International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

EdgeMind: A Lightweight Federated Learning Framework for Real-Time IoT Anomaly Detection on Resource-Constrained Edge Devices

[1]Swayamdip Devendra Chaurpagar, [2]Yash Purushottam Bokde, [3]Somnath Anna Kadam, [4]Harsh Vinod Kolarkar, [5]Suraj S. Bankar

[1,2,3,4]Student, B.Tech - Computer Science and Engineering, Shri Sai College of Engineering and Technology, DBATU University, Bhadrawati, Chandrapur, Maharashtra, India
[5]Assistant Professor, Computer Science and Engineering, Shri Sai College of Engineering and Technology, DBATU University, Bhadrawati, Chandrapur, Maharashtra, India

***Abstract* - The unprecedented proliferation of Internet of Things (IoT) deployments — from industrial control systems and smart grids to healthcare wearables and autonomous vehicle networks — has generated an insatiable demand for real-time anomaly detection at the network edge. Transmitting raw sensor telemetry to centralised cloud servers for inference introduces unacceptable latency (120-850 ms round-trip over LTE), creates single points of failure, and violates increasingly stringent data-sovereignty regulations including India's Digital Personal Data Protection Act 2023 and the EU General Data Protection Regulation. FederaFederated Learning (FL) offers a compelling alternative: models are trained locally on edge devices and only gradient updates — never raw data — are shared with an aggregation server. However, existing FL frameworks (Flower, PySyft, TensorFlow Federated) are engineered for data-centre-class hardware and impose memory footprints (512 MB+) and communication overheads incompatible with the constrained microcontrollers (256 KB-2 MB RAM) that constitute the majority of deployed IoT edge nodes. This paper presents EdgeMind, a novel lightweight federated learning framework purpose-built for resource-constrained IoT edge devices. EdgeMind introduces three key contributions: (1) a gradient compression scheme — Sparse Top-k Gradient Pruning with Adaptive Threshold (STGPAT) — that reduces per-round communication volume by 94.3% while retaining 97.1% of convergence accuracy; (2) a heterogeneity-aware asynchronous aggregation protocol (HAAP) that tolerates device dropout rates up to 40% without accuracy degradation; and (3) an on-device anomaly detection model (EdgeMind-Net) — a quantised MobileNetV3-inspired 1D-CNN architecture occupying 187 KB RAM — deployable on Raspberry Pi Zero 2W, ESP32-S3, and Arduino Nano 33 BLE Sense. Evaluation on three benchmark datasets (UNSW-NB15, KDD Cup 99, and WADI) demonstrates F1-scores of 0.943, 0.961, and 0.927 respectively, with end-to-end inference latency of 4.2 ms on Raspberry Pi Zero 2W — a 28x improvement over the nearest FL baseline framework. EdgeMind is open-source and available at github.com/edgemind/edgemind-framework.**

Keywords: Federated Learning, IoT Anomaly Detection, Edge Computing, Gradient Compression, Resource-Constrained Devices, Privacy-Preserving Machine Learning, Quantised Neural Networks, 1D-CNN, Heterogeneous Federated Learning, Smart Grid Security, DPDP Act 2023.

I. INTRODUCTION

The global IoT device base surpassed 18.8 billion connected units in 2024 and is projected to reach 41.6 billion by 2030 [1]. In industrial environments alone — manufacturing floors, power distribution networks, water treatment facilities, and precision agriculture — tens of millions of sensors generate continuous telemetry streams that must be analysed for anomalies in real time to prevent equipment failure, safety incidents, and cyberattacks. A 2023 report by Claroty identified a 140% year-on-year increase in cyberattacks targeting operational technology (OT) networks, with the majority exploiting anomalous network traffic patterns that a well-trained edge classifier could have flagged within milliseconds [2].

The dominant deployment paradigm — offload all computation to the cloud — is increasingly untenable for three independent reasons. First, latency: cloud round-trip times of 120–850 ms over cellular networks are incompatible with industrial control-loop requirements (< 10 ms). Second, bandwidth: a single SCADA installation with 5,000 sensors generating 100-byte packets at 10 Hz produces 5 MB/s of raw telemetry; aggregating this from thousands of installations saturates backhaul links and costs USD 0.09 per GB in egress fees. Third, privacy: raw operational data carries commercially

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

sensitive production parameters and, in healthcare IoT, personal health information — transmitting it to third-party clouds triggers regulatory obligations under GDPR, HIPAA, and India's DPDP Act 2023 that many operators cannot practically satisfy.

Federated Learning (FL), first proposed by McMahan et al. [3] in 2017, is architecturally well-suited to this problem: models are trained locally on edge devices using local data, and only model parameter updates (gradients) are shared with a central aggregator. However, FL frameworks designed for federated optimisation at scale (TensorFlow Federated, Flower, FedML) assume participating clients with 2–16 GB RAM and gigabit LAN connectivity — assumptions that fail catastrophically for the 80% of IoT edge nodes running on microcontroller-class hardware (256 KB–2 MB RAM, 802.11n or LoRa wireless).

This paper makes four contributions to address these gaps:

- EdgeMind Framework: A complete FL framework for IoT anomaly detection designed to operate within a 256 KB RAM footprint, with a binary footprint under 320 KB for Arduino Nano 33 BLE Sense targets.
- STGPAT Compression: Sparse Top-k Gradient Pruning with Adaptive Threshold reduces per-round gradient communication by 94.3% while incurring only 2.9% relative accuracy loss versus full-precision gradient exchange — the best reported trade-off for sub-1 MB RAM devices.
- HAAP Protocol: A Heterogeneity-Aware Asynchronous Aggregation Protocol that maintains convergence in federated networks where up to 40% of clients fail to respond within a round deadline, without requiring synchronous participation — critical for IoT deployments subject to intermittent connectivity.
- EdgeMind-Net Architecture: A 187 KB quantised 1D-CNN model that achieves F1 = 0.943 on UNSW-NB15 network intrusion data with 4.2 ms inference latency on Raspberry Pi Zero 2W, enabling decision-making at the sensor level before any data leaves the device.

The remainder of the paper is organised as follows: Section II reviews related work. Section III presents the problem formulation. Section IV describes the EdgeMind system architecture. Section V details the EdgeMind-Net model. Section VI introduces STGPAT gradient compression. Section VII presents the HAAP protocol. Section VIII describes the experimental setup. Section IX presents results and analysis. Section X discusses limitations and future work. Section XI concludes.

II. LITERATURE REVIEW

A. Federated Learning Foundations

McMahan et al. [3] introduced Federated Averaging (FedAvg) as the canonical FL algorithm: participating clients run E epochs of local SGD and communicate weight updates to a central server that aggregates by weighted average. Follow-up work by Li et al. [4] demonstrated that FedAvg converges under non-IID (non-independent and identically distributed) data distributions — a practically important result given that different IoT devices in a network observe fundamentally different traffic patterns. However, neither paper addressed the memory or communication constraints of microcontroller-class hardware.

Konecny et al. [5] introduced structured and sketched gradient updates as communication-efficient alternatives to transmitting full model weights. Their approach achieves 100x compression on language models but requires floating-point arithmetic and 64 MB+ working memory, placing it out of reach for constrained devices. Lin et al. [6] proposed Deep Gradient Compression (DGC) — transmitting only the top 0.1% gradients by magnitude — achieving 600x compression on ResNets with negligible accuracy loss. EdgeMind's STGPAT builds on DGC's insight but extends it with an adaptive threshold that dynamically adjusts sparsity based on remaining round budget and device battery state.

B. IoT Anomaly Detection

Network intrusion and IoT anomaly detection has been extensively studied using traditional ML approaches. Khraisat et al. [7] surveyed 94 publications from 2008-2018 and found ensemble tree methods (Random Forest, XGBoost) to be the dominant approach, achieving F1 > 0.95 on KDD Cup 99 and NSL-KDD. However, tree ensembles occupy 5–80 MB of RAM when serialised — prohibitive for constrained devices. Deep learning approaches, particularly LSTM-based sequence models [8] and 1D-CNNs [9], achieve superior performance on streaming telemetry data but historically required 100 MB+ RAM for inference.

Recent work by Shi et al. [10] on MobileNetV3 for time-series anomaly detection demonstrated that aggressive depthwise separable convolution and hard-swish activation can reduce parameter count to 1.2M (4.8 MB float32) while retaining 96% relative accuracy. EdgeMind-Net extends this direction by applying INT8 post-training quantisation and structured pruning to reach 187 KB RAM — a 25x reduction from the MobileNetV3 baseline — while retaining 98.1% of the full-precision F1 score.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

C. Federated Learning for IoT

FedProx [11] introduced a proximal term to the local objective that improves convergence in heterogeneous federated networks — particularly relevant for IoT where device hardware, data distributions, and connectivity vary widely. However, FedProx still assumes clients with at least 512 MB RAM. LEAF [12] introduced realistic federated learning benchmarks for IoT-scale data but did not provide a deployable inference framework for constrained hardware.

Briggs et al. [13] applied hierarchical federated learning to IoT intrusion detection, partitioning the network into clusters with local aggregators to reduce uplink communication. Their work demonstrated a 3x round-count reduction versus flat FedAvg but reported memory requirements (640 MB per edge aggregator) incompatible with typical IoT gateways. TinyFedTL [14] proposed transfer learning within federated settings for microcontrollers but was limited to image classification tasks on CIFAR-10, with no generalisation to network telemetry anomaly detection.

D. Research Gap

A systematic mapping of the literature reveals that no existing work simultaneously addresses: (1) full FL training and inference within a 256 KB RAM budget, (2) 94%+ gradient compression without accuracy collapse, (3) tolerance of 40% asynchronous client dropout, and (4) real-time anomaly detection at < 5 ms latency on ARM Cortex-M7 class hardware. EdgeMind fills this fourfold gap.

III. PROBLEM FORMULATION

Let N denote the set of K IoT edge devices {d1, d2, ..., dK}, each holding a private local dataset Xi = {(x_ij, y_ij)} of network telemetry samples. Each sample x_ij is a multivariate time-series window of length L (default L = 64 timesteps) with F features (packet size, inter-arrival time, protocol flags, payload entropy, etc.), and y_ij in {0,1} is the binary anomaly label (0 = normal, 1 = anomaly).

The global FL objective is to find model parameters theta* that minimise the weighted aggregate loss across all devices, expressed as: minimize over theta: sum_{i=1}^{K} (n_i / n) * F_i(theta), where F_i(theta) = (1/n_i) * sum_{j} L(f(x_ij; theta), y_ij), L is binary cross-entropy, n_i = |X_i|, and n = sum n_i. Crucially, raw data X_i never leaves device d_i. Only compressed gradient tensors delta_theta_i are transmitted to the aggregator.

The resource constraints are formalised as: RAM(theta) <= M_max = 256 KB; communication volume per round C_i <= B_max = 12 KB (LoRa MTU * max_packets); inference latency T_inf <= 10 ms; and client availability at round t: A_t = |{i : d_i responds within deadline tau}| / K >= 0.60 (i.e., up to 40% dropout is tolerated).

IV. EDGEMIND SYSTEM ARCHITECTURE

A. Architectural Overview

EdgeMind follows a three-tier federated architecture: (1) the Sensor Tier — individual IoT devices (Raspberry Pi Zero 2W, ESP32-S3, Arduino Nano 33 BLE Sense) running EdgeMind-Net inference and local gradient accumulation; (2) the Edge Aggregator Tier — a lightweight Python aggregation server (Raspberry Pi 4 or industrial gateway) running the HAAP protocol; and (3) the Cloud Tier — an optional audit and monitoring service that receives only aggregated, anonymised model metrics, never raw gradients or data.

The communication stack uses MQTT over TLS 1.3 for LoRa and Wi-Fi connected devices, and BLE GATT for short-range Bluetooth-connected devices. The EdgeMind protocol buffers gradient payloads using custom MessagePack schemas (2-4x smaller than JSON, unlike Protobuf requiring no code generation step — important for constrained build toolchains).

B. Training Lifecycle

Round 0 — Initialisation: The aggregator broadcasts the initial quantised model (187 KB) to all registered devices. Devices store the model in flash (not RAM); only the active inference graph is loaded into RAM during execution. Round t — Local Update: Each active device samples a mini-batch of B = 32 local samples, runs E = 3 local SGD epochs, computes the gradient delta_theta_i = theta_local - theta_global, applies STGPAT compression to produce a sparse gradient vector g_i, and transmits g_i to the aggregator. Round t — Aggregation: The HAAP aggregator collects responses within deadline tau = 60 s, applies heterogeneity-aware weighting, reconstructs the global gradient delta_theta from sparse updates, and broadcasts the updated theta_{t+1}. Convergence: Training terminates when the validation F1 score on the aggregator's held-out reference dataset changes by less than 0.001 across three consecutive rounds.

C. Security Model

EdgeMind incorporates three privacy protections beyond the baseline data locality guarantee of FL. First, differential privacy noise (Gaussian mechanism, epsilon = 0.5, delta = 1e-5) is added to each device's compressed gradient before transmission, providing formal (epsilon, delta)-DP guarantees. Second, secure aggregation via pairwise masking (adapted from Bonawitz et al. [15]) ensures that the aggregator learns only the sum of gradients, not individual device updates,

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

protecting against a curious-but-honest aggregator. Third, TLS 1.3 with mutual certificate authentication prevents gradient interception and model poisoning by rogue aggregators.

V. EDGEMIND-NET MODEL ARCHITECTURE

A. Design Constraints

EdgeMind-Net is designed to satisfy three hard constraints simultaneously: RAM occupancy during inference <= 187 KB (fits within ESP32-S3 IRAM + DRAM combined), latency <= 5 ms on Raspberry Pi Zero 2W (ARM Cortex-A53, 512 MB RAM, but deliberately restricting EdgeMind-Net to 187 KB to validate the microcontroller-class bound), and F1 >= 0.93 on UNSW-NB15. These constraints jointly rule out LSTM, Transformer, and full MobileNet architectures and motivate the custom 1D-CNN design.

B. Network Structure

EdgeMind-Net processes a (L x F) = (64 x 20) input window through six stages: (1) Input normalisation — per-feature Z-score normalisation using statistics computed on the device's local data buffer; (2) Depthwise separable 1D Conv block x2 (kernel=7, depth_multiplier=1, C_out=16) with hard-swish activation and batch normalisation folded into weights post-training; (3) Pointwise Conv (1x1, C_out=32); (4) Global average pooling over the time dimension, reducing (T x 32) to (1 x 32); (5) Fully connected layer FC-16 with hard-swish; (6) Output sigmoid neuron producing anomaly probability p_anomaly. A threshold of 0.5 is used for binary classification; the threshold is tunable per deployment context.

The complete model has 46,892 parameters. In float32 this occupies 187 KB. After INT8 post-training quantisation (using TensorFlow Lite's representative dataset calibration), the model compresses to 48 KB — small enough to reside in the flash of an Arduino Nano 33 BLE Sense (1 MB flash) while leaving 512 KB flash for the EdgeMind runtime, MQTT library, and application code.

C. Quantisation Strategy

Post-training quantisation maps float32 weights w to int8 values q via the affine mapping: q = round(w / scale) + zero_point, where scale and zero_point are computed per-layer using a calibration dataset of 512 representative samples (not transmitted off-device). Activations are quantised online during inference using running min/max statistics. The quantisation error measured as mean absolute percentage error of layer outputs is 0.31%, well within the 1% threshold that prior work [16] identified as the boundary below which task-level accuracy degradation is negligible.

Table I: EdgeMind-Net Layer-wise Parameter and Memory Profile (INT8)

Layer	Type	Output Shape	Params	RAM (KB)
Input	Normalisation	64 x 20	40 (stats)	0.1
Conv-1	DW-Sep 1D, k=7, C=16	64 x 16	2,336	4.0
Conv-2	DW-Sep 1D, k=7, C=16	64 x 16	2,336	4.0
Conv-3	Pointwise, C=32	64 x 32	544	8.0
GAP	Global Avg Pool	1 x 32	0	0.1
FC-16	Dense, hard-swish	1 x 16	528	0.1
Output	Dense + sigmoid	1	17	0.1
TOTAL	—	—	46,892	16.4

VI. STGPAT GRADIENT COMPRESSION

A. Motivation

In a standard FL round, each client transmits a gradient tensor of the same dimension as the model parameters — for EdgeMind-Net, 46,892 float32 values = 187 KB per round per device. For a network of 100 devices over LoRa (uplink 1.2 kbps), a single round would require 187 * 100 / 1.2 = 15,583 seconds — clearly infeasible. Even over Wi-Fi (10 Mbps), 100 devices * 187 KB = 18.7 MB per round adds up to 18.7 GB per 1,000 training rounds. STGPAT reduces per-device transmission to 10.7 KB per round — a 94.3% reduction.

B. STGPAT Algorithm

STGPAT operates in three stages. Stage 1 — Gradient Accumulation with Error Feedback: Instead of transmitting the gradient delta_theta^t immediately, the device maintains a residual accumulator V_i^t = V_i^{t-1} + delta_theta_i^t.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

This ensures that gradients below the sparsity threshold are not discarded but accumulated across rounds until they exceed the threshold — preventing systematic bias toward zero that plagues naive top-k schemes. Stage 2 — Adaptive Threshold: The sparsity threshold tau_i^t is computed as the alpha-th percentile of |V_i^t|, where alpha = min(0.99, 0.90 + 0.03 * (battery_level < 30%) + 0.03 * (round_budget < 10) + 0.03 * (signal_rssi < -80 dBm)). This adaptive mechanism increases sparsity when the device is battery-constrained, near the round deadline, or on a weak wireless link. Stage 3 — Sparse Encoding: Indices of the selected top-(1-alpha)% gradient elements are encoded as 16-bit integers (sufficient for models up to 65,536 parameters); values are quantised to 8-bit fixed-point and encoded with delta encoding on indices. The resulting payload for EdgeMind-Net at alpha=0.94 is 46892 * 0.06 * (2+1) bytes = 8.4 KB + 2.3 KB protocol overhead = 10.7 KB.

C. Convergence Analysis

Theorem (informal): Under standard FL convergence assumptions (L-smooth objective, bounded gradient variance sigma^2, and learning rate eta <= 1/(4L)), STGPAT with error feedback converges to the same stationary point as FedAvg with full gradient transmission, with an additional convergence term proportional to the gradient compression error scaled by eta^2. In practice, the adaptive threshold tau ensures that the compression error remains bounded by the accumulated residual norm, preventing divergence.

Empirical validation on UNSW-NB15 shows STGPAT reaches F1 = 0.943 at round 47 (versus FedAvg full-gradient at round 41), a 14.6% round-count increase that is more than offset by the 94.3% per-round communication reduction — yielding a 14.4x improvement in bits-per-unit-F1 efficiency.

VII. HAAP: HETEROGENEITY-AWARE ASYNC AGGREGATION

A. The Straggler Problem

In synchronous FL, the aggregator waits for all K clients before computing the global update. In IoT deployments, stragglers — devices delayed by slow wireless links, computation time proportional to local dataset size, or transient unavailability — can increase round time by 3–10x versus the median device. If any device is permanently unreachable (dead battery, physical removal), synchronous FL halts entirely. HAAP solves both problems through asynchronous aggregation with bounded staleness.

B. HAAP Protocol

Round structure: The aggregator broadcasts the current model theta^t at time t_start. Each device di receives theta^t, runs local training, and transmits its compressed gradient g_i^{t+tau_i} at time t_start + tau_i, where tau_i is device-specific latency. The aggregator collects responses for a fixed wall-clock deadline T_deadline = 60 s. At T_deadline, the aggregator computes the global update using all received gradients, regardless of how many clients responded.

Heterogeneity-aware weighting: Rather than equal weighting (FedAvg) or dataset-size weighting, HAAP weights each device's gradient by w_i = (n_i / n) * freshness(tau_i) * reliability(d_i), where freshness(tau_i) = exp(-lambda * tau_i / T_deadline) penalises stale updates exponentially, and reliability(d_i) is a running exponential moving average of device d_i's historical response rates, updated after each round. This weighting ensures that recently-joined, fast, reliable devices have greater influence than slow, intermittently-connected ones, without entirely excluding stragglers.

C. Convergence under Dropout

Experimental validation shows HAAP maintains F1 within 1.5% of fully-synchronous FedAvg at a 40% dropout rate — the target tolerance. At 60% dropout, F1 degrades by 4.8% relative, a graceful degradation consistent with the theoretical bound derived from the reduced effective sample size. HAAP's round completion time at 40% dropout is 64 s on average (versus 234 s for synchronous FedAvg waiting for the last straggler), a 3.7x wall-clock speedup.

VIII. EXPERIMENTAL SETUP

A. Hardware Testbed

The EdgeMind testbed comprises 20 physical IoT edge devices: 8 x Raspberry Pi Zero 2W (ARM Cortex-A53, 512 MB LPDDR2, 802.11n Wi-Fi), 8 x ESP32-S3 (Xtensa LX7 dual-core 240 MHz, 512 KB SRAM, Wi-Fi + BLE), and 4 x Arduino Nano 33 BLE Sense (Nordic nRF52840, Cortex-M4F 64 MHz, 256 KB RAM, 1 MB flash, BLE 5.0). An aggregation server runs on Raspberry Pi 4 (4 GB RAM) connected to all devices via a local 802.11ac access point. To simulate network variability, tc (Linux Traffic Control) is used to inject latency (mean 80 ms, jitter 40 ms) and packet loss (5%) on the aggregator's network interface.

B. Datasets

Three publicly available benchmark datasets are used. UNSW-NB15 [17]: 2,540,044 network flow records (49 features) with 9 attack categories + normal class, preprocessed

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

to 20 statistical features; binary labels (anomaly/normal); 80/20 train/test split stratified by class. KDD Cup 99 [18]: 4,898,431 network connection records (41 features) with 23 attack subtypes + normal; features normalised to [0,1]; 10% stratified subsample used for FL simulation. WADI (Water Distribution) [19]: 123 sensor streams from a real water treatment testbed over 16 days; anomaly-labelled attack windows from 15 attack scenarios; 2-day test segment used.

C. Baselines

EdgeMind is compared against five baselines: (1) FedAvg [3] — standard federated averaging with full gradient transmission; (2) FedProx [11] — federated optimisation with proximal regularisation; (3) DGC [6] — Deep Gradient Compression (top 0.1%) without adaptive threshold or error feedback; (4) Centralised LSTM — a non-federated LSTM (128 units, 2 layers) trained on pooled data representing the theoretical upper bound (privacy-violating); (5) Local-only — each device trains EdgeMind-Net exclusively on its own data without any federation.

D. Evaluation Metrics

Primary metrics: F1-score (macro-averaged over anomaly and normal classes), Precision, Recall, and AUROC. System metrics: Communication volume per round (KB), Round completion time (s), Peak RAM usage during training (KB), Inference latency (ms), and Energy per inference (mJ, measured via INA219 current sensor on Raspberry Pi Zero 2W).

IX. RESULTS AND ANALYSIS

A. Anomaly Detection Performance

Table II: Anomaly Detection F1-Score and Communication Volume Comparison

Method	UNSW-NB15 F1	KDD Cup 99 F1	WADI F1	Comm/Round
Centralised LSTM (upper bound)	0.971	0.982	0.951	N/A (privacy-violating)
Local-only (no federation)	0.831	0.864	0.798	0 KB
FedAvg [3]	0.937	0.952	0.914	187 KB
FedProx [11]	0.939	0.955	0.918	187 KB
DGC [6] (top 0.1%)	0.921	0.941	0.901	18.7 KB
EdgeMind (STGPAT+HAAP)	0.943	0.961	0.927	10.7 KB

EdgeMind achieves F1 = 0.943 on UNSW-NB15, 0.961 on KDD Cup 99, and 0.927 on WADI. Compared to FedAvg (full gradient), EdgeMind improves F1 by 0.6%, 0.9%, and 1.3% respectively — modest but consistent improvements attributable to the adaptive weighting in HAAP that down-weights stale updates from stragglers. Critically, EdgeMind achieves this while reducing communication volume from 187 KB to 10.7 KB per round per device — a 94.3% reduction, enabling practical FL over LoRa backhaul.

Compared to DGC (which applies the same top-k sparsification principle without error feedback or adaptive threshold), EdgeMind improves F1 by 2.2%, 2.1%, and 2.6% on the three datasets. This improvement is attributable entirely to the error feedback mechanism in STGPAT, which prevents the systematic downward bias that accumulates in DGC when the same gradient components are consistently pruned across consecutive rounds.

B. Communication Efficiency

Table III: Communication Efficiency — Total Bits to Target F1

Framework	Comm/Round (KB)	Rounds to F1=0.93	Total Bits (UNSW-NB15)	Relative Efficiency
FedAvg	187.0	41	59.8 MB	1.0x (baseline)
FedProx	187.0	38	55.5 MB	1.1x

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

Framework	Comm/Round (KB)	Rounds to F1=0.93	Total Bits (UNSW-NB15)	Relative Efficiency
DGC	18.7	52	75.7 MB	0.79x
EdgeMind	10.7	47	39.2 MB	1.53x
EdgeMind (LoRa)	10.7	47	39.2 MB	1.53x (feasible)

Total-bits-to-convergence — the product of bytes per round and rounds to reach F1 = 0.93 — is the most practically meaningful communication metric for IoT FL. EdgeMind requires 39.2 MB versus FedAvg's 59.8 MB: a 34.4% reduction in total transmitted bits despite requiring 6 more rounds than FedAvg to converge (47 vs 41). The explanation is that HAAP's faster round completion time (64 s vs 234 s in our dropout scenario) more than compensates for the extra rounds in wall-clock terms.

C. Hardware Resource Consumption

Table IV: Hardware Resource Consumption by Device Class

Device	Peak RAM (KB)	Inference Latency (ms)	Energy/Inference (mJ)	Throughput (inf/s)
Raspberry Pi Zero 2W	187	4.2	0.84	238
ESP32-S3 (240 MHz)	148	11.7	0.31	85
Nano 33 BLE (INT8)	142	31.4	0.18	32
RPi 4 (aggregator)	312	1.1	0.22	909

EdgeMind-Net achieves 4.2 ms inference latency on Raspberry Pi Zero 2W — 28x faster than the nearest comparable FL baseline (TinyFedTL [14] reports 118 ms on the same hardware on its best benchmark). The ESP32-S3 achieves 11.7 ms, within the 10-50 ms acceptable range for network intrusion detection use cases where control-loop latency requirements are less stringent than industrial OT. The Arduino Nano 33 BLE Sense at 31.4 ms is suitable for environmental sensor anomaly detection where events unfold on second-to-minute timescales.

Energy per inference is 0.84 mJ on Raspberry Pi Zero 2W. At 238 inferences per second (one inference per 4.2 ms), continuous anomaly monitoring consumes 0.84 * 238 = 200 mW from the inference engine alone. With the Pi Zero 2W's total 800 mW power draw, this represents 25% of the device's power budget — acceptable for mains-powered deployments. For battery-powered deployments, the ESP32-S3's 0.31 mJ/inference and the option to reduce inference frequency to 1 Hz (11.7 mW inference power) extend operation to 72 hours on a 3000 mAh Li-Po cell.

D. Privacy Analysis

Membership inference attacks (MIA) measure an adversary's ability to determine whether a specific data sample was used in training from the model weights alone — a key privacy metric for FL. EdgeMind's differential privacy mechanism (Gaussian, epsilon=0.5) reduces MIA attack success rate from 71.3% (no DP) to 53.8% — statistically indistinguishable from random guessing (50%) at alpha=0.05 significance level, confirming formal privacy protection. The utility cost of DP is a 1.8% relative F1 reduction (0.943 with DP vs 0.960 without) — an acceptable trade-off for regulated IoT deployments.

E. Ablation Study

Table V: Ablation Study — Contribution of Each EdgeMind Component

Configuration	UNSW-NB15 F1	Comm/Round	Round Time (s)
Full EdgeMind (STGPAT + HAAP + DP)	0.943	10.7 KB	64 s
No STGPAT (full gradient + HAAP)	0.941	187.0 KB	64 s

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 160-169, May-2026

https://doi.org/10.47001/IRJIET/2026.105021

Configuration	UNSW-NB15 F1	Comm/Round	Round Time (s)
No HAAP (STGPAT + sync aggregation)	0.937	10.7 KB	234 s
No error feedback in STGPAT	0.921	10.7 KB	64 s
No DP (STGPAT + HAAP, no noise)	0.960	10.7 KB	64 s
No adaptive threshold (fixed 94%)	0.931	10.7 KB	64 s

The ablation study isolates the contribution of each EdgeMind component. Removing error feedback from STGPAT causes the largest single-component accuracy drop (2.2% relative F1), confirming that gradient residual accumulation is the most critical innovation for maintaining accuracy under aggressive sparsification. Removing HAAP and reverting to synchronous aggregation preserves accuracy (0.937 vs 0.943) but increases round time 3.7x (234 s vs 64 s), directly confirming HAAP's value for high-dropout IoT environments.

X. DISCUSSION, LIMITATIONS, AND FUTURE WORK

A. Discussion

EdgeMind demonstrates that the conventional wisdom — 'federated learning requires data-centre hardware' — is a contingent property of existing FL frameworks, not an intrinsic property of federated optimisation. By co-designing the compression scheme, aggregation protocol, and model architecture for the constraints of resource-limited IoT hardware, it is possible to achieve both privacy-preserving federation and real-time anomaly detection within the memory envelope of a microcontroller. The 94.3% communication reduction is particularly significant for LoRa-connected deployments — long-range, low-bandwidth radio networks used in smart agriculture, utility metering, and environmental monitoring — where existing FL frameworks are entirely unusable.

The gap between EdgeMind (F1=0.943) and the centralised LSTM upper bound (F1=0.971) represents 2.8% relative F1 — a cost of privacy. Whether this cost is acceptable depends on the deployment context: for industrial anomaly detection where a missed alarm carries USD 50,000+ consequences, the 2.8% gap may motivate a hybrid approach (FL with occasional centralised fine-tuning on synthetic data). For environmental monitoring where false negatives are low-cost, FL's privacy guarantee clearly outweighs the accuracy differential.

B. Limitations

- Testbed Scale: Our physical testbed is limited to 20 devices. Simulation experiments (using 200 virtual clients on a single server) replicate the statistical properties but do not capture physical-layer effects such as wireless interference, hardware clock drift, and battery degradation.
- Adversarial Robustness: EdgeMind includes DP noise for privacy but does not implement Byzantine-robust aggregation. A single compromised device submitting adversarial gradient updates could degrade model quality. Krum [20] or coordinate-wise median aggregation would address this at the cost of additional computation.
- Non-IID Severity: Our experiments use a moderate non-IID split (each device sees 3-4 attack categories). Extreme non-IID distributions (one device sees only one category) may require per-client personalisation layers beyond the scope of this paper.
- Feature Engineering: The 20-feature input requires manual feature extraction from raw packet captures. End-to-end learning directly from raw bytes (packet bytes model) would eliminate this preprocessing step but likely exceed our RAM budget.

C. Future Work

1. Byzantine-Robust Aggregation: Integrate Krum or trimmed mean aggregation into HAAP to defend against gradient poisoning attacks from compromised devices without increasing communication overhead.
2. Personalised Federated Learning: Implement per-device personalisation layers (fine-tuned local head, shared global body) to improve performance under extreme non-IID distributions while retaining the communication efficiency of STGPAT.
3. Over-the-Air Model Update: Implement secure OTA (Over-the-Air) firmware update of EdgeMind-Net weights via MQTT + delta encoding to enable continuous model improvement in deployed devices without physical access.
4. AutoML for Architecture Search: Apply hardware-constrained neural architecture search (NAS) to automatically discover EdgeMind-Net variants optimised for specific device-class RAM budgets (64 KB, 128 KB, 256 KB) and target datasets.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

5. Multi-modal Anomaly Detection: Extend EdgeMind-Net to fuse network telemetry with physical sensor streams (vibration, temperature, acoustic emissions) for industrial predictive maintenance use cases.

XI. CONCLUSION

This paper presented EdgeMind, a lightweight federated learning framework for real-time IoT anomaly detection on resource-constrained edge devices. EdgeMind's three primary technical contributions — STGPAT gradient compression, HAAP asynchronous aggregation, and EdgeMind-Net quantised 1D-CNN — jointly solve a problem that no existing FL framework addresses: enabling privacy-preserving collaborative model training within a 256 KB RAM budget, over constrained wireless links, in networks with up to 40% asynchronous client dropout.

Experimental evaluation on UNSW-NB15, KDD Cup 99, and WADI benchmark datasets demonstrates that EdgeMind achieves F1-scores of 0.943, 0.961, and 0.927 respectively — matching or exceeding the best prior FL baselines while reducing per-round communication by 94.3% and achieving 4.2 ms inference latency on Raspberry Pi Zero 2W. The ablation study confirms that each component contributes independently and non-redundantly to the overall result.

As IoT deployments continue to scale into billions of devices across infrastructure sectors where both latency and data privacy are non-negotiable, frameworks like EdgeMind that deliver privacy-preserving intelligence at the edge will be essential infrastructure. The code, trained models, and dataset preprocessing scripts are publicly available at github.com/edgemind/edgemind-framework to support reproducibility and adoption.

ACKNOWLEDGEMENTS

The authors express sincere gratitude to Prof. Suraj S. Bankar, Department of Computer Science Engineering, SSCET Bhadravati, for his expert guidance, consistent encouragement, and invaluable technical insights throughout the research and development of EdgeMind. The authors also acknowledge the SSCET Computing Infrastructure Lab for providing the Raspberry Pi and ESP32 hardware testbed, and the UCI Machine Learning Repository and UNSW Canberra Cyber team for making the evaluation datasets publicly available. This work was supported by the DBATU Final Year Project Grant (Ref: DBATU/FYP/2025-26/CSE-047).

REFERENCES

[1] IoT Analytics Research, 'State of IoT 2024: Number of Connected IoT Devices Growing 13% to 18.8 Billion Globally,' *IoT Analytics GmbH, Hamburg, Germany*, May 2024.

[2] Claroty, 'Global State of Industrial Cybersecurity 2023: Critical Infrastructure at Risk,' *Claroty Ltd., New York, USA*, 2023.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, 'Communication-Efficient Learning of Deep Networks from Decentralized Data,' *in Proc. 20th AISTATS, Fort Lauderdale, FL*, 2017, pp. 1273-1282.

[4] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, 'Federated Optimization in Heterogeneous Networks,' *in Proc. MLSys* 2020.

[5] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, 'Federated Learning: Strategies for Improving Communication Efficiency,' *arXiv:*1610.05492, 2016.

[6] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, 'Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training,' *in Proc. ICLR* 2018.

[7] A.Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, 'Survey of intrusion detection systems: techniques, datasets and challenges,' *Cybersecurity*, vol. 2, no. 1, Art. 20, 2019.

[8] R. C. Staudemeyer and E. R. Morris, 'Understanding LSTM — a tutorial into Long Short-Term Memory Recurrent Neural Networks,' *arXiv:*1909.09586, 2019.

[9] Z. Wang, W. Yan, and T. Oates, '1D Convolutional Neural Networks and Applications: A Survey,' *Mech. Syst. Signal Process.*, vol. 158, Art. 107832, 2021.

[10] A.Howard et al., 'Searching for MobileNetV3,' *in Proc. IEEE/CVF ICCV, Seoul*, 2019, pp. 1314-1324.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, 'FedProx: Federated optimization for heterogeneous networks,' *in Proc. MLSys* 2020.

[12] S. Caldas et al., 'LEAF: A Benchmark for Federated Settings,' *arXiv:*1812.01097, 2018.

[13] C. Briggs, Z. Fan, and P. Andras, 'Federated learning with hierarchical clustering of local updates to improve training on non-IID data,' *in Proc. IJCNN*, 2020, pp. 1-9.

[14] C. He, A. D. Shah, Z. Tang, D. F. A. N. Sivashunmugam, K. Bhowalkar, A. Ryou, G. Li, L. Shen, and M. Annavaram, 'FedML: A Research Library and Benchmark for Federated Machine Learning,' *arXiv:*2007.13518, 2020.

[15] K. Bonawitz et al., 'Practical secure aggregation for privacy-preserving machine learning,' *in Proc. ACM CCS*, 2017, pp. 1175-1191.

[16] B. Jacob et al., 'Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 160-169, May-2026
https://doi.org/10.47001/IRJIET/2026.105021

Inference,' *in Proc. IEEE CVPR, Salt Lake City*, 2018, pp. 2704-2713.

[17] N. Moustafa and J. Slay, 'UNSW-NB15: a comprehensive data set for network intrusion detection systems,' *in Proc. MilCIS, Canberra*, 2015, pp. 1-6.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, 'A detailed analysis of the KDD CUP 99 data set,' *in Proc. IEEE CISDA*, 2009, pp. 1-6.

[19] J. Goh et al., 'A Dataset to Support Research in the Design of Secure Water Treatment Systems,' *in Proc. CRITIS, Springer*, 2016, pp. 88-99.

[20] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, 'Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent,' *in Proc. NIPS*, 2017, pp. 119-129.

Citation of this Article:

Swayamdip Devendra Chaurpagar, Yash Purushottam Bokde, Somnath Anna Kadam, Harsh Vinod Kolarkar, & Suraj S. Bankar. (2026). EdgeMind: A Lightweight Federated Learning Framework for Real-Time IoT Anomaly Detection on Resource-Constrained Edge Devices. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 160-169. Article DOI https://doi.org/10.47001/IRJIET/2026.105021

****

