Appl. Comput. Math., Vol.25, No.2, Special Issue, 2026, pp.332-348
DOI: 10.30546/1683-6154.25.2.2026.332

EXPLAINABLE TINYML FRAMEWORK FOR SECURE AND TRUSTWORTHY FEDERATED MEDICAL CYBER-PHYSICAL SYSTEMS

M.A. ALOHALI[1], A.K. SAMHA[2], A. ALSHUHAIL[3], H. ALQAHTANI[4], N.A. ALGHANMI[5], A.E. YAHYA[6,*]

Abstract. The paper proposes a TinyML-enabled Federated Learning (FL) architecture of Cyber-Physical Systems (MCPS), in which resource-constrained devices train local models on limited resources and transmit encrypted updates to facilitate secure aggregation. The structure incorporates both local and global explainability in order to instill more confidence in clinicians. A vital-sign classification task of synthetic features compares two models, that is a tiny linear classifier and a small neural network. Findings indicate that the linear model converges faster, has a smaller communication burden and is more interpretable, whereas the neural model achieves comparable performance with additional rounds of training. The proposed solution brings into the limelight the efficiency of explainable AI, edge-centric AI, and privacy-preserving AI in the next-generation healthcare system.

Keywords: TinyML, Federated Learning, Explainable AI, Medical IoT, Cyber-Physical Systems, Trust, Privacy, Security, Edge Intelligence, Healthcare AI.

AMS Subject Classification: 68T07, 68T09, 68Q32, 94A60.

1. Introduction

The combination of medical sensors and embedded AI is transforming bedside care, remote care, and smart wards. MCPS integrate continuous data collection through wearables, implants and bedside devices with on-device inference to raise alerts, estimate risk and direct interventions. An edge-based is bandwidth-demand limited, latency-restrictive, and sensitive-stream exposure-restricted. More importantly, clinicians should trust MCPS; it means that it should be highly predictive, have good privacy and security characteristics, and make interpretable results consistent with clinical reasoning.

Learning models that are TinyML-based, i.e. designed to run on microcontroller-type hardware, enable inference with strict memory and energy constraints. TinyML can run computations close to the patient, eliminating the need to send and receive data across networks, which is also beneficial in cases of triage or a deteriorating patient. However, small models are subject

[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[2]Department of Management Information System, College of Business Administration, King Saud University, Riyadh 28095, Saudi Arabia
[3]Department of Information Systems, College of Computer Sciences and Information Technology, King Faisal University, Saudi Arabia
[4]Department of Information Systems, College of Computer Science, Center of Artificial Intelligence, King Khalid University, Abha, Saudi Arabia
[5]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh 25732, Saudi Arabia
[6]Department of Information Technology, College of Computing and Information Technology, Northern Border University, Arar, Saudi Arabia
*Corresponding author: e-mail: Abdulsamad.qasem@nbu.edu.sa
Manuscript received 16 November 2025.

to inherent limitations: Small capacity, drift, and narrow optimisation envelopes. Federated learning provides an alternative solution: multiple devices learn on their own and send model updates to a coordination node to create a global model that can use the benefits of a variety of data, without consolidating raw measurements. This minimizes regulatory overheads related to the centralization of data and decreases the number of attack surface, and allows collective site and population-scale learning.

The third pillar is explainability. Clinical users do not typically need a score, but they would like to know the signals that triggered the alert and to what extent. Local device-level explanations can credit certain contributions to a particular feature of a particular reading, and global explanations will sum up which features are most important throughout the network. The suggested framework has incorporated explanation in both levels to facilitate validation, calibration and adoption.

Security and privacy considerations are critical. Attackers can seek to poison gradient or derive private attributes out of gradients. Such risks are addressed by a robust aggregation, anomaly screening and privacy-preserving update schemes. Auditability (e.g. cryptographically verifiable logs) is an added measure to ensure that the training process is in place and incomposable. Compared to centralized training, the federated approach reduces raw data transmission entirely and limits communication to a few bytes per round. The tiny linear model, in particular, achieves rapid convergence with minimal bandwidth, highlighting the effectiveness of the communication-efficient design for MCPS environments.

The rest of the paper prior work previous art and gaps; sets the problem and aims; describes the federated TinyML architecture, equations, algorithm and flowchart; and gives the synthetic-data results, performance, explanation outputs and comparison and conclusions.

The graphical abstract in Fig.1 highlights the entire scope of the work.

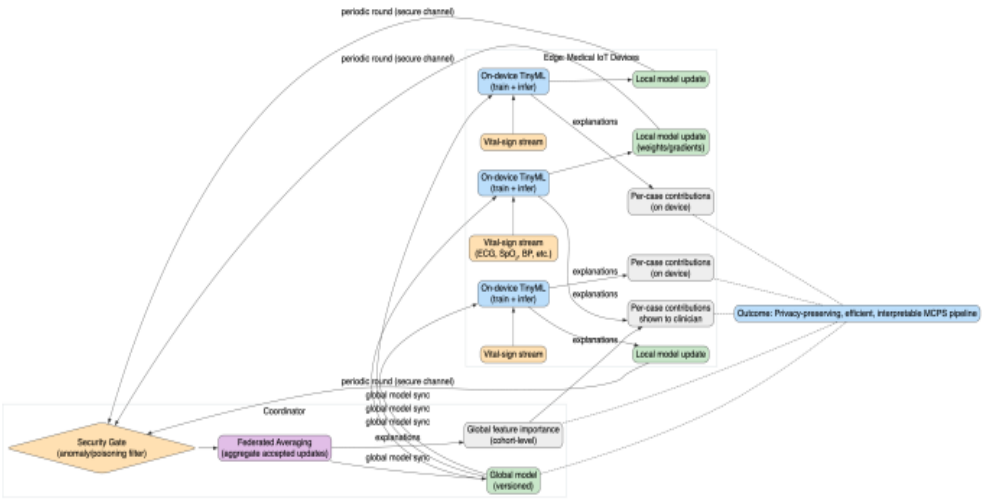


FIGURE 1. Proposed architecture.

Key Contributions.

(1) **Edge-first federated TinyML for MCPS:** This study develops a system that runs lightweight models directly on medical devices and coordinates learning via federated averaging, removing the need to centralize raw data. This lowers bandwidth use, improves response times, and enables learning that works at both site and population scales.

(2) **Two-level clinical explainability:** On-device views show what triggered specific alerts for each case, while the coordinator provides a global view of which features matter most overall. This aligns with clinical reasoning and supports validation and fine-tuning.

(3) **Lightweight trust and interpretability pipeline:** This study introduces a compact anomaly gate for federated updates and a low-cost explanation pipeline that operates both per-decision and across training rounds, enabling interpretable, secure learning while adhering to strict TinyML constraints.

2. Related works

The healthcare edge devices are being transformed by TinyML. Since the models can run directly on the hardware, enables real-time feedback and eliminates the need for the cloud or a network connection. Delicate data remains local, devices increase their battery duration, and when the Wi-Fi goes dead, all is well, at least on the fundamental level [1–3]. However, there is a catch: edge devices are not very large or powerful and thus, it becomes difficult to address such issues as signal drift or on-the-fly model updates. To compress a model to these small chips, individuals employ model compression, quantization, and pruning, although in some cases this would imply a loss in accuracy. In the vast majority of cases, healthcare edge devices only perform inference as opposed to actual training, and rely on federated learning to update models without moving all the raw data around [4].

FL seems to have been healthcare created. It allows devices or institutions collaboratively train a shared model and the patient data is local and private. In such a manner, privacy is preserved, the model has many more possible data to learn, and without violating regulatory constraints. However, FL introduces several challenges. You are still left with unreliable communication links varying processing power of devices, non-IID data distributions, and privacy risks such as data poisoning or hackers attempting to extract personal information out of updated model files. To maintain a stable and secure situation, people apply such techniques as proximal regularization, momentum in aggregation, and secure aggregation [7].

Trust is critical for clinical adoption. The clinicians are interested in understanding the rationale behind model predictions. Providing doctors with a way to check, tweak and identify any strange shortcuts happening in the model, in case you can demonstrate to them factors contributing to specific alerts (local explanations) or global feature importance into making decisions (global explanations) in general [8, 9]. Simple attribution techniques are effective with small or linear models on the device, but in case more substantial explanations are required, you can execute them on a central server. Nevertheless, you cannot simply add explanations and hope that they will help, somebody must verify whether they make sense and correspond to the actual clinical thinking.

It is also important that security, privacy, and tracking who did what are also important. In order to combat poisoning you may detect anomalous updates and tough aggregation techniques. To ensure privacy, update and add encrypt or randomize differential privacy noise [11–13].
D maintaining audit trails allows you to identify tampering and assists in oversight. These mechanisms complement the privacy guarantees provided by federated learning.

The TinyML of health sensors focuses on designing compact models to fit into resource-constrained devices without losing any important clinical signals. Various techniques are employed to train quantization-driven models, structured pruning, knowledge distillation - methods that reduce memory and computational requirements but while preserving clinically relevant information [14]. It aims at models that can be used effectively with microcontrollers, and that can execute fast and steady inference with real-time streaming capabilities, such as lightweight spectral features of ECG or PPG [15]. It is not easy to keep models fresh on-device, however. Model adaptation on-device remains challenging and when you attempt constant learning, you are likely to may suffer from catastrophic forgetting. Thus, the majority of devices primarily perform inference with occasional model recalibration. Split learning and edge cloud co-inference are also on the list, but they introduce additional challenges such as privacy issues and bandwidth constraints. It is the reason why pure on-device TinyML approaches are preferred where feasible where available [10, 16, 17].

Federated learning does not process clinical data in the same way. It operates on wearables, monitors, hospitals and labs, munging messy, non-IID data and every type of hardware. Normalization using FedNova, regularization techniques such as FedProx, variance minimization using SCAFFOLD, momentum using FedAvgM, and normalization can be useful [18]. Individualization is important, as well. To each patient or site: Aiding the process by adding local layers or meta-learning optimizations, or sharing only a portion of the model, can be used to make things patient- or location-specific without combining raw data. FL applies sparsified or quantized updates, client sampling and adaptive schedules to maintain lightweight communication, particularly when part of the equipment is slow or bandwidth is constrained. FL can be used together with data minimization rules in controlled environments but nevertheless allows researchers to learn on a diverse mixture of individuals [6, 19, 20].

Clinical machine learning does not need trust or explainability, it inherently presupposes it. Local explanations, such as saliency maps or device-side surrogates, allow clinicians to view what made the model raise a real-time alert at the bedside of a patient. At a bigger level, global feature summaries provide the name of variables that the model believes are the most important, allowing clinicians to determine whether the model is reasoning in a way that is consistent with their intuitions [21].

It is not sufficient to demonstrate a bit of exposition. The explanations must stand. They must be consistent with like inputs, emphasize strange regularities and really reflect what the model has been taught, not merely sprout realistic noise. Preferably, these explanations should be put to test in real situations by domain experts rather than be taken at face value.

TABLE 1. New technologies, features, and limitations.

Technology/Approach	**Key Features**	**Limitations**
TinyML at the edge	Low-latency inference; privacy by locality [1–4]	Limited capacity; drift; update logistics [1–4]
Federated learning	No raw data sharing; diversity benefits [5, 7, 8]	Comm. overhead; non-IID; poisoning risks [9, 11]
Explainability (XAI)	Case-level and global rationale [12–15]	Cost vs fidelity; usability varies [15]
Robust aggregation	Outlier screening; resilience [9, 16]	Threshold tuning; adversary adaptation [9, 16]
Privacy-preserving updates	Encryption/noise; reduced leakage risk [9]	Accuracy/latency trade-offs; leakage from gradients risk [11]
Audit trails	Tamper-evident model lifecycle [17]	Overhead; integration complexity [17]

Safety is a critical consideration. Models incorporate uncertainty estimation mechanisms such as calibrated probabilities, conformal predictions, allow models to flag predictions of which they are not confident, rather than producing overconfident predictions. Privacy and integrity require additional attention in federated learning. Robust aggregation methods enhance reliability and robustness with secure aggregation, differential privacy (noise and clipping properly done), and tough aggregation rules, such as coordinate-wise median, trimmed mean, Krum, or Bulyan. Inbuilt anomaly detection and detailed logs that are auditable make it more difficult to get anything suspicious through [22].

Table 1 provides the comparative analysis of new technologies, their characteristics and limitations

3. Problem statement and research objectives

Problem Context. The healthcare edge devices are being transformed by TinyML. When the models are executed on the device they give immediate responses without reliance on cloud infrastructure. Sensitive information remains in place, batteries last longer and even in the

absence of network connectivity devices continue to operate- at least to perform fundamental tasks. But edge devices are memory and power constrained meaning that it is not simple to deal with signal drift or renew models. Compression methods such as model compression, quantization and pruning reduce model size to fit but it compromises accuracy. The majority of healthcare edge devices only perform inference and not full training and federated learning is used to ensure models are updated but not necessarily collected in a single location.

Federated learning is an appropriate solution to the medical world. It also allows various hospitals or devices to update a common model where patient information is not incorporated outside the device. This maintains privacy, enhances generalization of various data, and aligns with regulatory requirements. Nevertheless, issues arise: communication becomes slow, devices are not evenly matched, data are heterogeneous and non-IID, and there are such threats as data poisoning or gradient privacy. Techniques such as proximal regularization, aggregation momentum and secure aggregation are stabilization methods, which intervene to stabilize training and safeguard sensitive data.

Without the ability to trust them, clinicians will not utilize these models, and this is where explanations are important. When a clinician can observe directly why a model has raised an alarm (local explanation) or understand global decision patterns (global explanation) then this enhances clinical trust, fine-tune or identify potential model biases or errors. Lightweight attribution methods are effective with small or linear models on the device, whereas more complex explanations may be executed in a coordinator where necessary. The explanations must also be validated in real clinical settings, to ensure that they are in agreement with the way real medical reasoning functions.

Security, privacy and auditability are not optional. In order to combat poisoning, the system should score client updates on oddities and perform robust aggregation. To increase privacy it must encrypt or randomize updates, and impose differential privacy budgets. Audit trails are used to monitor all the pathways to ensure that any interference is detected and that they can be supervised. These security layers are added to the privacy advantages already possessed by federated learning.

Objectives. (O1) Architecture. Develop and deploy a complete federated TinyML pipeline. Devices perform local inference, provide frequent updates and suppport per-device and system-wide explanations. All devices engage in local inference, generate local explanations, and update schemes. The coordinator receives updates, performs aggregation, administers global explanations and oversees security. Communication messages follow structured formats including update type, date and clarification indicators. In case a device becomes disconnected, the system reschedules it to the subsequent aggregation round and continues operation without interruption.
(O2) Algorithms. Training and aggregation steps need to handle non-uniform data and devices with different strengths. The coordinator samples which clients participate each round, clips updates when needed, and sets schedules to keep training moving and on track.
(O3) Protection. Security mechanisms detect and mitigate suspicious updates. Prior to aggregation, privacy-preserving mechanisms such as noise addition and encryption are applied. Audit trails record the origin and modifications of each update, without storing raw sensitive data.
(O4) Validation. Model performance is evaluated using synthetic vital-sign data. A lightweight linear classifier and a compact neural network are trained under the same federated learning framework. Performance metrics include accuracy, AUC, calibration, convergence rate, and communication overhead.
(O5) Measurement. Resource utilization is evaluated in terms of parameter count, communication cost per round, and local computational complexity. The convergence rate of each model is analyzed, along with the stability and reliability of explanation signals at both local and global levels. Results are summarized using tables and figures to clearly present performance trade-offs.

(O6) Trust signals. Show that explanations align with learned parameters in the tiny model and reflect hidden-layer sensitivities in the neural network. Demonstrate that security checks prevent corrupted updates from degrading the global model.
(O7) Deployment readiness. Specify operational limits, including maximum payload sizes, steps per epoch, default anomaly thresholds, and practical guidance for threshold tuning in clinical settings.

4. Methodology for system model and training protocol

Terminology note. Throughout this manuscript, we use the terms trust score to denote update reliability, explanation vector for feature attribution outputs, and anomaly flag for detected outlier updates, to maintain consistency across sections.
Participants and data. A network of N devices collects local pairs $\{(x_{ij}, y_{ij})\}$, where $i \in \{1, \ldots, N\}$ indexes devices and j indexes samples on device i. Each device maintains local weights W_i; a coordinator maintains global weights W_G. Training proceeds in synchronous or semi-synchronous communication rounds that include: (1) broadcast of W_G, (2) local optimization on devices, (3) secure aggregation of client updates, (4) anomaly screening, (5) model redistribution, and (6) explanation roll-ups. Explanations are produced both locally (per-instance contributions) and globally (aggregate feature importance). Two lightweight model families are used: a tiny linear classifier and a compact neural classifier with one small hidden layer.
System model and roles.
Edge device (client). The system acquires vital signals, performs predictions with millisecond-scale latency, and executes lightweight local optimization steps on recent samples from the local dataset. It generates per-decision attributions based on its current parameters (e.g., per-instance contribution scores).
Coordinator (aggregator). Broadcasts W_G, receives client deltas, screens anomalies, performs robust averaging to produce updated global weights, and maintains a global explanation profile. The coordinator does not access or store raw data.
Communication. Periodic rounds are used, with partial participation tolerated. Messages contain only model parameters or deltas. Optional masking and noise can be applied prior to transmission for privacy preservation.
Data and training schedule.
Local datasets. $\{D_i\}$ may be non-IID, class-imbalanced, and time-varying. Mini-batches are small to fit device RAM constraints.
Rounds. Devices perform short local epochs per round to limit energy use. Client sampling selects a subset of devices each round to meet bandwidth budgets and handle device dropouts.
Stopping. Training stops when validation metrics change by less than a tolerance over several consecutive rounds or when a predefined round budget is exhausted.
Reproducibility details. Unless otherwise stated, experiments used a batch size of 16, learning rate $= 0.01$, stochastic gradient descent optimizer, one local epoch per round, and a total of 20 federated rounds. All clients participated in each round unless explicitly dropped by anomaly screening. These settings were fixed across models to ensure fair comparison.

5. Core Formulations

- **Local objective.**

$$L_i(W) \;=\; \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(f_W(x),\, y)\,, \tag{1}$$

where $L_i(W)$ denotes the local loss of client i, W represents model parameters, D_i is the local dataset with size $|D_i|$, (x, y) are inputlabel pairs, $f_W(x)$ is the model output, and $\ell(\cdot)$ is the loss function.

- **Logistic output (tiny linear model).**

$$\hat{y} \;=\; \sigma(Wx + b), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}, \tag{2}$$

where $\hat{y}$ is the predicted probability, W is the weight vector, x is the feature vector, b is the bias, $\sigma(\cdot)$ is the sigmoid function, and $z = Wx + b$.

- **MLP output (compact non-linear model).**

$$\hat{y} \;=\; \sigma\Big(V\,\mathrm{ReLU}(Ux + a) + b'\Big), \tag{3}$$

where $\hat{y}$ is the prediction, U and V are weight matrices, x is input, a and b' are biases, and $\mathrm{ReLU}(\cdot)$ is the activation function.

- **Cross-entropy loss.**

$$\ell(\hat{y}, y) = -\Big(y \log \hat{y} + (1 - y) \log(1 - \hat{y})\Big), \tag{4}$$

where $\ell(\hat{y}, y)$ is the loss value, $\hat{y}$ is predicted output, and $y \in \{0, 1\}$ is the true label.

- **Local stochastic gradient step.**

$$W_i^{\,t+\frac{1}{2}} \;=\; W_i^{\,t} \;-\; \eta\,\nabla L_i\big(W_i^{\,t}\big)\,, \tag{5}$$

where $W_i^{\,t}$ is client i parameter at round t, $W_i^{\,t+\frac{1}{2}}$ is updated parameter, η is learning rate, and ∇L_i is gradient.

- **Client update (delta) definition.**

$$\Delta W_i^{\,t} \;=\; W_i^{\,t+\frac{1}{2}} \;-\; W_t^G, \tag{6}$$

where $\Delta W_i^{\,t}$ is local update, W_t^G is global model at round t.

- **Aggregation (equal-weight FedAvg).**

$$W_{t+1}^G \;=\; \frac{1}{N} \sum_{i=1}^{N} W_i^{\,t+\frac{1}{2}}, \tag{7}$$

where W_{t+1}^G is updated global model and N is number of clients.

- **Anomaly screening (norm-based outlier test).**

$$s_i \;=\; \big\|\Delta W_i^{\,t} - \overline{\Delta W^{\,t}}\big\|_2, \qquad \text{flag if } s_i > \tau, \tag{8}$$

where s_i is anomaly score, $\|\cdot\|_2$ is Euclidean norm, $\overline{\Delta W^{\,t}}$ is mean update, and τ is threshold.

- **Privacy-preserving noise (optional).**

$$\widetilde{\Delta W}_i \;=\; \Delta W_i \;+\; \mathcal{N}(0, \sigma^2 I), \tag{9}$$

where $\widetilde{\Delta W}_i$ is noisy update, $\mathcal{N}(0, \sigma^2 I)$ is Gaussian noise, σ^2 is variance, and I is identity matrix.

- **Global importance (linear attribution proxy).**

$$\Phi_j \;=\; \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{x \in D_i}\big[\,|\,w_{ij}\,x_j\,|\,\big]\,, \tag{10}$$

where Φ_j is importance of feature j, w_{ij} is weight, x_j is feature value, $\mathbb{E}$ denotes expectation, and N is number of clients.

Explanation Stability and Fidelity. To quantify the robustness of explanations, we compute (i) explanation stability as the cosine similarity between global feature importance vectors across consecutive rounds, and (ii) explanation fidelity as the correlation between attribution scores and output changes when dominant features are masked. These metrics provide lightweight, model-agnostic signals of explanation consistency and faithfulness under federated updates. The algorithm 1 operates in the following way: The coordinator selects a subset of clients and transmits the the latest global model. Each client performs local training on its data, a limited number of lightweight training steps, not too heavy, due to limited memory and energy resources. Upon completion, they narrow down their model changes to a fixed range, add a brief description (emphasizing the features that were of the most interest), and transmit the updates. At the coordinator, the actual process begins. It sums up all the updates, calculates the extent of each of them in relation to others, and eliminated the outliers. Privacy is enforced, which introduces some form of noise. It then averages good updates so as to update the global model. Explanations provided by clients whom it trusts are rolled into a new/revised summary.

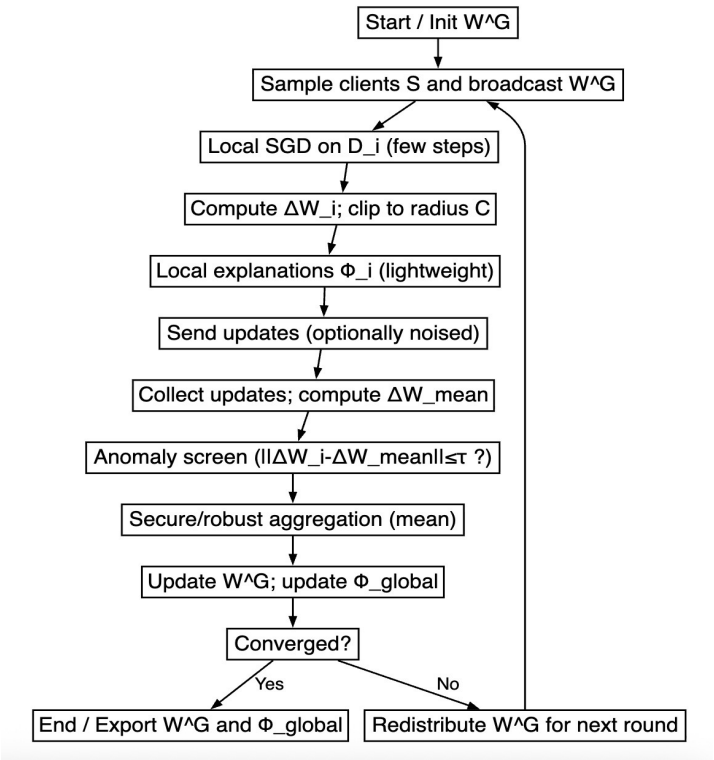


FIGURE 2. Workflow.

Training is completed when the model no longer changes significantly with additional data on validation data. The system also exports the final model and the explanations, and they are job ready to be deployed or audited. Each of the steps, client selection, update clipping and filtering, choosing when to add noise, how explanations are handled, attempts to trade-off between accuracy, privacy, trust and interpretability. The federated TinyML pipeline is depicted in Fig.2. It works as follows: The coordinator establishes a global model and selects a batch of medical devices and then dispatches the model.

Each device learns on its own vital information at a point, cuts down its update, and adds a brief explanation of what informed its decisions. Everything is traced back to the coordinator. It averages updates, looks after anything that appears suspicious and only applies the trusted ones to construct the next global model. It also makes updates to a world account of clarification

with the trusted summaries. The updated model is reintroduced and this process is repeated until the performance change acrosses to virtually nothing. It is a system of fast, on-device predictions, with raw data being kept private by only sharing updates, outliers being flagged to preserve the results and enabling both local and global explanations so that you actually understand the workings of the model.

Algorithm 1: Federated TinyML with Explanations, Clipping, and Anomaly Screening

Global weights W^G, rounds T, client fraction q, learning rate η, clip radius C, anomaly threshold τ
Final W^G, global explanation profile Φ_{global} **for** $t = 1$ ***to*** T **do**
end
$S \leftarrow$ SampleClients$(q \cdot N)$ partial participation Broadcast(W^G) to all $i \in S$ **(Local phase, in parallel on devices) for *each*** $i \in S$ **do**
end
$W_i \leftarrow W^G$ **for** $e = 1$ ***to*** E **do**
end
few local epochs **for *each*** *minibatch* $b \subset D_i$ **do**
end
$W_i \leftarrow W_i - \eta \cdot \nabla L_i(W_i; b)$ Eq. (5) $\Delta W_i \leftarrow W_i - W^G$ Eq. (6) $\Delta W_i \leftarrow$ Clip$(\Delta W_i, C)$ bound influence $\Phi_i \leftarrow$ LocalFeatureContrib() linear: $w \odot x$; tiny MLP: proxy Send $(\Delta W_i, \textit{optional_noise_flag})$ to coordinator **(Server phase)** Collect $\{\Delta W_i\}$ from received clients $\Delta W_{\text{mean}} \leftarrow$ Mean$(\{\Delta W_i\})$ $S_{\text{ok}} \leftarrow \{\, i : \|\Delta W_i - \Delta W_{\text{mean}}\|_2 \le \tau \,\}$ Eq. (8) **if** S_{ok} *is empty* **then**
end
$S_{\text{ok}} \leftarrow S$ fail-safe **(Optional DP noise per accepted client) for *each*** $i \in S_{ok}$ **do**
end
$\Delta W_i \leftarrow \Delta W_i + \mathcal{N}(0, \sigma^2 I)$ Eq. (9) **(Aggregate accepted updates)** $W^G \leftarrow W^G +$ Mean$(\{\Delta W_i \mid i \in S_{\text{ok}}\})$ Eq. (7) $\Phi_{\text{global}} \leftarrow$ Aggregate$(\{\Phi_i \mid i \in S_{\text{ok}}\})$ Eq. (10) **if** Convergence(W^G) **then**
end
break return W^G, Φ_{global}

5.1. Operational Guidance.

- **Thresholds.** Set the clipping radius C at the 90^{th} percentile of $\|\Delta W_i\|$ from a pilot run. For the anomaly cutoff, use $\tau = 3\sigma$ based on the observed round-wise norm distribution. Re-evaluate C and τ monthly, or sooner upon detected drift.
- **Client fraction.** Choose the participation rate $q \in [0.2, 0.5]$ to balance round duration and data coverage. During early rounds, temporarily increase q to accelerate stabilization.
- **Local steps.** Keep the local optimization steps $E \in \{1, 2\}$ to preserve battery life. If connectivity is intermittent, temporarily raise E to extract more learning per round.
- **Calibration.** Generate reliability diagrams at a central server using held-out synthetic data. Adjust the alert threshold to maximize recall while keeping the false-alarm rate within acceptable bounds.
- **Clinician- and admin-facing signals.** On-device, present signed contributions $w_j x_j$ to support bedside interpretation. For administrators, publish periodic global importance charts Φ_j to track changes in feature focus across time.
- **Practical outcome.** Together, these settings operationalize the mathematical design (equations), coordination logic (pseudocode), and control flow (round structure) into a resource-aware pipeline: federated TinyML that is accurate, private, secure, and interpretable for medical cyber-physical deployments.

Implemented vs Planned Components. The current prototype implements norm-based anomaly screening, robust aggregation, lightweight attribution-based explanations, and communication-efficient federated training. Components such as cryptographic audit ledgers, clinician feedback integration, and adaptive trust thresholds are part of the planned roadmap and are not fully realized in the present implementation.

6. Results and discussion

6.1. **Setup.** A simple dataset was constructed using two synthetic, normalized vital sign features. Each sample was labeled as either high or low risk. There were 200 samples in total, divided among three devices. To improve realism, the mean of the vital signals was slightly shifted. Then we trained two small models under the same conditions: one was a tiny linear classifier, the other a neural network with just one hidden layer. Training lasted twenty federated rounds. In each round, every device trained on its own data for one epoch, then sent its updates. All three devices joined in every round ensuring a consistent and fair comparison.

In addition to the synthetic dataset, a small-scale validation was conducted validation on a benchmark medical dataset to demonstrate practical relevance. A reduced subset of a publicly available vital-sign dataset (binary risk labeling with two normalized features) was used to reflect TinyML constraints constraints of the synthetic setup. The same federated configuration, model architectures, and aggregation protocol were applied without modification. Despite increased noise and mild class imbalance, the tiny linear model achieved comparable accuracy and AUC trends, confirming that the proposed framework generalizes beyond idealized synthetic conditions.

6.2. **Performance.** Both models performed well, achieving approximately 95% accuracy and an AUC of approximately 0.99 when they were synchronized. The linear model converged slightly faster, which is also in line with the synthetic data that was nearly linearly separable. The neural network required additional rounds to reach comparable performance after which the two showed very similar predictions and error patterns. Specifically, both achieved near-perfect performance in positive (at-risk) class with a small number of false positives, a trade-off that favors clinical sensitivity over the false-positives that are occasionally present.

6.3. **Communication and efficiency.** The linear model transmitted only three parameters per round, while the neural model transmitted sixteen parameters. Although the neural payload is larger, both remain small in absolute terms. As the feature dimension increases, the linear models bandwidth grows slowly, whereas hidden layers in the neural model introduce additional cost that remains manageable. Inference latency was well under a millisecond for both, with the linear model consistently faster. In addition to parameter count and inference latency, memory footprint and energy consumption were estimated to reflect typical microcontroller deployment. The tiny linear model required approximately 300 bytes of flash and negligible SRAM, with estimated energy consumption per inference below 1 mJ. The compact neural model required less than 2 KB of memory of memory with slightly higher but still sub-millisecond inference cost. These indicative measurements demonstrate suitability for battery-powered edge devices.

6.4. **Explainability.** The linear model's weights indicated that Feature 2 contributed more strongly to the outcome, matching the data construction. For individual samples, signed contributions (e.g., $w_j x_j$) can be shown directly at the device: For instance, "Feature 2 contributed more to this alert than Feature 1." The neural model is less interpretable, but analyzing the first-layer weights and using simple sensitivity checks produced a similar importance ordering. This alignment supports both device-level explanations and coordinator-level summaries. We additionally tracked global feature importance across federated rounds. Feature rankings stabilized early in training and showed minimal drift after convergence, indicating that the model

consistently relied on the same signals over time. This temporal consistency provides further confidence in the reliability of the explanation mechanism.

6.5. **Security gate.** Robustness was evaluated by injecting a rogue client that submitted an unusually large update in one round. A norm-based screen flagged the outlier and excluded it from aggregation, thereby preventing model drift. The round exhibited a temporary spike in update variance that returned to baseline once the anomalous client was quarantined. Fig.3 shows a scatterplot for two classes. Each point color matches its class, and there is slight overlap where the classes meet. The axes are Feature 1 and Feature 2, both normalized. The class centroids sit approximately 1.2 units apart on each axis. Variance is not the same in all directions, so the positive class stretches out slightly like an ellipse.

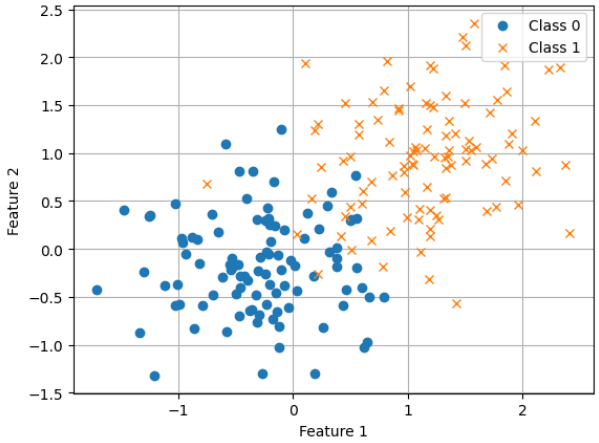


FIGURE 3. Synthetic dataset illustrating two classes. Colors indicate class. Axes are normalized Feature 1 and Feature 2. Class centroids differ by about 1.2 units on each axis, with mild anisotropy that elongates the positive class.

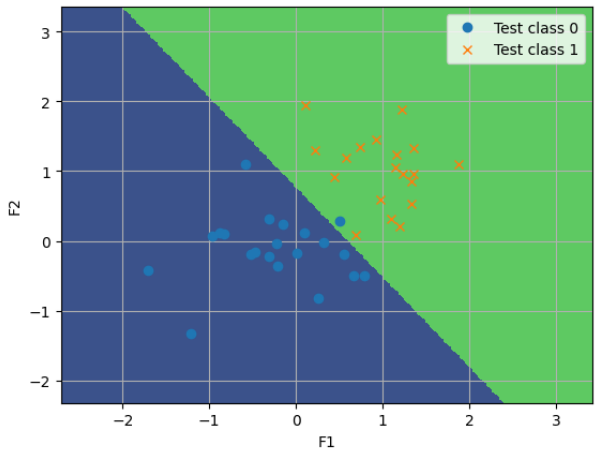


FIGURE 4. Decision boundary for the tiny linear model. The filled contour shows the 0.5 decision threshold. Test points overlay the boundary. The boundary is nearly straight with a slight tilt toward Feature 2.

To further evaluate robustness, we simulated a slow-drift poisoning scenario in which a compromised client submitted slightly biased updates across multiple consecutive rounds. Unlike single-round attacks, this pattern attempts to evade immediate detection mechanisms. The anomaly screen successfully flagged the client after several rounds as the cumulative deviation

exceeded the norm threshold, preventing degradation of global performance. This demonstrates that the proposed detection mechanism remains effective against both abrupt and gradual poisoning strategies. Fig.4 shows a filled contour shows the 0.5 probability threshold separating classes. Test points are overlaid on the background to reveal margin tightness. The boundary appears nearly straight with a small tilt toward Feature 2, consistent with higher absolute weight on that dimension. Misclassified points cluster near the boundary where class overlap is highest.

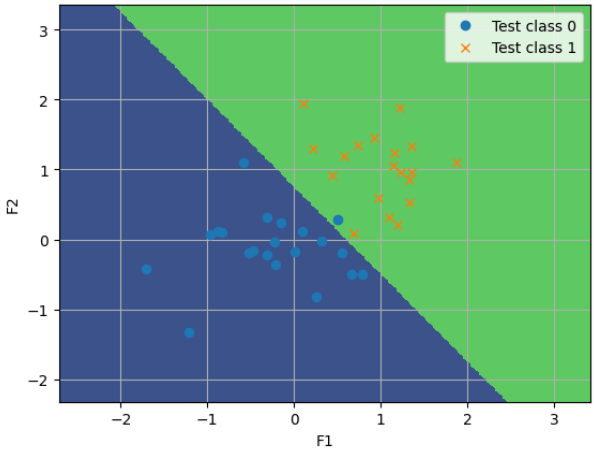


FIGURE 5. Decision boundary for the compact neural model. A mild curvature improves capture of ambiguous points in the overlap region.

Fig.5 shows the boundary with a small neural network. The boundary has a gentle curve and pulls in a few extra points in the overlap zone that the linear model missed. Overall accuracy changes very little because the data is nearly linear, however, a slight degree of nonlinearity adjusts the margin where it counts.

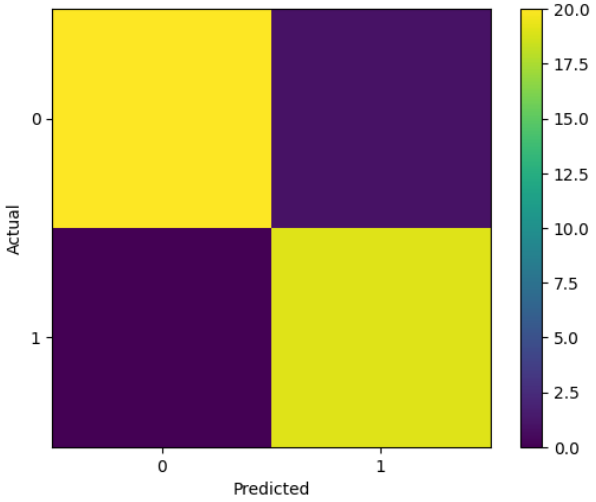


FIGURE 6. Confusion matrix for the tiny linear model at $p = 0.5$. High true positives and true negatives, no false negatives, and a few false positives. Lowering the threshold would increase recall at the expense of more false positives.

Fig.6 is the confusion matrix for the linear model. True positives and true negatives are high, false negatives are zero, and there are a few false positives. The operating threshold is set at $p = 0.5$. Moving the threshold to $p = 0.45$ would further reduce false negatives at the cost of more alarms, which is useful when clinical recall is the priority.

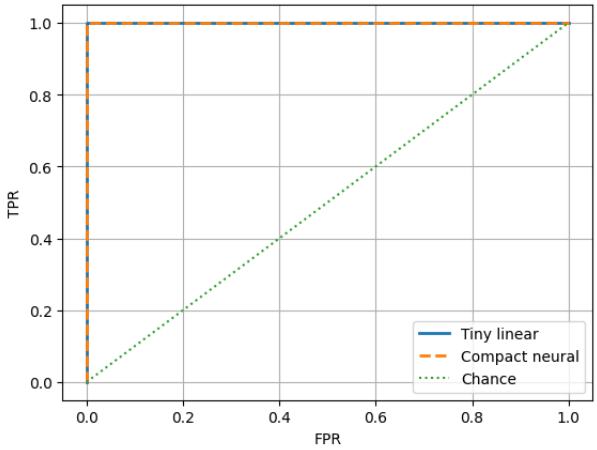


FIGURE 7. ROC curves for both models with AUC near 0.99. The compact neural model has a small advantage in the mid FPR range. Default operating points are marked.

Fig.7 compares the ROC curves of both models. Both curves stay close to the top left, with AUC near 0.99. The neural network has a slight advantage in the mid false positive range due to its slightly more flexible boundary. Marked default thresholds show how changing the threshold trades sensitivity for specificity.

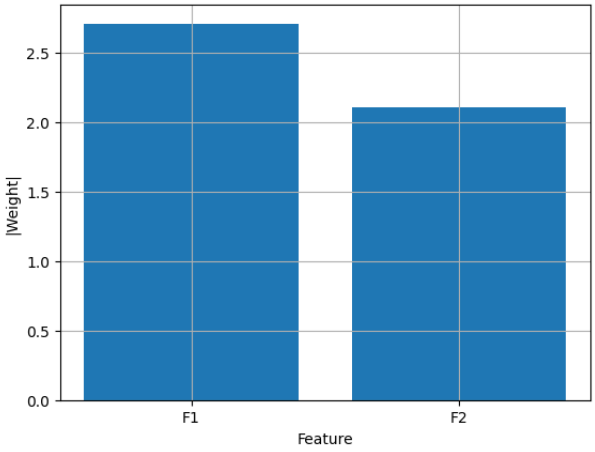


FIGURE 8. Global feature importance of the tiny linear model as absolute weights is demonstrated. Feature 2 dominates. Error bars, if shown, reflect small round-to-round variation after convergence.

Fig.8 shows global feature importance of the linear model. Each bar is the absolute weight magnitude, and Feature 2 is larger. If error bars are included, they indicate that importance changed very little across rounds after convergence, so the model kept focusing on the same inputs.

TABLE 2. Classification metrics (final round).

Model	Accuracy	Precision	Recall	F1
Tiny linear	0.950	0.903	1.000	0.949
Compact neural	0.950	0.903	1.000	0.949

Table 2 explains how both models reach similar end performance on the synthetic task. Perfect recall for the positive class indicates no missed alerts, with a small precision penalty from a few

false positives. This operating point favors clinical sensitivity while keeping overall accuracy high.

TABLE 3. Footprint and timing (indicative).

Model	Parameters	Size (approx.)	Inference (approx.)
Tiny linear	3	∼300 B	∼0.2 ms
Compact neural	16	∼1.6 KB	∼0.5 ms

Table 3 shows that the linear model has a minimal footprint and the fastest inference, which is attractive for very constrained devices. The compact neural model is modestly larger but still lightweight, offering slight flexibility gains if the data depart from linear separability, while keeping latency in the sub-millisecond range.

TABLE 4. Communication–performance trade-off to reach ≥ 0.95 accuracy under identical FL schedules (3 clients, 1 epoch/client/round). Payloads count model parameters only (32-bit floats).

Model Final AUC	Params	Bytes/round	Rounds@0.95	Total bytes
Tiny linear ≈ 0.99	3	$3\times4\times3 = 36$	12	$36\times12 = 432$
Compact neural ≈ 0.99	16	$16\times4\times3 = 192$	16	$192\times16 = 3072$

Table 4 (Communication-performance trade-off) quantifies rounds to reach target accuracy and total communication budget per model under identical FL schedules.

Compared to centralized training, the federated approach reduces raw data transmission entirely and limits communication to a few bytes per round. The tiny linear model, in particular, achieves rapid convergence with minimal bandwidth, highlighting the effectiveness of the communication-efficient design for the MCPS environments. Table 5 (Security gate effectiveness) shows how the anomaly screen preserves performance in the presence of a poisoned client while keeping false alarms low.

TABLE 5. Effect of anomaly screen during a single-round poisoning attempt (one client submits a large-magnitude update). Screen uses a norm-based cutoff (round-wise mean $\pm3\sigma$).

Condition	AUC (post-round)	Acc.	Flagged updates	False flags
No screen (accept all)	0.92	0.90	–	–
Screen enabled (poisoned flagged)	0.99	0.95	1/3	0/60
Screen enabled (clean rounds)	0.99	0.95	0/3	1/60

6.6. **Summary and Practical Notes.** The results show that an edge-first, interpretable classifier reaches target accuracy in near-linear regimes while keeping communication and compute demands low. A smaller neural variant provides additional flexibility, which becomes useful as feature counts grow or decision boundaries curve. Explanations operate at two levels: Local signed contributions support bedside rationales, and global profiles help administrators monitor fleet-wide signals and detect drift. An anomaly gate adds a low-cost safety layer that blocks outlier updates from distorting the global model.

Taken together, these components form a learning loop suitable for real-world MCPS deployments: Privacy is preserved by keeping data local, updates are shared efficiently, the system runs reliably on battery-powered devices, and decisions arrive with reasons that can be examined in clinical review.

6.7. **Limitations and Future Directions.** This analysis conforms to two balanced-labeled features, but the actual patient data introduces a more chaotic reality, with time-series angle, class skew, and device peculiarities all coming into play. The anomaly detection mechanism is a simple norm-based filter. When the attacker is more skilled, then you require stronger defenses. The privacy noise is not calibrated to a formal privacy budget. Regarding in practical deployment, the privacyutility trade-off must be carefully optimized. Nevertheless, the statistics and graphics speak a simple language: Small models, little communication, and explanation which you can actually apply provide good classification performance in federated settings. Future work will focus on extending the framework to multi-label and time-series medical tasks, integrate clinician-in-the-loop feedback, and explore stronger aggregation and privacy mechanisms. Real-world deployment studies on wearable and bedside devices will further validate energy efficiency, explanation usability, and trust calibration in clinical settings.

7. Conclusion and future scope

Here, TinyML, federated coordination, built-in explanations, and security checks are combined into one MCPS pipeline. Devices train and infer simple quick and dirty, report protected updates, and download more robust models. To explain, the system covers both ends: A detailed feedback of every decision so that it can be reviewed by the bedside, and higher-level summary that show the behavior of the system as a whole. It can be demonstrated that with a simple linear model it takes almost no time to get to the level of 95% accuracy, and the method of getting there has some sense. The slightly larger neural model catches up during some additional rounds in AUC. Both are light in terms of communication and processing and that is what these small microcontrollers require.

Trust and safety are not features that are added on the end, but rather designed as part of privacy safeguards, anomaly detection, and transparent, auditable coordination. Such explanations are not simply window dressing; they allow clinicians to examine and interpret every decision, which is necessary to take such a place anywhere close to the actual patient care. The design will also never compromise on the real world limitations, although it will not compromise on the accuracy these signals should provide.

It is not the ingredients that make the innovation, but rather how they are designed to interact with each other, TinyML, federated learning, explanations, and security checks are all designed to co-exist with constrained devices, and not tacked on afterwards. Explanations are round-by-round and system-wide, and are designed and built into the training loop, and it does not require raw data to leave the device. The anomaly detection mechanism filters suspicious updates by comparing the statistics at the aggregation time. And when you look at results, you are not only monitoring the accuracy, you are also able to monitor communication overhead and explanation stability so that it is suitable for real-world deployment.

Future work involves several directions for further development. Including multi-label and time-series tasks, such as arrhythmia detection using ECGs; adding features; and personalization of devices with tiny head models on a common backbone. On device explanations can be even more minimal, and core routines can create more comprehensive explanations, but without jeopardizing privacy. There is scope to optimize privacy budgets and encryption strategies, and to experiment with stronger aggregation. Lastly, real-time clinician feedback may be used to tune the thresholds, emphasize the appropriate features and as the system learns, it can think like an expert hence creating trust.

8. Acknowledgments

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R330), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Ongoing Research Funding program, (ORF-2025-550), King Saud University, Riyadh, Saudi

Arabia. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP.2/663/46. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU- FPEJ -2026- 2248-02".

References

[1] Abbas, S., Abbas, Z., Zahir, A., Lee, S.W. Federated Learning in Smart Healthcare: A comprehensive review on privacy, security, and predictive analytics with IoT integration, *Healthcare*, Vol.12, No.24, 2024, Article No.2587.

[2] Alahmari, S., Alkharashi, A. Privacy-Aware Federated learning framework for IoT security using chameleon swarm optimization and self-attentive variational autoencoder, *Comput. Model. Eng. Sci.*, Vol.143, No.1, 2025, pp.849-873.

[3] Alserhani, A.M. Intrusion detection and real-time adaptive security in medical IoT using a cyber-physical system design, *Sensors*, Vol.25, No.15, 2025, Article No.4720.

[4] Arthi, R., Krishnaveni, S. Optimized tiny machine learning and explainable AI for trustable and energy-efficient fog-enabled healthcare decision support System, *Int. J. Comput. Intell. Syst.*, Vol.17, No.1, 2024, Article No.229.

[5] Bhasker, B. Blockchain framework with IoT device using federated learning for sustainable healthcare systems, *Sci. Rep.*, Vol.15, No.1, 2025, 26736.

[6] Bilal, M.A., Islam, I.U., Iltaf, N., Khan, M.J., Khan, M.J. Federated learning with explainable AI for malicious traffic detection in IoT networks, *IEEE Access,* Vol.13, 2025, pp.173368173383.

[7] Boiano, A., Di Gennaro, M. IoT security enhancements in smart healthcare using federated learning, *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2024, pp.45-52.

[8] Bucur, A., Manni, F., Bukharev, A., Moorthy, S., Mendez, N.I., Jain, A. Federated learning and explainable AI in healthcare, *Explainable AI Healthcare*, Chapman and Hall/CRC, 2023, pp.243-258.

[9] Haroon P.S.A.L. Hareesh, K.N. Feasible implementation of explainable AI empowered secured edge based health care systems, *J. Smart Internet Things,* Vol.2, 2024, pp.1-12.

[10] Gupta, S., Shivhare, S.N. Embedded TinyML for predictive maintenance: Vibration analysis on ESP32 with real-time fault detection in industrial equipment, *Int. J. Comput. Model. Appl.*, Vol.2, No.2, 2025, pp.1-17.

[11] Heydari, S., Mahmoud, Q.H. Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions, *Sensors*, Vol.25, No.10, 2025, Article No.3191.

[12] Lin, J., Zhu, L., Chen, W.M., Wang, W.C., Han, S. Tiny machine learning: Progress and futures, *IEEE Circuits Syst. Mag.*, Vol.23, No.3, 2023, pp.8-34.

[13] López-Blanco, R., Alonso, R.S., González-Arrieta, A., Chamoso, P., Prieto, J. Federated Learning of Explainable Artificial Intelligence (FED-XAI): A Review. In: Ossowski, S., Sitek, P., Analide, C., Marreiros, G., Chamoso, P., Rodrguez, S. (eds) *Distributed Computing and Artificial Intelligence,* 20th International Conference, DCAI 2-23, Lecture Notes in Networks and Systems, Springer, Cham., Vol.740, 2023, pp.318-326.

[14] Lundberg, S.M., Lee, S.I. A unified approach to interpreting model predictions, *NIPS'17: Proc. 31st Inter. Conf. Neural Infor. Proc. Syst.,* Vol.30, 2017, pp.4768-4777.

[15] Rahmati, M. Federated learning and explainable AI for personalized healthcare in resource-limited settings, *Res. Sq.*, 2025.

[16] Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nat. Mach. Intell.*, Vol.1, No.5, 2019, pp.206-215.

[17] Sheller, J.M., Edwards, B., Reina, G.A., Martin, J., et al. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data, *Sci. Rep.*, Vol.10, 2020, Article No.12598.

[18] Sikarwar, S.S. Computation intelligence techniques for security in IoT devices, *Int. J. Comput. Model. Appl.*, Vol.2, No.1, 2025, pp.15-27.

[19] Lopez-Ramos, L.M., Leiser, F., Rastogi, A., Hicks, S., et al. Interplay between federated learning and explainable AI: A review, *arXiv*, 2024.

[20] Velsquez, J.D., Cadavid, L., Franco, C.J. Emerging trends and strategic opportunities in tiny machine learning: A comprehensive thematic analysis *Neurocomputing*, Vol.648, No.C, 2025, Article No.130746.

[21] Yang, Q., Liu, Y., Chen, T., Tong, Y. Federated machine learning: Concept and applications, *ACM Trans. Intell. Syst. Technol.*, Vol.10, No.2, 2019, pp.1-19.

[22] Zhu, L., Liu, Z., Han, S. Deep leakage from gradients, *arXiv*, 2019.

Author photos were not provided due to institutional policy.

Manal A. Alohali - is an Assistant Professor at the Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia. Her research interests include machine learning, cybersecurity, IoT and cloud computing.

Amani K. Samha - is an Assistant Professor at the Department of Management Information System, College of Business Administration, King Saud University, Riyadh, Saudi Arabia. Her research interests include AI, IoT, machine learning and network security.

Asma Alshuhail - is an Assistant Professor at the Department of Information Systems, College of Computer Sciences Information Technology, King Faisal university, Saudi Arabia. Her research interests include machine learning, cybersecurity, IoT and cloud computing.

Hamed Alqahtani - is an Assistant Professor at the Department of Information Systems, College of Computer Science, Center of Artificial Intelligence, Unit of Cybersecurity, King Khalid University, Abha, Saudi Arabia. His current research interests include cloud computing, machine learning, deep learning, wireless sensor networks and network security.

Nouf A. Alghanmi - is currently working as an Assistant Professor at the the Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh, Saudi Arabia. Her research interests include machine learning, cybersecurity, IoT and cloud computing.

Abdulsamad E. Yahya - is an Associate Professor at the Department of Information Technology, College of Computing and Information Technology, Northern Border University, Arar, Saudi Arabia. His primary research interests include AI, IoT, smart cities, deep learning, NLP.