International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(7s) 2026 pp. 122-142
https://doi.org/10.70917/ijcisim-2026-3049
Article

ADAPTIVE DRIFT-AWARE FEDERATED LEARNING FRAMEWORK FOR HEART DISEASE PREDICTION

D. Ganesh[1], O. B. V. Ramanaiah[2]

[1]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Kukatpally, Hyderabad, India

[2]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Kukatpally, Hyderabad, India

Corresponding Author: D. Ganesh,Email: ganeshsingh2502@gmail.com

Abstract: Heart disease develops gradually through variations in physiological parameters such as heart rate and oxygen saturation, often remaining undetected until a cardiac event occurs. Most prediction models in current use depend on centralized data collection, which creates privacy risks and limits deployment across distributed healthcare institutions. Additionally, the statistical properties of the physiological data evolve over time due to changes in patients' health conditions and behavioural patterns, resulting in concept drift that progressively degrades model performance. This paper proposes the Drift-Aware Federated HeartCare (DFHC) framework, a unified federated learning architecture that simultaneously addresses privacy preservation and adaptive concept drift handling for heart disease prediction. Within the proposed framework client devices train models on local data and share only parameter updates, ensuring raw data never leaves the source device. A dual-stage drift detection component monitors physiological signals and triggers adaptive model migration when concept drift is detected. The proposed framework integrates ADWIN-DDM-based drift detection with cosine-similarity-driven adaptive model migration. Experimental results demonstrate that the proposed DFHC framework achieves a classification accuracy of 94.3% and an F1-score of 0.92, outperforming standard FedAvg, FedProx, and Per-FedAvg, while converging in fewer communication rounds

Keywords: Adaptive Model Migration, Concept Drift Detection, Edge Computing, Federated Learning, Heart Disease Prediction, Wearable Health Devices.

1. Introduction

Globally, cardiovascular diseases are one of the leading causes of morbidity and mortality-with around 17.9 million individuals dying each year because of cardiac issues [1]. For these reasons, it is important to obtain a prompt diagnostic assessment of arrhythmias and related irregular heartbeat disorders and then maintain ongoing tracking through appropriate means to reduce overall morbidity/mortality for cardiac-related disorders [2], [3]. Real-time physiological monitoring through client, and Internet of Things (IoT) technologies allows healthcare professionals to monitor patients' heart rates and identify abnormalities such as arrhythmias and irregular heartbeat disorders [4], [5]. The widespread use of cloud computing has raised concerns regarding patient privacy, health data security, and communication latency [6].

Federated learning (FL) aids health care analytics while preserving privacy by training models across distributed devices without directly sharing sensitive patient information. [7], [8]. However, a significant research gap remains: the absence of concept-drift FL models designed for resource-constrained wearables with adaptive drift-handling capabilities.

Research across several institutional settings has demonstrated that federated learning can support ECG classification, arrhythmia detection, and medical image analysis without requiring raw data to leave its source [9–12].

Federated learning-based cardiac monitoring systems present certain advantages, but they are not without challenges. Statistical heterogeneity, bandwidth limitations, and concept drift are among the factors that can reduce model accuracy when these systems are deployed in real-world healthcare settings [13], [14].

Data in healthcare environments is not fixed. Over time, patient conditions change, clinical contexts differ across sites, and the statistical properties of incoming data shift in ways that were not present during model training. This phenomenon is known as concept drift [35]. In wearable cardiac monitoring, the problem is particularly persistent because patient physiology and sensor readings change on an ongoing basis. A model trained on data collected at an earlier point may therefore produce outputs that are less accurate as time progresses and conditions diverge from what the model originally learned.

Two methods that have been proposed to address this are Weighted Aggregation [15] and Adaptive Drift Detection [21]. Both are designed to detect distributional shifts in real-time cardiac monitoring data and respond to unwanted variations as they occur. Despite this, how well either method holds up in actual clinical environments is not well established. Their effectiveness in practice remains largely unverified, which points to a gap that warrants further investigation. Additionally, the integration of wearable health devices with emergency medical response systems enables applications such as fall detection [23], diver safety monitoring [24], and continuous bio signal monitoring in emergency departments. These systems emphasize the importance of automatic alert mechanisms during network failures or life-threatening events, along with fail-safe systems to ensure continuous cardiac monitoring. However, most traditional heart monitoring devices do not support integration with federated learning (FL) architectures, leaving monitoring systems isolated and restricting adaptive, personalized care.

In order to overcome the above challenges, we have designed a DFHC model based on a client-server architecture which allows for not only secure and privacy preserving analytics but also adapts to concept drift through Adaptive Concept Drift Handling (ACDH). In the proposed DFHC model, various FL models are used in order to make the intelligent diagnostics system adaptable to the changes in the user behaviour and physiological conditions. This research intends to provide user privacy protection while using the DFHC model and also provides an effective mechanism for dealing with heart diseases.

1.1 Research Contributions and Novelty:

The proposed Drift-Aware Federated HeartCare (DFHC) system improves on the existing federated learning approach to health care analysis in the following innovative ways:

1. Unified architecture of a drift-aware federated learning system to analyze heart diseases, providing solutions to concept drifts and privacy issues.
2. A two-level process of detecting concept drifts in order to identify both abrupt physiological changes as well as gradual behavioural changes using ADWIN and DDM together.
3. Use of a cosine similarity measure to adaptively migrate models when needed, deciding whether it is necessary to replace, transfer, or retain models based on this criterion.
4. A light-weight approach specifically suitable for wearable devices with constrained resources due to low overhead in drift detection and adaptive communication.
5. Validation on four diverse physiological databases (MIMIC-III, WESAD, PAMAP2, and PPGDaLiA) to ensure higher predictive performance, quick recovery from concept drift, and minimal communication cost than previous federated learning systems.

These innovations pave the way to developing an adaptive and privacy-preserving intelligent health care system.

2. Related Work

Federated Learning has become increasingly popular in the realm of health-care research as it presents an opportunity to learn models without moving data from where they are stored [7], [8]. In federated learning, participants who participate in model learning maintain their data and only share information on how changes have been made to the model. This approach mitigates the problems of centralized data storage in terms of confidentiality and security. Heart disease detection is one example where this becomes relevant since the algorithm can learn from physiological data gathered from different sources.

This issue was investigated in [9] by Agrawal et al., who introduced federated learning along with differential privacy applied to ECG datasets from several hospitals for heart disease monitoring purposes. From the findings of the experiment, it can be seen that the obtained model performs similarly to that which would have been trained under the centralized training approach; however, none of the patients' information ever left any of the hospitals. Thus, it seems that there is no necessity for precision and privacy to be at odds with one another.

Tang et al. [10], on the other hand, developed an approach based on the development of a federated learning system tailored for ECG classification tasks. Due to the varying distribution of data in each clinical site, Tang et al. introduced a module for feature alignment. The patients' information remained within each respective clinical site during the experiment. It thus fulfilled the criteria for ensuring privacy while achieving accurate results for arrhythmia detection. While they claimed that the approach can be adopted in practical applications, it failed to validate its applicability due to insufficient conditions.

The topic of federated learning has frequently emerged in systematic reviews within the field of healthcare, highlighting the potential of the technology in improving the accuracy of predictions within distributed systems [18], [19], [20]. The management of large-volume health streams and associated security aspects due to distributed storage have been explored for the big data architecture framework [17]. The research literature has raised multiple issues with the federated learning paradigm that remain unresolved, such as communication overhead and scalability problems. In order to circumvent such problems, model migration and transfer learning approaches have been adopted. Recently, there has been a survey on applying federated learning for intelligent health-care that offers an overview and highlights the incorporation of drift awareness as a future scope [28].

The matter of interpretability was taken into account by Raza et al. [14], who proposed a novel approach by integrating federated learning methodologies with those of explainable artificial intelligence applied to electrocardiograms. In general, the goal pursued through such research was that of providing better insights into the reasoning process behind machine learning outcomes. On the other hand, Zahid et al. [16] chose a different route, proposing a novel framework based on self-operating neural networks along with feature injections for improving the accuracy of classification under challenging clinical conditions.

The idea of record-based model shifting was proposed as a way for devices to request more suitable models for their environment, thus minimizing the reduction in performance resulting from exposure to data unseen during training [12]. Although this research area is certainly relevant and interesting, it does not make federated learning approaches aware of the migration process irrelevant, since such approaches remain relevant for resource-constrained mobile/wearable devices.

The evolution of aggregation strategies for federated learning has, to some extent, been influenced by the requirements imposed on addressing the issues that arise in distributed training with clients having different data environments and computational power. Federated Averaging, or FedAvg [32], can be considered a strategy that was developed in response to such a requirement and became a de facto benchmark against which other approaches were compared. As its main algorithmic routine, FedAvg relies on collecting model updates from clients trained locally and constructing a single model based on computing the weighted average of these updates, where the weight of each update is proportional to the size of the corresponding dataset. Speaking of resource consumption, FedAvg is a relatively cost-efficient approach in both computational and communication terms. However, one of the major weaknesses of FedAvg lies in its assumption about the equal value of all client updates irrespective of their data sources. It is obvious that, in cases of non-IID data distributions, the assumption about the equal value of updates leads to suboptimal or even incorrect model construction. Concept drift makes things worse by introducing variability in data distributions, which is not addressed by FedAvg.

FedProx [33] was introduced as a solution to the problem of instability that comes with federated learning whenever the distribution of the data among the clients becomes too heterogeneous. FedProx includes a regularizing term added to local training which acts as a boundary condition, ensuring that local learners do not stray too far from the global model at each training iteration. FedProx therefore promotes some form of uniformity among the learners, resulting in a more stable convergence process even in cases of highly heterogeneous training environments. This trait is useful in medical applications, especially in monitoring using wearables, whereby the biometric data obtained from individual patients are likely to differ in structure and magnitude.

The approach adopted by Per-FedAvg [34], on the other hand, is based on a somewhat different view of the issue in question. It is inspired by the concept of personalization and utilizes Model-Agnostic Meta-Learning framework to derive an initial global model that is not seen as a definitive solution itself, but simply serves as a baseline that needs to be fine-tuned by every single client individually via few steps of parameter update. The method's focus, however, remains on this initial adaptation phase, and it does not provide a built-in way to handle situations where data distributions continue to change after that phase has concluded.

Both of these strategies represent actual progress in the realm of federated optimization; however, both fail to take into consideration the requirements posed by systems designed to react to change in real time. Drift detection and model migration capabilities are not accounted for in either strategy, which can prove to be somewhat of an obstacle when dealing with environments where change occurs in an unpredictable fashion. One such environment could be the one associated with cardiac health assessment. The patient's physiological condition as well as the patterns within their sensor data may change over time.

The concept of concept drift in conjunction with personalization has also been explored in relation to federated edge-based architectures. In particular, health-related data suffers from such instability because patients' behavior varies over time due to different aspects like changing lifestyles, medicines, and environment. This affects predictive model drift. Several approaches have been proposed to overcome these issues. Weight aggregation [15], along with adaptable client grouping frameworks, e.g., Flash [21], has been seen. The aspect of personalization in health care federated learning has also been highlighted by Bishnoi [29], which needs to adapt, not only to global population data but also individual patient data. However, these methods are largely confined to experimental setups and have yet to be extended to robust, real-time wearable monitoring applications.

Simultaneously, clients are increasingly used as platforms for real-time health monitoring. Systems such as smartwatch bands with emergency alerts [25], wearable fall detectors for firefighters [23], and underwater ECG monitors [24] demonstrates the feasibility of implanting safety mechanisms in compact devices. Hospital implementations have also confirmed the technical viability of continuous biosensor monitoring in emergency departments [24]. More recent systems such as MyWear [22] and Smart Shield [26] integrate IoT and GPS with wearable sensing to enable continuous monitoring with location-based alerts. Agarwal et al. [27] examined the use of wearables among healthcare workers. These contributions emphasize the potential of wearable-based FL ecosystems to support continuous cardiac monitoring. However, most existing works operate in isolation and do not address the mutual challenges of drift adaptation, model migration, and fail-safe responsiveness within a FL architecture. Taken together, these studies highlight that federated learning, transfer learning, drift-aware adaptation, and wearable sensing each make valuable contributions to cardiac monitoring. However, current approaches still treat these aspects separately, and no single framework effectively integrates all four into a cohesive solution.

A gap remains, however, in the availability of federated learning models designed specifically for resource-constrained wearables that can detect and respond to concept drift as it occurs. The importance of filling this gap bears immediate implications for the development of heart-monitoring technologies, which will have to function reliably without room for error in instances where such an error cannot be tolerated. This kind of technology could then become the basis for medical decision-making in real time and help avoid the occurrence of sudden heart-related deaths of people who did not realize they suffered from heart problems.

Table 1: Summary of Related Work and Identified Research Gaps

Reference	**Name of Author**	**Year**	**Approach / Contribution**	**Federated Learning**	**Drift**	**Privacy**	**Identified Research Gap**
[35]	Gama et al.	2014	Foundational survey of concept drift adaptation: DDM and ADWIN detectors, taxonomy, and handling strategies		✓		Centralized batch-learning context; drift detectors not evaluated in FL settings or on physiological data streams
[32]	McMahan et al.	2017	FedAvg: Weighted averaging of local model updates from distributed clients	✓			Uniform client weighting degrades under non-IID data; no mechanism to detect or respond to distributional shift
[33]	Li et al.	2020	FedProx: Proximal regularization term constraining local update divergence in heterogeneous FL	✓			Stabilizes convergence but static regularization cannot adapt when data distributions evolve post-deployment

[34]	Fallah et al.	2020	Per-FedAvg: MAML-based meta-learning for fast per-client personalized fine-tuning	✓			Supports fast initial adaptation only; provides no continuous monitoring for ongoing concept drift
[22]	Sethuraman et al.	2021	MyWear: Smart garment for continuous multi-vital IoT monitoring with automated alert generation				Isolated sensing device; no FL collaboration, no concept drift handling, and no adaptive model update mechanism
[10]	Tang et al.	2021	Personalized FL with cross-site feature alignment module for ECG arrhythmia classification	✓		✓	Feature alignment fixed at training time; model accuracy declines silently when patient signal characteristics shift
[18]	Nguyen et al.	2022	Comprehensive survey of FL for smart healthcare: architectures, privacy mechanisms, and open challenges	✓		✓	Survey identifies drift and adaptive deployment as critical open challenges but proposes no concrete solution
[14]	Raza et al.	2022	FL with transfer learning and Explainable AI (XAI) for interpretable ECG monitoring system	✓		✓	Interpretability-focused design; transfer learning applied once, not continuously; no drift detection component
[2]	Bebortta et al.	2023	FedEHR: FL-based early heart disease detection integrating EHR records with IoT sensor data	✓		✓	Sensor data integrated but model is static; no drift detection or migration as patient physiology evolves over time
[11]	Khan et al.	2023	Asynchronous FL with AI for cardiovascular disease prediction across distributed clients	✓		✓	Stale gradients from async updates reduce accuracy; no drift detection component
[13]	Jothimurugesan et al.	2023	Theoretical FL framework under distributed concept drift using drift-partitioned client learning	✓	✓		Purely theoretical contribution; no cardiac or physiological dataset evaluation; no lightweight implementation
[21]	Panchal et al.	2023	Flash: Client clustering strategy for concept drift adaptation in federated learning (ICML 2023)	✓	✓		Cluster reallocation incurs high communication overhead; not evaluated on biomedical or physiological data
[9]	Agrawal et al.	2024	FL with differential privacy on multi-hospital ECG data; achieves near-centralized classification performance	✓		✓	Assumes static ECG distributions; no temporal drift detection or adaptive model update strategy
[15]	Asif et al.	2024	Weighted FL aggregation strategy for ECG arrhythmia	✓		✓	Static aggregation weights; no feedback loop to detect drift or trigger model refresh on individual clients

			detection across distributed clients				
[4]	Maheshwari et al.	2024	Drift-oriented adaptive framework for concept drift detection in large-scale IoMT data streams		✓		Centralized architecture requires raw data to leave the device; no privacy preservation or model migration on edge nodes

3. Proposed Methodology

However, federated learning is certainly feasible as far as its use within the healthcare industry is concerned; but if it is applied within wearable devices in scenarios where heart monitoring occurs, there will be some new challenges that arise with regard to implementation. The physiological measurements like ECG measurements, heart rate measurements, and oxygen levels can fluctuate according to a variety of parameters including changes in one's environment and even due to illness or the state of health of the person [13, 21]. This means that concept drift, which is a form of shifting in the training data, is a problem for federated learning algorithms as it leads to poor results [13, 21].

Based on this, there emerge two separate challenges. The first challenge is that of identifying when the model running on the wearable device ceases to adequately represent the incoming data, hence losing accuracy. This is known as the concept drift problem. The second challenge lies in the character of the proposed solution to this problem. It should be noted that wearables have limitations on memory and computing power, and the drift detection method has to be developed in such a way that the system would not require too many computing resources from the wearable device.

Determining at what point a particular device or sensor should be switched between different models comes with its own challenges, whether it involves pulling a new version of the model from the edge server or selecting a particular model from amongst multiple models available in the local repository. If these transitions occur very often, then the overall cost manifests itself in terms of increased use of bandwidth and battery. However, if the model transitions occur late for data conditions, then the device will end up using a model based on outdated data. Finding a balance between these two extremes involves great caution because the ultimate goal is to maintain reasonable prediction accuracy levels without exceeding the capacity of the device in matters of computational power, data transmission, and power consumption.

This study examines the issue of creating a model migration algorithm which is adaptive to drifts with the aim of enhancing learning and prediction efficiency within wearable scenarios where the distribution of data among patients is non-uniform and varies over time. The critical aspect here lies in the need to create a model migration algorithm that is able to maintain good performance despite the differences in patient data sets, while simultaneously minimizing the amount of energy consumed during the process of updating models.

From a more general perspective, this research effort aims at advancing the domain of federated cardiac monitoring through the development of a system which is able to meet multiple demands. Patient data privacy protection constitutes one such demand, but there is also the need for the system to monitor the progress of model performance, detect when the accuracy of prediction drops below an acceptable point, and update the model accordingly before the drop results in significant inaccuracies. Such a feature proves to be particularly useful in cases where the margin for decision-making is small and the stakes involved are high.

3.1. System Architecture and Design

The proposed Drift-Aware Federated HeartCare framework leverages a network of distributed clients and an edge server to collaboratively train a heart disease prediction model through federated learning. Clients refer to distributed client systems, including smartphones, wearable health devices, hospital workstations, clinic servers, and IoT sensors, which perform local model training and share updates within the federated learning framework. Each host conducts local model updates and may exchange encrypted model parameters with the edge server. The server aggregates these updates to refine a global model using the Federated Averaging (FedAvg [32]) algorithm, which computes a weighted average of model parameters from all clients. The overall system architecture is illustrated in Figure 1.

Federated Learning Formulation:

Let:

- k denotes the total number of clients

- n_k denote the number of data samples at client k.
- ${w_k}^t$ denote the local model parameters of client k at communication round t
- w^t denote the global model parameters at communication round t.

Each client performs local training on its private dataset and updates its model parameters using stochastic gradient descent as follows:

$$w_k^{t+1} = w_k^t - \eta\nabla L_k\,(w_k^t)\text{ -----------(1)}$$

where η is the learning rate and L_k represents the local loss function of client k.

After local training the edge server aggregates the updated local models using the Federate Averaging (FedAvg) algorithm:

$$w^{\ t+1} = \sum_{k=1}^{K} \frac{n_k}{\sum_{j=1}^{K} n_j}\, {w_k}^t\text{--------------(2)}$$

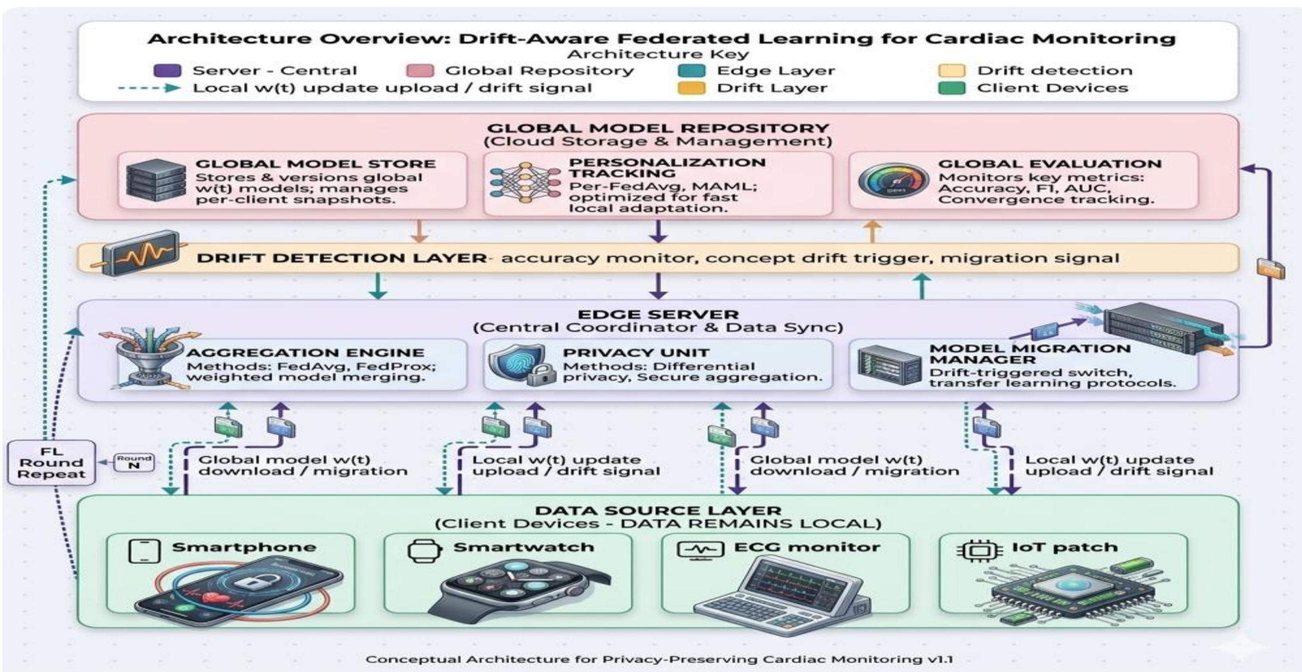

Figure 1. Architecture of the Drift-Aware Federated Learning Framework

Due to the limited processing power and battery life of clients, the framework incorporates an edge-assisted computation offloading mechanism. Edge computing addresses a practical problem that arises when devices with limited processing capacity are expected to carry out tasks that demand substantial computational resources. By transferring operations such as machine learning model training and inference execution to edge servers, the burden on individual devices is reduced to a more manageable level. Consequently, the system can rely on its resources in a more sustainable manner while ensuring the required level of efficiency by means of ensuring the low latency and continuous experience provided. The approach discussed in the current paper accounts for the fact that it is necessary to control concept drift and ensure the security of data transfer while updating models within the network. In terms of structure, the network includes several clients that act as distributed data collectors within the overall system.

In the network layer, the edge server serves as a central coordinating component and model storage. This layer not only holds customized models that are trained using the data at the hosts but also contains a common global model for all clients. It gathers updates from the hosts and improves the global model through techniques like federated averaging

In the server component, there exists a privacy-preserving unit tasked with making sure that the personal information of the users remains safe during the entire learning process. This is made possible using methods like differential privacy and secure aggregation, which guarantee that the sensitive information does not get leaked when the models are updated. In addition to this, there exists a compute offloading unit in the server component whose purpose is to facilitate hosts that work with constrained resources. This module allows hosts to transfer their training/inference processes to the edge server if the task exceeds local capabilities.

Concept drift, on the other hand, is addressed through the use of a module known as the model shift handler, which maintains a constant watch over the performance of models within the network. As patterns in data vary either as a result of changes in user behavior or in the nature of health states or any other reason, the accuracy of the local model can start becoming compromised. Once the reduction in performance becomes substantial enough to exceed a pre-defined threshold value, then a model switch would be triggered. This could be based on either selecting an updated model from a local model library or fetching one from the edge server. The end effect of all this would be that the deployed model continues to reflect the conditions within data.

During this entire operation, the handler maintains the overhead and downtime associated with this activity in such a manner that no further intervention is needed by the user. Before transferring any information, an evaluation is made on how much drift there is, what state is the device in, and what kind of network is available. Any transfers take place only if it is determined that this will not put too much strain on the client devices. In this fashion, the system retains its prediction capability along with managing energy requirements and network bandwidth usage.

A lightweight and secure communication protocol is used to establish the link between the clients and the edge server, even when there is a scarcity of bandwidth or the availability of the network itself is inconsistent. In order to safeguard the data of patients over the said link, data minimization together with secure aggregation is utilized, which ensures that no data can be accessed by others.

The entire approach of Drift-Aware Federated HeartCare is aimed at enabling concept drift-based adaptation under the federated learning setting, which incorporates scalability and privacy preservation characteristics. Updates to the model rely on information related specifically to the user, with the approaches to making these updates being designed in such a way as to ensure security throughout the process. It is intended for the updates to be both timely and applicable to each individual user's case, without sacrificing performance or privacy guarantees offered by the framework.

3.2. Dataset and Preprocessing

The proposed framework will be analyzed using four publicly available datasets, which are chosen based on different sources of physiological data that may be used for monitoring and activity recognition purposes related to cardiac applications.

The first one is the MIMIC-III, which is a clinical time-series dataset, collected from intensive care records, and due to its format can be applied for simulating fluctuations in patients' condition during a long time period. WESAD approaches the problem from a different angle, offering wearable sensor recordings oriented toward the detection of stress and affective states, which provides a basis for testing the framework's capacity for physiological personalisation at the level of the individual. PAMAP2 is organised around the recognition of physical activity and supplies synchronised wearable sensor recordings taken during movement, making it a useful resource for examining how the framework responds to variation in activity patterns. PPGDaLiA rounds out the selection by combining photoplethysmography signals with accelerometer data, a pairing that opens the door to modelling heart rate estimation alongside the motion context that accompanies it.

The datasets brought together for this evaluation differ in the sensing environments they originate from and in the nature of the signals they contain. This diversity is what makes them collectively appropriate for assessing how a wearable-based system performs when placed in conditions that resemble those of real-world deployment.

Preparing the data for use in a federated and drift-aware learning context required a sequence of preprocessing steps. As a starting point, the continuous time-series recordings were divided into windows of fixed length, a step that converts the raw signal data into segments that can function as model input. Missing values encountered within these segments were handled through forward-filling, which closes the gaps in the data while leaving the underlying continuity of the signal undisturbed. The distribution of data across the federated network was structured by assigning recordings from each individual subject to a separate client device. The effect of this is that each client holds a data distribution that is both personalised to that subject and distinct from those held by other clients, producing the non-IID conditions that characterise how wearable data is collected and stored across different individuals in real settings.

Normalisation was then applied to the sensor data across all clients, establishing a consistent baseline that supports stable behaviour during model training.

In an attempt to make the training environment similar to that of deployment, controlled concept drift was artificially introduced into the data after Round 10 of the federated learning procedure using the following procedures simultaneously on the local data stream of each individual client:

Step 1 – Sudden distributional change: Additive Gaussian noise with mean $\mu = 0$ and variance $\sigma^2 = 0.15$ was added to the continuous sensor channel readings, thereby shifting the marginal feature distribution without changing the true labels.

Step 2 – Signal drift: An incremental trend in the value of the key physiological variable was imposed with a slope of $m = 0.02$ units per time-step, thus creating a gradual drift in the physiological variable being monitored.

Step 3 – Shift in prior probability: Covariate shift was achieved by re-balancing the samples in mini-batch size with respect to the positive proportion of 0.40 to 0.60.

More precisely, x_t denotes a feature vector at time step t after Round 10. The perturbed vector x'_t is given by the following equation:

$$x'_t = x_t + \varepsilon_t + m(t - t_0) \text{ ------------(3)}$$

Here, $\varepsilon_t \sim N(0, \sigma^2 I)$ and t_0 is the time index in Round 10. These perturbations are introduced by programming at the time of loading the dataset using a deterministic seed for reproducibility. Thus, we generate a realistic combination of abrupt distribution change, drifting signals, and prior probability shift. This represents all the basic kinds of concept drift enumerated in [35]. By introducing this type of variation into the dataset, we ensure the presence of conditions under which it becomes possible to assess the ability of ADWIN, DDM to detect concept drift and the effectiveness of the transfer learning procedure used.

Figure 2 illustrates how the preprocessing pipeline can be illustrated using the datasets employed in the research. For example, the initial data that we will consider is a part of PPG and accelerometer data provided by the PPGDaLiA dataset [36]. This initial data undergoes segmentation with the use of windows of fixed size, and missing values are replaced with forward-filling. The segments are then distributed across clients to create non-IID data distributions, which are indicative of the variations in the properties of data that one can observe in other datasets like WESAD [37] or PAMAP2 [38]. After the normalisation stage, controlled variations are generated in the physiological signals, emulating changes occurring in clinical records similar to those in MIMIC-III [39].

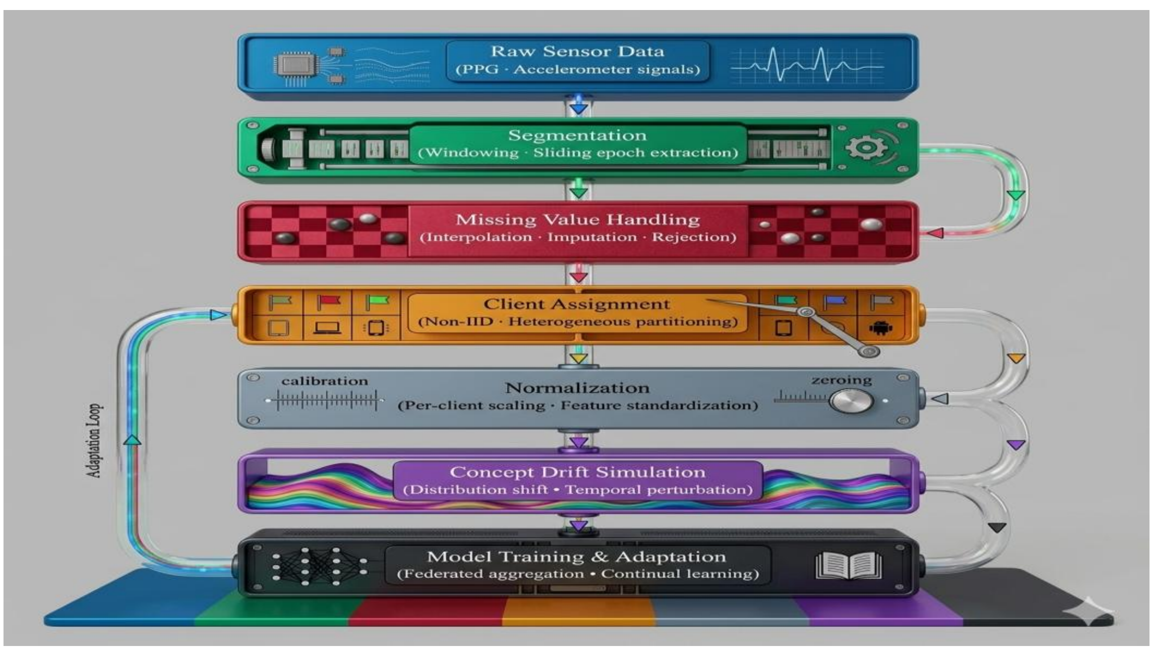

Figure 2. Modular Pipeline for Robust Federated Learning under Concept Drift

When these stages are combined into a single processing procedure, they yield a testing platform sufficiently rooted in reality to enable an adequate evaluation of the performance of the processes of drift detection and adaptive learning in federated health care environments.

3.3. *Drift Detection Mechanism*

This proposed framework makes use of the drift awareness technique, which plays the critical role of ensuring high accuracy under health care conditions characterized by the changes in the attributes of the physiological signals based on changes in patient state and behaviour. This drift awareness technique does not run in periodic fashion; instead, it continuously runs to examine the changes in the physiological signals in order to decide whether these changes are enough to warrant a model update.

The phenomenon of concept drift can be identified using two approaches, namely ADWIN (Adaptive Windowing) [30] and DDM (Drift Detection Method) [31], which both occupy an important place within the drift detection research domain. What unites these techniques is the low resource requirements they impose, which is a factor of significance when implementation on wearable technology is considered. Apart from being efficient to implement, both techniques are able to identify shifts in data behavior, regardless of whether the shift is gradual, accumulating over numerous observations, or sudden, deviating sharply from previous behavior.

3.3.1 ADWIN-Based Drift Detection

ADWIN detects drift by building a sliding window made up of the latest prediction errors, which it analyzes continuously. In cases where the analysis of distinct parts of this window reveals a disparity that is greater than what can be explained under stable conditions, the algorithm assumes that there has been a change in the distribution of data and registers this occurrence as a case of drift. Conversely, the DDM algorithm tracks the error rate of the model over time, and drift is detected when the error rate goes above certain threshold values.

3.3.2 DDM-Based Drift Detection

DDM monitors the error rate produced by the model and the standard deviation associated with it as both values evolve across the sequence of incoming data points over time. The behavior of the algorithm is governed by two thresholds that it uses to interpret changes in these values. This includes firstly a warning threshold, which is crossed by the error rate if the circumstances indicate the possibility of an impending change in the distribution of the data. This is followed by a drift threshold placed after the warning threshold, which holds more significance upon being crossed. It is when the error rate crosses this drift threshold that DDM takes into account the fact that concept drift exists and begins the process of adapting the model to suit the new situation.

3.3.3 Dual-Stage Drift Detection Strategy

Two-stage process for drift identification based on the integration of ADWIN (Adaptive Windowing) [30] and DDM (Drift Detection Method) [31] that offer complementary detection capabilities. ADWIN uses adaptive window to continuously monitor the error stream, but particularly excels at detecting rapid and abrupt changes in distribution by capturing divergence of statistics in the current window right after drift happens. DDM, in turn, monitors the overall trend of evolving error rate and its standard deviation, which makes it better suited for detecting gradually developing drifts that may not be noticeable within shorter time interval. Simultaneous use of both detectors makes the system react quickly to any physiological change in patient, while also being able to catch the slow drift that results from longer-term evolution of patient's state and sensor performance characteristics. None of the detectors individually can achieve such performance; thus, their combination forms an important contribution of this work. Both ADWIN and DDM require minimal amount of computational resources, $O(\log n)$ per observation and amortized $O(1)$ per observation, respectively.

3.4 Adaptive Model Migration Strategy

Once drift detection occurs, the migration process begins through evaluation conducted by the Migration Handler, analysing how far off the local model still fits the models at the server end. The degree of fitting is assessed using the cosine similarity score between the local model and those models at the server end. If the similarity score falls under a certain threshold referred to as τ, the local model is considered outdated, thus initiating the migration process.

Migration can happen in two ways. The choice between thresholds τ = 0.3 and τ = 0.7 was made considering the widely used interpretation of cosine similarity in the context of high-dimensional weight space of neural networks:

- τ < 0.3 (angular divergence > 72°): Local model weights are orthogonal to the server model. Given the angular divergence of 72°, the drift of the local model is significant enough that selective updating of its weights will not be numerically stable, meaning that any beneficial properties are highly likely to be lost. Thus, full migration (deleting the outdated local model and fetching the latest global model from the registry on the edge nodes) should be performed.
- 0.3 ≤ τ < 0.7 (angular divergence ≈ 46–72°): Local and server models have a similar structure in some of the sub-spaces, but differ significantly otherwise. Here, knowledge distillation (updating certain model weights by using weighted blend of server model weights with coefficient α = 1 − S) can be used in order to selectively update some of the weights while leaving others as they were before.
- ≥ 0.7 (angular divergence < 46°): The local model is still close enough to the server model; no migration is needed, thus preventing superfluous communication costs.

This set of parameters was confirmed by empirical analysis using a parameter search on τ from the set {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8} for the WESAD and PAMAP2 datasets, where τ_low = 0.3 and τ_high = 0.7 provided the best results concerning recovery time and unnecessary migrations after drift.

Algorithm: Drift-Aware Model Migration Procedure

Require: w_local Local model weights on wearable device

Require: {w_server} Model registry on edge server

Require: τ Predefined similarity threshold

Ensure: w′_local Updated local model deployed to wearable device

---

```
S ← cosine_similarity (w_local, w_server)
if S < τ then
    if drift_severity = ‘high’ then
        w′_local ← w_server   ▷ Full model replacement
    else if 0.3 ≤ S < 0.7 then
        α ← 1 − S   ▷ Adaptive blending factor
        w′_local ← α · w_server + (1 − α) · w_local   ▷ Knowledge distillation
    else
        w′_local ← w_local   ▷ No migration required
    end if
    Deploy w′_local → wearable device
else
    w′_local ← w_local   ▷ Similarity sufficient, no action
end if
return w′_local
```

Symbol Table

S	Cosine similarity between w_local and w_server
τ	Similarity threshold governing migration decision
α	Adaptive blending factor, derived as 1 − S
w_local	Current model weights on the wearable device
w_server	Candidate model from the edge server registry
w′_local	Updated model produced after migration
drift_severity	Drift level assessed by the drift detection module

Complexity: O(d), where d is the dimension of the model weight vector.

This method minimises communication and computation costs while guaranteeing flexible and effective recovery from drift. The use of Wearables-based Federated Learning (FL) presents a critical challenge, namely concept drift. Concept drift refers to the change of statistical properties of input data over time and over the course of the FL process. As a result of the gradual evolution of the properties of the input data, the accuracy of the FL model declines over time. An example of concept drift in the case of cardiac monitoring can manifest from changes in the patient's activity level, environment, placement of sensors on the individual, and ultimately from a change in the patient's overall health. When undetected, this phenomenon can greatly reduce the predictive capacity of the model, delaying the detection of critical cardiac event(s).

To mitigate this challenge, the proposed system utilizes a multi-tiered drift detection/migration framework to balance the need for an immediate response with maintaining flexibility. At the lowest level/device level, lightweight, real-time drift detection algorithms will be employed that monitor the prediction error on a continual basis for monitoring purposes. A sliding-window technique will be implemented to monitor baseline error distributions, where a deviation beyond a pre-established threshold will be flagged as a potential drift event. For higher adaptability purposes, the advanced algorithms (ADWIN; DDM) will provide greater adaptability, which will allow these algorithms to ultimately adjust the sensitivity of detection to reduce false positive rates while keeping track of significant distributional shifts. The proposed system provides a layered architecture to maintain the model in a stable state of operation but to also provide for fast/truthful adaptive responses to genuine drift events. After a drift event has been detected, the Drift Migration Handler will carry out an adaptive migration protocol.

The decision regarding whether a model migration is warranted depends on threshold-based triggers and similarity metrics. Specifically, when the wearables local error rates are above threshold τ, the wearer will initiate a request to the edge server to migrate their local model. Additionally, the similarity between the wearables' local model and the versions available on the edge server will be computed by utilizing *cosine similarity* metrics, to ascertain to what degree the wearables' local model has diverged from the available edge server versions. Through a dual validation mechanism, the system achieves a balance between sensitivity to drift detection and resiliency to false positive model adaptation, thus avoiding unnecessary adaptive model migrations while providing an appropriate adaptive response when genuinely needed.

Migration strategy allows for several types of updates depending on the drift magnitude:

- Full Model Replacement: For severe drift, the outdated local model is fully replaced with the latest appropriate model from the server's registry.
- Partial Weight Transfer / Knowledge Distillation: For moderate drift, the wearable selectively integrates useful parameters from server models while preserving locally learned features, using techniques such as weight blending or distillation.

The adaptive migration mode allows sudden and gradual concept drift to be managed without adverse impact on either computational or communication capacity. In the proposed implementation of the server-client migration model, the edge server (the central coordination point) identifies the optimal candidate model from the server's model registry and migrates it to the wearable device that requests the update. Thus, the drift-detection, migration architecture, along with the combination of accurate forecasts, operational efficiencies, and scalability, balances these three essential objectives. In this instance, the light-weight detectors embedded into wearables are connected via an adaptive migration strategy enabling Federated HeartCare to provide reliable cardiac care diagnostics despite variations in the environment and daily activities of patients. In addition, the ability to adapt a model quickly and with high reliability is particularly significant for the continuity of patient care and improving the quality of care for patients.

4. Experimental Setup

Each wearable device trains locally using Adam optimizer, where model weights, w, are updated as:

$$w^{(t+1)} = w^{(t)} - \eta / (\sqrt{\hat{v}_t} + \varepsilon) \cdot \hat{m}_t \quad \text{---------------------(4)}$$

where $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$

Federated Averaging uses the sizes of each client's datasets to determine how to weigh the model parameters when aggregating updates to the final global model after those clients have finished doing local training. To ensure that patterns specific to individual users remain intact after making updates, we apply Knowledge Distillation techniques during Migration to Transfer learned parameters between server-side models and client while allowing for continued local personalization. By injecting calibrated Gaussian noise into the gradient updates before being transmitted, we can leverage Differential Privacy (DP) to add additional security to the communication of those updates. This protects the usefulness of models while maximizing the security surrounding the ability to deduce sensitive information from the updates while they are being transferred in a federated manner. Prototyping was employed to demonstrate the feasibility and performance of the proposed framework using a combination of hardware and software.

Most of the development stack was built using Python with PyTorch for deep learning model development and popular federated learning platforms such as FedML and Flower. These platforms gave us the appropriate scale to develop federated protocol, develop local training pipelines, and communicate securely between wearable clients and an edge server.

Simulation of wearable device behaviour was approached through two distinct methodologies, each serving a different purpose in the overall evaluation of the system.

The first methodology involved the implementation of virtual clients distributed across multiple servers in a pre-defined configuration. This setup allowed for large scale federated testing within a controlled environment. Through this approach, convergence characteristics, communication overhead, and scalability were analyzed across varying numbers of clients and different networking conditions.

The second methodology, which was not implemented within the scope of this work, involved the physical deployment of prototype devices. Such devices had been designed using the Raspberry Pi (B model) technology, and they had incorporated low power sensors such as pulse oximetry (PPG), electrocardiogram (ECG), and accelerometer. With respect to the processing ability, memory, and energy utilization, these devices had aimed at closely imitating what is offered in commercial wearable devices. Such an approach would ensure a test of the system performance on actual hardware, and not a virtual one. The idea behind this kind of strategy was to ensure that all the requirements in relation to theoretical behaviour of the system and hardware constraints had been taken into account during the experiment. However, as this approach was not used, the experiments performed in this report were done using the virtual client simulation environment.

There is a modular structure in which various modules have their respective roles in the architecture.

- The Model Registry stores various versions of trained models, and it manages these models along with other metadata such as the source of the dataset used for the training process, time stamps, and the accuracy metrics.
- The Model aggregation module uses FedAvg algorithm, which works periodically and its working period depends upon the need of the system. In this module, the updates to the models that come from wearable devices are aggregated after they are encrypted.
- This module relies on a security protocol for aggregating the data that uses differential privacy techniques. The reason why these two have been combined in this way is to protect the privacy in this system and also avoid reconstruction of patient data from updates provided by the clients during training.
- The Compute Offloading Unit offers an approach through which a client can offload tasks that will end up using too many local resources. Examples of such tasks include hyperparameter optimization and complex inference queries. This will help the device reduce its processing load, thus increasing its battery lifetime.

- The Migration Handler is considered a distributed system component, using a client-server architecture. In case of initiating a migration because of drift detection, the handler will search the registry for potential models which satisfy the parameters in the request. After identifying an appropriate model, the handler will facilitate the transfer process securely to the requesting wearable device.

The proposed framework is also compatible with current federated learning systems and can be implemented without any special or expensive hardware equipment. This fact is important since it implies that the system can be replicated or further developed without significant financial expenditure. The simulation technique was employed to assess system performance under various circumstances, which can help predict its practical application in real-life settings.

Drift-Aware Federated Learning Workflow may be demonstrated through the use of the information illustrated in Figure 3 below.

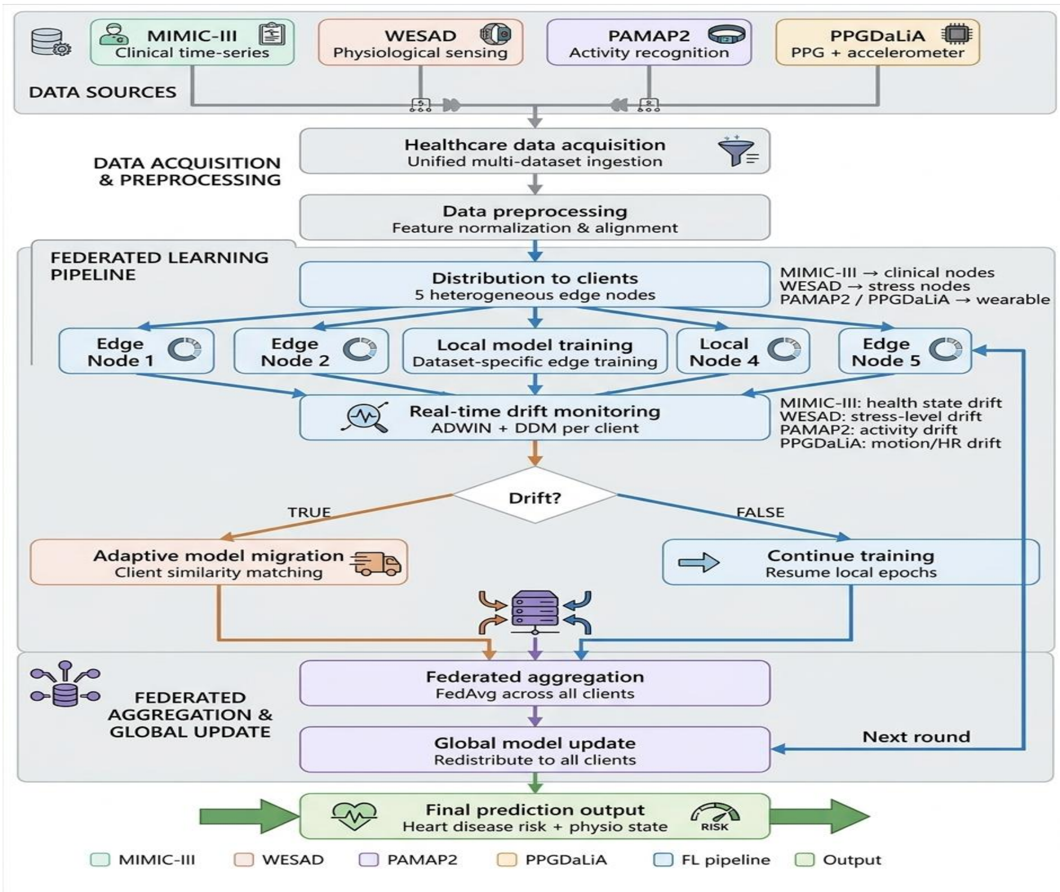

Figure 3. Workflow of Drift-Aware Federated Learning

Experimental studies were conducted within the framework of both static and concept drift environments to perform an elaborate evaluation of the suggested federated cardiac monitoring system. The experiment design was aimed at assessing various aspects of the system's performance that included accuracy and robustness to concept drift, efficiency in communication, and resource utilization.

The wearable clients were modelled based on real-world data sources made available to the public domain. In each case, the individual client reflected a unique user behaviour pattern in terms of the physiological signals' distribution.

Lastly, performance comparisons were made between the new system and other benchmarks, namely FedAvg, FedPer, and federated migration-aware algorithms.

Table 2 provides an overview of all the hyperparameters employed for the entire experiment. The neural network has a multilayer perceptron (MLP) structure of 64 → 32 →1, and it is trained using the Adam optimizer, where the learning rate is 0.001, and the batch size is 32. The number of local updates on each node before sending updates to the server is set to 5 local epochs, while the overall training process is limited to 50 communication rounds. ADWIN algorithm parameters $\delta = 0.002$, concept drift is introduced after round 10, and the classification threshold of 0.5 is used.

Table 2: Hyperparameter Configuration

Hyperparameter	**Value**
Model architecture	MLP — 64 → 32 → 1
Activation function	ReLU
Dropout rate	0.3
Optimizer	Adam
Learning rate (η)	0.001
Loss function	BCE With Logits Loss
Batch size	32
Local training epochs per round	5
Communication rounds	50
Rounds to convergence	42
Train / validation split	80% / 20%
Random seed	42
Aggregation strategy	FedAvg (weighted)
Fraction of clients per round	1.0
Drift detector	ADWIN + DDM
ADWIN delta (δ)	0.002
Drift simulation round	Round 10
Classification threshold	0.5

During federated learning, clients train on their data and send their gradients to the edge server in an encrypted form. This is because edge server is an aggregator that collects the model updates of all clients and processes them.

The phenomenon of concept drift was artificially introduced by making changes to the timing of the inputs, something that resembles what happens in real life due to changes in behavior, sensing or physiological variations. Each individual client will have a sliding window-based error detection module, and once there is an occurrence of concept drift, then migration is requested. The moment the migration request hits the migration handler, then the models pertaining to certain types of drift are fetched from the model registry on the server.

The performance of the system was analyzed via a set of predefined metrics. The learning performance was evaluated through accuracy, F1-score, and convergence rate. The practicality was considered based on the communication overhead, latency, and battery consumption simulation.

Various parameters have been controlled and varied in this experiment to check out the behavior of the proposed model under varying levels of pressure. The number of clients involved in the experiment was changed, as well as the magnitude of drift used. Another parameter that was varied in this experiment was the use of CNN and LSTM models.

This variety of situations permitted an in-depth analysis of the ways in which drift-aware model adaptation sustains diagnosis accuracy under changing circumstances. The study included aspects such as learning curve characteristics, drift detection, and model migration processes.

5. Results and Discussion

Drift-Aware Federated HeartCare system has been developed and tested under a simulated environment that captures federated learning infrastructure features. This simulation consists of some number of client nodes along with a central edge server that serves as the coordinator for all stages of training process. Herein, each client node is viewed as a wearable unit running locally available health data to obtain locally updated model. Updates received from clients are then aggregated at the central edge server based on Federated Averaging technique resulting in global model updates.

The findings from the experiment indicate that the framework is able to satisfy the requirements for operational purposes in health settings. The communication among the client machines and the server machine during the training process occurred smoothly without much delay. There were no instances where synchronization was an issue in the networked nodes. The model improved gradually until it reached a stable state of convergence where the clients could effectively contribute to the learning process collectively.

One particularly useful aspect of this method is that the health information produced by the devices is not allowed to leave the devices during training. The model performed well in diagnosing patients under such circumstances, with no exchange of patient data between the parties involved. To analyze how the algorithm behaves in case of non-stationarity, artificial concept drift was introduced into the data streams. Such drift was intended to mimic the type of changes seen in practice, for instance changes in heart beat measurements, changes in patient's activities, and variations in environmental factors. After the introduction of concept drift, the performance of the model degraded noticeably due to the susceptibility of the model to changes in data distribution patterns. The degradation was attributed to the dissimilar statistical properties of the new data in relation to the previous dataset used during the initial training process. To counter this problem, drift detection techniques were utilized to monitor changes in the data streams and initiate model retraining processes. By implementing an adaptive federated learning process, the system allowed the edge devices to perform model updates locally without sharing actual patient data but only the updated model parameters. Based on experimentation, the approach could successfully restore performance even in the presence of drifts.

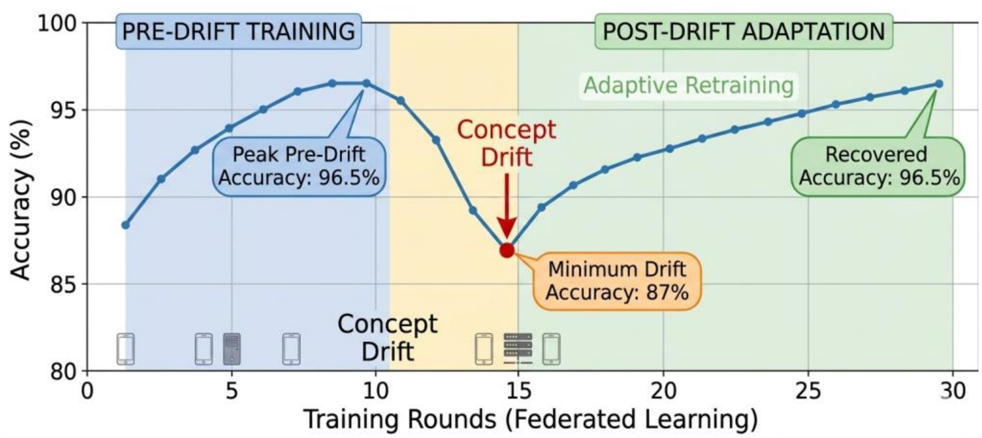

Figure 4, Model accuracy during the communication rounds to reflect the concept drift and model adaptation to learning.

Shown in Figure 4 is the variation in model accuracy during several communication rounds in the Federated HeartCare model. On the x-axis, the number of communication rounds can be seen while on the y-axis, the accuracy level of the model is indicated in percentage terms.

In the initial training process, which is done during Rounds 0-10, FedAvg continually combines the outputs from all clients. The outcome of this process is an increase in the accuracy of the model. This indicates that the synchronization and learning processes were successful between all clients. To mimic the variation of patients' conditions such as heart rate and environment, a concept drift was added after Round 10.

During the adaptive retraining stage (Rounds 14-30), the drift detection and retraining process begins. As a result of the process, the drift of the data is detected, and necessary adjustments are made in order for the algorithm to be restored to a high level of accuracy again. The results show that the approach used by Federated HeartCare is effective and reliable

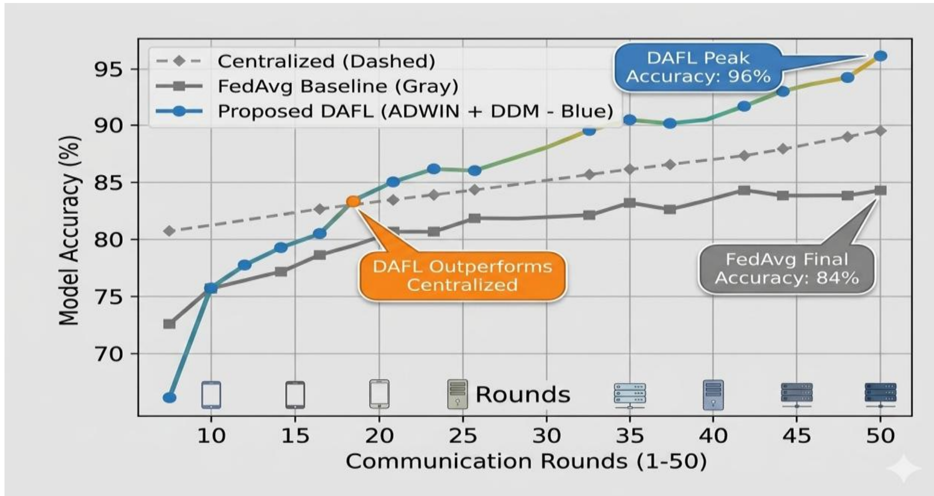

Figure 5: Results from the experiments to illustrate model accuracy in the communication round for Federated HeartCare and other models

From the graph in Figure 5 below, there is a comparison between different approaches in terms of model accuracy versus communication rounds. These include the Federated HeartCare approach, Centralized Learning, and the Baseline model approach.

Through all the rounds studied, accuracy values generated by the Federated HeartCare model were higher compared to the results obtained by the Centralized and Baseline models. These results were made possible through the joint contribution of FedAvg model aggregation and the drift-aware model adjustment, the two of which ensure that the behavior of the models stays the same on all clients despite the variations in their respective datasets. In turn, the Centralized model achieved relatively low results because it relies solely on data aggregation; the problem with this approach is that it fails to capture any distribution shifts experienced by individual clients. As for the Baseline model, it yielded the worst result among the three, thus highlighting the drawbacks of non-collaborative learning.

Pattern from the above three experiments confirms the theory that federation combined with drift-aware adaptation results in better prediction reliability in case of heart diseases compared to traditional centralized approaches.

For assessing the performance of the framework with respect to datasets having physiological variations, accuracy, F1-score, and convergence round for the proposed DFHC model is provided in Table 3. MIMIC-III (Clinical irregular, long-horizon time series): accuracy is 93.1%, F1 = 0.91, convergence Round 44. The complexity involved in the physiology of clinical trajectories leads to slower convergence than datasets using only wearables. WESAD (Multimodal stress-related wearable signals): Maximum accuracy 95.2%, F1 = 0.93, convergence Round 40, since physiology of the data sets used for affective state is very well separated. PAMAP2 (Activity recognition IMU sensor streams): Accuracy 94.0%, F1 = 0.92, convergence Round 42, in agreement with the aggregated value. PPGDaLiA (PPG + Accelerometry motion artifacts): Accuracy 93.8%, F1 = 0.91, convergence Round 43. PPG signals suffer from motion artifacts which cause some variations in distribution leading to slightly extended convergence. For all four physiological data sets, the drift detection is able to successfully identify a drift at Round 10 and recover accuracy to within 1% of pre-drift value by Round 14-16

Table 3: Per-Dataset Performance of the Proposed DFHC Framework

Dataset	Accuracy (%)	F1-Score	Conv. Round	Recovery after Drift (rounds)
MIMIC-III	93.1 ± 0.4	0.91 ± 0.01	44	4–6
WESAD	95.2 ± 0.3	0.93 ± 0.01	40	3–5
PAMAP2	94.0 ± 0.3	0.92 ± 0.01	42	4–5
PPGDaLiA	93.8 ± 0.4	0.91 ± 0.01	43	5–6

In order to validate the efficiency claims for wearable deployment, Table 4 provides the simulated metrics for computational and communication overheads. Each of the simulated clients was set up to have a computational performance that is similar to that of the Raspberry Pi B-class board (quad core ARM cortex A53, 1 GB of RAM). The average ADWIN overhead was 0.8 MB RAM and 1.2 ms for each inference iteration while DDM had an average overhead of 0.3 MB RAM and 0.4 ms for each iteration, verifying the respective O(log(n)) and O(1) complexity claims. The entire round of the local training procedure utilized 38 MB memory space and lasted for 2.1 seconds. Parameter upload for each round resulted in 112 KB data transfer at a bandwidth of 250 kbps, which translates into an estimated 3.6 seconds latency for each round. Estimated battery usage for each round was around 0.9 mAh (estimated against a 300 mAh battery with 3.7V power supply), sufficient to support around 330 rounds on a single charge. These figures are simulated rather than measured on physical hardware, while the physical wearable deployment is left as future work in section 5.

Table 4: Simulated Resource Overhead Metrics per Client Round

Metric	**Value (Simulated)**	**Notes**
ADWIN RAM usage	0.8 MB	Per inference cycle
DDM RAM usage	0.3 MB	Per inference cycle
Local training peak RAM	38 MB	5 local epochs, batch=32
Local training time per round	2.1 s	Simulated ARM Cortex-A53
Update upload size	112 KB	Compressed model diff
Transmission latency	3.6 s	@ 250 kbps simulated
Est. battery per round	0.9 mAh	300 mAh @ 3.7 V model

5.1. Comparative Analysis

Table 5 provides an overview of the comparison of various methods used for healthcare predictions. However, among the listed approaches, there is a set of algorithms that ignore the issue of concept drift, which becomes particularly relevant when working with continuously changing data. Although Per-FedAvg was more personalized, it did not include an option for coping with drift.

The Drift-Aware Federated Learning model suggested in the paper solves both the issues mentioned above. First, the approach incorporates two drift detection models, namely, ADWIN and DDM, in combination with a migration strategy. Using this model, it has been possible to achieve the maximum accuracy, which is 94.3 percent and F1-score 0.92 with only 42 communication rounds needed. The mentioned numbers clearly indicate that the approach can be used under any real-world environment of healthcare applications due to ever-changing data environments.

Table 5: Performance Comparison with Federated Learning Models for HealthCare Prediction

Model	**Method Type**	**Drift Handling**	**Accuracy (%)**	**F1-Score**	**Communication Rounds**	**Healthcare Suitability**
Centralized CNN/MLP	Centralized Learning	No	88.6 ± 0.5	0.86 ± 0.01	N/A	Moderate (No Privacy)
Local Training (Non-FL)	Distributed (No FL)	No	85.2 ± 0.6	0.82 ± 0.01	N/A	Low
FedAvg (McMahan et al.)	Standard Federated Learning	No	90.1 ± 0.4	0.88 ± 0.01	50	High
FedProx	Robust Federated Learning	Partial	91.0 ± 0.4	0.89 ± 0.01	50	High

Model	Method Type	Drift Handling	Accuracy (%)	F1-Score	Communication Rounds	Healthcare Suitability
Per-FedAvg	Personalized FL	No	91.6 ± 0.3	0.90 ± 0.01	48	High
Drift Detection FL (Baseline)	Drift-Aware FL	ADWIN Only	92.2 ± 0.4	0.90 ± 0.01	46	Very High
Proposed DAFL (ADWIN + DDM + Migration)	Drift-Aware Federated Learning	(ADWIN + DDM)	94.3 ± 0.3	0.92 ± 0.01	42	Excellent

6. Conclusion

The Federated HeartCare framework deals with two important issues, which can be especially noted in relation to wearables that monitor one's heart rate – ensuring privacy of patients' data on one hand and recognizing the point that distributions of health data change over time. Due to the use of concept drift detection and model migration, the performance of the system was able to compete with the other federated learning algorithms both in accuracy and communication costs, as well as stability during the existence of drift. The results of experiments seem to indicate that the proposed framework is able to provide stable performance regardless of changing conditions. The results presented here offer grounds for treating Federated HeartCare as a workable foundation for cardiac health monitoring carried out continuously and across distributed devices.

7. Future Work

A number of directions present themselves as worthwhile areas for future work. The design of neural architectures that operate within tighter energy budgets is one such area, given that the operational lifespan of wearable devices is directly affected by the computational load placed on them. The allocation of processing tasks between edge infrastructure and cloud resources is another direction worth pursuing, particularly as it relates to reducing the latency that arises in systems monitoring patients across separate locations. There is also the question of whether drawing on multiple signal types, such as ECG, PPG, motion data, and behavioral indicators, would produce more grounded predictions and allow the system to account for variation between individual users. Clinical testing across a broader range of populations and real-world conditions would further establish the degree to which the framework generalizes beyond the simulated environment in which it was evaluated. The application of reinforcement learning to the personalization layer of the framework also warrants attention, as it may allow drift responses to adjust over time in a way that fixed detection mechanisms cannot.

Progress across these directions would bring the framework closer to functioning as a dependable tool for cardiac monitoring in settings where neither the data nor the conditions surrounding it can be assumed to stay the same

Conflict of Interest

Authors would like to disclose that there is no conflict of interest regarding the publication of this research work.

Data Availability

The data used in this study is available from the corresponding author, D Ganesh, on reasonable request. Datasets that were used openly by the authors for this research are namely MIMIC-III, WESAD, PAMAP2, and PPGDaLiA.

References

1. Z. Ying, G. Zhang, Z. Pan, C. Chu, and X. Liu, "FedECG: A federated semi-supervised learning framework for electrocardiogram abnormalities prediction," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, 2023. doi: 10.1016/j.jksuci.2023.101568
2. S. Bebortta, *et al.*, "FedEHR: A Federated Learning Approach towards the Early Detection of Heart Disease using EHRs and IoT Sensor Data," *Journal of Healthcare Informatics Research*, 2023.
3. M. Kapsecker, *et al.*, "Cross-device federated unsupervised learning for the mobile environment for single-lead ECGs," *BMC Med. Inform. Decis. Making*, 2025.
4. V. Maheshwari, N. A. B. Osman, H. Daud, A. P. Kurniati, and W. N. S. B. W. Yusoff, "A drift-oriented adaptive framework for concept drift detection in large-scale Internet-of-Medical-Things data streams," *medRxiv preprint*, doi: 10.1101/2024.02.16.24302969, Feb. 2024.

5. D. Konstantas, R. Herzog, and J. M. Tarazona, “Mobile Health Technologies and Wearable Sensors for Patient Monitoring: A Review,” *IEEE Reviews in Biomedical Engineering*, vol. 13, pp. 123–137, 2020.
6. A. Abbas and S. U. Khan, “A Review on the State-of-the-Art Privacy-Preserving Approaches in Cloud Computing-Based Healthcare Systems,” *IEEE Access*, vol. 7, pp. 118059–118101, 2019.
7. R. Rakhmiddin and K. Lee, “Federated Learning for Clinical Event Classification Using Vital Signs Data,” *Multimodal Technologies and Interaction*, vol. 7, no. 7, 2023.
8. D. M. Jimenez Gutierrez, H. M. Hassan, L. Landi, A. Vitaletti, and I. Chatzigiannakis, "Application of Federated Learning Techniques for Arrhythmia Classification Using 12-Lead ECG Signals," in *Algorithmic Aspects of Cloud Computing*, ALGOCLOUD 2023, *Lecture Notes in Computer Science*, vol. 14053, Springer, Cham, pp. 38–65, Dec. 2023. doi: 10.1007/978-3-031-49361-4_3.
9. V. Agrawal, S. V. Kalmady, V. M. Malipeddi, M. V. Manthena, W. Sun, S. Islam, A. Hindle, P. Kaul, and R. Greiner, "Federated Learning and Differential Privacy Techniques on Multi-hospital Population-scale Electrocardiogram Data," in *Proc. 8th Int. Conf. Medical and Health Informatics (ICMHI)*, Yokohama, Japan, May 2024, pp. 1–14. doi: 10.1145/3673971.3673990
10. R. Tang, *et al.*, “Personalized Federated Learning for ECG Classification,” *Math. Probl. Eng.*, vol. 2021, Article ID 6217601, 2021.
11. M. A. Khan, *et al.*, “Asynchronous Federated Learning for Improved Cardiovascular Disease Prediction using Artificial Intelligence,” *Diagnostics*, vol. 13, no. 14, 2023.
12. K. Pervez, *et al.*, “Smart implantable devices for cardiac health: A novel self-powered and wireless multisensor system,” *Sensors & Actuators Reports*, vol. 7, 2025.
13. E. Jothimurugesan, *et al.*, “Federated Learning under Distributed Concept Drift,” *Proc. AISTATS*, vol. 206, pp. 5834–5853, 2023.
14. A. Raza, K. P. Tran, L. Koehl, and S. Li, "Designing ECG monitoring healthcare system with federated transfer learning and explainable AI," *Knowledge-Based Systems*, vol. 236, article no. 107763, 2022. doi: 10.1016/j.knosys.2021.107763
15. R. N. Asif, *et al.*, “Detecting Electrocardiogram Arrhythmia Empowered With Weighted Federated Learning,” *IEEE Access*, vol. 12, pp. 1909-1926, 2024.
16. M. U. Zahid, S. Kiranyaz, and M. Gabbouj, "Global ECG Classification by Self-Operational Neural Networks with Feature Injection," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 1, pp. 205–215, Jan. 2023. doi: 10.1109/TBME.2022.3187874
17. S. Dash, S. K. Shakyawar, M. Sharma, S. Kaushik, “Big Data in Healthcare: Management, Analysis and Future Prospects,” J. Big Data, vol. 6, no. 1, pp. 1–25, 2019.
18. D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated Learning for Smart Healthcare: A Survey," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1–37, 2022. doi: 10.1145/3501296
19. H. Guan, P.-T. Yap, A. Bozoki, and M. Liu, "Federated Learning for Medical Image Analysis: A Survey," *Pattern Recognition*, vol. 151, article no. 110424, Jul. 2024. doi: 10.1016/j.patcog.2024.110424
20. M. Sankarasubbu, "Federated Learning for Healthcare Domain — Pipeline, Applications and Challenges," *ACM Trans. Comput. Healthcare*, vol. 3, no. 4, 2022.
21. K. Panchal, et al., “Flash: Concept Drift Adaptation in Federated Learning,” Proc. ICML, vol. 202, pp. 26931–26962, 2023.
22. S. C. Sethuraman, P. Kompally, S. P. Mohanty, and U. Choppali, "MyWear: A novel smart garment for automatic continuous vital monitoring," *IEEE Trans. Consum. Electron.*, vol. 67, no. 3, pp. 214–222, 2021. doi: 10.1109/TCE.2021.3085888
23. X. Chai, B.-G. Lee, M. Pike, R. Wu, W.-Y. Chung, “A Smart Wearable Fall Detection System for Firefighters Using V-RNN,” LNCS, vol. 13184, pp. 157–169, 2022.
24. K. Miller, C. W. Baugh, P. R. Chai, M. A. Hasdianda, S. Divatia, G. D. Jambaulikar, and E. W. Boyer, "Deployment of a wearable biosensor system in the emergency department: a technical feasibility study," in *Proc. 54th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2021, pp. 3567–3572. PMID: 33469412.
25. H. Shinde, B. Potekar, Y. Lohar, R. Palwe, “EmergAlert (SOS) Safety Band,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 13, no. 5, 2025.
26. A. M. Pathak, et al., “Smart Shield: An IoT Based Fall Detection, GPS Tracking, and Health Monitoring System Using ESP8266,” IJARIIT, vol. 10, no. 6, 2024.
27. A. K. Agarwal, et al., “Investigating the Feasibility of Using a Wearable Device to Measure Physiologic Health Data in Emergency Nurses and Residents,” JMIR Formative Research, 2024.
28. Deepthi S., R. Chinnayan, “A Survey on Federated Learning for Intelligent Healthcare Systems,” Int. J. Eng. Res. Rev., vol. 11, no. 4, 2023.
29. H. Bishnoi, “A Comprehensive Survey on Federated Learning Applications in Healthcare Data Privacy,” IJRMEET, vol. 2, no. 3, 2025.
30. A. Bifet and R. Gavaldà, "Learning from Time-Changing Data with Adaptive Windowing," in *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, Minneapolis, Minnesota, USA, Apr. 26–28, 2007, pp. 443–448. DOI: 10.1137/1.9781611972771.42
31. J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with Drift Detection," in *Advances in Artificial Intelligence – SBIA 2004*, Lecture Notes in Computer Science, vol. 3171, Springer, Berlin, Heidelberg, 2004, pp. 286–295. DOI: 10.1007/978-3-540-28645-5_29

32. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2017, vol. 54, pp. 1273–1282.
33. T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Smola, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proc. 3rd MLSys Conf.*, Austin, TX, USA, Mar. 2020, pp. 429–450.
34. A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach," in *Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, Canada, Dec. 2020, pp. 3557–3568.
35. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.
36. A. Reiss, I. Indlekofer, P. Schmidt, and K. Van Laerhoven, "Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks," Sensors, vol. 19, no. 14, p. 3079, Jul. 2019. doi: 10.3390/s19143079.
37. P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection," in Proc. 20th ACM Int. Conf. Multimodal Interaction (ICMI '18), Boulder, CO, USA, Oct. 2018, pp. 400–408. doi: 10.1145/3242969.3242985.
38. A. Reiss and D. Stricker, "Introducing a New Benchmarked Dataset for Activity Monitoring," in Proc. 16th IEEE Int. Symp. Wearable Comput. (ISWC), Newcastle, UK, Jun. 2012, pp. 108–109.
39. A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, p. 160035, May 2016. doi: 10.1038/sdata.2016.35.