# Privacy-preserving patient clustering for personalized federated learning

**Ahmed Elhussein** AE2722@CUMC.COLUMBIA.EDU
*Department of Biomedical Informatics, Columbia University*
*New York City, NY, U.S.A*

**Gamze Gürsoy** GAMZE.GURSOY@COLUMBIA.EDU
*Department of Biomedical Informatics, Columbia University*
*New York Genome Center*
*New York City, NY, U.S.A*

## Abstract

Federated Learning (FL) is a machine learning framework that enables multiple organizations to train a model without sharing their data with a central server. However, it experiences significant performance degradation if the data is non-identically independently distributed (non-IID). This is a problem in medical settings, where variations in the patient population contribute significantly to distribution differences across hospitals. Personalized FL addresses this issue by accounting for site-specific distribution differences. Clustered FL, a Personalized FL variant, was used to address this problem by clustering patients into groups across hospitals and training separate models on each group. However, privacy concerns remained as a challenge as the clustering process requires exchange of patient-level information. This was previously solved by forming clusters using aggregated data, which led to inaccurate groups and performance degradation. In this study, we propose Privacy-preserving Community-Based Federated machine Learning (PCBFL), a novel Clustered FL framework that can cluster patients using patient-level data while protecting privacy. PCBFL uses Secure Multiparty Computation, a cryptographic technique, to securely calculate patient-level similarity scores across hospitals. We then evaluate PCBFL by training a federated mortality prediction model using 20 sites from the eICU dataset. We compare the performance gain from PCBFL against traditional and existing Clustered FL frameworks. Our results show that PCBFL successfully forms clinically meaningful cohorts of low, medium, and high-risk patients. PCBFL outperforms traditional and existing Clustered FL frameworks with an average AUC improvement of 4.3% and AUPRC improvement of 7.8%.

## 1. Introduction

The use of deep learning on Electronic Health Records (EHR) has been widely and successfully implemented for a range of goals such as for disease risk prediction, diagnostic support, and Natural Language Processing (Esteva et al. (2019); Gulshan et al. (2016); Miotto et al. (2016); Choi et al. (2016)). However, to leverage the predictive ability of deep learning models on the inherently high dimensionality of EHR data, a large number of samples are needed. Undersampled or overspecified models are more likely to overfit on training datasets and generalize poorly when applied to new datasets (Hosseini et al. (2020); Miotto et al. (2018)). This is especially important in rare disease settings where a single institution cannot have enough power to develop predictive models. One solution to

enable more sophisticated and accurate models is to increase available training data. While some attempts to build large and diverse cohorts have been successful (*e.g.,* All Of Us and UK Biobank), they are largely volunteer based and limited in the number and diversity of patients enrolled, with the majority of patients coming from healthy populations. Another alternative is institutional data-sharing but the regulatory framework (*e.g.,* HIPAA) and the ethical need to respect patient privacy limit widespread data sharing across institutions. One solution to support collaborative learning across sites while minimizing privacy concerns is Federated learning (FL) (McMahan et al. (2017); Kaissis et al. (2020)). FL is a distributed machine learning approach that enables multiple sites to collaboratively train a model while keeping data local. The process involves sites sharing locally trained model parameters with a central server, which then aggregates these parameters to create a global model. This process is repeated for a number of training rounds until a final global model is obtained. The parameters are aggregated via a commonly used algorithm, Federated Averaging (FedAvg), which uses sample-size weighted averaging to combine model parameters. FL enables model training on larger and more diverse patient groups across many sites while keeping datasets local. Furthermore, it has the benefit of allowing sites with limited training data (*e.g.,* rural hospitals) to be involved in model building. That is, FL has the potential to improve model performance, generalizability, and fairness (Rieke et al. (2020)). As such, FL has become increasingly popular in healthcare, and has been implemented on a range of tasks including disease risk prediction, diagnosis, and image recognition (Rieke et al. (2020); Dayan et al. (2021); Pati et al. (2022)).

FL still has some limitations. FedAvg underperforms when data is non-identically independently distributed (non-IID) across sites. This is a particular concern for EHRs, where a range of factors can lead to distribution shifts including patient composition, institutional treatment guidelines, and institutional data capture processes (Zhao et al. (2018)). Patient composition, *i.e.,* differences in demographics and clinical presentation is one of the most significant sources of distribution shift (Prayitno et al. (2021)). Personalized FL, which aims to account for distribution shifts across datasets, is a potential solution for training models on non-IID data (Fallah et al. (2020)). Clustered Federated Learning is a variant of personalized FL that has demonstrated success in handling non-IID data when datasets naturally partition into clusters (*e.g.*, clinical groups) (Ghosh et al. (2020)). In this scenario, training separate models for each cluster has been shown to improve performance on downstream tasks. The challenge lies in identifying the clusters and partitioning the datasets accordingly. Recent patient clustering preprocessing using individual patient embeddings demonstrate improvements in downstream task performance in the centralized setting (Xu et al. (2020); Zeng et al. (2021)). This suggests clustering can be a promising avenue for healthcare tasks in a federated setting as well.

Addressing the absence of a privacy-preserving federated approach for clustering using individual patient embeddings is a critical step for personalized FL. Frameworks such as Differential Privacy (DP) have been introduced to address this issue. DP works by adding noise to summary-level data (*e.g.,* model parameters) prior to sharing information with a central server. But in the clinical context, the amount of noise needed to achieve privacy can compromise model performance (Ficek et al. (2021); Dwork and Roth (2014)). One can use cryptographic techniques that provide mathematical privacy guarantees without adding noise and can work with both summary and individual level data. One such technique that

is appropriate in a multi-site setting is Secure MultiParty Computation (SMPC). SMPC enables multiple parties to jointly compute a function over their inputs while keeping those inputs secret from each other by using a secret-sharing scheme (Evans et al. (2018)).

In this study, we introduce Privacy-preserving Community-Based Federated machine Learning (PCBFL), a privacy-preserving framework that incorporates a clustering preprocessing step into FL (Clustered FL). Using SMPC, PCBFL securely calculates patient-level embedding similarities across all sites while preserving privacy. We assume an honest-but-curious adversary scenario, in which the computing parties cannot learn the input from the secrets and will not intentionally collude with each other to learn the input (Evans et al. (2018)). By using individual patient embedding similarity scores to cluster patients into groups, we aim to improve downstream task performance. We evaluate PCBFL algorithm against two main federated comparators on a downstream mortality prediction task: Community-Based Federated machine Learning (CBFL) and FedAvg. CBFL, a state-of-the-art method, also employs a clustering preprocessing step, however, unlike PBCFL, CBFL uses aggregate hospital embeddings for patient clustering. FedAvg is the standard algorithm with no preprocessing for non-IID data. Additionally, we assess the performance of non-federated algorithms that conduct only model training: single site training and centralized training. Single site performance serves as a baseline that all federated algorithms should surpass, while centralized performance represents the gold standard against which federated algorithms are compared. We show that our PCBFL approach results in improved performance compared to both standard FedAvg and CBFL on the mortality prediction task. PCBFL also outperforms FedAvg and CBFL in the majority of the individual sites. In addition, our results demonstrate that PCBFL produces clinically meaningful clusters, grouping patients in low, medium, and high-risk cohorts. This suggests that PCBFL has the potential to support other clustering tasks such as federated phenotyping. Future work could explore the utility of our approach in a wider range of clinical applications.

### 1.1. Generalizable Insights about Machine Learning in the Context of Healthcare

In healthcare, protecting patient privacy while leveraging data to improve clinical outcomes is a crucial challenge. This challenge is particularly relevant for complex deep learning models that require large sample sizes. Our paper introduces PCBFL, a privacy-preserving framework that uses SMPC to incorporate a clustering preprocessing step into federated learning. Our approach provides a practical solution to protect patient privacy while enabling patient-level calculations across different hospitals. Another key issue in healthcare is dealing with non-IID data, where data from different sites have different distributions. Our clustering-based approach effectively handles non-IID data by partitioning patients into clinically relevant groups. This clustering procedure can be used in various healthcare tasks, including unsupervised patient clustering for phenotyping. Our methodological contributions could be extended to conduct large-scale phenotyping across many sites, enabling more accurate and granular sub-phentoypes to be discovered.

## 2. Related work

In FL, several works have studied the statistical heterogeneity of users' data and linked high heterogeneity to performance degradation and poor convergence (Li et al. (2018)). To address this, researchers have attempted to personalize learning to each user (Tan et al. (2022)). The proposed solutions typically occur at the preprocessing, learning, or postprocessing stages. Preprocessing solutions include data-augmentation and client partitioning (Sattler et al. (2020); Zhao et al. (2018)). Learning solutions include meta-learning and modifications to the FedAvg algorithm (*e.g.,* addition of regularization parameters) (Fallah et al. (2020); Deng et al. (2020); Li et al. (2020)). Post-processing techniques involve adaptation of the global model by the local site after federated training is complete (Hanzely and Richtárik (2020)). In healthcare, personalized FL has mostly focused on preprocessing steps. An example is CBFL, which uses embeddings to cluster patients (Huang et al. (2019)). The authors showed clustering improved performance on a downstream mortality prediction task compared to the standard FedAvg technique. However, due to privacy constraints, CBFL's patient clustering is based on average embeddings per site, *i.e.,* it does not use individual patient embeddings. This led to patient clusters based on the geography of hospitals and not based on patient characteristics.

## 3. Methods

### 3.1. Cohort and feature extraction

We used the eICU collaborative research database, which contains critical care data for 200,859 patients at 208 hospitals across the United States (Pollard et al. (2018)). We followed a similar data-processing step as Huang et al. (2019). The outcome of interest was mortality in the ICU, defined as the unit discharge status (0 for alive and 1 for expired). The independent variables are diagnosis, drugs, and physical exam markers in the first 48 hours of admission. We limited our features to the first 48 hours to ensure consistency on patient follow-up times and clinical relevancy of model predictions.

For diagnosis and drugs, we used the count of times the feature appears in the dataset for that patient. For physiological markers, we used the first recorded instance. Physical exam markers used include: Glasgow Coma Scale (GCS) Motor, GCS Verbal, GCS Eye, Heart Rate (HR), Systolic Blood pressure (SBP), Respiratory rate (RR) and Oxygen Saturation (O2%), age, admission weight, admission height. Drug and physiologic features were kept as is (1,056 and 7 features in total, respectively). Diagnosis codes were rolled up to 4 digits, *i.e.,* all 5 digit codes were converted to 4 digit codes resulting in 483 diagnosis codes. Note, compared to Huang et al. (2019), we also added diagnosis and physical exam markers to the features as these have been shown to improve predictive performance in related tasks (Sheikhalishahi et al. (2020)). All data was 0-1 normalized prior to training by the models.

We extracted patients from the dataset who had data for all three variable groups and a recorded outcome. This was done to avoid the need for imputation which could introduce bias to the data. Doing so would affect evaluation of the training architectures. We then filtered for sites that had a minimum of 250 patients, resulting in 20 sites and 20,221 patients. We randomly subsample 250 patients from each site, creating a final cohort of

5,000 patients. This subsampling approach was intended to create a more realistic FL scenario, where size of the dataset in each site is limited.

### 3.2. Privacy-preserving CBFL

A schematic of the steps involved in PCBFL is presented in Figure 1 and the procedures for each step are detailed in Supplementary Algorithms 1-4 (Appendix E). PCBFL is composed of four procedures: creating patient embeddings, estimating patient similarity securely, clustering patients, and predicting mortality.

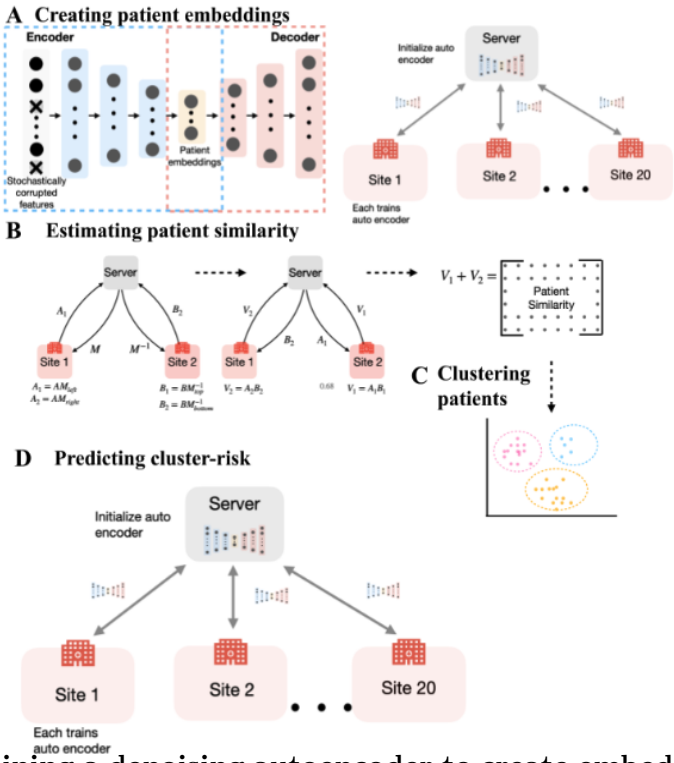



Figure 1: **(A) Training a denoising autoencoder to create embeddings.** A federated autoencoder is trained to obtain latent variables for each feature domain. Latent variables are concatenated to form a patient embedding vector. **(B) SMPC protocol to calculate the cosine similarity between vectors.** SMPC uses a secret sharing scheme to jointly calculate the dot product between pairs of vectors. **(C) Spectral clustering to cluster the patients** using similarity matrix generated from pairwise cosine similarities of embeddings. **(D) Cluster-based FL training.** Each model is separately trained per cluster.

#### 3.2.1. Creating patient embedding

Following Huang et al. (2019), we trained a federated denoising autoencoder made up of 6 layers including a three-layer encoder and an identical three-layer decoder to create patient embeddings. To reduce overfitting, 30% of the features are stochastically corrupted during training, *i.e.,* 30% features are forced to 0. A separate autoencoder was trained for each feature domain, *i.e.,* drugs, diagnosis, and physical examination. We used a ReLU activation

in the hidden layers, a sigmoid activation in the final output layer, and a Mean Squared Error loss. We used an Adam optimizer with a learning rate of $1e^{-3}$ and batch size of 32. Federated models were trained for 20 rounds with 10 epochs per round and centralized models were trained for 200 epochs. For one patient's embedding, we concatenated the latent variables of each feature domain

#### 3.2.2. Estimating patient similarity securely

We used cosine similarity as the similarity metric as it is invariant to scaling effects and works well with high dimensional vectors compared to euclidean distance (Strehl et al. (2000); Li et al. (2022)). We used SMPC to securely calculate patient embedding similarity across sites while preserving privacy. SMPC is a cryptographic technique that allows parties to jointly compute a function over their inputs while keeping the inputs secret, *i.e.,* only the output is made available (Evans et al. (2018)). The benefit of SMPC is that it protects privacy against both outside adversaries and other involved parties with mathematical guarantees and allows for exact calculation of cosine similarity across sites. We adapted a protocol from Du et al. (2004) to calculate the dot product across sites (see SMPC protocol) using secret sharing for an honest-but-curious adversary model. This protocol involves the following steps:

1. Create a $dxd$ invertible matrix $M$ and send $M$ to $site_1$ and $M^{-1}$ to $site_2$ (where $d$ is the embedding dimension)
2. Each site divides their dataset into submatrices and masks them with $M$ or $M^{-1}$
3. A limited number of masked submatrices are shared between sites
4. The submatrices are combined to produce the final dot product without revealing any information about the dataset

**SMPC PROTOCOL**

$Site_1$ hold dataset A $(N_1 \times d)$, $Site_2$ hold dataset B $(N_2 \times d)$ where $d$ =embedding dim and $N_i$ = number of patients.

1. Server creates a random invertible matrix $M_{d \times d}$ using Reed-Hoffman encoding and sends $M$ to Site 1 and $M^{-1}$ to Site 2.
2. Site 1 computes $A_1 = A \times M_{\text{left}}$, $A_2 = A \times M_{\text{right}}$ and sends $A_1$ to the server.
3. Site 2 computes $B_1 = B \times M^{-1}_{\text{top}}$, $B_2 = B \times M^{-1}_{\text{bottom}}$ and sends $B_2$ to the server.
4. Server sends $B_2$ to Site 1 and $A_1$ to Site 2.
5. Site 1 computes $V_a = A_2 \times B_2$ and sends it to the server.
6. Site 2 computes $V_b = A_1 \times B_1$ and sends it to the server.

We have $\mathbf{AB} = AMM^{-1}B = \begin{pmatrix} A_1 & A_2 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} = V_a + V_b$.

Secure calculation is possible as no party has sufficient information to reconstruct the original dataset with only some of the submatrices. More concretely, it can be seen that as long as sites only share half of their encoded matrices ($A_1$ and $B_2$) there remains an infinite number of solutions to the problem. The method relies on the construction of a secure matrix $M$. This matrix can be generated using maximum distance separable (MDS) codes such as Reed-Solomon codesDu et al. (2004) . MDS codes ensure that any subset of columns are linearly independent of each other making it impossible to recover the original data. For a more detailed introduction on MDS codes please see MacWilliams and Sloane (1977). Note that all embeddings were first L2-normalized prior to calculating the dot product so this product is equivalent to cosine similarity. In our case, we conduct pairwise calculation of cosine similarity between sites and concatenate these together to construct the final similarity matrix across all patients.

#### 3.2.3. Clustering patients

We employed spectral clustering to cluster patients using the cosine similarity matrix. Spectral clustering is suited to this task as it utilizes the similarity matrix's global structure to capture complex relationships (von Luxburg (2007)). We used the elbow method to determine the optimal number of clusters. For this, we first calculated the Within-Cluster-Sum-of-Squares (WCSS) for clusters 1-10 (see Supplement: Appendix A.1 WCSS). WCSS is a metric to measure the compactness of the clusters. We selected the 'elbow' point of the plot after which additional clusters do not lead to substantial improvements in WCSS (*i.e.*, compactness of the clusters). This is a heuristic that determines the minimum number of clusters necessary to account for the majority of the variance in the dataset (Madhulatha (2012)). A smaller WCSS implies that the data points are more compact, indicating tighter clustering of similar points. We assessed a range of clusters between 1 to 10 and ultimately choose 3 (Supplementary Figure 1).

#### 3.2.4. Predicting mortality

We trained a FeedForward neural network with 3 input heads and a classification module. Each input head processes a feature domain into a 5-dimensional representation. These representations are concatenated and fed through the classification module to generate predictions. The multihead structure integrates distinct data domains more effectively by first processing each type separately before combining them for prediction. We followed a similar training algorithm as FedAvg, but we trained a model on each cluster separately. That is, the server initializes a separate model for each cluster and each site trains a cluster model only on the patients in that respective cluster. Weights were aggregated based on each site's sample size for that cluster (Figure 1D).

### 3.3. Other models

We evaluated the performance gain from PCBFL against other training methods including: single site, centralized, FedAvg and CBFL. Single site training refers to the average performance of each site if it were to train a model separately. Centralized training refers to the

performance of a model trained on all data together as if it were one site. Centralized training performance is the gold-standard benchmark hoped to be achieved by FL. FedAvg refers to standard Federated Averaging procedure, which aggregates model parameters based on sample size in each site. CBFL refers to the community based clustering procedure described by Huang et al. (2019) that uses K-means clustering on average hospital embeddings. The generated clusters were sent to each site to assign a cluster to each patient. Models are then trained separately for each cluster. Note that the major difference between CBFL and PCBFL is in the clustering approach; CBFL uses average site embeddings while PCBFL enables privacy-preserving clustering of patients based on individual patient embeddings.

### 3.4. Model training

We implemented the feedforward models with ReLU activation in the hidden layers and a sigmoid activation in the final output layer. We employed Binary Cross Entropy loss. All feedforward models were trained using the same hyperparameters: an Adam optimizer with a learning rate of $1e^{-3}$ for all models and batch size of 32. For all federated training methods, we used 20 training rounds with 10 epochs per round. For all central models, we used 200 epochs. This kept the training epochs consistent across all models.

### 3.5. Evaluation

#### 3.5.1. Cohort analysis for clustered FL algorithms

We examined the clusters generated by PCBFL and CBFL by comparing patients' mortality and feature distributions between clusters. We used one-way ANOVA testing for continuous variables and Negative Binomial testing for count variables to determine statistical significance. We chose Negative Binomial over Poisson to account for the overdispersion in the count data (Hilbe (2011)). A p value of $<0.05$ with Bonferroni correction was used to determine statistical significant differences between clusters. We also examined whether the clusters generated by PCBFL and CBFL capture the regional distribution of hospitals. We grouped hospitals by region as defined by eICU Collaborative Research Dataset (Midwest, Northeast, South, West) and conducted a chi-squared test examining the relationship between region and cluster distribution.

#### 3.5.2. Prediction task

Data was randomly split into training and testing datasets in a 70:30 ratio. Since only 20% of the labels were positive, we evaluated performance with both AUC and AUPRC scores. We ran the models for 100 times and calculated the mean scores and bootstrapped estimates (1000 iterations) of the 95% confidence intervals. We calculated the overall performance of a protocol as the weighted average of individual site scores. This weighting accounts for the number of samples used in model development at each site. For the cases of Single Site and FedAvg, where each site trains one model and has the same number of patients, this is a simple average. In the cases of CBFL and PCBFL, where sites train three separate models, the weighting is based on the proportion of patients that belong to the site and cluster (see Supplement: Appendix B. Calculating overall performance). We also compared the performance of the models at each site to determine if there are sites that fail to benefit from

FL, or PCBFL in particular. This is important as we wanted to ensure that all sites benefit from FL to incentivize FL collaboration Li et al. (2020). All code was written in Python 3.9.7 and Pytorch 1.12.1 and is available on GitHub https://github.com/G2Lab/pcfbl.

## 4. Results

### 4.1. PCBFL provides privacy-preserving and accurate patient similarity scores using Secure MultiParty Computation

We first evaluated whether the use of SMPC affects the accuracy of the patient similarity scores calculated using cosine similarity. This was done by comparing PCBFL results to the results of a plaintext and centralized calculation (referred to as True). The root mean squared error between the True and privacy-preserving scores was $<5x10^{-11}$, indicating that the protocol is highly accurate. Figure 2 displays the comparison of the True cosine similarity scores and privacy-preserving cosine similarity scores, where each point represents a pairwise comparison of two patients. The graph demonstrates that the SMPC protocol accurately calculates the cosine similarity between patients' embeddings at all ranges.

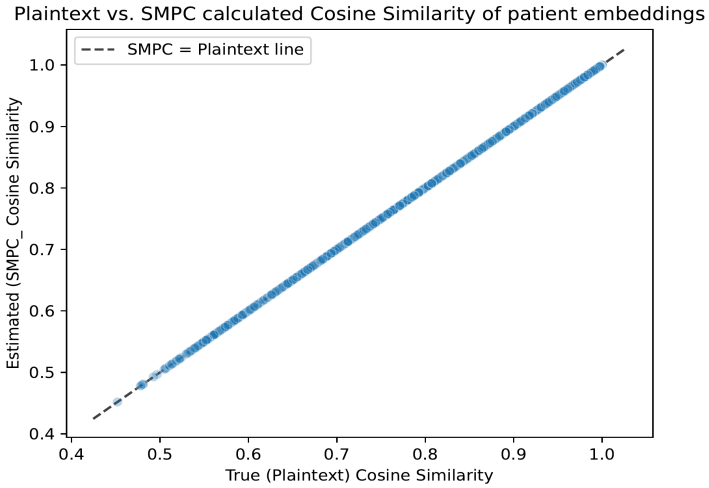



Figure 2: **Comparison of cosine similarity scores** calculated using PCBFL's SMPC protocol and a plaintext centralized truth (True). Each point is a cosine similarity score between 2 patient embeddings.

### 4.2. PCBFL provides clinically meaningful clusters

We examined the clusters determined by PCBFL in terms of their mortality and physical examination scores (Table 1). The resulting three clusters were found to correspond to three distinct levels of severity based on both true mortality rates and physical examination scores.The high risk group was more likely to have higher mortality, lower GCS, higher age and worse vital sign measurements ($p < 0.005$). In contrast, an examination of CBFL did not yield distinct clinical severity groups, with only significant differences in age and RR. Full mortality and physical examination feature distributions of PCBFL and CBFL clusters are shown in Supplementary Tables 2 and 3, respectively (Appendix F).

Table 1: Outcome and physical examination distribution for 3 clusters identified by PCBFL

| **PCBFL clusters:** | **Low** | **Medium** | **High** | **p-value** |
|---|---|---|---|---|
| **Mortality** | 11.7% | 20.8% | 26.3% | <0.005* |
| **GCS** | 13.1 | 12.7 | 12.5 | <0.005* |
| **Age** | 55.4 | 67.8 | 69.2 | <0.005* |
| **HR** | 86.7 | 89.1 | 92.6 | <0.005* |
| **SBP** | 127.0 | 118.0 | 116.1 | <0.005* |
| **RR** | 20.2 | 20.1 | 21.4 | <0.005* |
| **$O_2$%** | 97.2 | 96.3 | 96.1 | <0.005* |

* statistically significant

We also compared the distribution of diagnosis counts across clusters and found that conditions indicative of severe disease are more likely to occur in the high-risk group (Table 2, see Methods 3.5.1 for statistical tests used). Overall, we identified 28 out of 483 diagnoses that were more likely to occur in the high risk group ($p<0.0001$). These included clinically relevant diagnoses for cardiovascular disease, respiratory disease, renal disease, infectious disease, metabolic disorders, hematological disorders, and nutritional disorders. A similar analysis on CBFL clusters yielded no statistically significant differences in diagnosis counts. Full diagnosis feature distributions of PCBFL and CBFL clusters are shown in Supplementary Tables 4 and 5, respectively (Appendix F).

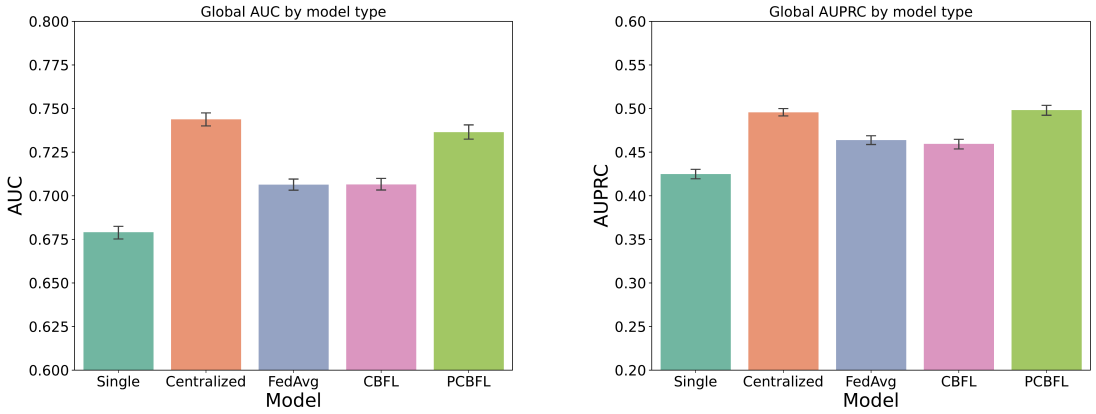
We performed the same comparison for the distribution of medication counts between clusters and found 154 drugs out of 1,056 that differed between clusters. Specifically, 45 and 109 in drugs were more likely to be prescribed in the high and medium risk groups, respectively ($p<5e^{-5}$). Same analysis for CBFL clusters resulted in 48 out of 1,056 drugs that differed between clusters. Full medication count distributions of PCBFL and CBFL clusters are shown in Supplementary Tables 6 and 7, respectively (Appendix F). Finally, comparing the distribution of clusters by region showed that PCBFL clusters are not associated with region ($p=0.10$) but CBFL clusters are ($p¡1e^{-3}$). See Supplementary Table 1 (Appendix C) for the full cluster distribution breakdown by region.

### 4.3. PCBL increases predictive performance of federated learning

Next, we evaluated performance on the mortality prediction task. Table 3 shows the global AUC and AUPRC scores of model training for Single site, Centralized, FedAvg, CBFL and PCBFL. PCBFL achieves statistically significant improvements against Single site, FedAvg and CBFL. Compared to CBFL and FedAvg, PCBFL improves mean AUC by 4.4% (3.0-5.5% at 95% CI) and 4.2% (2.8-5.8% at 95% CI) and AUPRC by 7.3% (3.4-11.6% at 95% CI) and 8.4% (3.4-13.8% at 95% CI) , respectively. Figures 3a and 3b show global AUC and AUPRC scores for each model. Note that we calculated average per site performance for FedAvg and Single site training, while performance was measured as a weighted average of per cluster and site performance for CBFL and PCBFL (see Supplement Appendix B for definitions).

Table 2: Conditions more likely to occur in PCBFL high-risk group

| Category | Condition more likely to occur ($p < 0.0001$) |
|---|---|
| **Cardiovascular** | Atrial Fibrillation and Flutter, Congestive Heart Failure, Hypertension, Tachycardia |
| **Respiratory** | Asphyxia and Hypoxemia, Obstructive Chronic Bronchitis, Paralysis of Vocal Cords or Larynx |
| **Renal** | Acute Kidney Failure, Chronic Kidney Disease, Cystitis |
| **Infectious** | Septicemia, Fever |
| **Metabolic** | Abnormal Blood Chemistry, Acidosis, Disorders of Magnesium Metabolism, Hyperlipidemia, Hyperpotassemia, Hypothyroidism, Obesity |
| **Hematologic** | Anemia, Coagulation Defects, Disease of White Blood Cells, Thrombocytopenia |
| **Nutritional** | Protein-calorie Malnutrition, Dehydration |

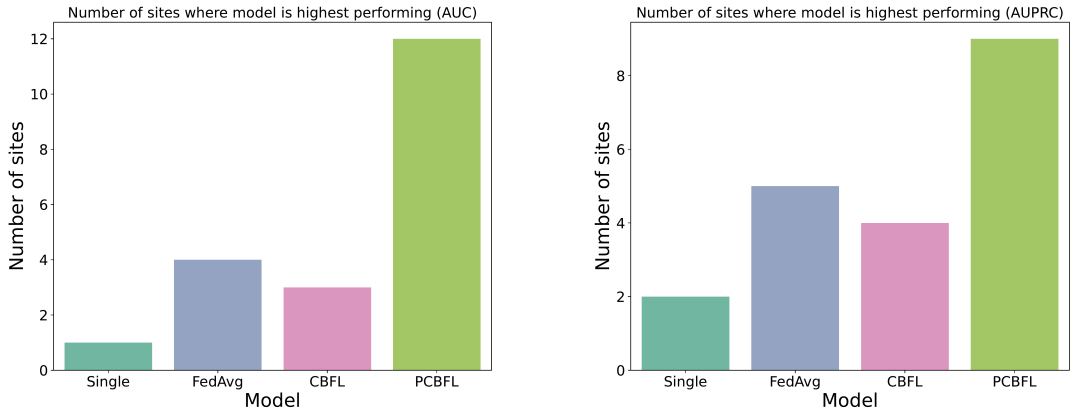



(a) (b)

Figure 3: **Performance by model type**, AUC (a) and AUPRC (b) for Single, Centralized, FedAvg, CBFL and PCBFL.

### 4.4. PCBFL enables better predictive performance at most sites

Figure 4 shows the number of sites where each model has the highest performance. We compared Single site, FedAvg, CBFL and PCBFL at 20 sites. Centralized training was not compared as there are no per-site results. PCBFL performs best at 12 and 9 sites in terms of AUC and AUPRC, respectively. Figure 5 shows the AUC scores for each site and cluster (see Supplementary Figure 2 for AUPRC). PCBFL outperforms single site training at 16 and 18 sites, FedAvg at 14 and 14 sites, and CBFL at 13 and 13 sites for AUC and AUPRC, respectively.




$(a)$ $(b)$

Figure 4: **Number of sites where model has highest** AUC (a) and AUPRC (b) for Single, Centralized, FedAvg, CBFL and PCBFL.

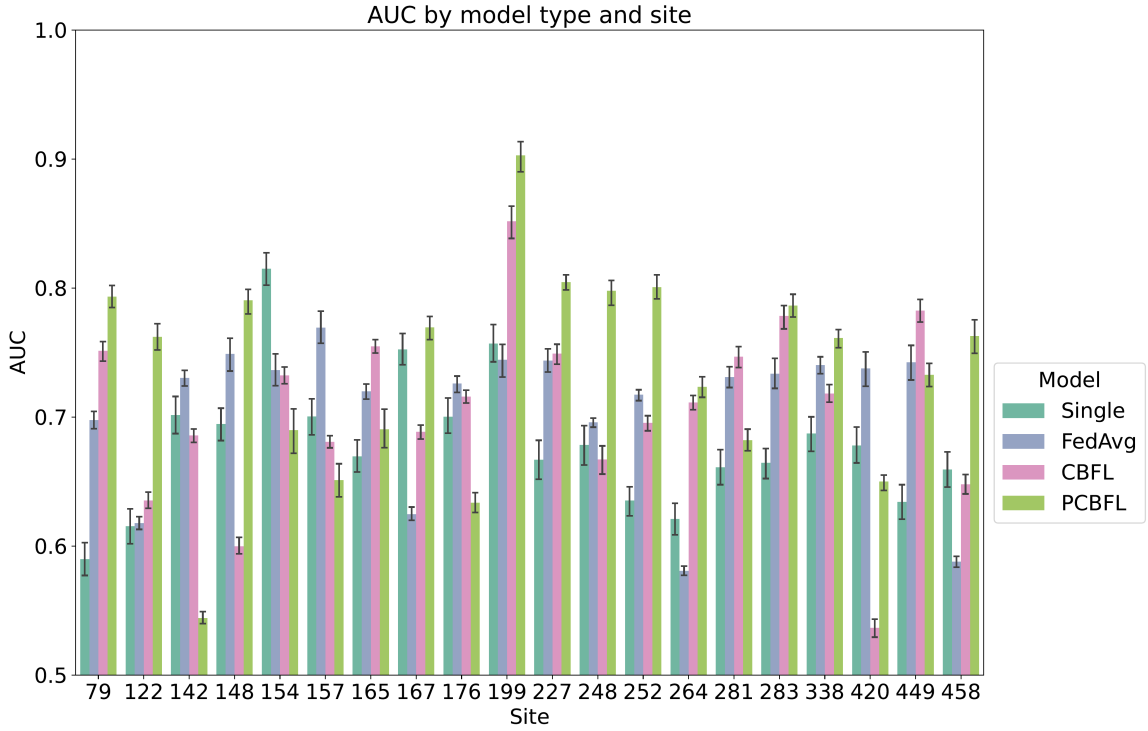



Figure 5: **Model performance by site**, AUC. Results for Single, FedAvg, CBFL and PCBFL.

## 5. Discussion

We present a new personalized FL framework based on privacy-preserving patient clustering (PCBFL). We show that this algorithm enables better performance in a downstream mortality prediction task across 20 ICU datasets when compared to traditional FL and existing

Clustered FL techniques. Furthermore, per site analysis shows that PCBFL is most likely to achieve best performance at any given site. PCBFL also generates clinically meaningful clusters categorizing patients into low, medium, and high mortality risk groups based on physical exam, diagnosis, and medication values. Our results show that PCBFL can be used to implement personalized FL, addressing the challenges of non-IID EHR data and patient privacy by securely clustering patients and optimizing model performance on each cluster. The ability to generate clinically meaningful subgroups suggests PCBFL can be extended to other clinical use cases such as phenotyping, risk stratification, and advancing disease understanding.

We demonstrated that PCBFL is able to generate clinically meaningful groups that were categorized mostly based on patient severity. This is in contrast to CBFL, which has been shown to cluster patients based on geographical distribution of the hospitals (Huang et al. (2019)). PCBFL also enables clustering over a very large number of sites and patients and is able to do so without introducing error into the calculation. These findings suggest that PCBFL can be extended to support the discovery of novel subgroups without the need for a prediction task (*e.g.,* unsupervised phenotyping, risk stratification, disease subtyping, treatment selection, and trial recruitment) (Robinson (2012)). Previous studies have shown success of clustering patients in centralized settings and we believe it can now be extended to the federated settings (Xu et al. (2020); Zeng et al. (2021)). Overall, our study highlights the potential of PCBFL as a powerful tool for collaborative analysis of healthcare data.

We also found that PCBFL demonstrates a meaningful improvement in global performance compared to traditional FL frameworks (FedAvg) and other personalized FL frameworks (CBFL). We believe this improvement can be attributed to our clustering technique, which is able to divide the federated datasets into more IID cohorts by using patient similarity scores between individual patients. This likely improves the performance of the downstream tasks by reducing the impact of site-specific biases and allowing the model to focus on cohort-specific features necessary for predictions. As a result, it has the potential to improve the generalizability of the models (Prayitno et al. (2021); Fallah et al. (2020)). Moreover, we found that PCBFL has the best per-site performance compared to other methods, which increases motivation for sites to participate in federated learning (Cho et al. (2022)).

**Limitations:** PCBFL has some limitations. First, it relies on sufficient sample sizes per cluster, which can be an issue in cases where sites have limited datasets. However, given the improved performance against single site training, in cases where sufficient samples are available, clustering should be preferred. Second, the secure clustering algorithm requires pairwise cosine similarity calculation across all sites. This results in additional communication costs as each pair of hospitals must use their own separate masking matrix and secret sharing protocols. A new secret sharing scheme with central server coordination can be developed that reuses secret shares across multiple calculations, thus reducing the communication cost. In addition, PCBFL requires training of two deep learning models and a clustering algorithm. As such, the communication cost is higher than traditional FL frameworks. However, we feel the trade-off between communication cost and improved model performance is acceptable in the context of healthcare, where higher accuracy is preferred over compute power. Finally, this analysis is limited to eICU dataset and a mortality pre-

diction task. PCBFL should be assessed on a range of clinical prediction tasks and datasets to fully evaluate its performance.

## 6. Conclusion

We present a new personalized FL framework based on a novel privacy-preserving patient clustering algorithm (PCBFL) that addresses the challenge of non-IID data and patient privacy in federated settings. Our study demonstrates that PCBFL enables better model performance than existing methods in a mortality prediction task. We showed that the clustering technique used by PCBFL divides the federated datasets into clinically meaningful cohorts suggesting it can be extended to other phenotyping tasks. These findings highlight the potential of PCBFL as a powerful tool for collaborative analysis of healthcare data. In future work, we plan to explore the generalizability of PCBFL to other healthcare datasets and domains.

## References

Y J Cho, D Jhunjhunwala, T Li, V Smith, and others. To federate or not to federate: Incentivizing client participation in federated learning. *arXiv preprint arXiv*, 2022.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor AI: Predicting clinical events via recurrent neural networks. *JMLR Workshop Conf. Proc.*, 56:301–318, August 2016.

Ittai Dayan, Holger R Roth, and Aoxiao et al. Zhong. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nat. Med.*, 27(10):1735–1743, September 2021.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Wenliang Du, Yunghsiang S Han, and Shigang Chen. Privacy-Preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, Proceedings, pages 222–233. Society for Industrial and Applied Mathematics, April 2004.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nat. Med.*, 25(1):24–29, January 2019.

David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure Multi-Party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3): 70–246, 2018.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

Joseph Ficek, Wei Wang, Henian Chen, Getachew Dagne, and Ellen Daley. Differential privacy in health research: A scoping review. *J. Am. Med. Inform. Assoc.*, 28(10):2269–2276, September 2021.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. pages 19586–19597, June 2020.

Varun Gulshan, Lily Peng, and Marc et al. Coram. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, December 2016.

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

Joseph M Hilbe. *Negative Binomial Regression*. Cambridge University Press, March 2011.

Mahan Hosseini, Michael Powell, John Collins, Chloe Callahan-Flintoft, William Jones, Howard Bowman, and Brad Wyble. I tried a bunch of things: The dangers of unexpected overfitting in classification of brain data. *Neuroscience & Biobehavioral Reviews*, 119: 456–467, 2020.

Li Huang, Andrew L Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *J. Biomed. Inform.*, 99:103291, November 2019.

Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on Non-IID data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. ieeexplore.ieee.org, May 2022.

Tian Li, Anit Kumar Sahu, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

T Soni Madhulatha. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*, 2012.

B McMahan, E Moore, D Ramage, and others. Communication-efficient learning of deep networks from decentralized data. *Artif. Intell.*, 2017.

Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Sci. Rep.*, 6:26094, May 2016.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Brief. Bioinform.*, 19(6): 1236–1246, November 2018.

Sarthak Pati, Ujjwal Baid, and Brandon et al. Edwards. Federated learning enables big data for rare cancer boundary detection. *Nat. Commun.*, 13(1):7346, December 2022.

Tom J Pollard, Alistair E W Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eICU collaborative research database, a freely available multi-center database for critical care research. *Sci Data*, 5:180178, September 2018.

Prayitno, Chi-Ren Shyu, Karisma Trinanda Putra, Hsing-Chung Chen, Yuan-Yu Tsai, K S M Tozammel Hossain, Wei Jiang, and Zon-Yin Shae. A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications. *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, 11(23):11191, November 2021.

Nicola Rieke, Jonny Hancox, and Wenqi et al. Li. The future of digital health with federated learning. *npj Digital Medicine*, 3(1):1–7, September 2020.

Peter N Robinson. Deep phenotyping for precision medicine. *Hum. Mutat.*, 33(5):777–780, May 2012.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. Benchmarking machine learning models on multi-centre eICU critical care dataset. *PLoS One*, 15(7): e0235424, July 2020.

Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, volume 58, page 64, 2000.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Ulrike von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416, December 2007.

Zhenxing Xu, Fei Wang, and Prakash et al. Adekkanattu. Subphenotyping depression using machine learning and electronic health records. *Learn Health Syst*, 4(4):e10241, October 2020.

Xianlong Zeng, Simon Lin, and Chang Liu. Transformer-based unsupervised patient representation learning based on medical claims for risk stratification and analysis. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, number Article 17 in BCB '21, pages 1–9, New York, NY, USA, August 2021. Association for Computing Machinery.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

# Appendix

## Appendix A. Selecting number of clusters

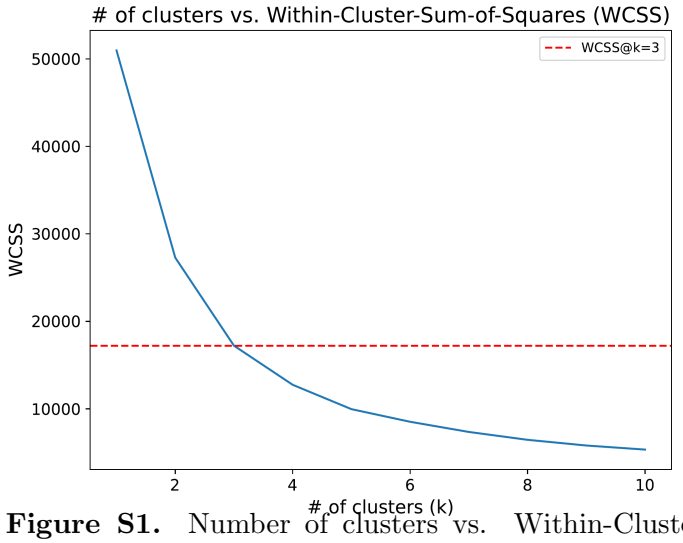
### A.1. Within-Cluster-Sum-Of-Squares

Within-Cluster-Sum-Of-Squares (WCSS) is a metric used in clustering to determine the optimal number of clusters. It calculates the total sum of the squared distances between each data point within a cluster and the center of that cluster (centroid). A smaller WCSS implies that the data points are more compact, indicating tighter clustering of similar points. It is defined as:

$$\text{WCSS} := \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_{ki} - \mu_k)^2 \tag{1}$$

where cluster $k$ ranges from $1..K$m data points, $x_{ki}$ in cluster $k$ range from $1..n_k$, and $\mu_k$ is the mean of the points in cluster $k$ *i.e.*, $\frac{1}{n_k} \sum_{i=1}^{n_k} x_{ki}$

### A.2. Elbow plot




**Supplementary Figure S1.** Number of clusters vs. Within-Cluster-Sum-Of-Squares (WCSS). Dashed red-line indicates WCSS at k=3.

## Appendix B. Calculating overall performance

### B.1. Single site and FedAvg

To calculate the overall AUC and AUPRC for Single Site and FedAvg (Global R), we use a simple average (or uniform weighted average). This is because each site trains only one model and has the same number of patients:

$$\text{Global R} = \frac{1}{N}\sum_{c=1}^{C} R_c n_c \tag{2}$$

where $R_c$ is the result (AUC or AUPRC) for site $c$, $n_c$ is the sample size for site $c$, and $N$ is the total number of samples across all sites.

### B.2. PCFBL and CBFL

To calculate the overall AUC and AUPRC (Global R) for CBFL and PCBFL, we use a weighted average of results that takes into account the number of patients each site contributes to a cluster. This is necessary as each site trains 3 separate models (1 per cluster) and their sample size contribution to a given cluster varies:

$$\text{Global R} = \frac{1}{N}\sum_{c=1}^{C}\sum_{k=1}^{K} R_{ck} n_{ck} \tag{3}$$

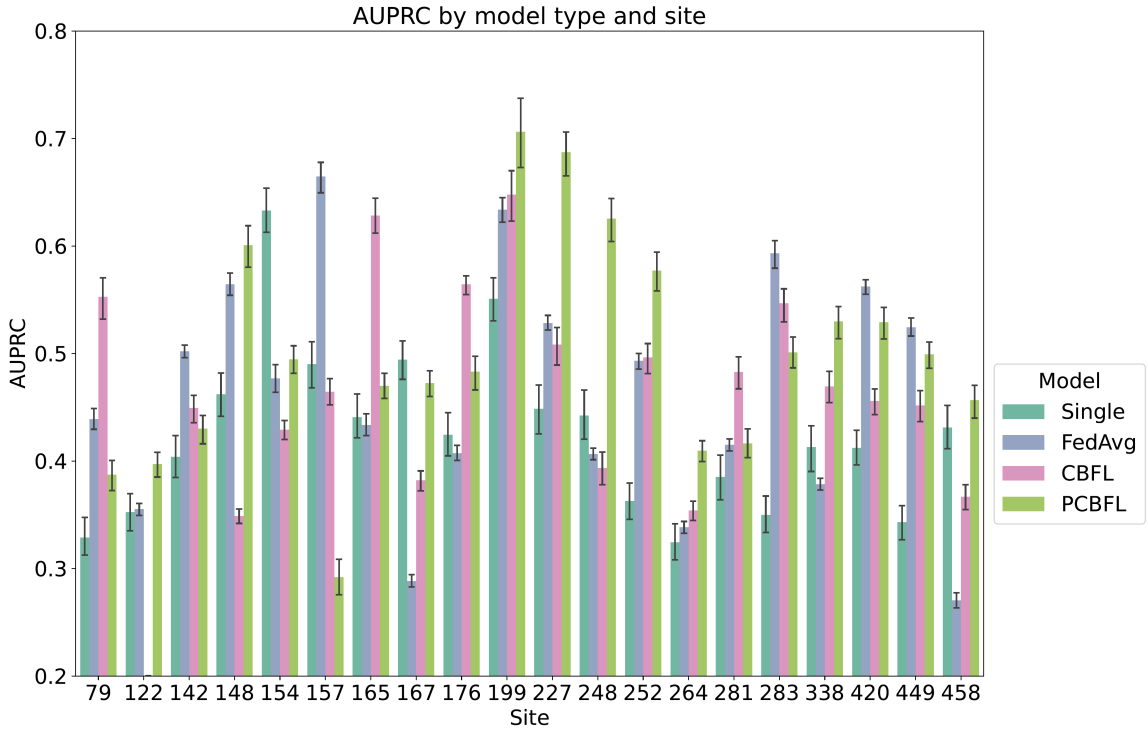
where $R_{ck}$ is the result (AUC or AUPRC) for site $c$ and cluster $k$, $n_{ck}$ is the sample size for site $c$ and cluster $k$, and $N$ is the total number of samples across all sites.

## Appendix C. Cluster regional distribution

| Region | Cluster | PCBFL | CBFL |
|---|---|---|---|
| **Midwest** | Low | 36.7 | 46.4 |
| | Medium | 19.85 | 48.25 |
| | High | 43.45 | 5.35 |
| **Northeast** | Low | 44.2 | 36 |
| | Medium | 15.6 | 61.2 |
| | High | 40.2 | 2.8 |
| **South** | Low | 39.53 | 18.27 |
| | Medium | 16.47 | 81.47 |
| | High | 44 | 0.27 |
| **West** | Low | 51.87 | 24.6 |
| | Medium | 22.6 | 10.73 |
| | High | 25.53 | 64.67 |

**Supp. Table 1.** Cluster distribution of each region for PCBFL and CBFL. All values expressed as a % of patients in the region

## Appendix D. Per site results: AUPRC




**Supplementary Figure S2.** Model performance by site, AUPRC. Results for Single, FedAvg, CBFL and PCBFL.

## Appendix E. Pseudocode

**Algorithm 1** Creating patient embeddings

**Input:** A set of patients across multiple sites, each associated with domains: observation, diagnosis, drug.

**Output:** The patient embedding matrix $\mathbf{E}$.

**for** *each* $t$ *in domains observation, diagnosis, drug* **do**
  Initialize weights, $w_{ae}$, of autoencoder model $f_{ae}$ **for** *each round* $i$ *from* 1 *to* 20 **do**
    **for** *each site* $c$ *from* 1 *to* $C$ *in* ***parallel*** **do**
      Train $f_{ae}$ for 20 epochs to obtain $w^c_{i,ae}$ `//` $c$ `is the site index and` $i$ `is the training round index`
      **return** $w^c_{i,ae}$ to server
    **end**
    Server updates weights using $w_{i+1,ae} \leftarrow \sum_{c=1}^{C} \frac{n^c}{N} w^c_{i,ae}$ `//` $n^c$ `is number of patients at site` $c$, $N$ `is total number of patients`
  **end**
  set final model $f^g_{ae} \leftarrow f_{ae}$
**end**
**for** *each site* $c$ *from* 1 *to* $C$ *in* ***parallel*** **do**
  Create embedding matrix $\mathbf{E}^{|P_c| \times D}$ `//` $|P_c|$ `is the number of patients at site` $c$ `and` $D$ `is the embedding dimension`
  **for** *each patient* $p$ *in patient set,* $P_c$ **do**
    Create empty patient embedding vector, $\mathbf{E}_p^{1 \times D}$ **for** *each* $t$ *in domains observation, diagnosis, drug* **do**
      Create domain patient embedding, $\mathbf{E}_{p_t} = f^g_{ae}(p_t)$ Concatenate $\mathbf{E}_p$ with $\mathbf{E}_{p_t}$
    **end**
    $\mathbf{E}[p] \leftarrow \mathbf{E}_p$
  **end**
  Return $\mathbf{E}$
**end**

**Algorithm 2** Estimating patient similarity

**Input:** Patient embeddings across multiple sites, $\mathbf{E}_p$.

**Output:** Global similarity matrix $S$.

Initialize global similarity matrix, $S$, as a $P \times P$ zero matrix, where $P$ is the total number of patients **for** *every unique pair of sites c, d ∈ sites C in* ***parallel*** **do**
  Construct invertible matrix, $M_{f\times f}$ using Reed-Hoffman encoding with degree $f > D$ *i.e.,* degree greater than embedding dimension Send $M$ to $c$ and $M^{-1}$ to site $d$ **for** *site c* **do**
    Set matrix $A$ to hold all patient embeddings $\mathbf{E}_p$ within site $c$ Calculate $A_1 = A \times M_{left}$, $A_2 = A \times M_{right}$ Send $A_1$ to the server
  **end**
  **for** *site d* **do**
    Set matrix $B$ to hold all patient embeddings $\mathbf{E}_p$ within site $d$ Calculate $B_1 = B \times M^{-1}_{top}$, $B_2 = B \times M^{-1}_{bottom}$ Send $B_2$ to the server
  **end**
  **for** *site c* **do**
    Receive $B_2$ from the server Calculate $V_2 = A_2B_2$ Send $V_2$ to the server
  **end**
  **for** *site d* **do**
    Receive $A_1$ from the server Calculate $V_1 = A_1B_1$ Send $V_1$ to the server
  **end**
  Server completes $V$=$V_1$+$V_2$ **for** $\forall$ *patient* $i \in c$ **do**
    **for** $\forall$ *patient* $j \in d$ **do**
      Update similarity matrix $S$ with the corresponding value in $V$: $S_{c_i,d_j}$, $S_{d_j,c_i} \leftarrow V_{i,j}$
    **end**
  **end**
**end**

Return Global similarity matrix $S$

**Algorithm 3** Clustering patients

**Input:** Global similarity matrix $S$.

**Output:** Patient clusters for each site $c$ in site $C$.

Initialize list to hold within-cluster sum of squares (WCSS) for each $k$, denoted as WCSS[] **for** *each k ∈ 1, 2, ..., 10* **do**
  Apply spectral clustering to global similarity matrix $S$ with $k$ clusters Compute WCSS for current $k$, denoted as $\text{WCSS}_k$ Store $\text{WCSS}_k$ in WCSS[$k$]
**end**

Determine optimal number of clusters, $k_{opt}$, using elbow method on WCSS **for** *each c ∈ site C* **do**
  Apply spectral clustering on $S$ using $k_{opt}$ clusters Return patient clusters for site $c$
**end**

**Algorithm 4** Predicting Outcomes

**Input:** Clusters $K$
**Output:** Global AUC, Global AUPRC
**for** *each cluster* $k \in K$ **do**
  Initialize weights $w_{pk}$ of prediction model, $f_{pk}$ **for** *each round* $i$ *in 1 to 20* **do**
    **for** *each site* $c \in$ *sites* $C$ **do**
      Train $f_{pk}$ for 20 epochs to obtain $w^c_{i,pk}$ `// c is the site index and i is the training round index`
      **return** $w^c_{i,pk}$ to server
    **end**
    Server updates weights $w_{i+1,pk} \leftarrow \sum_{c=1}^{C} \sum_{k=1}^{K} \frac{n^{ck}}{N^k} w^c_{i,pk}$ `// ` $n^{ck}$ `is number of patients at site` $c$ `in cluster` $k$`,` $N^k$ `is total number of patients in cluster` $k$
    **for** *each site* $c \in C$ **do**
      Measure $\text{AUC}_{c,k}$ and $\text{AUPRC}_{c,k}$ in test set and send results
    **end**
  **end**
  Global AUC $\leftarrow \sum_{c=1}^{C} \sum_{k=1}^{K} \frac{n^{ck}}{N^k}$ $\text{AUC}_{c,k}$ Global AUPRC $\leftarrow \sum_{c=1}^{C} \sum_{k=1}^{K} \frac{n^{ck}}{N^k}$ $\text{AUPRC}_{c,k}$
**end**
**return** Global AUC, Global AUPRC

## Appendix F. Feature distributions by cluster

Supplementary tables 2-7 can be found here.