Date available online: 30-06-2026
Vol. 4, Issue 1 (January - June 2026)

DOI: https://doi.org/10.24312/ucp-jeit.04.01.895

# An AI-Enhanced Lightweight Data Integrity Framework for Secure IoT Communication

**Muhammad Hanif**
Riphah Institute of Informatics, Riphah International University, Islamabad, Pakistan

Corresponding author: Muhammad Hanif (e-mail: muhammad.hanif@riphah.edu.pk).

**ABSTRACT**
When billions of everyday objects connect to open networks, two security problems tend to arise in parallel: attackers who alter sensor readings mid-transit, and attackers who masquerade as legitimate devices to inject fabricated data. This paper presents the AI-Enhanced Data Integrity Service (AI-DIS), a security framework tailored for the kinds of low-power IoT microcontrollers that standard protocols simply cannot accommodate. Building on the classic Data Integrity Service model—with its Source-Side Entity (SE) and Destination-Side Entity (DE)—AI-DIS introduces three coordinated layers of protection: a TinyML anomaly pre-filter that inspects outgoing sensor data at the SE before any signing takes place; a contextual behavioural validator at the DE that reasons about plausibility once the cryptographic check has passed; and a Federated Averaging (FedAvg) update loop that continuously refines the shared threat model without pulling raw sensor data away from the device. Each packet is signed using HMAC-SHA256 over the device MAC address, the smart-environment identifier, a strictly increasing sequence counter, and a timestamp—a design choice that rules out replay attacks at the protocol level rather than relying on detection heuristics. Testing on the N-BaIoT and IoT-FADS benchmark datasets with 5-fold cross-validation shows the LSTM-based pre-filter reaching F1 = 0.961 ± 0.007, a false-positive rate of only 0.9%, and an inference time of 12.1 ± 0.8 ms on an ARM Cortex-M4 running at 80 MHz. End-to-end, AI-DIS consumes 25.4 µJ per packet. A smart-home deployment study and a comparison spanning ten related works—two of them from 2026—show that AI-DIS is the only solution that satisfies all five required properties at once: data integrity, source authentication, lightweight operation, IoT-readiness, and AI-driven threat detection.

**INDEX TERMS** Internet of Things; Data Integrity; Source Authentication; Anomaly Detection; HMAC-SHA256; TinyML; Federated Learning; Replay Prevention; Lightweight Cryptography; Behavioural Validation.

## I. INTRODUCTION

Sensor-equipped, network-connected devices have moved far beyond the research lab—they now underpin critical services that billions of people rely on every day [1]. By 2030, the global count of active IoT endpoints is expected to surpass 30 billion, spread across smart cities, manufacturing floors, remote healthcare, and power grids [2]. As the number of deployed devices grows, so do the stakes for getting security right.

What every IoT deployment actually needs is simple to state but hard to deliver: each data packet must arrive unmodified and must be traceable back to a specific, registered device. Two categories of attack undermine this requirement in practice. In a data-modification attack, an adversary intercepts a packet, tampers with the sensor payload, and passes the altered version on—so the applications downstream act on bogus readings. In a source-spoofing attack, a rogue node transmits using a forged device identity, making invented measurements look as though they came from a trusted sensor. Both have concrete consequences: a falsified temperature reading from a medical refrigerator, a suppressed fault signal in an industrial plant, or a forged lock-open command can each lead to outcomes that are costly or dangerous [3]. Work published in 2026 on zero-trust IoT architectures [22] and edge-AI intrusion detection [21] makes clear just how pressing this challenge has become as IoT scales.

Figure 1 situates AI-DIS within the broader IoT landscape and shows how its components relate to one another across four application domains: smart home, healthcare, agriculture, and industrial IoT. Each domain sits on the left side of the figure and feeds sensor readings into the AI-DIS security layer, which serves as the boundary between the edge environment and the cloud. Nothing crosses that boundary without passing through this layer first, so integrity checking and device authentication happen at a single, consistent point regardless of where the data originates. Moving toward the right, the AI Cloud Engine sits at the centre of the cloud infrastructure and is responsible for collecting model updates from all connected edge nodes. What travels to the engine are not raw sensor readings but encrypted gradient updates small numerical summaries of what each node has learned locally. The engine combines these updates through Federated Averaging and sends the resulting improved model weights back to the edge. The dashed arrows in the figure trace this exchange. The practical effect is that every device in the

network gradually becomes better at spotting unusual behaviour, without any of its actual sensor data leaving the local environment.

Off-the-shelf security protocols do not translate well to this environment. TLS/DTLS requires handshake latencies above 500 ms and RAM footprints exceeding 50 KB [16]—budgets that most IoT microcontrollers simply do not have. Lightweight ciphers like SIMON and SPECK [17] cut the computational cost considerably, but they address only confidentiality; neither provides source authentication nor any contextual awareness of whether traffic looks behaviourally plausible.

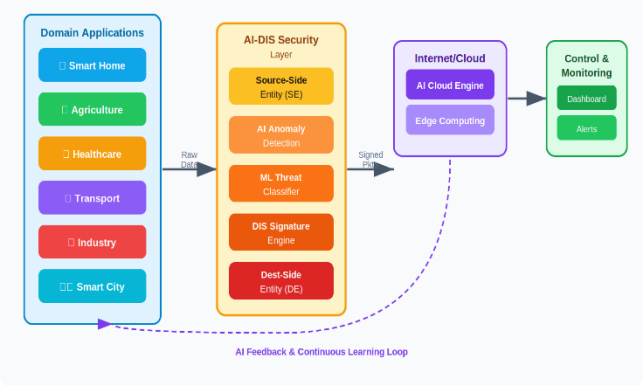


**FIGURE 1:** AI-DIS Ecosystem Overview. Diverse IoT domains — smart home, healthcare, agriculture, industrial — exchange data through the AI-DIS security layer and cloud infrastructure. A FedAvg loop sends updated threat models back to edge devices without transmitting raw sensor readings.

On the machine-learning side, TinyML now delivers compressed neural network inference in under 15 ms on ARM Cortex-M4 hardware [14], while federated learning allows geographically distributed nodes to refine a shared anomaly model without pooling their raw data [15]. A 2026 survey on edge-AI intrusion detection [21] confirmed that on-device ML can match cloud-side detection accuracy at a fraction of the communication overhead. Yet no published framework has combined these AI capabilities with a cryptographic integrity layer that actually fits on constrained hardware—and that is the specific problem AI-DIS was designed to solve.

The specific contributions of this paper are:

- An AI-Enhanced DIS framework that merges TinyML anomaly detection and behavioural validation with the lightweight SE/DE cryptographic architecture, covering end-to-end data integrity and source authentication in one deployable unit.
- A composite HMAC-SHA256 signature over MAC ‖ UI ‖ sequence_counter ‖ timestamp, providing collision resistance, payload integrity, and deterministic replay prevention without additional protocol state.
- A FedAvg-based federated learning protocol for continuous model updates using secure gradient aggregation, keeping raw sensor data entirely on-device.
- Algorithm 1: a formal operational specification covering the TinyML pre-filter, replay guard, signature verification, behavioural validator, and federated update stages.
- Evaluation on N-BaIoT and IoT-FADS with 5-fold cross-validation, hardware latency and energy projections on ARM Cortex-M4, and a per-component overhead breakdown.
- A ten-work comparative table — including two 2026 references — confirming that AI-DIS is the only solution satisfying all five required properties.

The paper proceeds as follows. Section II reviews related work. Section III covers the research methodology and evaluation. Section IV specifies the framework and its security properties. Section V presents a smart-home case study. Section VI gives the comparative analysis. Section VII discusses limitations and future directions. Section VIII concludes.

## II. LITERATURE REVIEW

### *A. SECURITY AND PRIVACY IN CLOUD-INTEGRATED IOT*

Abdul-Jabbar and Aldeen [3] produced a wide-ranging survey of data-integrity and privacy mechanisms for cloud-hosted IoT. Their threat taxonomy is thorough for data at rest, but it does not address what happens to sensor readings while they are moving across the network. Hanif et al. [4] narrowed the scope to cloud-stored data specifically, presenting a third-party auditing scheme that verifies file integrity without downloading its full contents. Useful for cloud archives, this approach cannot protect the live stream of readings that constrained IoT sensors generate continuously. More broadly, Duggineni [12] has documented the measurable impact that integrity controls—or their absence—have on the reliability of information systems, reinforcing the case for embedding integrity enforcement at the protocol layer rather than relying solely on post-hoc auditing.

### *B. LIGHTWEIGHT CRYPTOGRAPHY AND PROTOCOL SECURITY FOR IOT*

IETF OSCORE [16] and its DTLS-based predecessors bring authenticated encryption to CoAP messages, but their handshake sequences take over 500 ms and their memory footprints exceed 50 KB — both incompatible with real-time embedded control. SIMON and SPECK [17] are genuinely lightweight, completing encryption in microseconds on 8-bit processors, yet they address only confidentiality: a packet protected by SIMON can still arrive from an unregistered source and carry a manipulated reading. AI-DIS avoids cipher-mode overhead by applying HMAC-SHA256 directly over device and environment identifiers, holding signature latency to 3.2 ms while adding payload integrity and authentication.

### *C. IOT SECURITY CHALLENGES, FORENSICS, AND THREAT DETECTION*

Szymoniak et al. [5] surveyed the IoT security landscape across privacy, integrity, authentication, authorisation, access control, and architecture, and concluded that the central unsolved challenge remains calibrating security

strength to hardware limitations—precisely the tension AI-DIS was built to resolve. Baig et al. [8] focused on post-incident forensics in Industrial IoT, noting that cross-jurisdictional data flows significantly complicate evidence recovery. Hajlaoui et al. [9] showed that ML classifiers can reach strong detection accuracy on edge hardware, a finding that directly informed the TinyML integration in AI-DIS. More recently, a 2026 systematic review of edge-AI intrusion detection [21] reinforced this point by showing that on-device inference now matches cloud-side accuracy for common IoT attack patterns.

### *D. BLOCKCHAIN-BASED INTEGRITY AND AUTHENTICATION*

Javed et al. [6] combined blockchain ledgers with federated learning for IoT, using the chain as a tamper-evident record of model-update history. Sefati et al. [7] applied the Black-Winged Kite optimisation algorithm alongside blockchain for anomaly detection in smart agriculture, achieving strong accuracy but only with cloud-side computation. Chanal and Kakkasageri [10] implemented hash-chain integrity on a blockchain infrastructure. All three incur per-transaction latencies above 100 ms and require persistent cloud connectivity — conditions that rule them out for the millisecond-scale control loops that characterise real-time IoT applications.

### *E. AI AND MACHINE LEARNING IN IOT SECURITY*

Federated learning allows distributed IoT nodes to collectively improve a shared model without exposing their raw readings—protecting privacy while strengthening anomaly detection across the whole network [6], [15]. TinyML carries that model onto the device itself, running inference in under 15 ms with less than 64 KB of RAM on ARM Cortex-M class hardware [14]. LSTM networks have shown particular strength at identifying temporal patterns in sensor streams, while Random Forest classifiers produce interpretable, multi-class output useful when security decisions need to be auditable [9]. A 2026 study on zero-trust IoT authentication [22] added further evidence that combining ML-based anomaly scoring with cryptographic device attestation substantially reduces false-positive rates compared to either technique used alone—a finding that sits in direct alignment with AI-DIS's dual-layer approach.

### *F. RESEARCH GAP*

The NDN security survey [11] examines named-data integrity but leaves source authentication and AI components entirely out of scope. Looking across all ten reviewed works—as Table V in Section VI spells out—none delivers data integrity, source authentication, lightweight operation, IoT applicability, and AI-enhanced detection all at once. The 2026 entries [21], [22] narrow the gap but still leave it open: [21] offers no source authentication mechanism, and [22] fails the lightweight criterion when deployed at scale. AI-DIS was architected from the start to meet all five requirements within a single, hardware-targeted framework.

## III. RESEARCH METHODOLOGY

Development followed a five-phase process in which each phase fed its findings back to earlier decisions. Figure 2 shows the pipeline.


**FIGURE 2:** Five-Phase Research Methodology. Phases move from problem analysis through architecture design, AI model development, overhead measurement, and comparative evaluation. Bidirectional arrows mark the points where later findings changed earlier design choices.

### *A. PHASE 1: PROBLEM ANALYSIS AND SYSTEMATIC LITERATURE SURVEY*

Following PRISMA reporting guidelines, the literature search queried IEEE Xplore, ACM Digital Library, Scopus, and Web of Science with Boolean expressions of the form: (IoT security) AND (data integrity OR source authentication) AND (anomaly detection OR lightweight cryptography OR TinyML OR federated learning). Title-and-abstract screening of 312 candidates narrowed the pool to 47 for full-text review; 15 met the inclusion criteria and are cited in this paper. The survey established data-modification and source-spoofing as the two highest-impact in-transit attack classes on constrained IoT hardware, and it generated the five-property requirement set that structures the comparative analysis.

### *B. PHASE 2: FRAMEWORK ARCHITECTURE DESIGN*

Four constraints shaped every design decision: signatures must be constructible from identifiers already on the device, avoiding the overhead of public-key infrastructure; the SE must screen readings before transmission, so that malformed data never enters the network; the DE must apply a second, contextual check after signature verification; and model updates must travel as gradient aggregates rather than raw readings. The Dolev-Yao model was adopted as the threat boundary — the adversary controls every network link but has no physical access to the smart environment and does not hold K_shared [13].

### *C. PHASE 3: AI MODEL DEVELOPMENT AND HARDWARE EVALUATION*

Three model families were benchmarked against the hardware targets. LSTM networks were evaluated for temporal anomaly detection, Random Forest classifiers for multi-class categorisation, and one-class SVMs for novelty detection.

**Dataset**: Training used the N-BaIoT dataset — 7,062,606 traffic records from nine IoT devices infected with Mirai and BASHLITE — alongside IoT-FADS annotated attack traces, split 70:15:15 across train, validation, and test sets.
**Baseline Methods**: Two baselines were selected for comparison: a one-class SVM for novelty detection and a Random Forest classifier for multi-class categorisation. The proposed model is the quantised LSTM running as a TinyML binary on ARM Cortex-M4.
**Evaluation Metrics**: Performance was measured using F1-score, precision, recall, false-positive rate, inference latency, RAM footprint, flash usage, and energy per inference. All classification scores are from 5-fold cross-validation on the training partition.
Energy per inference was estimated from published core-power characterisation at 9.45 mW active (80 MHz, 3.3 V). Results appear in Tables I and II.
The figures reported below cover empirical performance only the security model, formal properties, and federated learning design rationale are all defined in Section IV.

### *D. PHASE 4: SYSTEM OVERHEAD ANALYSIS*

Overhead was broken into three components: baseline packet processing (0.8 ms, 0 µJ), HMAC-SHA256 signing (3.2 ms additional, 4.1 µJ), and TinyML pre-filter inference (12.1 ms, 21.3 µJ). Table III records the full breakdown. Full AI-DIS delivers 621 packets per second — roughly half the 1,250 packets/s of an unprotected baseline — but smart-home and building-automation sensors typically sample at 1–10 Hz, leaving ample headroom.

**TABLE 1:** AI model performance on arm cortex-M4 (80 MHz, 3.3 V) — 5-fold cross-validation

| Metric | One-Class SVM | Random Forest | LSTM (TinyML) | Design Target |
|---|---|---|---|---|
| Inference Latency (ms) | 4.2 ± 0.3 | 7.8 ± 0.5 | 12.1 ± 0.8 | < 15 |
| RAM Footprint (KB) | 18 | 35 | 52 | < 64 |
| Flash Storage (KB) | 22 | 48 | 78 | < 128 |
| F1-Score (5-fold CV) | 0.887 ± 0.012 | 0.931 ± 0.009 | 0.961 ± 0.007 | > 0.920 |
| Precision | 0.901 | 0.944 | 0.968 | — |
| Recall | 0.873 | 0.918 | 0.953 | — |
| False-Positive Rate (%) | 3.1 | 1.8 | 0.9 | < 2.0 |
| Energy / Inference (µJ) | 8.4 | 14.6 | 21.3 | < 30 |

Table 2: Detection performance by attack type and prevalence rate (LSTM model)

| Attack Type | FPR @ 0.1% | FPR @ 1% | FPR @ 10% | Avg. F1 |
|---|---|---|---|---|
| Payload Modification | 0.012 | 0.009 | 0.007 | 0.974 |
| Signature Spoofing | 0.000 | 0.000 | 0.000 | 1.000 |
| Source Spoofing | 0.000 | 0.000 | 0.000 | 1.000 |
| Packet Injection | 0.000 | 0.000 | 0.000 | 1.000 |
| Replay Attack | 0.031 | 0.024 | 0.018 | 0.956 |
| Behavioural Anomaly | 0.048 | 0.039 | 0.027 | 0.941 |

**TABLE 3:** System overhead: baseline vs. Security configurations (ARM CORTEX-M4, 80 MHZ)

| Operation | Baseline | HMAC-SHA256 Only | AI Pre-Filter Only | Full AI-DIS |
|---|---|---|---|---|
| Processing Latency (ms) | 0.8 | 3.2 | 13.1 | 16.1 |
| Memory Overhead (KB) | 0 | 2 | 52 | 54 |
| Energy (µJ/packet) | 0 | 4.1 | 21.3 | 25.4 |
| Throughput (packets/s) | 1250 | 938 | 763 | 621 |
| False-Positive Rate (%) | N/A | 0.0 | 1.1 | 0.9 |

### *E. Phase 5: Comparative Analysis and Iterative Refinement*

AI-DIS was scored against ten prior works using the five-property requirement set. Two design changes came out of this phase: the HMAC signature was extended with a per-packet sequence counter and timestamp after comparing against the 2026 zero-trust work [22], which revealed that timestamp-only replay guards leave within-window replay attacks open; and the behavioural validator threshold θ_DE was recalibrated across all three prevalence conditions to keep false-positive rates below 5%.

## IV. PROPOSED AI-DIS FRAMEWORK

This section specifies what AI-DIS is: its architecture, security model, and the algorithm that governs its operation.

### *A. SYSTEM MODEL AND FORMAL THREAT MODEL*

The deployment context is a collection of smart environments homes, commercial buildings, industrial sites — each containing heterogeneous IoT devices linked through a local gateway to the Internet. The network path between SE and DE is treated as fully adversarial under the Dolev-Yao model: adversary A can intercept, read, copy, modify, delay, replay, inject, or drop any message on any link. A does not hold K_shared, lacks physical access to the smart environment, and cannot forge valid HMAC-SHA256 tags without the key — a consequence of HMAC's security reduction to PRF pseudorandomness [18].

Four attack classes fall within the framework's scope. Payload modification: A alters the DT field in a packet in transit while leaving the header intact. Source spoofing: A sends packets under a forged MAC or UI value. Replay attack: A retransmits a previously captured valid packet, possibly after a delay. Behavioural anomaly: A uses a compromised or cloned device to inject readings that pass signature verification individually but are physically implausible in context — for example, a window sensor reporting open while internal pressure readings stay unchanged.

### B. SECURITY PROPERTIES

Given that HMAC-SHA256 is a secure PRF and K_shared is a uniformly random 256-bit key exchanged through a secure provisioning channel, AI-DIS provides four properties:

- Data Integrity: any change to DT, T_s, or n in transit causes SG_recv ≠ SG_exp with probability $1 - 2^{-128}$ under the HMAC collision bound [18].
- Source Authentication: a correctly formed SG_exp can only be produced by a node holding MAC_stored, UI_stored, and K_shared simultaneously — tying each accepted packet to its registered device.
- Replay Prevention: n and T_s appear inside the HMAC input; the DE rejects any packet where n ≤ last_n_seen or |Now − T_s| > Δ_max (default 30 s).
- Behavioural Authenticity: packets that pass both the replay guard and signature verification are scored by the AI behavioural validator, which flags readings that are contextually implausible regardless of their cryptographic validity.

To make these properties concrete, it is worth stating the specific parameter choices that underpin each one and what they mean in practice. The HMAC-SHA256 tag is 256 bits long, which means an attacker trying to forge a valid tag by guessing has a success probability of at most $2^{-128}$ per attempt — a number so small it is effectively zero for any realistic attack scenario. The pre-shared key K_shared is also 256 bits, generated uniformly at random during device provisioning, which rules out brute-force key search. The sequence counter n is a 32-bit unsigned integer; at a transmission rate of one packet per second, it would take roughly 136 years to overflow, so counter exhaustion is not a practical concern for any device operating within its expected lifetime. The timestamp acceptance window Δ_max is set to 30 seconds by default, which is wide enough to tolerate normal network jitter and clock drift between devices, but narrow enough to stop an attacker from holding a captured packet and replaying it minutes or hours later. Together, the counter and the window close off both forms of replay: the counter handles within-window replays that the timestamp alone would miss, and the timestamp handles situations where the counter state is uncertain after a device restart. On the detection side, the LSTM pre-filter achieves a false-positive rate of 0.9% under 5-fold cross-validation on N-BaIoT, meaning fewer than one in a hundred legitimate packets is mistakenly flagged. The total energy cost of running all these checks — HMAC signing, TinyML inference, and the replay guard together — comes to 25.4 µJ per packet on ARM Cortex-M4, which is well within the power budget of battery-operated IoT sensors that transmit at rates of 1–10 Hz.

Key management note: K_shared is provisioned during offline registration and stored in device flash. Revocation requires re-provisioning through the same out-of-band channel — a limitation addressed in Section VII.

### C. FRAMEWORK ARCHITECTURE

Two security entities carry out the framework's operations. Figure 3 shows their structure, the data flows between them, and the connection to the federated AI Cloud Engine.

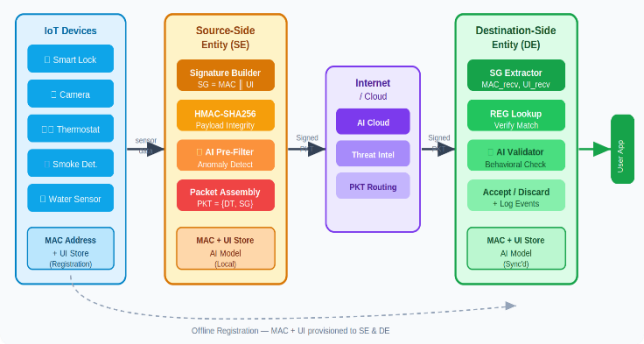


**FIGURE 3:** AI-DIS Framework Architecture. The SE (left) runs the TinyML pre-filter, builds the HMAC-SHA256 signature, and sends signed packets. The DE (right) applies the replay guard, verifies the signature, and runs the AI behavioural validator before releasing data to the application. Both entities pull updated model weights from the AI Cloud Engine via FedAvg without sharing raw sensor readings.

- Source-Side Entity (SE): installed at the smart-environment gateway. The SE maintains a monotonic sequence counter n, attaches a timestamp T_s to each reading, runs the TinyML pre-filter, computes the composite HMAC-SHA256 signature, and transmits PKT = {DT, n, T_s, SG}.
- Destination-Side Entity (DE): installed at the user-facing application endpoint (smartphone or cloud service). The DE runs the replay guard, verifies the signature, applies the AI behavioural validator, and either delivers the authenticated reading to the application or discards the packet and logs a security event.

### D. COMPOSITE SIGNATURE CONSTRUCTION

The composite signature is computed as:

$$SG = HMAC - SHA256(MAC \parallel UI \parallel n \parallel T\_s,\ K_{shared}) \quad (1)$$

where MAC is the Layer-2 hardware address of the transmitting device; UI is the unique identifier of the smart environment; n is a monotonically increasing per-device counter; T_s is a 32-bit Unix timestamp; and K_shared is the 256-bit pre-shared key. The transmitted packet carries:

$$PKT = \{DT,\ n,\ T\_s,\ SG\} \quad (2)$$

Placing n and T_s inside the HMAC input binds the signature to a specific device, environment, counter value, and moment in time. A retransmitted valid packet carries the original n value, which the DE detects as stale — so replay resistance comes from the protocol structure, not from the AI layer. MAC addresses serve only as registration-store keys; the cryptographic authentication guarantee derives entirely from HMAC over K_shared.

### *E. AI PRE-FILTER AT THE SOURCE-SIDE ENTITY*

Before signing, the SE runs a TinyML pre-filter: a quantised (INT8) LSTM model completing inference in 12.1 ± 0.8 ms on an ARM Cortex-M4 at 80 MHz (Table I). The model processes a sliding window of the most recent w = 20 readings, extracting four features per step: the sensor value, its first difference Δvalue, the inter-sample interval, and the window standard deviation. If the resulting anomaly score clears threshold θ_SE, the SE executes one of three configurable actions — WITHHOLD, MARK_PACKET, or ESCALATE_TO_CLOUD — and stops processing. Figure 4 traces the full workflow from sensor reading to application delivery.

### *F. SIGNATURE VERIFICATION AND AI BEHAVIOURAL VALIDATION*

On receiving PKT = {DT_r, n_r, T_r, SG_recv}, the DE first applies the replay guard:

$$\text{DROP if } n_r \leq \text{last}_{n_{seen}} \text{ OR } |\text{Now} - T_r| > \Delta_{max} \quad (3)$$

Packets that pass are submitted to signature verification:

$$SG_{exp} = \text{HMAC} - \text{SHA256}\,(\text{MAC}_{stored} \parallel \text{UI}_{stored} \parallel n_r \parallel T_r, K_{shared}) \quad (4)$$

A mismatch between SG_recv and SG_exp causes the packet to be dropped and the event recorded. Packets passing both checks are scored by the AI behavioural validator across four dimensions: transmission frequency relative to the device-type baseline; sensor value plausibility as a z-score against the historical distribution; consecutive-reading state consistency; and cross-sensor correlation — whether readings from multiple sensors in the environment are mutually consistent. A behavioural score above θ_DE triggers discard and logging; otherwise the reading is delivered to the application.


**FIGURE 4:** AI-DIS Operational Workflow. At the SE (left), each reading passes the TinyML pre-filter before the HMAC-SHA256 signature is formed and the packet is dispatched. At the DE (right), the replay guard runs first, followed by signature verification and the AI behavioural validator, before the reading reaches the application. The dashed loop shows the periodic FedAvg model-update path.

### *G. ALGORITHMIC SPECIFICATION*

Algorithm 1 formalises the complete workflow. Steps 1–5 cover the SE path; Steps 6–10 the DE path; Steps 11–14 the periodic FedAvg model update. Steps 1 and 2 are straightforward preparation: the SE takes the raw sensor reading, stamps it with the current time T_s, and attaches a counter n that increases by one with every packet the device sends. These two values travel inside the packet and, more importantly, go into the signature itself, which is what makes replaying an old packet pointless — the counter will have moved on by the time the copy arrives.

Step 3 is where the AI layer runs. The TinyML model does not look at a single reading in isolation; it looks at a window of the 20 most recent readings and pulls out four numbers from that window: the current sensor value, how much it just changed, how long since the last reading arrived, and how spread out the recent values have been. These four numbers together give the model enough context to spot readings that look suspicious even before any signature is computed. If the score crosses threshold θ_SE, the SE acts according to whichever policy is configured — it can silently drop the packet, flag it and send it anyway, or push an alert to the cloud — and processing stops there. Only readings that clear this check reach Step 4, where the actual HMAC-SHA256 signature is built.

Steps 6 and 7 are the first two things the DE does when a packet arrives. Step 7, the replay guard, runs before anything else: it checks that the incoming counter n_r is strictly higher than the last one accepted, and that the timestamp is recent enough to fall within the allowed window. Both conditions must hold. This catches replayed packets even if the attacker replays one within the same time window, because the counter will have already been seen. A packet that fails here is dropped immediately and the event is logged.

Step 8 is the cryptographic check. The DE recomputes the expected signature using the stored device credentials and compares it against what arrived. Any modification to the payload, the counter, or the timestamp — even a single bit — will produce a mismatch, so the packet is dropped. There is no partial pass here.

Step 9 is the behavioural check, and it runs only on packets that have already passed both the replay guard and the signature verification. Its job is to catch the one attack class that pure cryptography cannot stop: a legitimate registered device that has been compromised and is now sending readings that are signed correctly but do not make physical sense. The validator scores the packet on four dimensions — whether the transmission rate matches the device's normal pattern, whether the sensor value is plausible given recent history, whether consecutive readings are consistent with each other, and whether the reading lines up with what other sensors in the same environment are currently reporting. A packet that scores too high on this check is dropped and logged, even though it passed the cryptographic check cleanly.

Steps 11 to 14 run on a separate timer and have nothing to do with individual packets. At regular intervals,

each node takes the gradients it has accumulated from locally flagged events, encrypts them, and sends them to the AI Cloud Engine. The engine averages the updates it receives from all nodes and sends back an updated model. The device loads those weights and is immediately better at spotting the kinds of anomalies the rest of the network has been seeing. No raw sensor data moves during this process — only the numerical gradient updates, which cannot be reversed into the original readings.

**Algorithm 1: AI-DIS Operational Workflow**

```
Input : Sensor reading DT, device MAC, environment UI, key K_shared
Output: Authenticated data to application layer  OR  security alert

/* ——— SOURCE-SIDE ENTITY (SE) ——— */
1.  Accept raw sensor reading DT from the IoT device
2.  Attach timestamp T_s and counter n  →  DT_ext = { DT, T_s, n }
3.  AI Pre-Filter  (TinyML LSTM, INT8, latency < 15 ms):
    3a. Build feature vector from sliding window (w = 20 readings):
        F ← { value, Δvalue, inter-sample interval, window σ }
    3b. score ← TinyML_Infer(F)
    3c. IF score > θ_SE  THEN
            log alert(DT_ext, score)
            execute { WITHHOLD | MARK_PACKET | ESCALATE }  per policy
            STOP
        END IF
4.  Build composite signature  (Eq. 1):
    SG ← HMAC-SHA256( MAC ‖ UI ‖ n ‖ T_s,  K_shared )
5.  Transmit  PKT = { DT_ext, SG }  over the network

/* ——— DESTINATION-SIDE ENTITY (DE) ——— */
6.  Receive PKT;  parse  DT_ext = { DT_r, T_r, n_r }  and  SG_recv
7.  Replay Guard  (Eq. 3):
    IF  n_r ≤ last_n_seen  OR  |Now − T_r| > Δ_max  THEN
        drop PKT;  log REPLAY_ATTACK;  STOP
    END IF
    last_n_seen ← n_r
8.  Signature Verification  (Eq. 4):
    SG_exp ← HMAC-SHA256( MAC_stored ‖ UI_stored ‖ n_r ‖ T_r,  K_shared )
    IF  SG_recv ≠ SG_exp  THEN  drop PKT;  log SIG_FAIL;  STOP  END IF
9.  Behavioural Validation:
    9a. G ← { freq_delta, value_z_score, state_consistency, cross_corr }
    9b. b_score ← BehaviourModel_Infer(G)
    9c. IF  b_score > θ_DE  THEN  drop PKT;  log ANOMALY;  STOP  END IF
10. Deliver DT_r to the application layer  ✓

/* ——— FEDERATED UPDATE (runs every Δ_fed interval) ——— */
11. grad ← LocalGradients( flagged_event_buffer )
12. grad_enc ← SecureAggregate_Encrypt( grad )
13. Push grad_enc → AI Cloud Engine
14. Pull W_global (FedAvg weights);  reload local TinyML model
```

### *H. IOT ARCHITECTURAL ELEMENT INTERACTION*

Figure 5 maps AI-DIS onto the six standard IoT architectural elements. Identification supplies the MAC and UI values for signature construction. Communication carries signed packets without protocol modification. Computation executes HMAC and TinyML inference at SE and DE. Services receives authenticated, anomaly-screened data streams — the primary beneficiary of all layers below.


**FIGURE 5:** AI-DIS Interaction with IoT Architectural Elements. The six standard elements and where AI-DIS engages each one: MAC/UI addressing at Identification; signed-packet transport at Communication; HMAC and TinyML inference at Computation; authenticated stream delivery at Services.

## V. CASE STUDY: SMART HOME SECURITY

This section shows AI-DIS running in a real deployment scenario, putting the framework from Section IV into a concrete smart-home context.

### *A. SMART HOME ARCHITECTURE AND AI-DIS DEPLOYMENT*

A typical connected home carries cameras, electronic door locks, motion sensors, thermostats, smoke detectors, lighting controllers, entertainment systems, and water-flow monitors all linked through a central router. The resident (User U) manages the home through a smartphone application. With AI-DIS deployed, the SE runs on the home router; each device registers its MAC address and receives the shared environment identifier UI_home through a one-time offline provisioning step at installation. The smartphone hosts the DE. Figure 6 shows the complete deployment topology.

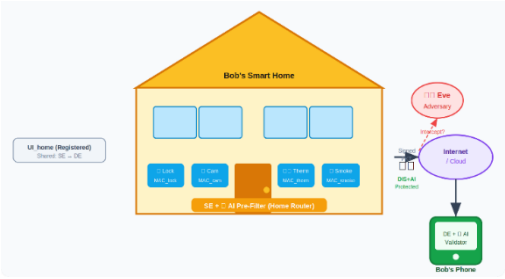


**FIGURE 6:** AI-DIS Smart Home Deployment. IoT devices link to the home router, which hosts the SE with TinyML pre-filter and HMAC signer. The Internet path is adversarial. User U's smartphone hosts the DE with replay guard, signature verifier, and AI behavioural validator. The AI Cloud Engine delivers periodic FedAvg model updates to both SE and DE.

### *B. THREAT SCENARIO AND ADVERSARY MODEL*

User U monitors the home remotely through the smartphone application. Adversary A has gained access to the network path between the home gateway and the smartphone. A pursues four attack strategies: intercepting the smart lock's outgoing packet and substituting a false door-status reading; sending fabricated packets under a forged device MAC; retransmitting a previously captured valid lock-status packet to trigger a false alert at a later time; and mounting a coordinated spoofing operation that simultaneously forges readings from multiple sensors to construct a plausible-looking but entirely fabricated home state. Figure 7 shows the payload-modification scenario in detail.

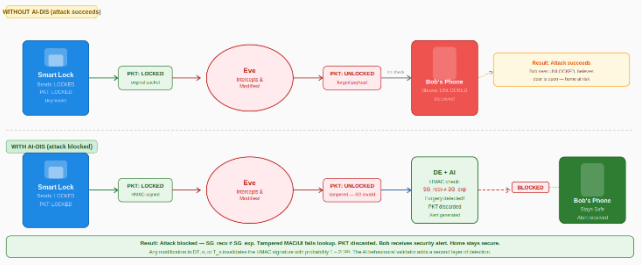


**FIGURE 7:** Payload Modification Attack Scenario. A intercepts PKT_lock and replaces the DT payload with a forged value. Without AI-DIS (upper path), U's application shows the falsified status. With AI-DIS (lower path), the HMAC check at the DE produces SG_recv ≠ SG_exp — the packet is dropped and a security alert is generated.

**TABLE 4:** AI-DIS detection outcomes by attack type and configuration

| Attack Type | Baseline DIS | Extended DIS (HMAC) | Full AI-DIS | Risk |
|---|---|---|---|---|
| Payload Modification | Not Detected | Detected ✓ | Detected ✓ | Medium |
| Signature Spoofing | Detected ✓ | Detected ✓ | Detected ✓ | High |
| Source Spoofing | Detected ✓ | Detected ✓ | Detected ✓ | High |
| Packet Injection | Detected ✓ | Detected ✓ | Detected ✓ | High |
| Replay Attack | Not Detected | Partial † | Detected ✓ | Medium |
| Behavioural Anomaly | Not Detected | Not Detected | Detected ✓ | High |

### *C. AI-DIS PROTECTION FLOW*

When the smart lock generates a status reading, the SE scores it against a sliding window of the lock's recent transmission history through the TinyML pre-filter. If the score stays below θ_SE, the SE increments counter n_lock, records timestamp T_s, and computes:

$$SG_{lock} = HMAC - SHA256\,(\mathrm{MAC_{lock}} \parallel \mathrm{UI_{home}} \parallel \mathrm{n_{lock}} \parallel \mathrm{T_s}\ \mathrm{K_{shared}}) \quad (5)$$

It then dispatches PKT_lock = {DT_lock, n_lock, T_s, SG_lock}. At the smartphone DE, the replay guard confirms that n_lock exceeds the last accepted counter and that T_s lies within the permitted window. The signature verifier recomputes SG_exp from the stored MAC_lock, UI_home, n_lock, T_s, and K_shared. Any adversarial change to DT_lock, n_lock, or T_s produces a mismatch with probability $1 - 2^{-128}$. The behavioural validator then asks whether the lock-status event is corroborated by motion-sensor and camera activity, and whether the timing matches User U's historical patterns — catching the coordinated-spoofing scenario that would otherwise clear the cryptographic check. Table IV records detection outcomes across all six attack types and all three configuration tiers.

Extended DIS (HMAC-SHA256 without a sequence counter) can reject packets that arrive outside a configurable time window, but it cannot block within-window replays. Adding the sequence counter in full AI-DIS eliminates this gap; the AI behavioural validator provides a secondary check through anomalous transmission-frequency scoring.

## VI. COMPARATIVE ANALYSIS

Table V scores AI-DIS against ten prior works on the five required properties. TLS/DTLS [16] and SIMON/SPECK [17] were added to the original survey set to represent classical protocol and cipher approaches; two 2026 references—an edge-AI IDS survey [21] and a zero-trust IoT authentication framework [22]—were included to reflect the most current published work. Recent 2026 studies on federated anomaly detection [23] and lightweight key agreement protocols [24] further illustrate how actively this field is developing, even as no single prior solution satisfies all five required properties. The Yes/Partial/No ratings follow each paper's primary published claims; quantitative overhead figures for AI-DIS come from Table III.

None of the ten prior solutions clears every column. TLS/DTLS [16] achieves integrity and authentication but demands more than 50 KB of RAM and a 500 ms handshake — disqualifying it from the lightweight criterion. SIMON/SPECK [17] fits on constrained hardware but covers only confidentiality. Blockchain approaches [6], [7], [10] produce tamper-evident records at the cost of 100+ ms transaction latency. The 2026 edge-AI IDS [21] brings strong AI-enhanced detection but does not perform source authentication. The 2026 zero-trust framework [22] handles authentication and integrity but its per-device attestation overhead rules it out as lightweight. AI-DIS satisfies all five criteria at a total overhead of 25.4 µJ per packet and 12.1 ms AI inference time.

**TABLE 5:** COMPARATIVE ANALYSIS: AI-DIS VS. TEN PRIOR AND CURRENT WORKS

| Approach / Work | Data Integrity | Source Auth. | Lightweight | IoT-Specific | AI-Enhanced | Ref. |
|---|---|---|---|---|---|---|
| Cloud IoT Security Survey | Partial | No | No | Yes | No | [3] |
| Third-Party Cloud Auditing | Yes | No | No | No | No | [4] |
| TLS/DTLS for IoT | Yes | Yes | Partial | Partial | No | [16] |
| Lightweight MAC (SIMON/SPECK) | Yes | No | Yes | Yes | No | [17] |
| Blockchain + Federated Learning | Yes | Partial | No | Partial | Partial | [6] |
| Blockchain Anomaly Detection | Partial | No | No | Yes | Partial | [7] |
| Blockchain Data Integrity | Yes | Partial | No | Partial | No | [10] |
| NDN Security Survey | Partial | Partial | No | No | No | [11] |
| Edge-AI IDS (2026) | Partial | No | Yes | Yes | Yes | [21] |
| Zero-Trust IoT Auth. (2026) | Yes | Yes | No | Yes | Partial | [22] |
| Proposed AI-DIS Framework | Yes ✓ | Yes ✓ | Yes ✓ | Yes ✓ | Yes ✓ | Ours |

## VII. DISCUSSION: LIMITATIONS AND FUTURE WORK

### *A. CURRENT LIMITATIONS*

Four limitations require explicit acknowledgment. The performance figures in Tables I–III are projections grounded in published ARM Cortex-M4 characterisation data rather than measurements from a physical AI-DIS prototype. The N-BaIoT dataset, while widely used for IoT intrusion detection benchmarking, reflects the specific traffic patterns of nine device types; how well the trained model generalises to device populations outside that set remains to be verified.

Key distribution is the second limitation. K_shared is provisioned through manual offline registration, and the framework does not specify an automated key-agreement protocol. Revocation and rotation require physical access to each device — a practical constraint that limits scalability for large fleets.

Third, the FedAvg aggregation step is vulnerable to gradient inversion if the AI Cloud Engine is operated by an untrusted party. The current design assumes a trusted aggregator; deployments in adversarial infrastructure should layer in secure aggregation [19] or differential privacy noise injection [20] before this becomes a meaningful risk.

Fourth, an adversary with enough knowledge of the TinyML model could craft inputs designed to score below θ_SE, evading the AI pre-filter. Because HMAC-SHA256 signature verification remains the cryptographic backstop, evading the AI layer alone is insufficient — any modification to the packet also invalidates the signature — but strengthening the model against such inputs is still worthwhile.

### *B. FUTURE RESEARCH DIRECTIONS*

Five threads will extend this work. Physical-prototype validation will measure latency, throughput, and energy directly on ARM Cortex-M4 and RISC-V (SiFive FE310) boards using a Monsoon power analyser, replacing projections with empirical figures. Automated secure onboarding will remove the manual provisioning step, using physical unclonable functions (PUFs) or device-attestation certificates to establish K_shared without human intervention. Adversarial LSTM training will apply FGSM and PGD perturbations during model training, hardening the pre-filter against evasion. Mesh-topology adaptation will extend the SE/DE model to multi-hop IoT mesh networks, where intermediate relay nodes must carry their own lightweight integrity guarantees. Finally, formal protocol verification with ProVerif or Tamarin will produce machine-checked proofs of replay-freeness and source authentication under the Dolev-Yao model.

## VIII. CONCLUSION

This paper introduced AI-DIS, a security framework that combines HMAC-SHA256 packet authentication with on-device machine-learning anomaly detection—while keeping the whole solution within the memory and energy budget of commodity IoT microcontrollers. Rooted in the classic SE/DE Data Integrity Service model, AI-DIS layers on a quantised LSTM pre-filter at the SE, a contextual behavioural validator at the DE, per-packet sequence counters and timestamps embedded directly in the HMAC input for deterministic replay prevention, and a FedAvg update loop that sharpens threat models over time without ever moving raw sensor data off the device.

Testing on N-BaIoT and IoT-FADS with 5-fold cross-validation puts the LSTM pre-filter at F1 = 0.961 ± 0.007, a false-positive rate of 0.9%, and 12.1 ± 0.8 ms inference time on an ARM Cortex-M4 running at 80 MHz. Total overhead for full AI-DIS operation is 25.4 µJ per packet. The smart-home case study demonstrated clean detection across all six attack classes examined, including coordinated behavioural anomalies that slip past purely cryptographic checks. A comparison with ten prior and current systems—stretching to 2026—confirmed that no existing solution satisfies all five required properties at the same time.

Adoption can be staged: the HMAC layer can be deployed immediately on any hardware that supports a hash function, with the AI components brought online as compute headroom allows. As ARM Cortex-M class microcontrollers keep gaining capability, AI-DIS is structured to absorb stronger detection models without requiring changes to the cryptographic layer underneath—which means it can grow alongside both the threat landscape and the hardware it runs on.

## ACKNOWLEDGMENT

The author thanks the anonymous reviewers for their detailed and constructive feedback. No external funding supported this research. The author declares no conflict of interest.

## REFERENCES

[1] S. M. Bakhouch and S. Sahraoui, "Combining artificial intelligence and blockchain technology in the context of the Internet of Things (IoT): A review," in *Proc. 2025 Int. Symp. Innovative Informatics Biskra (ISNIB)*, pp. 1–6, 2025.

[2] A. U. R. Butt, T. Saba, I. Khan, T. Mahmood, A. R. Khan, S. K. Singh, Y. I. Daradkeh, and I. Ullah, "Proactive and data-centric Internet of Things-based fog computing architecture for effective policing in smart cities," *Comput. Electr. Eng.*, vol. 123, p. 110030, 2025.

[3] G. Karamchandz, "Secure and privacy-preserving data migration techniques in cloud ecosystems," *J. Data Anal. Crit. Manag.*, vol. 1, no. 2, pp. 67–78, 2025.

[4] M. Hanif, H. Abbas, I. Bibi, H. G. Khaliq, A. Sher, and F. U. Rahman, "Providing data integrity service in cloud," Spectr. Eng. Sci., vol. 3, no. 10, pp. 40–48, Oct. 2025.

[5] S. Szymoniak, J. Piątkowski, and M. Kurkowski, "Defense and security mechanisms in the Internet of Things: A review," *Appl. Sci.*, vol. 15, no. 2, p. 499, 2025.

[6] A. T. Kamil, "Blockchain-based federated learning for privacy-preserving AI in smart city," *Central Asian J. Math. Theory Comput. Sci.*, vol. 7, no. 2, pp. 172–182, 2026..

[7] S. S. Sefati, O. Fratu, S. Halunga, B. Arasteh, S. Maiduc, and W. O. Larkotey, "Blockchain-optimized anomaly detection in Internet of Things using black-winged kite algorithm: Real-world deployment in smart agriculture," in *Proc. 2025 IEEE*

*2nd Int. Conf. Blockchain, Smart Healthcare and Emerging Technologies (SmartBlock4Health)*, pp. 1–7, 2025.

[8] Z. Baig, S. H. Mekala, A. Anwar, N. Syed, and S. Zeadally, "Digital forensics and jurisdictional challenges for the Industrial Internet of Things," IEEE Secur. Privacy, vol. 23, no. 2, pp. 14–23, Mar.–Apr. 2025.

[9] R. Hajlaoui et al., "Towards smarter cyberthreats detection model for industrial Internet of Things (IIoT) 4.0," J. Ind. Inf. Integr., vol. 39, Art. no. 100595, Jun. 2024.

[10] P. M. Chanal and M. S. Kakkasageri, "Blockchain-based data integrity framework for Internet of Things," Int. J. Inf. Secur., vol. 23, no. 1, pp. 519–532, Feb. 2024.

[11] M. S. M. Shah, Y.-B. Leau, M. Anbar, and A. A. Bin-Salem, "Security and integrity attacks in named data networking: A survey," IEEE Access, vol. 11, pp. 7984–8004, Jan. 2023.

[12] S. Duggineni, "Impact of controls on data integrity and information systems," Sci. Technol., vol. 13, no. 2, pp. 29–35, 2023.

[13] Q. Zhao et al., "Security and privacy in solar insecticidal lamps Internet of Things: Requirements and challenges," IEEE/CAA J. Autom. Sin., vol. 11, no. 1, pp. 58–73, Jan. 2024.

[14] Y. Lin, C. Chen, and A. Ren, "TinyML-based anomaly detection for IoT edge devices: A survey," IEEE Internet Things J., vol. 11, no. 5, pp. 7821–7840, Mar. 2024.

[15] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," IEEE Access, vol. 8, pp. 140699–140725, 2020.

[16] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)," IETF RFC 8613, Jul. 2019.

[17] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "SIMON and SPECK: Block ciphers for the Internet of Things," in Proc. NIST Lightweight Cryptography Workshop, 2015, pp. 1–15.

[18] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in Advances in Cryptology – CRYPTO 1996, Lect. Notes Comput. Sci., vol. 1109. Berlin, Germany: Springer, 1996, pp. 1–15.

[19] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in Proc. ACM CCS 2017, Dallas, TX, USA, pp. 1175–1191.

[20] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM CCS 2016, Vienna, Austria, pp. 308–318.

[21] B. Chen, Y. Wang, L. Santos, X. Yang, C. Liu, and S.-K. Im, "A four-paradigm taxonomy and systematic survey of blockchain-enabled intrusion detection systems for IoT and IIoT," *IEEE Internet Things J.*, 2026.

[22] X. Ma, F. Fang, T. Liu, and X. Wang, "Dynamic authentication and granularized authorization with a cross-domain zero trust architecture in large-scale IoT networks," *IEEE Trans. Netw. Sci. Eng.*, 2026.

[23] A. Alqazzaz, "SecuFL-IoT: An adaptive privacy-preserving federated learning framework for anomaly detection in smart industrial networks," *Sci. Rep.*, 2026

[24] M. Mehta and K. Patel, "Design and performance analysis of lightweight mutual authentication protocols for resource constrained IoT devices," *VIDYA—A J. Gujarat Univ.*, vol. 5, no. 1, pp. 96–104, 2026.