Context-Aware Health Monitoring using Federated Learning

1st Abylay Salimzhanov*, 2nd Ayaulym Tynyssova*, 3rd Temirlan Tursynbekov*,
4th Saida Tulebayeva*, 5th Ayazhan Yelmagambetova*, 6th Adnan Yazici*, 7th Enver Ever*
*Computer Science, School of Engineering and Digital Science, Nazarbayev University, Astana, Kazakhstan

***Abstract*—Recent advances in the Internet of Things (IoT) and communication technologies have enabled the emergence of the Internet of Medical Things (IoMT), which integrates IoT into healthcare for real-time monitoring and intelligent intervention. IoMT applications are increasingly used for continuous health tracking, but traditional machine learning (ML) approaches—dependent on centralized data collection—raise critical concerns around patient privacy, data security, and regulatory compliance. Federated Learning (FL) addresses these issues by enabling collaborative model training across stakeholders (e.g., hospitals, homes) without sharing sensitive raw data, thereby preserving privacy and supporting distributed intelligence within IoMT ecosystems.**

This project presents a privacy-preserving health monitoring and emergency detection system that integrates IoT devices, FL, and Active Learning (AL). The system uses a two-stage architecture: physiological and contextual data (heart rate, SpO_2, ECG, age, gender) are collected via wearable sensors, while multimedia devices perform Human Activity Recognition (HAR) for anomaly validation. FL ensures secure model training across distributed sources. Experimental results show an anomaly detection accuracy of ˜0.89 using SensorECGCNN_Light and ˜0.85 HAR accuracy using a Temporal Convolutional Network (TCN). A Meta Federated Model (MFL) combines these outputs, while AL enables model refinement through human feedback, improving adaptability and system robustness over time.

***Index Terms*—Artificial intelligence (AI), Data interoperability, Federated learning (FL), Internet of Things (IoT), Internet of Medical Things (IoMT), Data science, Machine Learning (ML), Wearable sensors**

I. Introduction

The global increase in chronic illnesses , aging populations, and the demand for continuous health monitoring have driven the development of intelligent, real-time, and privacy-preserving healthcare systems [1], [2]. While these solutions offer computational power and accessibility, they suffer from significant drawbacks such as latency, scalability limitations, single-point failure risks, and — most critically — data privacy concerns [3].

In response to these limitations, the Internet of Medical Things (IoMT) has emerged as a promising framework that enables real-time data collection from wearable and ambient sensors [4], [5]. However, integrating these devices into a secure, intelligent, and responsive healthcare infrastructure remains challenging. Data heterogeneity, sensitivity of patient information, and the need for low-latency decision-making further complicate the adoption of centralized learning models. This necessitates a paradigm shift toward decentralized, edge-enabled, and privacy-conscious healthcare monitoring systems [6], [7].

A. Motivation and Significance

Privacy is paramount in medical applications, as health-related data is among the most sensitive categories of personal information. Regulations such as HIPAA and GDPR impose strict constraints on how such data can be handled, processed, and shared. Moreover, continuous monitoring of elderly and chronically ill individuals requires immediate response capabilities, which cannot be efficiently achieved through cloud-only architectures due to latency and connectivity issues [2], [8]. Therefore, it is essential to build a healthcare monitoring system that not only ensures privacy and security but also supports real-time anomaly detection, contextual awareness, and user personalization.

The motivation behind this project stems from the need to combine these qualities into a unified system capable of delivering proactive, accurate, and secure health monitoring in home and assisted-living environments.

B. Proposed Solution: Smart-Care

To address these challenges, we propose **Smart-Care**, an edge-intelligent federated health monitoring system that integrates scalar sensors (e.g., heart rate, temperature, ECG SpO_2), video-based context recognition, and wearable data streams. Smart-Care leverages:

- **Federated Learning (FL)** to collaboratively train anomaly detection models across multiple edge devices without transferring sensitive raw data [9], [10].
- **Edge computing** to perform real-time inference close to the data source, reducing latency and bandwidth usage [11], [12].
- **Multimodal data fusion** to integrate physiological, behavioral, and contextual information for accurate and personalized health anomaly detection [13].
- **Secure communication protocols** to ensure end-to-end data protection and compliance with privacy regulations [14], [15].

Additionally, Smart-Care employs a **meta-decision model** that combines results from federated anomaly detection, video-based Human Activity Recognition (HAR), and wearable sensor analysis to determine emergency alerts. A fail-safe mechanism is included where, in case of anomalies, the user is prompted

to confirm their status. If unresponsive, caregivers are notified through integrated communication services.

C. Organization of the Report

This report is organized as follows:

- **Section 2: Background and Related Work** provides a comprehensive survey of existing FL-based healthcare systems, edge computing strategies, multimodal fusion techniques, and security frameworks.
- **Section 3: Project Approach** describes the core components of Smart-Care, including sensor configuration, learning modules, communication flow, and emergency handling.
- **Section 4: Methodology** details the tools, technologies, datasets, and testing environments used to develop and validate the Smart-Care prototype.
- **Section 5: Evaluation** presents quantitative and qualitative results demonstrating the effectiveness and efficiency of the proposed system.
- **Section 6: Implemented Software** provides an overview of the developed software components, highlighting implementation details and system functionalities.
- **Section 7: Conclusion and Future Work** summarizes key findings and discusses potential directions for improvement and deployment.

II. Background and Related Work

The integration of FL, edge computing, multimodal sensor fusion, and secure IoMT has significantly advanced smart healthcare monitoring. This section reviews existing research extensively, analyzes various methodologies, compares different approaches, and provides a robust justification for the Smart-Care architecture, demonstrating a comprehensive understanding of relevant computing-based solutions.

A. Federated Learning Architectures and Privacy Preservation

FL allows distributed model training without compromising data privacy, crucial in healthcare applications. Liang *et al.* [7] introduced MD-FEEL, optimizing federated updates using an Age of Update (AoU) metric, effectively handling heterogeneous data sources. Jenifer *et al.* [16] validated the feasibility of lightweight federated models such as LASSO regression on edge devices, achieving 93.4% accuracy in heart disease prediction, underscoring FL's efficacy in resource-constrained scenarios.

Addressing privacy in heterogeneous environments, Hsu *et al.* [9] proposed HPFL, enabling selective data sharing per modality, thus enhancing privacy control. Anbarasan *et al.* [20] further secured FL with homomorphic encryption in convolutional neural networks (CNNs), highlighting robust privacy in medical diagnostics. Meanwhile, Vucovich et. al. *et al.* [21] introduced FedSAM, which addresses non-IID data challenges using sharpness-aware minimization, significantly improving the FL model performance.

Blockchain integration for enhanced security in FL was explored by Djenouri *et al.* [15], who created a blockchain-secured federated intrusion detection system, providing transparency and auditability. This approach aligns with Xu *et al.* [3] and Yang *et al.* [22], who extensively reviewed federated machine learning frameworks and their practical implementations in healthcare informatics.

Further advancements include dual-stage differential privacy by Lin *et al.* [14], which applies a two-stage mechanism to protect sensitive data during training of the cross-institutional model. Malekkhazadeh et. al. [18] demonstrated a federated approach for diabetic retinopathy detection using differentially private SGD, showing high accuracy while preserving privacy in retinal image analysis. The intracranial hemorrhage detection framework [23] applied DenseNet and GRU models within an FL paradigm to enable real-time, collaborative diagnosis of brain CT scans without data sharing.

Chen et. al. [10] introduced a federated transfer learning system tailored to wearable data, enabling personalized Parkinson's disease classification across heterogeneous client environments. FedHome [19] addressed in-home health monitoring by integrating cloud-edge architecture with a personalized FL scheme for elderly care using a generalized convolutional autoencoder (GCAE). ADDETECTOR [24] applied audio data collected from smart-home devices in a federated learning setting to detect early signs of Alzheimer's disease, demonstrating strong performance on noisy and privacy-sensitive speech inputs. VAFL [25] tackled variability in cross-source medical image data by incorporating CycleGAN-based latent space mapping into FL, improving generalizability across institutions without needing shared labels or formats.

These studies collectively demonstrate the versatility of FL across various healthcare domains, from medical imaging and chronic disease diagnosis to personalized, home-based patient monitoring.

B. Edge Intelligence and IoMT Integration

Edge computing has emerged as essential for real-time responsiveness and privacy enhancement in IoMT. Shi *et al.* [6] detailed foundational principles and benefits of edge computing, setting the stage for real-time, latency-sensitive healthcare applications. Doulani and Adhikari [17] developed FuXAI, an interpretable AI model using LightGBM and explainability methods (SHAP, LIME) deployed on edge devices, achieving real-time infectious disease detection.

Hayyolalam *et al.* [12] combined AdaBoost with Black Widow Optimization (BWO) for efficient edge-based diagnostics, achieving significant accuracy improvements in detecting heart anomalies. Sharma and Sharma [20] successfully deployed federated CNN models on edge devices for cardiovascular diagnostics, achieving 94.99% accuracy. Practical feasibility studies such as Paulraj *et al.* [11], who tested lightweight logistic regression models on Raspberry Pi devices, further highlight edge computing's viability in IoMT.

Paper	Key Topic	FL	AL	Sensors	HAR (video)	HAR (sensor)	Meta Classifier	Key Findings
[7]	Asynchronous Federated Learning on Edge Devices	✓	×	×	×	×	×	Introduced MD-FEEL for asynchronous FL in heterogeneous edge settings, improving efficiency and convergence.
[16]	FL for Heart Disease Prediction	✓	×	✓	×	×	×	Used LASSO-based logistic regression on edge devices for accurate heart disease detection.
[17]	Edge AI for Contagious Disease Monitoring	✓	×	✓	×	×	×	Deployed SHAP and LIME-enhanced LightGBM at the edge to provide interpretable predictions for disease detection.
[13]	Fall Detection with Hybrid Sensor Fusion and FL	✓	×	✓	×	✓	×	Achieved personalized fall detection via hybrid fusion of sensor data using federated learning.
[18]	Differentially Private FL for Retinal Image Analysis	✓	×	×	×	×	×	Proposed DPSGD-based federated framework with privacy guarantees for diabetic retinopathy detection.
[10]	Personalized Federated Learning for In-Home Healthcare Monitoring	✓	×	✓	×	✓	×	FedHome combines cloud and edge resources to achieve personalized healthcare monitoring through federated learning. Show significant personalization improvement.
[19]	Attention-based Anomaly Detection in FL	✓	×	×	×	✓	×	Used attention mechanisms to enhance robustness in federated anomaly detection with non-IID client data.
[**?**]	Federated Evidential Active Learning (FEAL) for domain shifts in medical imaging	✓	✓	×	×	×	×	Introduced FEAL using a Dirichlet-based evidential model to robustly handle uncertainties, significantly improving federated learning performance in heterogeneous medical scenarios.
Our Work	Health Monitoring and Emergency Detection System	✓	✓	✓	✓	✓	✓	A comprehensive work on deploying a health monitoring system with the emergency system using FL with a meta classifier through wearable sensors to track vital signs /ECG and video on HAR. Additionally, using AL to improve system performance.

Table I: Literature Review

Lin *et al.* [26] integrated YOLO-LW with SVM classifiers for efficient fall detection on neuromorphic edge hardware, enhancing real-time emergency response capabilities.

C. Multimodal Sensor Fusion and Context-Aware Health Monitoring

Sensor fusion techniques are critical in enhancing the accuracy and context-awareness of healthcare monitoring systems. Wu et.el. [19] categorized sensor fusion methods into early, late, and hybrid strategies, recommending hybrid fusion for balancing accuracy and responsiveness. Besrour *et al.* [13] introduced FedHSFD, which leverages multimodal data fusion and federated learning for personalized fall detection, significantly reducing false alarms.

Sharma and Rani [27] discussed quantified self-systems, emphasizing the value of multimodal data in personalized health monitoring. Sharon [28] provided critical ethical analysis of self-tracking technologies, highlighting transparency, autonomy, and user trust as essential components in healthcare systems.

D. Secure and Personalized E-Health Frameworks

Security and personalization remain pivotal challenges addressed by recent healthcare monitoring research. Lu *et al.* [4] explored blockchain integration in FL for secure and traceable data exchanges, significantly enhancing data integrity. Cvach [8] identified alarm fatigue in healthcare monitoring systems, recommending context-aware, intelligent alert systems for effective emergency management.

Yazici *et al.* [29] developed a comprehensive smart e-health framework utilizing multimodal sensors (ECG, inertial, and video) for accurate fall detection and emergency escalation. Similarly, the real-time monitoring system by Doulani and Adhikari [17] utilized lightweight, interpretable models deployed at the edge, demonstrating effective disease monitoring and caregiver engagement in clinical decision-making.

E. Comparative Analysis and Justification of Smart-Care

Traditional centralized models often face critical privacy, latency, and scalability limitations [30], which federated learning and edge computing effectively mitigate [2], [6]. Smart-Care synthesizes federated learning with edge intelligence, sensor fusion, and advanced security methodologies, addressing critical gaps noted in previous studies. By leveraging lightweight neural models deployed at the edge, Smart-Care ensures rapid anomaly detection and personalized health monitoring while safeguarding user data.

Unlike purely centralized or edge-only solutions, Smart-Care employs a hybrid federated-edge architecture, optimizing model

accuracy, reducing communication overhead, and ensuring privacy. By integrating blockchain-inspired security principles and differential privacy techniques [14], Smart-Care ensures robust data protection, compliance, and user trust.

Moreover, Smart-Care's multimodal approach, inspired by Sharma and Rani [27] significantly enhances anomaly detection reliability. It employs a meta-classification strategy that integrates physiological, environmental, and behavioral data streams, minimizing false alarms and optimizing caregiver responsiveness.

In conclusion, Smart-Care effectively integrates the most innovative aspects of federated learning, edge computing, multimodal fusion, and robust security frameworks derived from extensive prior research. This comprehensive synthesis positions Smart-Care at the forefront of next-generation smart health monitoring systems, directly addressing the limitations highlighted in existing literature and advancing personalized, real-time, and secure healthcare.

III. Project Approach

A. Design Philosophy and Goals

There are three main concepts that we focus on in the development of our system, namely patient data privacy, personalized real-time anomaly detection, and, most importantly, federated learning. To achieve this, we adopted a federated learning framework where sensitive physiological data is never sent to the central server. Instead, edge devices perform local computations and communicate only model weights which are anonymized. This secure design also supports personalization, as model adapts to each individual patient patterns. Also, our architecture had to be adaptable in order to support smartwatches, ECG monitors and camera systems. These goals shaped every aspect of our software and hardware integration and allowed us to support effective caregiver interaction without sacrificing latency and security.

B. Layered System Architecture

The system is structure into five interconnected layers to manage the flow of data and responsibilities efficiently: Sensor Layer, Edge Device Layer, Communication Layer, Cloud Aggregation Layer, and Visualization Layer. These are illustrated in Figure 1, which describes the physical and logical interactions between layers.

At the base, the Sensor Layer includes devices such as ECG monitors, smartwatches and cameras which collect real-time physiological and behavioral signals. Then, the collected signals are passed to the Edge Device Layer, typically implemented on devices like Raspberry Pi or smartphones, where they undergo pre-processing and anomaly detection using lightweight models. If and anomaly is detected, it is forwarded through the Communication Layer, which securely transmits alerts and model updates to the Cloud Layer. At this point, a key scenario unfolds: if an anomaly is detected and metamodel, which is described in more detail in the Methodology section, is not available, a notification is sent to the user, who confirms if the alert is accurate. If the user selects "OK", this feedback is added

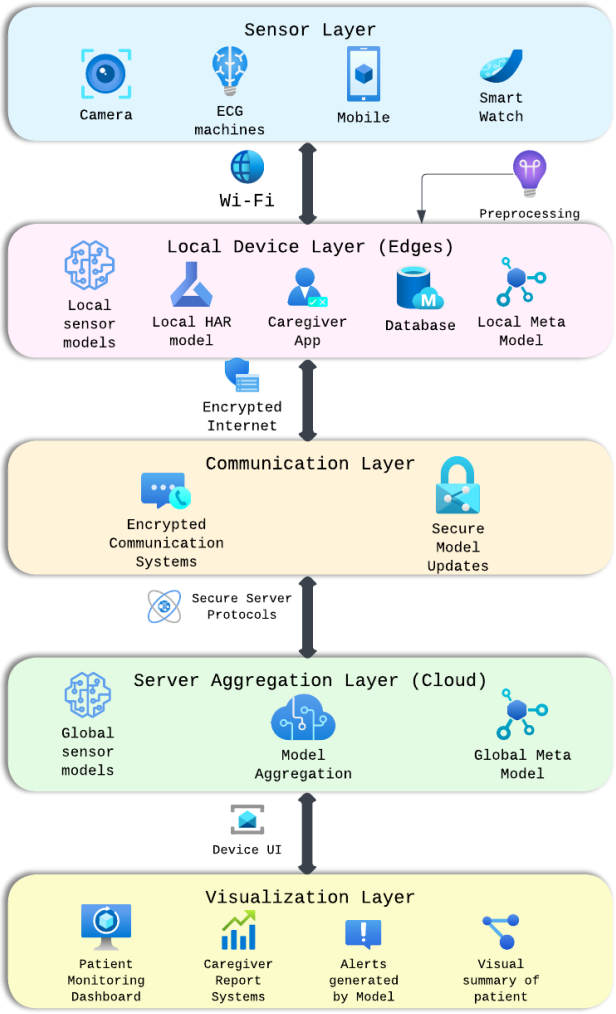


Figure 1: Layer Structure

to a report log for future local model retraining. However, if the user is indeed "Not OK", an alert is escalated to a caregiver, who receives both the prediction and supporting video footage with predicted activity. This response is then also added to the feedback pool. The Visualization Layer supports the final step, allowing caregivers to verify or correct predictions and close the feedback loop. This architectural layering ensures modular development, supports privacy at every stage, and enables supervised real-time decision-making.

C. Federated Learning Architecture

To enable the improvement of the collaborative model while making the system secure, we adopted a federated learning approach, for which the architectural diagram is illustrated in Figure 2.

In this framework, each edge device independently trains a local model using collected sensor data and caregiver feedback. Once enough feedback is accumulated, typically after a defined batch of confirmed or corrected predictions, the model updates

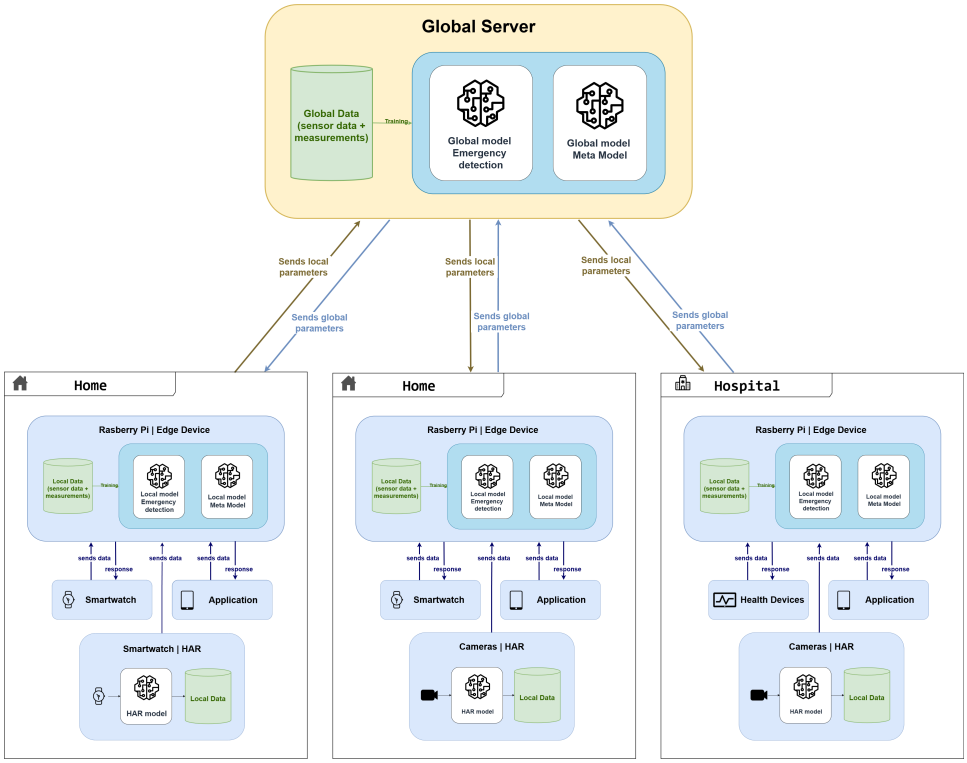


Figure 2: FL Architecture. End-to-end federated-learning loop for anomaly detection. Each edge pair trains a local 1-D CNN, uploads only weight deltas, receives the averaged global weights, and repeats. Caregiver feedback is folded into the next mini-batch before weights are sent.

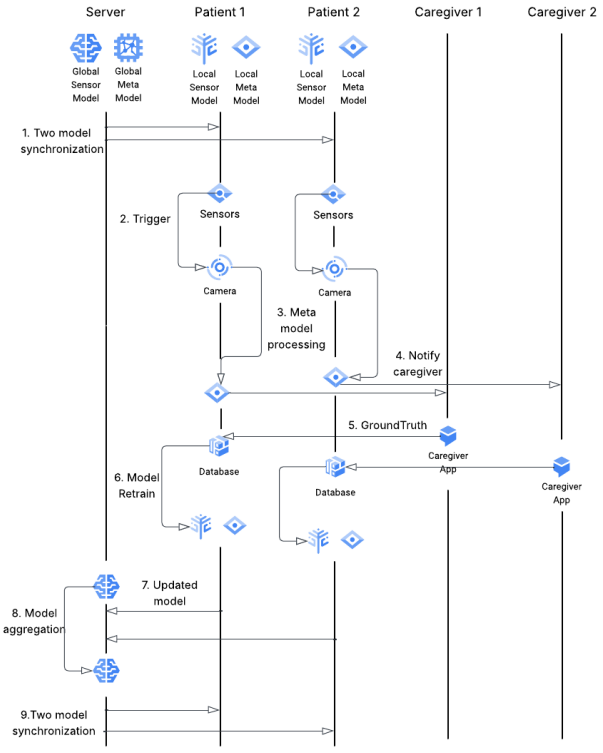


Figure 3: Sequence Diagram

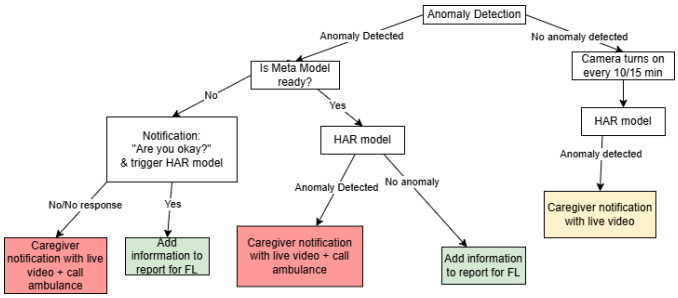


Figure 4: State Diagram. System operational state diagram illustrating the decision-making process from anomaly detection to caregiver notification and data logging for federated learning.

are sent to the central server. Only updated weights are sent, which is significant for privacy. The server aggregates updates from all participating clients using federated averaging techniques to form a new updated global model. This model is then redistributed to each client, allowing all local models to benefit from system-wide learning while remaining adapted to their individual patient environments. This cyclical process increases generalizability, reduces false positives, and respects privacy constraints, which are especially critical in medical settings.

D. System Workflow

The full operational flow of the system can be seen in Figure 3 and Figure 4. The process starts when the global model is sent

to all registered edge devices. These devices begin collecting sensor and video data from patients. When an anomaly is detected locally, a notification is generated and immediately sent to the patient for self-assessment. The user is prompted with two options: "OK" or "Not OK." This scenario is similar to the previously mentioned in Layer Diagram subsection. The feedback is logged and used in future training rounds.

Once a threshold of validated entries is reached, the local device initiates a federated update, sending model weights to the central server. After aggregation, a refined global model is broadcast to all clients. This cycle repeats continuously, enabling both real-time responsiveness and system-wide improvement through collective learning.

IV. Methodology

This section details the comprehensive methodology employed in this research, encompassing the development and evaluation of the sensor model for ECG anomaly classification, the contextual Human Activity Recognition (HAR) information, the meta-learning framework for anomaly verification, and the simulated federated learning environment designed to assess performance under heterogeneous conditions.

A. Sensor Model for ECG Anomaly Classification

The primary role of the sensor model is to perform real-time binary classification on incoming 187-feature ECG segments, denoted as $\mathbf{x}_{ecg} \in \mathbb{R}^{187}$, categorizing them as either 'Normal' ($y_{ecg} = 0$) or 'Abnormal' ($y_{ecg} = 1$).

1) Architecture: `SensorECGCNN_Light`*:* A lightweight 1D Convolutional Neural Network (CNN), designated `SensorECGCNN_Light`, was designed and implemented for this task. This architecture was selected to balance effective feature extraction from sequential ECG data with computational efficiency suitable for potential deployment on resource-constrained edge devices. The architecture, illustrated in Figure 5, processes the input ECG segment through a series of convolutional and pooling layers, followed by fully connected layers for classification. A key feature of this model is its ability to optionally output an intermediate 50-dimensional embedding, $\mathbf{e}_{cnn} \in \mathbb{R}^{50}$, which serves as a crucial input for the downstream meta-model.

The `SensorECGCNN_Light` model comprises three convolutional blocks. Each block consists of a 1D convolutional layer (`Conv1D`), a Rectified Linear Unit (`ReLU`) activation function, and a 1D Max Pooling layer (`MaxPool1D`). The filter counts for the convolutional layers are 16, 32, and 64, respectively, with corresponding kernel sizes of 7, 5, and 3. Appropriate padding is used. After these convolutional stages, features are flattened and passed through a dense layer (FC1) with ReLU activation to produce the 50-dimensional embedding. For classification, this embedding is processed by a dropout layer (p=0.5) and a final dense layer (FC2) with 2 output units, yielding logits for the 'Normal' and 'Abnormal' classes.

B. Human Activity Recognition (HAR) for Contextual Enrichment

Recognizing that the interpretation of physiological signals like ECG is often context-dependent, a Human Activity Recognition (HAR) component was developed to provide crucial situational awareness. For instance, an elevated heart rate is expected during physical exertion ('running') but could be indicative of an anomaly if observed during a sedentary state ('laying'). The HAR system aims to classify the subject's activity, and this information is conceptually vital for the meta-model to accurately verify sensor-detected ECG anomalies.

1) HAR Data Acquisition and Preprocessing Pipeline: The HAR model was developed using video data as input. The pipeline involves several stages:

1) **Video Input:** The system processes video sequences (e.g., .mp4, .avi) capturing subject movement.
2) **Frame Sampling:** Frames are sampled from the video at a specific rate to form input sequences for the pose estimator.
3) **Pose Estimation:** A pose estimation model is applied to each sampled frame to extract 2D skeletal keypoints. The system architecture supports multiple pose estimation backends (e.g., YOLOv8-pose, RTMPose, HRNet). For development and evaluation, YOLOv8-pose (specifically, lightweight variants like 'yolov8n-pose.pt' or 'yolov8s-pose.pt') was predominantly used due to its favorable balance of accuracy and computational speed suitable for near real-time applications. This backend typically extracts $V = 17$ keypoints per detected person, each comprising (x, y) coordinates and a confidence score.
4) **Feature Extraction:** From the raw keypoint coordinates, a feature vector is constructed for each frame. This typically involves concatenating the x and y coordinates of all detected keypoints (e.g., $17 \times 2 = 34$ features per frame if confidence scores are not directly included in this stage). Normalization techniques (e.g., relative to a root joint like the pelvis, or scaling based on bounding box dimensions) are often applied to achieve pose invariance to camera distance and subject size, although specific normalization details are implementation-dependent within the HAR module.
5) **Sequence Buffering and Formatting:** The frame-wise feature vectors are aggregated into fixed-length sequences. A sequence length of $T = 30$ frames was configured for the HAR models, resulting in an input tensor of shape (T, D_{feat}) for the HAR classifier, where D_{feat} is the dimensionality of the per-frame feature vector (e.g., 34). These processed sequences are then stored (e.g., as .npy files) for training and evaluation.

2) HAR Model Architecture: Temporal Convolutional Network (TCN): Several deep learning architectures suitable for sequential data classification were evaluated for the HAR task, including LSTMs with Attention, Transformers, and Temporal Convolutional Networks (TCNs). Based on comprehensive evaluations (yielding an accuracy of 86.1% on a held-out

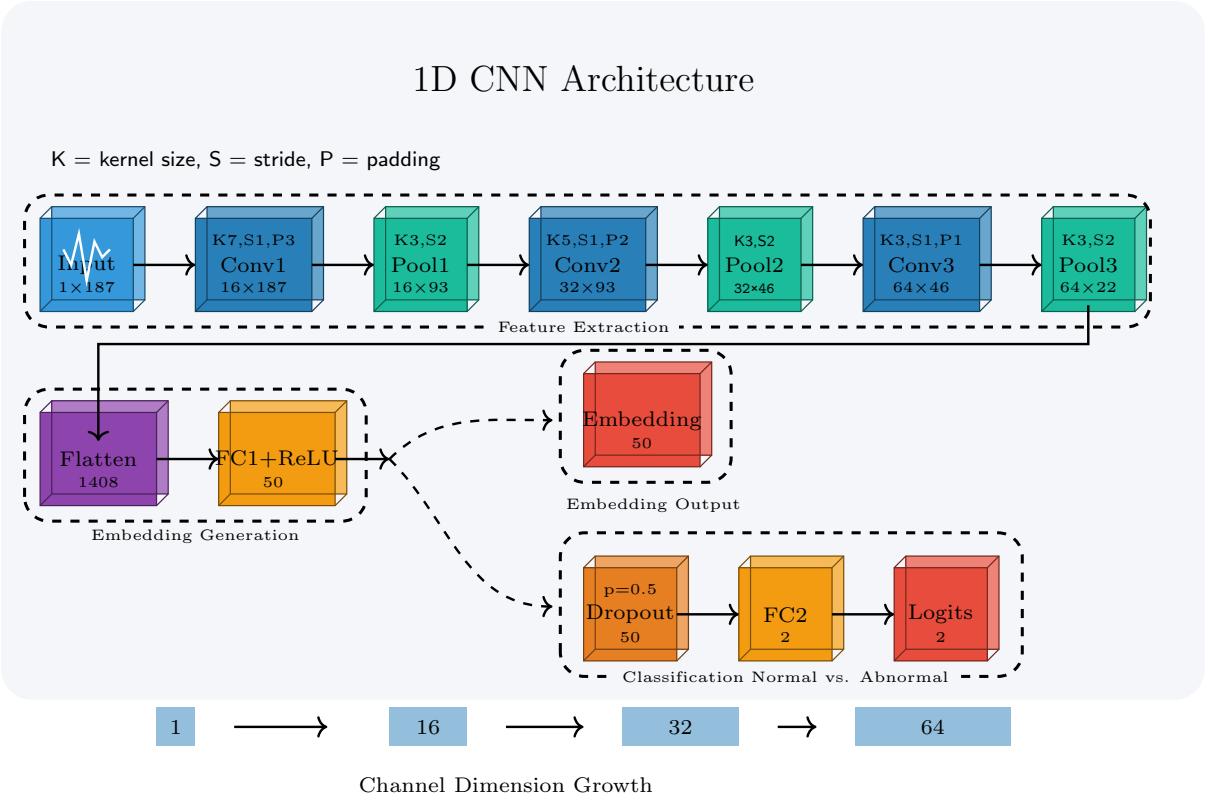


Figure 5: Architecture of the `SensorECGCNN_Light` model. It processes a 187-feature ECG segment, extracting features through three convolutional blocks. An intermediate 50-dimensional embedding is generated after the first fully connected layer (FC1) and ReLU activation. For direct classification, the flow continues through a dropout layer and a final classification layer (FC2).

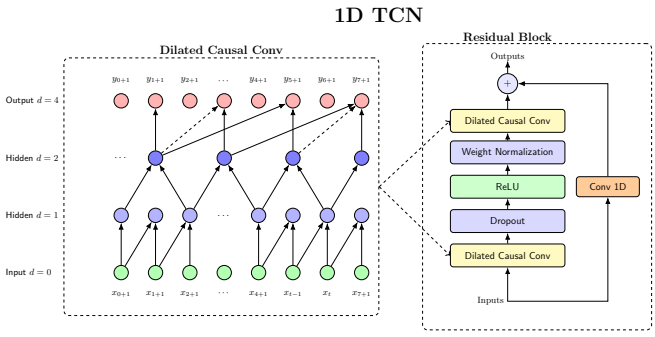


Figure 6: Conceptual data preprocessing pipeline for the HAR component: from raw video input to structured feature sequences ready for model training/inference.

validation set, as detailed in project worklogs), the TCN architecture was selected as the primary model for HAR.

TCNs are particularly well-suited for activity recognition from pose sequences due to their ability to capture long-range temporal dependencies efficiently using stacked layers of dilated causal convolutions. Causal convolutions ensure that the prediction for a given timestep only depends on past and current inputs, making them suitable for online processing. Dilated convolutions allow the receptive field of the network to grow exponentially with depth without a corresponding increase in parameters or computational cost, enabling the model to learn patterns over extended time windows. The TCN architecture typically incorporates residual blocks, each containing dilated causal convolutions, weight normalization, ReLU activation, and dropout, as illustrated in Figure 6. This structure helps in training deeper networks and improving gradient flow. The output of the TCN is a probability distribution over the pre-defined activity classes.

3) Simulated HAR Context for Meta-Model Integration: As described in Section 3, for the Federated Learning simulations focused on ECG anomaly verification, the direct output of this fully developed HAR component was not used in every iteration due to computational constraints of the simulation setup. Instead, the HAR context ($\mathbf{p}_{har} \in [0,1]^7$) provided to the meta-model was programmatically simulated. This simulation was, however, informed by the understanding gained from the HAR component's development:

- The seven activity classes match those targeted by the TCN-based HAR model.
- The conditional base probability distributions ($\mathbf{b}_{\text{normal}}$ and $\mathbf{b}_{\text{abnormal}}$) used in the simulation reflect plausible activity states during normal versus abnormal cardiac events (e.g., higher baseline probabilities for 'fall' or 'seizure' when $y_{ecg} = 1$).

- The introduction of jitter and dropout in the simulated $\mathbf{p}_{har}$ models the types of noise and unreliability that a real-world HAR system might exhibit.

This approach allowed for controlled study of the meta-model's sensitivity to varying quality and relevance of HAR contextual information, while leveraging the insights from the separately developed HAR system. In a fully deployed end-to-end system, the output of the TCN-based HAR component (potentially running on a dedicated HAR server or an edge device with sufficient capacity) would directly provide the $\mathbf{p}_{har}$ vector to the meta-model, as conceptually illustrated in Figure 7.

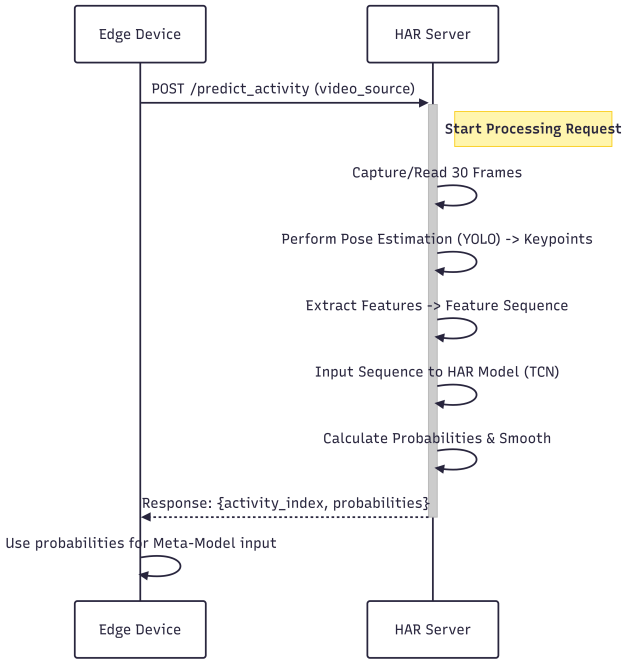


Figure 7: Conceptual interaction for a fully integrated system: The Edge Device, upon a sensor event, could request HAR context from a dedicated HAR Server, which processes video to provide activity probabilities for meta-model input. (Note: HAR context was simulated in the reported FL experiments).

C. Meta-Model for Anomaly Verification

A meta-learning layer, implemented as a `MetaClassifier`, was introduced to improve the overall system reliability by verifying anomalies flagged by the `SensorECGCNN_Light` model.

1) Task Definition: Anomaly Verification: The meta-model is invoked only when the sensor model predicts an ECG segment as 'Abnormal' (denoted $\hat{y}_{sensor} = 1$). Its objective is to predict the correctness of this specific sensor alert. The meta-model's target label, $y_{meta} \in \{0, 1\}$, is defined as:

$$y_{meta} = \begin{cases} 1 & \text{if } \hat{y}_{sensor} = 1 \wedge y_{ecg} = 1 \\ & \text{(Sensor True Positive)} \\ 0 & \text{if } \hat{y}_{sensor} = 1 \wedge y_{ecg} = 0 \\ & \text{(Sensor False Positive)} \end{cases} \quad (1)$$

This targeted verification aims to reduce false alarms from the primary sensor.

2) Architecture: `MetaClassifier` MLP: The `MetaClassifier` is a Multi-Layer Perceptron (MLP), chosen for its ability to effectively fuse the heterogeneous input features. The architecture is depicted in Figure 8. It takes a 58-dimensional input vector $\mathbf{z}_{meta} \in \mathbb{R}^{58}$. This input is processed through two hidden layers with ReLU activations and dropout (p=0.3). The first hidden layer comprises 32 units, and the second hidden layer comprises 16 units. The output layer is a fully connected layer producing 2 logits for the binary verification task.

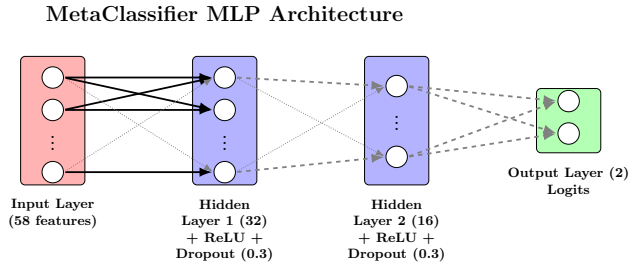


Figure 8: Architecture of the `MetaClassifier` MLP. It takes a 58-dimensional feature vector and processes it through two hidden layers (32 units and 16 units, respectively, with ReLU and Dropout) before producing a 2-class output for the anomaly verification task.

3) Meta-Model Input Features: The 58-dimensional input vector $\mathbf{z}_{meta}$ for the `MetaClassifier` is constructed by concatenating three distinct feature sets, as illustrated in Figure 9. This process is triggered only when $\hat{y}_{sensor} = 1$.

1) **Sensor CNN Embedding ($\mathbf{e}_{cnn} \in \mathbb{R}^{50}$):** The 50-dimensional embedding vector extracted from the penultimate layer (FC1 + ReLU) of the client's current `SensorECGCNN_Light` model corresponding to the alarmed ECG segment.
2) **Sensor Output Entropy ($H \in \mathbb{R}$):** Calculated from the sensor model's softmax output probabilities $\mathbf{p}_{sensor} = [p_0, p_1]$ for the two classes as $H = -\sum_{i=0}^{1} p_i \log_2(p_i)$. This quantifies the sensor's predictive uncertainty.
3) **Simulated HAR Context ($\mathbf{p}_{har} \in [0, 1]^7$):** A 7-dimensional probability vector representing simulated Human Activity Recognition context over 7 activities. This is generated via conditional structured mocking. Based on the true ECG label y_{ecg} of the sample that triggered the sensor, a base probability distribution over activities ($\mathbf{b}_{\text{normal}}$ or $\mathbf{b}_{\text{abnormal}}$) is selected. A sample $\mathbf{p}_{har}$ is then drawn from a Dirichlet distribution, $\mathbf{p}_{har} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where the concentration parameters $\boldsymbol{\alpha}$ are biased towards a dominant activity sampled from the selected base distribution. Client-specific noise (jitter/dropout) is subsequently applied to this vector, as detailed in Section **??**.

Thus, the meta-input is formed as $\mathbf{z}_{meta} = [\mathbf{e}_{cnn}; H; \mathbf{p}_{har}]$.

D. Simulation Design

1) Preliminary Sensor Model Comparative Analysis: To inform the choice of the primary sensor model, an initial

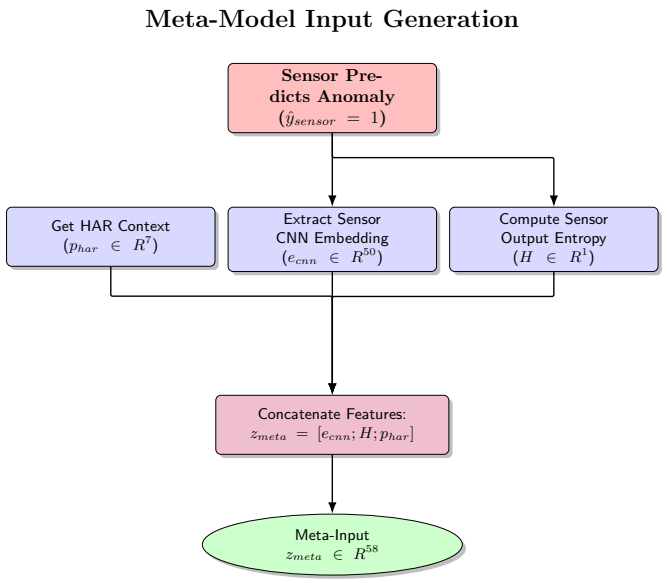


Figure 9: Meta-model input generation pipeline, triggered when the primary sensor predicts an anomaly ($\hat{y}_{sensor} = 1$). The sensor's CNN embedding ($\mathbf{e}_{cnn}$), output entropy (H), and simulated HAR context ($\mathbf{p}_{har}$) are extracted/generated and concatenated to form the 58-dimensional input vector $\mathbf{z}_{meta}$.

comparative study was conducted. This involved simulating five clients in FL mode, each processing 10,000 ECG samples. The client archetypes and their specific noise parameters (f_{n} for ECG Noise Factor, σ_{j} for HAR Jitter Sigma, p_{d} for HAR Dropout Probability, and p_{l} for Meta Label Noise Probability) used in this preliminary phase are detailed in Table II. Architectures evaluated included MLP with PCA, `SensorECGCNN_Light`, GRU, BiLSTM, and a ResNet1D variant. This study, whose results are presented in Section V-B1, guided the selection of `SensorECGCNN_Light` for subsequent, more extensive heterogeneous simulations due to its favorable balance of accuracy and computational cost.

Table II: Client Archetype Noise Parameters for Preliminary Sensor Model Exploration ($N = 5$, 10k rows/client).

User	Archetype	ECG Noise (f_{n})	HAR Jitter (σ_{j})	HAR Dropout (p_{d})	Meta Label Noise (p_{l})
1	Hospital 1	0.01	0.01	0.00	0.00
2	Wearable 1	0.35	0.25	0.15	0.05
3	Clinic	0.20	0.15	0.50	0.05
4	Wearable 2	0.50	0.30	0.20	0.15
5	Hospital 2	0.02	0.02	0.00	0.01

2) Main Federated Learning Simulation Framework: The primary simulation phase focused on an in-depth evaluation of the `SensorECGCNN_Light` and `MetaClassifier` within the FL paradigm.

a) Dataset Partitioning and Server Initialization:: The combined MIT-BIH Arrhythmia [**?**] and PTB Diagnostic ECG [31] datasets were preprocessed. For this main simulation, 1% of the total available training data was utilized to pre-train the global sensor model (`SensorECGCNN_Light`) on the server for 15 epochs. A distinct 20% of the data was held out as a global server validation set. The remaining data was then partitioned among $N = 5$ simulated edge clients, with each client receiving approximately 20,000 ECG samples (representing 15.8% of this remainder per client). The global meta-model on the server commenced with random parameter initialization.

b) Operational Modes and Heterogeneity Simulation:: Two operational modes were compared: Federated Learning (FL) and isolated Local-Only training (Figure 10). Client heterogeneity was simulated by applying the same five distinct archetype-specific noise profiles detailed in Table II (previously used for preliminary sensor exploration) to each client's respective 20,000-sample training dataset. This process, illustrated in Figure 11, included introducing ECG noise, HAR context noise, and meta-label noise. Each client employed an 80/20 stratified split of its (noisy) data for local training and local validation sets. All features were scaled on a per-client basis using a `MinMaxScaler` fitted exclusively on its local training data. The evaluation strictly adhered to a "Train on Noisy, Validate on Clean" paradigm.

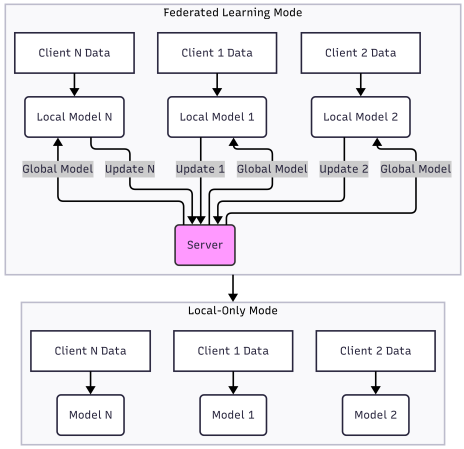


Figure 10: Comparison of operational modes: Isolated Local-Only training versus the Federated Learning (FL) topology with a central server.

c) Local Training and Update Protocol:: Clients initiate local model retraining (10 epochs for the sensor model; 15 epochs for the meta-model) when their respective data accumulation buffers reach dynamically adjusted thresholds. Training utilizes the Adam optimizer [**?**] and weighted Cross-Entropy loss. A client contributes its model update to the server only if the model's loss on its local clean validation set improves by more than 1% compared to its state prior to the local training round, ensuring only meaningful updates are propagated.

d) Federated Aggregation (FL Mode):: The server aggregates a minimum of $K_s = 3$ sensor model updates and $K_m = 3$ meta-model updates using Federated Averaging (FedAvg) [**?**]. To specifically address statistical heterogeneity for the meta-

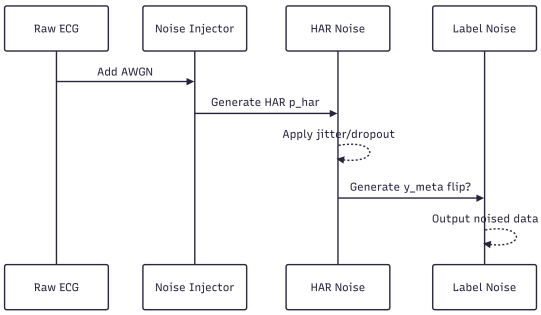


Figure 11: Simulated noise injection sequence for client training data generation, including AWGN for ECG, jitter/dropout for HAR context, and potential meta-label flipping.

model, clients employ the FedProx algorithm [2] (with proximal term coefficient $\mu = 0.01$) during their local meta-model training. This adds the regularization term $\frac{\mu}{2}||\mathbf{w} - \mathbf{w}_{\text{global}}||^2$ to the local loss function, where $\mathbf{w}$ are the local model parameters and $\mathbf{w}_{\text{global}}$ are the parameters of the current global meta-model. Clients synchronize with the server to download the latest global models every 45 simulated seconds.

e) Evaluation Strategy and Metrics:: Performance was assessed via local, client-side global, and server-side global evaluations using Accuracy, Macro F1-Score, Area Under the ROC Curve (AUC), and Cross-Entropy Loss. Per-class precision, recall, and F1-score were logged by the server to facilitate the construction of confusion matrices for the final global models. Meta-model validation data was dynamically generated at each evaluation stage (client or server) using the then-current sensor model on the respective clean ECG validation data to ensure relevance to the anomaly verification task.

V. Experimental Results and Evaluation

This section presents the evaluation outcomes for the Human Activity Recognition (HAR) component and the Federated Learning (FL) framework for ECG anomaly detection and verification.

A. Human Activity Recognition (HAR) Component Performance

The HAR component, leveraging a Temporal Convolutional Network (TCN), was independently trained and evaluated. Training utilized a custom dataset of 200 videos augmented with public data, while testing was performed on a held-out set including sequences from the Harvard Dataverse and 50 additional proprietary videos.

1) HAR Model Architecture Comparison: Table III summarizes the performance of various deep learning architectures explored for HAR. The TCN model demonstrated the highest overall accuracy.

2) Selected TCN Model: Detailed Performance: The confusion matrix for the selected TCN model is shown in Figure 12. Per-activity accuracies are detailed in Table IV, indicating strong recognition for critical activities such as 'fall' (94.12%) and common activities like 'walking' (95.00%).

Table III: Performance Comparison of Implemented HAR Model Architectures.

Model Architecture	Overall Acc. (%)	Critical Acts. Acc. (%)	Regular Acts. Acc. (%)
ST-GCN	74.78	72.06	73.35
DeepConvLSTM	33.91	29.41	27.09
Transformer	74.78	68.75	71.78
TCN	**86.09**	**78.31**	**85.94**
LSTM w/ Attention	78.26	62.87	79.35

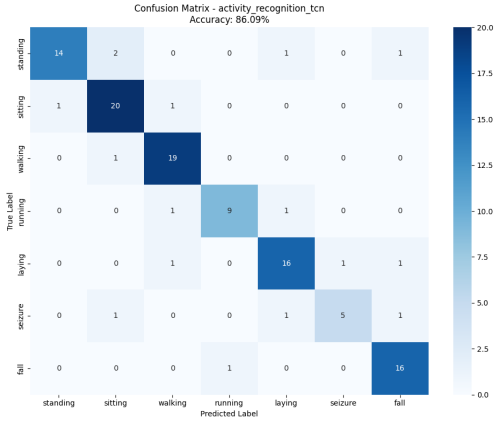


Figure 12: Confusion Matrix for the TCN-based HAR model (Overall test accuracy: 86.09%).

Table IV: Per-Activity Accuracy (%) for the TCN-based HAR Model on Test Data.

Activity	Accuracy (%)
standing	77.78
sitting	90.91
walking	95.00
running	81.82
laying	84.21
seizure	62.50
fall	94.12

B. Federated Learning for ECG Anomaly Detection and Verification

1) Preliminary Sensor Model Exploration: Accuracy vs. Computational Cost: An initial FL study (5 heterogeneous clients, 10k rows/client, archetypes from Table II) compared several sensor architectures. Table V summarizes their final average client accuracy and relative computational cost (average simulation time per client). While ResNet1D (∼94% accuracy) and BiLSTM (∼90% accuracy) achieved high performance, the `SensorECGCNN_Light` model (∼87% accuracy) provided a superior balance with substantially lower computational overhead (611.79s simulation time) compared to the more complex BiLSTM (2927.51s) and ResNet1D (4852.14s) models. This informed its selection for the main, more extensive FL simulations.

2) Main Simulation: `SensorECGCNN_Light` Performance (FL vs. Local-Only): In the main FL simulation ($N = 5$ clients, ∼20k samples/client, using archetypes from Table II),

Table V: Preliminary Sensor Model Exploration: Final Average Client Accuracy and Relative Computational Cost (5 Clients, 10k rows/client, FL Mode).

Sensor Model Architecture	Avg. Client Accuracy	Avg. Sim. Time per Client (s)
MLP with PCA	0.75	300.15
CNN Light	0.87	611.79
GRU	0.75	491.53
BiLSTM	0.90	2927.51
ResNet1D	0.94	4852.14

the FL mode generally enhanced sensor model performance compared to Local-Only training. Table VI details the final accuracy and loss achieved by each client on its local clean validation set. Notably, clients exposed to higher data noise levels, such as Edge 2 (Wearable 1 archetype), demonstrated a marked improvement with FL, with accuracy increasing from 0.81 (Local-Only) to 0.87 (FL). Similarly, Edge 4 (Wearable 2 archetype) improved from 0.86 to 0.88. This trend underscores FL's capability to foster more robust models, especially for clients operating with compromised local data quality.

Table VI: Final Sensor (`SensorECGCNN_Light`) Metrics on Local Clean Validation Sets: FL vs Local-Only Training (Lower Loss is Better).

Edge ID	Archetype Description	Accuracy ↑		Loss ↓	
		FL	Local	FL	Local
1	Hospital 1 (Clean)	0.91	0.89	0.31	0.26
2	Wearable 1 (Noisy)	**0.87**	0.81	0.31	0.41
3	Clinic (Mixed)	0.91	0.88	0.33	0.32
4	Wearable 2 (Noisy+)	**0.88**	0.86	0.33	0.30
5	Hospital 2 (Clean)	0.90	0.88	0.26	0.31

3) Main Simulation: `MetaClassifier` Performance (FL vs. Local-Only): The efficacy of the `MetaClassifier` in verifying sensor-detected anomalies was significantly enhanced through FL with client-side FedProx regularization. Table VII presents the final F1-score and AUC achieved by the meta-model on each client's dynamically generated clean local validation data. Clients training in the FL mode consistently outperformed their Local-Only counterparts. For instance, Edge 2 (Wearable 1, Noisy) saw its F1-score improve from 0.68 (Local-Only) to 0.80 (FL), and its AUC from 0.75 to 0.88. Edge 4 (Wearable 2, Very Noisy) experienced an F1-score increase from 0.60 to 0.76 and an AUC increase from 0.70 to 0.79 with FL. This demonstrates FL's effectiveness in leveraging collective knowledge to build a more accurate and reliable anomaly verification layer, especially mitigating the impact of severe local data degradation.

4) Global Model Performance and Convergence in FL Mode: The server-aggregated global models demonstrated consistent learning and improvement over 46 federation rounds. The global sensor model achieved a final AUC of approximately 0.95 on the server's held-out validation set. The global meta-

Table VII: Final Meta-Model Metrics on Local Clean (Dynamic) Validation: FL (FedProx) vs Local-Only (Higher F1/AUC is Better).

Edge ID	Archetype Description	F1-Score ↑		AUC ↑	
		FL	Local	FL	Local
1	Hospital 1 (Clean)	0.85	0.83	0.92	0.91
2	Wearable 1 (Noisy)	**0.80**	0.68	**0.88**	0.75
3	Clinic (Mixed)	0.82	0.75	0.90	0.84
4	Wearable 2 (V.Noisy)	**0.76**	0.60	**0.79**	0.70
5	Hospital 2 (Clean)	0.84	0.82	0.91	0.90

model, evaluated on dynamically generated server validation data (to reflect its dependency on the current sensor model), achieved a final F1-score of 0.881. The confusion matrix for this final global meta-model (Figure 13) details its performance in the anomaly verification task. It correctly confirmed 7382 sensor True Positives (TPs) and correctly rejected 1529 sensor False Positives (FPs). The primary error category was 573 False Negatives, where true sensor alerts were incorrectly dismissed by the meta-model.

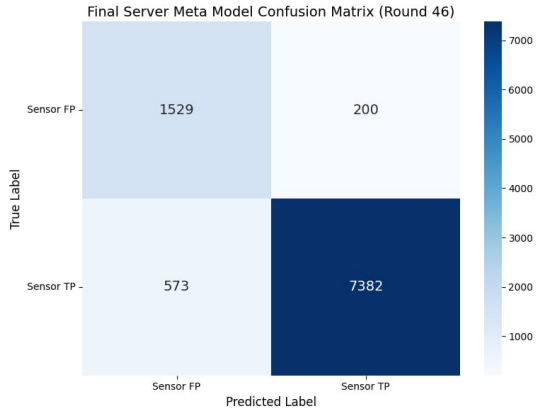


Figure 13: Confusion Matrix for the final global `MetaClassifier` model (Round 46) on server validation data, illustrating performance in verifying sensor 'Abnormal' predictions. True Label 'Sensor FP' means the sensor was wrong (actual normal); 'Sensor TP' means the sensor was right (actual abnormal).

C. Computational Cost and Edge Device Feasibility

Edge device operations were simulated on a MacBook Air M2, 16GB RAM, with artificial delays targeting a 1.5 GFLOPs/sec hardware profile to estimate performance on resource-constrained devices. Table VIII summarizes these results for the main FL simulation (5 clients, 20k rows/client). Sensor inference averaged a highly efficient 1.52 ms per sample. Periodic sensor model retraining cycles averaged 41.06s, while meta-model retraining cycles were substantially quicker at 0.65s. These timings suggest that real-time inference is readily achievable, while the frequency and impact of on-device retraining would need careful consideration in a deployment scenario, potentially favoring less frequent updates or offloading for more complex models.

Table VIII: Average Computational Metrics per Simulated Edge Device (FL Mode, 20k rows/client, Simulated Target 1.5 GFLOPs/sec).

Metric	Average Value
Total Simulation Time per Client (s)	1562.72
Sensor Model Inference (ms/sample)	1.52
Sensor Model Retraining (s/cycle)	41.06
Meta-Model Retraining (s/cycle)	0.65

VI. Implemented Software

This section outlines the developed software components of the Smart-Care system, categorized into backend services, frontend interfaces for caregivers and patients, and visualization features for real-time interaction and monitoring.

A. Backend

The backend system facilitates communication between patient devices and caregiver applications. To securely expose the local development environment to the internet, **ngrok** was used as a tunneling service. This enabled seamless data exchange between distributed clients without requiring static IPs or public hosting.

The backend server was responsible for receiving sensor data from the patient's Apple Watch, locally using **TinyDB**, and making it accessible to the caregiver application upon request. **TinyDB**, a lightweight document-oriented database, was chosen for its simplicity and suitability in handling key-value storage in constrained environments. It also acted as a relay point for real-time communication via **WebRTC**.

To ensure timely alerts and emergency notifications, **Firebase Cloud Messaging (FCM)** was integrated into the system. FCM enabled the backend to push instant notifications to the caregiver application when critical health events were detected or when real-time attention was needed.

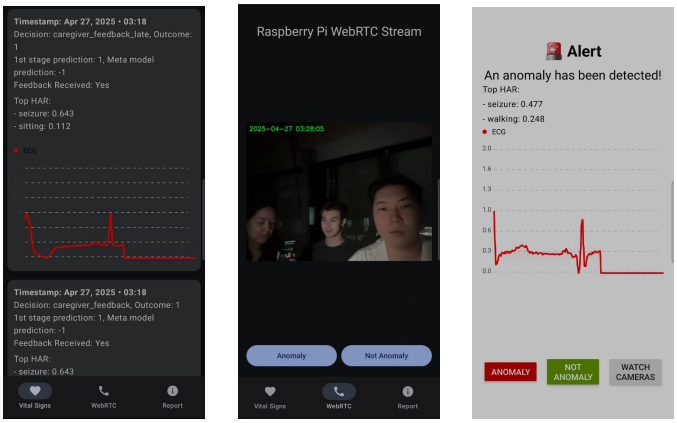
B. Frontend

Two frontend applications were developed as part of the Smart-Care system. The Caregiver Application is an Android-based app designed to assist caregivers by providing functionalities such as accessing health reports sent from the patient's device, receiving real-time alerts via FCM, and viewing live video streams through WebRTC for continuous patient monitoring.

On the other hand, the Patient Application is an Apple Watch app that collects real-time physiological data—such as heart rate, ECG and movement — and transmits it to the local server. This application runs unobtrusively in the background, enabling continuous and seamless health monitoring without disrupting the patient's daily routine.

C. Visualization

Figures 16a, 14b, and 15a illustrate the major user interface components of both the caregiver and patient applications.



(a) Report **(b)** Live Stream **(c)** Notification

Figure 14: Screens from the Caregiver Android App

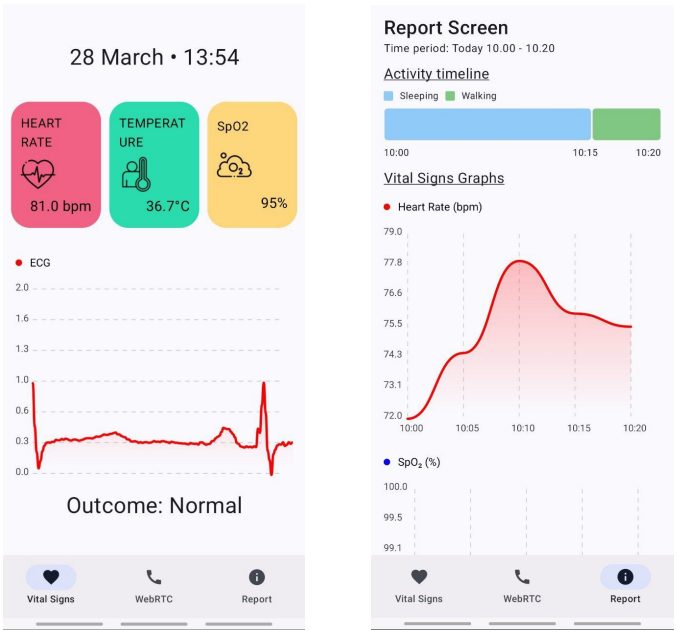
The screens in Figure 14 represent the system interface where ECG is the primary parameter used for anomaly detection. In contrast, Figure 16 demonstrates an enhanced version that includes additional vital signs such as heart rate, body temperature, and blood oxygen saturation alongside ECG. Notably, Figure 16 showcases a reporting interface that allows the caregiver to select a custom time range for generating detailed reports, complete with an activity timeline synchronized to specific timestamps. This timeline is rendered using a custom drawing function implemented via Jetpack Compose's built-in Canvas library.

(a) App Interface **(b)** Notification

Figure 15: Screens from the Apple Watch Patient App

To enable advanced visualization capabilities in the Caregiver application, two open-source libraries were integrated. For charting and graphical representation of sensor data, the "AAY-Chart" library developed by TheChance101 was used. For the live video stream interface, the "stream-video-android" SDK from GetStream was employed, which provides a reliable and customizable WebRTC-based solution for real-time communication.

On the patient side, the Apple Watch application provides an intuitive interface for continuous health monitoring. As shown in Figure 15, the app uses ”CoreMotion” and ”HealthKit” frameworks from Apple to access real-time physiological signals, including heart rate and motion data.

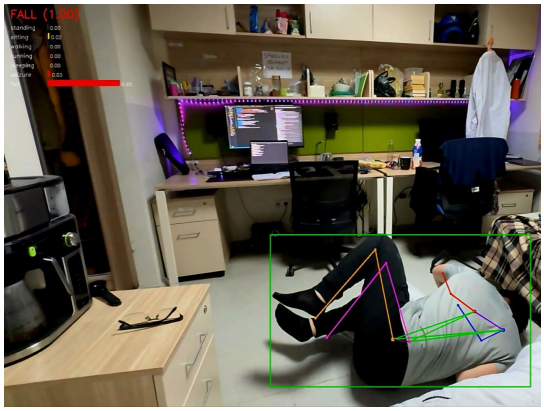


(a) Real Time Monitoring **(b)** Report Visualization

Figure 16: Screens for Representing Vital Signs

To provide immediate feedback and insight into the system's operation, our HAR visualizations are designed for clarity and real-time understanding, as illustrated in Figure 17. These visualizations prominently display the currently recognized activity along with its associated confidence score, exemplified by the 'RUNNING' detection in Figure 17(a) and the 'Fall' event in Figure 17(b). Complementing this, a dynamic bar graph, visible in both subfigures, illustrates the confidence distribution across a predefined set of activities (e.g., standing, sitting, walking, seizure), offering a comprehensive view of the model's predictions. Furthermore, as seen in the examples, a bounding box localizes the detected individual, while a skeletal pose estimation overlay maps their key body joints, aiding in the visual interpretation of the recognized action. Crucially, upon confident detection of these activities, the system logs this information—including the activity label, confidence score, and timestamp—and saves it to local storage on the device. This ensures a persistent record of significant events, facilitating offline review, data analysis, or triggering further actions based on the historical activity data.

(a) Visualization of the system detecting a 'Running' activity, including pose estimation and confidence scores.



(b) Visualization of the system detecting a 'Fall' event, with corresponding pose and activity confidence levels.

Figure 17: Examples of HAR visualizations for 'Running' and 'Fall' scenarios. The system displays detected activity, confidence scores, and pose estimation overlays.

VII. Conclusion and Future Work

A. Future Work

For the future phases of the system, a number of significant extensions and improvements are planned:

- **Authentication and Access Control:** To differentiate roles (administrators, caregivers, and patients), limit access, and protect patient data confidentiality, implement secure authentication features.
- **Full System Integration and Automation:** Include in place a completely automated pipeline that, upon turning on the system (such as a Raspberry Pi at home), initiates the required scripts, sets up services, establishes connections with sensors and mobile apps, and plugs in monitoring.
- **HAR Models in FL Framework:** To enhance context validation in emergency detection and further customize the system, incorporate sophisticated HAR models into the federated learning configuration.
- **Encrypted Federated Training:** To increase security without sacrificing performance, look into training models on encrypted data via safe multiparty computation or homomorphic encryption.
- **Cloud Infrastructure and Scalability:** To accommodate large-scale and multi-user scenarios, deploy components (data storage, federated aggregation server, and authentication) on scalable cloud platforms such as AWS or Google Cloud.

- **Cross-Platform Mobile App and UI Enhancements:** Complete and optimize iOS and Android apps for dependable, low-latency operation. Increase the functionality of device diagnostics, Bluetooth Low Energy pairing, and offline mode.
- **Broad Sensor Support Beyond Smartwatches:** Although the current implementation uses smartwatches to collect vital signs, future iterations will investigate integration with additional ambient or wearable sensors, including EMG sleeves, chest bands, ECG patches, and environmental Internet of Things devices (e.g., bed pressure sensors, fall detectors).

These future steps will bring the system closer to being ready for the real world by making sure it is flexible, private, medically relevant, and scalable in a variety of assisted living and healthcare environments.

B. Conclusion

This project demonstrated a reliable and privacy-preserving real-time health monitoring and emergency detection system. The solution combines wearable IoT sensors, federated learning (FL), and active learning (AL) into a two-stage detection framework. By combining physiological signals and activity recognition, the system ensures accurate anomaly detection while reducing false positives.

The architecture includes edge computing for local inference, federated learning for decentralized model training, and real-time notifications via mobile apps. A WebRTC-based communication feature enhances caregiver-patient interaction during emergency situations. The mobile app interface, dashboard visualizations, and notification modules provide an intuitive and responsive user experience for real-world deployment.

REFERENCES

[1] A. Gupta, S. Misra, N. Pathak, and D. Das, "Fedcare: Federated learning for resource-constrained healthcare devices in iomt system," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1587–1596, 2023.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of healthcare informatics research*, vol. 5, pp. 1–19, 2021.

[4] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[5] U. Anliker, J. A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. B. Schenker, F. Catarsi, L. Coluccini *et al.*, "Amon: a wearable multiparameter medical monitoring and alert system," *IEEE Transactions on information technology in Biomedicine*, vol. 8, no. 4, pp. 415–427, 2004.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[7] X. Liang, J. Tang, and T. Q. Quek, "Large-scale decentralized asynchronous federated edge learning with device heterogeneity," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 4566–4571.

[8] M. Cvach, "Monitor alarm fatigue: an integrative review," *Biomedical instrumentation & technology*, vol. 46, no. 4, pp. 268–277, 2012.

[9] C.-F. Hsu, Y.-C. Li, C.-C. Tsai, J.-K. Wang, and C.-H. Hsu, "Federated learning using multi-modal sensors with heterogeneous privacy sensitivity levels," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 11, pp. 1–27, 2024.

[10] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[11] G. J. L. Paulraj, I. J. Jebadurai, S. P. Janani, M. S. Aarthi *et al.*, "Edge-based heart disease prediction using federated learning," in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)*. IEEE, 2024, pp. 294–299.

[12] V. Hayyolalam, S. Otoum, and Ö. Özkasap, "A hybrid edge-assisted machine learning approach for detecting heart disease," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 2966–2971.

[13] S. Besrour, G. S. Mubibya, C. B. Abdeljelil, and J. Almhana, "Generalization vs personalization: A trade-off for better data heterogeneity impact mitigation in fl," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 3207–3212.

[14] Q. Lin, K. Xu, Y. Huang, F. Yu, and X. Wang, "Privacy-enhanced data fusion for federated learning empowered internet of things," *Mobile information systems*, vol. 2022, no. 1, p. 3850246, 2022.

[15] Y. Djenouri and A. N. Belbachir, "Empowering urban connectivity in smart cities using federated intrusion detection," in *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2023, pp. 1–9.

[16] A. J. J, G. J. Leelipushpam Paulraj, G. R. M, I. J. Jebadurai, S. P. Janani, and M. S. Aarthi, "Edge-based heart disease prediction using federated learning," in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*, 2024, pp. 294–299.

[17] K. Doulani and M. Adhikari, "Real-time detection and monitoring of contagious diseases using wearable sensors and lightweight model in edge networks," *IEEE Sensors Journal*, 2025.

[18] M. Malekzadeh, B. Hasircioglu, N. Mital, K. Katarya, M. E. Ozfatura, and D. Gündüz, "Dopamine: Differentially private federated learning on medical data," *arXiv preprint arXiv:2101.11693*, 2021.

[19] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2020.

[20] P. Sharma and S. Sharma, "An effective fl-cnn based data securing model for heart disease prediction," in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, vol. 6. IEEE, 2023, pp. 1862–1866.

[21] M. Vucovich, A. Tarcar, P. Rebelo, A. Rahman, D. Nandakumar, C. Redino, K. Choi, R. Schiller, S. Bhattacharya, B. Veeramani *et al.*, "Anomaly detection via federated learning," in *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023, pp. 259–266.

[22] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[23] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*. Springer, 2019, pp. 92–104.

[24] J. Li, Y. Meng, L. Ma, S. Du, H. Zhu, Q. Pei, and X. Shen, "A federated learning based privacy-preserving smart healthcare system," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, 2021.

[25] Z. Yan, J. Wicaksana, Z. Wang, X. Yang, and K.-T. Cheng, "Variation-aware federated learning with multi-source decentralized medical image data," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 7, pp. 2615–2628, 2020.

[26] B.-S. Lin, T. Yu, C.-W. Peng, C.-H. Lin, H.-K. Hsu, I.-J. Lee, and Z. Zhang, "Fall detection system with artificial intelligence-based edge computing," *IEEE Access*, vol. 10, pp. 4328–4339, 2022.

[27] R. Sharma and S. Rani, "Quantified self: From self-learning to machine learning," *IT Professional*, vol. 23, no. 4, pp. 69–74, 2021.

[28] T. Sharon, "Self-tracking for health and the quantified self: Re-articulating autonomy, solidarity, and authenticity in an age of personalized healthcare," *Philosophy & Technology*, vol. 30, no. 1, pp. 93–121, 2017.

[29] A. Yazici, D. Zhumabekova, A. Nurakhmetova, Z. Yergaliyev, H. Y. Yatbaz, Z. Makisheva, M. Lewis, and E. Ever, "A smart e-health framework for monitoring the health of the elderly and disabled," *Internet of Things*, vol. 24, p. 100971, 2023.

[30] A. Akhmetov, Z. Latif, B. Tyler, and A. Yazici, "Enhancing healthcare data privacy and interoperability with federated learning," *PeerJ Computer Science*, vol. 11, p. e2870, 2025, accessed: 2025-04-25. [Online]. Available: https://doi.org/10.7717/peerj-cs.2870

[31] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000, [Online]. [Online]. Available: https://doi.org/10.1161/01.CIR.101.23.e215