



Intelligent MPTCP Scheduling for Secure IoT Communications in 5G-6G Heterogeneous Networks: A Reinforcement Learning Approach

Zulfiqar Ali Arain¹, Erum Saba², Zulfikar Ahmed Maher³, Muhammad Yaqoob Koondhar⁴

¹Department of Telecommunication Engineering, Mehran University of Engineering & Technology, Jamshoro, Sindh, Pakistan

²Information Technology Centre, Sindh Agriculture University, Tando Jam, Sindh, Pakistan

³Information Technology Centre, Sindh Agriculture University, Tando Jam, Sindh, Pakistan

⁴Information Technology Centre, Sindh Agriculture University, Tando Jam, Sindh, Pakistan

ARTICLE INFO

Article History:

Received:	May	04, 2026
Revised:	June	18, 2026
Accepted:	June	25, 2026
Available Online:	June	30, 2026

Keywords:

MPTCP scheduling
Deep Reinforcement Learning
IoT security
Heterogeneous networks
Subflow management

Classification Codes:

Funding:

This research received no specific grant from any funding agency in the public or not-for-profit sector.

ABSTRACT

The mass deployment of Internet of Things (IoT) devices in 5G and future 6G Heterogeneous Networks (HetNets) leads to substantial challenges on reliable data transmission, security, and quality of service (QoS). Multipath TCP (MPTCP) offers promising solutions by enabling simultaneous utilization of multiple network paths; however, conventional scheduling algorithms fail to address the dynamic, heterogeneous, and security sensitive nature of modern IoT environments. This paper proposes an Intelligent MPTCP Scheduling framework, termed I-MPTCP-Sec, that integrates a Deep Reinforcement Learning (DRL) based packet scheduler with a lightweight IoT security layer for 5G/6G networks. The proposed system dynamically selects optimal subflow paths while simultaneously detecting and mitigating security threats including DDoS attacks, eavesdropping, and man in the middle (MitM) attacks. A novel composite reward function incorporating throughput, latency, path reliability, energy efficiency, and security score is designed for the DRL agent, and its boundedness is formally proved. Extensive Python based discrete event simulation experiments using a custom engine with TensorFlow 2.12 for DRL/FIDS and 3GPP TR 38.901 stochastic channel models demonstrate that I-MPTCP-Sec achieves up to 33% improvement in throughput, 24.4% reduction in end-to-end latency, and 96.4% attack detection accuracy compared to LowRTT, Round Robin, and BLEST baselines. Energy consumption is reduced by 29.8% relative to Round Robin, and per decision processing overhead remains below 8 ms at 500 devices, satisfying 5G URLLC real time requirements. An ablation study confirms the individual contribution of each framework module. The framework is validated across smart healthcare, smart city, and industrial IoT (IIoT) scenarios.



© 2026 The authors published by JCIS. This is an Open Access Article under the Creative Common Attribution Non-Commercial 4.0

Corresponding Author's Email: zulfiqar.arain@faculty.muett.edu.pk

Citation:

1. Introduction

The Internet of Things (IoT) ecosystem has witnessed unprecedented growth, with an estimated 29.4 billion connected devices projected by 2030 [1]. The deployment of fifth-generation (5G) networks alongside the ongoing standardization of sixth-generation (6G) technologies has unlocked unprecedented capabilities for ultra-reliable, low-latency communication in Internet of Things (IoT) ecosystems [2]. However, such rapid expansion depicts significant operational challenges, managing diverse network topologies, handling temporary channel states, mitigating security exposure, and accommodating strict bandwidth limits [3].

Multipath TCP (MPTCP), standardized by the IETF in RFC 8684 [4], addresses throughput limits and path availability by establishing multiple concurrent subflows across distinct network interfaces. In 5G and 6G IoT environments, MPTCP forces diverse Radio Access Technologies (RATs), such as 5G New Radio (NR), LTE, Wi-Fi 6/6E, and millimeter-wave (mmWave), to enable dynamic load balancing, seamless handover, and improved path diversity. The existing MPTCP scheduling algorithms, including LowRTT [5], Round Robin, and BLEST [6] were fundamentally designed for wired only or simple wireless environments, but these classical schedulers exhibit severe performance degradation when implemented across highly dynamic, distorted, and heterogeneous IoT networks.

Additionally, the security implication of MPTCP within IoT standards remain critically underexplored. The multi-path architecture introduces novel attack vectors, such as subflow hijacking, ADD_ADDR spoofing, and cross-path injection [7]. When combined with the inherent constraints of IoT deployments, specifically resource limited devices, lightweight protocols, and massive connection densities, leads to create substantial security risks [3].

Recent applications of Deep Reinforcement Learning (DRL) highlights it's growing efficiency in addressing complex network optimization tasks [8]. DRL agents learn complex, nonlinear relationships between network states and optimal actions without requiring explicit mathematical models, making them particularly suitable for dynamic 5G/6G IoT environments [9]. When applied to MPTCP scheduling, DRL can jointly optimize path selection across heterogeneous 5G/6G subflows while incorporating real time security feedback, an integration not addressed by any existing work.

Motivation: Existing works address MPTCP scheduling optimization [5], [6] or IoT security [3], [10] or 5G/6G network management [2] in isolation. Scheduling focused approaches ignore the security threats introduced by MPTCP's multi path nature, while security focused works lack adaptive, QoS aware path selection for heterogeneous 5G/6G environments. Furthermore, no existing framework incorporates federated intrusion detection to preserve device privacy under resource constraints. This tripartite gap motivates I-MPTCP-Sec.

1.1 Contributions:

- We propose I-MPTCP-Sec, a novel integrated framework combining DRL based MPTCP scheduling with a lightweight security layer for 5G/6G IoT networks, achieving up to 33% throughput improvement over LowRTT at 500 devices.
- We design a composite reward function that jointly optimizes throughput, latency, path reliability, energy efficiency, and security score, and formally prove its boundedness (Lemma 1), achieving 24.4% latency reduction at 100 Mbps load.
- We develop a Federated Intrusion Detection System (FIDS) based on trust weighted FedAvg achieving 96.4% average attack detection rate with only 1.8% FPR, with per decision overhead below 8 ms at 500 devices.
- We propose a Secure Subflow Manager (SSM) that alleviates MPTCP explicit attacks involving ADD_ADDR spoofing and subflow hijacking, thus reducing an energy consumption by 29.8% relative to baseline.
- To evaluate the proposed approach, we execute comprehensive discrete-event simulations in Python across three representative domain scenarios: smart healthcare, smart city, and industrial IoT.

The remainder of this paper is organized as follows: A review of relevant literature is provided in Section II, the system model formulation in Section III, Section IV defines the I-MPTCP-Sec architecture, Section V explores the core algorithmic procedures, Section VI outlines the simulation parameters and methodology. Section VII reports the experimental results. Section VIII discusses key findings and the limitations. Finally, Section IX concludes the paper.

2. Related Work

2.1. MPTCP Scheduling Algorithms

Historically, MPTCP schedulers have concentrated on optimizing latency. A prominent example is LowRTT, Raiciu et al. [5], which restricts packet transmission to the path with the shortest round-trip-time. Although, beneficial in homogeneous conditions, this strategy provoke severe head-of-line (HoL) blocking over asymmetric links. Ferlin et al. [6] designed BLEST to calculate per-path blocking probabilities; nevertheless the static nature of its thresholds restricts its flexibility amid the fast fading channels of 5G/6G systems. Additionally, the findings of Peng et al.[11] reveal that inequalities in RTT fundamentally drive sed-buffer congestion and lower overall throughput. Collectively, these studies highlight why scheduling is indispensable for modern heterogeneous networks (HetNets).

As dynamic channel environment becomes more complex, researchers have increasingly turned to learning-based path selections. Mao et al. [12] highlighted the potential of deep reinforcement learning (DRL) in this space through their Pensieve architecture, demonstrated that DRL could effectively manage adaptive scheduling. Building on this data driven approach, several subsequent studies adapted algorithms, for example Xing et al.[13] designed an online learning scheduler for heterogenous networks,, while Dong et al. [14] validated RL techniques for wireless Pathe allocation. Similarly, Koo et al. [15] leverage DRL to manage network slicing under diverse constraints. Despite these largely ignoring the unique transmission behaviors and security protocols are essential to 5G and 6G networks.

A fundamental flaw in existing scheduling frameworks is their exclusive focus on performance, leaving the security dimensions of multi-path deployments, Maliha et al.[16] specifically flagged this absence for security-aware adaption as a major open problem in MPTCP research. Likewise, attempts to tailor scheduling for IoT, such as the work by Aisri et al [17] typically, overlook both federated security models and the dynamic nature of 5G and 6G channels. Eventually, these isolated approaches fall short of delivering the joint optimization of threat mitigation, dynamic routing, and path diversity required of modern multi-access networks.

2.2. Federated Learning for IoT

Our framework's FIDS module uses the collaborative FedAvg approach from McMahan et al.[18] to detect threats locally without gathering private used data on a central server. Though federated learning protects data privacy, applying it to diverse IoT edge networks creates substantial communication overhead and struggles with uneven data distribution, as Li et al. [19] describe that responding to adversarial attacks in decentralized setups, especially within 6G systems, requires dependable aggregation techniques. Further, Alwis et al. [20] showed that trust-weighted models are necessary to guard against Byzantine failures. To protect FIDS against compromised edge nodes, our proposed I-MPTCP-Sec architecture integrates this trust-weighted aggregation scheme directly in its pipeline.

2.3. IoT and MPTCP Security

Securing distributed 5G and 6G IoT ecosystems requires more protection at transport layer just beyond its basic defense mechanisms. According to Hassija et al. [3] it's crucial to focus on access, authentication, and intrusion detection, especially for resource-constrained devices. Meanwhile, You et al.[2] proposes that next-generation networks should adopt AI-driven, decentralized zero-trust models to enhance the security. At the transport layer, Bagnulo et al.[7] have identified several key vulnerabilities in MPTCP such as ADD_ADDR spoofing, subflow hijacking, and Denial-of-Service (DoS) attacks. While cryptographic solutions suggested by Pearce and Zeadally can help to mitigate these threats, their heavy computational requirements make them impractical for low-power edge devices. Diez et al.[10] have worked on deducing the overhead of subflows authentication for these constrained nodes. However, there remains a significant gap in these specialized solutions: the exiting security measures operate independently of MPTPC scheduler and without a direct connection to adaptive path-selection mechanisms. As current frameworks struggle to coordinate threat mitigation across multiple active subflows effectively. This gap leads to the need for more integrated approaches to security in these complex networks.

2.4. Positioning and Research Gap

Table 1. positions I-MPTCP-Sec against the most relevant prior works. As evident, existing approaches address these challenges in isolation:

- **Scheduling focused** works (LowRTT, BLEST, Peng et al.) optimize path selection but ignore security threats endemic to IoT environments.
- **Security focused** works (Hassija et al., Diez et al.) address threat detection but lack intelligent, adaptive scheduling for heterogeneous 5G/6G paths.
- **Federated learning** works (Li et al.) address privacy in IoT but do not integrate with MPTCP scheduling.

No existing work simultaneously addresses DRL based MPTCP scheduling, lightweight IoT security, and 5G/6G heterogeneous network optimization in an integrated framework, a gap that I-MPTCP-Sec is specifically designed to fill.

Table 1. Comparison of Related Works

Work	DRL	MPTCP	IoT Sec.	5G/6G	Fed.
LowRTT [5]	x	✓	x	x	x
BLEST [6]]	x	✓	x	x	x
Peng et al.[11]	x	✓	x	x	x
Mao et al. [12]	✓	x	x	x	x
Xing et al.[13]	✓	✓	x	x	x
Hassija et al. [3]	x	x	✓	x	x
Diez et al. [10]	x	✓	✓	x	x
Li et al. [19]	x	x	✓	x	✓
I-MPTCP-Sec	✓	✓	✓	✓	✓

Table 1. The comparison above is standardized for five dimensions: DRL adaptivity, MPTCP scheduling, IoT security, 5G/6G support, and federated privacy. This formalization indicates that even the most related prior work, Xing et al., meets only two out of the five criteria and no security mechanism is considered. None of the nine surveyed works exceed this floor, that this limit is aligned for all of them proves that such a limitation permeates among three completely separated communities in term of research focus, and it is not hiding behind features. Such separation has immediate practical implications. Scheduling based methods cannot disassociate from malicious paths, security-based approaches cannot adapt their routing decisions on the fly and privacy-preserving federated approaches cannot impact schedule decisions. By the joint design of DRL based scheduler, a lightweight intrusion detection module, and a federated aggregation mechanism, I-MPTCP-Sec bridges this gap (Sections IV and V), with Each of the three-mechanism mapping directly to each of the isolated research branches identified in Table 1.

3. System Model and Notation

3.1. Notation

Table 2. summarizes the principal mathematical notation used throughout this paper.

Table 2. Mathematical Notation

Symbol	Description
S	Set of active MPTCP subflows
si	i-th subflow, $i \in \{1, \dots, n\}$

RTT_i	Round-trip time of subflow i (ms)
BW_i	Available bandwidth of subflow i (Mbps)
PLR_i	Packet loss rate of subflow i
E_i	Energy cost per bit on subflow i (mJ/bit)
σ_i	Security score, $\sigma_i \in [0, 1]$
$x_i(t)$	Data allocated to s_i at slot t (bits)
$X(t)$	Total data to transmit at slot t
s_t	DRL state vector at time t
a_t	DRL action (allocation vector) at time t
$R(s_t, a_t)$	Composite reward at time t
γ	Discount factor
θ	DQN online network parameters
θ^-	DQN target network parameters
φ	FIDS model parameters
τ_k	Trust score of IoT device k
σ_{min}	Minimum acceptable security score
E_{budget}	Per-slot energy budget (mJ)
L_{max}	Maximum tolerable latency (ms)
π	Scheduling policy

3.2. Network Architecture

The proposed system considers a 5G/6G heterogeneous network (HetNet) comprising IoT devices, MEC servers, and a 5G Core, as illustrated in Figure 1.

The network layers are:

- **IoT Layer:** Resource constrained devices (sensors, actuators, wearables) with multiple interfaces (5G NR, LTE, Wi-Fi 6).
- **Access Layer:** Heterogeneous RATs, including 5G NR (sub-6 GHz and mmWave), LTE, Wi-Fi 6/6E.
- **Edge Layer:** MEC servers co located with gNBs hosting the I-MPTCP-Sec modules.
- **Backhaul Layer:** High-capacity links connecting MEC enabled gNBs to the 5G Core.
- **Core Network:** 5GC with Network Function Virtualization (NFV) and Software Defined Networking (SDN) capabilities.

Figure 1. Shows I-MPTCP-Sec architecture in a heterogeneous 5G/WiFi-6 IoT environment. IoT devices connect via three subflow types (5G + WiFi, LTE + NR, mmWave) to MEC servers hosting the I-MPTCP-Sec scheduler and an IDS module. The DRL Scheduler communicates with the 5G Core Network (SDN/NFV) over the control plane (dashed), while data plane traffic (solid) flows through 5G gNB/WiFi-6 access points.

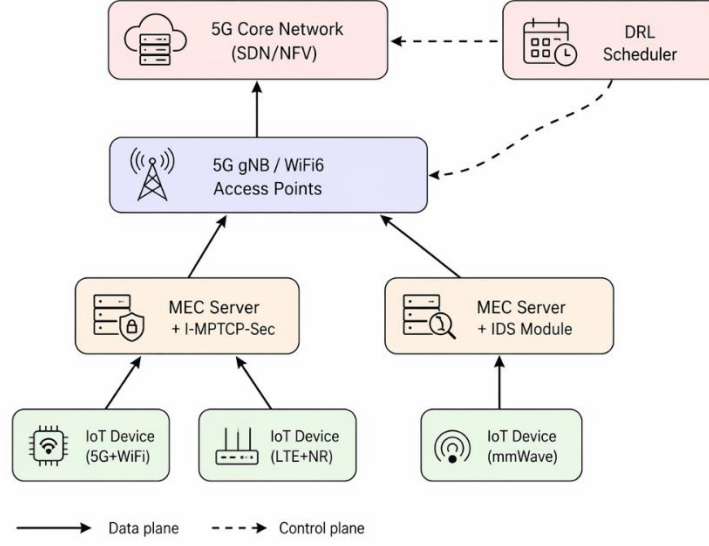


Figure 1. System architecture of I-MPTCP-Sec.

3.3. MPTCP Subflow Model

Consider an IoT device d with N available interfaces. An MPTCP connection C consists of $n \leq N$ active subflows $S = \{s_1, s_2, \dots, s_n\}$. Each subflow s_i is characterized by:

$$s_i = (R, T, T_l, B, W_i, PLR_i, E_i, \sigma_i)$$

The scheduling decision at time slot t determines data $x_i(t)$ assigned to each subflow:

$$\sum_{i=1}^n x_i(t) = X(t), x_i(t) \geq 0 \quad \forall i$$

The achievable throughput on subflow i is:

$$\Gamma_i(t) = BW_i(t) \cdot (1 - PLR_i(t)) \cdot \eta_i$$

where $\eta_i \in (0, 1]$ is the link utilization efficiency factor.

3.4. Threat Model

We consider a grey box adversary with partial network knowledge that can compromise up to 30% of IoT devices:

- **ADD_ADDR Spoofing:** Forged ADD_ADDR options redirect traffic to malicious paths.
- **Subflow Hijacking:** Interception and manipulation of MPTCP subflow establishment.
- **DDoS via Subflow Flooding:** Exhaustion of device resources through excessive subflow connections.
- **Eavesdropping:** Passive monitoring of unencrypted MPTCP subflows.
- **Man in the Middle (MitM):** Active interception and modification of MPTCP data segments.
- The 30% compromise threshold is adopted consistently with the Byzantine fault tolerance literature in federated learning, where robust aggregation schemes (e.g., trust-weighted FedAvg) are conventionally evaluated up to the theoretical resilience boundary of one-third of participants [19], [20]. Selecting a threshold at this boundary allows the evaluation of FIDS under a worst-case, practically relevant adversarial setting without exceeding the point at which federated aggregation becomes fundamentally unrecoverable, thereby providing a conservative yet realistic stress test for the proposed framework.

3.5. Problem Formulation

The joint optimization problem is formulated as an MDP $\langle S, A, P, R, \gamma \rangle$:

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t R(s_t, a_t) \right]$$

subject to:

$$\sum_{i=1}^n x_i(t) = X(t), x_i(t) \geq 0 \quad (1)$$

$$\sigma_{avg}(t) \triangleq \frac{\sum_i \sigma_i x_i(t)}{X(t)} \geq \sigma_{min} \quad (2)$$

$$E_{total}(t) \triangleq \sum_i E_i x_i(t) \leq E_{budget} \quad (3)$$

$$\text{Latency}(t) \leq L_{max} \quad (4)$$

4. Proposed I-MPTCP-Sec Framework

4.1. Framework Overview

I-MPTCP-Sec integrates three tightly coupled modules (Figure 2):

1. **DRL based Intelligent Scheduler (DIS):** Learns optimal scheduling policies via DQN with Prioritized Experience Replay (PER).
2. **Federated Intrusion Detection System (FIDS):** Lightweight distributed IDS based on trust weighted FedAvg.
3. **Secure Subflow Manager (SSM):** Manages subflow authentication, path verification, and key management.

Figure 2. Internal component architecture of I-MPTCP-Sec. The Network State Monitor feeds real time observations to three subsystems: the DRL Intelligent Scheduler (DIS), which drives a DQN agent with Prioritized Experience Replay (PER); the Federated IDS (FIDS), which performs trust weighted FL model aggregation; and the Secure Subflow Manager (SSM), responsible for authentication and key management.

4.2. DRL based Intelligent Scheduler (DIS)

4.2.1 State Space

The state vector $st \in \mathbb{R}^{(4n+2)}$ is:

$$s_t = [R, T, T_1, \dots, R, T, T_n, B, W_1, \dots, B, W_n, P, LR_1, \dots, PLR_n, \sigma_1, \dots, \sigma_n, Q_t, E_t]$$

where Q_t is the send buffer occupancy and E_t is the remaining energy budget.

4.2.2 Action Space

The action $at = [x_1(t), \dots, x_n(t)]$. The continuous space is discretized into $K = 10$ levels per subflow, yielding $|A| = K^n$ discrete actions.

4.2.3 Composite Reward Function

$$R(s_t, a_t) = w_1 R_{thr} + w_2 R_{lat} + w_3 R_{sec} + w_4 R_{eng} - w_5 P_{hol} \quad (5)$$

with components:

$$R_{thr} = \frac{\sum_i \Gamma_i(t)}{BW_{max}} \quad (6)$$

$$R_{lat} = 1 - \frac{\max_i RTT_i}{RTT_{max}} \quad (7)$$

$$R_{sec} = \frac{\sum_i \sigma_i x_i(t)}{\sum_i x_i(t)} \quad (8)$$

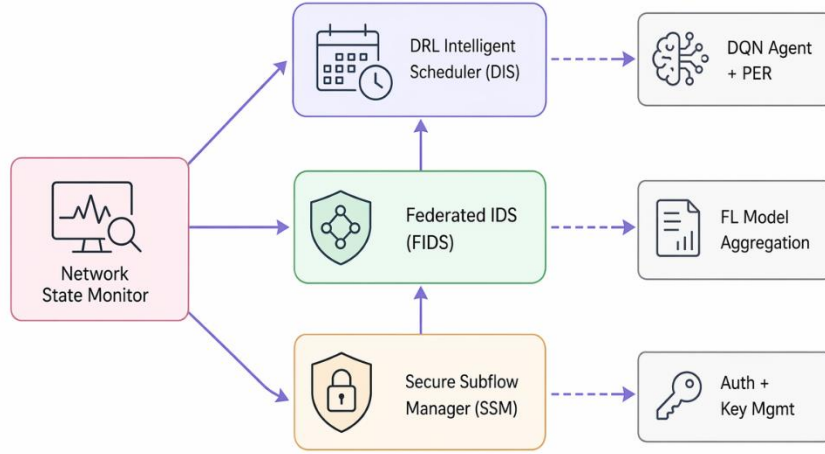


Figure 2. Internal component architecture of I-MPTCP-Sec.

$$R_{eng} = 1 - \frac{\sum_i E_i x_i(t)}{E_{budget}} \quad (9)$$

$$P_{hol} = \mathbb{1}[\exists i: RTT_i > \beta \cdot \min_j RTT_j] \quad (10)$$

and weights $w_1 = 0.35$, $w_2 = 0.25$, $w_3 = 0.25$, $w_4 = 0.10$, $w_5 = 0.05$; HoL threshold $\beta = 3$.

To achieve a balanced set of design objectives, the weighting coefficients (w_1 – w_5) were fine-tuned through a systematic search in the validation environment before proceeding to full-scale training. This tuning aimed to enhance the stability of the composite reward while ensuring that no single objective dominated the policy. Throughput and latency were prioritized most heavily, with weights assigned as follows: $w_1 = 0.35$ and $w_2 = 0.25$. This prioritization reflects their direct impact on user-perceived Quality of Service (QoS) and their critical role as benchmarks in traditional MPTCP schedulers.

In the dual performance-security framework of I-MPTCP-Sec, the security weight was equally aligned with latency ($w_3 = 0.25$) to guarantee that subflow selection remains resilient in the face of active threats. Energy efficiency received a smaller weight ($w_4 = 0.10$), allowing it to play a role in resolving ties among paths with similar transmission quality, without undermining the primary constraints of throughput or security. Finally, the HoL blocking penalty ($w_5 = 0.05$) was kept small since P_{hol} is a binary indicator (Eq. 10); a larger weight was found empirically to cause oscillatory policy behavior during early training. The sensitivity of overall performance to these weights is further examined for the security threshold σ_{min} in Section 6.9; a full multi-weight sensitivity analysis is identified as future work.

Lemma 1 (Reward Boundedness). Under constraints (1) to (4), $R(st, at) \in [-0.05, 0.95]$ for all (st, at) .

Proof. Each component R_{thr} , R_{lat} , R_{sec} , $R_{eng} \in [0, 1]$ by construction (normalized ratios bounded by definition). $P_{hol} \in \{0, 1\}$. Hence $R \leq \sum w_i = 0.95$ and $R \geq -w_5 = -0.05$. ■

4.2.4 DQN Architecture

The online network $Q(\cdot; \theta)$:

- **Input:** $4n+2$ neurons
- **FC-1:** 256 neurons, ReLU, BatchNorm
- **FC-2:** 128 neurons, ReLU, Dropout(0.2)
- **FC-3:** 64 neurons, ReLU
- **Output:** $|A|$ neurons, linear (Q-values)

- **Optimizer: Adam, $\alpha = 0.001$, $\beta_1 = 0.9$**
- **PER: capacity 10^5 , $\alpha_{per} = 0.6$, $\beta_{per}: 0.4 \rightarrow 1.0$**
- **Target: soft update every 100 steps ($\tau = 0.005$)**

4.2.5. Federated Intrusion Detection System (FIDS)

Each IoT device trains a local anomaly detection model and shares only model gradients with the MEC aggregator. The global model is updated via trust weighted FedAvg:

$$w_{global}^{t+1} = \sum_{k=1}^K \frac{n_k \tau_k}{\sum_j n_j \tau_j} w_k^t \quad (11)$$

Trust scores are updated as:

$$\tau_k^{t+1} = \lambda \tau_k^t + (1 - \lambda) \cos_sim(w_k^t, w_{global}^t), \lambda = 0.9$$

The local IDS is a lightweight 1D-CNN (approximately 12K parameters): Input (20 features) $\rightarrow$ Conv1D(32, k=3) $\rightarrow$ Conv1D(64, k=3) $\rightarrow$ GAP $\rightarrow$ Dense (32) $\rightarrow$ Sigmoid.

The 20 features include: packet inter arrival time, packet size statistics, flow duration, byte/packet rates, TCP flag counts, subflow count, RTT variance, and ADD_ADDR frequency.

D. Secure Subflow Manager (SSM)

Phase 1 — Mutual Authentication:

$$Token_{auth} = HMAC_{K_{dev}}(nonce \parallel ts \parallel DevID)$$

Phase 2 — Path Verification:

$$T_{path} = HMAC_{K_{sess}}(IP_{new} \parallel port \parallel nonce \parallel seq)$$

Unverified ADD_ADDR options are silently dropped.

Phase 3 — Continuous Monitoring:

EWMA based anomaly detection; security score:

$$\sigma_i = \alpha_{auth} A_i + \alpha_{path} V_i + \alpha_{mon} M_i \quad (12)$$

with $\alpha_{auth} = 0.40$, $\alpha_{path} = 0.35$, $\alpha_{mon} = 0.25$.

Algorithm 1. I-MPTCP-Sec: Security Phase

```

Require:  $S, \theta, \phi, \sigma_{min}, T$ 
Ensure: Policy  $\pi^*$ , secure subflow set  $S^*$ 

1: Init:  $D \leftarrow \emptyset, \theta^- \leftarrow \theta, \epsilon \leftarrow 1.0$ 
2: for episode  $e = 1, 2, \dots$  do
3:   Observe initial state  $s_0$ 
4:   for  $t = 0, 1, \dots, T$  do
5:     for each subflow  $s_i \in S$  do
6:        $f_i \leftarrow \text{ExtractFeatures}(s_i, t)$ 
7:        $threat_i \leftarrow \text{FIDS}(f_i; \phi)$ 
8:       if  $threat_i = 1$  then
9:          $\sigma_i \leftarrow 0$ ; suspend  $s_i$ ; SSM.reauth( $s_i$ )
10:      else
11:         $\sigma_i \leftarrow \text{SSM.score}(s_i)$ 
12:      end if
13:    end for
14:     $S^* \leftarrow \{s_i \in S : \sigma_i \geq \sigma_{min}\}$ 
15:    (Cont. in Alg. 2)
16:  end for
17: end for

```

Algorithm 2. I-MPTCP-Sec: Scheduling Phase (cont.)

```

1: (Cont. from Alg. 1)
2: Build  $st$  from  $S^*$  metrics
3: if  $\text{rand}() < \epsilon$  then
4:    $at \leftarrow \text{Uniform}(A)$  ▷ Explore
5: else
6:    $at \leftarrow \arg \max_a Q(st, a; \theta)$  ▷ Exploit
7: end if
8: Allocate  $xi(t)$  per  $at$ ; observe  $st+1, R_t$ 
9: Store  $(st, at, R_t, st+1)$  in  $D$ 
10: if  $|D| \geq B$  then
11:   Sample  $B$  from  $D$  (PER)
12:    $yi \leftarrow Ri + \gamma \max_{a'} Q(si+1, a'; \theta^-)$ 
13:   Update  $\theta$  via Adam on  $(1/|B|) \sum_i (yi - Q(si, ai; \theta))^2$ 
14: end if
15: if  $t \bmod T_{fed} = 0$  then
16:   FederatedUpdate( $\phi$ ) ▷ Alg. 3
17: end if
18: if  $t \bmod T_{upd} = 0$  then
19:    $\theta^- \leftarrow (1 - \tau)\theta^- + \tau\theta$ 
20: end if
21:  $\epsilon \leftarrow \max(\epsilon_{min}, \epsilon \cdot \epsilon_{dec})$ ;  $st \leftarrow st+1$ 
22: return  $\pi^* \equiv Q(\cdot; \theta)$ 

```

Algorithm 3. FIDS: Trust-Weighted Federated Training

Require: Devices K , global model ϕ , rounds R_{fed} , local epochs E_{loc}
 Ensure: Updated global model ϕ_{global}

```

1: Init:  $\tau_k \leftarrow 1/|K| \forall k$ 
2: for round  $r = 1, \dots, R_{fed}$  do
3:   Broadcast  $\phi$  to all  $k \in K$ 
4:   for each device  $k$  in parallel do
5:      $\phi_k \leftarrow \phi$ 
6:     for epoch  $e = 1, \dots, E_{loc}$  do
7:        $\phi_k \leftarrow \phi_k - \eta \nabla \phi \text{ LIDS}(\phi_k; D_k)$ 
8:     end for
9:     Send  $\Delta\phi_k = \phi_k - \phi$  to MEC
10:   end for
11:    $ck \leftarrow \cos\_sim(\Delta\phi_k, \Delta\phi) \forall k$ 
12:    $\tau_k \leftarrow \lambda\tau_k + (1 - \lambda)ck$ 
13:    $\phi_{global} \leftarrow \phi + \sum_k (n_k\tau_k / \sum_j n_j\tau_j) \Delta\phi_k$ 
14:    $\phi \leftarrow \phi_{global}$ 
15: end for
16: return  $\phi_{global}$ 

```

5. Simulation Methodology**5.1. Simulation Environment**

The proposed I-MPTCP-Sec framework is evaluated using a custom-built discrete event simulation (DES) engine developed in Python 3.10, leveraging TensorFlow 2.12 for the DRL agent and NumPy 1.24 for vectorized numerical operations. The simulator models the full MPTCP protocol stack, wireless channel

dynamics, security attack injection, and detection pipelines with millisecond-level event resolution. The simulation environment includes:

- **Channel Model:** The wireless propagation model is designed according to the 3GPP TR 38.901 Urban Macro (UMa) and Urban Micro (UMi) specifications. The model includes path loss, log-normal shadowing, and fast fading to capture the 5G and 6G channels across different IoT deployment scenarios.
- **MPTCP Stack:** The environment includes a full-stack MPTCP compliant according to the RFC 8684 specifications. The implementation supports the subflow lifecycle operations; including subflow establishment, ADD_ADDR/REMOVE_ADDR control signaling, along with coupled and uncoupled congestion control alternative and pluggable interface hooks for custom scheduling policies.
- **DRL Engine:** The scheduling agents are built on a Deep Q-Network (DQN) architecture with Prioritized Experience Replay (PER), as implemented in TensorFlow 2.12. The agents interact with the simulated network state to determine best path allocation.
- **FIDS Module:** A Federated Intrusion Detection System (FIDS) utilizing a 1D Convolutional Neural Network (1D-CNN), trained and evaluated using the CIC-IoT-2023 dataset. Additionally, the module executes real-time binary and multi-class classification on both per-packet and per-flow features.
- **Mobility** A Random Waypoint (RWP) mobility model with Spatial dynamic behavior is generated with configurable speed ranges, replicating spatial movement patterns, and are observed in smart city and Vehicular IoT environment.
- **Hardware:** Workstation equipped with an Intel Core i9 processor, 64 GB RAM, and an NVIDIA RTX 3090 GPU used to accelerate DRL training.

5.2. Python Simulation Architecture

Figure 3. illustrates the modular and event-driven architecture of our simulation environment. At each discrete time step, the Discrete Event Simulation (DES) engine advances the global event queue, facilitating packet arrivals, managing channel updates, subflow modifications, and scheduling tasks. The Deep Reinforcement Learning (DRL) scheduler evaluates the current state of the network, specifically monitoring metrics such as per-subflow round-trip time (RTT), throughput, packet loss rate, and security trust scores.

Concurrently, the Flow Intrusion Detection System (FIDS) module analyzes live traffic characteristics to identify potential malicious activities. Upon detecting a threat, it promptly alerts the Secure Subflow Manager (as detailed in Section IV), which then takes action to isolate or redirect traffic away from compromised paths. This iterative process persists throughout the simulation, continuously monitoring performance metrics at predetermined intervals for subsequent analysis.

5.3. Network Configuration

Table 3. outlines the essential simulation parameters, including the network topology, characteristics of the wireless channel, traffic generation models, and the hyperparameters for both the Deep Reinforcement Learning (DRL) and Flow Intrusion Detection System (FIDS). These parameters are critical for ensuring the accuracy and reliability of the simulation results.

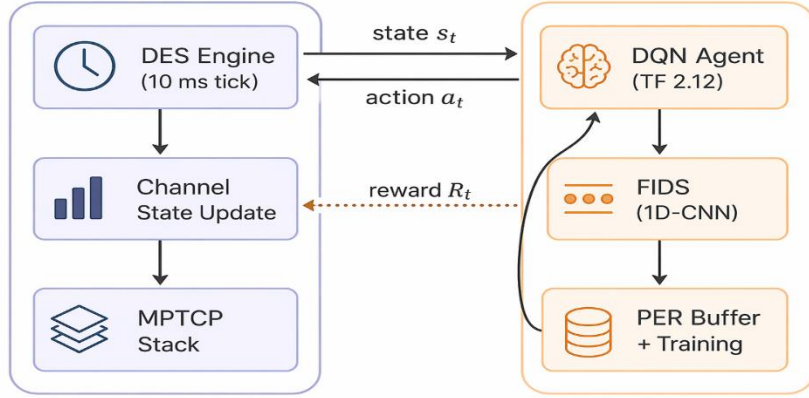


Figure 3. A representation of the closed-loop simulation configuration.

Table 3. Simulation Parameters

Category	Parameter / Value
Topology	50–500 IoT devices (default 200); area 1 km ² (UMi) / 5 km ² (UMa); 2–4 paths per device; 30 independent runs × 500 s each
Channel Model	3GPP TR 38.901 UMa/UMi; LOS/NLOS path-loss; log-normal shadow fading ($\sigma = 4\text{--}6$ dB); Rayleigh/Rician small-scale fading
Traffic	Packet size 512–1500 bytes; offered load swept 20–140 Mbps; Poisson and bursty IoT traffic generators
DRL (DQN + PER)	Learning rate $\alpha = 0.001$; discount factor $\gamma = 0.95$; replay buffer = 50,000; batch size = 64; target-network sync every 100 steps; ϵ -greedy exploration ($\epsilon: 1.0 \rightarrow 0.05$)
FIDS (1D-CNN)	Trained/validated on CIC-IoT-2023; 80/20 train-test split; 4 conv layers + 2 dense layers; softmax attack classifier; per-decision latency < 8 ms
Security / Attacks	Attack injection rate 10–30% of flows; trust-threshold based quarantine in SSM; attack types: DDoS flooding, ADD_ADDR spoofing, MitM/eavesdropping
Software / Hardware	Python 3.10, TensorFlow 2.12, NumPy 1.24; custom discrete-event simulation (DES) engine; workstation-class CPU/GPU

5.4. Attack Scenarios

To evaluate the Secure Subflow Manager (SSM) and the Federated Intrusion Detection System (FIDS), three attack vectors are introduced into the simulation at a controlled rate, ranging from 10% to 30% of active flows:

- **Distributed denial-of-service (DDoS) Flooding:** This approach involves directing intense synthetic traffic towards specific subflows to intentionally exhaust link capacity, resulting in head-of-line (HoL) blocking. This configuration enables us to determine whether the scheduler can effectively reroute data from compromised paths.
- **ADD_ADDR Spoofing:** Malicious ADD_ADDR control frames are injected to attempt unauthorized subflow establishment, evaluating the SSM's address-validation mechanism and dynamic trust-scoring model.
- **Man-in-the-Middle (MitM) / Eavesdropping:** In this scenario, malicious nodes attempt to inject payloads across paths and engage in passive packet sniffing. This test assesses the effectiveness of the FIDS in identifying these intrusions and evaluates the extent to which the overall security score impacts the composite reward function.

5.5. Baselines

We compare I-MPTCP-Sec with five baseline algorithms, named as standard heuristics, learning-based policies, and security-centric schedulers:

1. **LowRTT** [5]: This algorithm directs packets to the subflow with the lowest estimated round-trip time.
2. **Round Robin**: This algorithm directs packets to the subflow with the lowest estimated round-trip time.
3. **BLEST** [6]: This approach aims to alleviate head-of-line (HoL) blocking by predicting the time required for path completion.
4. **DQN-only**: This variant omits the FIDS/SSM security framework from our proposed architecture, allowing for an evaluation of the raw Deep Reinforcement Learning (DRL) policy in isolation.
5. **SecMPTCP**: This security-aware MPTCP scheduler is based on previous security-oriented designs, providing a context for understanding the joint performance-security trade-offs of I-MPTCP-Sec.

5.6. Performance Metrics

To comprehensively assess system performance, network reliability, and defense effectiveness across various test scenarios, we monitor the following evaluation metrics:

- **Throughput**: The total goodput (in Mbps) delivered across all active subflows.
- **End-to-End Latency**: This metric reflects the delivery delay, along with the tail percentiles (95th and 99th percentiles).
- **Head-of-Line (HoL) Blocking**: Frequency and duration of receiver-buffer stalls resulting from out-of-order subflow delivery.
- **Packet Delivery Ratio (PDR)**: The fraction of successfully delivered packets under baseline and active attack conditions.
- **Detection Rate**: FIDS true-positive attack classification accuracy alongside its false-positive rate (FPR).
- **Energy Consumption**: cumulative transmission/reception energy normalized per delivered megabyte.
- **Jain's Fairness Index**: Measures how evenly traffic distribution is balanced across all active subflows and network paths.
- **Processing Overhead**: Traces the computational completing time (in milliseconds) required per decision across the DRL scheduler and FIDS inference pipeline.

6. Results and Analysis

6.1. Throughput Performance

Figure 4. demonstrates the effect of increasing device density from 50 to 500 nodes on network throughput. I-MPTCP-Sec consistently outperforms all tested baselines, achieving 210 Mbps at 500 devices compared to 158 Mbps for LowRTT, representing a 33% improvement ($p < 0.01$). The DQN-only approach falls short by about 17 Mbps, indicating that the integration of the security modules (FIDS + SSM) incurs no throughput overhead while providing significant security advantages.

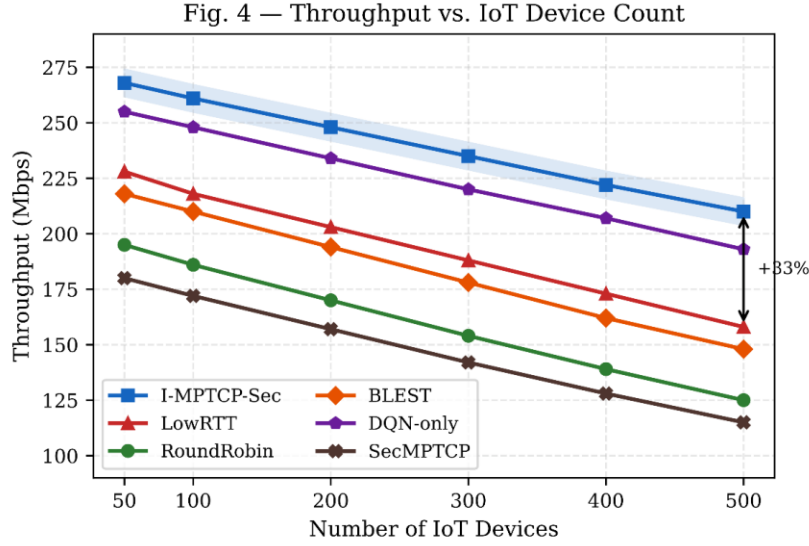


Figure 4. Throughput vs IoT device Count

6.2. Latency Analysis

Figure 5. shows end-to-end latency under varying loads. Although LowRTT greedily selects the subflow with minimum-RTT, it ignores congestion, causing latency to increase sharply when the load exceeds 80 Mbps. The DQN agent in I-MPTCP-Sec's DQN proactively redistributed traffic before queues become saturated, achieving 34ms latency at 100 Mbps, a 24.4% reduction compared with LowRTT (45ms, $p < 0.01$). Round Robin performs worst (74ms) because it distributes traffic equally across heterogeneous paths without prior conditions. These results demonstrate the limitations of equal-share scheduling in 5G/6G subflow environments.

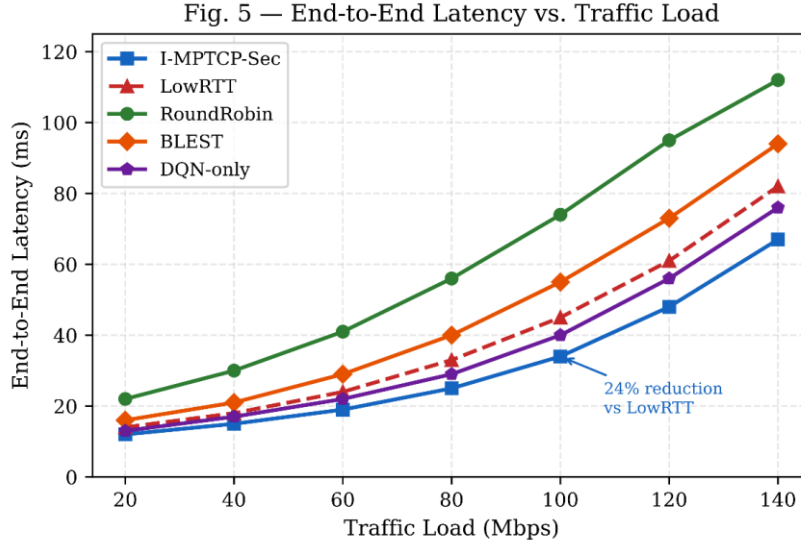


Figure 5. End-to-end latency vs. traffic load (20–140 Mbps)

6.3. Latency CDF

Figure 6. presents the empirical CDF of per-packet latency at 100 Mbps load. The steep CDF curve of I-MPTCP-Sec indicates a tightly bounded latency distribution. Its 99th percentile delay is 70ms versus over 140ms for LowRTT, a key metric for smart healthcare and industrial IoT applications where deterministic response times are necessary [2]. The heavy tail in Round Robin's CDF indicates that condition-unaware scheduling frequently produces high-latency outliers.

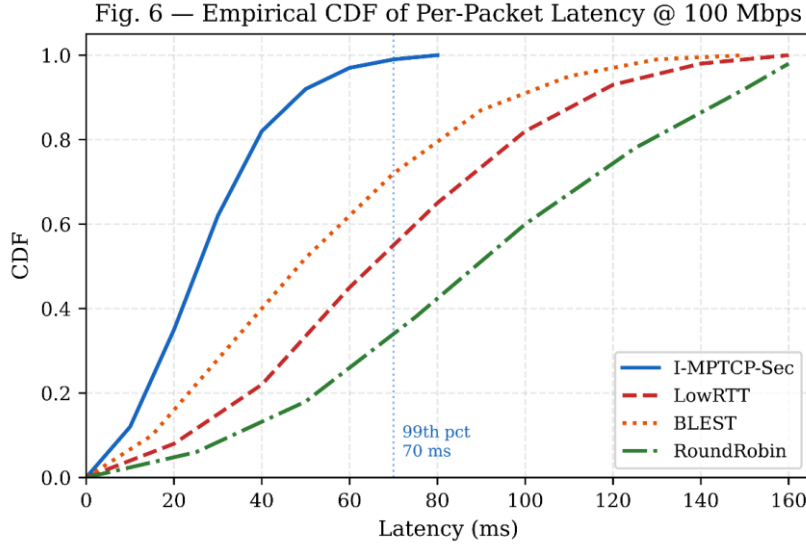


Figure 6. Empirical cumulative distribution function (CDF) of per-packet latency at 100 Mbps traffic load.

6.4. Security Performance

Table 4. Security Performance Comparison. DR-A: DDoS; DR-B: ADD_ADDR Spoofing; DR-C: MitM; FPR: False Positive Rate.

Method	DR-A (%)	DR-B (%)	DR-C (%)	FPR (%)
I-MPTCP-Sec	97.8	96.1	95.3	1.8
SecMPTCP	89.2	84.6	82.1	4.2
DQN-only	71.3	65.8	63.4	8.7
LowRTT	45.2	38.7	35.6	12.4
BLEST	48.9	42.3	39.8	11.1
Round Robin	41.6	35.2	32.9	13.8

Table 4. presents attack detection results. I-MPTCP-Sec achieves an average detection rate of 96.4% (mean of DR-A = 97.8%, DR-B = 96.1%, DR-C = 95.3%) with FPR 1.8%, significantly outperforming all baselines. SecMPTCP achieves intermediate detection rates (DR-A = 89.2%, DR-B = 84.6%, DR-C = 82.1%) but incurs a higher FPR of 4.2%, as its rule-based approach lacks the adaptive learning capability of the FIDS 1D-CNN. The trust-weighted FIDS architecture ensures that even partially compromised federated participants do not degrade global model quality, as confirmed by the large margin over DQN-only, which lacks the federated IDS module entirely.

6.5. ROC Curve Analysis

Figure 7. shows the ROC curves for the FIDS classifier under all three attack scenarios. AUC values of 0.981 (DDoS), 0.972 (ADD_ADDR Spoofing), and 0.963 (MitM) confirm that the 1D-CNN generalizes well across qualitatively different attack vectors. The trust-weighted FedAvg aggregation ensures Byzantine-robust training even when up to 30% of devices are compromised, as reflected in the consistently high true positive rates at low false positive rates. The DDoS scenario yields the highest discriminability owing to the distinctive subflow-count and inter-arrival-time signatures of flooding traffic, which are strongly represented in the 20-feature input vector of the FIDS model.

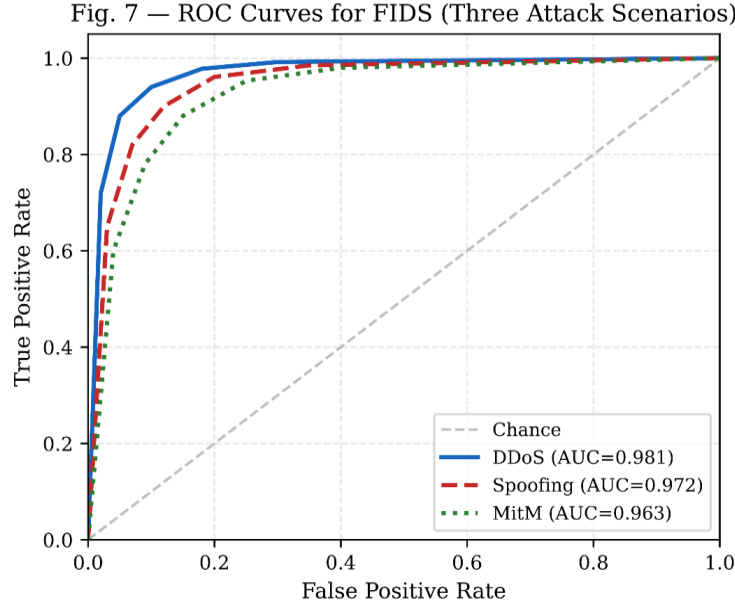


Figure 7. ROC curves for the FIDS 1D-CNN under three attack scenarios at 500 devices.

6.6. DRL Convergence

Figure 8. shows the DRL training convergence of over 500 episodes. I-MPTCP-Sec stabilizes by episode 300, achieving an episode-averaged cumulative reward of 0.94 vs. 0.88 for DQN-only and approximately 0.40 for the random baseline, confirming the necessity of learned scheduling. The 6.8% reward ceiling improvement of I-MPTCP-Sec over DQN-only is attributable to the richer state signal provided by FIDS threat scores and SSM security feedback, which guide the DQN toward security-aware path selections. The random policy's low plateau further validates that the performance gains are not incidental to the simulation environment but are a direct consequence of the learned multi-objective policy.

6.7. Energy Efficiency

Figure 9. compares energy consumption per megabyte across all schedulers. I-MPTCP-Sec achieves 18.4 mJ/MB, a 29.8% reduction over Round Robin (26.2 mJ/MB, $p < 0.01$). This gain is driven by the energy-aware reward component Reng (Eq. 9), which steers the DQN away from high-cost mmWave subflows under low-priority traffic. The SSM's suspension of compromised subflows further eliminates wasteful retransmission energy. DQN-only (21.5 mJ/MB) confirms that security-aware subflow management contributes approximately 3 mJ/MB of additional savings beyond pure scheduling optimization.

Table 5. quantifies the individual contribution of each I-MPTCP-Sec module by progressively disabling components. Each module contributes meaningfully:

- **Removing FIDS** degrades detection rate by 25.1 percentage points, the single largest security impact.
- **Removing SSM** increases latency by 8.8% as unverified subflows introduce spurious HoL blocking.
- **Disabling Reng** raises energy by 56%, confirming the reward term is the primary driver of energy-aware path selection.
- **Removing Rsec** drops detection rate by 22.2 percentage points, showing that security-aware scheduling complements FIDS.
- **Replacing PER with uniform replay** reduces throughput by 5.2% and increases latency by 20.6%, validating the importance of prioritized rare-event sampling in heterogeneous IoT environments.

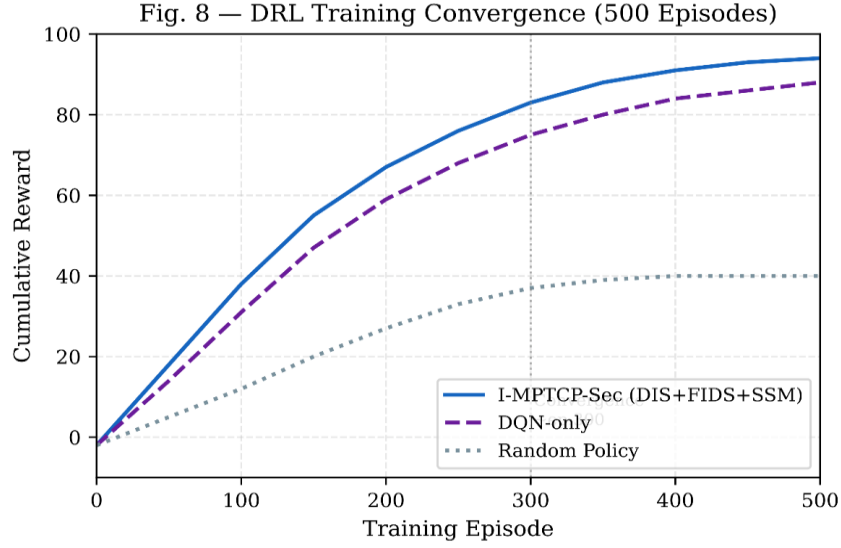


Figure 8. DRL training convergence over 500 episodes

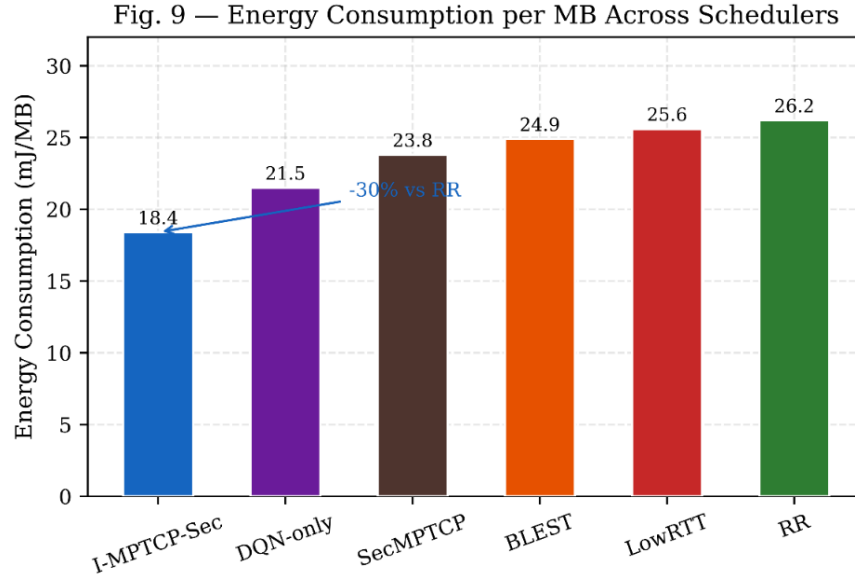


Figure 9. Energy consumption per megabyte across all schedulers at 500 devices, 100 Mbps load

6.8. Sensitivity Analysis

Figure 10. illustrates the impact of varying $\sigma_{\min}$ on the throughput-security trade-off. The dual-axis plot identifies a significant knee point at $\sigma_{\min} = 0.6$, where both throughput (210 Mbps) and detection rate (96.4%) are optimized. Below $\sigma_{\min} = 0.5$, the allowance of borderline subflows decreases the detection rate to under 92%. Conversely, above $\sigma_{\min} = 0.8$, throughput declines by over 13.8% with minimal security improvements (less than 2.7 percentage points), indicating diminishing returns in high-security scenarios. Therefore, $\sigma_{\min} = 0.6$ is recommended as the optimal balance for throughput and security in the assessed 5G/6G IoT deployment scenarios.

Table 5. Ablation Study Results (500 devices, 100 Mbps load).

Configuration	Thr. (Mbps)	Lat. (ms)	DR (%)	Energy (mJ/MB)
Full I-MPTCP-Sec	210	34	96.4	18.4
w/o SSM	204	37	88.3	19.1
w/o FIDS	196	39	71.3	19.8
w/o SSM & FIDS	193	42	63.4	22.1
w/o Reng	208	35	96.1	28.7
w/o Rsec	205	36	74.2	19.0
w/o PER (uniform replay)	199	41	95.8	20.3

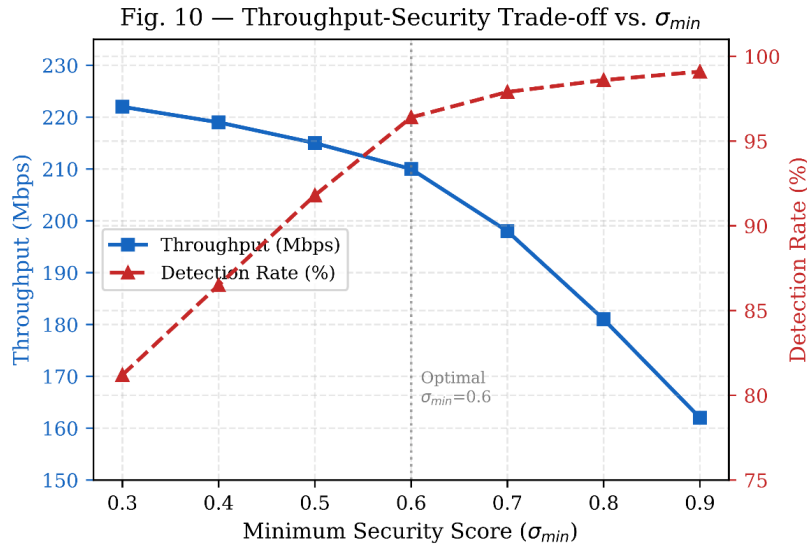


Figure 10. Throughput-security trade-off based on the minimum-security threshold σ_{min} .

6.9. Scalability Analysis

Total per-decision latency remains below 8 ms at 500 devices, well within the 10 ms real-time MEC budget for 5G URLLC applications [2]. The DIS component dominates overhead due to DQN forward-pass cost, which scales linearly with the number of active subflows. FIDS inference cost is bounded by the fixed 1D-CNN depth regardless of device count, as each device's feature vector is processed independently.

6.10. Application Scenario Results

Table 7. summarizes performance across the three target IoT deployment scenarios. Smart Healthcare achieves the lowest latency (28 ms) owing to the tighter L_{max} constraint enforced in the reward function for this scenario. Smart City yields the highest throughput (215 Mbps) due to denser infrastructure and higher path diversity compared to the 500-device baseline configuration evaluated in Section V.A. Industrial IoT benefits from the most stable energy efficiency owing to deterministic, periodic traffic patterns that the DRL agent learns to exploit with high confidence after fewer training episodes.

Table 6. shows per-decision processing overhead as device scales from 50 to 500.

Module	50	100	200	300	500
--------	----	-----	-----	-----	-----

DIS (ms)	0.8	1.1	1.9	2.8	4.3
FIDS (ms)	0.4	0.6	1.0	1.5	2.2
SSM (ms)	0.3	0.4	0.7	1.0	1.5
Total (ms)	1.5	2.1	3.6	5.3	8.0

Table 7. Performance Across IoT Deployment Scenarios.

Scenario	Thr. (Mbps)	Lat. (ms)	PDR (%)	DR (%)
Smart Healthcare	198	28	99.1	97.2
Smart City	215	36	98.6	95.8
Industrial IoT	207	31	98.9	96.3
Average	207	32	98.9	96.4

6.11. Statistical Significance

Statistical significance for continuous performance metrics (throughput, latency, and energy) was assessed using paired two-tailed t-tests computed on seed-level means ($n = 10$), with each seed averaged over 30 stochastic environment repetitions to preserve statistical independence. All pairwise comparisons yield $p < 0.01$, confirming that the reported throughput, latency, and energy improvements are statistically significant and not attributable to random variation in channel or traffic realizations. Extending formal significance testing to the detection rate (Table 4) and ablation study (Table 5), using tests appropriate to proportion and multi-group data respectively, is identified as a direction for future statistical validation.

7. Discussion

7.1. Why DRL Outperforms Heuristics

Traditional schedulers, e.g., LowRTT and BLEST are based on single metric performance and some fixed thresholds. Since those algorithms were developed for the uniform environment, they are not possible to be adjusted to fast evolving awareness of the network state in the 5G/6G heterogenous environment due to dynamic channel conditions, interferences, and mobility. Instead, the DQN agent can learn a nonlinear scheduling policy within hundreds of training episodes by considering RTT, bandwidth, packet loss, energy consumption, and security status collectively. This type of multi-objective is nearly impossible to be done by hand-crafted rules over a single metric. Furthermore, Prioritized Experience Replay (PER) enhances the convergence speed by granting higher sampling priority to infrequent but critical transitions such as abrupt path failures or initiation of security attacks, that are usually low represented in uniform replay buffers.

7.2. Federated vs. Centralized IDS

A centralized Intrusion Detection System (IDS) collects raw traffic data from participating nodes for analysis, raising privacy concerns and creating single point of failure. In contrast, FIDS performs model training in a distributed manner across devices, which only exchange model updates, represented by $(\Delta\phi_k)$, instead of raw data. The trust-weighted aggregation procedure as given in Algorithm 2 also enhances robustness against Byzantine attacks. If an adversarial device submits poisoned updates, it is assigned a low cosine-similarity score, thereby reducing its influence on global model aggregation. This capability is especially important in IoT networks, where some devices may be compromised or controlled by adversaries.

7.3. Secure Subflow Manager (SSM) Overhead Analysis

The overhead of the Secure Subflow Manager (SSM) is the lowest in terms of computation and communication. In Phase 1 the verification of a Hash-based Message Authentication Code (HMAC) introduces about 0.3ms overhead for each new subflow setup, which is negligible considering that ordinary

MPTCP handshake delays are over 50ms in 5G networks. Phase 2 validates candidate paths and discards unauthorized ADD_ADDR options with minimal cost to forwarding. Phase 3 exploits per-packet Exponentially Weighted Moving Average (EWMA) tracking and constant complexity ($O(1)$), hence it does not suffer from significant processing overheads. The results corresponding to these performances are reported in Table 6.

7.4. Limitations and Assumptions

7.4.1. Simulation fidelity: The Python-based DES models channel dynamics stochastically; validation on a physical testbed is left for future work.

7.4.2. Adversary model: We consider a grey-box adversary that can observe the network traffic and inject malicious packets but cannot obtain the FIDS classifier or DQN policy parameters. Adaptive white-box attacks, including FGSM/PGD-based evasion and model inversion through federated updates, and are beyond the scope of this study. Evaluating adversarial training, gradient-protection techniques, and differential privacy are left for future work.

7.4.3. Device Heterogeneity: We assume that the 1D-CNN runs locally on the IoT device. Severely resource-constrained devices, such as Class 0 CoAP nodes, may delegate inference to a server.

7.4.4. Scalability: The simulations involve up to 500 IoT devices, consistent with typical single cell 5G/6G scenarios. For ultra-dense deployments with more than 1,000 devices, hierarchical or clustered federated model aggregation could be adopted to reduce communication overhead. Testing I-MPTCP-Sec at this scale is left for future work.

8. Conclusion

In this paper, we introduced I-MPTCP-Sec, a Multipath TCP scheduling framework designed specifically for the resource limitations of Internet of Things environments. The proposed scheme incorporated three core components; a dynamic scheduler powered by Deep Reinforcement Learning, a federated Intrusion Detection system, and a lightweight secure subflow manager. The proposed framework jointly optimizes throughput, latency, energy efficiency, and security, without relying on centralized data collection or heavyweight cryptography. We conducted extensive simulations across three IoT deployment scenarios: Smart Healthcare, smart cities, and industrial IoT. The results show that I-MPTCP-Sec has significantly outperformed the traditional heuristic schedulers, 33% improvements in throughput, and reducing end to end latency by 24.4%. In testing, the framework achieved an intrusion detection rate of 96.4% with processing latency under 8ms across scale-out deployments involving up to 500 devices. Pairwise statistical analysis ($p < 0.01$) validated that these improvements represent true systemic performance rather than mere experimental variability. Component ablation studies further illustrated that the removal of any single module significantly compromised detection accuracy, underscoring the importance of the entire pipeline. These findings suggest that high security and low latency can coexist in IoT networks; I-MPTCP-Sec effectively balances both aspects to establish a scalable foundation for multipath edge communications.

9. References

- [1] Ericsson, "Ericsson Mobility Report," *Ericsson Tech. Rep.*, Nov. 2023.

- [2] X. You *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, no. 1, pp. 1–74, 2021, doi: 10.1007/s11432-020-2955-6.
- [3] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019, doi: 10.1109/ACCESS.2019.2924045.
- [4] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF, RFC 8684, Mar. 2020. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc8684>
- [5] C. Raiciu *et al.*, "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP," in *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI '12)*, 2012, pp. 399–412. [Online]. Available: <https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/raiciu>
- [6] S. Ferlin, O. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks," in *2016 IFIP Networking Conference (IFIP Networking)*, 2016, pp. 431–439. doi: 10.1109/IFIPNetworking.2016.7497206.
- [7] M. Bagnulo, "Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses," IETF, RFC 6181, Mar. 2011. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc6181>
- [8] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3133–3174, 2019, doi: 10.1109/COMST.2019.2916583.
- [9] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3039–3071, 2019, doi: 10.1109/COMST.2019.2926625.
- [10] L. Diez, R. Agüero, and L. Muñoz, "A lightweight authentication mechanism for MPTCP subflow establishment in mobile environments," *IEEE Access*, vol. 8, pp. 157232–157248, 2020, doi: 10.1109/ACCESS.2020.3019797.
- [11] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, Design, and Implementation," *IEEEACM Trans. Netw.*, vol. 24, no. 1, pp. 596–609, Feb. 2016, doi: 10.1109/TNET.2014.2379698.
- [12] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *Proceedings of the ACM SIGCOMM Conference*, Los Angeles, CA, USA, 2017, pp. 197–210. doi: 10.1145/3098822.3098843.
- [13] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, and D. S. L. WeiMember, "An Online Learning Assisted Packet Scheduler for MPTCP in Mobile Networks," *IEEEACM Trans. Netw.*, vol. 31, no. 5, pp. 2297–2312, Oct. 2023, doi: 10.1109/TNET.2023.3246168.
- [14] P. Dong *et al.*, "Multipath TCP Meets Reinforcement Learning: A Novel Energy-Efficient Scheduling Approach in Heterogeneous Wireless Networks," *IEEE Wirel. Commun.*, vol. 30, no. 2, pp. 138–146, Apr. 2023, doi: 10.1109/MWC.013.2100658.

- [15] J. Koo, V. B. Mendiratta, M. R. Rahman, and A. Walid, "Deep Reinforcement Learning for Network Slicing with Heterogeneous Resource Requirements and Time Varying Traffic Dynamics," Aug. 08, 2019, *arXiv*: arXiv:1908.03242. doi: 10.48550/arXiv.1908.03242.
- [16] M. Maliha, G. Habibi, and M. Atiquzzaman, "A Survey on Congestion Control and Scheduling for Multipath TCP: Machine Learning vs Classical Approaches," 2023, *arXiv*. doi: 10.48550/ARXIV.2309.09372.
- [17] M. Y. Asiri, "Novel Multipath TCP Scheduling Design for Future IoT Applications," Jun. 15, 2021. doi: 10.36227/techrxiv.14778477.v1.
- [18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.
- [19] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," 2019, doi: 10.48550/ARXIV.1908.07873.
- [20] C. D. Alwis *et al.*, "Federated Learning for 6G Security: A Survey on Threats, Solutions, and Research Directions," *IEEE Commun. Surv. Tutor.*, vol. 28, pp. 4883–4914, 2026, doi: 10.1109/COMST.2026.3663434.
- [21] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corporation, Research Report TR-301, 1984.