Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.074274

ARTICLE

Safety-Aware Reinforcement Learning for Self-Healing Intrusion Detection in 5G-Enabled IoT Networks

Wajdan Al Malwi[1], Fatima Asiri[1], Nazik Alturki[2], Noha Alnazzawi[3], Dimitrios Kasimatis[4] and Nikolaos Pitropakis[5,*]

[1]College of Computer Science, Informatics and Computer Systems Department, King Khalid University, Abha, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, Saudi Arabia
[3]Computer Science and Engineering Department, Yanbu Industrial College, Royal Commission for Jubail and Yanbu, Yanbu, Saudi Arabia
[4]Blockpass ID Lab, Edinburgh Napier University, Edinburgh, UK
[5]Department of Information Technology, Cybersecurity and Computer Science, The American College of Greece, Athens, Greece

*Corresponding Author: Nikolaos Pitropakis. Email: npitropakis@acg.edu

Received: 07 October 2025; Accepted: 20 January 2026; Published: 12 March 2026

ABSTRACT: The expansion of 5G-enabled Internet of Things (IoT) networks, while enabling transformative applications, significantly increases the attack surface and necessitates security solutions that extend beyond traditional intrusion detection. Existing intrusion detection systems (IDSs) mainly operate in an open-loop manner, excelling at classification but lacking the ability for autonomous, safety-aware remediation. This gap is particularly critical in 5G environments, where manual intervention is too slow and naive automation can lead to severe service disruptions. To address this issue, we propose a novel Self-Healing Intrusion Detection System (SH-IDS) framework that develops a closed-loop cyber defense mechanism. The main technical contribution is the integration of a deep neural network-based threat detector, which offers uncertainty-quantified predictions, with a safety-aware reinforcement learning (RL) engine formulated as a Constrained Markov Decision Process (CMDP). The CMDP explicitly models operational safety as cost constraints, and a new runtime safety shield actively adjusts any unsafe action proposed by the RL agent to the nearest safe alternative, ensuring operational integrity. Additionally, we introduce a composite utility function for the comprehensive evaluation of the system. Empirical analysis on the 5G-NIDD dataset demonstrates the superior performance of our framework: the detector achieves 98.26% accuracy, while the safe RL agent learns effective mitigation policies. Importantly, the safety shield blocked up to 70 unsafe actions under strict constraints, and analysis of the learned Q-tables confirms that the agent internalizes safety, avoiding overly disruptive actions, such as isolating nodes for minor threats. The system also maintains high efficiency with a compact model size of 121.7 KB and sub-millisecond latency, confirming its practical deployability for real-time 5G-IoT security.

KEYWORDS: Cybersecurity; internet of things; intrusion detection; 5G/6G security; reinforcement learning

1 Introduction

The rapid deployment of 5G networks is reshaping the Internet of Things (IoT) ecosystem by enabling a broad spectrum of applications, including industrial automation, smart cities, and connected healthcare [1,2]. The distinctive features of 5G, including ultra-reliable low-latency communication (URLLC) and massive machine-type communication (mMTC), offer unprecedented connectivity and real-time data

exchange [3,4]. However, this expanded and heterogeneous environment also enlarges the attack surface, exposing critical IoT infrastructures to increasingly sophisticated cyber threats [5].

Conventional Intrusion Detection Systems (IDSs), although fundamental to network defense, are not adequately equipped to secure modern, highly dynamic 5G-enabled IoT environments [6,7]. Anomaly-based systems offer some adaptability; however, they often exhibit high false-positive rates and lack the contextual awareness needed for reliable threat interpretation [8]. Moreover, traditional IDSs operate in a reactive, open-loop manner, focusing primarily on detection and alerting while leaving mitigation and remediation to human operators [9]. In 5G settings where latency requirements are measured in milliseconds, any reliance on manual intervention results in unacceptable delays that allow attacks to evolve, propagate, or disrupt services before appropriate countermeasures can be executed [10].

Although recent intrusion detection studies designed for 5G networks have made notable progress in areas such as data preprocessing, feature engineering, and the use of advanced deep learning models, the vast majority remain predominantly detection-oriented [11]. These systems can identify malicious behavior. Still, they do not provide a mechanism for autonomously and safely translating detections into timely corrective actions, a capability crucial to latency-sensitive 5G-IoT infrastructures [12]. Consequently, even high-performing IDS solutions continue to rely on manual or semi-automated responses, which introduce delays and increase the likelihood that threats will propagate [13]. Existing automation efforts also rarely include explicit operational safety constraints, which raises the risk of overly disruptive remediation actions, such as isolating critical nodes in response to minor anomalies [14]. These limitations reveal a substantial gap in the current literature and emphasize the need for a practical self-healing IDS that integrates high-accuracy detection with intelligent, safety-aware remediation, capable of maintaining both effectiveness and service continuity.

To address these limitations, we propose a Self-Healing Intrusion Detection System (SH-IDS), a closed-loop cyber defense framework that combines advanced threat detection with safe, autonomous remediation. The system works in two stages. First, a deep neural network trained on the complexities of 5G network traffic functions as the detection module, delivering real-time, high-accuracy classification of intrusions. Second, the identified threat is sent to a Safe Reinforcement Learning (SRL) decision-making engine, structured as a Constrained Markov Decision Process (CMDP) [15]. The SRL agent learns an optimal policy that maps detected threats to specific healing actions, such as isolating compromised nodes, reducing malicious traffic, or patching vulnerable services, while adhering to explicit safety constraints that penalize disruptive responses. This guarantees both effective and stable autonomous operation.

The primary contributions of this article are summarized below:

1. A closed-loop self-healing intrusion detection architecture is introduced, integrating high-accuracy deep learning–based threat detection with autonomous remediation. This unified design contrasts with traditional IDS approaches that operate exclusively as detection mechanisms and lack integrated, continuous defense loops suitable for real-time 5G-IoT environments.
2. A safety-aware decision-making framework based on a Constrained Markov Decision Process (CMDP) is formulated, enabling reinforcement learning [16] to generate remediation strategies that simultaneously consider mitigation effectiveness and operational safety. This formulation addresses a notable limitation in existing IDS research, where automated responses rarely incorporate explicit safety constraints [16].
3. A runtime safety shield is incorporated to guarantee safety-compliant action selection during both training and deployment. The shield monitors proposed remediation actions, identifies those that violate predefined safety conditions, and projects them onto the nearest safe alternatives. Such mandatory

runtime enforcement is absent from current IDS and automated defense systems, which typically lack mechanisms to prevent operationally disruptive actions.

4. A detailed architectural integration of uncertainty-aware deep neural network detection, CMDP-driven safe reinforcement learning, and runtime safety shielding is presented. This combination of calibrated uncertainty estimation, constrained optimization, and safety-enforced action correction distinguishes the proposed SH-IDS from existing schemes that remain detection-centric, do not incorporate prediction uncertainty into decision making, or employ automated responses without verifiable safety guarantees.
5. The proposed framework achieves 98.26% detection accuracy, blocks up to 70 unsafe actions via the safety shield, and maintains operational efficiency with a 121.7 KB model and sub-millisecond latency, confirming its deployability in real-time 5G-IoT environments.

The remainder of the article is organized as follows. Section 2 presents an overview of the related studies. Section 3 describes the proposed architecture. Section 4 presents the experimental methodology and a discussion of the results. Section 5 concludes the research.

2 Related Work

The increasing deployment of 5G networks, which support critical infrastructure from smart grids to industrial IoT, has heightened the focus on robust cybersecurity measures. The unique design of 5G, defined by network softwarization, network slicing, and the growth of endpoints, creates a broad and complex attack surface. As a result, IDSs have adapted to these new challenges, with recent research using advanced artificial intelligence (AI) and machine learning (ML) techniques. This section reviews current IDS research specifically targeting 5G environments, emphasizing progress toward more adaptive, efficient, and specialized solutions.

Rani et al. [17] proposed a new Target Projection Regressed Gradient Convolutional Neural Network (TPRGCNN) for smart grid security, combining feature selection and classification to achieve high detection accuracy while reducing computational load. Similarly, Gurushanker et al. [18] emphasized the need for 5G-specific IDS by testing ML models on both TCP/IP flow data and Packet Forwarding Control Protocol (PFCP) signaling data, achieving high accuracy with TCP/IP data and highlighting the difficulty of detecting control-plane-specific attacks. These studies show a clear trend toward customizing detection models to suit the unique data features of 5G networks.

Recognizing the evolving threat landscape, several studies have added mechanisms for adaptability and robustness. Neha and Bhatia [19] presented an IDS framework using dynamic neural networks and adversarial training, enabling incremental learning to detect new attacks and resist data poisoning, a key vulnerability in systems that learn continuously. Building on robustness, Reis [20] introduced a hybrid AI-driven framework combining autoencoders, LSTMs, and CNNs to detect a broader range of spatial and temporal anomalies. This work also incorporated federated learning and edge AI, addressing scalability and data privacy issues in distributed smart city environments. These approaches represent a major advancement beyond static, signature-based detection.

Further specialization of the 5G core network has been a key area of innovation. Radoglou-Grammatikis et al. [21] developed 5GCIDS, an IDS specifically designed to protect the N4 interface between the Session Management Function (SMF) and User Plane Function (UPF). Their contribution includes a PFCP flow statistics generator and the integration of explainable AI (XAI) via TreeSHAP, providing crucial transparency for security analysts. In the area of decentralized and resource-efficient learning, Adjewa et al. [22] explored the use of an optimized BERT model within a federated learning framework. Their work demonstrates the viability of large language models for intrusion detection on resource-constrained edge devices while

maintaining data privacy and achieving high accuracy even after model compression. Mahmood et al. [23] applied different machine learning algorithms to 5G intrusion detection, finding that a Linear Regression algorithm could achieve high accuracy. This highlights that even traditional models can be effective for specific tasks in 5G security.

While the surveyed literature provides essential building blocks, including specialized feature engineering, adversarial robustness, hybrid model architectures, privacy-preserving federated learning, and core-network-specific detection, a significant gap remains. Most of these studies are primarily detection-focused. They excel at identifying malicious activity but lack an integrated, autonomous, and safety-aware remediation mechanism. Common limitations include the absence of end-to-end evaluations that address detection and mitigation together, insufficient consideration of the operational costs of automated responses, and a lack of explicit constraints to prevent service-disruptive actions. Naive automation, without formal safety guarantees, can cause serious service issues, such as the unnecessary isolation of critical nodes.

Our work directly bridges this gap by proposing a closed-loop SH-IDS. We integrate a high-accuracy deep neural network (DNN) detector, trained on the 5G-NIDD dataset, with a safety-aware reinforcement learning engine designed as a Constrained Markov Decision Process. This is complemented by a runtime safety shield. This combined approach ensures that effective threat mitigation always adheres to explicit operational safety constraints, making the system not only highly effective in detection but also trustworthy and cautious in autonomous actions.

3 The Proposed Self-Healing Intrusion Detection System

This section presents the mathematical foundation and operational workflow of the Proposed SH-IDS. This closed-loop cyber defense framework integrates high-precision threat detection with safe autonomous remediation for 5G-IoT networks. The workflow of the proposed framework is presented in Fig. 1. Table 1 presents the utilized notations and their description in this section.

3.1 System Overview and Operational Workflow

The SH-IDS operates through four integrated phases that form a complete cyber defense loop:

1. Threat Detection & Confidence Estimation: Raw network traffic and telemetry data represented as feature vectors $x \in \mathcal{X} \subseteq \mathbb{R}^d$ are continuously monitored and classified by a deep neural network. The detector not only predicts threat categories but also computes confidence scores for its predictions, enabling uncertainty-aware decision making.
2. State Representation & Context Enrichment: The detector's outputs (predicted labels and confidence scores) are fused with real-time contextual information about the network environment. This enriched state representation provides a comprehensive view of the current security situation and operational context.
3. Safe Autonomous Decision-Making: A Safety-Aware Reinforcement Learning agent, formulated as a Constrained Markov Decision Process, selects optimal remediation actions. A runtime safety shield intercepts each proposed action and enforces operational constraints by projecting unsafe actions to safe alternatives.
4. Remediation Execution & System Update: Approved healing actions are executed in the network environment, which transitions to a new state. The agent receives reward signals based on the effectiveness and safety of the taken action, completing the feedback loop for continuous learning.

This closed-loop architecture enables the system to autonomously adapt to evolving threats while maintaining operational stability through explicit safety constraints.

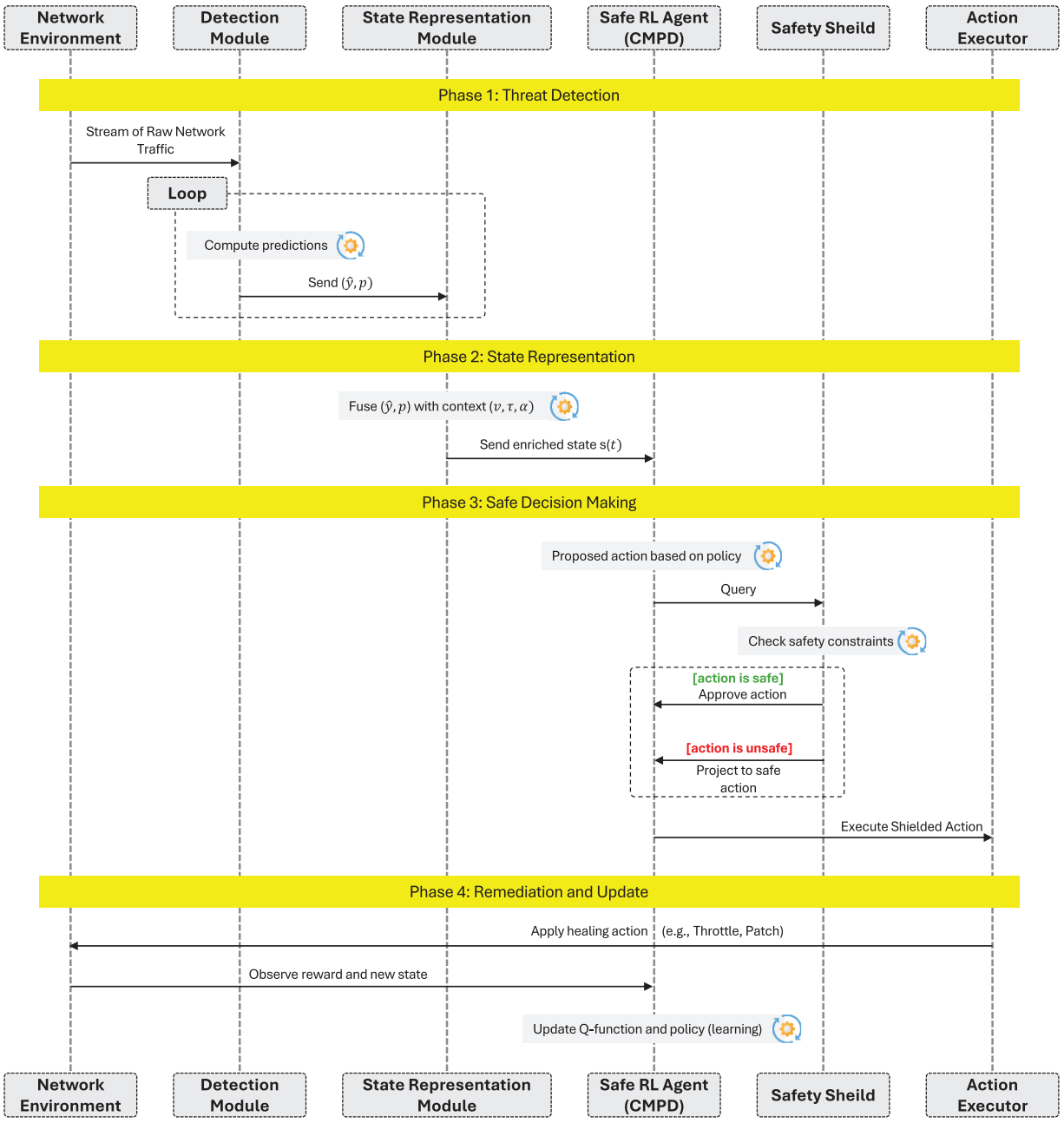

Figure 1: Workflow of the proposed SH-IDS architecture.

Table 1: List of symbols and notations.

Symbol	Description	Symbol	Description
x	Feature vector of raw network traffic	$\mathcal{X}$	Feature space ($\subseteq \mathbb{R}^d$)
d	Dimensionality of feature vectors	K	Number of threat classes
θ	Parameters of the deep neural network detector	$p_k(x;\theta)$	Predicted probability of class k given input x

(Continued)

Table 1 (continued)

Symbol	Description	Symbol	Description
$\hat{y}_t$	Predicted threat class at time t	ρ_t	Detection confidence score at time t
v_t	Node criticality level (categorical variable)	τ_t	Time-of-day indicator (peak/off-peak)
α_t	Recent alert density (continuous security metric)	s_t	State at time t
$\mathcal{S}$	State space	$\mathcal{A}$	Action space
r_t	Reward at time t	γ	Discount factor
C	Safety cost function	$B(s)$	Set of unsafe actions in state s
S_{shield}	Safety shield operator	$Q(s,a)$	Action-value estimate
a_{default}	Predefined fallback safe action (e.g., No Action)	π	Policy mapping states to action distributions
ϕ_i	Component i of the composite utility function	α_i	Weight for utility component ϕ_i

3.2 Threat Detection with Uncertainty Quantification

Threat Detection & Confidence Estimation: Raw network traffic and telemetry data represented as feature vectors $x \in \mathcal{X} \subseteq \mathbb{R}^d$ are continuously monitored and classified by a deep neural network. The detector not only predicts threat categories but also computes confidence scores for its predictions, enabling uncertainty-aware decision making.

$$f_\theta(x) = (p_0(x;\theta), \ldots, p_{K-1}(x;\theta)) \tag{1}$$

where each $p_k(x;\theta) = \Pr(Y = k \mid X = x;\theta)$ represents the estimated probability that input x belongs to class k, with the constraint that probabilities sum to unity: $\sum_{k=0}^{K-1} p_k(x;\theta) = 1$.

The predicted threat class is determined through maximum a posteriori estimation:

$$\hat{y}(x;\theta) = \arg\max_{k \in \mathcal{Y}} p_k(x;\theta) \tag{2}$$

Beyond classification, the module quantifies prediction confidence as the maximum softmax probability:

$$\rho(x;\theta) = \max_k p_k(x;\theta) \tag{3}$$

This confidence measure is crucial for downstream safety-aware decision making, particularly in handling uncertain detections. The detector is trained by minimizing the categorical cross-entropy loss over labeled training data:

$$\mathcal{L}_{CE}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=0}^{K-1} 1\{y_i = k\} \log p_k(x_i;\theta) \tag{4}$$

3.3 Enriched State Representation

The reinforcement learning agent operates on a comprehensive state representation that extends beyond simple threat classification to incorporate detection confidence and operational context [16]. This enriched

state enables context-aware remediation policies that consider both the security situation and network operational status.

The state at time t is formally defined as the tuple:

$$s_t = (\hat{y}_t, \rho_t, \nu_t, \tau_t, \alpha_t) \in \mathcal{S} \tag{5}$$

where the components are:

- $\hat{y}_t$: Predicted threat class from the detection module
- ρ_t : Detection confidence score $\in [0,1]$
- ν_t: Node criticality level (categorical variable)
- τ_t: Time-of-day indicator (peak/off-peak operation)
- α_t: Recent alert density (continuous security metric)

This multi-faceted state representation allows the agent to make nuanced decisions, such as avoiding aggressive remediation on critical nodes during peak hours or adjusting response strategies based on detection confidence levels.

3.4 Constrained Markov Decision Process Formulation

The core decision-making process is formally modeled as a Constrained Markov Decision Process (CMDP) to optimize threat mitigation while simultaneously respecting operational safety constraints. The CMDP is defined as:

$$M = (S, A, T, r, \gamma, C) \tag{6}$$

where S is the state space, A the action space, T the transition dynamics, r the reward, γ the discount factor, and C the safety costs.

3.4.1 Reward Function with Uncertainty-Aware Expectation

The reward mechanism explicitly accounts for detection uncertainty to prevent overreaction to potentially false positives. The immediate reward is computed as an expectation over possible true states, weighted by the detector's confidence distribution:

For an action a_t taken in state s_t, the expected reward is:

$$r_t = \mathbb{E}_{Y \sim \mathbf{p}(x_t;\theta)} \left[r\left(s_t, a_t, Y\right) \right] = \sum_{k=0}^{k-1} p_k\left(x_t;\theta\right) r\left(g\left(k, \rho_t, c_t\right), a_t\right), \tag{7}$$

where $g\left(k, \rho_t, c_t\right)$ maps the predicted class, confidence, and context variables c_t into the evaluation state.

The reward function decomposes into three components:

$$r(s, a, Y) = R_{\text{mit}}(s, a, Y) - C_{\text{op}}(s, a) - \lambda C_{\text{safe}}(s, a) \tag{8}$$

where R_{mit} rewards mitigation success, C_{op} penalizes resource and service impact, C_{safe} penalizes unsafe actions, and $\lambda > 0$ tunes safety-aggressiveness.

3.4.2 Safety Shield with Fallback Guarantee

The safety shield constitutes a runtime enforcement mechanism that guarantees operational constraints are never violated. Define the set of unsafe actions in state s as:

$$B(s) = \{a \in A \mid a \text{ violates safety constraints in state } s\}. \tag{9}$$

The shielded action operator ensures safe action selection:

$$S_{\text{shield}}(s,a) = \begin{cases} a, & a \notin B(s), \\ \arg\max_{a' \in A \setminus B(s)} Q(s,a'), & B(s) \neq A, a \in B(s), \\ a_{\text{default}}, & B(s) = A, \end{cases} \tag{10}$$

where $Q(s,a)$ is the action-value estimate, and a_{default} is a predefined minimally disruptive fallback action (e.g., No Action or Throttling). This guarantees system safety even in degenerate cases.

3.4.3 Constrained Optimization Objective

The learning objective is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes expected cumulative rewards subject to safety constraints:

$$\max_{\pi} J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{11}$$

while satisfying safety requirements expressed as expected cost constraints:

$$J_{C_i}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t)\right] \leq d_i, \quad i = 1, \ldots, L \tag{12}$$

3.5 Safe Q-Learning with Action Projection

The Q-learning algorithm is augmented with the safety shield to ensure constraint satisfaction throughout the learning process. The Q-function update incorporates shielded next-state actions:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \delta_t \tag{13}$$

where the temporal difference error is computed as:

$$\delta_t = r_t + \gamma \max_{a'} Q(s_{t+1}, \mathcal{S}_{\text{shield}}(s_{t+1}, a')) - Q(s_t, a_t) \tag{14}$$

The resulting optimal safe policy naturally integrates the safety mechanism:

$$\pi^*(s) = \mathcal{S}_{\text{shield}}\left(s, \arg\max_{a \in \mathcal{A}} Q^*(s,a)\right) \tag{15}$$

This formulation guarantees that the deployed policy is both performance-optimized and safe by construction.

3.6 Composite Performance Metric

Given the multi-dimensional nature of system requirements in 5G-IoT environments, we evaluate SH-IDS performance using a composite utility function that balances competing objectives:

$$U(\theta, \pi) = \sum_{i=1}^{6} \alpha_i \cdot \phi_i(\theta, \pi) \tag{16}$$

The utility components represent key performance indicators:

- ϕ_1: Detection accuracy A—Measures classification performance
- ϕ_2: Mitigation gain $\bar{G}$—Quantifies remediation effectiveness
- ϕ_3: Negative model size—$S(\theta)$—Encourages lightweight detection
- ϕ_4: Negative prediction latency—L_{pred}—Ensures real-time operation
- ϕ_5: Negative decision latency—L_{decide}—Maintains responsive control
- ϕ_6: Negative comm. payload—B_{comm}—Reduces network overhead

The weights $\alpha_i \geq 0$ are configurable parameters that allow stakeholders to prioritize objectives based on specific deployment constraints and operational requirements.

3.7 Training Pipeline of the Proposed SH-IDS

The SH-IDS training pipeline employs a structured three-phase approach that progressively builds from supervised detection learning to safe autonomous decision-making. The pipeline ensures robust performance while maintaining operational safety throughout the training process.

1. Phase 1: Detector Pre-training phase presented in Algorithm 1 establishes the foundation by training a deep neural network for accurate threat classification with calibrated confidence estimates. This phase uses historical network data to learn discriminative patterns between normal and malicious traffic, employing standard deep learning techniques with validation-based early stopping.
2. Phase 2: Safe RL Policy Learning phase illustrated in Algorithm 2 focuses on learning optimal remediation policies within safety constraints. The pre-trained detector from Phase 1 provides state representations, while the safety shield ensures all exploratory actions comply with operational limits. This phase utilizes experience replay and target networks to facilitate stable learning.
3. Phase 3: Joint Fine-tuning phase presented in Algorithm 3 performs iterative refinement of both detection and remediation components based on the composite utility metric. This phase addresses performance bottlenecks and adapts the system to challenging edge cases, ensuring balanced optimization across all system objectives.

The pipeline incorporates curriculum learning, beginning with basic threat scenarios and progressively introducing complex attack patterns. Safety constraints are enforced at every phase, making the training process suitable for critical infrastructure applications.

Algorithm 1: Detector pre-training phase

Require: Labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, validation split ratio α, maximum epochs E_{det}, learning rate η
Ensure: Trained detector parameters θ^* (best validation performance)
1: Initialize detector network f_θ with random weights θ_0
2: Split $\mathcal{D}$ into training and validation sets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ using ratio α
3: Initialize best validation accuracy $acc_{\text{best}} \leftarrow 0$

(Continued)

Algorithm 1 (continued)

Initialize early stopping counter $p \leftarrow 0$
for epoch = 1 to E_{det} **do**
  Shuffle $\mathcal{D}_{\text{train}}$
  for each mini-batch (X_b, Y_b) in $\mathcal{D}_{\text{train}}$ **do**
    Compute class probability predictions: $\mathbf{p} = f_\theta(X_b)$
    Compute cross-entropy loss: $\mathcal{L} = \mathcal{L}_{CE}(\mathbf{p}, Y_b)$
    Update detector parameters via gradient descent: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
  end for
  Evaluate detector on validation set:
    $acc_{\text{val}} = \text{Accuracy}(f_\theta(\mathcal{D}_{\text{val}}))$
  if $acc_{\text{val}} > acc_{\text{best}}$ **then**
    Update best validation accuracy: $acc_{\text{best}} \leftarrow acc_{\text{val}}$
    Store optimal parameters: $\theta^* \leftarrow \theta$
    Reset early stopping counter: $p \leftarrow 0$
  else
    Increment early stopping counter: $p \leftarrow p + 1$
  end if
  if early stopping criterion satisfied (e.g., $p \geq p_{\max}$) **then**
    Terminate training successfully
    break
  end if
end for
return θ^* ▷ Return best-performing detector parameters after successful convergence

Algorithm 2: Safe RL policy learning phase

Require: Trained detector f_{θ^*}, safety constraints $\mathcal{C}$, discount factor γ learning rate η, exploration parameters $(\epsilon, \epsilon_{\min}, \epsilon_{\text{decay}})$, maximum episodes E_{RL}, maximum steps per episode $T_{\max}$
Ensure: Learned Q-function Q_ϕ and runtime safety shield $\mathcal{S}_{\text{shield}}$
Initialize Q-network Q_ϕ with random parameters ϕ
Initialize target Q-network $Q_{\phi'} \leftarrow Q_\phi$
Initialize replay buffer $\mathcal{B}$
Initialize exploration rate $\epsilon \leftarrow 1.0$
Configure safety shield $\mathcal{S}_{\text{shield}}$ using constraints $\mathcal{C}$
for episode = 1 to E_{RL} **do**
  Sample initial network observation x_0
  Encode initial state using detector: $s_0 \leftarrow f_{\theta^*}(x_0)$
  for t = 0 to $T_{\max}$ **do**
    Observe current state s_t
    if random uniform $< \epsilon$ **then**
      Select exploratory action: $a_t \leftarrow \text{RandomAction}()$
    else
      Select greedy action: $a_t \leftarrow \arg\max_a Q_\phi(s_t, a)$
    end if

(Continued)

Algorithm 2 (continued)

Enforce safety via runtime shield:
  $\tilde{a}_t \leftarrow \mathcal{S}_{\text{shield}}(s_t, a_t)$
Execute $\tilde{a}_t$, observe reward r_t and next observation x_{t+1}
Encode next state: $s_{t+1} \leftarrow f_{\theta^*}(x_{t+1})$
Store transition $(s_t, \tilde{a}_t, r_t, s_{t+1})$ in replay buffer $\mathcal{B}$
if $|\mathcal{B}| \geq B_{\min}$ **then**
  Sample mini-batch $(s_j, a_j, r_j, s_{j+1}) \sim \mathcal{B}$
  Compute target Q-values:
    $y_j = r_j + \gamma \max_{a'} Q_{\phi'}(s_{j+1}, \mathcal{S}_{\text{shield}}(s_{j+1}, a'))$
  Update Q-network by minimizing:
    $\sum_j (y_j - Q_\phi(s_j, a_j))^2$
  if $t \bmod f_{\text{update}} = 0$ **then**
    Soft update target network:
      $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
  end if
end if
Update state: $s_t \leftarrow s_{t+1}$
end for
Decay exploration rate:
  $\epsilon \leftarrow \max(\epsilon_{\min}, \epsilon \cdot \epsilon_{\text{decay}})$
end for
return $Q_\phi, \mathcal{S}_{\text{shield}}$ ▷ Terminate successfully after completing E_{RL} episodes or policy convergence

Algorithm 3: Joint fine-tuning phase

Require: Trained detector f_θ, learned Q-function Q_ϕ, runtime safety shield $\mathcal{S}_{\text{shield}}$, weights $\{\alpha_i\}_{i=1}^6$, fine-tuning learning rate η_{ft}, maximum fine-tuning iterations I_{ft}, utility convergence threshold $U_{\text{threshold}}$

Ensure: Optimized detector parameters θ^{**} and Q-function parameters ϕ^*

Initialize fine-tuning dataset $\mathcal{D}_{\text{ft}} \leftarrow \varnothing$
Initialize best utility score $U_{\text{best}} \leftarrow -\infty$
Define safety-compliant policy:
  $\pi(s) = \mathcal{S}_{\text{shield}}(s, \arg\max_a Q_\phi(s, a))$
for iteration = 1 to I_{ft} **do**
  Interact with environment using policy π to collect challenging scenarios
  Augment fine-tuning dataset $\mathcal{D}_{\text{ft}}$ with newly observed samples
  Evaluate composite system utility:
    $U_{\text{current}} = \sum_{i=1}^6 \alpha_i \cdot \phi_i(\theta, \pi)$
  if $U_{\text{current}} < U_{\text{threshold}}$ **then**
    Fine-tune detector using $\mathcal{D}_{\text{ft}}$:
      $\theta \leftarrow \theta - \eta_{\text{ft}} \nabla_\theta \mathcal{L}_{CE}(\mathcal{D}_{\text{ft}})$
    Update Q-function Q_ϕ using updated detector state representations
    Re-evaluate system utility after fine-tuning:
      $U_{\text{new}} = \sum_{i=1}^6 \alpha_i \cdot \phi_i(\theta, \pi)$

(Continued)

Algorithm 3 (continued)

```
        if U_new > U_best then
            Accept improvements:
                θ** ← θ, φ* ← φ
            Update best utility: U_best ← U_new
        else
            Revert to previously stored parameters        ▷ Reject performance degradation
        end if
    else
        Terminate fine-tuning successfully                ▷ Utility convergence achieved
        break
    end if
end for
return θ**, φ* ▷ Return jointly optimized parameters after convergence or maximum iterations
```

4 Experiments and Results

This section offers a thorough empirical evaluation of the proposed SH-IDS. We begin by describing the experimental setup and dataset, then detail the specific configurations tested. Subsequently, we analyze the system's performance, emphasizing threat-detection accuracy, the safety-aware reinforcement learning agent's effectiveness, and the computational efficiency of the framework.

4.1 Experimental Setup

The experiments were performed on the Google Colab Pro platform, using its cloud-based computing resources, including an NVIDIA T4 Tensor Core GPU and high-RAM capabilities. This environment provides the necessary processing power to efficiently train and evaluate deep learning and reinforcement learning models at the core of our proposed architecture.

To validate the SH-IDS in a relevant and contemporary network environment, we used the 5G-NIDD dataset [24]. This dataset is specifically designed for 5G networks. It includes a realistic mix of normal traffic and various modern cyberattacks, such as flooding, scanning, and Denial-of-Service (DoS). Its use ensures that our evaluation captures the challenges and complexities of securing next-generation IoT networks, making it an ideal benchmark for assessing the proposed system's performance.

Before training, the 5G-NIDD dataset underwent structured preprocessing to ensure high-quality, unbiased training data. First, all numerical features were scaled to a [0, 1] range using min–max normalization to prevent large-magnitude attributes from dominating during neural network training. Second, class imbalance, particularly between benign and minority attack types, was addressed through stratified sampling and moderate oversampling of underrepresented classes, thereby maintaining the statistical integrity of the traffic distribution. Third, incomplete or noisy samples with missing feature values or corrupted network fields were removed after threshold-based validation checks. Finally, all categorical attributes were encoded with one-hot encoding and combined into fixed-length feature vectors. These preprocessing steps made sure that the detector training in Algorithm 1 used clean, balanced, and normalized data, reducing bias and improving the model's robustness and generalizability.

4.2 Performance Evaluation Metrics

To rigorously assess the proposed SH-IDS, we employ a multidimensional evaluation framework that covers threat detection accuracy, reinforcement learning efficacy, and system efficiency. The utilized performance assessment metrics are summarized in the following.

4.2.1 Detection Performance Metrics

The detection module is evaluated using standard classification metrics derived from the confusion matrix, where *TP*, *TN*, *FP*, and *FN* represent True Positives, True Negatives, False Positives, and False Negatives, respectively:

- Accuracy: The proportion of total instances correctly classified.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

- Precision: The ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{18}$$

- Recall: The ratio of correctly predicted positive observations to all observations in the actual class.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{19}$$

- F1-Score: The harmonic mean of Precision and Recall, providing a balanced measure for imbalanced datasets.

$$F1 - \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

4.2.2 Self-Healing and RL Metrics

The performance of the autonomous remediation engine is quantified by the cumulative reward and the safety shield's intervention rate:

- Expected Cumulative Reward: Measures the agent's ability to maximize mitigation success while minimizing operational costs over time t.

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right)\right] \tag{21}$$

- Safety Shield Block Count: The total number of proposed actions $a \in B(s)$ that were intercepted and projected to safe alternatives $a' \in A \backslash B(s)$ to prevent service disruption.

4.2.3 Efficiency and Composite Utility

To ensure deployability in 5G-IoT environments, we measure latency and model size, integrated into a composite utility function $\mathcal{U}$:

- Inference Latency (L): The total time required for threat prediction $\left(L_{\text{pred}}\right)$ and decision making (L_{decide}).

- Composite Utility: A weighted sum of performance indicators (ϕ_i) and their respective importance weights (α_i):

$$\mathcal{U}(\theta, \pi) = \sum_{i=1}^{6} \alpha_i \cdot \phi_i(\theta, \pi) \tag{22}$$

4.3 Experimental Configurations

To rigorously evaluate the performance and behavior of the SH-IDS, we established seven distinct experimental configurations. Each configuration modifies a specific component or hyperparameter of the system to isolate its impact on the overall performance.

The configurations are described as follows:

1. Baseline SH-IDS: The standard implementation of our proposed architecture, featuring a standard deep neural network for detection and a Q-learning agent with baseline learning parameters and an active safety shield.
2. Complex Detector: This configuration replaces the standard detector with a more complex neural network containing additional hidden layers and neurons to assess the trade-off between model complexity and detection performance.
3. High Learning Rate RL: The reinforcement learning agent's learning rate is significantly increased to observe its effect on the speed and stability of policy convergence.
4. Fast Epsilon Decay RL: The exploration-exploitation trade-off in the RL agent is shifted by implementing a faster decay for the epsilon parameter, encouraging the agent to exploit known effective actions more quickly.
5. Extended RL Training: The RL agent undergoes a significantly longer training period (5000 episodes) to determine if more extensive training leads to a more optimal and stable remediation policy.
6. High-Safety Constraints: The safety shield is configured with a broader set of critical nodes, making the operational constraints more stringent to evaluate the system's performance under heightened safety requirements.
7. No Safety Shield (Baseline): This configuration deactivates the safety shield entirely, allowing the RL agent to select any action without constraint. It serves as a crucial baseline to demonstrate the value and impact of the safety-aware decision-making component.

4.4 Performance Evaluation of the Threat Detection Module

Fig. 2 illustrates the training and validation results for the seven experimental configurations. In all cases, training and validation accuracies increase rapidly and typically stabilize within the first 20–30 epochs. This indicates that the proposed model can efficiently learn useful patterns. Meanwhile, the training and validation loss consistently decline and stabilize at low values, with no significant differences between the two curves.

The close match between training and validation performance across all configurations, including the Baseline SH-IDS, Complex Detector, and High-Safety Constraints, suggests that the preprocessing steps and stratified sampling were effective in reducing overfitting. In Configuration 2 (Complex Detector), although the model architecture is more complex, the validation loss remains stable, indicating that the model generalizes well. In comparison, the "No Safety Shield" baseline exhibits slightly greater variation in validation loss (Fig. 2, Plot 7), suggesting that safety constraints also improve training stability. A detailed discussion of system performance is presented below.

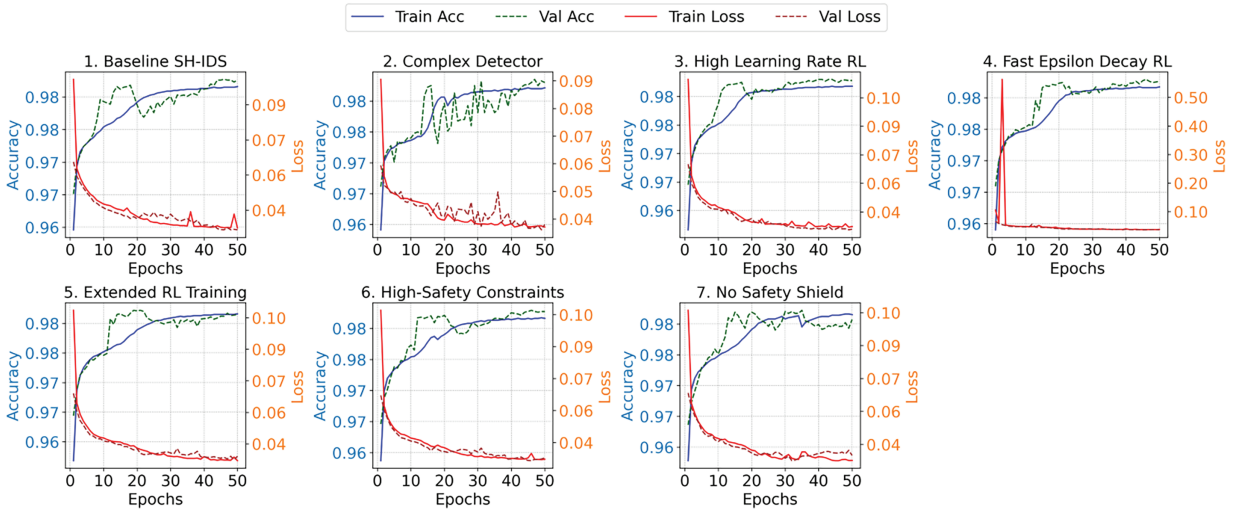

Figure 2: Training and validation curves under different experimental configurations.

The attack detection performance of the proposed design is presented in Fig. 3. The results consistently demonstrate high performance across all configurations, underscoring the robustness of the chosen approach in identifying threats within the 5G-NIDD dataset.

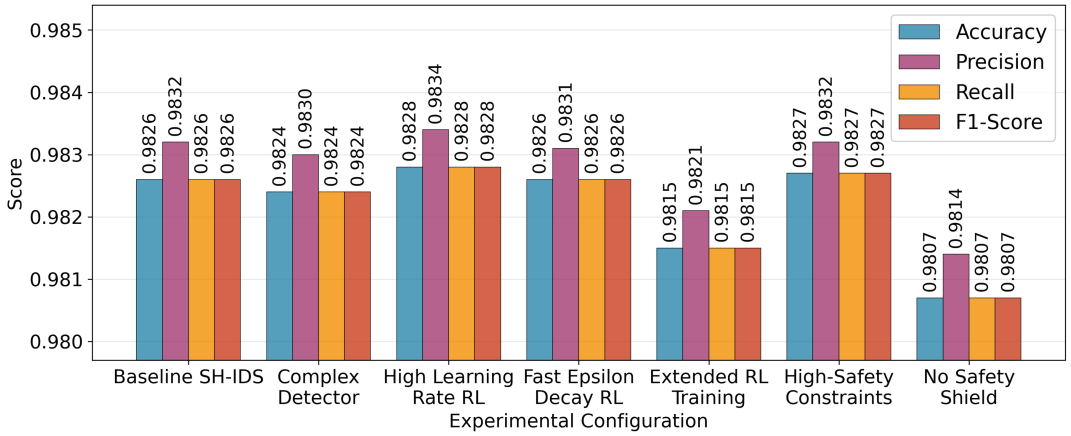

Figure 3: Attack detection performance under different experimental configurations.

The Baseline SH-IDS setup achieved an outstanding accuracy of 98.26%, with high precision (98.32%), recall (98.26%), and F1-Score (98.26%). These metrics indicate that the detector is highly effective in accurately identifying both normal and malicious traffic, with minimal false positives and false negatives. Interestingly, the Complex Detector setup did not produce a significant performance gain and even resulted in a slight decrease in accuracy to 98.24%. This suggests that the standard model architecture is already sufficiently complex to detect patterns in the data, and adding further complexity does not offer additional advantages, while increasing computational demands. The other setups, mainly changing the RL agent's parameters, showed little variation in detection performance, confirming that the effectiveness of the detection module does not depend on the configuration of the remediation module. The lowest accuracy recorded was 98.07% in the "No Safety Shield" experiment, which remains a remarkably high result.

4.5 Analysis of the RL-Based Self-Healing Module

The performance of the reinforcement learning agent and the safety shield is summarized in Table 2. This table provides critical insights into the effectiveness of the autonomous decision-making process. The two key metrics are the average reward obtained by the RL agent and the number of times the safety shield intervened to block a potentially disruptive action.

Table 2: RL agent and safety shield performance.

Experimental Configuration	Avg RL Reward	Safety Shield Blocks
Baseline SH-IDS	−3.195	31
Complex Detector	−2.820	28
High Learning Rate RL	−3.045	27
Fast Epsilon Decay RL	2.895	11
Extended RL Training	2.164	44
High-Safety Constraints	−7.080	70
No Safety Shield (Baseline)	−0.230	0

A key finding is the positive average reward achieved by the Fast Epsilon Decay RL (2.895) and Extended RL Training (2.164) configurations. This demonstrates that the agent has successfully learned a policy that not only effectively reduces threats but also minimizes unnecessary or costly actions, resulting in positive rewards over time. In contrast, the Baseline SH-IDS received a negative average reward (−3.195), indicating that its policy was not fully optimized within the 1000-episode training period.

The importance of the safety mechanism is clearly shown by comparing the Baseline SH-IDS with the No Safety Shield setup. While the "No Safety Shield" agent achieved a nearly neutral average reward (−0.230), it did so without any safety interventions. This means the agent was free to take actions, such as isolating critical nodes, that the shield would have blocked. The Baseline SH-IDS, however, blocked 31 such unsafe actions. This difference is even more evident in the High-Safety Constraints scenario, where the shield blocked 70 actions, resulting in a significantly lower average reward (−7.080) due to the penalties for attempting unsafe operations. This shows a clear trade-off: the safety shield effectively enforces operational limits but at the expense of immediate reward, favouring system stability over aggressive and potentially harmful fixes.

4.6 Computational and Communication Efficiency

Table 3 presents the efficiency metrics of the proposed SH-IDS, which are critical for its practical deployment in resource-constrained IoT environments. The Baseline SH-IDS model is relatively lightweight, with 8265 parameters and a serialized model size of approximately 121.7 KB. This small footprint makes it suitable for deployment on edge devices.

As expected, the Complex Detector has a much larger footprint, with 22,409 parameters and a model size of 292.2 KB. Since it showed no noticeable improvement in detection accuracy, the baseline model offers a better balance between performance and resource use. Latency is a key metric for 5G networks. The average prediction latency across all configurations is very low, ranging from 0.0466 to 0.0554 ms. This almost instant prediction capability meets the stringent low-latency requirements of 5G URLLC applications, enabling the SH-IDS to perform real-time detection without introducing significant network delay. The training times are also reasonable, with the baseline model training taking about 431 s on the specified hardware.

Table 3: Computational & communication efficiency.

Experimental Configuration	Model Parameters	Model Size (KB)	Avg Prediction Latency (ms)	Training Time (s)
Baseline SH-IDS	8265	121.696	0.0554	430.98
Complex Detector	22,409	292.191	0.0517	442.82
High Learning Rate RL	8265	121.696	0.0467	427.12
Fast Epsilon Decay RL	8265	121.698	0.0487	427.76
Extended RL Training	8265	121.698	0.0473	425.85
High-Safety Constraints	8265	121.698	0.0472	424.49
No Safety Shield (Baseline)	8265	121.698	0.0466	421.46

It is important to note that the reinforcement learning loop and safety shield introduce only constant per-node overhead, making the SH-IDS scalable for large, multi-node 5G-IoT deployments. The RL agent operates on a compact 5-dimensional state and a fixed, small action set, resulting in a single forward pass through the Q-network and a constant-time safety-shield check per decision. The shield itself performs only a bounded filtering step over the available actions, requiring no inter-node coordination. Therefore, the combined RL + shield inference latency (measured at 0.047–0.055 ms in Table 3) remains effectively unchanged as the number of protected nodes grows. Since decisions are made locally on each edge device without centralized synchronization, both computation and communication overhead scale linearly with the number of nodes, making it suitable for dense 5G-IoT environments.

4.7 Analysis of Learned Remediation Policies

The final set of tables (Tables 4–10) provides the learned Q-tables for each experimental configuration, revealing the specific action policies developed by the RL agent. A comparative analysis of these tables illuminates how different parameters and constraints shape the agent's decision-making logic.

Table 4: Q-Table for experiment: Baseline SH-IDS.

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	13.66	33.38	34.31	35.17	36.19	63.49
HTTPFlood	11.41	10.15	18.44	12.15	20.37	56.86
ICMPFlood	2.89	3.27	0.00	0.00	0.00	0.00
SYNFlood	2.21	11.07	6.23	1.78	0.00	0.00
SYNScan	2.19	7.93	6.86	2.67	9.00	16.61
SlowrateDoS	3.15	5.88	23.18	0.46	15.33	49.08
TCPConnectScan	1.90	7.88	18.88	8.45	6.96	21.08
UDPFlood	−4.53	33.97	38.02	40.56	36.35	64.00
UDPScan	−9.30	0.01	22.78	0.00	3.85	0.00

Table 5: Q-TABLE for experiment: Complex Detector.

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	22.87	39.14	35.14	43.34	39.40	66.47
HTTPFlood	8.96	19.97	27.87	11.65	24.58	58.06
ICMPFlood	2.38	0.00	0.00	0.00	0.00	0.00
SYNFlood	−15.25	6.27	0.00	0.45	1.94	0.00
SYNScan	−0.08	−0.81	0.00	0.00	19.05	0.55
SlowrateDoS	0.79	17.80	16.61	9.53	11.66	53.13
TCPConnectScan	−5.14	5.58	18.90	4.07	5.37	14.10
UDPFlood	22.07	42.74	37.63	39.63	34.44	66.08
UDPScan	4.04	0.00	5.24	2.14	14.91	0.00

Table 6: Q-Table for experiment: High Learning Rate RL.

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	87.88	88.05	87.68	88.69	86.21	99.03
HTTPFlood	84.07	72.06	85.11	86.95	87.54	98.95
ICMPFlood	0.00	0.00	0.00	0.00	0.00	0.00
SYNFlood	9.27	64.77	43.79	0.00	8.12	0.00
SYNScan	41.03	29.01	88.25	59.25	28.60	51.90
SlowrateDoS	−22.93	73.56	72.10	66.59	74.48	97.40
TCPConnectScan	0.00	0.00	48.58	25.91	60.66	98.59
UDPFlood	76.83	88.38	83.34	88.02	88.21	98.97
UDPScan	62.43	53.86	0.00	13.64	44.45	78.02

Table 7: Q-Table for experiment: Fast Epsilon Decay RL.

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	−22.74	−0.04	−1.14	3.38	5.20	84.35
HTTPFlood	0.38	2.62	0.00	−0.18	−0.36	81.83
ICMPFlood	6.18	0.00	0.00	0.00	0.00	0.00
SYNFlood	−9.31	46.18	0.00	0.00	0.00	0.00
SYNScan	−1.40	48.78	0.00	0.00	0.00	0.00
SlowrateDoS	−0.13	2.63	0.00	0.49	0.87	77.68
TCPConnectScan	0.00	0.00	−0.50	0.00	0.00	51.95
UDPFlood	−36.38	9.72	−0.55	1.93	3.01	84.40
UDPScan	0.00	0.00	0.00	0.00	0.00	34.80

Table 8: Q-Table for experiment: Extended RL Training.

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	65.16	78.14	76.90	81.53	82.21	99.97
HTTPFlood	18.29	38.61	67.44	54.47	47.98	99.97
ICMPFlood	25.59	0.00	0.00	0.00	−0.13	0.00
SYNFlood	0.00	−0.31	0.41	6.83	0.19	90.98
SYNScan	10.97	−0.59	89.45	−0.26	6.87	19.34
SlowrateDoS	12.30	52.75	25.05	23.24	11.09	99.97
TCPConnectScan	−8.50	0.00	10.41	18.08	4.12	99.50
UDPFlood	62.24	76.68	78.65	80.64	77.04	99.97
UDPScan	12.10	0.00	0.00	89.23	5.73	16.55

Table 9: Q-Table for experiment: High-Safety Constraints.

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	−14.37	30.51	39.48	31.41	33.64	62.93
HTTPFlood	−25.23	17.36	19.51	17.33	29.71	60.10
ICMPFlood	0.00	0.00	0.00	0.00	0.00	0.00
SYNFlood	−9.22	−0.27	0.00	0.00	12.76	0.00
SYNScan	−19.34	1.88	7.51	0.51	4.87	28.74
SlowrateDoS	−14.19	3.64	3.72	13.86	8.04	38.64
TCPConnectScan	0.00	1.07	24.63	6.28	1.28	1.95
UDPFlood	−33.55	30.81	35.03	33.12	35.12	62.91
UDPScan	−6.03	16.35	0.00	9.18	0.00	4.62

Table 10: Q-Table for experiment: No Safety Shield (Baseline).

Attack Type	Isolate Node	Throttle Traffic	Patch Service	Rotate Credentials	Rollback Deployment	No Action
Benign	35.38	36.68	33.50	35.60	27.82	59.23
HTTPFlood	16.82	18.31	18.71	23.96	15.09	56.80
ICMPFlood	0.00	0.00	0.00	0.00	4.43	0.00
SYNFlood	0.62	0.00	0.00	0.00	13.60	0.00
SYNScan	9.94	9.01	3.71	0.70	5.53	28.44
SlowrateDoS	11.80	9.06	4.49	5.54	4.03	48.91
TCPConnectScan	−0.45	3.62	7.01	1.49	3.92	19.76
UDPFlood	34.15	37.41	34.34	33.74	29.45	58.91
UDPScan	24.79	2.83	2.50	0.00	0.91	8.06

The Q-table for the Baseline SH-IDS, as demonstrated in Table 4, reveals a nuanced but evolving policy. For 'Benign' traffic, the "No Action" choice has the highest Q-value (63.49), indicating that the agent has correctly learned to avoid intervention during normal operation. For active threats like "UDPFlood", it assigned high values to "Throttle Traffic" (33.97) and "Patch Service" (38.02), demonstrating a basic ability to match threats to suitable countermeasures. The policy learned by the agent with the Complex Detector shown in Table 5 is generally similar, indicating the detection model's complexity had little impact on the agent's core strategy.

The impact of RL hyperparameters is clear in Tables 6–8. The High Learning Rate RL (Table 6) agent exhibits significantly inflated Q-values (e.g., 99.03 for "No Action" on "Benign"), indicating rapid and potentially unstable policy updates that did not result in a higher average reward. In contrast, the Fast Epsilon Decay RL (Table 7) demonstrates a more focused and converged policy. The Q-value for "No Action" on 'Benign' traffic is much higher (84.35) than for any other action in that state, showing a strong learned preference. Similarly, the agent in the Extended RL Training (Table 6) scenario developed a highly confident policy, with Q-values nearing 100 for optimal actions. This shows that both faster exploitation and longer training can help guide the agent to a more effective and stable policy than the baseline.

The most important insights come from comparing different safety constraint setups. The Q-table for High-Safety Constraints (Table 9) shows the agent learning to be cautious. For almost all attack types, the Q-values for "Isolate Node" are negative, indicating the strong penalties from the safety shield for trying this action on the expanded set of critical nodes. The agent has clearly learned to prefer less disruptive actions. This differs significantly from the No Safety Shield Q-table (Table 10). Without safety constraints, the agent assigns a relatively high Q-value (35.38) to "Isolate Node" for "Benign" traffic. In a live system, implementing this policy could cause significant service disruptions. This strongly confirms the need for the safety shield to guide the agent away from dangerous actions and toward a stable, reliable operation.

4.8 Performance Comparison with the State-of-the-Art

The analysis shown in Table 11 highlights the key operational benefits of the proposed SH-IDS compared to existing 5G-specific intrusion detection systems. While recent studies such as [23] and [17] have achieved high detection rates of up to 99.99% and 96.87%, respectively, these solutions primarily operate as open-loop systems. They are good at identifying threats but cannot automatically fix them, leaving the network exposed during the critical period between detection and manual response. When automated mitigation is used, as in [20], it relies on static, rule-based responses rather than adaptive learning and, importantly, lacks safety measures to prevent service interruptions. In contrast, the proposed SH-IDS achieves 98.26% detection accuracy and successfully completes the defense process with a Safe RL agent. This shift from passive monitoring to active, safety-conscious defense marks a significant step forward, filling the industry gap where automated responses have generally lacked the operational safety guarantees that our CMDP framework provides.

Furthermore, the proposed framework is better suited to resource-constrained 5G-IoT environments than current top models, especially in terms of computational efficiency. As shown in Table 11, complex hybrid models like those in [20] and federated approaches like [22] incur significant resource burdens, with model sizes of 72.9 and 30.2 MB and inference latencies of 287 and 450 ms, respectively. In contrast, the SH-IDS has an ultra-compact size of about 121.7 KB and delivers a combined prediction and decision time of roughly 0.055 ms. This efficiency, along with the unique integration of a runtime safety shield that ensures operational integrity, demonstrates that the proposed system offers a more practical, scalable, and safe solution for real-time 5G security than current detection-focused or resource-intensive alternatives.

Table 11: Performance comparison with the state-of-the-art.

Reference	Maximum Detection Accuracy (%)	Autonomous Remediation	Safety-Aware Decision Making	Model Size (KB)	Inference Latency (ms)
[17]	96.87%	No (Detection-only approach)	No (No operational safety constraints mentioned)	Not Reported	Not Reported
[19]	82.33%	No (Focus on robust detection and incremental learning)	No (Focus on adversarial robustness, not operational safety)	Not Reported	Not Reported
[20]	96.8% (F1-Score)	Yes (Rule-based autonomous mitigation)	No (No explicit safety constraints to prevent service disruption)	~72,900 KB (71.2 MB on Raspberry Pi)	287 ms (Median inference time)
[18]	97%	No (Focus is on detection and classification)	No (No mention of operational safety constraints)	Not Reported	Not Reported
[21]	85.5%, 64.1%	No (Focus is on detection and explainability)	No (No mention of operational safety constraints)	Not Reported	Not Reported
[22]	97.79%, 97.12%	No (Focus is on privacy-preserving detection)	No (No mention of operational safety constraints)	~30,200 KB (30.2 MB after quantization)	450 ms
[23]	99.99%	No (Focus is on detection accuracy)	No (No mention of operational safety constraints)	Not Reported	Not Reported
Proposed Scheme	98.26%	Yes (Safe RL agent formulates remediation policies)	Yes (CMDP formulation + Runtime Safety Shield)	~121.7 KB	~0.055 ms (Prediction + Decision)

5 Conclusion

This paper presented a closed-loop, self-healing IDS that effectively bridges the critical gap between accurate threat detection and safe, autonomous remediation in 5G-IoT networks. The core of our contribution is the formalization of the remediation problem as a CMDP and the implementation of a runtime safety shield, which together ensure the reinforcement learning agent's pursuit of optimal threat mitigation remains within operational safety constraints. Extensive evaluation on the 5G-NIDD benchmark confirms the viability of our approach, demonstrating not only high detection accuracy but also the emergence of intelligent, context-aware mitigation policies. The system's ability to learn to avoid service-disruptive actions, validated through comparative analysis of Q-tables with and without the safety shield, along with its computational efficiency, underscores its significant advantage over traditional, detection-centric IDS.

This work establishes a strong foundation for deploying trustworthy autonomous defense systems in critical and resource-constrained next-generation network environments.

While effective, the system is currently limited by its dependence on static safety constraints and labeled training data. Future work will address these limitations by exploring dynamic constraint adaptation and self-supervised learning to improve resilience against evolving zero-day threats. This work lays a solid foundation for deploying trustworthy autonomous defense systems in critical and resource-constrained next-generation network environments.

Acknowledgement: The authors state that Grammarly's AI tool was used solely to refine English in a few sections.

Funding Statement: The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through the Large Group Project under grant number (RGP2/245/46). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R333), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Wajdan Al Malwi: writing original draft, visualization, validation, software, project administration, methodology, investigation, formal analysis, conceptualization. Fatima Asiri: writing an original draft, visualization, methodology, investigation, formal analysis. Nazik Alturki: review & editing, visualization, methodology. Noha Alnazzawi: review & editing, software, project administration. Dimitrios Kasimatis: writing, review & editing, visualization, validation. Nikolaos Pitropakis: writing original draft, software, methodology, formal analysis. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The experiments conducted in this study are based exclusively on the publicly available, open-source 5G-NIDD dataset, which can be accessed from its original repository. The processed datasets and the Jupyter Notebooks developed for the proposed methodology are available from the authors upon reasonable request and subject to approval by the research group, for non-commercial research purposes.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Varga P, Peto J, Franko A, Balla D, Haja D, Janky F, et al. 5G support for industrial IoT applications—challenges, solutions, and research gaps. Sensors. 2020;20(3):828. doi:10.3390/s20030828.
2. Mukherjee S, Gupta S, Rawlley O, Jain S. Leveraging big data analytics in 5G-enabled IoT and industrial IoT for the development of sustainable smart cities. Trans Emerg Telecomm Technol. 2022;33(12):e4618. doi:10.1002/ett.4618.
3. Pokhrel SR, Ding J, Park J, Park OS, Choi J. Towards enabling critical mMTC: a review of URLLC within mMTC. IEEE Access. 2020;8:131796–813. doi:10.1109/access.2020.3010271.
4. Ji H, Park S, Yeo J, Kim Y, Lee J, Shim B. Ultra-reliable and low-latency communications in 5G downlink: physical layer aspects. IEEE Wirel Commun. 2018;25(3):124–30. doi:10.1109/mwc.2018.1700294.
5. Ahmed SF, Alam MSB, Afrin S, Rafa SJ, Taher SB, Kabir M, et al. Toward a secure 5G-enabled internet of things: a survey on requirements, privacy, security, challenges, and opportunities. IEEE Access. 2024;12(8):13125–45. doi:10.1109/access.2024.3352508.
6. Liao H, Murah MZ, Hasan MK, Aman AHM, Fang J, Hu X, et al. A survey of deep learning technologies for intrusion detection in Internet of Things. IEEE Access. 2024;12(1):4745–61. doi:10.1109/access.2023.3349287.
7. Alshehri M, Saidani O, Malwi W, Asiri F, Latif S, Khattak A, et al. A hybrid wasserstein gan and autoencoder model for robust intrusion detection in iot. Comput Model Eng Sci. 2025;143(3):3899–920. doi:10.32604/cmes.2025.064874.

8. Al-Fuhaidi B, Farae Z, Al-Fahaidy F, Nagi G, Ghallab A, Alameri A. Anomaly-based intrusion detection system in wireless sensor networks using machine learning algorithms. Appl Comput Intell Soft Comput. 2024;2024(1):2625922. doi:10.1155/2024/2625922.
9. Muneer S, Farooq U, Athar A, Ahsan Raza M, Ghazal TM, Sakib S. A critical review of artificial intelligence based approaches in intrusion detection: a comprehensive analysis. J Eng. 2024;2024(1):3909173. doi:10.1155/2024/3909173.
10. Kalodanis K, Papapavlou C, Feretzakis G. Enhancing security in 5G and future 6G Networks: machine learning approaches for adaptive intrusion detection and prevention. Future Internet. 2025;17(7):312. doi:10.3390/fi17070312.
11. Alnazzawi N, Asiri F, Alturki N, Laula Z, Zafar S, Latif S, et al. MGNN-IDS: a multi-graph neural network approach for robust intrusion detection in the internet of things. Telecommun Syst. 2025;88(4):1–20. doi:10.1007/s11235-025-01352-5.
12. Ahmad I, Kumar T, Liyanage M, Okwuibe J, Ylianttila M, Gurtov A. Overview of 5G security challenges and solutions. IEEE Commun Stand Mag. 2018;2(1):36–43. doi:10.1109/mcomstd.2018.1700063.
13. Anwar S, Mohamad Zain J, Zolkipli MF, Inayat Z, Khan S, Anthony B, et al. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. Algorithms. 2017;10(2):39. doi:10.3390/a10020039.
14. Bashendy M, Tantawy A, Erradi A. Intrusion response systems for cyber-physical systems: a comprehensive survey. Comput Secur. 2023;124(1):102984. doi:10.1016/j.cose.2022.102984.
15. Altman E. Constrained Markov decision processes. Oxfordshire, UK: Routledge; 2021.
16. Kheddar H, Dawoud DW, Awad AI, Himeur Y, Khan MK. Reinforcement-learning-based intrusion detection in communication networks: a review. IEEE Commun Surv Tutor. 2024;27(4):2420–69. doi:10.1109/comst.2024.3484491.
17. Rani SS, Shaaban MF, Ali A. An efficient convolutional neural network based attack detection for smart grid in 5G-IOT. Int J Crit Infrastruct Prot. 2025;48:100738. doi:10.1016/j.ijcip.2024.100738.
18. Gurushanker A, Anandakumaran K, Columbus CC. Enhancing intrusion detection systems for 5G networks using AI. In: Proceedings of the 2024 First International Conference on Software, Systems and Information Technology (SSITCON); 2024 Oct 18–19; Tumkur, India. p. 1–8.
19. Neha, Bhatia T. Adaptive intrusion detection system leveraging dynamic neural models with adversarial learning for 5G/6G Networks. In: Proceedings of the 2025 4th International Conference on Computer Technologies (ICCTech); 2025 Feb 20–23; Kuala Lumpur, Malaysia. p. 103–7. doi:10.1109/ICCTech66294.2025.00028.
20. Reis MJ. AI-driven anomaly detection for securing IoT devices in 5G-enabled smart cities. Electronics. 2025;14(12):2492. doi:10.3390/electronics14122492.
21. Radoglou-Grammatikis P, Nakas G, Amponis G, Giannakidou S, Lagkas T, Argyriou V, et al. 5GCIDS: An intrusion detection system for 5G core with AI and explainability mechanisms. In: Proceedings of the 2023 IEEE Globecom Workshops (GC Wkshps); 2023 Dec 4–8; Kuala Lumpur, Malaysia. p. 353–8.
22. Adjewa F, Esseghir M, Merghem-Boulahia L. Efficient federated intrusion detection in 5G ecosystem using optimized BERT-based model. In: Proceedings of the 2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob); 2024 Oct 21–23; Paris, France. p. 62–7.
23. Mahmood I, Alyas T, Abbas S, Shahzad T, Abbas Q, Ouahada K. Intrusion detection in 5G cellular network using machine learning. Comput Syst Sci Eng. 2023;47(2):2439–53. doi:10.32604/csse.2023.033842.
24. Samarakoon S, Siriwardhana Y, Porambage P, Liyanage M, Chang SY, Kim J, et al. 5G-NIDD: a comprehensive network intrusion detection dataset generated over 5G wireless network. IEEE dataport; 2022. doi:10.21227/xtep-hv36.