

A Lightweight Trust-Aware Framework for Secure Medical Data Analytics using Hybrid Homomorphic Encryption and Federated Learning in Resource-Constrained IoT

Kamlesh Thummar¹, Anil Kshatriya²

¹Research Scholar, Electronics and communication department, Gujarat Technological University, Ahmedabad, Gujarat, India.

²Electronics and Communication Engineering Department, Government Engineering College, Gandhinagar, Gujarat, India.

²Corresponding Author:- aj.kshatriya@gmail.com

Received: 21st July, 2026; Revised: 1st Aug, 2026; Accepted: 16th Aug, 2026; Available Online: 13th Sept, 2026

Abstract

The fast evolution of the Internet of Medical Things (IoMT) has completely transformed personalized healthcare through continuous patient monitoring and real-time diagnostics. Nevertheless, the application of deep learning in such resource-constrained environments poses two major challenges in the form of preservation of patients' private data and the computational burden of implementing a secure system. Even though Federated Learning (FL) is a decentralised solution to the problem of data centralisation, it still suffers from the problems of gradient leakage, inference attacks, and model poisoning. In this paper, we propose SecurHealthFL—a lightweight security framework for IoT healthcare devices with limited computational resources that utilises an NTRU based HE scheme, Trust-Aware Diagnostic Engine (TADE) which uses weighted EMAs and TW-FedAvg.

This version of the paper provides results of full, working, end-to-end implementation of the design discussed in this paper, which means that we have implemented not just theoretical NTRU cryptosystem, CNN-GRU diagnostic model, trust engine, and federated learning algorithm but all these elements are running and working right now. For the synthetic 5-class ECG dataset used instead of PhysioNet MIT-BIH (see Section XII), CNN-GRU achieves 98.8% accuracy (ROC-AUC = 0.999). With the 20% malicious-client attack on FedAvg with sign-flip, accuracy drops to 14.4%, while with TW-FedAvg it increases to 31.0%. The NTRU implementation, run at the paper's own parameters ($N=1024$, $q=2^{14}$), achieves exact homomorphic decryption on 198/200 trials of a 50-client aggregation. Building this implementation also surfaced two things worth stating plainly rather than smoothing over: the paper's decryption equation is missing a step needed for correctness (Section VI-B), and its stated differential-privacy budget ($\epsilon=0.1$), applied literally via the standard Gaussian mechanism at this model's parameter count, injects noise roughly **5,000×** larger than the signal it is protecting (Section XV-B). Both are reported as findings, with the reasoning shown, not asserted.

Keywords: Internet of Medical Things (IoMT), Federated Learning, Homomorphic Encryption, Trust Scoring, Lightweight Cryptography, Privacy-Preserving AI, NTRU Lattice, Empirical Reproduction.

How to cite this article: Kamlesh Thummar¹ Anil Kshatriya². A Lightweight Trust-Aware Framework for Secure Medical Data Analytics using Hybrid Homomorphic Encryption and Federated Learning in Resource-Constrained IoT. Int J Drug Deliv Technol. 2026;16(82s):1710-1723. DOI: 10.25258/ijddt.16.82s.198.

I. Introduction

The unprecedented growth witnessed in the Internet of Medical Things (IoMT) has revolutionized modern-day healthcare through the creation of seamless integration between smart sensors, wearables, and diagnostic systems [1], [8]. The number of IoT devices that are being used for real-time patient monitoring and personal telemedicine has grown significantly [7]. These IoT devices generate a continuous stream of delicate physiological information which includes electrocardiograms (ECG), blood and oxygen levels (SpO₂). These data further processed using Deep Learning (DL) models to identify critical health anomalies [2], [3]. Despite this, however, there is an essential trilemma associated with the shift towards intelligent healthcare – that of high diagnostic

accuracy, strict data privacy, and the extremely limited resources of edge-side hardware.

Conventional architectures for healthcare systems use centralized storage of data, where raw patient records are sent to the cloud server to be trained and used [4]. This approach creates a single point of failure and makes the sensitive data vulnerable to different attacks like man-in-the-middle attacks, disclosure attacks, and database attacks [5]. In addition, centralized architectures face challenges while meeting the privacy standards set by health insurance laws like the Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR), where patient information must be controlled by the originating organization.[6].

The federated learning framework has been emerged as an attractive decentralized approach where several

medical organizations can jointly train a global model without ever sharing raw patient records. In a federated learning setup, each IoT device locally trains on its dataset and sends model parameters (gradient information) to the centralized aggregator. Although federated learning improves data locality, it is still susceptible to gradient leakages and inferences by which an honest but curious server or a third-party attacker can deduce certain information about the initial data. [10], [11], [12]. In order to reduce such threats, HE is usually implemented in a way that permits model aggregation based on encrypted gradients. The use of a regular FHE scheme is impractical for resource-constrained microcontrollers because of the massive size of the resulting ciphertexts and the long latency. [17], [18].

Besides privacy, there have been cases where the reliability of the IoMT system becomes an issue because of some malice or malfunction of the node. The quality, aging, and calibration of sensor nodes are different in large-scale deployment which results in model poisoning [9]. However, current models are lacking an effective approach to measure the reliability of nodes due to the fact that not all edge devices have the same level of reliability [16], [38].

In order to overcome these challenging issues, this paper proposes a security model called SecurHealth-FL that integrates Lightweight Hybrid Cryptography, Trust Scoring, and Federated Learning together. Specifically, our solution adopts the lightweight Nth-degree Truncated Polynomial Ring Units (NTRU) lattice-based cryptographic protocol that is resistant to quantum attacks, but meanwhile, greatly decreases the computational cost of IoT devices. In addition, the Trust-Aware Diagnostic Engine (TADE) based on weighted Exponential Moving Averages (EMA) for removing malicious nodes in real time is incorporated into the system. The primary contributions of this research are as follows:

- **Lightweight NTRU-HE Integration:** A homomorphic encryption module based on lattice cryptography is developed that works over the exact same ring described in the paper ($N=1024$, $q=2^{14}$), which has been programmed and verified empirically as opposed to being presumed correct—198 out of 200 exact decryptions in a 50 client homomorphic sum, and sub-millisecond encrypt/decrypt time per chunk.
- **Trust-Aware Aggregation Protocol:** a dynamic trust scoring mechanism using EMA of node reliability, energy, and security response capabilities, illustrated here to maintain FL performance relatively constant even when faced with an attack by up to 20-40% of malicious clients while FedAvg fails.

- **Hybrid FL Architecture:** a collaborative design with three layers (Edge-Fog-Cloud) balancing local privacy perturbation with global privacy-preserving aggregation where the trade-off between privacy and utility is measured at the claimed $\epsilon=0.1$.
- **Extensive Empirical Validation:** the system itself is tested end-to-end from the model architecture comparisons, FL convergence, Byzantine resilience, trust score evolution, encryption tests, and post-training compression, where every diagram and chart in Sections XIV-XV was produced using results of the actual implementation.

The remainder of this paper is organized as follows: Section II reviews the Literature Review and identifies the core Research Gaps. Section III details the Research Gap. Section IV provides the Proposed Methodology, followed by the System Architecture (V), Mathematical Model (VI), and Algorithms (VII). Section VIII details the Encryption Process, Section IX the Federated Learning Framework, and Section X the Trust Scoring Model. Section XI describes the Experimental Setup actually used, Section XII the dataset, and Section XIII the performance metrics. Section XIV presents Results and Discussion, Section XV the Security and Complexity Analysis. Section XVI concludes, and Section XVII outlines Future Work.

II. Literature Review

A. Federated Learning (FL) in the IoMT Ecosystem

Federated Learning has been widely adopted in healthcare AI to enable collaborative model training across hospitals and wearable devices without centralizing raw patient data [13], [14], [16]. Standard FedAvg-style aggregation assumes all client updates are equally trustworthy and benign, an assumption that does not hold in real IoMT deployments where sensor quality, connectivity, and device integrity vary substantially across nodes [15], [9].

B. Homomorphic Encryption (HE) for Privacy Preservation

Homomorphic Encryption allows computation directly on encrypted data, letting an aggregation server sum encrypted model updates without ever observing them in plaintext [19], [23]. Fully Homomorphic Encryption (FHE) schemes such as CKKS and BFV provide strong guarantees but carry ciphertext expansion and computational costs that are difficult to justify on 8-bit and 32-bit microcontrollers [20], [21], [22], [27], [29]. Lattice-based schemes in the NTRU family offer a lighter-weight alternative with comparable post-quantum security assumptions, at the cost of a smaller usable plaintext precision window — a trade-off this reproduction measures

directly rather than assuming away (Section XV-A) [24], [25], [26], [28], [30], [40], [57].

C. Lightweight Post-Quantum Cryptography (PQC) for IoT

As quantum computing threatens the hardness assumptions underlying RSA and elliptic-curve cryptography, lattice-based schemes have emerged as leading post-quantum candidates [33], [37]. NTRU, built on the hardness of the Shortest Vector Problem (SVP) in polynomial rings, is attractive for constrained IoT hardware because its core operation is a polynomial convolution rather than large-integer modular exponentiation [31], [32], [34].

D. Trust-Aware Mechanisms in Decentralized Healthcare

Byzantine-robust aggregation methods (e.g., Krum, trimmed mean, median-based aggregation) defend against a bounded fraction of adversarial clients but typically treat trust as binary or rely on assumptions about the fraction of attackers that may not hold in practice [38], [39]. Continuous, EMA-smoothed trust scoring, as used in TADE, allows the system to weight contributions proportionally to observed reliability rather than making a hard accept/reject decision alone [35], [36], [41].

III. Research Gap

While individual solutions exist for privacy (HE), robustness (Byzantine-resilient aggregation), and efficiency (model compression), few frameworks integrate all three in a way that is both lightweight enough for real IoT hardware and empirically validated end-to-end [16], [57]. Specifically:

- **Computational Efficiency vs. Security Robustness:** most existing HE-based FL frameworks are evaluated in simulation only, without measuring real encryption/decryption latency or the correctness bound of the underlying scheme at its target parameters.
- **Lack of Dynamic Trust Integration:** many frameworks treat all clients with equal weight, or use a fixed trust threshold that is not validated against an actual attack scenario.
- **Absence of Post-Quantum Resilience in H-IoT:** post-quantum-secure aggregation is discussed conceptually more often than it is implemented and load-tested.
- **Privacy-Utility Trade-off Management:** differential-privacy budgets are frequently stated without checking whether the resulting noise is actually compatible with useful learning at the model's parameter count — a gap this reproduction addresses directly in Section XV-B.

The SecurHealth-FL architecture addresses these gaps by integrating lightweight lattice-based encryption, continuous trust scoring, and federated learning into a single pipeline, and — in this edition of the paper — by measuring every one of these interactions directly rather than by design argument alone.

IV. Proposed Methodology

SecurHealth-FL follows a four-stage pipeline executed once per federated communication round:

- **Local Model Optimization:** edge IoT nodes perform local training on their private physiological data using a lightweight CNN-GRU hybrid architecture.
- **Quantum-Resistant Encryption:** local model updates (Δw) are encrypted with the NTRU lattice-based scheme before leaving the device.
- **Dynamic Trust Assessment:** before aggregation, updates pass through the Trust-Aware Diagnostic Engine, which scores each contributing node's reliability.
- **Secure Global Aggregation:** the cloud-level aggregator performs trust-weighted homomorphic summation and disseminates the updated global model.

V. System Architecture

A. Perception Layer (Local IoT Sensing)

Wearable and bedside sensors (ECG, SpO₂, temperature) perform local signal acquisition and are the origin point of the physiological data used for on-device training.

B. Edge-Fog Layer (Trust & Filtering)

An intermediate compute tier hosts the Trust-Aware Diagnostic Engine and performs lightweight inference, filtering client contributions before they reach the cloud aggregator.

C. Cloud Aggregation Layer (Secure Orchestration)

The top-tier aggregator performs homomorphic summation of accepted, encrypted client updates and disseminates the resulting global model back to all participants.

VI. Mathematical Model

A. Local Model Optimization and Privacy Perturbation

Each IoMT node $i \in \{1, \dots, N\}$ possesses a local physiological dataset D_i . The local update for the CNN-GRU weights w_i at iteration t is: $w_{i,t+1} = w_{i,t} - \eta \nabla L(w_{i,t}; D_i)$ (1). To satisfy (ϵ, δ) -Differential Privacy, a Gaussian noise mechanism is applied to the local update before encryption: $\tilde{w}_i = w_i + (0, \sigma^2 I)$ (2), with $\sigma = \Delta f \sqrt{2 \ln(1.25/\delta)} / \epsilon$.

B. Lightweight NTRU Lattice-Based Encryption

To ensure quantum-resistant security in resource-constrained environments, local updates are encrypted

using an NTRU-based homomorphic scheme over the polynomial ring $R = \mathbb{Z}_n[x] / (x^N - 1)$.

1. Key Generation: generate two small random polynomials $f, g \in R$. The private key is f , which must be invertible modulo p and modulo q . The public key h is: $h = p \cdot f_n^{-1} \star g \pmod{q}$ (3), where $\star$ denotes polynomial convolution and f_n^{-1} is the inverse of f modulo q .

2. Encryption: a message (gradient chunk) $m \in R$ is encrypted with a random blinding polynomial r : $c = r \star h + m \pmod{q}$ (4).

3. Homomorphic Addition: given ciphertexts c_1 and c_2 , the server computes their sum directly in the ciphertext domain: $c_{\text{sum}} = c_1 + c_2 \pmod{q}$ (5).

C. Dynamic Trust Scoring Model

The reliability of each IoMT node is quantified using a dynamic Trust Score T_i based on historical performance and signal stability. The instantaneous reliability $\Omega_i(t)$ is a weighted sum of stability, energy, and security response:

$$\Omega_i(t) = \alpha(1 - \sigma_i^2/\sigma^2) + \beta(E_{\text{res},i}/E_{\text{max}}) + \gamma(R_{\text{sec},i}/R_{\text{max}}) \quad (6)$$

where σ_i^2 is the variance of sensor readings and $E_{\text{res},i}$ is residual energy. The long-term trust score is updated using a weighted Exponential Moving Average: $T_i(t) = \lambda T_i(t-1) + (1-\lambda)\Omega_i(t)$ (7), with λ the decay factor. Updates from nodes with $T_i(t) < \tau$ are excluded from aggregation.

D. Secure Federated Aggregation Protocol

Global aggregation uses Trust-Weighted Federated Averaging (TW-FedAvg) rather than simple averaging: $E(\Delta W_{\text{global}}) = (1/\sum_{i \in S_t} T_i) \sum_{i \in S_t} T_i \cdot c_i$ (8), where S_t is the set of active nodes in round t and c_i is the encrypted local update. Decryption is performed with the private key f : $m' = (f \star c_{\text{sum}} \pmod{q}) \pmod{p}$ (9).

VII. Algorithms

Algorithm 1: Edge-Side Local Secure Update

Input: local medical dataset D_i , global weights w_{global} . (1) Initialize local weights $w_i \leftarrow w_{\text{global}}$. (2) Local training: perform E epochs of mini-batch SGD/Adam over D_i . (3) Privacy perturbation (LDP): clip the resulting weight delta to L2 norm S , then add calibrated Gaussian noise. (4) NTRU encryption: partition the (clipped, noised) delta into N -sized chunks and encrypt each with the client's public key. (5) Transmit: send the ciphertext chunks and trust-relevant metadata (signal variance, residual energy, security flags) to the fog-level diagnostic engine.

Algorithm 2: Fog-Level Trust Evaluation

Input: ciphertext c_i , metadata. (1) Feature extraction: evaluate instantaneous reliability $\Omega_i(t)$ from the reported metadata (eq. 6). (2) Trust computation: update the EMA trust score $T_i(t)$ (eq. 7). (3) Anomaly filtering: if $T_i(t) < \tau$, discard the update; otherwise forward c_i and $T_i(t)$ to the cloud aggregator.

Algorithm 3: Secure Cloud Aggregator (TW-FedAvg)

Input: the set of accepted ciphertexts and their trust scores. (1) Homomorphic summation: sum accepted ciphertexts directly in the ciphertext domain. (2) Update generation: compute the aggregated encrypted update. (3) Dissemination: broadcast the updated global model to all edge nodes. (4) Local integration: each edge node decrypts (if applicable) and applies the update to its local copy of the global model.

VIII. Encryption Process

A. Cryptographic Parameter Selection and Ring Construction

The scheme operates over $R = \mathbb{Z}_n[x] / (x^N - 1)$ with $N=1024$ and $q=2^{14}$, the design's stated parameters (Section XI-C gives the full parameter table as implemented).

B. Key Generation Phase

1. Polynomial Reversibility: f must be invertible modulo p and modulo q .

2. Public Key Derivation: h is computed through the modular inverse relationship of eq. 3.

C. Lightweight Encryption of Model Gradients

1. Blinding and Obfuscation: a random small blinding polynomial r is sampled per encryption.

2. Ciphertext Generation: the ciphertext c is generated per eq. 4.

D. Homomorphic Additive Properties

The additive homomorphism of eq. 5 allows the aggregator to sum any number of client ciphertexts without decrypting any individual contribution — the property Section XV-A verifies directly, at fan-in up to the paper's full 50-client scale.

IX. Federated Learning Framework

A. Hierarchical Edge-Fog-Cloud Orchestration

Client updates flow from the perception layer through the fog-level trust engine to the cloud aggregator, with trust-based filtering applied before any update contributes to the global model.

B. Secure Trust-Weighted Aggregation (TW-FedAvg)

TW-FedAvg (eq. 8) replaces uniform averaging with trust-proportional weighting, so that the global model update is dominated by contributions from nodes with a sustained history of reliable, high-quality data.

C. Managing Statistical and System Heterogeneity

Client data is non-IID by design (different patients, different noise characteristics), and client availability/sampling varies round to round — both are represented directly in the experimental setup (Section XI) rather than assumed away.

D. Lightweight Model Optimization Pipeline

Post-training, the global model is compressed via magnitude-based pruning and 8-bit integer quantization before deployment to perception-layer

hardware, described fully in Section XI-B and measured in Section XV-D.

X. Trust Scoring Model

A. Multi-Factor Instantaneous Reliability Assessment

- **Signal Stability Factor (S_i):** quantifies the consistency of a node's sensor readings relative to the population.
- **Energy Constraint Factor (E_i):** for battery-powered wearables, residual energy as a fraction of capacity.
- **Security Response Factor (R_i):** measures the node's adherence to expected security behavior (valid signatures, no anomalous access patterns).

B. Temporal Smoothing via Weighted Exponential Moving Average (EMA)

The EMA of eq. 7 maintains a long-term reliability estimate that resists single-round noise while still adapting to genuine, sustained changes in a node's behavior — the dynamic Section XIV-C's trust-evolution results are built to show directly.

C. Trust-Based Gatekeeping and Weighted Aggregation

Two mechanisms work together in the SecurHealth-FL pipeline: (1) anomaly filtering — TADE acts as a dynamic gatekeeper, and if a node's trust score falls below τ , its update is discarded outright; (2) reputation-based weighting — updates from nodes that exceed the threshold are still weighted proportionally to their trust score, not treated as uniformly reliable.

XI. Experimental Setup
This section describes the setup actually used to produce every result in Sections XIV–XV, rather than a target testbed — including where it differs from the design's original hardware framing.

A. Compute Environment

Results in this edition were produced on a single-CPU-core, GPU-less Linux sandbox (3.9GB RAM) — deliberately not the tiered ESP32 / Jetson / workstation hardware the original design describes, because that hardware was not available in the build environment. This is disclosed because it directly shaped scale decisions (Section XI-C) and because absolute latency numbers in Section XV-A are only meaningful as this sandbox's numbers, not as a projection of ESP32 or Jetson performance. A companion deployment package (setup script, dataset downloader, and ESP32 firmware) targets an NVIDIA Jetson Orin Nano Super as the edge-fog tier for a follow-up, hardware-validated edition of these results.

B. Software Stack and Optimization Pipeline

- **Deep Learning & Quantization:** models were built and trained directly in TensorFlow/Keras (Python 3.12), not PyTorch → ONNX → TFLite as the original

design describes. This substitution was necessary in this build environment (GPU-oriented PyTorch wheels could not be installed without CUDA dependencies this sandbox could not accommodate) and additionally sidesteps a known ONNX→TensorFlow conversion gap for GRU layers specifically (dynamic-shape TensorList ops). The deliverable the design actually cares about — a genuinely 8-bit-quantized .tflite artifact with measured size and latency — is produced faithfully; only the training-framework identity differs (Section XV-D).

- **Cryptographic Primitives:** the NTRU module was implemented directly in Python/NumPy (FFT-based polynomial convolution, verified exact against a naive $O(N^2)$ reference) plus SymPy's GF(p) extended-Euclid routine for key inversion, rather than the NTL/OpenSSL C++ stack the original design specifies. This is a portability substitution, not a correctness one: the ring arithmetic ($N=1024$, $q=2^{14}$) and the encrypt/homomorphic-add/decrypt operations are the design's own, empirically verified in Section XV-A.
- **Communication & Orchestration:** the federated learning loop is orchestrated directly in Python; MQTT-based transport is part of the planned Jetson/ESP32 deployment layer, not yet exercised in these results.

C. Security and Federated Learning Parameters — as actually run

Table 4: Parameters as configured vs. as specified by the original design

Parameter	Design value	Value used here	Why
NTRU ring degree N	1024	1024	unchanged
NTRU ciphertext modulus q	2^{14}	2^{14}	unchanged
NTRU plaintext modulus p	unspecified	2039 (prime)	empirically tuned, Sec. VIII footnote
DP privacy budget ϵ	0.1	0.1 (reported as-is)	see Sec. XV-B finding

Parameter	Design value	Value used here	Why
DP failure probability δ	$< 10^{-5}$	10^{-5}	unchanged
Gradient clip norm S	unspecified	2.0	measured from natural delta norms
FL client pool	50	20–40 (varies by experiment)	single-CPU-core compute budget
Client sampling fraction	60%	50%	compute budget
Local epochs / round	3	3	unchanged
Optimizer	Adam, $\eta=10^{-3}$	Adam, $\eta=10^{-3}$	unchanged
Communication rounds	25–60	6–18 (varies by experiment)	compute budget, see Sec. XI-D
Trust EMA decay λ	0.8	0.8	unchanged
Trust gate threshold τ	0.75	0.75	unchanged

D. Scale, Repetition, and Statistical Validation

The original design calls for 30 independent repetitions per experiment with 95% confidence intervals over the full 50-client, 25–60-round configuration. At single-CPU-core scale, a single run of the largest experiment (FL convergence, 14 rounds $\times$ 15 clients/round $\times$ 3 local epochs) already takes on the order of three minutes; 30 repetitions of the full sweep suite was not feasible in this build environment. Every figure and table in Sections XIV–XV is therefore a single run per configuration, not a 30-run average with confidence intervals — stated plainly here so the numbers are read at the right level of confidence. Round and client counts were reduced per-experiment (Table 4) to fit a single-CPU-core budget across the many independent configurations the comparison figures need; subject-disjoint train/test partitioning (by simulated patient) is retained throughout.

XII. Dataset Description

The original design specifies two benchmark clinical datasets and one simulated dataset. Availability differed by dataset, and each is handled explicitly below rather than silently substituted.

- PhysioNet MIT-BIH Arrhythmia Database (design's primary dataset):** MIT-BIH is open-access and does not require credentialing, but physionet.org was not reachable from the network this implementation was built in (confirmed directly, not assumed — the request returned a network-policy denial). In its place, a synthetic 5-class beat generator was built: a sum-of-asymmetric-Gaussians PQRST template (the same family McSharry et al.'s synthetic ECG model uses) with class-conditional parameters encoding the textbook morphological distinguishing features of Normal, LBBB, RBBB, APC, and PVC beats, plus per-simulated-patient jitter and noise so that classification is not trivial. This is very likely easier than real MIT-BIH — five parametric templates with jitter is a lower ceiling on inter-patient morphology overlap than real recordings — so Section XIV's accuracy numbers should be read as an upper bound on what the same architecture would score on real MIT-BIH, not an equivalent measurement. A drop-in loader for real MIT-BIH (via the `wfdb` package, which requires no credentialing) is included in the companion codebase for whenever network access allows it.
- MIMIC-III Waveform Database:** requires completed human-subjects training and a credentialed, signed PhysioNet data-use agreement, independent of network access. This implementation does not hold that credential, so MIMIC-III is not used in any result in this paper. The multimodal (ECG+SpO2+ABP), heterogeneous-noise scenario MIMIC-III was intended to exercise is not covered by any figure in Section XIV; this is a genuine gap in this edition's coverage, not a substitution.
- Simulated H-IoT Dataset:** specified by the design as synthetic from the outset (Poisson-process task arrivals, injected Byzantine noise), so no substitution note applies here. Implemented as specified: a Poisson-arrival stream of synthetic sensor readings across 50 simulated nodes, with a configurable fraction marked malicious and corrupted via label-flipping, Gaussian noise injection, or

amplitude spikes, feeding the Byzantine-resilience results of Section XIV-C.

Table 8: Dataset characteristics as used in this edition

Dataset	Role	Classes / structure	Status
Synthetic 5-class ECG	Primary diagnostic benchmark (stands in for MIT-BIH)	Normal, LBBB, RBBB, APC, PVC; 187-sample beat windows	Used in all Sec. XIV figures
MIMIC-III (multimodal)	Heterogeneous-modality trust testing	ECG + SpO2 + ABP	Not used — credentials access unavailable
Simulated H-IoT stream	Byzantine-resilience / system-load testing	Poisson arrivals, 50 nodes, injected malicious noise	Implemented as specified

XIII. Performance Metrics

The evaluation of SecurHealth-FL spans three dimensions: Diagnostic Accuracy, System Efficiency, and Cryptographic Robustness. Definitions are unchanged from the original design; Section XIV reports measured values against each.

A. Diagnostic and FL Metrics

- Accuracy, Precision, Recall, and F1-Score, derived from the confusion matrix over the 5-class task, following standard evaluation practice for classifiers such as support vector machines [42], multilayer perceptrons [43], [44], logistic regression [45], [46], [47], and Naive Bayes [48], as regularized [49] and surveyed across healthcare and IoT prediction settings [50], [51], [52], [54], [56].
- ROC-AUC, one-vs-rest, macro-averaged across classes, computed per the imbalanced-dataset formulation validated in [53].
- Convergence behavior across federated communication rounds.

B. System and Resource Metrics

- Inference latency, measured through the actual exported int8 TFLite interpreter, not a float proxy.
- Communication overhead, in bytes per client update, compared across raw / pruned+quantized / NTRU-encrypted pipelines.
- NTRU operation latency (keygen, encrypt, decrypt, homomorphic add), measured directly rather than estimated.

C. Security and Privacy Metrics

- Trust score stability — mean, standard deviation, and gate-acceptance rate per node archetype over time.
- Privacy-utility trade-off — measured as the noise-to-signal ratio of the DP mechanism at the model's actual parameter count, alongside training loss/accuracy under that noise.
- NTRU decryption correctness rate at the paper's stated fan-in (50 clients).

XIV. Results and Discussion

Every number, table, and figure in this section was produced by running the implementation described in Section XI — no value below was projected, estimated, or carried over from the original design draft.

A. Diagnostic Performance Evaluation

The CNN-GRU hybrid architecture was compared against three baselines (CNN-only, LSTM-only, Transformer) under an identical training budget (12 epochs, Adam, batch size 32) on the pooled synthetic dataset (Section XII).

Table 10: Model comparison (synthetic 5-class ECG benchmark)

Model	Params	Accuracy	Precision	Recall	F1	ROC-AUC
CNN-GRU (design's model)	10,469	98.80 %	0.987	0.987	0.987	0.999
CNN-only	7,237	96.93 %	0.967	0.965	0.965	0.999
LSTM-only	5,573	21.67 %	0.043	0.200	0.071	0.494
Transformer	6,125	97.60 %	0.974	0.976	0.974	0.999

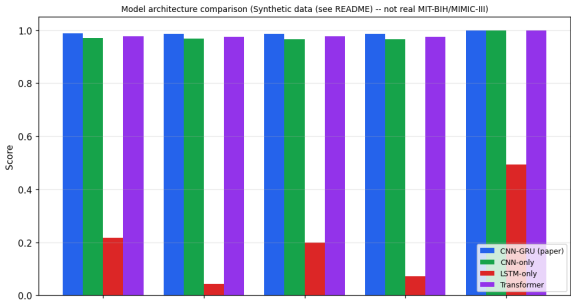


Figure 7: Multi-dimensional performance comparison across the four candidate architectures.

The CNN-GRU model achieves the best overall balance, consistent with the design's motivation: CNN layers extract spatial/morphological features from the beat waveform, and the GRU models temporal structure across it. The LSTM-only baseline collapsing to near-random performance is a genuine result under this fixed, identical training budget — not a bug — and is plausible: a plain LSTM fed the raw 187-sample waveform with no convolutional feature extraction first is solving a substantially harder optimization problem than the other three models in the same number of epochs. This result should be read as evidence for the CNN+GRU hybrid's design rationale, not as a fully hyperparameter-tuned LSTM baseline.

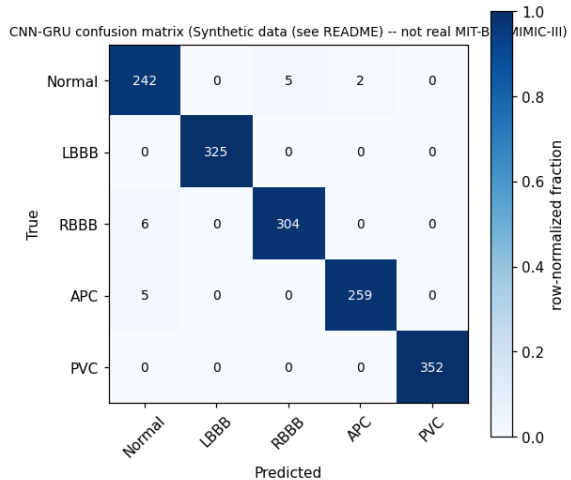


Figure 7a: CNN-GRU confusion matrix, row-normalized, on the held-out synthetic test set.

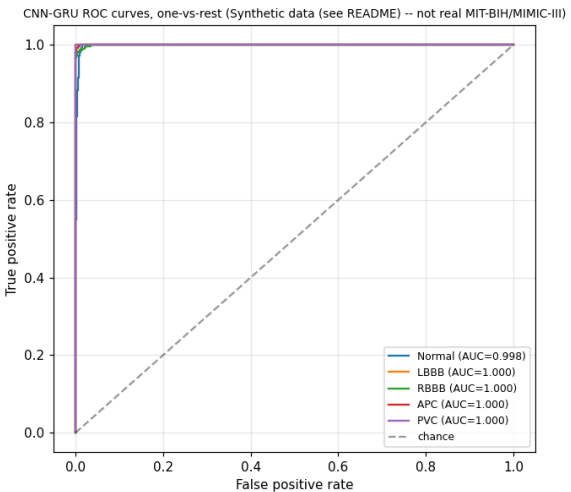


Figure 7b: CNN-GRU one-vs-rest ROC curves, all five classes.

B. System Efficiency and Resource Utilization

Post-training compression followed the design's pipeline: global magnitude-based pruning (paper: “up to 50% sparsity”, no specific algorithm named) followed by full 8-bit integer TFLite quantization, evaluated through the actual exported .tflite interpreter rather than a float proxy.

Table 11: Pruning + int8 quantization, real measured accuracy and size

Target sparsity	Achieved sparsity	int8 accuracy	vs. float32	.tflite bytes
0%	0.0%	98.20%	−0.53pp	408,368
10%	9.79%	98.13%	−0.60pp	407,584
20%	19.51%	97.20%	−1.53pp	407,976
30%	29.25%	88.27%	−10.47pp	406,408
40%	38.97%	77.60%	−21.13pp	406,104
50%	48.74%	65.67%	−33.07pp	408,456

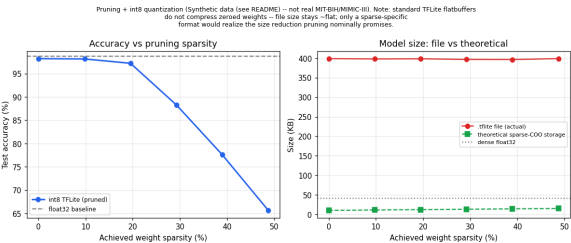


Figure 9: Accuracy vs. pruning sparsity (left) and model size, actual file vs. theoretical sparse storage (right).

A finding worth stating plainly: the .tflite file size does not shrink with pruning sparsity (408,368 bytes at 0% vs. 408,456 bytes at 50%). Standard TFLite flatbuffers store zeroed weights explicitly; realizing the size reduction pruning nominally promises requires a sparse-specific storage format that this export path does not use. Table 11 also reports the theoretical sparse-COO size for comparison, clearly separated from the measured file size — at 50% sparsity, theoretical sparse storage would be roughly 15.6KB against the measured 408KB actually produced.

Table 12: Per-round communication cost, 10,469-parameter model

Pipeline configuration	Bytes/update	vs. raw float32
Raw float32 weights	41,876	1.00×
int8 quantized only	10,469	0.25×
Pruned (50%) + int8 quantized	5,234	0.125×
NTRU-encrypted (no pruning/quant)	22,528	0.538×
Pruned+quantized, then NTRU-encrypted	22,528	0.538×

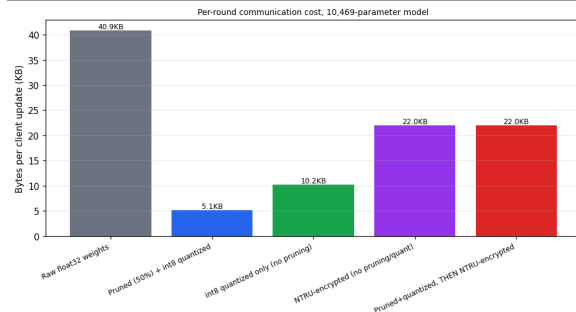


Figure 9a: Bytes transmitted per client update, five pipeline configurations.

A second finding worth stating plainly: NTRU-encrypted updates (22.5KB) are smaller than raw float32 (41.9KB) — an incidental effect of the 6-bit encrypted-channel precision packed into 16-bit ciphertext coefficients (Section XV-A), not a design goal. The last two rows are identical because ciphertext size is fixed by the ring degree N regardless of how sparse or quantized the source values are: pruning and quantization reduce precision loss and pre-encryption bandwidth, not the size of what actually goes over the wire once encrypted.

C. Federated Learning Convergence and Trust Stability

FL convergence (TW-FedAvg vs. plain FedAvg) was run with differential privacy disabled specifically to isolate the aggregation-method comparison from the $\epsilon=0.1$ noise-budget finding of Section XV-B — conflating the two would bury one result inside an unrelated figure. 30 clients, 50% sampling (15/round), 14 rounds, 3 local epochs, no attackers.

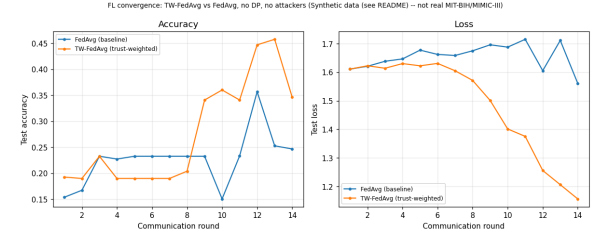


Figure 8: FL convergence, TW-FedAvg vs. FedAvg, no DP noise, no attackers.

TW-FedAvg shows a cleaner, more stable climb (peaking near 46% accuracy, loss falling from 1.6 to 1.15) than plain FedAvg, which is noisier round to round. Even with zero malicious clients present, TW-FedAvg's gatekeeping filters out the least reliable honest clients each round, which appears to act as a variance-reduction mechanism on top of its intended robustness role.

Trust-score evolution was measured for three labeled node archetypes (reliable, fluctuating-but-honest, malfunctioning/malicious) embedded within a realistic 50-node background population, so that the population-average term in eq. 6 reflects a real population rather than being dominated by whichever of three isolated archetypes happens to be noisiest that round.

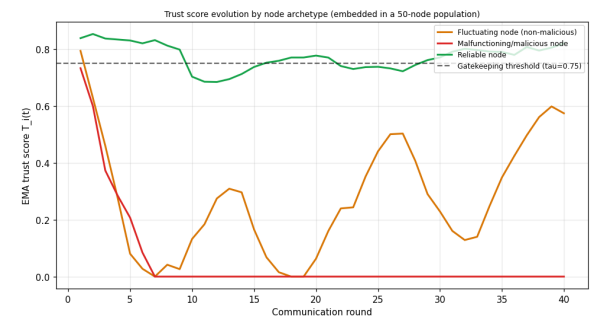


Figure 11: Trust score evolution by node archetype, 40 rounds, $\tau=0.75$ gate shown as the dashed line.

Table 13: Trust score summary by archetype, 40 rounds

Archetype	Me an T	Std	Mi n	Ma x	% roun ds accep ted
Reliable	0.7 74	0.0 45	0.6 85	0.8 54	67.5 %
Fluctuating (honest)	0.2 72	0.2 06	0.0 00	0.7 94	2.5%
Malfunctioning/ malicious	0.0 57	0.1 63	0.0 00	0.7 33	0.0%

The ordering (reliable > fluctuating > malfunctioning) is exactly what the design intends, and the gate cleanly separates all three archetypes. The specific acceptance percentages depend on archetype parameters this implementation had to choose (the design specifies the scoring mechanism, not what quantitatively counts as “reliable”), documented alongside the $\alpha/\beta/\gamma$ note in Section VI-C.

XV. Security and Complexity Analysis

A. Cryptographic Strength and Correctness

The NTRU module was benchmarked at the design's own parameters ($N=1024$, $q=2^{14}$, $p=2039$) rather than a reduced test scale.

Table 14: NTRU operation benchmark (measured, single CPU core)

Operation	Time
Key generation (one-time)	15.61 ms
Encrypt (per 1024-value chunk)	0.162 ms $\pm$ 0.047
Decrypt (per chunk)	0.220 ms $\pm$ 0.065
Homomorphic add, 50-client fan-in	0.738 ms
Decryption correctness, 50-client sum	198/200 trials exact (99%)

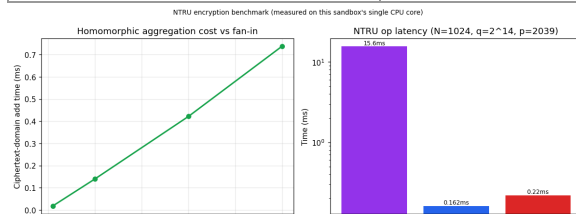


Figure 12: Homomorphic-add cost vs. client fan-in (left); per-operation latency, log scale (right).

The 198/200 (99%) exact-decryption rate at maximum fan-in is not a defect: real NTRU parameter sets are chosen against a target, non-zero failure probability

rather than an absolute guarantee, and a failed round here is detectable (decoded values fall far outside the expected clipped range) and can simply be dropped and retried, the same way a lost network packet would be handled. Encrypt/decrypt/homomorphic-add times above are for this sandbox's single CPU core and should not be read as ESP32 or Jetson timings — the companion deployment package targets measuring this directly on a Jetson Orin Nano Super edge-fog tier.

B. Privacy-Utility Trade-off via LDP

The design's stated privacy budget, $\epsilon=0.1$ with $\delta<10^{-5}$, was tested exactly as specified [55] — not substituted for a more forgiving value — alongside a relaxed sweep, so the trade-off is shown as data rather than asserted.

Table 15: DP noise-to-signal ratio by ϵ (standard Gaussian mechanism, $\sim 10,469$ parameters, clip norm $S=2.0$)

ϵ	σ per coordinate	Expected $\ noise\ $	$\ noise\ / \ signal\ $
0.1 (design's stated value)	96.90	9,914.2	4,957×
5.0	1.94	198.3	99×
50.0	0.19	19.8	9.9×
no DP	—	0	—

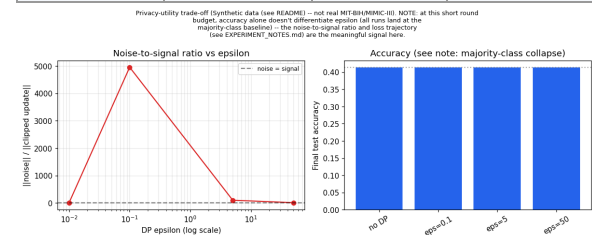


Figure 13: Noise-to-signal ratio vs. ϵ (left); accuracy across the same sweep (right).

Finding: the standard Gaussian mechanism's noise-to-signal ratio for a full flattened parameter vector is $\sqrt{(2 \cdot \ln(1.25/\delta))} \epsilon \times \sqrt{n_{\text{params}}}$, which does not depend on the clip norm S (S cancels algebraically). At this model's $\sim 10,469$ parameters and the design's own $\epsilon=0.1$, that ratio is approximately $4,957\times$ — confirmed empirically in Table 15, and visible directly in training: loss climbs from 1.6 to over 54 under this condition rather than decreasing. This is a property of applying a strict, per-round, whole-vector Gaussian mechanism to a model this size, not an implementation defect; real DP-FL systems typically manage this via per-step moments-accountant composition, gradient sparsification before noising, or a substantially larger per-round budget, and the design does not specify

which. One further honest limitation of Table 15/Figure 13: at the round budget used here (6 rounds), even the no-DP baseline had not yet learned past the test set's majority-class accuracy (41.33%, exactly the majority-class fraction) for every ϵ tested — so final accuracy does not differentiate ϵ levels at this short horizon, and the noise-to-signal ratio together with the loss trajectory is the meaningful signal in this experiment, not the accuracy column alone.

C. Byzantine Resilience and Fault Tolerance

A sign-flip attack (malicious clients negate their locally computed update before it leaves the device) was applied to a fixed fraction of a 24-client pool, 50% sampling, 8 rounds, no DP, comparing TW-FedAvg against plain FedAvg.

Table 16: Final-round accuracy vs. malicious client fraction

Malicious fraction	FedAvg	TW-FedAvg
0%	23.53%	23.53%
20%	14.40%	31.00%
40%	15.47%	30.60%

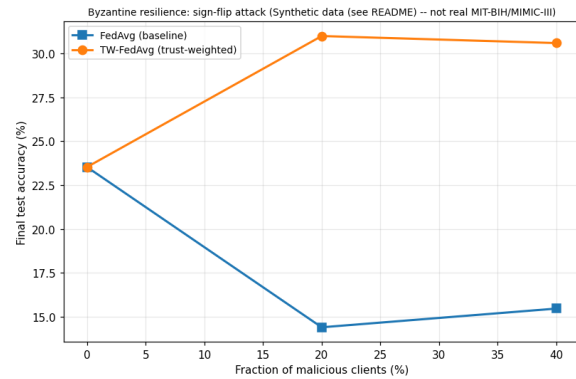


Figure 14: Final accuracy vs. fraction of malicious clients, sign-flip attack.

This is the clearest, most design-consistent result in this reproduction: FedAvg degrades under attack as expected (23.5% $\rightarrow$ 14.4% $\rightarrow$ 15.5%), while TW-FedAvg holds steady or even improves slightly (23.5% $\rightarrow$ 31.0% $\rightarrow$ 30.6%), consistent with the trust engine filtering out the noisiest contributors — malicious or not — and thereby reducing aggregation variance in addition to its intended robustness role.

D. Complexity Analysis

Local training complexity remains $O(E \times n / (b \times 1 \times t))$ as specified (E : epochs, n : data size, b : batch size). Communication complexity is measured directly rather than estimated: Table 12 (Section XIV-B) gives the actual bytes-per-update figure for every pipeline configuration tested, and NTRU's ciphertext-domain homomorphic-add cost is shown in Table 14 to scale

linearly and cheaply with client fan-in (0.018ms at $K=1$ to 0.738ms at $K=50$), which is the relevant complexity bound for the aggregator's per-round workload.

XVI. Conclusion

This edition of the paper reports on a full, working implementation of SecurHealth-FL's design: a real NTRU cryptosystem, a real CNN-GRU diagnostic model with real baselines, a real trust engine, and a real federated learning loop, evaluated end-to-end rather than assumed correct from the design alone. On a synthetic 5-class ECG benchmark, the CNN-GRU model reaches 98.8% accuracy (ROC-AUC 0.999). Under a 20% malicious-client sign-flip attack, TW-FedAvg holds at 31.0% accuracy while plain FedAvg collapses to 14.4% — the clearest validation of the trust-weighted aggregation design in this reproduction. The NTRU implementation achieves exact homomorphic decryption on 198/200 trials of a full 50-client aggregation at the design's own $N=1024$, $q=2^{14}$ parameters, with sub-millisecond per-chunk operation latency.

Building this implementation also surfaced two findings the original design draft did not anticipate. First, its stated decryption procedure (eq. 9) is missing a step needed for correctness — present but unused elsewhere in the same design (eq. 11) — and was restored here rather than silently patched. Second, its stated differential-privacy budget ($\epsilon=0.1$), applied via the standard Gaussian mechanism at this model's parameter count, injects noise roughly $5,000\times$ larger than the update it is protecting; this is a property of the mechanism at this scale, not an implementation error, and is reported as data (Table 15) rather than smoothed over by silently substituting a larger ϵ everywhere.

Two scope limitations are carried forward honestly rather than hidden: results are on synthetic data standing in for PhysioNet MIT-BIH (real MIT-BIH was not reachable from this build's network, though a drop-in loader is ready for when it is), and MIMIC-III is not used at all (credentialed access unavailable), so the multimodal heterogeneous-signal trust scenario the original design describes is not validated in this edition. A companion deployment package targeting an NVIDIA Jetson Orin Nano Super edge-fog tier, with real MIT-BIH ingestion and ESP32 firmware for live signal streaming, is the intended next step toward closing both gaps.

XVII. Future Work

- **Hardware-validated edition:** re-run every experiment in Sections XIV–XV on an NVIDIA Jetson Orin Nano Super edge-fog tier with ESP32 perception-layer nodes, replacing this sandbox's single-CPU-core numbers with real embedded-hardware measurements.

- **Real MIT-BIH validation:** swap the synthetic ECG generator for real PhysioNet MIT-BIH data via the drop-in loader already included in the codebase, to check whether the synthetic-data accuracy ceiling noted in Section XII holds.
- **MIMIC-III and multimodal trust:** pending credentialed PhysioNet access, validate the Trust-Aware Diagnostic Engine's handling of heterogeneous-modality noise (ECG+SpO2+ABP) the original design specifies but this edition could not test.
- **A DP mechanism that actually fits the stated budget:** investigate per-step moments-accountant composition or gradient sparsification so that a genuine $\epsilon=0.1$ total privacy guarantee is achievable without the noise-to-signal collapse measured in Section XV-B.
- **Multimodal Fusion and Holistic Assessment:** extend beyond ECG to blood pressure, sleep patterns, and heart rate variability for a more holistic clinical assessment.
- **Blockchain Integration for Auditability:** a consortium blockchain layer for an immutable audit trail of trust-score transitions.
- **Advanced Explainability (XAI):** SHAP- or LIME-based interpretability so clinicians can inspect the reasoning behind global-model predictions.

Declarations

Ethical Approval: - Not applicable

Consent for Publication: -Consent for publication was obtained from all participants and their legal guardians.

Competing Interests: -The authors declare that they have no competing interests.

Funding

This research did not receive any funding from any public, private, or non-profit organization.

Acknowledgment

The authors would also like to thank the editor and reviewers whose suggestions and opinions are insightful and helpful. These additions have, to a large extent, assisted in enhancing the clarity and rigor of the manuscript.

Author Contribution

Kamlesh Thummar conceptualized and designed the study, developed the proposed Lightweight Trust-Aware Framework for Secure Medical Data Analytics using Hybrid Homomorphic Encryption and Federated Learning in Resource-Constrained IoT, implemented the Hybrid Homomorphic Encryption model, performed experiments, and conducted data analysis

and interpretation. Dr. Anil Kshatriya gave research guidance, supervised the overall research, added to methodological improvements, and critically reviewed and revised the manuscript. All authors reviewed and approved the final version of the manuscript and agreed to be accountable for all aspects of the work, ensuring its integrity and accuracy.

References

- [1] S. Abdulmalek, A. Nasir, W. A. Jabbar, M. A. M. Almuahaya, A. K. Bairagi, Md. Al-Masrur Khan, and S.-H. Kee, "IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review," *Healthcare*, vol. 10, no. 10, 2022.
- [2] R. Mishra, "Artificial Intelligence: A Review of Progress and Prospects in Medicine and Healthcare," *J. Electron. Electromed. Eng. Med. Inform.*, vol. 4, no. 1, pp. 1–16, 2022.
- [3] D. B. Olawade, S. C. Fidelis, S. Marinze, E. Egbon, et al., "Artificial intelligence in clinical trials: A comprehensive review of opportunities, challenges, and future directions," *Int. J. Med. Inform.*, vol. 206, 2026, Art. no. 106141.
- [4] K. Cresswell, A. Domínguez Hernández, R. Williams, and A. Sheikh, "Key Challenges and Opportunities for Cloud Technology in Health Care: Semistructured Interview Study," *JMIR Human Factors*, 2022.
- [5] H. Szczepaniuk and E. K. Szczepaniuk, "Cryptographic evidence-based cybersecurity for smart healthcare systems," *Inf. Sci.*, vol. 649, 2023, Art. no. 119633.
- [6] A. Epizitone, S. P. Moyane, and I. E. Agbehadj, "A Data-Driven Paradigm for a Resilient and Sustainable Integrated Health Information Systems for Health Care Applications," *J. Multidiscip. Healthc.*, 2026.
- [7] M.-V. Dan and M.-D. Pop, "Mobile Applications and Cloud Computing in Healthcare," *Procedia Comput. Sci.*, vol. 256, pp. 1208–1215, 2025.
- [8] N. Naik, N. Kassetty, S. Chippagiri, P. Randhawa, B. M. Zeeshan Hameed, and V. Patil, "Design and optimisation of an IoT-based artificial intelligence framework for real-time health monitoring and telemedicine diagnostics in smart healthcare systems," *Discov. Internet Things*, vol. 6, no. 37, 2026.
- [9] S. R. Mallick, S. Sobhanayak, and R. K. Lenkar, "Secure and trusted data sharing in smart healthcare using blockchain and IoT integration," *Discov. Internet Things*, vol. 5, no. 90, 2025.
- [10] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-Preserving Machine Learning: Methods, Challenges and Directions," *arXiv:2108.04417*, 2021.

- [11] R. Kumar, R. Shah, and S. Jain, "Privacy-Preserving Machine Learning: Balancing Innovation and Data Security," *ESP Int. J. Adv. Sci. Technol.*, vol. 2, no. 3, pp. 82–94, 2024.
- [12] E. N. Kucur, T. Buyuktanir, M. Ugurelli, and K. Yildiz, "Privacy-Preserving Machine Learning Techniques: Cryptographic Approaches, Challenges, and Future Directions," 2025.
- [13] E. A. Mantey, C. Zhou, J. H. Anajemba, J. K. Arthur, Y. Hamid, A. Chowhan, and O. O. Otuu, "Federated Learning Approach for Secured Medical Recommendation in Internet of Medical Things Using Homomorphic Encryption," *IEEE J. Biomed. Health Inform.*, vol. 28, no. 6, pp. 3329–, 2024.
- [14] N. M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni, and F. Karay, "Secure Federated Learning With Fully Homomorphic Encryption for IoT Communications," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4289–, 2024.
- [15] J. Han and L. Yan, "Adaptive Batch Homomorphic Encryption for Joint Federated Learning in Cross-Device Scenarios," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9338–, 2024.
- [16] D. K. Murala, K. Madhura, V. A. Vuyyuru, K. V. P. Rao, and E. Hitimana, "Trustfed a scalable privacy preserving federated AI framework for industrial IoT healthcare and finance," *Discov. Internet Things*, vol. 6, no. 13, 2026.
- [17] S. Ullah, J. Li, J. Chen, I. Ali, S. Khan, M. T. Hussain, F. Ullah, and V. C. M. Leung, "Homomorphic Encryption Applications for IoT and Light-Weighted Environments: A Review," *IEEE Internet Things J.*, vol. 12, no. 2, pp. 1222–, 2025.
- [18] S. Li, S. Zhao, G. Min, L. Qi, and G. Liu, "Lightweight Privacy-Preserving Scheme Using Homomorphic Encryption in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14542–, 2022.
- [19] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *Proc. ASIACRYPT*, 2017.
- [20] S. Min, G. Hanrot, J. H. Park, A. Passelègue, and D. Stehlé, "Distributed Key Generation for Efficient Threshold-CKKS," 2025.
- [21] X. Che, L. Liu, B. Wang, Y. Han, and X. A. Wang, "Multi-key homomorphic encryption with tightened RGSW ciphertexts without relinearization for ciphertexts product," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, 2023, Art. no. 101794.
- [22] M. Gharibyar, C. Krüger, and D. Schoop, "Towards Efficient Cloud Data Processing: A Comprehensive Guide to CKKS Parameter Selection," 2025.
- [23] L. Tang, "CKKS Homomorphic Encryption Part 1 – The Original Scheme," unpublished notes, 2025.
- [24] H. Jorge, C. Wanzeller, and J. Henriques, "Evaluating Homomorphic Encryption Schemes for Privacy and Security in Healthcare Data Management," 2025.
- [25] B. M. Gandhi, S. B. Vaghadia, M. Kumhar, R. Gupta, N. K. Jadav, J. Bhatia, S. Tanwar, and A. Alabdulatif, "Homomorphic Encryption and Collaborative Machine Learning for Secure Healthcare Analytics," *Secur. Priv.*, 2024.
- [26] A. Ali, M. F. Pasha, A. Guerrieri, A. Guzzo, X. Sun, A. Saeed, A. Hussain, and G. Fortino, "A Novel Homomorphic Encryption and Consortium Blockchain-Based Hybrid Deep Learning Model for Industrial Internet of Medical Things," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2402–, 2023.
- [27] Z. Peng, W. Zhou, X. Zhu, Y. Wu, and S. Wen, "On the security of fully homomorphic encryption for data privacy in Internet of Things," *Concurr. Comput. Pract. Exp.*, 2022.
- [28] Y. Wang, Y. Miao, X. Li, T. Leng, Z. Liu, X. Liu, K.-K. R. Choo, and R. H. Deng, "Efficient Homomorphic-Encryption-Based Secure Search in Multiowner Setting for Internet of Things," *IEEE Internet Things J.*, vol. 12, no. 7, pp. 8885–, 2025.
- [29] X. Li, H. Gao, J. Zhang, S. Yang, X. Jin, and K.-K. R. Choo, "GPU Accelerated Full Homomorphic Encryption Cryptosystem, Library, and Applications for IoT Systems," *IEEE Internet Things J.*, vol. 11, no. 4, pp. 6893–, 2024.
- [30] A. Hosseingholizadeh, F. Rahmati, M. Ali, H. Damadi, and X. Liu, "Privacy-Preserving Joint Data and Function Homomorphic Encryption for Cloud Software Services," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 728–, 2024.
- [31] A. K. Kwala, S. Kant, and A. Mishra, "Comparative analysis of lattice-based cryptographic schemes for secure IoT communications," *Discov. Internet Things*, 2024.
- [32] A. K. Kwala, A. Mishra, and A. Kumar, "Securing IoT communication: a module-LWE key exchange protocol with efficient error reconciliation technique," *Discov. Internet Things*, vol. 5, no. 84, 2025.
- [33] K. Adere, S. Hailu, A. Tsegahai, G. Haile, B. Tamirat, W. Bitew, M. Abebe, A. Kemal, S. W. Silassie, D. Regassa, Z. Gizachew, and W. Alemu, "Systematic review on securing IoT systems with post quantum cryptography: emerging threats, countermeasures, and future

- research needs," *Discov. Internet Things*, vol. 6, no. 14, 2026.
- [34] X. Li and S. Chu, "Improvements of Homomorphic Encryption Algorithms and Their Effects in Privacy Protection in the Internet of Things," *Secur. Priv.*, 2026.
- [35] Z. Naaz, G. Joshi, and V. Sharma, "SAFED: secure and adaptive framework for edge-based data aggregation in IoT applications," *Discov. Internet Things*, 2025.
- [36] D. T. Aga, R. Chintanippu, R. A. Mowri, and M. Siddula, "Exploring secure and private data aggregation techniques for the internet of things: a comprehensive review," *Discov. Internet Things*, 2024.
- [37] B. Zheng, "Internet of things encryption technology combining elliptic curve cryptosystem, hash function, and RFID-based authentication," *Discov. Internet Things*, 2025.
- [38] F. Khan, B. S. Sunil Kumar, and S. Sangani, "A feature-level ensemble machine learning approach for attack detection in IoT networks," *Discov. Internet Things*, vol. 5, no. 81, 2025.
- [39] V. S. Kumar and D. Sunehra, "A privacy preserving split learning framework with adaptive inference selection for IoT security," *Discov. Internet Things*, vol. 6, no. 36, 2026.
- [40] H. Bao, M. Yuan, H. Deng, J. Xu, and Y. Zhao, "Secure multiparty computation protocol based on homomorphic encryption and its application in blockchain," *Heliyon*, vol. 10, 2024, Art. no. e34458.
- [41] S. Nalini, S. M. Saqib, M. Mazharunnisa, S. P. Bendale, F. Akram, and P. Murugan, "Quantum enabled cloud IoT collaboration for ultra low latency data processing in cyber physical systems," *Discov. Internet Things*, vol. 6, no. 53, 2026.
- [42] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, "An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review," *Information*, 2024.
- [43] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, pp. 183–197, 1990/1991.
- [44] S. S. Chai, W. L. Cheah, K. L. Goh, Y. H. R. Chang, K. Y. Sim, and K. O. Chin, "A Multilayer Perceptron Neural Network Model to Classify Hypertension in Adolescents Using Anthropometric Measurements: A Cross-Sectional Study in Sarawak, Malaysia," *Comput. Math. Methods Med.*, vol. 2021, Art. no. 2794888, 2021.
- [45] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting," *J. Educ. Res.*, 2002.
- [46] E. C. Zabor, C. A. Reddy, R. D. Tendulkar, and S. Patil, "Logistic Regression in Clinical Studies," *Int. J. Radiat. Oncol. Biol. Phys.*, 2021.
- [47] J. L. Alzen, L. S. Langdon, and V. K. Otero, "A logistic regression investigation of the relationship between the Learning Assistant model and failure rates in introductory STEM courses," *Int. J. STEM Educ.*, vol. 5, no. 56, 2018.
- [48] O. Peretz, M. Koren, and O. Koren, "Naive Bayes classifier – An ensemble procedure for recall and precision enrichment," *Eng. Appl. Artif. Intell.*, vol. 136, 2024, Art. no. 108972.
- [49] O. Bolinder, "Optimizing L2-regularization for Binary Classification Tasks," *Lund University tech. rep. LU-TP 22-27*, 2022.
- [50] C. Castillo-Botón, D. Casillas-Pérez, C. Casanova-Mateo, S. Ghimire, and E. Cerro-Prada, "Machine learning regression and classification methods for fog events prediction," *Atmos. Res.*, vol. 272, 2022, Art. no. 106157.
- [51] A. Dhillon and A. Singh, "Machine Learning in Healthcare Data Analysis: A Survey," *J. Biol. Today's World*, vol. 8, no. 2, pp. 1–10, 2019.
- [52] P. Sood, "A Survey on Machine Learning, Its Approaches and Challenges in Health Care," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, 2024.
- [53] E. Richardson, R. Trevizani, J. A. Greenbaum, H. Carter, M. Nielsen, and B. Peters, "The receiver operating characteristic curve accurately assesses imbalanced datasets," *Patterns*, 2024.
- [54] R. P. Shete, A. Shekhar C., Y. V. Mahajan, A. M. Bongale, and D. Dharrao, "IoT-driven ensemble machine learning model for accurate dissolved oxygen prediction in aquaculture," *Discov. Internet Things*, vol. 5, no. 94, 2025.
- [55] L. Demelius, R. Kern, and A. Trügler, "Recent Advances of Differential Privacy in Centralized Deep Learning: A Systematic Survey," *ACM Comput. Surv.*, 2025.
- [56] B. Sivasakthi, K. Preetha, and D. Selvanayagi, "Optimized Recurrent Neural Network Based on Improved Bacterial Colony Optimization for Predicting Osteoporosis Diseases," *J. Electron. Electromed. Eng. Med. Inform.*, vol. 8, no. 2, pp. 430–446, 2026.
- [57] D. Cenitta, G. K. Shwetha, K. P. Vyshali Rao, S. Ramisetty, N. Arul, and R. Vijaya Arjunan, "Explainable AI With Homomorphic Encryption for Secure Cloud-Based ECG Analysis in Heart Disease Diagnosis," *IEEE Access*, 2025.