International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

CardioFL: A Federated Learning Prototype for Heart Disease Prediction Using Simulated Hospital Clients

Author Details:

Guru Raghavendra D S[1*], Chaithanya M[1], Chandrashekarmurthy K M[1], Rakesh Gowda D S[1], Dr. Manjunath N[1]

[1] Department of Computer Science and Engineering, Vidya Vikas Institute of Engineering and Technology, Mysuru, Karnataka, India

Corresponding Author Email: gururaghavendra286@gmail.com

https://doi.org/10.55041/ijsmt.v2i11.008

Cite this Article: S, G. R. D., M, C., M, C. K., S, R. G. D. & N, M. (2026). CardioFL: A Federated Learning Prototype for Heart Disease Prediction Using Simulated Hospital Clients. International Journal of Science, Strategic Management and Technology, *02*(11), 1-9. https://doi.org/10.55041/ijsmt.v2i11.008

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract—

Machine-learning models for heart-disease prediction are commonly trained on data collected in a single repository, which may raise data-governance and privacy concerns when multiple healthcare organizations are involved. This paper presents CardioFL, an educational federated-learning prototype that demonstrates collaborative Logistic Regression training across three simulated hospital clients using the Cleveland heart-disease dataset. The prototype implements local mini-batch training, weighted Federated Averaging (FedAvg), and a FedProx experiment for heterogeneous client data. In the centralized-versus-federated experiment (240 training and 57 test records), the centralized model achieved 85.96% accuracy (cross-entropy loss 0.3769) and the federated model 87.72% (loss 0.3884). In a separate experiment (238 training and 59 test records), IID + FedAvg, Non-IID + FedAvg, and Non-IID + FedProx achieved 84.75%, 81.36%, and 86.44% accuracy, respectively. A separate PyTorch reference experiment (237 training and 60 test records) reported 88.33% centralized and 85.00% federated accuracy. These experiments use different configurations and are not directly interchangeable. The three clients are simulated within one application backend; the prototype implements neither secure aggregation nor differential privacy and has not undergone clinical validation. The results demonstrate a federated-learning workflow on a small benchmark dataset, not clinical readiness or formal privacy protection.

Keywords— Federated Learning; Heart Disease Prediction; Logistic Regression; FedAvg; FedProx; Non-IID Data

I. INTRODUCTION

A. Background and Motivation Cardiovascular disease is an important public-health concern, and machine-learning methods have been investigated for supporting risk assessment. Predictive models can learn relationships between clinical attributes and disease labels. However, healthcare data are often held by separate organizations, and combining records into one repository may raise privacy, governance, and operational concerns. Federated learning (FL) offers an alternative training arrangement in which participating clients train local models and a coordinating server aggregates model updates.

B. Federated Learning in Healthcare In a typical FL workflow, a server distributes a global model to participating clients. Each client trains the model using its local training records and returns model parameters or updates. The server combines these updates to form a new global model. This approach can reduce the need to pool raw records for model training. However, sharing model updates does not automatically guarantee privacy [1], [2]. Model updates can contain information about training examples, and additional techniques such as differential privacy [3] and secure aggregation [4] are needed to address particular information-leakage threats.

C. Objectives and Contributions The objectives of CardioFL are to: (1) develop a Logistic Regression model for heart-disease prediction; (2) demonstrate collaborative training across three simulated hospital clients; (3) implement weighted FedAvg for global model aggregation; (4) compare IID and Non-IID client distributions and evaluate FedProx; and (5)

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

compare centralized and federated model performance using accuracy and cross-entropy loss. The contribution is an educational software prototype and a reproducible set of benchmark experiments, not a deployed multi-hospital clinical system.

II. LITERATURE REVIEW

McMahan et al. introduced Federated Averaging (FedAvg), in which locally trained model updates are aggregated using client data sizes [5]. Kairouz et al. surveyed the opportunities and open problems in federated learning, including statistical heterogeneity, communication, and privacy challenges [1]. Li et al. proposed FedProx, which adds a proximal regularization term to local optimization to address variability among participating clients [6]. Detrano et al. reported an internationally applicable probability algorithm for diagnosing coronary artery disease using clinical data, and the Cleveland dataset has subsequently been used in cardiovascular prediction research [7]. Healthcare applications and challenges of federated learning have been discussed by Li et al., Rieke et al., and Brisimi et al. [8], [2], [9].

III. METHODOLOGY

A. System Architecture CardioFL consists of a browser-based interface and a Python backend. Its main functions include patient assessment, simulated hospital login, dataset upload, local Logistic Regression training, model-update submission, global aggregation, and experiment comparison. The three participating clients are labelled H001, H002, and H003. They are simulated clients operated within a single application/backend environment; they are not independent physical hospitals or independently administered servers.

B. Dataset The experiments use the Cleveland heart-disease dataset containing 297 records and 13 input attributes with a binary target field named condition. For demonstration, the application divides the full 297-record dataset into three deterministic 99-record partitions ($3 \times 99 = 297$), one for each simulated client. These demo partitions are not the experimental training sets: each reported experiment first holds out its own test records (57, 59, and 60 records, respectively), leaving 240, 238, and 237 training records. The dataset is a small benchmark and does not represent the full range of clinical populations, diagnostic practices, or patient characteristics.

C. Feature Processing The backend converts the input attributes into numeric model features, applies fixed continuous-feature scaling statistics, and imputes missing continuous values using the configured reference means. Chest-pain category values are mapped through a hand-designed numeric encoding. The mathematical/code audit identified that this encoding does not preserve the empirical direction of the dataset's chest-pain association: category cp = 3 has a higher observed positive-label proportion than cp = 0, while the implemented mapping assigns cp = 3 a lower encoded value. In addition, several categorical fields are treated as numeric scalar inputs. These preprocessing choices are limitations and may affect model behavior; they should be improved and re-evaluated in future work.

D. Logistic Regression For an input vector x, Logistic Regression estimates the positive-class probability as $p(y = 1 \mid x) = \sigma(w^{\top}x + b)$, where $\sigma(z) = 1 / (1 + e^{-z})$, w is the weight vector, and b is the bias. The model is optimized using binary cross-entropy. The implementation uses mini-batch gradient updates. Accuracy and binary cross-entropy loss are reported for the held-out evaluation data.

E. Federated Averaging At communication round t, each client receives the current global parameters, performs local training, and returns a model-parameter update. The server computes a weighted average based on the number of local training records. The update is expressed as $w(t+1) = w(t) + \Sigma_k [(n_k/N) \times \Delta w_k]$, where n_k is the training-record count at client k and N is the total number of training records. The bias parameter is aggregated in the same way [5].

F. FedProx FedProx adds a proximal penalty to the local objective to discourage local parameters from moving too far from the current global parameters. Its objective adds a penalty proportional to the

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

squared distance between the local parameter vector and the current global parameter vector; μ controls the penalty strength [6]. The CardioFL comparison uses μ = 0.15 for the Non-IID FedProx configuration.

G. Experiment Configurations The centralized-versus-federated experiment uses 240 training records and 57 held-out test records, with three client partitions, 10 federated rounds, 20 local epochs per round, learning rate 0.08, and batch size 32. The IID/Non-IID/FedProx comparison uses 238 training records and 59 held-out test records, 10 rounds, 15 local epochs, learning rate 0.20, batch size 32, and FedProx μ = 0.15. The PyTorch reference experiment is separate and uses 237 training records, 60 test records, three label-skewed clients, and 10 rounds. Results from these experiments are reported separately because the splits and experiment configurations differ.

H. Privacy and Deployment Scope In the prototype, CSV datasets are uploaded to the application backend for the simulated client workflow. Therefore, the system does not demonstrate that raw data remain on independently controlled hospital infrastructure. The update-submission step aggregates model parameter updates rather than sending raw patient rows as part of the aggregation message. Nevertheless, the prototype does not implement secure aggregation or differential privacy, and it does not provide a formal privacy guarantee. The backend uses in-memory sessions and demo credentials; production-grade identity management, deployment security, and independent-client isolation are outside the scope of this implementation.

IV. RESULTS AND DISCUSSION

A. Centralized versus Federated Evaluation The centralized-versus-federated endpoint was executed on the same held-out records for both approaches. The deterministic experiment used 240 training records and 57 test records, as summarized in Table I. The centralized Logistic Regression model achieved 85.96% accuracy and 0.3769 cross-entropy loss. The federated Logistic Regression model using weighted FedAvg achieved 87.72% accuracy and 0.3884 loss. The federated model's accuracy was 1.76 percentage points higher in this run, while its reported loss was slightly higher. These metrics describe this particular split and configuration; they do not establish general superiority of federated training.

Table I: Centralized versus Federated Evaluation

Model	Train	Test	Accuracy	BCE loss
Centralized LR	240	57	85.96%	0.3769
Federated LR (FedAvg)	240	57	87.72%	0.3884

B. IID, Non-IID, and FedProx Evaluation A separate three-way experiment used 238 training records and 59 held-out test records (Table II). IID + FedAvg achieved 84.75% accuracy and 0.3387 loss. Non-IID + FedAvg achieved 81.36% accuracy and 0.4031 loss. Non-IID + FedProx achieved 86.44% accuracy and 0.4085 loss. In this experiment, FedProx improved accuracy by 5.08 percentage points over Non-IID + FedAvg, while its loss was higher. Accuracy and loss therefore provide different views of model performance and should both be reported.

Table II: IID, Non-IID, and FedProx Comparison

Configuration	Train	Test	Accuracy	BCE loss
IID + FedAvg	238	59	84.75%	0.3387
Non-IID + FedAvg	238	59	81.36%	0.4031
Non-IID + FedProx (μ=0.15)	238	59	86.44%	0.4085

C. Independent PyTorch Reference Experiment A separate PyTorch Logistic Regression reference experiment used 237 training records and 60 held-out test records, with three label-skewed clients and 10 rounds. It achieved 88.33% centralized accuracy and 85.00% federated accuracy, a 3.33 percentage-point difference. This is an independent reference experiment and is not merged with the backend experiment results in Tables I and II.

D. Engineering Validation The project audit reported 10/10 engineering sanity checks passed and 19/19 backend API/workflow integration checks passed for the audited source snapshot. The checks covered dataset schema, client partitioning, backend import and API behavior, local training, aggregation arithmetic, preprocessing, prediction probability

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

bounds, authentication checks, and update-submission handling. Python syntax checks also passed. The frontend production build and browser-based end-to-end workflow were not verified because dependency installation did not complete in the audit environment. The backend integration checks used two local epochs for smoke testing, not the full experiment configuration. These checks are software verification results, not clinical validation or a security certification.

E. Discussion The experiments show that the prototype executes local training, weighted aggregation, and comparisons under IID and Non-IID client partitions. The reported scores differ across experiment configurations and should not be compared as though they were produced by one common split. The small dataset limits statistical confidence and generalizability. The audit also identified preprocessing limitations, including a hand-designed chest-pain encoding whose direction conflicts with the observed dataset association, and categorical features represented as scalar numeric values. These limitations should be addressed before interpreting the model as a robust clinical predictor.

F. Limitations and Future Work The current work has several limitations. First, the three hospital clients are simulated in a single backend, and uploaded CSV data are handled by that backend; the prototype does not demonstrate independent institutional data residency. Second, secure aggregation and differential privacy are not implemented. Third, the benchmark contains only 297 records, which limits generalizability. Fourth, the current preprocessing includes fixed scaling constants and a hand-designed chest-pain encoding, while some categorical attributes are treated as scalar values. Fifth, the backend uses demo credentials and in-memory sessions. Sixth, the frontend production build and browser workflow were not verified in the audit environment. Finally, no prospective clinical evaluation has been performed.

Future work should improve categorical feature encoding and fit preprocessing parameters only from training data, evaluate the model with repeated or external validation, test on larger and more diverse datasets, deploy independently administered clients, add secure aggregation and differential privacy, strengthen authentication and transport security, complete frontend/browser testing, and assess calibration and subgroup performance. Clinical use would require independent clinical validation and appropriate regulatory review.

V. CONCLUSION

This paper presented CardioFL, an educational federated-learning prototype for heart-disease prediction using Logistic Regression, three simulated clients, weighted FedAvg, and a FedProx comparison. In the centralized-versus-federated experiment, the centralized model achieved 85.96% accuracy and the federated model achieved 87.72% on 57 held-out records. In a separate IID/Non-IID experiment, Non-IID + FedProx achieved 86.44% accuracy compared with 81.36% for Non-IID + FedAvg. A separate PyTorch reference experiment reported 88.33% centralized and 85.00% federated accuracy. These results are tied to their respective experiment configurations. The prototype demonstrates federated-learning mechanics on a small benchmark dataset but does not establish clinical validity, production security, or formal privacy guarantees.

ACKNOWLEDGMENT

The authors express their sincere gratitude to Dr. Manjunath N, Project Guide, Department of Computer Science and Engineering, Vidya Vikas Institute of Engineering and Technology, for his guidance and support throughout the project. The authors also thank the Department of Computer Science and Engineering for providing academic support and facilities.

REFERENCES

[1] P. Kairouz et al., “Advances and open problems in federated learning,” Foundations and Trends in Machine Learning, vol. 14, nos. 1–2, pp. 1–210, 2021.

[2] N. Rieke et al., “The future of digital health with federated learning,” npj Digital Medicine, vol. 3, art. 119, 2020.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

[3] C. Dwork, "Differential privacy: A survey of results," in Theory and Applications of Models of Computation, Springer, pp. 1–19, 2008.

[4] K. Bonawitz et al., "Towards federated learning at scale: System design," Proceedings of Machine Learning and Systems, 2019.

[5] 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. Artificial Intelligence and Statistics, PMLR, vol. 54, pp. 1273–1282, 2017.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," Proceedings of Machine Learning and Systems, vol. 2, pp. 429–450, 2020.

[7] R. Detrano et al., "International application of a new probability algorithm for the diagnosis of coronary artery disease," The American Journal of Cardiology, vol. 64, no. 5,
pp. 304–310, 1989.

[8] L. Li, Y. Fan, M. Tse, and K. Y. Lin, "A review of applications in federated learning for healthcare," Computers S Industrial Engineering, vol. 149, art. 106854, 2020.

[9] T. S. Brisimi et al., "Federated learning of predictive models from federated electronic health records," International Journal of Medical Informatics, vol. 112, pp. 59–67, 2018.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.