

Privacy-Preserving Federated Learning System for Heart Disease Prediction

¹Okokon, Kate Edidiong, ²Sako, D.J.S., ³Anireh, V. I. E.

Department of Computer Science,
Rivers State University, Port-Harcourt, Nigeria

¹okokon.kate@ust.edu.ng, ²dum.sako@ust.edu.ng, ³anireh.ike@ust.edu.ng

<https://doi.org/10.56201/rjpst.vol.9.no9.2026.pg32.44>

Abstract

Smart healthcare wearable devices generate physiological data that can support disease prediction and remote patient monitoring, but centralising such data increases privacy, governance, and patient-trust risks. This paper presents a privacy-preserving federated learning system for heart disease prediction. The system was implemented as a simulation-based Python system and combined data preprocessing, Dirichlet-based non-independent and identically distributed (non-IID) client partitioning, Federated Averaging, gradient clipping, and Gaussian noise addition and performance evaluation. The Cleveland Heart Disease dataset was used as wearable-representative healthcare data rather than live wearable data. Although the variables are relevant to wearable-assisted cardiovascular monitoring, the dataset was not collected directly from smart wearable devices. The evaluation was simulation-based and did not include physical wearable hardware, real edge devices, network latency, battery usage, memory constraints, or client dropout. Overall, the application interface provides presentation-level evidence that the proposed system is not only a static report output as it supports parameter selection, live process visibility, result inspection, visual evidence review, and prediction demonstration. This strengthens the practical demonstration of the simulation-based framework while preserving the reproducibility of the original experiment script and result files.

Keywords: Federated learning, Privacy Preservation, Smart Healthcare, Wearable Devices, Heart Disease, Prediction, Differential Privacy, Simulation.

1. Introduction

Smart healthcare wearable devices have expanded from basic activity trackers to health-monitoring systems that collect physiological measurements such as heart rate, blood oxygen level, electrocardiogram-derived indicators, blood pressure, sleep patterns, and activity signals. These measurements can support disease prediction, early warning, remote patient monitoring, and continuous care outside traditional clinical environments [1], [2]. As wearable devices become integrated into Internet of Medical Things (IoMT) systems, they produce data that are analytically valuable because they describe health status over time, but privacy-sensitive because the same signals may reveal disease status, lifestyle patterns, and individual physiological baselines.

Machine learning can extract useful patterns from wearable and clinical health data, but conventional centralised training requires data to be transferred to a central server. This creates privacy, security, ethical, and regulatory concerns because health records may expose diagnostic status, physiological trends, and other sensitive personal information [3], [4]. In

healthcare settings, such risks are not only technical. They affect patient trust, institutional compliance, and the willingness of patients to participate in digital monitoring. Federated learning addresses part of this problem by allowing multiple clients to train a shared model while keeping raw data at the local source [5]. Instead of sending patient records to a central server, each client trains locally and sends model updates for aggregation. Federated Averaging (FedAvg) is widely used for this purpose because it combines client updates into a global model through weighted averaging [5], [6]. However, federated learning alone does not remove privacy risk. Shared gradients or model parameters may still leak sensitive information through membership inference, attribute inference, or gradient reconstruction attacks [7], [8]. Differential privacy provides a formal mechanism for limiting the influence of individual records on model outputs or updates by adding calibrated noise [9], [10]. When combined with federated learning, differential privacy can reduce leakage from shared updates while retaining the data-locality benefit of distributed training. Its benefit depends on parameter choice. A small privacy budget gives stronger privacy but requires more noise, while a larger privacy budget may preserve accuracy but offers weaker privacy protection. The computational cost of clipping and perturbing gradients is also relevant for wearable and edge-healthcare settings. Healthcare wearable data also present a distribution challenge. Patient populations differ across devices, institutions, age groups, health profiles, and disease prevalence. These differences create non-IID client data, which can reduce federated model convergence and make evaluation under ideal balanced partitions unrealistic [6], [11]. A privacy-preserving healthcare model therefore needs evidence under heterogeneous client distributions as well as under different privacy budgets.

In this study, we developed a privacy-preserving federated learning system for heart disease prediction. The system is implemented as a simulation-based Python system and combined data preprocessing, Dirichlet-based non-independent and identically distributed (non-IID) client partitioning, Federated Averaging, gradient clipping, and Gaussian noise addition and performance evaluation. The Cleveland Heart Disease dataset was used as wearable-representative healthcare data rather than live wearable data. Although the variables are relevant to wearable-assisted cardiovascular monitoring, the dataset was not collected directly from smart wearable devices

2. Related Works

Federated learning was introduced as a distributed machine learning approach for training models from decentralised data. McMahan et al. [5] proposed FedAvg, where clients train local models and the server aggregates their parameters; their experiments showed that communication rounds could be reduced substantially compared with synchronised stochastic gradient descent. Kairouz et al. [6] later reviewed federated learning advances and identified data heterogeneity, system constraints, communication cost, and privacy leakage as persistent open problems. These studies establish the algorithmic foundation used in the present work, but they do not by themselves resolve the healthcare-specific question of how privacy noise and non-IID patient distributions affect disease prediction performance.

Healthcare is a major application domain for federated learning because clinical and wearable data are often distributed across institutions or devices and cannot always be pooled centrally. Rieke et al. [12] described federated learning as a route for collaborative medical modelling without direct data sharing, while Nguyen et al. [13] reviewed its use in smart healthcare and IoMT environments. Teo et al. [3] reviewed healthcare federated machine learning and reported that only a minority of studies incorporated privacy mechanisms, with many remaining proof-of-concept evaluations. This evidence is important for the present study because it shows that data locality alone should not be treated as a complete privacy solution.

Differential privacy is one of the main privacy-preserving techniques used in machine learning. Dwork et al. [9] formalised differential privacy as a mathematical privacy guarantee based on limiting the effect of a single record on the output distribution of a computation. Liu et al. [10] surveyed differential privacy for medical data analysis and showed its relevance to health data publishing and machine learning. In gradient-based federated learning, differential privacy is commonly implemented through gradient clipping and noise addition before updates are shared or aggregated [14]. This design reduces information leakage risk, but its practical value depends on whether the selected privacy budget preserves enough utility for the clinical prediction task.

Recent studies have combined federated learning and differential privacy in medical prediction tasks. Ahmed et al. [14] applied adaptive DP-FL to COVID-19 chest X-ray detection and reported 94.3% accuracy, with differential privacy introducing a measurable but limited accuracy cost. Shukla et al. [15] used FL with differential privacy for breast cancer diagnosis and reported 96.1% accuracy at $\epsilon = 1.9$. Barnawi et al. [16] proposed an FL and DP scheme for IoMT data protection. These works support the feasibility of DP-FL in healthcare, but their disease targets and data types differ from structured heart disease prediction with wearable-representative variables. They also do not remove the need to test several epsilon values under heterogeneous client distributions.

Data heterogeneity remains a central challenge in healthcare federated learning. Taha et al. [11] reviewed federated learning from a healthcare data perspective and emphasised the effect of non-IID distributions, data skew, and institutional variation. Thakur et al. [17] addressed federated learning over heterogeneous healthcare data views, showing that healthcare heterogeneity may affect both feature representation and model training. In wearable-health settings, heterogeneity may arise from differences in patient age, disease prevalence, device usage, measurement quality, and physiological baselines. These findings justify the use of Dirichlet-based client partitioning in this study rather than assuming that every federated client receives an equivalent class distribution.

Wearable healthcare prediction has also received growing attention. Alimbayeva et al. [18] developed a wearable ECG and machine learning system for cardiovascular monitoring and reported 0.926 accuracy for cardiovascular disease prediction. Pereira et al. [19] mapped machine learning applications in edge computing and wearable healthcare and highlighted resource constraints as a practical concern. These constraints are directly relevant to DP-FL because gradient clipping and noise addition can increase training time and processing burden. Khan et al. [20] reported that differential privacy in IoT-oriented FL studies can cause 5-15% accuracy degradation depending on privacy budget, while Truex et al. [21] reported additional computational overhead for hybrid privacy-preserving FL. These findings indicate that a wearable-health privacy framework should report efficiency results, not only predictive performance.

This study presents a privacy-preserving federated learning system for heart disease prediction using differential privacy. The system is implemented and evaluated as a simulation-based Python system and combines data preprocessing, Dirichlet-based non-independent and identically distributed (non-IID) client partitioning, Federated Averaging, gradient clipping, and Gaussian noise addition and performance evaluation.

3. Proposed System

The proposed framework consists of five main components: data preprocessing, non-IID client partitioning, federated learning, differential privacy, and performance evaluation. Figure 1 shows the architecture of the proposed privacy-preserving federated learning framework.

The data preprocessing module loads the heart disease dataset, handles missing values, encodes categorical variables, scales numerical variables, and produces the training and test matrices. The non-IID partitioning module distributes the training data across simulated federated clients using Dirichlet allocation. The federated learning module trains local client models and aggregates them using FedAvg. The differential privacy module applies gradient clipping and Gaussian noise addition during private training. The evaluation module computes prediction metrics, accuracy drop, training time, and overhead ratio.

The local prediction model was a compact multilayer perceptron with 22 input features, a 32-unit hidden layer with rectified linear activation, dropout of 0.10, a 16-unit hidden layer, and a single binary-output logit. Stochastic gradient descent was used for local client training with a learning rate of 0.05 and batch size of 16. The same model architecture, random seed, data partitions, communication-round settings, and local-epoch settings were used when comparing matched private and non-private configurations.

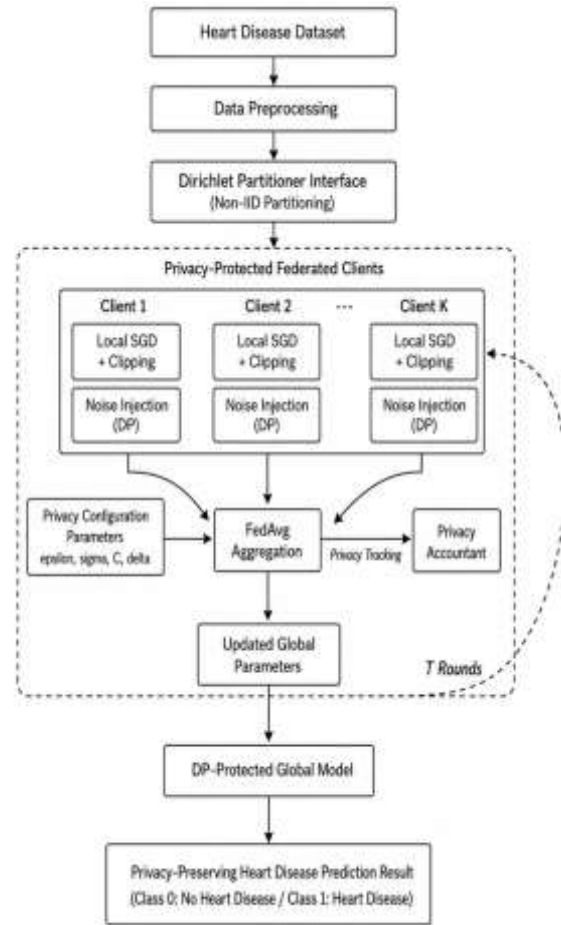


Figure 1: Proposed privacy-preserving federated learning architecture.

3.1 Non-IID Client Simulation

The study simulated heterogeneous client data using a Dirichlet distribution. The Dirichlet concentration parameter alpha controlled the degree of heterogeneity. Lower alpha values generated more uneven client class distributions, while higher alpha values produced more balanced distributions. The tested values were alpha = 0.1, 0.5, 1.0, 5.0, and 10.0. Client counts were 5, 10, and 20. This design represented healthcare scenarios where patients differ across wearable user groups, clinical environments, or edge-healthcare nodes.

3.2 Federated Learning and Differential Privacy

FedAvg was used as the aggregation method. For each communication round, the server distributed the current global model to clients. Each client trained locally on its assigned data partition and returned model updates. The server aggregated client updates using weighted averaging:

$$\theta_{t+1} = \sum_{k=1}^K \frac{n_k}{n} \theta_k^{t+1} \quad (2)$$

where θ_{t+1} is the updated global model, K is the number of clients, n_k is the number of records at client k , n is the total number of distributed training records, and θ_k^{t+1} is the updated local model from client k .

Differential privacy was implemented through gradient clipping and Gaussian noise addition during local training. Gradient clipping bounded the sensitivity of model updates, while Gaussian noise reduced the risk that a shared update would reveal information about an individual record. The clipping operation is represented as:

$$\tilde{g}_i = g_i \cdot \min\left(1, \frac{C}{\|g_i\|_2}\right) \quad (3)$$

where g_i is the gradient contribution and C is the clipping threshold. The private update is represented as:

$$g_{\text{private}} = \tilde{g} + \mathcal{N}(0, \sigma^2 C^2) \quad (4)$$

where $\mathcal{N}(0, \sigma^2 C^2)$ is Gaussian noise with noise scale determined by the privacy configuration. The experiment tested epsilon = 0.1, 0.5, 1.0, 2.0, 5.0, and 10.0 with delta = $1/n^2$. The clipping threshold was fixed at 1.0. Privacy was evaluated through these epsilon-controlled noise settings; direct attack experiments such as membership inference or gradient inversion were outside the implemented evaluation. The performance evaluation of this model has been well documented and reported in Sako & Okokon (2026).

3.3. System Modelling

The object-oriented design of the proposed system follows a modular architecture with defined class responsibilities and dependencies. The class structure uses a composition pattern in which a central orchestrator class manages component instances for data partitioning, federated training, differential privacy, and performance evaluation. Figure 2 represents the class relationships and interface specifications for the modular privacy-preserving federated learning architecture.

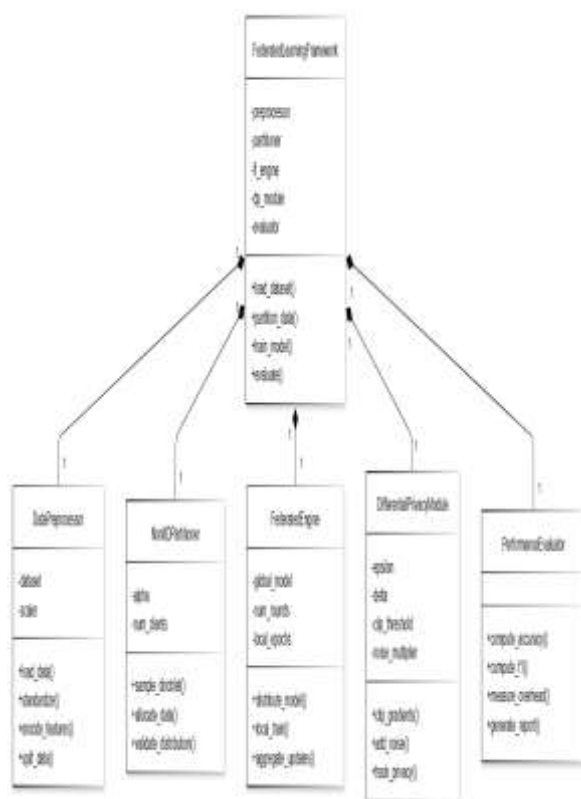


Figure 2: System Class Diagram Showing Modular Privacy-Preserving Federated Learning Architecture.

The architecture comprises six primary classes following the composition design pattern.

The **FederatedLearningFramework** class serves as the main orchestrator, maintaining composition relationships (1:1 multiplicity) with five component instances. Its four public methods coordinate component interactions: `load_dataset()` delegates to `DataPreprocessor`, `partition_data()` invokes `NonIIDPartitioner`, `train_model()` coordinates with `FederatedEngine`, and `evaluate()` triggers `PerformanceEvaluator`. The remaining five classes (**DataPreprocessor**, **NonIIDPartitioner**, **FederatedEngine**, **DifferentialPrivacyModule**, **PerformanceEvaluator**) encapsulate the functional responsibilities detailed in the sub-component tables of Section 3.4.3 (Tables 3.6 through 3.10), with their attributes and methods corresponding to the interfaces specified therein.

The sequence diagram in Figure 3 presents the streamlined sequence of interactions between the user, framework orchestrator, and component modules. It presents the runtime interactions between system components during the complete privacy-preserving federated learning workflow, showing the temporal ordering of method invocations from initial data loading through final performance evaluation.

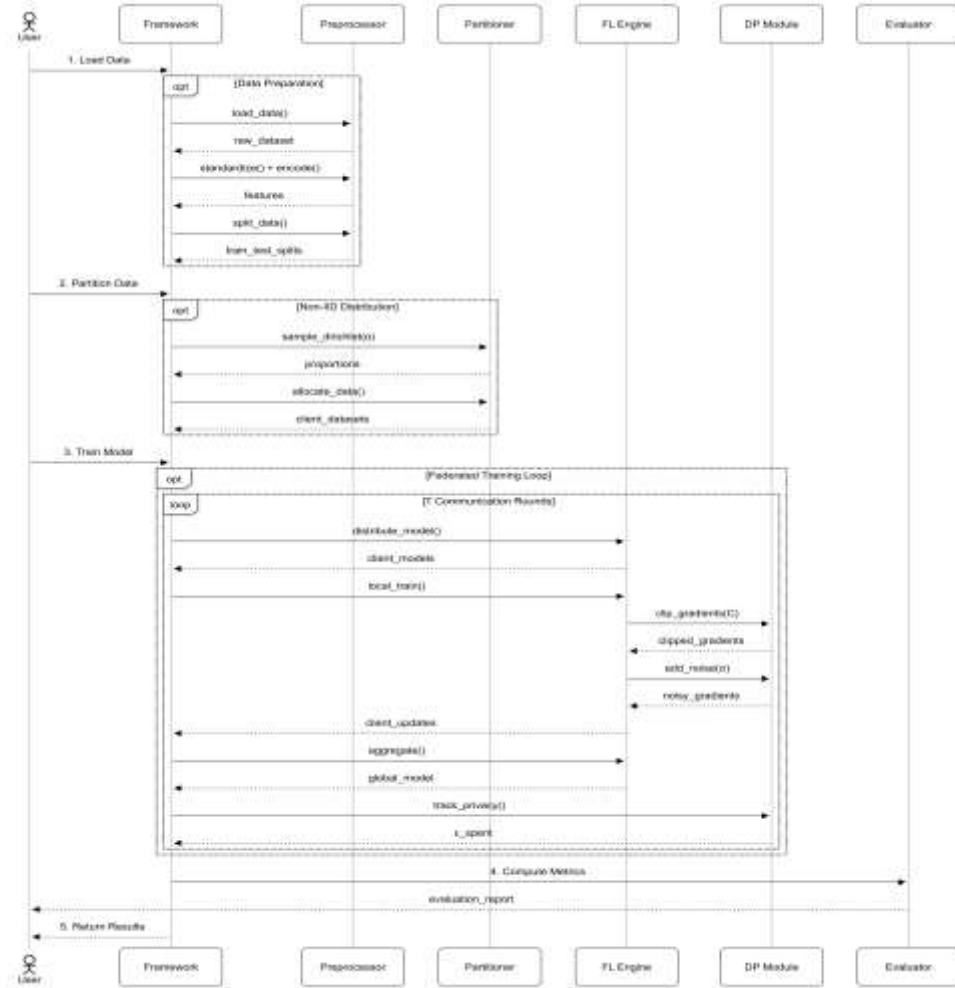


Figure 3: Streamlined Sequence Diagram of Privacy-Preserving Federated Learning Workflow.

The **FederatedLearningFramework** orchestrates the workflow.

- (1) The **User** loads data through the **DataPreprocessor**, which handles feature scaling, encoding, and stratified splitting.
- (2) The **NonIIDPartitioner** distributes training data across K clients using Dirichlet allocation with configurable heterogeneity.
- (3) The **FederatedEngine** executes T communication rounds, with local training at each client incorporating **DifferentialPrivacyModule** for gradient clipping and noise injection, followed by FedAvg aggregation and privacy budget tracking.
- (4) The **PerformanceEvaluator** computes accuracy, F1-score, and overhead metrics, and the **User** receives the evaluation report.

4. Experiment and Results

4.1 Dataset and experimental setup

The system was implemented as a simulation-based Python system because the study did not deploy physical smart wearable devices. The Cleveland Heart Disease dataset from the UCI Machine Learning Repository was used as the experimental data source. The simulation represents wearable or healthcare clients as distributed data partitions generated from the preprocessed training set. The dataset contained 303 records, consisting of 164 records for absence of heart disease and 139 records for presence of heart disease. After stratified splitting, 242 records were used for training and 61 records were used for testing. The 13 original input attributes were transformed into 22 model-ready features after imputation, standardisation, and categorical encoding. Table 1 presents a compact sample of records used in the experiment. The privacy failure probability was fixed at $\delta = 1/n^2 = 0.00001708$, where n represents the number of training records. The full experimental grid produced 420 completed runs. These comprised 60 non-private FedAvg baseline runs and 360 differentially private federated learning runs. The tested configurations covered all approved Chapter 3 settings: privacy budgets $\epsilon \in \{0.1, 0.5, 1.0, 2.0, 5.0, 10.0\}$, Dirichlet concentration values $\alpha \in \{0.1, 0.5, 1.0, 5.0, 10.0\}$, client counts $K \in \{5, 10, 20\}$, communication rounds $T \in \{20, 50\}$, and local epochs $E \in \{1, 5\}$. Each client trained locally on its own partition, while the central server aggregated model parameters using FedAvg. In the private configurations, local training included gradient clipping and Gaussian noise addition before client updates were aggregated. The data preprocessing component loaded the heart disease records, handled missing values, standardised numerical features, encoded categorical attributes, and performed stratified train-test splitting. The non-IID partitioning component distributed the training data across 5, 10, or 20 clients using the Dirichlet alpha values. The federated engine executed local training and server-side FedAvg aggregation. The differential privacy component applied the clipping threshold and calibrated Gaussian noise. The evaluation component computed accuracy, precision, recall, F1-score, confusion-matrix values, training time, accuracy drop, and overhead ratio.

Table 1: Sample records from the Cleveland Heart Disease dataset

age	sex	cp	trestbps	chol	thalach	oldpeak	raw target	binary target
63.0	1.0	1.0	145.0	233.0	150.0	2.3	0	0
67.0	1.0	4.0	160.0	286.0	108.0	1.5	2	1
67.0	1.0	4.0	120.0	229.0	129.0	2.6	1	1
37.0	1.0	3.0	130.0	250.0	187.0	3.5	0	0
41.0	0.0	2.0	130.0	204.0	172.0	1.4	0	0

4.2 Simulation Results

The simulation output consisted of one metrics dataset, one summary file, and ten figures. The metrics file contains the full 420-run experimental output. The summary file records the dataset summary, best baseline result, best DP-FL result, mean accuracy by mode, mean accuracy by epsilon, and mean overhead by epsilon. The prototype therefore demonstrates the complete experimental workflow through reproducible simulation outputs. It loads the dataset, produces non-IID client partitions, trains baseline and differentially private federated models, evaluates predictive performance, measures privacy-accuracy trade-offs, records computational overhead, and generates Seaborn-based visualisations for reporting. This interface does not replace the detailed experimental results reported in Sako & Okokong (2026); rather, it provides a practical way to demonstrate that the framework can be configured, executed,

monitored, and used for prediction without exposing the Python source code during presentation.

The application supports three run modes. The demo/custom mode executes a selected baseline, differentially private, or comparison run and prepares a trained model for prediction. The results mode loads the saved result files and figures used in the report. The full-grid regeneration mode runs the original script with resume support for the approved experimental grid. This structure separates fast demonstration from full experimental reproduction, which is important because the full differential privacy grid is computationally expensive.

Figure 4 shows the run configuration screen. The user can select the run mode, privacy mode, number of simulated clients, Dirichlet alpha value, privacy budget epsilon, communication rounds, and local epochs. The screen also displays system evidence and live processing messages so that execution steps are visible while the backend is running. Figure 5 shows the data and simulated clients screen. The screen summarises the Cleveland Heart Disease dataset and shows the client-level partitions produced by the Dirichlet allocation process.

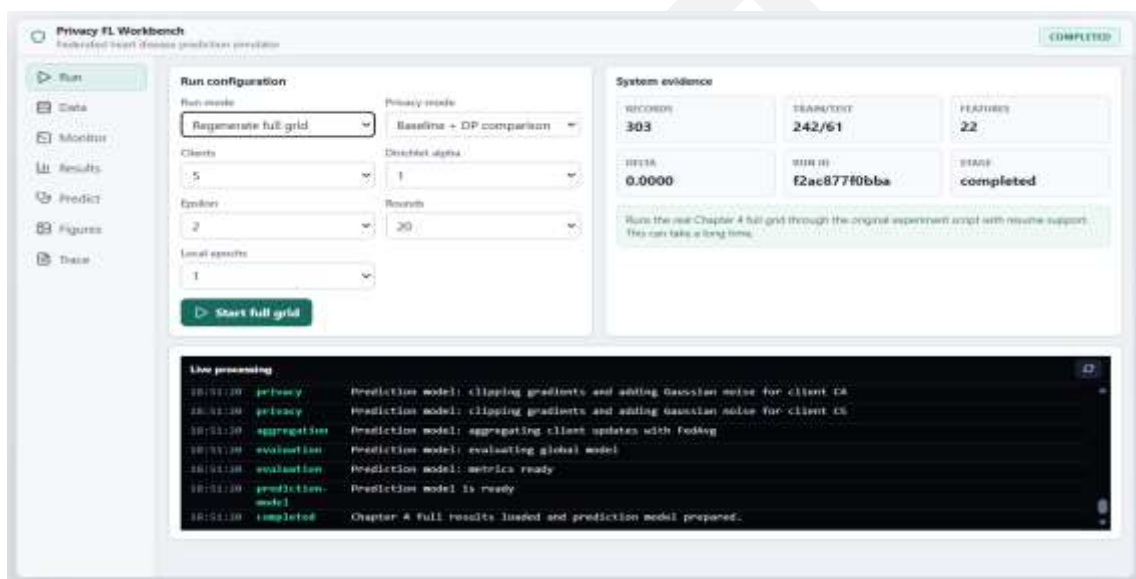


Figure 4: Prototype run configuration screen

Figure 6 presents the monitoring screen. The monitor presents run progress and backend processing messages, including dataset loading, preprocessing, client partitioning, local training, differential privacy processing, FedAvg aggregation, and evaluation. Figure 7 shows the results screen. It displays prediction metrics, training time, epsilon expenditure, accuracy drop, overhead ratio, and baseline-versus-DP comparison results. It also reports criteria checks such as whether the baseline accuracy exceeds the 80% target and whether moderate privacy accuracy drop remains below the defined threshold. Figure 8 presents the prediction screen. After a model is trained or prepared from the reported configuration, the user can select a sample health record and request a prediction. The prediction input is processed for the current request and is not stored by default, while the screen reports the predicted class and probability returned by the model. The figure gallery screen in Figure 9 displays the generated visual outputs, including class distribution, client distribution, privacy-budget trends, accuracy-drop trends, overhead, confusion matrices, and best-model comparison figures. Overall, the application interface provides presentation-level evidence that the proposed system is not only a static report output. It supports parameter selection, live process visibility, result inspection, visual evidence review, and prediction demonstration. This strengthens the practical

demonstration of the simulation-based framework while preserving the reproducibility of the original experiment script and result files.



Figure 5: Prototype dataset and simulated-client distribution screen.

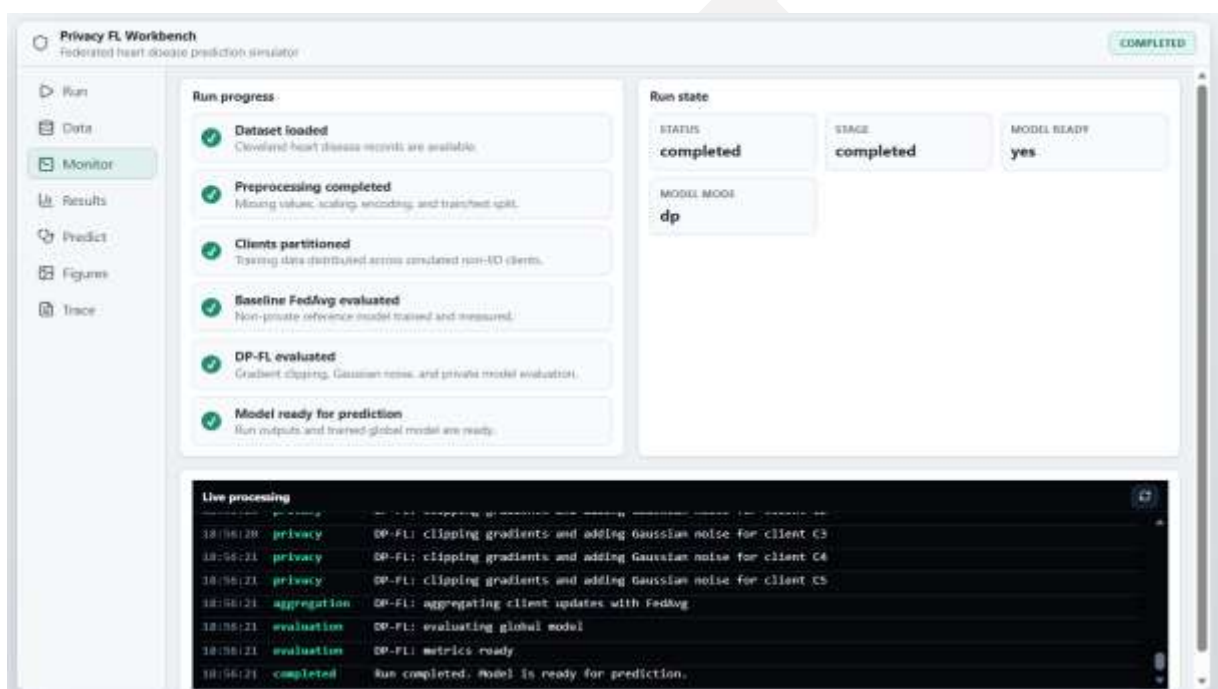


Figure 6: Live monitoring screen.

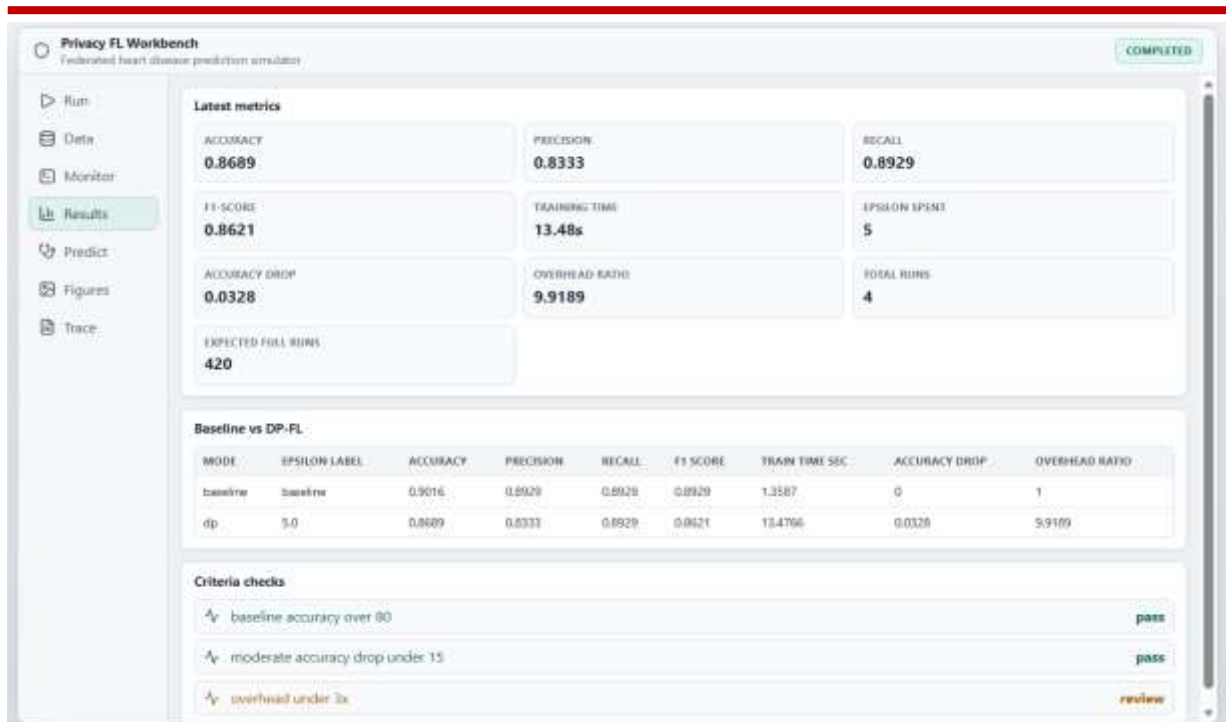


Figure 7: Prototype results and criteria screen

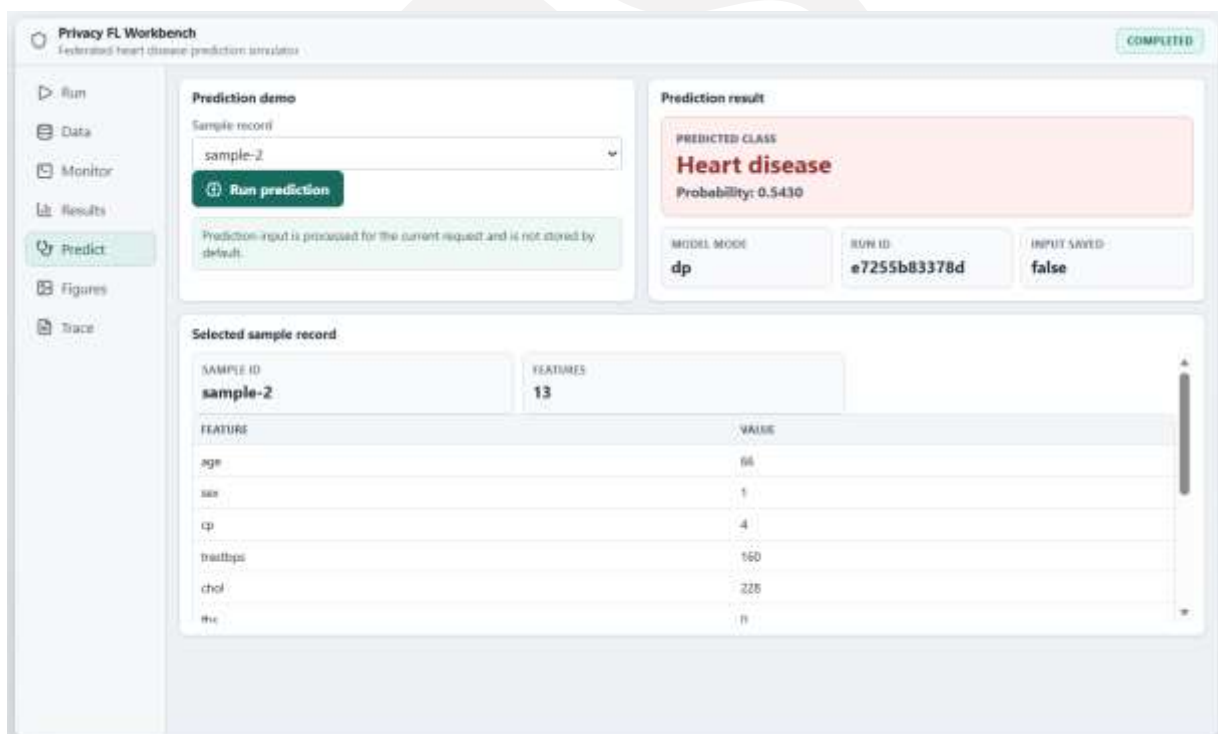


Figure 8: Prototype prediction screen

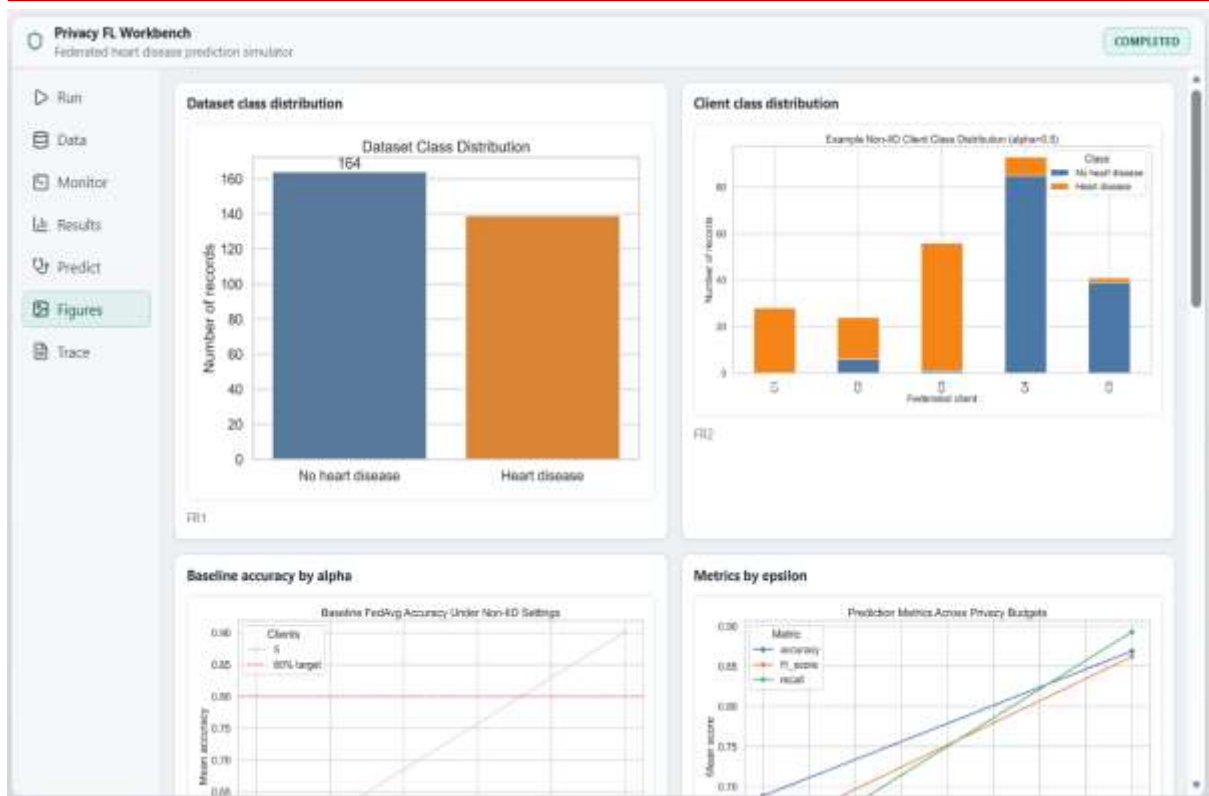


Figure 8: Prototype figure gallery screen.

5. Conclusion

This paper presented a privacy-preserving federated learning system for smart healthcare wearable devices using differential privacy. The system was developed for heart disease prediction using the Cleveland Heart Disease dataset as wearable-representative healthcare data rather than live wearable data. Although the variables are relevant to wearable-assisted cardiovascular monitoring, the dataset was not collected directly from smart wearable devices. It combined preprocessing, Dirichlet-based non-IID client partitioning, FedAvg training, gradient clipping, Gaussian noise addition, and performance evaluation in a simulation-based Python prototype. The evaluation was simulation-based and did not include physical wearable hardware, real edge devices, network latency, battery usage, memory constraints, or client dropout. Privacy was evaluated through epsilon-based differential privacy settings, but direct membership inference, attribute inference, or gradient reconstruction attacks were not implemented. The dataset size was small, so results should be interpreted as controlled experimental evidence rather than proof of clinical generalisability. Future work should evaluate the framework on larger and more recent wearable or IoMT datasets, deploy and test the approach on physical edge devices, include direct privacy attack evaluation, improve DP-SGD efficiency, examine model calibration and clinical thresholding, and combine differential privacy with secure aggregation or robust aggregation for broader federated learning protection.

References

- [1] S. R. Abbas, Z. Abbas, A. Zahir, and S. W. Lee, "Federated learning in smart healthcare: A comprehensive review on privacy, security, and predictive analytics with IoT integration," *Healthcare*, vol. 12, no. 24, 2587, 2024.
- [2] Q. Niu, H. Li, Y. Liu, Z. Qin, L. B. Zhang, J. Chen, and Z. Lyu, "Toward the Internet of Medical Things: Architecture, trends and challenges," *Mathematical Biosciences and Engineering*, vol. 21, no. 1, pp. 650-678, 2024.
- [3] Z. L. Teo, L. Jin, N. Liu, S. Li, D. Miao, X. Zhang, W. Y. Ng, T. F. Tan, D. M. Lee, K. J. Chua, J. Heng, Y. Liu, R. S. M. Goh, and D. S. W. Ting, "Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture," *Cell Reports Medicine*, vol. 5, no. 3, 101481, 2024.
- [4] P. Radanliev, "Privacy, ethics, transparency, and accountability in AI systems for wearable devices," *Frontiers in Digital Health*, vol. 7, 1431246, 2025.
- [5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273-1282, 2017.
- [6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1-210, 2021.
- [7] S. Pati, S. Kumar, A. Varma, B. Edwards, C. Lu, L. Qu, J. J. Wang, A. Lakshminarayanan, S. Wang, M. J. Sheller, K. Chang, P. Singh, D. L. Rubin, J. Kalpathy-Cramer, and S. Bakas, "Privacy preservation for federated learning in health care," *Patterns*, vol. 5, no. 7, 100974, 2024.
- [8] K. N. Kumar, C. K. Mohan, and L. R. Cenkeramaddi, "The impact of adversarial attacks on federated learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2672-2691, 2024.
- [9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Springer, pp. 265-284, 2006.
- [10] W. Liu, Y. Zhang, H. Yang, and Q. Meng, "A survey on differential privacy for medical data analysis," *Annals of Data Science*, 2023.
- [11] Z. K. Taha, C. T. Yaw, S. P. Koh, S. K. Tiong, K. Kadirgama, F. Benedict, J. D. Tan, and Y. A. L. Balasubramaniam, "A survey of federated learning from data perspective in the healthcare domain: Challenges, methods, and future directions," *IEEE Access*, vol. 11, pp. 45711-45735, 2023.
- [12] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, G. N. Galtier, B. Landman, K. Maier-Hein, et al., "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, 119, 2020.
- [13] 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.
- [14] R. Ahmed, P. K. R. Maddikunta, T. R. Gadekallu, N. K. Alshammari, and F. A. Hendaoui, "Efficient differential privacy enabled federated learning model for detecting COVID-19 disease using chest X-ray images," *Frontiers in Medicine*, vol. 11, 1409314, 2024.
- [15] S. Shukla, S. Rajkumar, A. Sinha, M. Esha, K. Elango, and V. Sampath, "Federated learning with differential privacy for breast cancer diagnosis enabling secure data sharing and model integrity," *Scientific Reports*, vol. 15, 95858, 2025.

- [16] A. Barnawi, P. Chhikara, R. Tekchandani, N. Kumar, and B. Alzahrani, "A differentially privacy assisted federated learning scheme to preserve data privacy for IoMT applications," *IEEE Transactions on Network and Service Management*, vol. 21, no. 4, pp. 4686-4700, 2024.
- [17] A. Thakur, S. Molaei, P. C. Nganjimi, F. Liu, A. Soltan, P. Schwab, K. Branson, and D. A. Clifton, "Knowledge abstraction and filtering based federated learning over heterogeneous data views in healthcare," *npj Digital Medicine*, vol. 7, 283, 2024.
- [18] Z. Alimbayeva, C. Alimbayev, K. Ozhikenov, N. Bayanbay, and A. Ozhikenova, "Wearable ECG device and machine learning for heart monitoring," *Sensors*, vol. 24, no. 14, 4543, 2024.
- [19] C. V. F. Pereira, E. M. de Oliveira, and A. D. de Souza, "Machine learning applied to edge computing and wearable devices for healthcare: Systematic mapping of the literature," *Sensors*, vol. 24, no. 19, 6322, 2024.
- [20] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 613-668, 2024.
- [21] S. Truex, N. Baracaldo, A. Anber, M. E. Gursoy, R. Jain, J. Li, H. Ludwig, and V. Gadepally, "A hybrid approach to privacy-preserving federated learning," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1-36, 2023.