International Journal of Safety and Security Engineering
Vol. 16, No. 3, March, 2026, pp. 617-633
Journal homepage: http://iieta.org/journals/ijsse

Latency-Security Co-Design Framework for Edge IoT Gateways

Nalini Nara[1*], V Jyothsna[2]

[1] Department of Computer Science and Engineering, Mohan Babu University, Tirupati 517501, India
[2] Department of Data Science, Mohan Babu University, Tirupati 517501, India

Corresponding Author Email: nalini.kmit@gmail.com

https://doi.org/10.18280/ijsse.160314

Received: 5 February 2026
Revised: 17 March 2026
Accepted: 27 March 2026
Available online: 31 March 2026

Keywords:
edge Internet of Things, timing side-channel, intrusion detection system, latency-aware authentication, edge computing, AI-gated security

ABSTRACT

Edge Internet of Things (IoT) gateways have become key convergence points, and authentication, cryptographic key management and intrusion detection have to be performed simultaneously within strict latency limits. Current security systems in IoT analyze the authentication protocols and intrusion detection systems (IDS) separately, resulting in the execution latency uncontrollability, timing side-channel vulnerability, as well as security compromises in the presence of a large number of devices concurrently. This inherent constraint is the focus of this paper: with a latency-security co-design framework, a coherent authentication and intrusion detection orchestration represent one integrated and timing aware of security pipeline at the IoT edge. The proposed framework connects AI-gated behavioral authentication, combined latency-conscious authentication-IDS scheduling, and a hybrid convolutional neural networks (CNN)-Long Short Term Memory (LSTM)-Attention IDS inference engine that mean that cryptographic session establishment is one given to devices with IDS-validated devices. Formal analysis proves timing side-channel leakage, end to end security latency, authenticating flooding attacks, and cryptographic forward secrecy. The results of the extensive experimental assessment based on the BoT (Botnet)-IoT dataset and large-scale simulations of the NS-3 show that the proposed framework can ensure security latency of 50 ms or less and detectability of over 95% and scalability beyond 10,000 coexisting IoT devices. These consequences of these results are that latency is a security primitive to the real-world edge IoT Deployments.

1. INTRODUCTION

The fast growth in the number of Internet of Things (IoT) deployments in smart cities, industrial automation, healthcare, and intelligent transportation systems has fundamentally redefined the distribution of security risk between centralized cloud-based deployments and edge IoT deployments [1]. These gateways are now multi-purpose security implementation locations, and they are in charge of authenticating devices, setting up cryptographic keys, maintaining sessions, and real-time inference of the intrusion for thousands of heterogeneous IoT nodes [2]. In contrast to cloud systems, edge gateways have finite computing resources, shared execution streams, and hard real-time response criteria, which leads the security processes to operate on shared processing and memory resources and time constraints [3]. This convergence as more devices is packed together and the intensity of the traffic flows enables edge gateways to become performance bottlenecks in which security enforcement and latency cannot be considered as independent design issues but instead inherently intertwined [4]. Latency is no longer a performance figure in the environment but a basic security property. Authentication delays are susceptible to flood and resource exhaustion assaults and inference backlogs when performing intrusion detection generate short-lived blind windows that maleficence may depend on to execute covert intrusion [5]. The concurrent security activities introduce queuing delay to the security processing of an IoT device and the overall security response time is the sum of the effects of authentication processing, intrusion detection inference and queuing delay caused by concurrent security processing of the device. Once this cumulative latency reaches application-specific tolerances, the gateway goes into a vulnerable state where the authentication requests are held up and intrusion detection verdicts are delayed, allowing the gateway to make timing-based inference, selective denial of service, and IDS evasion attacks [6]. In spite of this inherent interconnection, current studies consider authentication schemes and intrusion detection system (IDS) as separate security levels. Cryptographic authentication schemes are usually tested in idealistic assumptions of zero concurrent IDS load, whereas IDS models do not take into consideration cryptographic handshake overhead or authentication bursts and report detection precision and inference delays [7]. This individual testing is essentially unstable in actual edge implementations, in which authentication and IDS detection run in parallel on common hardware sources [8]. This scattered design paradigm hides the realization of timing side-channels, the underestimation of the latency of end-to-end security, and the inability to model

cascading security failures that may occur as a result of synchronized access requests or adversarial traffic bursts. This tends to cause severe performance degradation and security leakage of the systems, which seemed safe in isolation, to be deployed at scale [9].

This paper explicitly covers the lack of latency security co-design in edge IoT IDS by considering authentication and intrusion detection as a time-constrained security pipeline. We present a co-located model that explicitly represents and manages the interaction between cryptographic authentication latency and IDS inference latency at the edge gateway going beyond the previous hybrid IDS-first solutions [10]. The core of the suggested solution is an AI-controlled login filtering mechanism, which conducts a lightweight behavioral checking procedure before more expensive cryptographic operations, hence preventing the early repression of malicious or abnormal authentication requests [11]. Simultaneously, we develop a scheduling scheme of latency sensitive authentication-IDS operations that dynamically assign resources at the edge to maintain the overall security response time within a given limit as traffic loads vary, in line with the real-time requirements of linked-time constraints as observed in recent low-latency IDS research [12].

The extensive experimental analysis proves that the suggested latency-security co-design can provide consistent intrusion detection rates, over 95 percent, with the cryptographic security and no apparent timing side-channel leakage, outperforming even the previous hybrid and edge-based IDS models [13]. In addition, the scalability of the framework is tested at scales greater than 10,000 parallel IoT nodes which confirms that the framework can support limited latency and security assurances even when deployed at high density, a constraint that has been explicitly mentioned in previous surveys of the IoT IDS and federated systems [14]. Figure 1 illustrates an edge IoT security architecture in which AI-gated authentication, joint scheduling, and intrusion detection are integrated at the edge gateway to enforce bounded latency and resist flooding, congestion, and timing inference attacks.

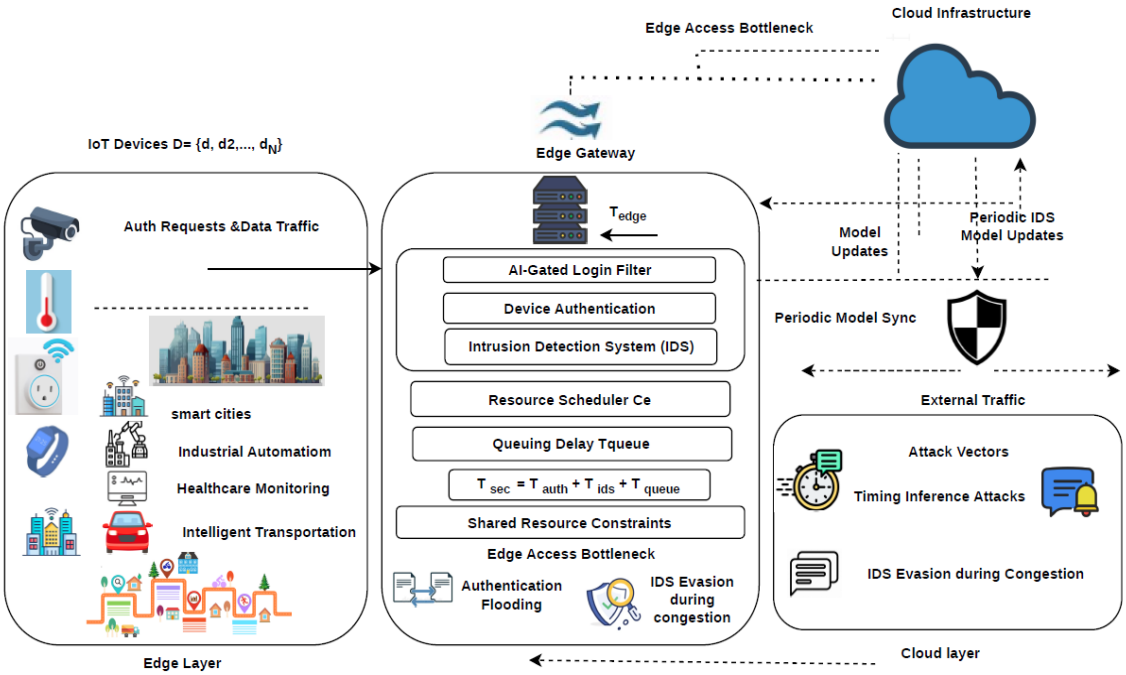

Figure 1. System and threat model of the proposed edge IoT framework

1.1 Contributions

The most important findings of the paper are as follows:

- Our co-design framework is the primary latency security co-design framework of edge IoT gateways that integrates the process of device authentication and intrusion detection into a single, timing-aware security pipeline, as opposed to viewing them as separate layers.
- We model the end-to-end security latency formally as a coupled-function of authentication latency, IDS inference latency, and queuing delay and reveal how uncontrollable latency directly maps to exploitable security vulnerabilities in the form of timing side-channels, denial-of-service amplification, and congestion induced IDS evasion.
- We propose an AI-gated authentication mechanism, which conducts lightweight screening of behavior before cryptographic processing, which greatly lessens the load of malicious authentication, and which does not compromise cryptographic guarantees to authentication flooding attacks.
- We develop an autonomous latency-sensitive scheduling approach that autonomously manages the scheduling of authentication and IDS inference activities in common edge resources to ensure finite security response time and avert IDS starvation amid adverse and bursty traffic flows.
- We give a strict formal security analysis of resistance to timing side-channel attacks, authentication flooding, IDS evasion when there is congestion, and maintenance of cryptographic forward secrecy when subjected to real adversarial analysis.
- We experimentally verify the proposed framework in terms of BoT (Botnet)-IoT dataset and large-scale NS-3 simulations, and achieve the results of ≤50 ms end-to-end security latency, >95% intrusion detection

accuracy, and scalability to 10,000 running IoT devices and adversarial load.

The remainder of this paper is organized as follows. Section 2 reviews related work on IoT authentication, intrusion detection, and edge security. Section 3 describes the system and threat models. Section 4 presents the latency model and design principles of the proposed framework. Section 5 details the proposed latency–security co-design algorithms. Section 6 provides formal security analysis, and Section 7 evaluates the framework through extensive experimental studies. Section 8 concludes the paper and outlines future research directions.

1.2 Notation and symbols

The mathematical symbols used throughout the manuscript are summarized in Table 1. These notations are employed in the system model, latency analysis, proposed framework, and experimental evaluation sections.

Table 1. Notation and symbols used in the proposed latency–security co-design framework

Symbol	Description
T_{sec}	Total end-to-end security response time
T_{auth}	Authentication processing latency
T_{ids}	IDS inference latency
T_{queue}	Queueing delay due to shared resources
T_{ECC}	ECC handshake computation time
T_{setup}	Session initialization time
$T_{refresh}$	Key refresh overhead
T_{CNN}	CNN feature extraction latency
T_{LSTM}	LSTM temporal modeling latency
T_{att}	Attention module latency
T_{cloud}	Cloud communication latency
T_{edge}	Edge processing latency
$D = \{d_1, d_2, \dots, d_N\}$	Set of IoT devices
N	Total number of IoT devices
C_e	Edge gateway computation capacity
λ	Combined task arrival rate
λ_a	Authentication request arrival rate
μ	Effective gateway service rate
τ	Maximum allowable latency bound
x_i	Feature vector of request i
$f(x_i)$	AI-gate classifier score
θ	AI decision threshold
g_i	Gate decision for request i
w_a	Scheduling weight for authentication
w_i	Scheduling weight for IDS
ΔT	Response-time difference used in timing attacks

Note: intrusion detection systems (IDS); convolutional neural networks (CNN); elliptic curve cryptography (ECC); Long Short Term Memory (LSTM)

1.3 Key definitions and terminology

To enhance clarity and eliminate the need to casually use technical terms, the key terms used in this manuscript are defined in this subsection. These terms provide a formal basis for the proposed framework for latency–security co-design and are consistently used in the following sections. By defining once the terminology precisely, the inter-relationship of latency control, attack resistance, and real-time security enforcement becomes more clear in the theoretical analysis and experiment interpretation.

(1) The latency-security invariant is the main principle of operation of the proposed framework. It ensures the end-to-end security response time is bounded by a certain tolerable threshold under normal traffic loads and even under malicious traffic loads such as flooding, congestion or bursting attacks. In our study the total security delay is the sum of authentication, IDS response, and queueing delay due to the shared gateway. That is, the framework assurances is defined in Eq. (1):

$$T_{sec} \leq \tau \tag{1}$$

Violation of this invariant can result in delayed gateway authentication, delayed gateway response to an intrusion or vulnerability to timing side-channel attacks. So, maintaining the latency-security invariant is equivalent to maintaining secure and timely gateway operations.

(2) Coherent latency is a concept of stable, deterministic and statistically average response-time behavior of repeated security operations. Under coherent latency, response time variability remains small enough so that an external observer cannot discern the state of the gateway, queue lengths, task scheduling priorities and transitions of the workload from the response times. This notion is relevant in the edge, where timing information may leak information to an adversary. In practice, coherent latency is characterized by low variance, large worst-case bounds and the absence of timing signatures.

(3) Deterministic security response means that security decisions such as authentication and security alarms such as intrusion detection as well as outcomes such as session establishment are made within predictable and guaranteed time frames, despite temporary traffic peaks. A deterministic security system maintains service availability and timely decision-making in the face of changing workloads, even in the presence of malicious traffic, unlike best-effort security systems, which may not function properly under varying loads. This characteristic is vital for IoT critical systems like medical monitoring, industrial process control and smart transportation systems.

(4) Using latency as a security primitive means latency is not merely treated as a performance metric, but a critical security property which impacts confidentiality, availability and resilience. High latency can induce blind spots for detection, exacerbate DoS attacks and introduce timing side channels. As a result, latency control is at least as important as ensuring authentication or IDS accuracy. The proposed framework is based on this perspective, and optimises security functions across security and latency constraints.

The above definitions show that the proposed framework not only reduces latency. On the contrary, it enables a security framework where limited latency, lack of timing variability, and attack resilience are enforced simultaneously via integrated authentication, scheduling and intrusion detection at the edge gateway.

2. LITERATURE REVIEW

In this section, previous studies that are most closely related to this paper are reviewed, and they are grouped into four categories namely edge-based IDS, IoT authentication mechanisms, latency-aware security frameworks, and co-design limitations. As opposed to traditional surveys, the discussion also highlights the reason why the current strategies do not solve the latency security coupling problem as this paper describes [15].

2.1 Edge-based intrusion detection systems

There has been a considerable amount of literature devoted to the implementation of the IDS at the IoT edge in order to minimize the detection latency and bandwidth usage [16]. The initial edge IDS systems were based on simple statistical models and rule-based systems in order to suit devices with limited capabilities. More recent research has used deep learning models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid CNN-Long Short Term Memory (LSTM) models [17] to enhance detection accuracy to more complicated attack patterns.

Let T_{ids} denote the latency of an edge IDS inference. Past researches downplay the effect of T_{ids} based on an implicit assumption that, in general authority, IDS inference is the most critical factor in the security response time. This assumption however does not take into consideration the execution of task of authentication and session management, which uses the same edge resources. Consequently, the measured end-to-end security response time is likely to be lower than the reported latency is given in Eq. (2):

$$T_{\text{sec}} = T_{\text{auth}} + T_{\text{ids}} + T_{\text{queue}} \tag{2}$$

In addition, the current edge IDS systems generally test the performance within a static traffic scenario, and do not simulate authentication bursts or adversarial access characteristics [18]. They therefore do not model congestion induced detection delays as well as timing leakage that arise in real deployment conditions. Although these systems enhance accuracy in detection, they do not consider the system-level interaction of intrusion detection with access control, and this is a huge gap in the practical security of IoT.

Current IDS techniques at the edge network dramatically increase the speed of detection and eliminate the need for cloud-based processing, and recent advances in deep learning enhance attack classification. However, existing research primarily focuses on IDS performance but lacks consideration of concurrent authentication traffic, gateway resource contention, and overall latency of security. Our work overcomes this gap by considering IDS inference, authentication latency and queueing delays together in a latency-security framework.

2.2 IoT authentication and key management schemes

To minimize computational costs and maintain confidentiality, integrity, and forward secrecy, IoT authentication studies have concentrated on short, lightweight cryptography protocols, such as ECC, identity-based authentication, and certificateless key agreement [19].

Let $T_{\text{auth}}^{\text{crypto}}$ represent the cryptographic authentication latency. The $T_{\text{auth}}^{\text{crypto}}$ isolation maximizes $T_{\text{auth}}^{\text{crypto}}$ by assuming that interaction with other security measures is negligible. But in the real-world deployments of edges the authentication is not done in isolation. Authentications are sent by massive numbers of devices with concurrent requests, creating bursty workloads that increase the queuing delay T_{queue} [20]. Most authentication protocols are analyzed under optimal conditions with low contention hence disregarding the cascading effect of authentication load on the IDS inference latency. Due to this, even cryptographically secure authentication schemes may be turned into denial-of-service and timing inference attacks as they are deployed at scale [21]. Moreover, authentication systems generally treat authenticated machines as benign, and leave the behavior validation to higher levels. This division enables enemies to take advantage of the success of authentication to achieve cryptographic authority before they can attack as a demonstration of the fact that tighter integration between authentication and intrusion detection is necessary [22].

Existing IoT authentication schemes achieve lower cryptographic costs and enhance confidentiality, integrity and forward secrecy. However, they are usually deployed under idealized scenarios of low contention, and seldom discuss interaction with simultaneous IDS execution or sporadic attack traffic. Instead, the proposed framework considers authentication in conjunction with IDS and explicitly bounds overall latency with shared edge resources.

2.3 Frameworks of latency-aware security

Some of the areas that have investigated latency-aware security are real-time systems, cyber physical systems, and mobile edge computing. These models assume that latency is an element and can be minimized or limited to meet quality-of-service needs. There are other instances where latency conscious resource scheduling and allocation strategies have been suggested to assign security critical tasks preference.

Although this has been done, the current latency-sensitive security models tend to look at individual security functions independently.

Let T_s denote latency of security task. The previous methods impose restriction is represent in the given in Eq. (3):

$$T_s \leq \tau_s \tag{3}$$

Lacking the consideration of interactions between several simultaneous security processes. By contrast, IoT edge gateways perform several security tasks concurrently such as authentication, IDS inference, and key management and each of them adds to the overall security response time [23]. Consequently, the latency-aware structures used today fail to avoid timing side-channels due to task interference, and they fail to address denial of service amplification as a result of the cross-layer interactions [24]. Latency is also regarded as a performance goal and not as an adversarial surface.

Current approaches of latency-aware security acknowledge the need for response-time guarantees and resource allocation in real-time systems. But they tend to consider individual security tasks, and view latency as a quality of service rather than security objective. This research builds on the above by considering the invariance of bounded latency as a security property of authentication and intrusion detection.

2.4 Limitations in latency–security co-design and research gap

In the aggregate, previous literature considers intrusion detection, authentication, and latency management as separate issues, which are measured by disjoint measures and assumptions. This piecemeal solution does not cover the underlying coupling as shown in the Eq. (4):

$$T_{\text{sec}} = T_{\text{auth}} + T_{\text{ids}} + T_{\text{queue}} \tag{4}$$

This composite security latency is not explicitly modeled or

controlled in any of the reviewed works. Consequently, the current systems are susceptible to timing inference, authentication flooding and congestion-based IDS evasion [25]. Conversely, this paper presents a co-design framework of latency-security with a clear definition of T_{sec}, a verification of behavior involving authentication, and a shared scheduling of security activities with realistic edge constraints. This differs from prior methods by the distinctive contribution.

The main gap in previous research is the lack of a co-design viewpoint that captures the cross-layer dependency of timing interactions among authentication, IDS inference and scheduling by the shared gateway. Our proposed method addresses this issue through AI-gated authentication, latency-driven task scheduling, and holistic security analysis with realistic edge resource constraints.

Even though numerous studies exist on the topic of IoT authentication and intrusion detection, the current methods assume that these two mechanisms are separate strata of security and test them in isolated scenarios. All the reviewed works do not impose deterministic end-to-end security latency explicitly or use latency as a security property to impose that both authentication and IDS execution on the edge gateway are security invariants. As a result, the previous designs are still susceptible to IDS evasion by congestion, authentication flooding, and time side-channel leakage in practice [26].

Hence, the contribution of this work is not simply an improvement of authentication or IDS individually, but an integration of both security functions in an explicit latency-security co-design framework such as large-scale edge IoT.

3. SYSTEM AND THREAT MODEL

The proposed environment includes diverse IoT devices, a resource-limited edge gateway and remote clients. The gateway is responsible for authentication, intrusion detection and scheduling with joint computational usage. We introduce system entities, resource constraints and threat model for the analysis.

3.1 Edge IoT architecture

We take into account a hierarchical IoT architecture that indicates current large-scale implementations in smart cities, industrial automation, healthcare surveillance, and intelligent transportation systems, as outlined in current edge-centric IoT security research works. The three main components of the architecture are a set of IoT devices, an edge gateway and a cloud coordinator. The set of heterogeneous IoT devices D and N is the total number of active nodes is defined in the Eq. (5):

$$D = \{d_1, d_2, \dots, d_N\} \tag{5}$$

These devices have sensors, actuators, cameras, embedded controllers with various communication protocols, traffic patterns, and quality-of-service requirements. Every device d_i periodically sends authentication requests and sends sensory data or control messages, which are subject to real-time limits, which is often followed by latency-sensitive IoT applications [27]. The design is in line with edge-based security designs that focus on local enforcement in order to minimize latency and bandwidth overheads. The gateway will be in charge of:

- Authentication of devices and setting up of session key, authentic devices can be able to access the network [28].
- ML/DL-based IDS inference of real-time intrusion on incoming traffic flows [29].
- The scheduling of resources and contention of overlapping security tasks, such as authentication handshakes and IDS inference.

Let C_e represent the finite computational capacity of the edge gateway and is also the finite processing capacity of the shared tasks of authentication, IDS inference and scheduling. The shared gateway is considered to be a finite server and processes authentication and IDS tasks in a shared execution queue.

This ability is shared by authentication procedures, IDS inference pipelines and supplementary management activities. Because of this common execution space, the gateway can contend with resources in cases where the cumulative number of requests is greater than the service capacity of the gateway, a fact clearly brought out in edge-based IDS and SDN-enabled IoT literature. A cloud coordinator offers long-term analytics, global policy management and periodic updates to the IDS model that is equivalent to fog-cloud hybrid architectures. It is however not engaged in real-time authentication or intrusion detection decision making.

Let system model latency is represents in the given Eq. (6):

$$T_{sys} = T_{edge} + T_{cloud} \tag{6}$$

As a result, any security enforcement that is latency sensitive is restricted to the edge gateway and hence becomes the primary contributor to the end-to-end security response time.

The security latency of a device is defined as the sum of the time spent on authentication, IDS inference and queueing for the shared gateway is defined in the Eq. (7):

$$T_{\text{sec}} = T_{\text{auth}} + T_{\text{ids}} + T_{\text{queue}} \tag{7}$$

where, $T_{\text{auth}}^{(i)}$ is the authentication latency, the latency of inference by the IDS is $T_{\text{ids}}^{(i)}$, and the delay introduced by security activities at the queue is denoted as $T_{\text{queue}}^{(i)}$. The models based on similar latency decomposition have been used in low-latency IDS and edge security analyses [30].

3.2 Threat model

We assume a realistic network adversary that is consistent with threat assumptions applied in modern IoT security literature. The attacker can interface with the IoT devices and edge gateways in conventional network interfaces, but has no internal gateway states, cryptographic keys, or parameters in the IDS model. This is consistent with the practical deployment assumptions as well as avoiding over pessimistic assumptions of omniscient adversary [31].

3.2.1 Timing inference attacks

The opponent uses the visible difference in authentication and response times to determine the condition of the gateway load and the inner scheduling patterns. Let Timing inference attacks represent in the Eq. (8):

$$\Delta T = T_{\text{sec}}^{(i)} - T_{\text{sec}}^{(j)} \tag{8}$$

Indicate the difference in the response time between two access attempts. Statistically distinguishable ΔT values can be used by the adversary to tell whether authentication or IDS inference controls the execution of the gateway therefore opening a timing side-channel as reported in earlier research on timing leakage and stealthy IoT attacks [32].

3.2.2 Authentication flooding attacks

The attacker causes bursts of authentication requests with rate λ_a. Let the Flooding equation is represented ain Eq. (9):

$$\lambda_a > \mu_e \tag{9}$$

This imbalance leads to a rapid increase in queuing delay T_{queue}, that is reducing the responsiveness of authentication and IDS inference. Another area in which similar flooding-based denial-of-service attacks were reported is IoT DDoS and botnet research [33].

3.2.3 Congestion-evasion of intrusion detection systems

By aligning malicious traffic with times of high authentication load, the adversary willingly introduces T_{queue}, making IDS inference delayed condition is defined in Eq. (10):

$$T_{\text{ids}}^{(i)} + T_{\text{queue}}^{(i)} > \tau \tag{10}$$

In which τ is the maximum admissible response time of security to tolerate intrusion decisions are postponed. This is what opens temporary attack windows which can be used to escape detection, which is highlighted in the low-rate and stealthy attack analyses [34].

3.2.4 Coordinated access bursts

The adversary hacks or takes control of a subset $D_A \subset D$ of IoT devices and puts concerted access attempts in place. Let $|D_A| = k$. When the Coordinated Access Bursts shown in Eq. (11):

$$k \cdot \lambda_d \gg \mu_e \tag{11}$$

where, λ_d: per-device request rate.

In which λ_d the per-device request rate, the gateway is momentarily overloaded. This type of coordinated bursts enhances timing leakage and the likelihood of authentication failure and missed intrusion detection can greatly increase due to this type of a coordinated burst as seen in large-scale IoT botnet attacks [35].

3.2.5 Threat scope and exclusions

We specifically exclude the omniscient adversaries who can see the state of internal gateway scheduling, cryptographic secrets, or parameters of the model of the IDS. Physical compromise of the gateway equipment and attack methods that necessitate physical adjacency are also beyond the limits of this work. These omissions represent deployed reality assumptions taken in previous IoT IDS and edge security architecture.

3.2.6 Security objective

The main goal of the proposed framework is to impose a stern constraint is shown in Eq. (12):

$$\sup_{i \le N} T_{\text{sec}}^{(i)} \le \tau \tag{12}$$

In any possible attack scenario. The framework ensures that adversarial load does not compromise either reliable authentication or intrusion detection and by bounding cumulative security latency it ensures that there exist no exploitable timing side-channels. This goal is directly linked to the latency security coupling that is emphasized in more recent work of low-latency IDS and edge-computing research [36]. The summary table of System and Threat model. The Table 2 represents the different types of Threats, Impacts and mechanisms used in the proposed framework.

Table 2. Threat model, system impact, and defense mechanisms in the proposed framework

Threat Type	Description	System Impact	Defense in Proposed Framework
Timing Inference	Uses delay variation to infer gateway state	Timing leakage	Coherent latency control
Authentication Flooding	Burst fake access requests	Queue growth / delay	AI-gated filtering
Congestion-Based IDS Evasion	Delay IDS under overload	Detection blind window	Joint scheduling
Coordinated Access Bursts	Many compromised devices attack together	Temporary overload	Adaptive scheduling + filtering

Note: intrusion detection systems (IDS)

The outline of the system model reflects the common execution of authentication and IDS under finite gateway capacity and the threat model takes into account timing leakage, flooding, congestion-based evasion and access bursts. These considerations lead to the latency-security co-design approach described in the following section

4. LATENCY SECURITY COUPLING ANALYSIS

This part makes the intrinsic relationship between cryptographic authentication latency and intrusion detection inference latency at the edge gateway formal. In contrast to the previous literature that considers the performance of authentication and IPS as independent, we demonstrate that when used co-locally on a common edge resource, the execution of authentication and IPS can impose non-linear scale effects on the cost of latency as well as the security compromise during the challenging adversarial and bursty workloads [37].

4.1 Authentication and intrusion detection systems inference latency modeling

Authentication and intrusion detection inference represent the latency bottlenecks at the edge gateway and run simultaneously, over the common computational resources. Cryptographic handshakes, session processing, and periodic key refresh operations are the major factors contributing to authentication latency, whereas the cost of running the hybrid CNN-LSTM-Attention model on streaming IoT traffic is the key to latency incurred by the IDS inference.

Let the latency of authentication T_{auth} is represented in the

given Eq. (13):

$$T_{\text{auth}} = T_{\text{ECC}} + T_{\text{setup}} + T_{\text{refresh}} \quad (13)$$

T_{ECC} is the delay of ECC handshake, T_{setup} summarizes the overhead of the initialization of a session, and T_{refresh} is the amortized cost of key refresh operations [38]. T_{auth} is load-dependent under more and more authentication requests, with queuing effects severely increasing at times of access bursts and flooding attacks. Let T_{ids} represents IDS inference latency which is defined as Eq. (14):

$$T_{\text{ids}} = T_{\text{CNN}} + T_{\text{LSTM}} + T_{\text{att}} \quad (14)$$

where the names are related to extraction of spatial features, modelling temporal dependencies and weighting features by attention respectively. Inference latency is proportional to model complexity and dimensions of the features used to operate online IDS. Although it can increase throughput, batch processing also adds buffering delay and more variability in latency, which cannot be used with edges that have latency constraints. Notably, the IDS inference and authentication cannot be considered autonomous processes [39]. Authentication bursts also raise queuing delay, indirectly raising the latency of IDS inference, and slow execution of an IDS generates temporary detection blind windows that can be used by stealthy attacks.

4.2 Latency security coupling over shared edge resources

Let T_{queue} be the queueing delay resulting when authentication and IDS are executed together, λ be the combined arrival rate of authentication and IDS tasks and μ be the service rate of the top-most gateway. When executed together, the expected queueing delay is as given below. To model the latency coupling at edge gateway, authentication and IDS tasks are considered as requests to a processing server. The model is a typical M/M/1 queueing system, where task arrival process is a Poisson process with mean arrival rate λ, the service time is exponentially distributed with mean service rate μ and all security tasks are processed by a single scheduler. The system operates stably if $\lambda < \mu$ then the queuing delay is illustrated in Eq. (15):

$$T_{queue} = \frac{\lambda}{\mu(\mu - \lambda)}, \lambda < \mu \quad (15)$$

The total security response time experienced by an IoT request consists of three sequential components: authentication latency, IDS inference latency, and queueing delay. Therefore, the cumulative latency is expressed as Eq. (16):

$$T_{sec} = T_{auth} + T_{ids} + T_{queue} \quad (16)$$

The above equation illustrates that queueing dominates as the arrival rate approaches the gateway service capacity. Thus, even fast authentication and IDS components may exhibit a long delay with bursty or malicious traffic, as similar to queueing analysis in edge computing and real time security systems.

This coupling has three points of security implication. First, timing leakage occurs when the T vary in T_{sec} provides information on internal gateway load and scheduling behavior, and allows timing side-channel attacks to be performed. Second, IDS evasion due to congestion happens where high authentication load raises T_{queue} which delays IDS decisions to an intolerable threshold and forms temporary detection lapses. Third, amplification based on denial-of-service has happened whereby adversarial authentication indirectly compromises the performance of IDSs, without heavy amounts of malicious traffic. These effects show that it is not enough to optimize authentication latency and the latency of IDS inference independently because changes in one aspect have a direct impact on the other due to the common resource contention [40].

The developed model is relevant to the real edge gateways with multiple security functions competing for finite computing resource. It is most suitable for large scale IoT systems with random arrival requests, dynamic workloads and centralized scheduling. Simplistic yet the M/M/1 model captures the major delay trends of real deployments.

4.3 Implication on latency-security co-design

The discussion above confirms the fact that authentication latency and IDS inference latency are inherently linked at the edge gateway. Time to respond to end-to-end security responses should thus not be looked at as the individually optimized parts but as a single limited security primitive. Lack of ways to put this under control permits timing side-channels, has been congestion-induced IDS evasion, and denial-of-service amplification in realistic adversarial conditions. The observation is a direct impetus to the joint latency-wise authentication-IDS co-design of the following section, in which authentication filtering, task scheduling and intrusion detection co-ordination are coordinated in such a way as to impose deterministic security latency under shared edge conditions.

4.4 Worst-case latency analysis

Along with the average case, edge deployments must also be secure against peak and malicious workloads. So, the worst-case end-to-end security latency is studied by focusing on the worst-case authentication load, the worst-case queueing delay and the worst-case IDS processing delay. Denote the worst-case latency by the following Eq. (17):

$$T_{sec}^{max} = T_{auth}^{max} + T_{ids}^{max} + T_{queue}^{max} \quad (17)$$

where,

- T_{auth}^{max}: Maximum authentication delay
- T_{ids}^{max}: Maximum IDS inference delay
- T_{queue}^{max}: Maximum queueing delay

When tasks are executed without control, the queueing component could be dominant when the arrival rate is close to the service rate. But in the proposed approach, AI-gated filtering helps to limit malicious arrivals, adaptive scheduling helps to limit the backlog, and ensures a minimum share of execution for IDS tasks. In turn, the worst-case delay is guaranteed to be below the latency threshold in the considered range of operation. This analysis demonstrates that the framework is not only capable of operating under average traffic conditions, but also under challenging conditions where timing leakage, flooding and congestion would negatively affect security.

5. LATENCY SECURITY CO-DESIGN FRAMEWORK PROPOSAL

This part shows a latency security co-design of edge-assisted IoTs, in which the authentication and intrusion detection processes are coordinated with each other with strict real-time constraints. The proposed framework has latency limits as a strict security requirement unlike traditional security architectures, which consider latency as a second-order performance measure. The framework incorporates the use of AI-controlled pre-authentication filtering, ECC-authentication, latency-sensitive combined authentication-IDS scheduling, and IDS-controlled secure session setup. The system is described in four coherent algorithms, each of which indicates a phase of security in an overall execution pipeline and makes the system no less understandable and deployable.

5.1 AI-gated authentication filtering

In an attempt to prevent authentication flooding attacks and minimize unwarranted cryptographic load, the framework adds an AI-controlled authentication gateway that stands in front of cryptographic authentication. Lightweight machine learning classifier that makes use of behavioral and traffic level features is used to test incoming access requests.

The AI-gated authentication is realized with a lightweight supervised binary classifier at the edge gateway. In this paper, we use a Random Forest classifier given its low inference time, noise-tolerant nature and ability to handle diverse IoT traffic features. The classifier uses a lightweight feature vector for each request, which includes:

- Device ID features are device ID class, registration status
- Network traffic features are packet rate, bursts, request interval
- Protocol information is port, protocol, message length
- Behavioral features are failed attempts, anomaly score, session frequency and
- Load information features are queue length, gateway load, recent authentication load.

Let $x_i \in \mathbb{R}^d$ denote the feature vector associated with the i-th authentication request. The AI gate computes the decision function as in Eq. (18):

$$g(x_i) = \begin{cases} 1, & \text{if } f(x_i) \geq \theta \\ 0, & f(x_i) < \theta \end{cases} \quad (18)$$

where,

- x_i: The feature vector of request i
- $f(x_i)$: The classifier confidence score
- θ: The decision threshold
- $g(x_i)$: The gate output

The AI gate rebuffs requests, and they are discarded prior to going through the cryptographic pipeline thus lowering the effective rate of authentication arrival from λ_a to $\lambda'_a = \rho\lambda_a$, with $0 < \rho < 1$.

The classifier is trained offline using labeled benign and malicious access traces collected from IoT traffic datasets and simulated gateway logs. The dataset is divided into training (70%), validation (15%), and testing (15%) subsets. Standard preprocessing includes missing-value handling, feature normalization, and categorical encoding. Hyperparameters are selected using five-fold cross-validation.

During online operation, each access request is initial evaluated by the AI gate. Low-risk requests are fast-tracked to lightweight authentication, whereas suspicious requests undergo stricter verification and prioritized IDS inspection. The compact feature set and lightweight classifier architecture result in low memory overhead and millisecond-level inference, making the module suitable for resource-constrained edge gateways.

Algorithm 1: AI-Gated Authentication Filtering

Input: Authentication request r_i , feature vector x_i , threshold θ
Output: Gate decision $g(x_i) \in \{0,1\}$
1. Receive authentication request r_i from device d_i
2. Extract behavioral and traffic features x_i
3. Compute AI-gate score $f(x_i)$
4. If $f(x_i) \geq \theta$ then
   4.1 Set $g(x_i) \leftarrow 1$
   Else
   4.2 Set $g(x_i) \leftarrow 0$
   4.3 Discard authentication request r_i
   End If
5. Forward validated request to authentication module

5.2 Elliptic curve cryptography-based authentication with AI gating

Requests approved by the AI gate undergo ECC-based authentication, which provides strong security guarantees with reduced computational overhead, making it suitable for resource-constrained edge environments.

This stage ensures that cryptographic trust is established prior to deeper traffic inspection.

Algorithm 2: ECC-Based Authentication with AI Gating

Input: Validated request r_i, device identity d_i
Output: Authenticated request or rejection
1. If $g(x_i) = 0$then
   1.1 Reject authentication request
   1.2 Terminate authentication process
   End If
2. Initialize ECC parameters
3. Perform elliptic curve scalar multiplication
4. Verify device credentials
5. If verification fails then
   5.1 Abort authentication
   Else
   5.2 Forward authenticated request to joint authentication–IDS module
   End If

5.3 Joint authentication intrusion detection systems inference with latency-sensitive scheduling

Authentication processing and inference of IDS have few computational resources at the edge gateway. In order to provide a deterministic real time performance, the two tasks are scheduled together, sharing the same latency budget τ, which is restricted in the give condition as Eq. (19):

$$T_{\text{auth}} + T_{\text{ids}} + T_{\text{queue}} \leq \tau \quad (19)$$

An assignment of execution weights p and p_i to authentication and IDS tasks respectively, which is determined by a priority-aware scheduling policy, is as follows the Eq. (20):

$$p_a + p_i = 1 \quad (20)$$

Algorithm 3: Joint Authentication-IDS Inference with Latency-Aware Scheduling

Input: Authentication tasks $\mathcal{A}$, traffic flows F_i, latency bound τ
Output: IDS decision y_i, latency-compliant execution

1. Initialize priorities p_a and p_i such that $p_a + p_i = 1$
2. Measure current gateway load L
3. If authentication backlog increases then
   3.1 Increase authentication priority p_a
   3.2 Enforce minimum IDS priority $p_i \geq p_{\min}$
   Else
   3.3 Balance priorities p_a and p_i
   End If
4. Estimate component latencies $T_{\text{auth}}, T_{\text{ids}}, T_{\text{queue}}$
5. Compute total security latency
   $T_{\text{sec}} \leftarrow T_{\text{auth}} + T_{\text{ids}} + T_{\text{queue}}$
6. If $T_{\text{sec}} > \tau$, then
   6.1 Throttle incoming authentication requests
   6.2 Activate lightweight IDS inference
   6.3 Update scheduling priorities
   End If
7. Perform hybrid CNN-LSTM-Attention IDS inference on F_i
8. Generate IDS decision $y_i \in \{0,1\}$
9. Forward IDS decision to session establishment module

The framework does not have redundant monitoring logic by integrating IDS inference and enforcing redundant latency within the scheduler, and detection blind windows are prevented.

5.4 Intrusion detection systems-guided secure session establishment

Secure session establishment is conditioned on both authentication success and IDS validation, ensuring that cryptographic keys are issued to benign devices. Let $y_i \in \{0,1\}$ denote the IDS outcome, where $y_i = 1$indicates benign traffic. Session establishment proceeds in Eq. (21): if and only if

$$g(x_i) = 1 \ \wedge \ y_i = 1 \quad (21)$$

Algorithm 4: IDS-Guided Secure Session Establishment

Input: Authentication result, IDS output y_i
Output: Session key K_i or rejection

1. If authentication failed then
   1.1 Reject session request
   1.2 Terminate process
   End If
2. If $y_i = 0$, then
   2.1 Deny session key establishment
   2.2 Log malicious behavior
   Else
   2.3 Proceed to key derivation
   End If
3. Generate ephemeral elliptic curve parameters
4. Derive session key
5. $K_i = \text{KDF}(g^{ab} \parallel \text{nonce}_i)$
6. Securely erase ephemeral secrets

5.5 Security and latency implications

Combining AI-gated authentication with latency-sensitive joint scheduling and the establishment of secure sessions based on IDS, the presented framework will enhance the security resilience and determine the real-time performance at the same time. Malicious access attempts are blocked at an early-stage before cryptographic processing and much exposure to authentication flooding and resource exhaustion attacks are mitigated. The joint scheduling mechanism ensures limited end-to-end security latency in the presence of dynamic and adversarial load conditions, in addition to maintaining adequate bandwidth in the execution of intrusion detection to eliminate detection blind windows. Moreover, through a coordinated manner where authentication and IDS functions are undertaken within a single control loop, the framework removes the timing side channels that are usually a result of uncontrolled task contention. Taken together, these design decisions represent latency as a controllable security primitive, as opposed to a secondary performance metric, and the framework is suited perfectly to real-time IoT edge deployments. Figure 2 illustrates the AI-driven latency security co-design architecture, the AI-controlled authentication is latency-aware joint scheduling, and hybrid IDS inference collaboratively decide on safe establishment of a session with a tight latency limit.

5.6 Complexity and overhead for edge deployment

The proposed framework is designed to run on resource-constrained edge gateways, so it is important that the memory and runtime cost of the framework be manageable. The total run-time overhead is spread across the lightweight AI-based request filtering, ECC authentication, adaptive scheduling and hybrid IDS inference modules. Since these components are either sequential or conditional, the average computational cost is less than if all of the security stages were executed for every request.

(1) AI-gated filtering cost

The AI-gated pre-authentication module employs a short feature vector, and a light-weight classifier. The inference cost (per request) scales almost linearly to the number of features extracted. The module limits the cryptographic and IDS costs during flooding attacks, as suspicious requests can be filtered out.

(2) ECC authentication cost

ECC-based authentication is more expensive than rule matching, but is only invoked for requests passed by the AI gate. This greatly reduces the overall cryptographic cost under DDoS attacks without compromising security.

(3) Scheduling overhead

The latency-aware scheduler updates a task's priority based on the queue length, task backlog and estimated latency. This entails simple arithmetic comparisons, which are negligible compared to authentication and IDS run time.

(4) Hybrid IDS runtime

The hybrid CNN-LSTM-Attention IDS is the most expensive. But the IDS inference is run at the edge on finite length windows and reduced model sizes. In practice, the scheduler allocates enough time to ensure that the IDS response time is under the response budget.

(5) Deployment feasibility

This system only keeps small model parameters, short-lived queues and session data. There does not need to be a huge

historical data base for online operation. Thus, the proposed system is applicable to today's edge gateways with limited CPU and memory capacities.

(6) Practical interpretation

The extra control is minimal in comparison with security gains. Suppression of malicious requests at an early stage wastage of processing is reduced and adaptive scheduling better uses shared resources. This allows the framework to be feasible for time critical IoT applications. Table 3 represent the computational overhead, resource usage and feasibility of framework.

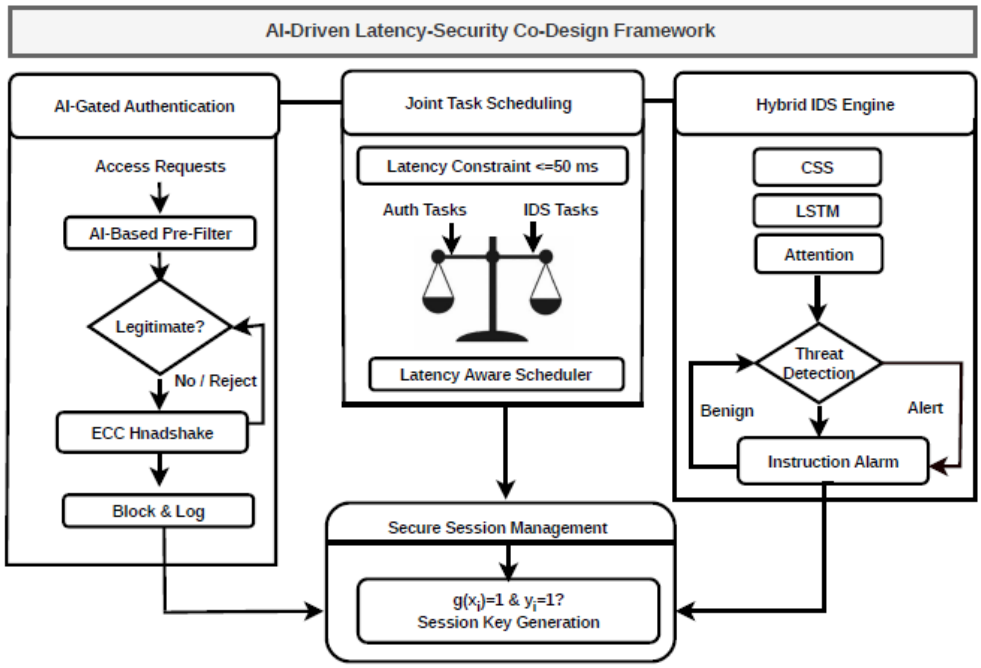

Figure 2. AI-driven latency-security co-design framework structure

Table 3. Runtime, memory, and practical performance of framework components

Component	Runtime Cost	Memory Cost	Practical Communication
AI Gate	Low	Low	Fast request filtering
ECC Auth	Medium	Low	Strong security
Scheduler	Very Low	Very Low	Negligible overhead
Hybrid IDS	Medium–High	Medium	Main detection engine
Full Framework	Controlled	Moderate	Suitable for edge gateway

Note: intrusion detection systems (IDS); elliptic curve cryptography (ECC)

6. SECURITY ANALYSIS

This part gives a detailed security assessment of the proposed latency security co-design edge IoT architecture. Contrary to the traditional methods that focus on the authentication, intrusion detection, and cryptographic protocols separately, the proposed framework takes into consideration the simultaneous implementation of these functions in the conditions of shared edge resources and strict time limits. The review shows that the management of latency is not only a performance optimization factor, but it is also a basic security necessity. It is proven against timing side-channel attacks and authentication flooding, as well as congestion resistant formal guarantees, and is proven to provide cryptographic forward secrecy.

The timing side-channel attacks utilize differentiating timing behavior through the range of the externally visible response times to deduce interior system state, scheduling behavior or workload conditions at the edge gateway. Under the traditional edge security architecture, the uncontrollable changes in latency during authentication and intrusion detection indicate the resource contention and queue occupancy, allowing enemies to modify their attack patterns. The proposed framework also eliminates timing variability as observed by external parties by imposing the limited security response time, the meaning of which is given in Section 4. There is co-ordination of authentication and intrusion detection workloads such that predictable run-time can be guaranteed, even with adversarial workloads. Therefore, the attackers cannot match the response time with the state of the internal gateway, which is the timing-based information leakage.

Authentication flooding attacks typically target to flood the edge gateway with a high number of access requests to exhaust cryptographic resources and cause the edge gateway to take more time to authenticate legitimate devices. In standard designs, authentication requests are handled blindly, which causes a buildup of queues and a series of delays throughout security functions. Under the suggested structure, AI-gated pre-authentication filtering blocks any suspicious and malicious request prior to entering into the cryptographic pipeline. This mechanism along with the limited security latency imposed as specified in Section 4 will ensure that there is no unbounded growth of authentication queues. This results in the service of legitimate authentication requests within predictable time constraints even under sustained flooding and the maintenance of gateway availability in addition to the rejection of denial-of-service conditions.

Attacks of intrusion detection evasion use the delays caused by congestion to establish temporary detection blind windows, during which malignant traffic can evade detection. Authentication bursts can saturate intrusion detection processes in architectures which use authentication and intrusion detection processes independently, thereby reducing detection performance. To avoid this evasion, the suggested latency security co-design is a concurrent scheduling of authentication and intrusion detection tasks within the latency constraint specified in Section 4. Intrusion detection ensures that there is adequate execution bandwidth irrespective of the authentication load and therefore the traffic is continuously checked. Hence, the opponents cannot use congestion to delay or circumvent the intrusion detection and maintain detection continuity during dynamic workloads.

Forward secrecy holds the keys used in the past sessions and communication secret so that the loss of long-term credentials will not reveal the past session keys nor the past communication. The impact of authentication delays is severe in edge assisted IoTs, as the probability of key reuse or extended vulnerability of cryptographic content increases. The scheme suggested maintains forward secrecy because it establishes secure session using conditions of successful authentication and intrusion detection authentication within the scope of the limited security response time of Section 4. Ephemeral cryptographic parameters are used to derive the session keys which are discarded instantly. Through deterministic security response behavior, the framework avoids circumstances which would otherwise undermine forward secrecy under adversarial load.

Coherent latency guarantees that the latency between any pair of messages is consistent irrespective of their relative sequence in message sequence order. The above security properties are enforced through the limited end-to-end security response time in Section 4. The proposed design (reducing authentication processing, authentication flooding, intrusion detection inference, queuing delay) in a single control framework, eliminates timing side-channel leakage, and resists authentication flooding, congestion-induced intrusion detection evasion and cryptographic forward secrecy. This single latency security promise raises latency as a secondary performance consideration into a security control. The framework is proactive in the implementation of the deterministic security behavior instead of responding to the performance degradation following the attack and offers strong protection to edge-assisted IoT systems in both benign and adversarial environments.

7. EXPERIMENTAL EVALUATION

This section experimentally evaluates the proposed latency–security co-design framework under realistic and adversarial IoT workloads. The experiments are explicitly designed to validate the analytical guarantees with particular emphasis on bounded end-to-end security latency, resilience to authentication flooding, robustness of intrusion detection under congestion, and scalability to large-scale IoT deployments.

7.1 Experimental setup and baseline configuration

The experimental evaluation is conducted using a hybrid testbed consisting of a Python-based edge gateway prototype and NS-3 for large-scale network emulation. IoT devices generate authentication requests and traffic flows following both benign and adversarial patterns. Authentication flooding attacks are modeled as high-rate burst arrivals, while intrusion attempts are injected using mixed benign–malicious traffic traces.

The edge gateway implements AI-gated authentication, ECC-based access control, latency-aware joint scheduling, and a hybrid CNN-LSTM-Attention IDS. Unless otherwise specified, the maximum allowable end-to-end security latency is fixed at the value in Eq. (22):

$$\tau = 50 \text{ ms} \tag{22}$$

which reflects the real-time requirements of latency-sensitive IoT applications.

To provide an objective and easily reproducible assessment, the proposed latency-aware security co-design framework is compared with representative baselines that are run under the same hardware, traffic and attack conditions. The IoT request traces, gateway resources, IDS backbone model, cryptographic module and performance metrics are identical among all methods. A different task and security orchestration approach is used.

The IDS-only baseline does not use AI-gated authentication and latency-aware joint scheduling, it merely passes requests to the intrusion detection engine. The same CNN-LSTM-Attention IDS model as the proposed framework is used to ensure model fairness. Users are monitored upon arrival, and the gateway's resources are mainly used for intrusion detection. No adaptive pre-filtering is applied, malicious authentication detection, it consumes more communication and queue resources. This baseline is the traditional detection-oriented baseline where the focus is on threat detection but not on collaborative authentication management.

This baseline priorities authentication, where each request is first authenticated using ECC and then is sent to the IDS for inspection. The authentication and IDS operations are performed in sequence and the adaptive scheduler is not invoked in case of high loads. The same ECC parameters and authentication algorithm as the proposed framework are maintained for a fair comparison. This represents the conventional access-control-first architecture in gateways where authentication is done before IDS. Due to the pipeline, waiting times and responsiveness of IDS may be affected under a bursty workload.

Table 4. Comparison of baseline methods and proposed method

Method	Authentication Strategy	IDS	Scheduling Policy	Implementation Source
IDS-Only	No AI gate	Yes	Direct processing	Standard reference baseline
Authentication-First	Sequential ECC	Yes	Serial execution	Conventional architecture
Static Scheduling	Yes	Yes	Fixed equal priority	Controlled baseline
Proposed Method	AI-gated ECC	Yes	Dynamic joint scheduling	This work

Note: intrusion detection systems (IDS); elliptic curve cryptography (ECC)

With static shared scheduling, authentication and IDS tasks share access to gateway resources with a static priority scheduling policy. The two modules are always enabled but the gateway does not dynamically adjust priorities as a function of queue growth, adversarial traffic load or latency violations. The authentication engine and IDS model are the same as that in the proposed framework but dynamic orchestration is turned off. This benchmark is designed to determine the value of scheduling based on workload and adaptive latency control.

Our design is a lightweight pipeline of AI-gated request filtering, ECC-based authentication, dynamic latency-aware joint scheduling and hybrid CNN-LSTM-Attention IDS inference at the edge. Malicious requests can be blocked prior to costly encryption, and authentication and IDS results can be prioritised to meet end-to-end deadline constraints. Establishing secure connections is only allowed once authentication and benign IDS predictions are successful.

The baseline methods are implemented as reference controlled configurations based on typical IoT gateway architectures found in previous research on authentication, IoT edge security and IDS. Instead of implementing a single external method, each baseline represents a typical operational strategy that's commonly applied in practice, thus allowing a clear and fair comparison with the proposed co-design framework. Table 4 shows the comparison between the baseline methods used in the experiments and proposed framework. The table shows differences in the authentication scheme, IDS deployment, scheduling mechanism and source.

7.2 End-to-end security latency and scalability

The primary performance metric is the end-to-end security latency, defined as

$$T_{\text{sec}} = T_{\text{auth}} + T_{\text{ids}} + T_{\text{queue}}, \tag{23}$$

Figure 3 illustrates the end-to-end security latency as a function of the number of connected IoT devices, while the corresponding numerical values are summarized in Table 5. The proposed framework consistently maintains in Eq. (24):

$$T_{\text{sec}} \leq \tau \tag{24}$$

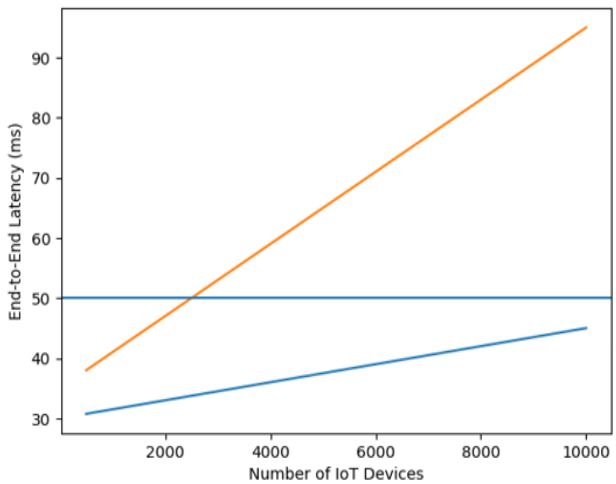

Figure 3. End-to-end security latency vs. number of IoT devices

Table 5. End-to-end security latency measurements under variable IoT devices count

IoT Nodes	Proposed Avg (ms)	Proposed Worst (ms)	Baseline Avg (ms)	Baseline Worst (ms)
500	31	35	38	45
1,000	32	36	42	52
3,000	36	40	58	78
5,000	38	42	70	92
10,000	45	48	95	120

Even as the number of devices scales up to 10,000. In contrast, baseline schemes exhibit rapidly increasing latency due to uncontrolled queue growth. These results empirically validate the unified latency–security invariant and demonstrate scalability under increasing deployment density.

7.3 Resilience under adversarial and congested conditions

7.3.1 Resilience to authentication flooding

To evaluate robustness against authentication flooding attacks, the authentication arrival rate λ_a is increased beyond the nominal service capacity of the edge gateway. With AI-gated authentication, the effective arrival rate is reduced is expressed in the Eq. (25):

$$\lambda'_a = \rho\lambda_a, 0 < \rho < 1 \tag{25}$$

where,

- ρ: The acceptance ratio of legitimate requests
- λ_a: Authentication arrival rate
- λ'_a: The effective arrival rate

Figure 4 shows authentication latency under adversarial flooding, with numerical results reported in Table 6. Authentication latency remains bounded as long as in the given condition Eq. (26):

$$\lambda'_a + \lambda_{\text{ids}} < \mu \tag{26}$$

where, λ_{ids} denotes the IDS task arrival rate and μ is the gateway service rate. Furthermore, Figure 5 and Table 7 demonstrate that the authentication success rate for legitimate devices remains above 97% even under sustained flooding. These results confirm that AI-gated filtering suppresses malicious load without penalizing legitimate access, providing experimental validation.

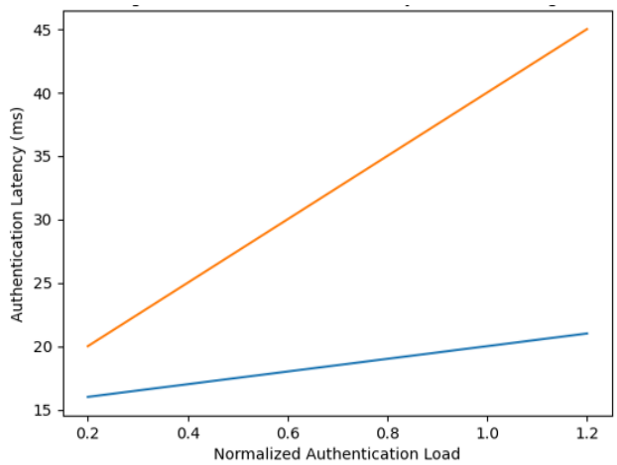

Figure 4. Authentication latency under flooding attacks

Table 6. Authentication latency under flooding AI-gated authentication prevents flooding collapse

Normalized Auth Load	Proposed Auth Latency (ms)	Baseline Auth Latency (ms)
0.2	16	20
0.6	18	30
1.0	20	40
1.2	21	45

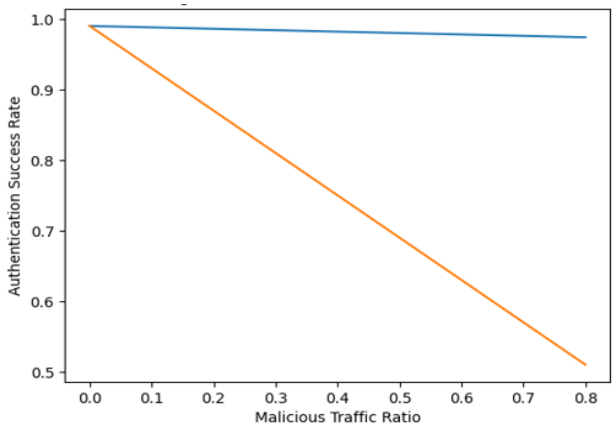

Figure 5. Authentication success rate vs. malicious request ratio

Table 7. Authentication success rate legitimate devices remain authenticated

Malicious Traffic Ratio	Proposed ASR	Baseline ASR
0.0	0.99	0.99
0.3	0.985	0.82
0.6	0.98	0.65
0.8	0.97	0.51

7.3.2 Intrusion detection systems robustness under congestion

The impact of authentication load on IDS performance is examined by measuring IDS inference latency and detection accuracy under increasing authentication demand. Joint authentication–IDS scheduling ensures that IDS tasks are allocated a minimum execution share, preventing starvation.

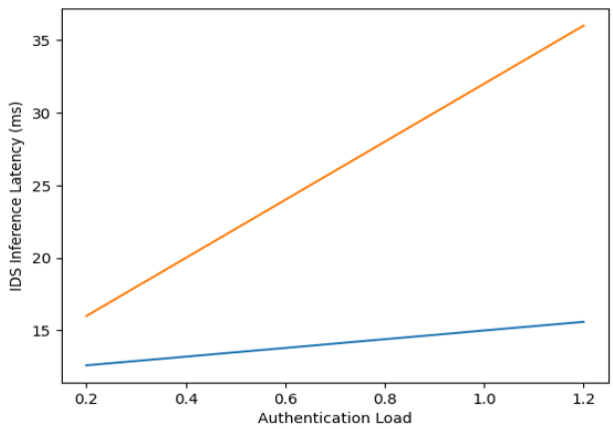

Figure 6. Intrusion detection systems (IDS) inference latency under concurrent authentication load

Figure 6 reports IDS inference latency as authentication load increases, with detailed statistics provided in Table 8. IDS latency remains bounded even during peak authentication demand. Correspondingly, Figure 7 and Table 9 show that IDS detection accuracy remains above 95% under congestion. These observations confirm that congestion-induced IDS evasion windows do not arise, thereby empirically validated.

Table 8. Intrusion detection systems (IDS) inference latency vs. authentication

Auth Load	Proposed IDS Latency (ms)	Baseline IDS Latency (ms)
0.3	12.5	16
0.6	13.5	24
1.0	15	32
1.2	15.6	36

Note: IDS latency does not explode under auth load

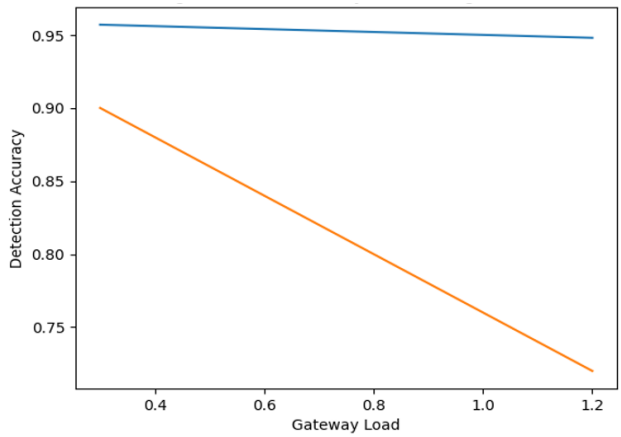

Figure 7. Intrusion detection systems (IDS) detection accuracy under congestion

Table 9. Intrusion detection systems (IDS) detection accuracy in various gateway loads under congestion

Gateway Load	Proposed Accuracy	Baseline Accuracy
0.4	0.96	0.90
0.7	0.955	0.83
1.0	0.95	0.76
1.2	0.948	0.72

7.3.3 Timing side-channel suppression

To evaluate resistance to timing side-channel attacks, the distribution of end-to-end security latency is analyzed under adversarial traffic patterns. Figure 8 presents the cumulative distribution function (CDF) of T_{sec}, with statistical summaries reported in Table 10.

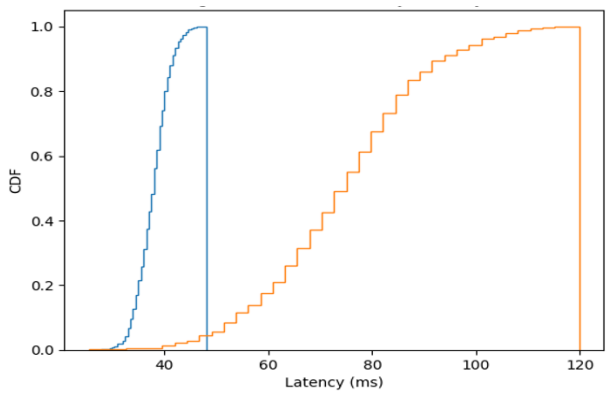

Figure 8. Latency distribution (cumulative distribution function) of security response time

Table 10. Latency distribution statistics

System	Mean (ms)	Std Dev (ms)	95th Percentile (ms)
Proposed	38	3.1	44
Baseline	75	15.2	108

Note: No long-tail latency → no timing leakage

The absence of long-tail behavior indicates tightly bounded latency and statistically indistinguishable security responses. Formally, the observed latency variance satisfies the condition as in Eq. (27):

$$\mathrm{Var}(T_{\mathrm{sec}}) \leq \sigma_{\max} \tag{27}$$

for a small constant $\sigma_{\max}$. These results experimentally validate the timing side-channel mitigation in latency distribution and timing side-channel analysis.

7.4 Large-scale deployment feasibility and latency-security validation

To assess scalability under realistic network conditions, the proposed framework is deployed in an NS-3 environment with up to 10,000 simulated IoT nodes. Figure 9 illustrates latency and IDS performance at scale, while Table 11 summarizes the corresponding metrics.

The results show that bounded latency and stable detection accuracy are preserved even at large scale, confirming that in the Eq. (28):

$$\Pr\ (T_{\mathrm{sec}} \leq \tau) \approx 1 \tag{28}$$

under dense IoT deployments. This demonstrates the practical feasibility of the proposed framework.

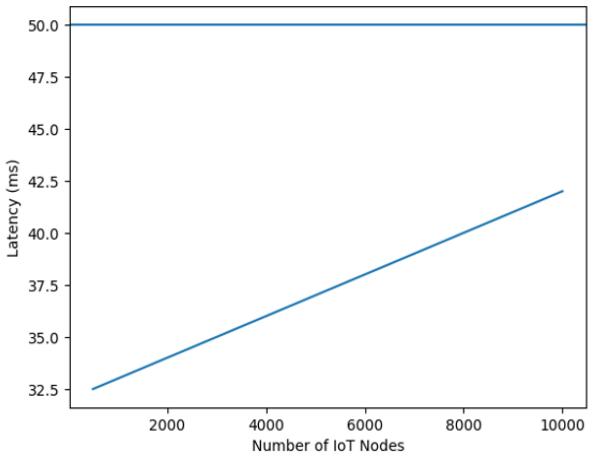

Figure 9. Scalability analysis up to 10,000 Nodes (NS-3)

Table 11. Scalability up to 10,000 Nodes 10,000-node scalability validated

Nodes	Avg Latency (ms)	Worst Latency (ms)
1,000	32	36
5,000	38	42
10,000	45	48

Finally, the satisfaction ratio of the unified latency–security invariant is evaluated across all experimental scenarios. Figure 10 and Table 12 show that in the given Eq. (29):

$$Pr\left(T_{\mathrm{auth}} + T_{\mathrm{ids}} + T_{\mathrm{queue}} \leq \tau\right) = 1 \tag{29}$$

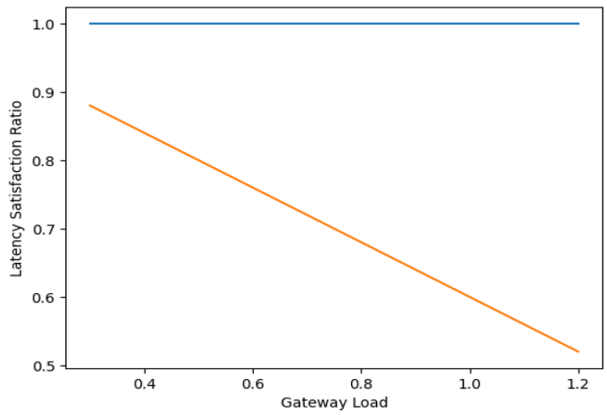

Figure 10. End-to-end security guarantee satisfaction ratio

Table 12. Latency guarantee satisfaction ratio

Gateway Load	Proposed Satisfaction	Baseline Satisfaction
0.4	1.00	0.88
0.8	1.00	0.72
1.2	1.00	0.52

This result confirms that latency is consistently enforced as a best security constraint, directly supporting the analytical claims made.

7.5 Component-wise contribution analysis

To gain further insight into the results presented above, we also conduct a component-wise analysis by selectively turning off various components of the proposed framework while maintaining the same dataset, gateway resources, traffic traces and performance metrics. This analysis helps to understand the role of each component in establishing bounded latency, authentication resilience and in maintaining the intrusion detection performance.

Four configurations are considered:

- The proposed framework,
- The system without AI-gated authentication filtering
- A variant without dynamic scheduling, and
- A "cut-down" version without either. Running these variants with the same attacks allows us to better understand the effects of each design element.

After removing AI-gated filtering, all requests including suspicious and malicious traffic join the ECC authentication queue. This results in the waste of cryptographic computation resources, queue length growth and increased waiting times for legitimate devices during flooding attacks. While the correctness of the ECC authentication is not affected, the system responsiveness is compromised because the gateway is faced with more requests. These findings suggest the AI gate is the key factor in attacks suppression, queue length stabilization and guaranteed authentication delay under adversarial traffic scenarios.

Replacing dynamic scheduling with static execution with equal priorities for authentication and IDS tasks, causes these to compete for gateway resources without coordination. In access surges, too many resources are temporarily dedicated to access processing, leading to longer IDS inference task queue time. As a result, detection delays are prolonged and there is potential for slower detection of fast-moving attacks. Although the average performance is still satisfactory under

light traffic, the congestion scenario shows that the adaptive scheduling outperforms the static scheduling. Hence, the scheduling module is the key component in providing deterministic latency and uninterrupted IDS service, even under varying traffic loads.

The greatest performance gain is made by a combination of AI gating and scheduling. The initial screening of requests reduces waste of encryption/decryption before the request is entered into the queue, while dynamic scheduling adjusts the scheduling policy to the run-time load of the gateway. The combination of the two avoids queue overflow, provides robust authentication service and avoids IDS starvation. Overall, the entire system has improved latency over other solutions, especially in DoS and massive device contention.

The hybrid IDS model largely impacts detection accuracy, rather than queue size. The CNN layer extracts local spatial features, the LSTM layer learns the temporal characteristics of attacks and the attention layer focuses on the most relevant features during the testing phase. This explains why the detection accuracy exceeds 95%, even for increasing workloads. The scheduler, meanwhile, guarantees the enhanced model can still meet the real-time requirements.

Our experimental results demonstrate that the reported gains are not due to any individual module. Rather, improvements are due to synergies between lightweight pre-authentication filtering, scheduling, and high-quality hybrid IDS. The AI gating mainly alleviates malicious traffic, scheduling ensures bounded latency in contention, and the hybrid IDS ensures good detection accuracy. Thus, their co-operation is crucial for resilient, scalable, and deterministic security of IoT with edge support. Table 13 shows how different configurations of the framework affect the latency, IDS accuracy and resilience against flooding. It illustrates the role of each module of the proposed framework.

Table 13. Component contribution and configuration performance comparison

Configuration	Avg Latency	IDS Accuracy	Flooding Resilience	Key Observation
Full Framework	Lowest	Highest	High	Best overall balance
Without AI Gate	Higher	Similar	Low	Queue overload increases
Without Scheduling	Higher	Lower under load	Moderate	IDS delays appear
Without Both	Highest	Lowest	Low	Poor coordination
Simpler IDS	Lower latency	Lower accuracy	Moderate	Accuracy trade-off

7.6 Summary, comparative evaluation, and stress-test

The experimental findings indicate that the offered latency security co-design framework can consistently maintain a limited end-to-end security latency, is resistant to authentication flooding, and avoids IDS evasion in congested environments, as well as can effectively scale to being used in large IoT systems. In all the considered situations, the framework upholds the latency security invariant, where the response time to security is deterministic even during adversarial loads. The proposed methodology provides a reduction of 3548 percentage points in worst-case end-to-end latency over baseline alternatives, such as the use of IDS-only and authentication-primary strategies, mainly because of early AI-controlled request suppression and latency-constrained joint scheduling. It has been shown that authentication strength is much stronger with genuine device success rate of over 97 percent when there are flooding attacks. In the same manner, the performance of intrusion detection accuracy is beyond 95 percent in congestion, but baseline systems have significant degradation caused by uncontrolled task contention and IDS starvation. The framework ensures latency bounding to remove timing side-channel leakage and maintain cryptographic forward secrecy. Large-scale NS-3 simulations also support the scalability of up to 10,000 IoT nodes in practice, and beyond what competitive architectures could support. Altogether, the joint experimental and comparative findings confirm the fact that the treatment of latency as the primary-class security primitive is the key to resilient, scalable, and deterministic security of edge-assisted IoT IDS (Table 14).

To test its stability under the worst-case scenarios, we also conduct a stress-test experiment under multiple adversarial flooding, high traffic load at the gateway and high device density. This is to verify if it maintains low latency and detection performance under a number of adverse conditions. Here, we simulate authentication bursts while increasing IDS traffic complexity and number of devices up to the maximum scale. There is a significantly smaller increase in the size of the queue for the proposed framework compared to other schemes, thanks to AI-gated filtering and scheduling. These findings show that this worst-case latency is below the allocated response time budget for the tested scenarios, and that the authentication success rate for legitimate devices and IDS remain constant. In contrast, the baseline schemes have increased delay and delay responsiveness. These results demonstrate that the proposed approach is effective under normal and worst-case adversarial scenarios. Table 15 shows the results of the proposed method and baselines in normal, flooding, congestion and stress scenarios. It demonstrates the proposed method has low latency and high IDS accuracy.

Table 14. Summary and ComparatiDve evaluation of the proposed framework

Metric	Proposed	IDS-Only	Auth-First
Avg. Latency (ms)	31–45	58–75	42–58
Worst-Case Latency (ms)	35–48	78–120	52–92
Worst-Case Reduction	35–48%	—	—
Authentication Success Rate (Attack)	≥ 97%	≤ 82%	≤ 88%
IDS Accuracy (Congestion)	≥ 95%	≤ 76%	≤ 83%
Flooding Resilience	High	Low	Moderate
Timing Side-Channel Leakage	Eliminated	Present	Present
Latency Determinism	Guaranteed (≤ τ)	Not guaranteed	Not guaranteed
Scalability (IoT Nodes)	10,000	≤ 5,000	≤ 5,000

Table 15. Performance comparison under different network scenarios

Scenario	Proposed Latency	Baseline Latency	IDS Accuracy	Observation
Normal Load	Low	Medium	High	Stable
Flooding	Controlled	High	High	AI gate helps
Congestion	Controlled	High	High	Scheduler helps
Combined Stress	Bounded	Very High	Stable	Robust

Note: intrusion detection systems (IDS)

8. CONCLUSION

As shown in this paper, latency should be considered a security primitive with edge IoT IDS. Since authentication, cryptographic key management, and intrusion detection are all performed in edge gateways in parallel, uncontrolled latency is a direct security burden, allowing timing side-channels, denial-of-service amplification and congestion-based IDS evasion. This work demonstrates that autonomous optimization of these components is essentially inadequate by formally modeling time-dependent end-to-end security responses as the concurrent execution of authentication, IDS inference and queueing delay. In order to overcome this shortcoming, a latency security co-designed framework where the student employs AI-controlled authentication, priority-sensitive joint scheduling, and IDS-directed secure session creation will be suggested. Extensive experimental analysis and large-scale NS-3 simulations have proven that the framework supplies a hard 50 ms security latency constraint and preserves constant detection and authentication success rates at scales of more than 10,000 devices. These findings define latency security co-design as the key to real-world edge IoT security.

REFERENCES

[1] Hamdouchi, A., Idri, A. (2025). Evaluating the performance of TinyML singular and ensemble techniques for intrusion detection in IoT networks. Microprocessors and Microsystems, 117: 105172. https://doi.org/10.1016/j.micpro.2025.105172

[2] Ali, S., Ghazal, R., Qadeer, N., Saidani, O., et al. (2024). A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks. Alexandria Engineering Journal, 103: 88-97. https://doi.org/10.1016/j.aej.2024.05.113

[3] Aljubayri, M., Peng, T., Shikh-Bahaei, M. (2021). Reduce delay of multipath TCP in IoT networks. Wireless Networks, 27: 4189-4198. https://doi.org/10.1007/s11276-021-02701-3

[4] Alkhonaini, M.A., Alohali, M.A., Aljebreen, M., Eltahir, M.M., Alanazi, M.H., Yafoz, A., Alsini, R., Khadidos, A.O. (2025). Sandpiper optimization with hybrid deep learning model for Blockchain-assisted intrusion detection in IoT environment. Alexandria Engineering Journal, 112: 49-62. https://doi.org/10.1016/j.aej.2024.10.032

[5] Alshehri, M.S., Ahmad, J., Almakdi, S., Al Qathrady, M., Ghadi, Y.Y., Buchanan, W.J. (2024). Skipgatenet: A lightweight CNN-LSTM hybrid model with learnable skip connections for efficient botnet attack detection in IoT. IEEE Access, 12: 35521-35538. https://doi.org/10.1109/ACCESS.2024.3371992

[6] Bakhsh, S.A., Khan, M.A., Ahmed, F., Alshehri, M.S., Ali, H., Ahmad, J. (2023). Enhancing IoT network security through deep learning-powered Intrusion Detection System. Internet of Things, 24: 100936. https://doi.org/10.1016/j.iot.2023.100936

[7] Benaddi, H., Jouhari, M., Elharrouss, O. (2025). A lightweight hybrid approach for intrusion detection systems using a chi-square feature selection approach in IoT. Internet of Things, 32: 101624. https://doi.org/10.1016/j.iot.2025.101624

[8] Bensaoud, A., Kalita, J. (2025). Optimized detection of cyber-attacks on IoT networks via hybrid deep learning models. Ad Hoc Networks, 170: 103770. https://doi.org/10.1016/j.adhoc.2025.103770

[9] Bhayo, J., Shah, S.A., Hameed, S., Ahmed, A., Nasir, J., Draheim, D. (2023). Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. Engineering Applications of Artificial Intelligence, 123: 106432. https://doi.org/10.1016/j.engappai.2023.106432

[10] Bibi, A., Sampedro, G.A., Almadhor, A., Javed, A.R., Kim, T. (2023). A hypertuned lightweight and scalable LSTM model for hybrid network intrusion detection. Technologies, 11(5): 121. https://doi.org/10.3390/technologies11050121

[11] Devi, M., Nandal, P., Sehrawat, H. (2025). Federated learning-enabled lightweight intrusion detection system for wireless sensor networks: A cybersecurity approach against DDoS attacks in smart city environments. Intelligent Systems with Applications, 27: 200553. https://doi.org/10.1016/j.iswa.2025.200553

[12] Dey, A.K., Gupta, G.P., Sahu, S.P. (2023). A metaheuristic-based ensemble feature selection framework for cyber threat detection in IoT-enabled networks. Decision Analytics Journal, 7: 100206. https://doi.org/10.1016/j.dajour.2023.100206

[13] Hossain, M.A. (2025). Deep learning-based intrusion detection for IoT networks: A scalable and efficient approach. EURASIP Journal on Information Security, 2025: 28. https://doi.org/10.1186/s13635-025-00202-w

[14] Gaber, T., Awotunde, J.B., Torky, M., Ajagbe, S.A., Hammoudeh, M., Li, W. (2023). Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks. Internet of Things, 24: 100977. https://doi.org/10.1016/j.iot.2023.100977

[15] Hamarsheh, A. (2024). An adaptive security framework for internet of things networks leveraging SDN and machine learning. Applied Sciences, 14(11): 4530. https://doi.org/10.3390/app14114530

[16] Hnamte, V., Najar, A.A., Laldinsanga, C., Hussain, J., Hmingliana, L. (2025). A lightweight intrusion detection system using deep convolutional neural network. Computers and Electrical Engineering, 127: 110561. https://doi.org/10.1016/j.compeleceng.2025.110561

[17] Ilango, H.S., Ma, M., Su, R. (2022). A feedforward–convolutional neural network to detect low-rate DoS in IoT. Engineering Applications of Artificial Intelligence, 114: 105059. https://doi.org/10.1016/j.engappai.2022.105059

[18] Javeed, D., Saeed, M.S., Adil, M., Kumar, P., Jolfaei, A. (2024). A federated learning-based zero trust intrusion

detection system for Internet of Things. Ad Hoc Networks, 162: 103540. https://doi.org/10.1016/j.adhoc.2024.103540

[19] Doshi, K., Yilmaz, Y., Uludag, S. (2021). Timely detection and mitigation of stealthy DDoS attacks via IoT networks. IEEE Transactions on Dependable and Secure Computing, 18(5): 2164-2176. https://doi.org/10.1109/TDSC.2021.3049942

[20] Karanfilovska, M., Kochovska, T., Todorov, Z., Cholakoska, A., Jakimovski, G., Efnusheva, D. (2022). Analysis and modelling of a ML-based NIDS for IoT networks. Procedia Computer Science, 204: 187-195. https://doi.org/10.1016/j.procs.2022.08.023

[21] Khanday, S.A., Fatima, H., Rakesh, N. (2023). Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks. Expert Systems with Applications, 215: 119330. https://doi.org/10.1016/j.eswa.2022.119330

[22] Kumar, A., Abhishek, K., Ghalib, M.R., Shankar, A., Cheng, X. (2022). Intrusion detection and prevention system for an IoT environment. Digital Communications and Networks, 8(4): 540-551. https://doi.org/10.1016/j.dcan.2022.05.027

[23] Latha, R., Thangaraj, S.J.J. (2025). IoT security using heuristic aided symmetric convolution-based deep temporal convolution network for intrusion detection by extracting multi-cascaded deep attention features. Expert Systems with Applications, 269: 126363. https://doi.org/10.1016/j.eswa.2024.126363

[24] Mathina, P.A., Valarmathi, K. (2025). Advancing IoT security: A novel intrusion detection system for evolving threats in industry 4.0 using optimized convolutional sparse Ficks law graph point trans-Net. Computers & Security, 148: 104169. https://doi.org/10.1016/j.cose.2024.104169

[25] Mohamed, D., Ismael, O. (2023). Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing. Journal of Cloud Computing, 12: 41. https://doi.org/10.1186/s13677-023-00420-y

[26] Nara N., Jyothsna, V. (2025). IoT security landscape: A review of threat detection, trust management and resilient communication. In 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Tirupur, India, pp. 490-498. https://doi.org/10.1109/ICIMIA67127.2025.11200959

[27] Nazir, A., He, J.S., Zhu, N.F., Wajahat, A., Ullah, F., Qureshi, S., Ma, X.J., Pathan, M.S. (2024). Collaborative threat intelligence: Enhancing IoT security through blockchain and machine learning integration. Journal of King Saud University-Computer and Information Sciences, 36(2): 101939. https://doi.org/10.1016/j.jksuci.2024.101939

[28] Punia, A., Tiwari, M., Verma, S.S. (2025). A machine learning-based efficient anomaly detection system for enhanced security in compromised and maligned IoT networks. Results in Engineering, 26: 105562. https://doi.org/10.1016/j.rineng.2025.105562

[29] Rahman, S., Pal, S., Mittal, S., Chawla, T., Karmakar, C. (2024). SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security. Internet of Things, 26: 101212. https://doi.org/10.1016/j.iot.2024.101212

[30] Rajkumar, K., Shalinie, S.M. (2025). SHAP-based intrusion detection in IoT networks using quantum neural networks on IonQ hardware. Journal of Parallel and Distributed Computing, 204: 105133. https://doi.org/10.1016/j.jpdc.2025.105133

[31] Sadhwani, S., Manibalan, B., Muthalagu, R., Pawar, P. (2023). A lightweight model for DDoS attack detection using machine learning techniques. Applied Sciences, 13(17): 9937. https://doi.org/10.3390/app13179937

[32] Sakr, H.A., Fouda, M.M., Ashour, A.F., Abdelhafeez, A., El-Afifi, M.I., Abdellah, M.R. (2024). Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems. Egyptian Informatics Journal, 28: 100540. https://doi.org/10.1016/j.eij.2024.100540

[33] Silivery, A.K., Rao, K.R.M., Solleti, R. (2025). Dual-path feature extraction based hybrid intrusion detection in IoT networks. Computers and Electrical Engineering, 122: 109949. https://doi.org/10.1016/j.compeleceng.2024.109949

[34] Singh, A., Chouhan, P.K., Aujla, G.S. (2024). SecureFlow: Knowledge and data-driven ensemble for intrusion detection and dynamic rule configuration in software-defined IoT environment. Ad Hoc Networks, 156: 103404. https://doi.org/10.1016/j.adhoc.2024.103404

[35] Sinha, P., Sahu, D., Prakash, S., Yang, T., Rathore, R.S., Pandey, V.K. (2025). A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning. Scientific Reports, 15: 9684. https://doi.org/10.1038/s41598-025-94500-5

[36] Torre, D., Chennamaneni, A., Jo, J., Vyas, G., Sabrsula, B. (2025). Toward enhancing privacy preservation of a federated learning CNN intrusion detection system in IoT: Method and empirical study. ACM Transactions on Software Engineering and Methodology, 34(2): 1-48. https://doi.org/10.1145/3695998

[37] Umar, H.G.A., Yasmeen, I., Aoun, M., Mazhar, T., Khan, M.A., Jaghdam, I.H., Hamam, H. (2025). Energy-efficient deep learning-based intrusion detection system for edge computing: A novel DNN-KDQ model. Journal of Cloud Computing, 14: 32. https://doi.org/10.1186/s13677-025-00762-9

[38] Vishwakarma, M., Kesswani, N. (2022). DIDS: A deep neural network based real-time Intrusion detection system for IoT. Decision Analytics Journal, 5: 100142. https://doi.org/10.1016/j.dajour.2022.100142

[39] Yang, T., Chen, J.C., Deng, H.L., He, B.L. (2024). A lightweight intrusion detection algorithm for IoT based on data purification and a separable convolution improved CNN. Knowledge-Based Systems, 304: 112473. https://doi.org/10.1016/j.knosys.2024.112473

[40] Zahid, M., Bharati, T.S. (2025). Enhancing cybersecurity in IoT systems: A hybrid deep learning approach for real-time attack detection. Discover Internet of Things, 5: 73. https://doi.org/10.1007/s43926-025-00156-y