



MedTrace: A Privacy-Preserving Federated Learning Framework For Clinical Drug Interaction Monitoring

1st Reena Satpute, 2nd Ananya Lokhande, 3rd Shravani Pimpale, 4th Anushri Mahajan, 5th Gopal Padalkar, 6th Rohit Jumna

¹Department of Computer Science and Engineering Nagpur Institute of Technology, Nagpur, Maharashtra, India.
reenasatpute2017@gmail.com

⁴Department of Computer Science and Engineering Nagpur Institute of Technology, Nagpur, Maharashtra, India.
mahajanandhri_6239cs23@nit.edu.in

²Department of Computer Science and Engineering Nagpur Institute of Technology, Nagpur, Maharashtra, India.
lokhandeananya_6222cs23@nit.edu.in

⁵Department of Computer Science and Engineering Nagpur Institute of Technology, Nagpur, Maharashtra, India.
padalkargopal_6829cs24s@nit.edu.in

³Department of Computer Science and Engineering Nagpur Institute of Technology, Nagpur, Maharashtra, India.
pimpaleshravani_6355cs23@nit.edu.in

⁶Department of Computer Science and Engineering Nagpur Institute of Technology, Nagpur, Maharashtra, India.
jumnaakerohit_6254cs23@nit.edu.in

ABSTRACT—

Drug-drug interactions (DDIs) are a significant concern in healthcare, particularly when patients are prescribed multiple medicines at the same time. One drug can change the effect of another, which may lead to adverse effects, reduced treatment effectiveness, or other unexpected outcomes. Machine learning can help identify such interactions, but many conventional approaches depend on collecting healthcare data in a central location. This is difficult in practice because clinical records contain sensitive patient and institutional information and cannot always be freely shared. To address this issue, this paper proposes MedTrace, a privacy-preserving framework for clinical drug interaction monitoring using Federated Learning (FL). In MedTrace, participating healthcare institutions keep their data locally and train the model on their own datasets. Instead of sending patient records to a central server, they share only the model updates required for collaborative training. The federated server combines these updates to create a global model and sends the improved model back to the participating clients. The application layer uses Python and Django, while Flower can be used to manage the federated learning process. The aim of MedTrace is to support the identification of possible drug interactions and medication-related anomalies without requiring institutions to give up control of their clinical data. By bringing together federated learning, clinical monitoring, and a web-based interface, the framework provides a practical starting point for privacy-aware healthcare analytics.

Index Terms—Federated Learning, Drug-Drug Interaction, Healthcare, Privacy-Preserving Machine Learning, Anomaly Detection, Django, Python, Clinical Decision Support.

Introduction

As the number of medicines prescribed to a patient increases, commonly referred to as polypharmacy, the chance of a drug-drug interaction also increases. A DDI occurs when one medicine changes the effect of another medicine taken at the same time [1]. Depending on the drugs involved, this can cause unwanted side effects, reduce treatment effectiveness, or create serious health risks. For this reason, computational prediction of DDIs has become an important area of research in drug discovery and pharmacovigilance.

Artificial intelligence and machine learning offer useful ways to analyse large biomedical datasets and uncover relationships that may be difficult to identify manually. Previous studies have explored several approaches, including methods based on chemical structures, interaction networks, natural language processing, and combinations of these techniques [2-4]. Deep learning and graph-based models are particularly useful because they can represent molecular structures and relationships between drugs when making interaction predictions.

Privacy is one of the main challenges when machine learning is applied to healthcare. Medical records can include patient details, medication histories, diagnoses, laboratory results, and other sensitive information. In a conventional centralized setup, data from different hospitals or organizations must be moved to a shared repository. Apart from increasing security risks, this can create problems related to data ownership, institutional policies, and regulatory requirements. These concerns can make collaboration between healthcare organizations difficult.

Federated Learning (FL) provides a different approach by allowing organizations to learn together without moving their original datasets to a central location. In an FL setup, each institution trains the model locally using the data it already has. The institution then sends model parameters or updates to a central server, which combines the contributions from participating clients [5]. This makes FL especially relevant to healthcare, where institutions may want to benefit from shared learning while continuing to keep raw clinical records within their own environments.

for aggregation. This architecture is particularly suitable for healthcare settings where institutions need collaborative intelligence without directly exchanging raw clinical records.

II. Problem statement

Many existing machine learning approaches for clinical drug interaction monitoring depend on access to sufficiently large datasets. Bringing data from several healthcare institutions into one central repository, however, can expose sensitive information and create privacy, security, ownership, and governance concerns. On the other hand, if every institution works completely independently, its model can learn only from the information available at that site and may miss useful patterns present in other institutions.

MedTrace focuses on the problem of enabling multiple healthcare institutions to work together on drug interaction and medication anomaly monitoring without moving sensitive clinical data outside their respective institutions. The proposed solution uses federated learning, where model updates are exchanged instead of raw clinical records. This creates a decentralized learning process in which each institution contributes to a shared model while retaining control of its local data [6]. A Django-based application layer is included to give authorized users access to predictions, alerts, model status, and other operational information.

The proposed system is therefore designed around three requirements: preserve local data ownership and locality, enable collaborative model training across participating clients, and provide a usable monitoring interface for authorized users. The primary objective of the proposed MedTrace framework is to develop a privacy-preserving system for monitoring clinical drug interactions while maintaining the confidentiality of sensitive patient data. The framework aims to utilize Federated Learning (FL) to enable multiple healthcare institutions to collaboratively train a machine learning model without sharing their raw clinical data with a central server. It further focuses on identifying potential drug-drug interactions and medication-related anomalies to support safer and more effective medication management [7]. In addition, the system is designed to provide an authenticated dashboard through which authorized users can access prediction results, alerts, model status, and performance information. The effectiveness of the proposed framework will be evaluated using standard machine learning metrics, including accuracy, precision, recall, F1-score, and Area Under the Curve (AUC), along with system-specific parameters such as training time, number of communication rounds, and convergence behaviour.

III. Scope of the study

MedTrace is designed as a student-scale prototype that demonstrates privacy-aware collaborative drug interaction monitoring [8]. The framework can represent different healthcare institutions as separate federated clients. Public benchmark datasets such as DrugBank or TWOSIDES may be considered when appropriate access and licensing are available. During development, several clients can be simulated on one computer or run as separate processes, with each client working on its own partition of the dataset.

The system covers data preprocessing, local model training, federated aggregation, prediction generation, web-based monitoring, and comparative evaluation. It is not intended to replace a clinician or provide autonomous clinical diagnosis. Real clinical deployment would require appropriate clinical validation, governance, patient-consent considerations, security controls, and regulatory compliance

IV. Literature review

a. Deep and Graph Learning for DDI Prediction

Deep and graph learning methods have expanded computational approaches to DDI prediction. The reviewed literature categorizes DDI models according to input representation, architecture, and prediction task. Molecular structures, drug networks, biomedical knowledge graphs, and sequence or text representations can be combined to capture complementary information. DrugBank and TWOSIDES are widely used benchmark resources in DDI research; however, access, licensing, and experimental assumptions must be considered before implementation.

b. Federated Learning in Healthcare

Federated learning has increasingly been studied as a way to support healthcare analytics while reducing the need to share raw data. A typical FL workflow keeps data at the local institution, performs training locally, sends model updates to a server, aggregates those updates, and distributes the resulting global model back to clients [9]. Research has considered FL for areas such as medical imaging, electronic health records, laboratory data, and physiological signals. However, FL by itself should not be considered a complete privacy solution, since model parameters or gradients can still contain information related to the data used during training.

c. Federated Learning for Pharmaceutical Tasks

Mittone et al. presented a federated learning benchmark for drug-target interaction using FedAvg with GraphDTA and the KIBA dataset. Their study considered different data-ownership distributions and levels of non-IID data. Another multi-party collaborative drug discovery framework investigated FL for drug-target affinity and DDI using multiple clients and benchmark datasets. These studies motivate an application-oriented system that connects federated pharmaceutical learning with a practical monitoring interface.

d. Research Gap

The literature shows that both DDI prediction and federated learning in healthcare are well-established research areas [10]. However, there is still a practical gap between studies that mainly evaluate model performance and systems that bring collaborative learning into an integrated monitoring application. MedTrace is intended to bridge this gap by combining local model training, federated coordination, prediction monitoring, and a Django-based web application within a reproducible student-scale framework.

TABLE I — COMPARISON OF APPROACHES

Approach	Data Location	Collaboration	Privacy	Main Focus
Centralized ML	Central Server	High	Lower	Prediction
Local ML	Individual sites	High	High	Site-specific mode
Federated ML	Individual sites	High	Higher*	Collaborative learning
MedTrace	Individual sites	High	Higher*	DDI monitoring + web applications

V. Proposed architecture and flow

The MedTrace architecture is organized into four main layers: clinical data, local machine learning, federated coordination, and application. Each layer has a specific role and can be developed separately [11]. This separation keeps the architecture manageable while ensuring that local data remain under the control of each participating institution and that the global model can still learn from multiple clients.

A. Clinical Data Layer

Each participating institution maintains its own local dataset containing the information needed for drug interaction or anomaly prediction. Depending on the dataset used, this may include drug identifiers, drug-pair information, interaction labels, dosage-related features, clinical observations, or derived numerical representations. Patient-identifiable information is kept within the local environment and is not intended to be sent to the federated server.

B. Local Learning Layer

Each participating institution is represented as an FL client. The client loads and preprocesses its local data, trains the selected model, evaluates the local results, and generates updated model parameters. Only the information required by the federated learning protocol is sent to the server.

C. Federated Coordination Layer

The federated server manages the overall training process. At the beginning of a round, selected clients receive the current global model and train it using their local datasets for a specified number of epochs. They then return their model updates to the server [12]. The server combines these updates to form the next global model, which is distributed to the clients for the following round.

D. Django Application Layer

The Django backend acts as the application layer around the machine learning system. It can handle user authentication, client registration, prediction requests, model status, logs, alerts, and result visualization. Django REST Framework can be used to provide controlled APIs between the web interface and the Python-based services [13]. Since the federated workflow is based on local training, the application does not need to centrally store raw patient records.

TABLE II — MEDTRACE ARCHITECTURE COMPONENTS

Layer	Component	Responsibility
Presentation	HTML/CSS/React	Dashboard, prediction and alert views
Application	Django	Authentication, APIs, workflow control
FL Coordination	Flower / FL server	Training rounds and aggregation
Client ML	Python ML	Local training and evaluation
Data	Local DB/files	Institution-specific clinical data

The proposed MedTrace architecture is a privacy-preserving, federated learning framework designed for clinical drug interaction monitoring. The architecture consists of multiple healthcare institutions that act as independent federated clients. Each client maintains its own local healthcare data and performs machine learning locally. Instead of transferring raw clinical data to a central server, only the trained model updates are transmitted to the Federated Server. The server aggregates these updates using the Federated Averaging (FedAvg) algorithm to create an improved global model. The global model is then distributed back to the participating clients. A Django-based web application provides an interface for authorized users to access prediction results, alerts, monitoring information, and model-related information.

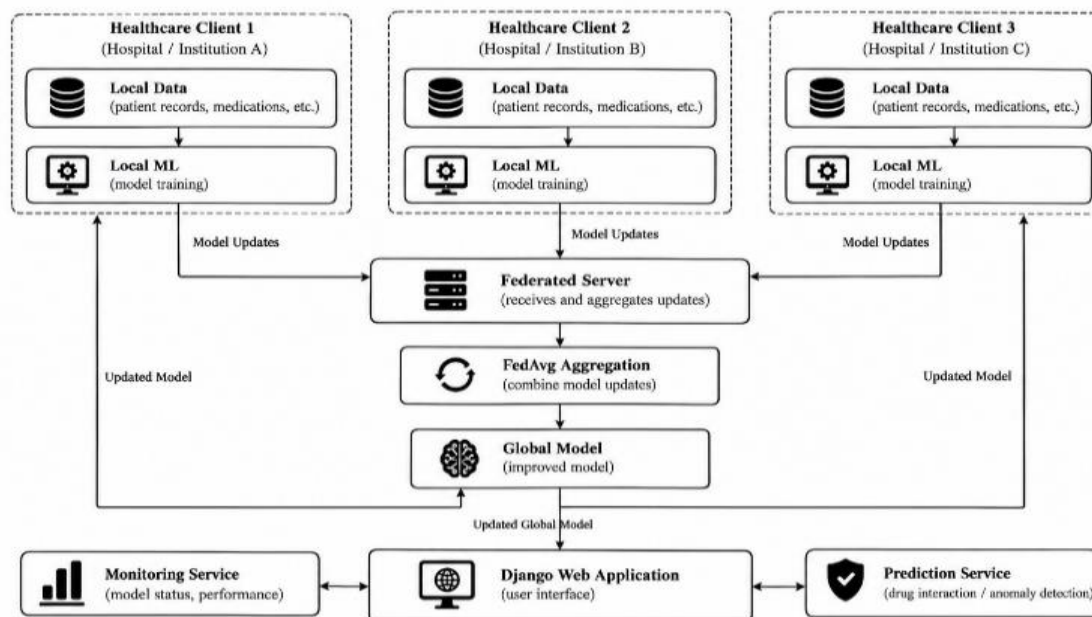


Fig 1. Proposed system Architecture

A. Healthcare Clients

Different hospitals or healthcare institutions participating in the federated learning process. Each client keeps its clinical data locally and trains the model independently.

B. Local Data

Patient/medication-related data available within each healthcare institution. The data is used for training but is not directly shared with the central server.

C. Local Machine Learning

Each healthcare client preprocesses its local data and trains the ML model using its own dataset. This allows the model to learn local drug-interaction patterns.

D. Model Updates

After local training, each client generates model parameters or updates representing what the model learned. These updates are sent to the federated server instead of raw clinical data.

E. Federated Server

The central coordination component that distributes the global model, receives updates from clients, and manages the federated training process.

F. FedAvg Aggregation

Federated Averaging (FedAvg) combines the model updates received from different healthcare clients to create an improved global model.

G. Global Model

The collaboratively trained model created from the aggregated client updates. It is sent back to participating clients for further training and can be used for predictions.

H. Django Web Application

The application layer that provides a web interface for authentication, prediction requests, alerts, monitoring, and communication with the ML services.

I. Prediction Flow

The process where an authorized user provides drug information, the system processes the request using the trained model, and a DDI/anomaly prediction is generated.

J. Risk Threshold & Alert Generation

The predicted score is compared with a selected risk threshold. If the prediction exceeds the threshold, MedTrace generates a potential risk alert.

K. Django Dashboard

The user interface where authorized users can view prediction results, alerts, model status, and performance information.

L. Operational Log

Stores application-related records such as prediction requests and system events for monitoring and auditing purposes.

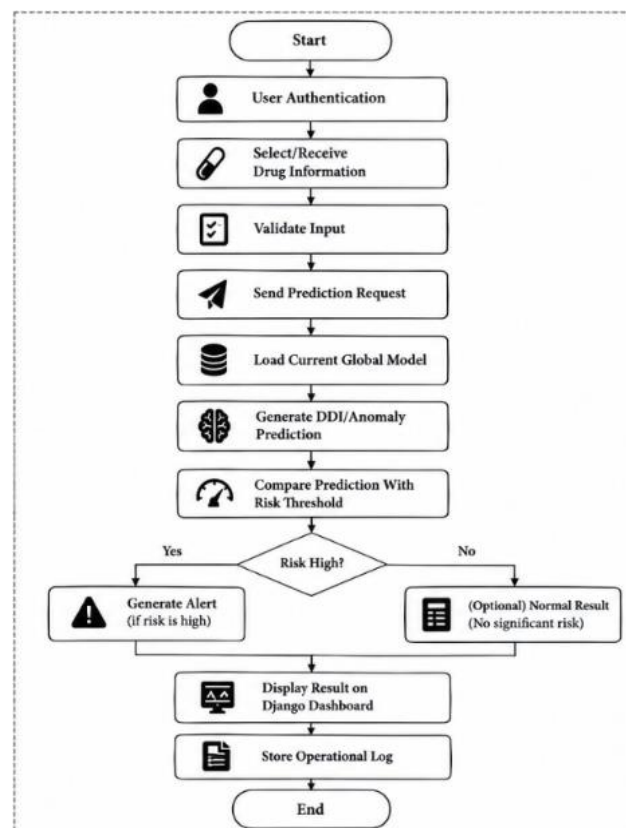


Fig 2. Prediction Flow

A. User Authentication

The user first logs into the MedTrace system. Only authenticated users can access the prediction functionality.

B. Select/Receive Drug Information

The user enters or selects the required drug information, such as the drugs or drug combination to be analysed.

C. Validate Input

The system checks whether the entered information is complete and valid before sending it for prediction.

D. Send Prediction Request

After validation, the information is sent to the prediction service for analysis.

E. Load Current Global Model – The system loads the latest global ML model produced through the federated learning process.

F. Generate DDI/Anomaly Prediction

The model analyses provided drug-related features and predicts whether a potential drug-drug interaction (DDI) or medication-related anomaly exists.

G. Compare with Risk Threshold

The prediction score is compared with a risk a selected threshold to determine whether the detected interaction represents a potential high-risk condition. The threshold should be selected using validation data.

H. Generate Alert

If the prediction exceeds the risk threshold, MedTrace generates a potential interaction/risk alert for the user.

I. Display Result on Django Dashboard

The prediction and alert are displayed through the authenticated Django dashboard, allowing the user to view the result and relevant system information.

J. Store Operational Log

The system records the operational event, such as the prediction request and system activity, for monitoring and auditing.

VI. Drug interaction and anomaly detection

MedTrace is intended to provide two related outputs: drug interaction prediction and anomaly monitoring. The interaction component estimates whether a particular drug pair is associated with an interaction class. The anomaly component looks for medication combinations or patterns that differ from the learned normal behaviour or exceed a defined risk threshold.

For a binary model, the predicted probability can be compared with a threshold T . If $P(\text{interaction} | X) > T$, the application generates a potential interaction alert [15]. The threshold should be selected using validation data rather than arbitrarily fixed because different operating points produce different precision and recall.

The alert displayed by Django is intended as a decision-support signal rather than an autonomous clinical diagnosis [16]. Real clinical use requires domain validation, human review, governance, and appropriate regulatory controls [17].

VII. Implementation and design

TABLE III — PROPOSED SOFTWARE STACK

Component	Technology
Programming	Python
Web Framework	Django
API	Django REST Framework
Federated Learning	Flower
Machine Learning	Scikit-learn / PyTorch
Database	PostgreSQL
Data Processing	Pandas, NumPy
Visualization	Matplotlib / Chart.js
Frontend	HTML, CSS, React
Aggregation	FedAvg

The implementation can be organized as separate services: a Django web server, a federated server, and one or more FL clients. During development, multiple clients may be simulated on one machine or across several processes [18]. Each client uses a separate local data partition to reproduce institutional data ownership.

The Django dashboard can display model version, training round, client participation status, evaluation metrics, prediction results, and alerts. PostgreSQL can store user accounts, application metadata, prediction logs, and other non-sensitive operational information [19]. Local clinical training data remain in the intended client environment.

VIII. Privacy and security

The primary privacy advantage of MedTrace is the avoidance of direct raw-data exchange during federated training. Each institution retains its local dataset while contributing model updates. This reduces the need for a centralized repository containing all participating clinical records [20].

However, federated learning should not be treated as a complete privacy solution. Model gradients or parameters may leak information about training data [21]. Secure aggregation can prevent the server from seeing individual client updates, while differential privacy can reduce information contained in model updates [22]. Encryption, strong authentication, authorization, audit logging, and adversarial defenses are additional measures for sensitive healthcare deployments [23].

The baseline student prototype can demonstrate data locality and federated aggregation, while advanced privacy mechanisms can be implemented and evaluated as future extensions [24].

IX. Experimental setup

The evaluation compares three learning settings: local learning, centralized learning, and federated learning. A public DDI benchmark such as DrugBank or TWOSIDES may be used where appropriate access and licensing are available. The selected dataset is partitioned into multiple simulated clients to represent different institutions [25].

TABLE IV — EXPERIMENTAL SETTINGS

Experiment	Training Data	Purpose
Local	One client only	Measure site-specific learning
Centralized	Combined training data	Upper/reference benchmark
Federated	Distributed client partitions	Measure collaborative learning

The federated experiment should vary the number of clients, local epochs, communication rounds, and data distribution [26]. Where feasible, both approximately IID and non-IID partitions should be tested because healthcare data can differ in sample size, feature distributions, patient populations, and clinical patterns

X. Evaluation metrics

For classification, the following metrics are recommended:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

$$\text{AUC} = \text{area under the receiver operating characteristic curve [27].}$$

In addition to predictive performance, the implementation should record communication rounds, training time, client participation, model convergence, and, where privacy mechanisms are implemented, their computational and communication overhead

XI. Result and discussion

At the present design stage, the paper does not contain measured performance values from an executed MedTrace implementation. Therefore, no numerical results are claimed here. The results section should be completed after the system is implemented and evaluated using the selected dataset and experimental settings [28]. The table below is provided as a template for recording those measured values.

TABLE V — EXPERIMENTAL RESULTS TO BE REPLACED WITH MEASURED VALUES

Method	Accuracy	Precision	Recall	F1	AUC
Local learning	TBD	TBD	TBD	TBD	TBD
Centralized learning	TBD	TBD	TBD	TBD	TBD
Federated learning	TBD	TBD	TBD	TBD	TBD

The expected analysis is that federated learning can provide a collaborative model that improves over individual local models when complementary information exists across clients. Its performance may approach the centralized benchmark, but the actual outcome depends on model architecture, data partitioning, number of clients, local epochs, and non-IID severity. Published pharmaceutical FL results should not be reported as MedTrace results. A second part of the discussion should analyse data heterogeneity. If client datasets become increasingly non-IID, global model quality may decrease or convergence may become less stable. Therefore, the final paper should report how performance changes as the number of clients, local training epochs, and heterogeneity are varied.

Recommended figures for the final version include: (1) accuracy/F1 versus communication rounds, (2) local versus centralized versus federated metric comparison, (3) confusion matrix for the final model, and (4) training time or communication overhead versus number of clients.

XII. Limitation

- Performance depends strongly on the quality, representativeness, and licensing of the selected dataset.
- Public benchmark data may not represent the complexity of real hospital records.
- Non-IID client distributions may reduce global model quality.
- Federated training introduces communication and coordination overhead.
- Model updates may still leak information without additional privacy mechanisms [29].
- Real clinical deployment requires validation, authorization, patient-consent considerations, governance, and regulatory compliance.

XIII. Conclusion

This paper presented MedTrace, a privacy-preserving framework for clinical drug interaction monitoring based on federated learning. The proposed approach addresses an important limitation of centralized healthcare machine learning by allowing participating institutions to train models locally and exchange model updates instead of raw clinical records. The system brings together Python, Django, machine learning, and federated learning to provide a practical architecture for collaborative DDI monitoring.

The framework also provides a reproducible way to compare local, centralized, and federated learning in a simulated multi-institution environment. Its main contribution is the combination of decentralized training with a web-based monitoring layer rather than treating the machine learning model as an isolated component. The next step is to implement and evaluate the system using a suitable DDI dataset, multiple client partitions, and measured performance and convergence metrics. Any numerical results should be reported only after the experiments have actually been performed.

XIV. Future scope

- Secure aggregation to protect individual client updates during server-side aggregation.

- Differential privacy to provide formal privacy guarantees for model updates.
- Non-IID-aware federated optimization for heterogeneous healthcare clients.
- Real-time medication monitoring and alert generation.
- Explainable AI to provide reasons for a predicted interaction or anomaly.
- Integration of biomedical language models for clinical-text analysis.
- Blockchain-based audit mechanisms for model-update traceability where appropriate.
- Multi-hospital pilot evaluation after security, governance, and clinical validation.

REFERENCES

- [1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.
- [2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020, doi: 10.1109/COMST.2020.2986024.
- [3] N. Rieke et al., "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, Art. no. 119, 2020, doi: 10.1038/s41746-020-00323-1.
- [4] M. J. Sheller et al., "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Scientific Reports*, vol. 10, Art. no. 12598, 2020, doi: 10.1038/s41598-020-69250-1.
- [5] A. Sadilek et al., "Privacy-first health research with federated learning," *npj Digital Medicine*, vol. 4, Art. no. 132, 2021, doi: 10.1038/s41746-021-00489-2.
- [6] M. Ali, F. Naeem, M. Tariq, and G. Kaddoum, "Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 778–789, Feb. 2023, doi: 10.1109/JBHI.2022.3181823.
- [7] V. A. Patel, P. Bhattacharya, S. Tanwar, R. Gupta, G. Sharma, P. N. Bokoro, and R. Sharma, "Adoption of federated learning for healthcare informatics: Emerging applications and future directions," *IEEE Access*, vol. 10, pp. 90792–90826, 2022, doi: 10.1109/ACCESS.2022.3201876.
- [8] Z. K. Taha, C. T. Yaw, S. P. Koh, S. K. Tiong, K. Kadirgama, F. Benedict, J.-D. Tan, and Y. 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, doi: 10.1109/ACCESS.2023.3267964.
- [9] M. Abaoud, M. A. Almuqrin, and M. F. Khan, "Advancing federated learning through novel mechanism for privacy preservation in healthcare applications," *IEEE Access*, vol. 11, pp. 83562–83579, 2023, doi: 10.1109/ACCESS.2023.3301162.
- [10] M. F. Khan and M. Abaoud, "Blockchain-integrated security for real-time patient monitoring in the Internet of Medical Things using federated learning," *IEEE Access*, vol. 11, pp. 117826–117850, 2023, doi: 10.1109/ACCESS.2023.3326155.
- [11] M. C. Tedeschini, S. Savazzi, R. Stoklasa, L. Barbieri, I. Stathopoulos, M. Nicoli, and L. Serio, "Decentralized federated learning for healthcare networks: A case study on tumor segmentation," *IEEE Access*, vol. 10, pp. 8693–8708, 2022, doi: 10.1109/ACCESS.2022.3141913.
- [12] M. F. Sohan and A. Basalamah, "A systematic review on federated learning in medical image analysis," *IEEE Access*, vol. 11, pp. 28628–28644, 2023, doi: 10.1109/ACCESS.2023.3260027.
- [13] S. H. Moon and W. H. Lee, "Privacy-preserving federated learning in healthcare," in *Proc. 2023 International Conference on Electronics, Information, and Communication (ICEIC)*, 2023, doi: 10.1109/ICEIC57457.2023.10049966.
- [14] M. J. Ali et al., "Privacy-preserving federated learning with domain adaptation for multi-disease ocular disease recognition," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 6, pp. 3219–3227, 2024, doi: 10.1109/JBHI.2023.3305685.
- [15] A. Padmavathi, N. Kushwaha, and S. Varma, "A privacy-preserving federated learning framework for adaptable cross-organizational collaboration," in *Proc. 2026 IEEE International Conference on AI Engineering and Innovations (AIEI)*, 2026, doi: 10.1109/AIEI69164.2026.11497944.
- [16] A. Shrivastava, H. M. Al-Jawahry, A. Kumar, and K. S. Kumar, "Federated foundation models for privacy-preserving multi-modal healthcare diagnostics," in *Proc. 2026 Contemporary Computing Innovations Conference (CCIC)*, 2026, doi: 10.1109/CCIC68129.2026.11486249.
- [17] G. Hernandez-Penalzo, S. Uribe, F. Moreno, B. Arroyo, and F. Alvarez, "Federated learning and medical device regulation: Bridging gaps in healthcare AI governance," in *Proc. 2025 IEEE International Conference on E-health Networking, Application & Services (Healthcom)*, 2025, pp. 25–31, doi: 10.1109/HealthCom60686.2025.11342827.
- [18] P. K. Shukla, P. Sharma, P. Rawat, J. Samar, R. Moriwai, and M. Kaur, "Efficient prediction of drug-drug interaction using deep learning models," *IET Systems Biology*, vol. 14, no. 4, pp. 211–216, 2020, doi: 10.1049/iet-syb.2019.0116.
- [19] M. Wang, H. Wang, X. Liu, and X. Ma, "Drug-drug interaction predictions via knowledge graph and text embedding: Instrument validation study," *JMIR Medical Informatics*, vol. 9, no. 6, Art. no. e28277, 2021, doi: 10.2196/28277.
- [20] A. K. Nyamabo, H. Yu, J.-Y. Shi, and Z. Liu, "SSI-DDI: Substructure-substructure interactions for drug-drug interaction prediction," *Briefings in Bioinformatics*, vol. 22, no. 6, 2021, doi: 10.1093/bib/bbab133.
- [21] A. K. Nyamabo, H. Yu, Z. Liu, and J.-Y. Shi, "Drug-drug interaction prediction with learnable size-adaptive molecular substructures," *Briefings in Bioinformatics*, vol. 23, no. 1, 2022, doi: 10.1093/bib/bbab441.
- [22] M. H. Al-Rabeah and A. Lakizadeh, "Prediction of drug-drug interaction events using graph neural networks based feature extraction," *Scientific Reports*, vol. 12, Art. no. 15590, 2022, doi: 10.1038/s41598-022-19999-4.
- [23] S. Liu, Y. Zhang, Y. Cui, Y. Qiu, Y. Deng, Z. Zhang, and W. Zhang, "Enhancing drug-drug interaction prediction using deep attention neural networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 2, pp. 976–985, 2023, doi: 10.1109/TCBB.2022.3172421.
- [24] F. Tanvir, K. M. Saifuddin, M. I. K. Islam, and E. Akbas, "Predicting drug-drug interactions using heterogeneous graph attention networks," in *Proc. 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2023, Art. no. 48, pp. 1–6, doi: 10.1145/3584371.3613065.
- [25] Y. Zhao, J. Yin, L. Zhang, Y. Zhang, and X. Chen, "Drug-drug interaction prediction: Databases, web servers and computational models," *Briefings in Bioinformatics*, vol. 25, no. 1, Art. no. bbad445, 2024, doi: 10.1093/bib/bbad445.
- [26] H. Luo, W. Yin, J. Wang, G. Zhang, et al., "Drug-drug interactions prediction based on deep learning and knowledge graph: A review," *iScience*, vol. 27, no. 3, Art. no. 109148, 2024, doi: 10.1016/j.isci.2024.109148.

-
- [27] Y. Wang, Z. Yang, and Q. Yao, "Accurate and interpretable drug-drug interaction prediction enabled by knowledge subgraph learning," *Communications Medicine*, vol. 4, Art. no. 59, 2024, doi: 10.1038/s43856-024-00486-y.
- [28] F.-I. Gheorghita, V.-I. Bocanet, and L. B. Iantovics, "Machine learning-based drug-drug interaction prediction: A critical review of models, limitations, and data challenges," *Frontiers in Pharmacology*, vol. 16, Art. no. 1632775, 2025, doi: 10.3389/fphar.2025.1632775.
- [29] W. Zhang, P. Peissig, Z. Kuang, and D. Page, "Adverse drug reaction discovery from electronic health records with deep neural networks," in *Proc. ACM Conference on Health, Inference, and Learning*, 2020, pp. 30–39, doi: 10.1145/3368555.3384459.
- [30] C. Y. Lee and Y.-P. P. Chen, "Prediction of drug adverse events using deep learning in pharmaceutical discovery," *Briefings in Bioinformatics*, vol. 22, no. 2, pp. 1884–1901, 2021, doi: 10.1093/bib/bbaa040.