Contents lists available at ScienceDirect

Informatics in Medicine Unlocked

journal homepage: www.elsevier.com/locate/imu

FedDeepInsight—A privacy-first federated learning architecture for medical data

Allan G. Duah [a], Roland V. Bumbuc [b,c,d], H. Ibrahim Korkmaz [c,d,e,f], Rory Wilding [g], Vivek M. Sheraton [b,*]

[a] *Informatics Institute, University of Amsterdam, Lab42, Amsterdam, Nord-Holland, The Netherlands*
[b] *Computational Science Lab, Informatics Institute, University of Amsterdam, Lab42, Amsterdam, Nord-Holland, The Netherlands*
[c] *Department of Plastic, Reconstructive and Hand Surgery, Amsterdam Movement Sciences (AMS) Institute, Amsterdam UMC, Location VUmc, Amsterdam, Nord-Holland, The Netherlands*
[d] *Department of Molecular Cell Biology and Immunology, Amsterdam Infection and Immunity (AII) Institute, Amsterdam UMC, Location VUmc, Amsterdam, Nord-Holland, The Netherlands*
[e] *Burn Center and Department of Plastic and Reconstructive Surgery, Red Cross Hospital, Beverwijk, Nord-Holland, The Netherlands*
[f] *Burn Research Lab, Alliance of Dutch Burn Care, Nord-Holland, Beverwijk, The Netherlands*
[g] *Supabase Limited Liability Company (LLC), San Francisco, CA, United States*

ARTICLE INFO

Keywords:
Federated learning
Deepinsight
Tabular data
Image data
Architecture
Privacy protection
GDPR

ABSTRACT

Medical data, hospital patient-specific data, are highly sensitive to privacy and are essential for research in the biomedical field. Although there are many new approaches to creating databases that ensure data must be FAIR and GDPR compliant, these approaches require the intervention of secured data handlers. To address this gap, this study investigates and designs a standardized Federated Learning (FL) architecture for medical data. Specifically, we examine traditional and novel methods for preprocessing, handling, and utilizing such data in FL. We develop "FedDeepInsight", a novel data transformation framework that enables tabular data augmentation and transformation into image data prior to neural network training and FL. Additionally, we analyze how the type of dataset influences the performance of federated learning algorithms and machine learning models in terms of accuracy and efficiency. Our results indicate that FedAvg is the most reliable aggregation algorithm, providing superior accuracy, stability, and convergence, and FedYogi is also viable with well-tuned hyperparameters. For privacy protection, we recommend Differential Privacy (DP) with calibrated noise multipliers and initial upper and lower bounds for stability. Ultimately, we emerge as a promising solution for secure, privacy-preserving federation learning in healthcare.

1. Introduction

In recent years, machine learning algorithms are being used for disease detection, drug discovery, and to improve the overall efficiency of health care [1]. They require a substantial amount of data to perform effectively. Unfortunately, in the field of healthcare, medical data from patients that contain sensitive information is subject to privacy regulations [2]. One of the most relevant is the General Data Protection Regulation (GDPR), a comprehensive data protection law in the European Union that aims to protect the personal data and privacy of individuals [3]. According to Recital 26 of the GDPR, if the data is truly anonymized, it can be shared, meaning it must be processed in such a way that individuals can no longer be identified, directly or indirectly. Data anonymization techniques such as k anonymity, l diversity, and t closeness offer some level of privacy protection, but often fall short of the rigorous standards of GDPR due to potential reidentification risks, their inability to fully anonymize data in all contexts, and the evolving nature of reidentification techniques [4,5]. Additionally, new methods such as the FAIR compliant database can offer some solutions, but it still requires careful implementation to ensure that all data usage complies with legal and ethical standards [6,7]. Medical institutions, such as the University Medical Center (UMC), are often labeled "data silos" due to their restricted data sharing capabilities [8] and often require many steps until research data is available [9–11]. To enable effective use of data from such "data silos", it is imperative that a privacy-compliant approach and its relevant data processing tools are deployed within the medical infrastructure. Federated learning (FL) is one such novel approach to address these challenges. This framework has gained traction due to its potential application in healthcare [12]. FL en-

[*] Corresponding author.
E-mail address: v.s.muniraj@uva.nl (V.M. Sheraton).

https://doi.org/10.1016/j.imu.2025.101691
Received 10 March 2025; Received in revised form 4 September 2025; Accepted 7 September 2025
Available online 14 September 2025

ables multiple parties to collaboratively train their machine learning models without the need to share raw data externally. Using multiple decentralized devices or servers, the models are trained with their own local data [5]. After training the local models, the parameters or gradients are transferred to a central model. The central model uses aggregation methods such as Federated Averaging (FedAvg) to combine these characteristics to improve the central model [13]. After aggregation, the updated model is sent back to each local device. This process continues to iterate, each iteration improving the models. This gives computational models the ability to make use of these research data and even train on these data without compromising privacy [14].

The current literature on FL lacks comprehensive studies on the development of a standardized FL architecture tailored to medical data. Research introducing a framework frequently lacks a thorough examination of the specifics of complex designs. There is a limited amount of literature focused on analyzing individual elements within a framework that prioritizes safeguarding personal patient information. When designing a Federated Learning Module (FLM), various components such as data distribution, privacy mechanics, communication architecture, FL algorithms, and model specifications need careful consideration [13]. A significant disparity between the models available for federated learning using imaging and tabular datasets is noticeable. Although image datasets have been extensively studied, tabular datasets, common in medical data, remain underutilized. Addressing this research gap is essential, as much medical data, including omics data, questionnaire reports, and medical summaries, is tabular in nature. Existing studies are heavily focused on image data, leaving a gap in the utilization of tabular data [15]. To address this gap, this study will investigate traditional and novel methods for using tabular data in federated learning. Specifically, we will explore two distinct approaches: using TabNet, a neural network architecture designed for tabular data. Additionally, we will convert tabular data into images with FedDeepInsight to potentially enhance the performance of the model [16,17].

Our research aims to develop an effective federated learning architecture for all types of medical data, achieving a balance between model performance and patient privacy. We will evaluate machine learning models, aggregation algorithms, and dataset types (image vs. tabular) to determine their impact on accuracy and efficiency. Essential security and data protection measures will be implemented, and we will refine the architecture to create a blueprint for a Federated Learning Module (FLM) for future applications.

2. Related work

Owing to heightened scrutiny and privacy regulations, medical institutions are prohibited from freely exchanging information with each other [18]. This unique predicament of data isolation has resulted in the terms ‘data silos’ or ‘data islands’ being attributed to hospitals and similar institutions. To achieve optimal performance, machine learning models require access to extensive and varied datasets. However, under the present conditions within the medical domain, these models cannot reach their full potential. Federated learning is proposed as a viable solution to mitigate privacy concerns of data silo. It enables machine learning models to be trained collaboratively on decentralized devices or servers without the need to share sensitive data [19]. As a relatively novel concept, there is a lack of comprehensive and standardized solutions specifically created to address the challenges inherent in the healthcare sector.

2.1. Data distribution

When designing a federated learning architecture, there are generally three approaches that can be used: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL) and Federated Transfer Learning [18,20]. We will discuss only the first two approaches due to resource limitations.

2.1.1. Horizontal federated learning

In Horizontal Federated Learning (HFL), each institution holds data from the same feature space, but has a different sample space [20]. For instance, consider two hospitals: Hospital A and Hospital B. These hospitals are located in different countries, and both are researching prostate cancer and want to train machine learning models. Their patients are unlikely to overlap, but the feature space will be the same. As local data sets share a common feature space, the parties involved can independently train their local models using the same architecture. Updating the global model becomes straightforward by averaging the parameters across all local models.

2.1.2. Vertical federated learning

Unlike HFL, in vertical federated learning (VFL), the parties possess data with different characteristics, but there is usually overlap in the sample space [20]. Consider two hospitals, Hospital A and Hospital C, located in the same city. Hospital A has detailed patient records, including demographic information and medical history, while lacking specific diagnostic details, such as MRI images. On the other hand, Hospital C specializes in diagnostic imaging, but has limited demographic and medical information. Vertical Federated Learning involves the aggregation of distinct features from multiple parties and the computation of training loss and gradients in a privacy-preserving manner.

2.2. Data privacy

When it comes to FL, the preservation of privacy is of utmost importance [5,13]. Even if raw data are not shared, there is still a potential risk of indirectly leaking sensitive data through exposure to information when exchanging model parameters. Several techniques can be used to protect sensitive information. A popular method to protect data privacy is Differential Privacy (DP) [21]. There are several approaches to implementing DP, but we will focus on central DP and local DP. Local DP is applied on the client side before any information is sent to the server, ensuring that updates sent to the server do not reveal any details about the client data. In contrast, the server applies the central DP to prevent the aggregated model from disclosing information about the data of each client [22]. Central DP consists of two key components: clipping client updates and adding noise to the aggregated model. When noise or randomness is added to the learning process, individual data points are protected. However, adding noise can negatively affect the accuracy. The challenge lies in striking a balance between robust privacy protection and maintaining the predictive performance of the model.

2.3. Communication systems

Communication systems architecture play an important role in the coordination of model updates between servers or devices, directly affecting the effectiveness of FL. There are two types of communication architecture: centralized and decentralized [5,23]. In a centralized architecture also known as the ’Client–Server architecture’, all information is passed through a central server which acts as a coordinating entity. The central server manages the communication and synchronization of the clients and trains the global model by aggregating all the parameters of the local model from each device or client. Due to its simplified nature, this architecture has been widely adopted in FL studies [13]. However, there are concerns that the central server becomes a vulnerability point because it contains all data [24].

A decentralized architecture, also known as the Peer-to-Peer architecture, does not rely on a central server. For this architecture, each client communicates directly with the others in the network [23]. Each client trains its model locally and updates its model using information from other clients [5]. In this way, all data remain localized on individual devices, reducing the risk of data exposure. However, designing a decentralized architecture can be challenging; coordinating communication and synchronization among multiple devices can become more complex to implement [13].

2.4. Machine learning models

Medical data collected from different hospitals may have different distributions due to variations in the demographics of patients and treatment protocols [25]. Also known as Non-Independent and Non-Identically Distributed (non-IID) data, dealing with this type of data is challenging [24,26]. As such, research is imperative for handling this type of data. Parametric and non-parametric ML models are often used in FL. Due to the differences in their training mechanisms, both types of model exhibit distinct behavior when dealing with non-IID data, resulting in different performance.

2.4.1. Parametric models

Non-IID data can affect the performance of parametric models, especially in HFL systems due to label distribution imbalances. It can cause a divergence between the local model and the global model [27]. Neural networks (NN) are widely adopted parametric models, due to their amazing performance in many areas, among them image classification and speech recognition [28,29]. Linear models such as linear regression and logistic regression are also commonly used in FL studies because they are relatively easier to implement. However, due to the simplicity of the models, individual private data is more likely to be reverse engineered or leaked [30].

2.4.2. Non-parametric models

Non-parametric models such as Decision Trees, Gradient Boosting Decision Trees (GBDT), and Random Forest are also commonly used in the field of FL. Because of their good performance in classification and regression tasks. GBDT has especially become popular in both HFL and VFL systems [31]. However, a potential downside is that these models can be computationally intensive.

2.5. Federated learning algorithms

There are various FL algorithms designed to aggregate the local parameters: FedAvg, FedProx, SCAFFOLD, FedMedian, FedOpt, FedYogi and more [13,32–34]. Federated averaging, or FedAvg, is one of the most widely adopted and straightforward algorithms, which works by computing averages of the weights of local models by multiple clients. FedProx addresses the challenges presented by non-IID data by limiting local changes and has been shown to be effective in privacy protection [33]. FedProx extends FedAvg by introducing a proximal term μ that can help stabilize the training process. FedYogi optimizes the training process by addressing common challenges in federated learning, such as communication efficiency, model convergence, and robustness to data heterogeneity among different clients. Each algorithm possesses unique properties, and this research aims to determine the most suitable one for the module [35].

2.6. Data types

There are various types of data used in machine learning, but tabular data and image data stand out due to their widespread applications. Understanding how these two types of data behave in the context of federated leaning can provide a deeper insight into their strengths and challenges in the medical field.

Image data has been favored in FL because of its complexity and rich information content. It consists of pixel values arranged in grids, making it inherently high-dimensional and unstructured. Convolutional Neural Networks (CNNs) are the primary models used for image data because they efficiently capture spatial hierarchies through convolutional and pooling layers. However, the large size of the image data imposes significant computational and communication costs. It should be noted that ‘image’ data is a broad category, covering different modalities such as ultrasound, MRI and X-ray, each with distinct characteristics and processing needs.

Tabular data structured into rows and columns is common in the field of healthcare. In FL, tabular data presents some challenges. One major issue is the non-IID nature of data across different clients. This heterogeneity can complicate the training process and requires specialized aggregation methods to ensure that the global model performs well across diverse datasets. Although neural networks can be used for tabular data, traditional models often perform better. The structured nature and typically smaller size of tabular data compared to image data result in lower computational and communication costs.

2.7. TabNet

TabNet, is a new approach designed to handle tabular data. Traditional deep learning models often struggle with tabular data due to overparameterization and lack of appropriate inductive biases [16]. Typical deep learning models can learn from relational patterns from images or text, which is not always the case for features in tabular data. TabNet, on the other hand, utilizes a unique approach that combines the strengths of tree-based learning and deep neural networks. This hybrid method not only improves the model performance but also enhances interpretability by calculating the importance of the features. This process relies on several key components for the architecture to function. Similarly to how a Decision Tree selects a feature, the feature transformer and attentive transformer work together to select and process features at each decision step. The feature transformer processes the input features, while the attentive transformer generates masks that highlight the most salient features. By focusing only on these selected features, TabNet effectively identifies the decision boundaries on the manifold. This approach is particularly well-suited for tabular data with sparse characteristics. Similarly, TabNet’s Encoder operates by using the output of the previous Encoder as feedback to update the feature masking for the next Encoder. This structure functions as an ensemble of encoders, mirroring the ensemble of trees in a tree-based model, allowing for progressive refinement and improved learning outcomes.

2.8. DeepInsight

Continuing the theme of using the capabilities of deep learning methods to train tabular data, we encountered a variety of innovative techniques during our research. One such method is DeepInsight, a technique that transforms tabular data into images, enabling the application of deep learning models used in image processing [17]. The first step is normalizing the tabular data to ensure that all features are on a comparable scale, crucial for accurately representing feature values as pixel intensities in images. The features are then mapped to a 2D grid using dimensional reduction techniques such as t-SNE (t-Distributed Stochastic Neighbor Embedding) or Principal Component Analysis (PCA). When transforming the high-dimensional data into a 2D grid (image), each feature is assigned a pixel location. This process can sometimes lead to multiple features being mapped to the same pixel, causing collisions. These collisions can degrade the quality of the transformation and the subsequent model’s ability to accurately learn and generalize. After mapping, each row of the tabular dataset is converted into a corresponding image, where each pixel value reflects the normalized value of a specific feature for that sample. This creates a visual representation of the tabular instance, allowing CNNs to process the data using their architectures. As mentioned earlier, CNNs with their hierarchical feature extraction capabilities can learn complex patterns and interactions within the data, potentially leading to improved predictive performance compared to traditional methods. Although this transformation enables effective learning, it complicates interpretability since the spatial structure does not directly reflect the original feature layout.

3. Methodology

3.1. Data

For the image dataset, we decided to use a COVID-19 radiography dataset containing X-ray images of the chest area. This dataset includes three classes: positive cases of COVID-19, normal cases, and viral pneumonia cases. The dataset consists of a total of 317 X-ray images, with 66 designated as test images and 251 as training images. This dataset was chosen because it is highly relevant for current medical challenges, includes diverse and high-quality images, and is publicly accessible on kaggle [36].

In the field of Federated Learning, little research has been done on tabular data compared to image data [37]. To address this gap, we used our models on different types of tabular data. Using Supabase (Supabase LLC, USA, https://supabase.com/), we extracted and downloaded the tabular data in CSV format. The first dataset is a breast cancer dataset obtained from the UCI Machine Learning Repository. This dataset is derived from digitized images of fine needle aspirates (FNA) of breast masses, describing the characteristics of the cell nuclei present in the images, resulting in a dataset with 30 features. The class distribution includes 357 benign cases and 212 malignant cases. This data set was chosen due to its relatively small size, balanced class distribution, and the presence of numerical values in all column of characteristics [38]. In the preprocessing steps, we removed all white spaces from the columns names to prepare the dataset for TabNet. Additionally, we label encoded the classification column.

For the second tabular dataset, our objective was to predict stroke occurrences based on various parameters of the patient. This dataset is particularly relevant given the high global impact of stroke, as highlighted by the World Stroke Organization (WSO), which states that stroke is the second leading cause of death globally, responsible for approximately 11% of total deaths [39]. The dataset comprises 12 columns, of which only two contain continuous values. The remaining columns are binary or categorical. The ‘bmi’ column had many missing values, so we replaced them with the median value within each age category. Each category was defined by a 5-year age range. Additionally, the categorical columns got encoded. The original dataset is highly imbalanced, with 249 stroke cases out of 5110 total cases. To address this imbalance, we employ SMOTE techniques to generate synthetic samples during pre-processing. This approach has also been explored in some federated learning studies to handle imbalanced data across distributed clients [40]. A preliminary evaluation comparing the original dataset and the SMOTE enhanced version is provided in Appendix C.2, demonstrating improvements in precision, recall and F1-score, which guided our decision to incorporate SMOTE in our pipeline. All the tabular datasets mentioned were divided into an 80/20 train-test split.

3.2. Approach

In terms of our approach, we examined the key components needed for designing an FLM through simulation. We selected the pathways that align closely with our design. For example, consider two pathways: Pathway 1, an FL architecture without DP using FedAvg for parameter aggregation, and Pathway 2, an architecture with DP using FedProx for aggregation. From the literature, we assume that Pathway 1 will have a high accuracy performance but is likely to leak patient data. However, Pathway 2 is less likely to leak patient data, but accuracy will suffer significantly [13,26,30]. If the initial selection did not meet our requirement, we proceeded to the next pathway. Through trial and error, we refined and identified the optimal FLM design adapted to medical data. All of our experiments were conducted using Horizontal Federated Learning (HFL) because it simplifies the process and is well supported by the frameworks we are using. Our priority is to select a flexible framework that offers security measures. In addition, it should be able to support different ML models and be scalable to ensure smooth implementation in a real-world setting. Furthermore, the framework should offer clear documentation that facilitates easier adoption by other parties. Based on these requirements, we have chosen the appropriate framework and use the Snellius supercomputer (SURF, Amsterdam, Netherlands; https://www.surf.nl/en/dutch-national-supercomputer-snellius) to run intensive tasks such as image training.

3.3. Security method

The primary objective is to improve privacy and security. As mentioned earlier, we are integrating Differential Privacy and other novel methods into the module. For this investigation, we used central differential privacy because it provides a good balance between privacy and utility. With the central differential privacy algorithm, we can choose whether to use server-side or client-side cutting in the first phase [41]. Each approach has its pros and cons. The first approach has the advantage of enforcing a uniform clipping on all clients and reducing communication overhead for clipping values. However, it increases the computational load on the server. The second approach reduces computational load, but there is a lack of centralized control. We decided to use server-side clipping with adaptive clipping because it allows for consistent and uniform application of clipping across all clients. Adaptive clipping dynamically adjusts the clipping values according to the data distribution, which can lead to better model performance and more efficient privacy management.

In the paper by Andrew et al. which this algorithm is based, they used noise multipliers (z) between 0.0 and 0.1 [42]. The results showed that the performance of the model begins to decrease significantly with values greater than $z = 0.1$. As such, we decided to use the smallest noise multiplier after 0 and the largest noise multiplier in the paper. Thus, we experimented with $z = 0.01$ and $z = 0.1$, to get a sense of a lower and upper bound. If the model demonstrates good performance, we consider adjusting the upper bound to further optimize the results. While a full grid search or sensitivity analysis would provide a more exhaustive view, our goal in this phase was to establish whether the model retains acceptable performance under practical differential privacy settings. These two values were selected to represent meaningful extremes within the empirically validated range, offering initial insight into the algorithm’s robustness. If promising results are observed, we consider these experiments a starting point for more granular tuning in future work. The other parameters of the algorithm were left at their default values as recommended by Andrew et al. All runs with differential privacy were performed using FedAvg, the algorithm is simple yet highly effective. Its straightforward approach makes it an excellent baseline for evaluating differential privacy methods and can serve as a starting point for researchers.

3.4. Model selection

3.4.1. CNNs

Selecting an appropriate ML model for FLM is a critical step in ensuring the effectiveness and performance of the module. Our module allows facilities to train various ML models based on the datasets. For the image dataset we chose CNNs due to their proven success and efficiency in handling image data. It is also possible to use a pre-trained model, according to Kieffer et al. pre-trained models typically achieve higher accuracy than those that are not pre-trained [43]. To maintain simplicity, we implemented a standard CNN model consisting of 2 convolutional layers, 2 max pooling layers, and 3 fully connected layers. We set the number of *local epochs* to *5*. As for FedDeepInsight, we used the SqueezeNet 1.1 architecture, consistent with the original paper, to train the converted images. We will discuss its integration and performance in greater detail in a later section.

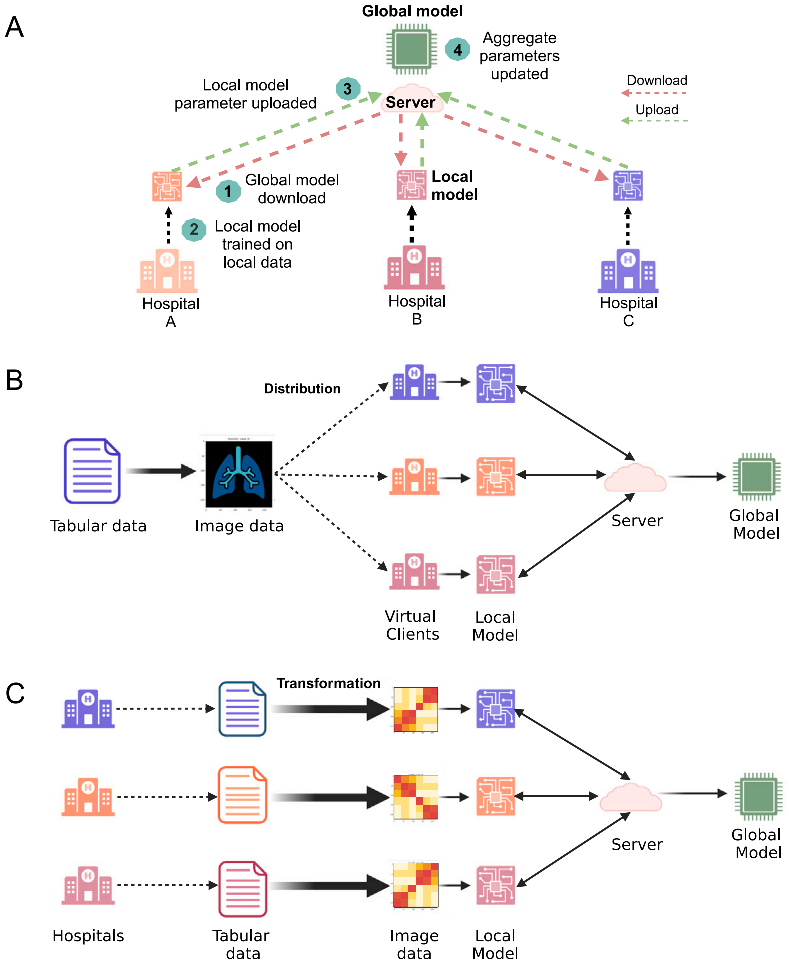

Fig. 1. Federated learning approaches and FedDeepInsight. (**A**) Cross-SILO Federated Learning. (**B** and **C**) FedDeepInsight Simulated Scenario vs. Real-Life Scenario Comparison, respectively.

3.4.2. *Logistic regression*

As for the tabular datasets we chose four different models for the investigations. Two traditional ML-models and two novel approaches. The chosen models are Logistic Regression (LR), XGBoost (XGB), TabNet, and DeepInsight. We chose LR for its simplicity and interpretability, linear models provide a strong baseline for comparison [44]. Regarding the hyper parameters, we chose *L2 regularization* as *penalty* to help prevent overfitting. We enabled the *warm_start* parameter to *True* to retain the previous weights and avoid reinitializing them.

3.4.3. *XGBoost*

XGB was selected for its high performance and robustness; it is known for outstanding results in tabular data. For the XGB model, we used NVFlare to carry out the investigations. The XGB model in NVFlare offers many options, such as data splits(uniform, linear, exponential, squared) and tree-based(cyclic, bagging) training. After investigating all the different configurations, we chose a *uniform data split* with the *bagging* model. We used a uniform data split to maintain consistency with the other models and use bagging because of its high accuracy.

3.4.4. *TabNet*

The new TabNet approach with the use of attention mechanisms and interpretable embeddings for feature selection allows for end-to-end learning, so the user can directly handle raw data, which can reduce preprocessing needs. This can save time and simplify the workflow of a researcher. We used the TabNet TensorFlow implementation [16]. For the hyperparameters, we applied the settings provided in the GitHub example. The only changes we made were dataset-specific parameters, such as the list of column names and the number of classes.

3.4.5. *FedDeepInsight*

Finally, we chose DeepInsight for its innovative method of converting tabular data into images, enabling the use of CNNs to uncover complex patterns and relationships that traditional models might miss. To develop our method, FedDeepInsight, we implemented several steps. In the original method, when the entire dataset is transformed at once, the DeepInsight image transformer captures the global structure of the data. This results in a consistent mapping where relationships and variances between all data points are considered; see Appendix B.1. If the dataset is split into parts and each part is transformed separately, the image transformer only captures the local structure of each part. This can lead to different mappings, as the relationships between data points in one part are not seen in the context of the other part. The images generated from separate transformations may not be consistent, affecting the ability of the model to generalize.

We reached out to the authors to inquire whether it was possible to minimize the loss caused by feature-to-pixel mapping collisions. They suggested that we could try changing the discretization method, even though it had not been tested. Discretization methods are used to map features (data points) to specific pixel locations. We decided to experiment with Linear Sum Assignment (LSA) and Coordinate Binning (BIN) as discretization methods. An in-depth explanation of these methods can be found in the source code of DeepInsight's image transformer. Subsequently, we examined the source code of two-dimensionality reducers: t-SNE and PCA. The first reducer was used in their GitHub example, and upon examining the t-SNE source code, we observed that some randomness might still be involved even when setting a random state seed for reproducibility. This is important because in theory small variations can generate completely different images. In centralized training, this variability is acceptable. However, in a federated learning environment, it is crucial that each client has the same parameters. As such we decided to look into linear dimensionality reducers instead, such as PCA, which were not investigated in the original paper [17].

Regarding the process of incorporating DeepInsight into our FL pipeline, the initial steps were similar to the original approach. First, we normalized the data. Then, we created a reducer object for the image transformer. As mentioned earlier, we experimented with t-SNE and PCA reducers. Starting with t-SNE, the *perplexity* hyperparameter was set to the *number of features in the dataset minus one*. The other settings remained the same as in their example. For PCA, we set *n_components* to *2*. For both reducers, the *random state* was set to *42*. Subsequently, we initialized the image transformer with the reducer and set the *pixel size* to *227 × 227*. The image transformer was then trained using the training data. Once the transformer was trained, it was applied to both the training and test datasets to convert the tabular data into images. Examples of transformed images are illustrated in Appendix B.2. The transformed images were then converted into tensors. Tensor data was trained with the SqueezeNet 1.1 model in our federated learning pipeline [17].

3.5. FL algorithms selection

FedAvg is a widely used algorithm and operates by averaging the model updates of the clients; as a foundational algorithm, it serves as a standard benchmark in this study to compare it with other algorithms. FedProx addresses data heterogeneity among clients and can mitigate non-IID data by incorporating the proximal term. We conducted investigations with (μ = 0.1, 1, 2), similar values were used at the Flower baseline [41]. Finally, we decided to use FedYogi because it has demonstrated superior performance on several benchmarks, often outperforming other federated optimization algorithms in terms of accuracy and convergence speed.

3.6. Evaluation

We evaluated the performance based on several factors including accuracy per round, the impact of increasing the number of clients on model performance, and the convergence behavior of the model under different configurations. Specifically, we analyze how the accuracy evolves with each training round and assess the convergence rate and stability of the model as more clients participate in the training process. As for our new method, FedDeepInsight, we also need to evaluate its viability in real-life settings. In a simulated federated learning environment Fig. 1B, the tabular data to image transformation occurs centrally, meaning that the transformation of the train set and the test set is performed before distributing them to virtual clients. This approach is not feasible in real-life settings and defeats the purpose of federated learning. The process in a real world scenario is illustrated in Fig. 1C. Earlier, we discussed the randomness that a dimensionality reducer like t-SNE can introduce and how the mappings can differ if the dataset is split and transformed separately. To evaluate the extent of this randomness, we conducted model training on two reducers: t-SNE and PCA. First, we applied the t-SNE and PCA transformations to the train set and trained the data in our FLM, saving the trained model from each round. After model training, we used the saved global models to make predictions on synthetic datasets generated from the two original datasets. By doing this, we can evaluate the predictions from the synthetic dataset. Our aim is to see how well the global FedDeepInsight model can generalize to new data with the same format and hyperparameters, simulating real-world scenarios. If the accuracy remains consistent, it indicates minimal impact of randomness; if not, this would highlight the challenges posed by its randomness and underscore the need for careful consideration when choosing dimensionality reduction techniques.

To this end, we included the trustworthiness score as a complementary metric, measuring how well the local structure of the high-dimensional data is preserved in the lower-dimensional space [45]. A high trustworthiness score indicates that the neighborhood relationships are maintained, which is particularly important in federated learning scenarios where each client may operate on only a small local subset of the data. In addition to local structure preservation, we also evaluated the preservation of global structure using Pearson and Spearman correlation scores between pairwise distances in the original and reduced spaces [46]. Spearman captures monotonic relationships and is better suited for assessing structural similarity in nonlinear embeddings such as t-SNE. While Pearson evaluates linear correspondence and is often more informative in linear methods like PCA. By incorporating these metrics into our evaluation, we gain a clearer understanding of how the choice of dimensionality reducer impacts not just accuracy, but also the consistency and reliability of data representation across clients.

4. Results

4.1. Image data

In this section, we present the performance analysis of three federated learning algorithms: FedAvg, FedProx and FedYogi on the COVID-19 X-ray dataset (Fig. 2A). The accuracy of FedAvg shows a steady increase initially, stabilizing around the 100th round. The accuracy remains consistently high and stable throughout the remaining rounds, with minor fluctuations around the 85%–90% range for both 5 & 20 clients. FedProx is less stable, showing significant fluctuations throughout the rounds. Based on the density plot in Appendix A.2, we chose μ = 0.1 as the proximal term. Despite initial improvement, accuracy experiences frequent and pronounced drops. The accuracy varies widely, generally between 60% and 90%. This is the case for both client configurations. The FedYogi algorithm also shows substantial variability, although it generally stabilizes after the initial rounds. The accuracy with 5 clients fluctuates between 70% and 90%, with some notable peaks reaching around 93%. With 20 clients, FedYogi shows improved stability compared to the 5-client configuration. Although there are still fluctuations, they are less pronounced, and the accuracy generally stabilizes around the 80%–90% range after the initial rounds. In the density plot of Fig. 2A, we can observe the differences in the accuracy distribution of the three federated learning algorithms. FedAvg has the highest density, indicating that it is the most reliable and stable algorithm in this context, consistently providing high accuracy with minimal fluctuations. FedProx, despite its potential, is hindered by significant instability, making it less suitable for consistent performance. FedYogi & FedProx offers high potential, but requires further optimization to ensure stability. It is evident from Appendix A.1 & A.2, that careful selection of the μ parameter can contribute to FedProx performance.

4.2. Tabular data

To address the mentioned research gap, our investigations with tabular data were more extensive, particularly with regard to FedDeepInsight. First, we compared the performances of the four proposed models. The performance of the models is illustrated in Fig. 2B & 2C. In the 5-client configuration with the cancer dataset, Logistic Regression (LR) initially increases rapidly, reaching around 90%. The accuracy stabilizes just after 50 rounds at around 88% with minimal fluctuations. The accuracy of TabNet also increases rapidly in initial rounds, stabilizing around 92% after approximately 150 rounds. Performance remains consistent with no fluctuations. The accuracy of FedDeepInsight shows significant fluctuations in the early rounds but decreases significantly after about 30 rounds. The model continues to fluctuate throughout the process, with the delta between the fluctuations decreasing while the accuracy increases, eventually reaching the heights of the XGB model

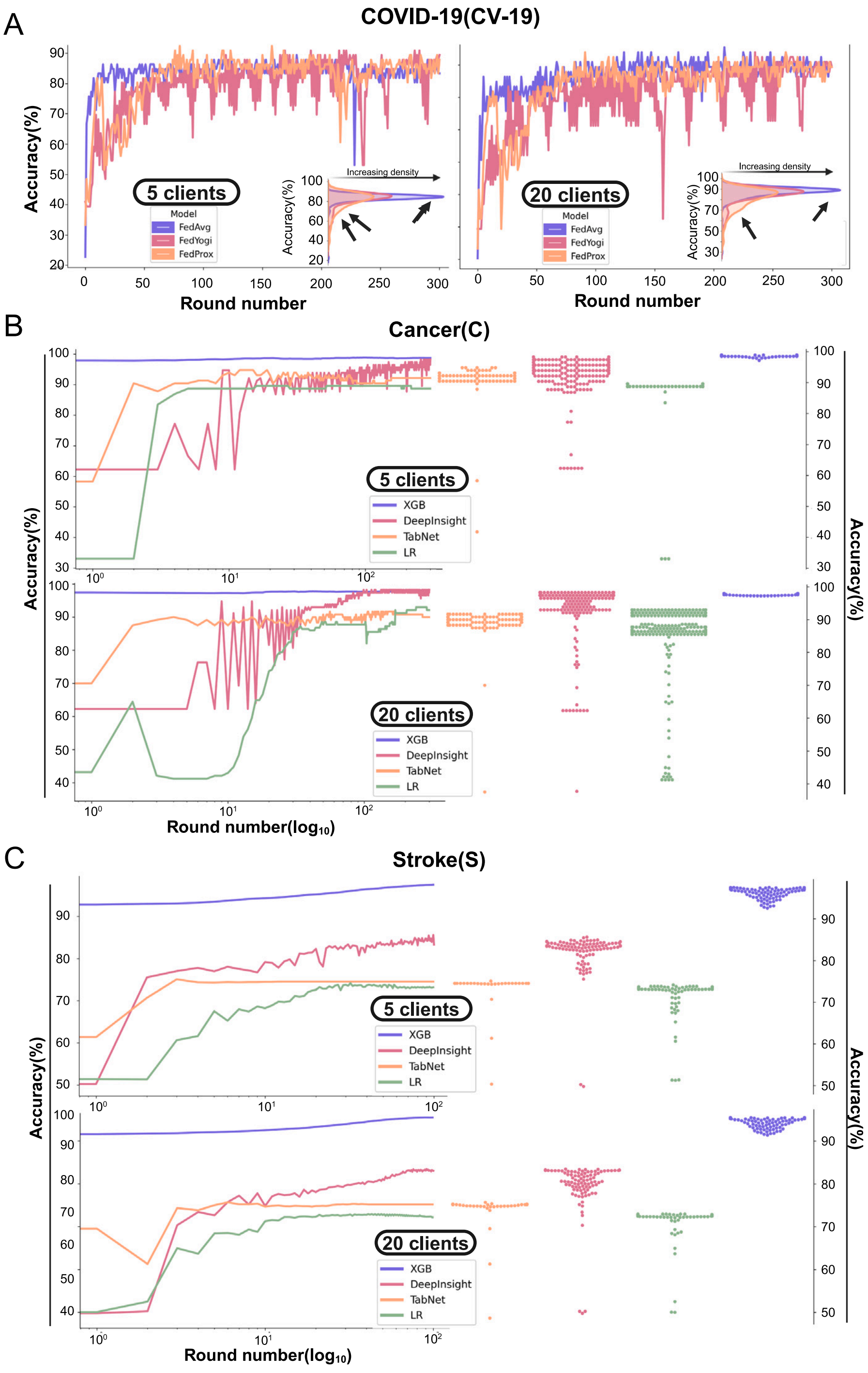

Fig. 2. (A) Accuracy over Rounds experiments with three FL-algorithms on the image dataset, where each color represents a different algorithm. The density plot in the lower right corner shows the distribution of accuracy for each FL algorithm. (B and C) Left side: Accuracy over rounds for various models on tabular datasets, with each color representing a distinct model. The round number is represented on a logarithmic scale. Right side: Swarm plot to illustrate the distribution of accuracy values for each model.

and outperforming the other two models. The accuracy of XGBoost (XGB) remains near-perfect, around 98%, throughout the rounds. For the stroke data set, the accuracy of the logistic regression has the lowest accuracy among all models at around 72%. The model converges after round 16 with minor fluctuations during the rest of the training process. The accuracy of TabNet achieves a higher accuracy than the previous model, reaching a peak around 75%. The model remains consistent throughout the training process with zero fluctuations. FedDeepInsight initially has the most unstable graph of all the models but quickly surpasses the previous models in terms of accuracy before the first 10 rounds, increasing to around 84% accuracy. XGBoost (XGB) model shows a steady increase in accuracy, maintaining around 95%, achieving the highest accuracy among all models. It is also the most stable model.

In the 20-client configuration for the cancer dataset, the accuracy of logistic regression shows a modest increase in the initial rounds compared to the others, stabilizing after 160 rounds with minor fluctuations between 90 and 93%. TabNet model experiences an initial rapid

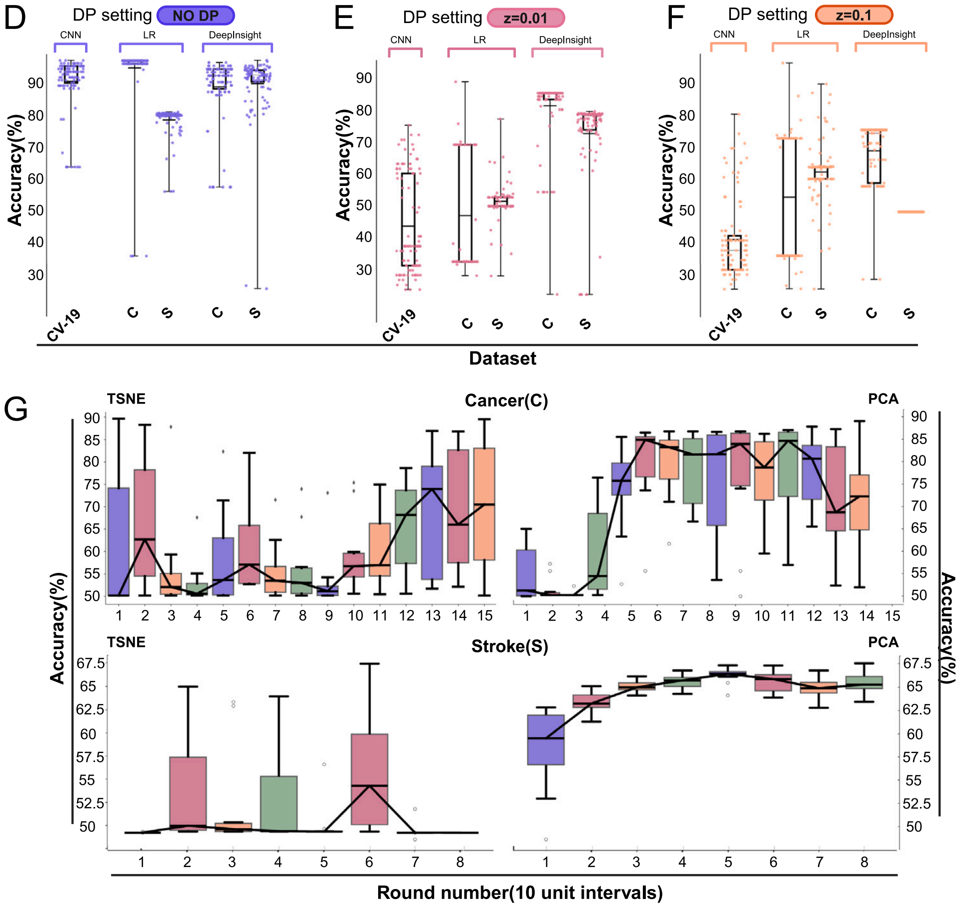

Fig. 3. (D, E, F) Experiments on implementing differential privacy in the models. Rain plots visualize the spread of accuracy distribution, with each color representing a different noise multiplier setting (z). (G) Accuracy over rounds performances for FedDeepInsight's saved model on synthetic datasets. The x-axis represents rounds, visualized in increments of 10 rounds. The colors in this plot are arbitrary and do not represent any specific variable.

increase, which also stabilizes after 160 rounds between 90%–91%. FedDeepInsight model shows significant early fluctuations, but the delta of the fluctuations becomes minimal just before round 50. The accuracy then climbs steadily, reaching 98%, outperforming XGB in some rounds. XGBoost model still maintains a near-perfect accuracy at 96% throughout the rounds, demonstrating consistent high performance with the best stability. For the stroke dataset, the logistic regression model stabilizes after 20 rounds around 72%. Performance remains stable with minimal fluctuations. TabNet model shows a rapid initial increase, stabilizing just before 20 rounds. Performance remains stable from round 40, with no fluctuations, and accuracy converges around 75%. FedDeepInsight model shows a gradual improvement in stability and accuracy over the rounds. It outperforms the two models mentioned above, achieving around 83% accuracy with minor fluctuations. XGBoost model demonstrates a steady increase in accuracy, with a value around 95%. Shows consistent and high performance throughout the rounds with no fluctuations.

4.2.1. Analysis

The cancer dataset comprises numerical columns and contains 569 rows. From the graph, it is evident that all models achieve high accuracy, but exhibit fluctuations and minor instability particularly within the first 100–150 rounds. The relatively small size of the dataset may contribute to the instability observed in model training. Furthermore, the numerical nature of the data likely contributes to the high performance seen in these models. However, the stroke dataset comprises mixed types, primarily categorical, and contains approximately 9700 rows. All models exhibit much greater stability with minimal fluctuations, indicating that a larger dataset contributes to the stability and convergence of federated learning training. In this dataset, XGBoost outperforms the other models, maintaining the highest accuracy. The mixed-type features in the dataset may contribute to the comparatively lower accuracy observed in the other models. In both datasets, XGBoost demonstrates exceptional performance and stability, making it a highly reliable model for both cancer prediction and stroke predictions. TabNet and logistic regression show consistent performance, but generally lower accuracy compared to the other two models. Our FedDeepInsight model performed competitively, at times surpassing XGBoost in the 20-client configuration. Although both TabNet and FedDeepInsight leverage deep neural networks, FedDeepInsight achieved higher peak accuracy in all situations, which may justify its use when maximum predictive performance is a priority especially in settings where deep architectures can be supported.

4.3. Differential privacy results

For the tabular data, we show the results of implementing DP with Logistic Regression and DeepInsight. For image data, we used convolutional neural networks (CNN). The results for both tabular and image data are presented in Figs. 3D, 3E, and 3F. This comparison allows us to observe how these models behave under DP and how various configurations influence the results. The performance of the logistic regression model under DP was poor across both datasets with the selected configurations. In each scenario, the models with DP exhibited high variability and did not show a clear improvement trend, with a mean hovering around the precision 50%. Furthermore, some configurations were abruptly stopped due to errors, preventing them from reaching the desired end-round. Based on overall performance, this would not have made a significant difference.

In CNN model investigations that compared FL algorithms, FedAvg demonstrated the most stable performance compared to FedYogi and FedProx. According to the study by Korkmaz et al. FedAvg also showed good overall performance, making it a reliable choice for federated learning on medical datasets [47]. Thus, we decided to conduct the DP investigations using FedAvg with a noise multiplier (z) of 0.1 & 0.01 . The accuracy from the model is distributed sporadically between 25% and 80%, with an average of around 45% for z = 0.01 and even lower for z=0.1. Given the instability observed in the preliminary training produced in Fig. 2A, similar trends were expected in the DP runs.

The DP implementation appears to work better with the DeepInsight model, demonstrating good accuracy across various configurations, unlike the previous models. Examining Figs. 3D, 3E, and 3F, the model using a noise multiplier of 0.01 shows that the cluster is concentrated around 88%. In contrast, in the stroke dataset, the same noise multiplier results in a cluster around 80%. A noise multiplier of 0.1 shows that the cluster is more spread between 60%–75% in the cancer dataset, while in the stroke dataset it is concentrated at 50%. Like the paper suggests: a lower noise multiplier might allow for better model performance but with less privacy [22].

4.4. Dimensionality reducers

In this section, we evaluate and compare PCA and t-SNE with respect to both structure preservation and model accuracy. First, we assess structure preservation by examining local and global fidelity. Local structure preservation is measured using trustworthiness scores across varying values of k [45]. As shown in Fig. 4 for the stroke dataset, t-SNE excels at preserving local neighborhoods (high trustworthiness at low k), but its performance declines rapidly as k increases, indicating weaker global consistency. In contrast, PCA exhibits more stable trustworthiness scores in k, suggesting better overall retention of the global structure of the data.

To complement this analysis, we computed Spearman and Pearson correlations between pairwise distances in the high-dimensional and reduced spaces (Table 1). PCA consistently outperforms t-SNE on both datasets, confirming its superior ability to preserve global relationships and supporting the trends observed in the trustworthiness curves.

Next, we examine how these structure preservation differences translate into downstream model accuracy. As discussed in the methodology, we are only interested in seeing how well the saved global models can generalize to a new dataset. Therefore, we ran the models up to the round where the preliminary accuracy in Figs. 2B & 2C plateaued. Based on the figures, this occurs at 150 rounds for the cancer dataset and 80 rounds for the stroke dataset. In the plots on the right side of Fig. 3G, PCA shows greater stability and a consistent improvement in accuracy throughout the rounds compared to t-SNE on the left side. The variability in the t-SNE boxplot indicates that the model performance is more sensitive to the randomness introduced by t-SNE. Models using PCA have an upward trend suggesting that the model can learn and generalize better with each round, which is less evident in t-SNE boxplots. The PCA reducer on the stroke dataset exhibits more stability and fewer outliers primarily due to the larger and more diverse dataset, which allows the model to generalize better. We also experimented with a different discretization method (LSA) as suggested by the authors, but the results were not as good. Therefore, we decided to stick with the default mapping setting (BIN). For more details, see Appendix B.3.

Table 1
Spearman and Pearson correlation scores for t-SNE and PCA on Stroke and Cancer datasets.

Dataset	Metric	t-SNE	PCA
Stroke	Spearman	0.0640	0.7637
	Pearson	0.0897	0.8806
Cancer	Spearman	0.7720	0.9730
	Pearson	0.7830	0.9896

5. Discussion

5.1. Trade-offs between privacy and utility

Based on the results, we found that DP can have a significant negative impact on the performance of the models, especially with traditional machine learning models. Not only does DP lower the accuracy of the models, but it also causes instability within some of the models resulting in extreme peaks and troughs. For this study, we adopted server-side differential privacy with adaptive clipping, a method in which noise is added after aggregating client updates. This approach is appealing due to its relative ease of deployment and its tendency to preserve model accuracy more effectively than other DP configurations. However, it involves important trade-offs. One key trade-off is the trust assumption: server-side DP relies on the server to honestly apply noise and discard raw updates. In real-world applications with stricter privacy requirements, this assumption may not be acceptable. More fundamentally, server-side DP embodies the classic trade-off where stronger privacy requires injecting more noise, which can hinder learning by reducing accuracy, slowing convergence, and introducing instability. While adaptive clipping mitigates some of these effects, hyperparameter tuning remains difficult, as the ideal noise level varies across datasets and model architectures. Furthermore, deploying DP at scale introduces additional computational and communication overhead, particularly when integrated with secure aggregation or cryptographic enhancements. Consistent with findings by Geyer et al. it is possible we used an insufficient amount of participants [48]. FedDeepInsight model performed well; however, as expected, the accuracy of the DP models dipped. Using a noise multiplier of 0.01 in this model resulted in an acceptable accuracy range. For the image model, the outcomes were less predictable. Although we anticipated similar results to the FedDeepInsight model due to the use of CNN architecture, the preliminary results shown in Fig. 2A showed significant fluctuations during training. These results highlight the need for careful design when applying differential privacy in federated learning. Balancing privacy and model utility remains challenging, especially in real-world settings. This raises the question of whether the privacy trade-off is justified when risks are low, but accuracy suffers. Ultimately, deploying differential privacy should depend on the specific privacy needs and operational context.

5.2. Model selection and performance analysis on image and tabular data in federated learning

Training the image data, FedAvg demonstrated overall better performance, but FedYogi showed potential by achieving the highest accuracy in some rounds. We believe that with tuned hyper parameters, as suggested by Reddi et al. we could have achieved a more stable and better performing model [35]. FedProx performed substantially worse than the other two FL algorithms in terms of convergence and model stability. However, we believe that optimizing the proximal term (μ) in the dataset with a grid-search, rather than choosing an upper and lower bound, as illustrated in Appendix A.1 & A.2, might have resulted in a different outcome.

For our model selection, we used CNNs to train the image data. The graph was less stable than we initially anticipated, which can be

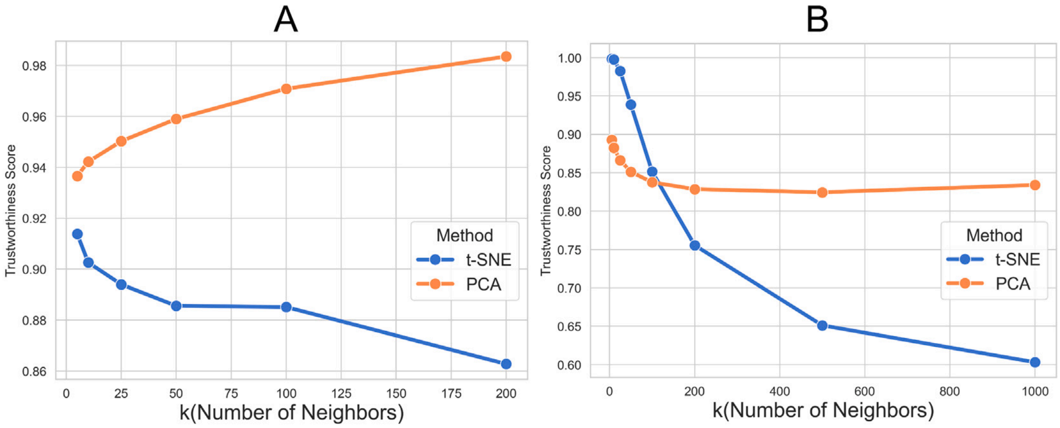

Fig. 4. Trustworthiness score PCA vs t-SNE for Cancer (A) and Stroke (B) datasets.

attributed to the smaller dataset we used for training. Furthermore, we used a simple CNN model to train the image dataset, using a pre-trained or sophisticated model may improve the overall performance. Based on the literature and the results of the image data, we decided to do all tabular data investigations with FedAvg, which can serve as a baseline for the other FL-algorithm configurations [47].

For tabular data, we used two types of datasets to test the capabilities of the proposed models. Starting with the cancer dataset, all models achieved high accuracy, above 90%. This high performance can be attributed to the fact that all columns, except the classification column, are numerical, which probably contributed to the model's effectiveness. When comparing the models in their 5- and 20-client configurations, the 5-client setup demonstrated more stability with fewer fluctuations, especially in the early rounds. In addition, these models achieved convergence much faster than their 20-client counterparts. Traditional machine learning approaches exhibited overall less fluctuation compared to novel approaches utilizing neural networks. We believe that this instability in the novel approaches is due to the relatively small size of the dataset. FedDeepInsight used SqueezeNet 1.1 CNN architecture. Comparing the results of FedDeepInsight and the CNN image, we observed that both models initially experience significant fluctuations. Although these fluctuations decreased as rounds progressed, they persisted throughout the training. However, when FedDeepInsight was applied to the stroke dataset, which is larger, the fluctuations in the graph were considerably smaller. This adds more validity to our hypothesis that neural network models achieve greater stability when trained with more data or using pre-trained models in the context of federated learning. Continuing with the stroke dataset, which comprises mostly categorical columns, the models appeared more stable and converged faster, hence the reason we only ran 100 rounds. XGBoost was the only model to achieve an accuracy in the 90% range. This suggests that XGBoost is also well suited for datasets with a higher proportion of categorical data.

We were initially unsure whether DeepInsight's ability to capture the global structure of tabular data would translate well in a federated learning environment. Since this is the first time this method is being used in an FL context, we devised a specific methodology to test its applicability. By communicating with the authors of the original study, we received valuable guidance in setting up the test methodology [17]. First, we attempted to change the mapping method to LSA as suggested, but this approach did not generalize well. As shown in Appendix B.3, the saved model did not generalize well even when using PCA as the reducer. In the study by Sharma et al. only nonlinear reducers such as t-SNE and k-PCA were used for the transformations [17]. Thus, we decided to test a linear reducer, PCA, to evaluate its performance in a federated learning environment. We discovered that the model with a PCA reducer could generalize well, but it requires that every detail, including the format of the dataset and the parameters of each client, be exactly the same when transforming the tabular data. Even slight discrepancies, such as an incorrect random state of a set of pixels, would likely result in completely different and ineffective images. The PCA model of the stroke dataset generally improved, as is evident in the right-side of Fig. 3G. The larger size of the dataset allowed more images to be transformed, providing the model with more training data. This consequently enhanced the model's generalization capabilities. As mentioned earlier, it is crucial that all the clients have consistent values in their settings. Ensuring that these values are kept hidden can add an extra layer of security and prevent potential data breaches mentioned by Kairouz et al. [24].

6. Conclusions

To meet our research objective, we conducted a comprehensive exploration of federated learning for medical data, focusing on both image and tabular data. Like an architect designing a building for functionality and safety, our blueprint outlines the key components and strategies needed to balance model performance with patient privacy. For image data, our results indicate that FedAvg is the most reliable aggregation algorithm, providing superior accuracy, stability, and convergence, and FedYogi is also viable with well-tuned hyperparameters. Although tested on a small dataset, we anticipate that larger datasets and pre-trained models could further improve performance [43]. For privacy protection, we recommend Differential Privacy (DP) with calibrated noise multipliers and initial upper and lower bounds for stability.

For tabular data, central DP is not recommended for traditional machine learning models due to performance constraints. However, it can be effective for neural networks when sufficient data are available. FedAvg is recommended as the baseline aggregation algorithm. Among the machine learning models tested, XGBoost (XGB) delivered the highest performance when DP was not applied. In contrast, our innovative FedDeepInsight approach demonstrated strong and consistent performance, excelling even in DP-enabled scenarios. Additionally, FedDeepInsight enhances security by protecting against attackers without access to model parameters.

Ultimately, FedDeepInsight emerges as a promising solution for secure, privacy-preserving federated learning in healthcare. Its potential to support collaborative, impactful medical research highlights its value in advancing the integration of federated learning into real-world medical applications.

6.1. Limitations & future work

Our study faced some minor limitations, most of which were due to hardware limitations. Firstly, working with image data posed significant challenges as larger datasets required extensive computing power.

To account for this, we propose to better parallelize the framework and explore minimization in communication costs. Federated learning is a relatively new concept, and our existing computing resources were not fully compatible with the demands of these larger datasets. Beyond technical challenges, practical limitations also impact the application of federated learning in clinical settings. Medical institutions may lack the computing power for local model training, and regulatory or network constraints may restrict model sharing and over-the-air updates. Furthermore, during our investigations with differential privacy, some models, such as Logistic Regression and TabNet, were fully compatible or functional.

Continuing with the theme of privacy, we plan to explore various forms of differential privacy, including local and client-side approaches. At this point, we can approach a possible hospital for testing and even possibly conduct a pilot study with patient data within the FAIR, GDPR compliance, and privacy guidelines required. In addition, we would like to perform experiments with cryptographic methods, such as homomorphic encryption [24]. These methods can enable computations on encrypted data without revealing raw information. Another topic of interest would be testing the limits of the models we used to train tabular data. Introducing sparse datasets and high-dimensional datasets will help us understand how these models handle data with many empty values or a large number of features. We anticipate that these experiments will reveal the strengths and weaknesses of the models in dealing with different types of complex data structures. For example, the generalizability of the DeepInsight model with a linear reducer like PCA might change depending on the dataset. Finally, in the future, we plan to evaluate these models using various metrics beyond accuracy, including precision, recall, and F1-score, as was done in the paper by Wilding et al. which use similar metrics to assess ultrasound images [49].

CRediT authorship contribution statement

Allan G. Duah: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation. **Roland V. Bumbuc:** Writing – review & editing, Visualization, Supervision, Software, Resources, Funding acquisition, Formal analysis, Data curation, Conceptualization. **H. Ibrahim Korkmaz:** Writing – review & editing, Validation, Formal analysis. **Rory Wilding:** Writing – review & editing, Validation, Resources, Formal analysis. **Vivek M. Sheraton:** Writing – review & editing, Supervision, Software, Project administration, Funding acquisition, Formal analysis, Conceptualization.

Ethical statement

The authors declare that no clinical or laboratorial experiments were performed that need ethical intervention and approval.

Data and code availability

The code and data required to reproduce our experiments are registered on Zenodo [50].

Funding

The authors declare that financial support was received for the research, authorship, and/or publication of this article. This research was partly funded by the Dutch Burns Foundation (DBF) – Health-Holland, Beverwijk, The Netherlands, project PPS 22.01 and partly funded by MetaHealth project number NWA.1389.20.080 of the research program Dutch Research Agenda - Research along Routes by Consortia (NWA-ORC) 2020/21 which is (partly) financed by the Dutch Research Council (NWO), The Netherlands.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Roland V. Bumbuc reports financial support was provided by Health Holland. Vivek M. Sheraton reports financial support was provided by Dutch Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is a follow-up study from a Master Thesis entitled “A Privacy-First Federated Learning Architecture for Medical Data”.

Appendix A. FedProx configurations

A.1. Line plot - Proximal term configurations

See Fig. A.5.

A.2. Density plot - Proximal term configurations

See Fig. A.6.

Appendix B. FedDeepInsight

B.1. 2D-grid feature-to-pixel map

See Fig. B.7.

B.2. Transformed images

See Fig. B.8.

B.3. Discretization method (LSA)

See Fig. B.9.

Appendix C. Dataset evaluation

See Table C.2.

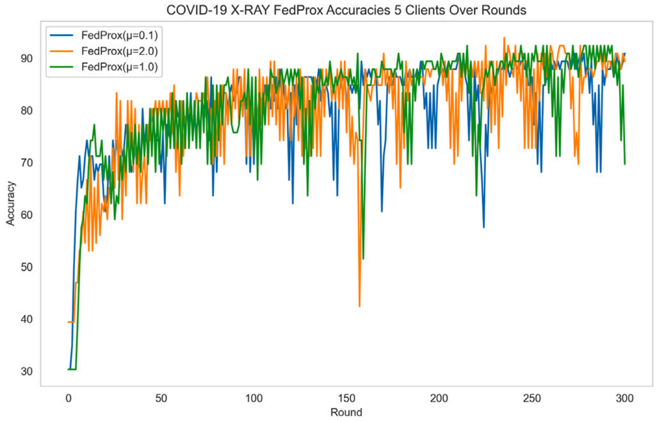

Fig. A.5. This plot illustrates the performance of three different proximal terms (μ) on the COVID-19 X-ray image dataset over multiple rounds. The proximal term settings are 0.1, 1, and 2, corresponding to the blue, green, and orange lines, respectively.

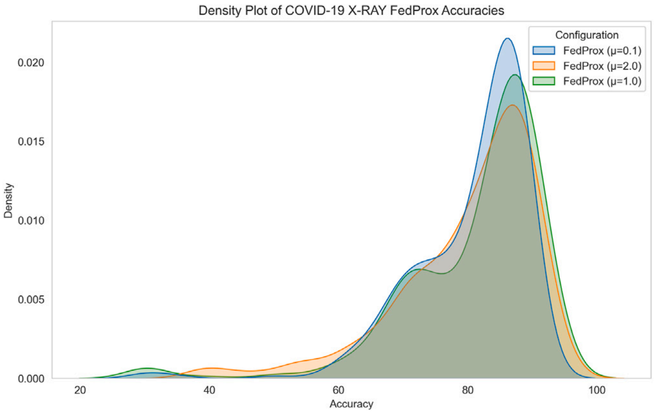

Fig. A.6. This plot illustrates the performance of three different proximal terms (μ) on the COVID-19 X-ray image dataset. The density plot allows us to see the differences in performance more clearly by showing the distribution of accuracy values.

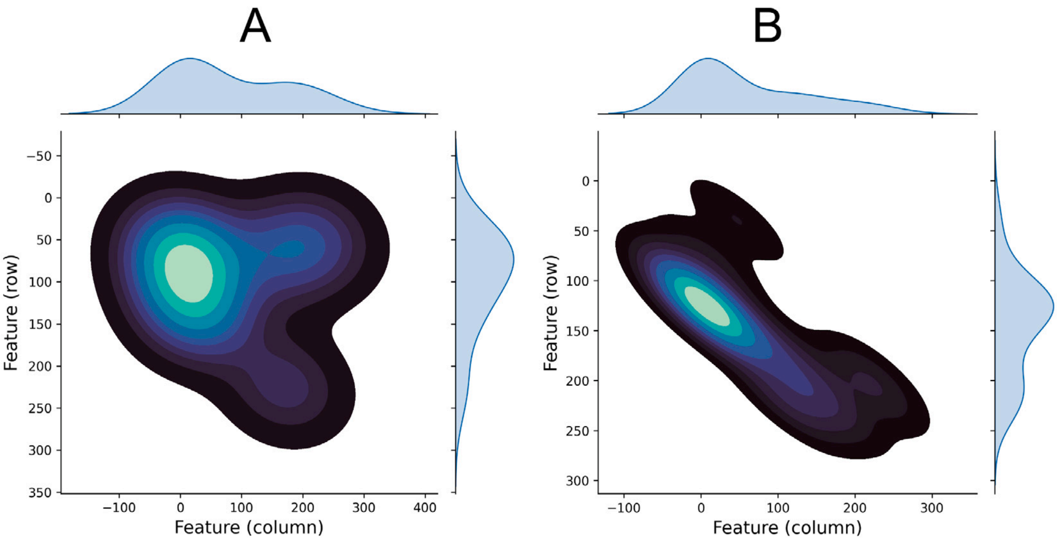

Fig. B.7. Feature-to-pixel kde plot for Stroke dataset (A) and Cancer dataset (B). The feature density matrix visualizes overall feature overlap by assigning each feature a pixel location. This transformation converts the high-dimensional data into a 2D grid, highlighting the position of pixels in the grid, lighter colors show higher density, while darker colors show decreasing density.

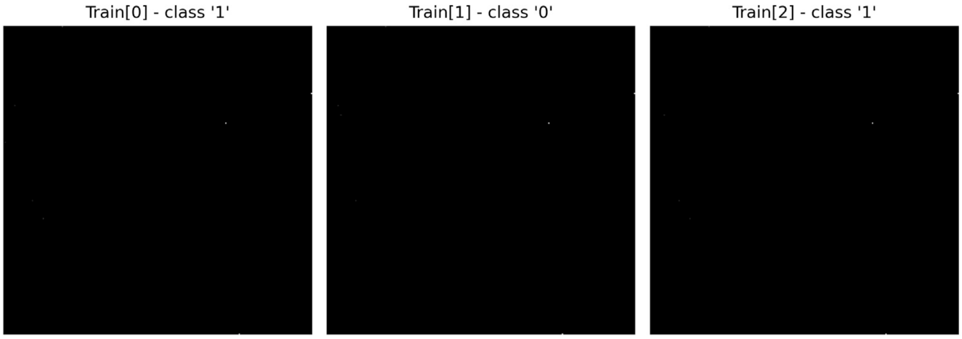

Fig. B.8. Constructed images where each pixel corresponds to a feature value, creating a visual representation of the tabular data. These images can be used with CNN architecture for model training.

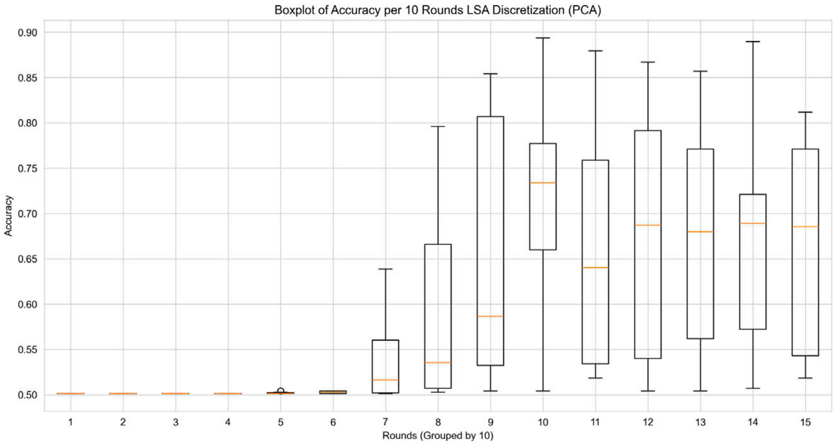

Fig. B.9. Boxplot - This plot illustrates the performance of the saved model on the synthetic cancer dataset using the LSA discretization method instead of the default BIN method. The model does not generalize as well with LSA compared to BIN. PCA was used as the dimensionality reducer.

Table C.2
Preliminary centralized evaluation comparing model performance with and without SMOTE. While the original dataset yields higher accuracy across all models, SMOTE significantly enhances performance on imbalanced classes. Logistic Regression (LR), which fails completely on the original dataset (precision and recall at 0%), shows strong gains with SMOTE. XGB improves dramatically in recall and F1-score, and DeepInsight benefits from balanced improvements in precision and recall. These results show SMOTE's effectiveness in improving model robustness under class imbalance.

Dataset	Model	Accuracy %	Precision %	Recall %	F1-score %
SMOTE	LR	82.00	80.00	84.00	82.00
SMOTE	XGB	95.00	92.00	97.00	95.00
SMOTE	DeepInsight	80.00	82.00	80.00	80.00
Original	LR	95.00	0.00	0.00	0.00
Original	XGB	95.00	22.00	39.00	6.00
Original	DeepInsight	95.00	48.00	50.00	49.00

References

[1] Chartrand G, Cheng PM, Vorontsov E, Drozdzal M, Turcotte S, Pal CJ, Kadoury S, Tang A. Deep learning: a primer for radiologists. Radiographics 2017;37(7):2113–31. http://dx.doi.org/10.1148/rg.2017170077.

[2] Rumbold JMM, Pierscionek B. The effect of the general data protection regulation on medical research. J Med Internet Res 2017;19(2):e47. http://dx.doi.org/10.2196/jmir.7108.

[3] European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council. 2016, URL https://data.europa.eu/eli/reg/2016/679/oj. [Accessed 25 June 2025].

[4] Marques JF, Bernardino J. Analysis of data anonymization techniques. In: KEOD. 2020, p. 235–41. http://dx.doi.org/10.5220/0010142302350241.

[5] Yin X, Zhu Y, Hu J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Comput Surv 2021;54(6). http://dx.doi.org/10.1145/3460427.

[6] Denton N, Molloy M, Charleston S, Lipset C, Hirsch J, Mulberg AE, Howard P, Marsh ED. Data silos are undermining drug development and failing rare disease patients. Orphanet J Rare Dis 2021;16(1):1–4. http://dx.doi.org/10.1186/s13023-021-01806-4.

[7] Dorst M, Zeevenhooven N, Wilding R, Mende D, Brandt BW, Zaura E, Hoekstra A, Sheraton VM. FAIR compliant database development for human microbiome data samples. Front Cell Infect Microbiol 2024;14:1384809. http://dx.doi.org/10.3389/fcimb.2024.1384809.

[8] Li L, Fan Y, Tse M, Lin K-Y. A review of applications in federated learning. Comput Ind Eng 2020;149:106854. http://dx.doi.org/10.1016/j.cie.2020.106854.

[9] Korkmaz HI, Sheraton VM, Bumbuc RV, Li M, Pijpe A, Mulder PP, Boekema BK, de Jong E, Papendorp SG, Brands R, et al. An in silico modeling approach to understanding the dynamics of the post-burn immune response. Front Immunol 2024;15:1303776. http://dx.doi.org/10.3389/fimmu.2024.1303776.

[10] Bumbuc RV, Korkmaz HI, van Zuijlen P, Vermeulen L, Sheraton VM. Understanding the dynamics of the proliferative phase in local burn wound healing: A computational model. In: 2023 IEEE international conference on bioinformatics and biomedicine. IEEE; 2023, p. 3676–83. http://dx.doi.org/10.1109/BIBM58861.2023.10385875.

[11] Yildirim V, Sheraton VM, Brands R, Crielaard L, Quax R, van Riel NA, Stronks K, Nicolaou M, Sloot PM. A data-driven computational model for obesity-driven diabetes onset and remission through weight loss. Iscience 2023;26(11). http://dx.doi.org/10.1016/j.isci.2023.108324.

[12] Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated electronic health records. Int J Med Inform 2018;112:59–67. http://dx.doi.org/10.1016/j.ijmedinf.2018.01.007.

[13] Li Q, Wen Z, Wu Z, Hu S, Wang N, Li Y, Liu X, He B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Trans Knowl Data Eng 2021;35(4):3347–66. http://dx.doi.org/10.1109/TKDE.2021.3124599.

[14] Papapanagiotou I, Bumbuc RV, Korkmaz HI, Krzhizhanovskaya V, Sheraton VM. From simulations to surrogates: Neural networks enhancing burn wound healing predictions. J Comput Sci 2025;102593. http://dx.doi.org/10.1016/j.jocs.2025.102593.

[15] Lindskog W, Prehofer C. A federated learning benchmark on tabular data: Comparing tree-based models and neural networks. In: 2023 eighth international conference on fog and mobile edge computing. IEEE; 2023, p. 239–46. http://dx.doi.org/10.1109/fmec59375.2023.10305887.

[16] Arik SO, Pfister T. TabNet: Attentive interpretable tabular learning. 2020, http://dx.doi.org/10.1609/aaai.v35i8.16826, arXiv:1908.07442.

[17] Sharma A, Vans E, Shigemizu D, Boroevich KA, Tsunoda T. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. Sci Rep 2019;9(1):11399. http://dx.doi.org/10.1038/s41598-019-47765-6.

[18] Zhang T, Mao S. An introduction to the federated learning standard. GetMobile: Mob Comp Comm 2022;25(3):18–22. http://dx.doi.org/10.1145/3511285.3511291.

[19] Li T, Sahu AK, Talwalkar AS, Smith V. Federated learning: Challenges, methods, and future directions. IEEE Signal Process Mag 2019;37:50–60. http://dx.doi.org/10.1109/MSP.2020.2975749.

[20] Nguyen DC, Pham Q-V, Pathirana PN, Ding M, Seneviratne A, Lin Z, Dobre O, Hwang W-J. Federated learning for smart healthcare: A survey. ACM Comput Surv 2022;55(3):1–37. http://dx.doi.org/10.1145/3501296.

[21] Xiong P, Zhu T, Wang X-F. A survey on differential privacy and applications. Chinese J Comput 2014;37:101–22. http://dx.doi.org/10.3724/SP.J.1016.2014.00101.

[22] Naseri M, Hayes J, C10063442ristofaro ED. Local and central differential privacy for robustness and privacy in federated learning. 2022, http://dx.doi.org/10.48550/arXiv.2009.03561, arXiv:2009.03561.

[23] Naik D, Naik N. An introduction to federated learning: Working, types, benefits and limitations. In: Naik N, Jenkins P, Grace P, Yang L, Prajapat S, editors. Advances in computational intelligence systems. Cham: Springer Nature Switzerland; 2024, p. 3–17. http://dx.doi.org/10.1007/978-3-031-47508-5_1.

[24] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R, et al. Advances and open problems in federated learning. Found Trends® Mach Learn 2021;14(1–2):1–210. http://dx.doi.org/10.1561/2200000083.

[25] Shae Z-Y, Chen K-Y, Chang C-Y, Tsai Y-Y, Chou C-Y, Baskett WI, Shyu C-R, Tsai JJ. Thoughts on non-iid data impact in healthcare with federated learning medical blockchain. In: 2022 IEEE 4th international conference on cognitive machine intelligence. IEEE; 2022, p. 20–6. http://dx.doi.org/10.1109/CogMI56440.2022.00013.

[26] Li Q, Diao Y, Chen Q, He B. Federated learning on non-iid data silos: An experimental study. In: 2022 IEEE 38th international conference on data engineering. IEEE; 2022, p. 965–78. http://dx.doi.org/10.1109/ICDE53745.2022.00077.

[27] Zhu H, Xu J, Liu S, Jin Y. Federated learning on non-IID data: A survey. Neurocomputing 2021;465:371–90. http://dx.doi.org/10.1016/j.neucom.2021.07.098.

[28] Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association, vol. 2012, 2012, p. 194–7, URL http://www.isca-archive.org/interspeech_2012/sundermeyer12_interspeech.pdf. [Accessed 25 June 2025].

[29] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges C, Bottou L, Weinberger K, editors. In: Advances in neural information processing systems, vol. 25, Curran Associates, Inc.; 2012, URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. [Accessed 25 June 2025].

[30] Lyu L, Yu H, Yang Q. Threats to federated learning: A survey. 2020, http://dx.doi.org/10.48550/arXiv.2003.02133, arXiv preprint arXiv:2003.02133.

[31] Li Q, Wen Z, He B. Practical federated gradient boosting decision trees. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, 2020, p. 4642–9. http://dx.doi.org/10.1609/aaai.v34i04.5895.

[32] Li X, Huang K, Yang W, Wang S, Zhang Z. On the convergence of fedavg on non-iid data. 2019, http://dx.doi.org/10.48550/arXiv.1907.02189, arXiv preprint arXiv:1907.02189.

[33] An T, Ma L, Wang W, Yang Y, Wang J, Chen Y. Consideration of FedProx in privacy protection. Electronics 2023;12(20):4364. http://dx.doi.org/10.3390/electronics12204364.

[34] Li Q, He B, Song D. Model-contrastive federated learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 10713–22. http://dx.doi.org/10.48550/arXiv.2103.16257.

[35] Reddi S, Charles Z, Zaheer M, Garrett Z, Rush K, Konečný J, Kumar S, McMahan HB. Adaptive federated optimization. 2021, http://dx.doi.org/10.48550/arXiv.2003.00295, arXiv:2003.00295.

[36] Raikot P. Covid-19 image dataset. 2020, kaggle dataset, URL https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset/data.

[37] Lindskog W, Prehofer C. A federated learning benchmark on tabular data: Comparing tree-based models and neural networks. In: 2023 eighth international conference on fog and mobile edge computing. 2023, p. 239–46. http://dx.doi.org/10.1109/FMEC59375.2023.10305887.

[38] WolbergWilliam SN, Street W. Breast cancer wisconsin (diagnostic). 1995, http://dx.doi.org/10.24432/C5DW2B, UCI Machine Learning Repository.

[39] Feigin V, Brainin M, Norrving B, Martins S, Sacco R, Hacke W, Fisher M, Pandian J, Lindsay P. World stroke organization (WSO): Global stroke fact sheet 2022. Int J Stroke 2022;17:18–29. http://dx.doi.org/10.1177/17474930211065917.

[40] Dolaat KMM, Erbad A, Ibrar M. Enhancing global model accuracy: Federated learning for imbalanced medical image datasets. In: 2023 international symposium on networks, computers and communications. 2023, p. 1–4. http://dx.doi.org/10.1109/ISNCC58260.2023.10323682.

[41] Beutel DJ, Topal T, Mathur A, Qiu X, Fernandez-Marques J, Gao Y, Sani L, Li KH, Parcollet T, de Gusmão PPB, et al. Flower: A friendly federated learning research framework. 2020, http://dx.doi.org/10.48550/arXiv.2007.14390, arXiv preprint arXiv:2007.14390.

[42] Andrew G, Thakkar O, McMahan B, Ramaswamy S. Differentially private learning with adaptive clipping. In: Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, editors. In: Advances in neural information processing systems, vol. 34, Curran Associates, Inc.; 2021, p. 17455–66, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/91cff01af640a24e7f9f7a5ab407889f-Paper.pdf. [Accessed 25 June 2025].

[43] Kieffer B, Babaie M, Kalra S, Tizhoosh HR. Convolutional neural networks for histopathology image classification: Training vs. Using pre-trained networks. In: 2017 seventh international conference on image processing theory, tools and applications. 2017, p. 1–6. http://dx.doi.org/10.1109/IPTA.2017.8310149.

[44] Lien D, Vuong Q. Selecting the best linear regression model: A classical approach. J Econometrics 1987;35:3–23. http://dx.doi.org/10.1016/0304-4076(87)90078-9.

[45] van der Maaten L. Learning a parametric embedding by preserving local structure. In: van Dyk D, Welling M, editors. Proceedings of the twelfth international conference on artificial intelligence and statistics. Proceedings of machine learning research, vol. 5, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR; 2009, p. 384–91, URL http://proceedings.mlr.press/v5/maaten09a/maaten09a.pdf. [Accessed 25 June 2025].

[46] Gildenblat J, Pahnke J. Preserving clusters and correlations: a dimensionality reduction method for exceptionally high global structure preservation. 2025, http://dx.doi.org/10.48550/arXiv.2503.07609, arXiv:2503.07609.

[47] Korkmaz A, Alhonainy A, Rao P. An evaluation of federated learning techniques for secure and privacy-preserving machine learning on medical datasets. In: 2022 IEEE applied imagery pattern recognition workshop. 2022, p. 1–7. http://dx.doi.org/10.1109/AIPR57179.2022.10092212.

[48] Geyer RC, Klein T, Nabi M. Differentially private federated learning: A client level perspective. 2017, http://dx.doi.org/10.48550/arXiv.1712.07557, arXiv preprint arXiv:1712.07557.

[49] Wilding R, Sheraton VM, Soto L, Chotai N, Tan EY. Deep learning applied to breast imaging classification and segmentation with human expert intervention. J Ultrasound 2022;1–8. http://dx.doi.org/10.1007/s40477-021-00642-3.

[50] Duah AG, Bumbuc RV, Korkmaz HI, Wilding R, Sheraton VM. A privacy-first federated learning architecture for medical data. 2024, http://dx.doi.org/10.5281/zenodo.14218577.