Journal of Biomedical Informatics 99 (2019) 103291

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Special Report

Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records

Li Huang[a,b], Andrew L. Shea[c], Huining Qian[d], Aditya Masurkar[e], Hao Deng[f,g,h], Dianbo Liu[c,h,i,*]

[a] Academy of Arts and Design, Tsinghua University, Beijing 10084, China
[b] The Future Laboratory, Tsinghua University, Beijing 10084, China
[c] Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, United States
[d] College of Applied Mathematical and Physical Science, Beijing University of Technology, Beijing 100124, China
[e] School of Engineering, Northeastern University, Boston, MA 02115, United States
[f] Department Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Boston 02115, United States
[g] School of Public Health, Johns Hopkins University, United States
[h] Boston Children's Hospital, Boston, MA 02115, United States
[i] Medical School, Harvard University, Boston, MA 02115, United States

ARTICLE INFO

Keywords:
Distributed clustering
Autoencoder
Federated machine learning
Non-IID
Critical care

ABSTRACT

Electronic medical records (EMRs) support the development of machine learning algorithms for predicting disease incidence, patient response to treatment, and other healthcare events. But so far most algorithms have been centralized, taking little account of the decentralized, non-identically independently distributed (non-IID), and privacy-sensitive characteristics of EMRs that can complicate data collection, sharing and learning. To address this challenge, we introduced a community-based federated machine learning (CBFL) algorithm and evaluated it on non-IID ICU EMRs. Our algorithm clustered the distributed data into clinically meaningful communities that captured similar diagnoses and geographical locations, and learnt one model for each community. Throughout the learning process, the data was kept local at hospitals, while locally-computed results were aggregated on a server. Evaluation results show that CBFL outperformed the baseline federated machine learning (FL) algorithm in terms of Area Under the Receiver Operating Characteristic Curve (ROC AUC), Area Under the Precision-Recall Curve (PR AUC), and communication cost between hospitals and the server. Furthermore, communities' performance difference could be explained by how dissimilar one community was to others.

1. Introduction

That EMRs improve the quality of healthcare has been endorsed by various evidences including enhanced performance of patients with chronic illness [1–5], reducing unnecessary medical examinations [6], cost saving for healthcare providers [7], better medical education [8] and more. To reap the benefits to a larger extent, machine learning applications have been developed on EMRs: for instance, ensemble learning of regression, k-nearest neighbor, decision trees and support vector machines for predicting type 2 diabetes (T2D) one year prior to diagnosis of diabetes [9], prediction of suicide risk via EMR-driven nonnegative restricted Boltzmann machines [10], classification of normal versus age-related macular degeneration OCT images using deep neural networks [11], and modeling of hospital readmission rates by a multistep Naïve Bayes-based learning strategy [12].

While such applications demonstrated promising perspectives towards translation of EMRs into improved human health [13], nevertheless they were developed under the premise that EMRs could be easily shared across silos and stored in centralized data warehouses. Generated by individual patients and in diverse hospitals/clinics, EMRs are distributed and sensitive in nature. This may impede adoption of machine learning on EMRs in reality, and has entailed researchers to raise concerns on central storage of EMRs and on security, cost-effectiveness, privacy and availability of medical data sharing [14–24]. These concerns can be addressed by FL that keeps both data and computation local in distributed silos and aggregates locally computational results to train a global predictive model [25,26]. Indeed, FL precludes the need of data collection and sharing, and thus can serve as

* Corresponding author at: Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, United States.
E-mail address: dianbo@mit.edu (D. Liu).

https://doi.org/10.1016/j.jbi.2019.103291
Received 3 April 2019; Received in revised form 16 September 2019; Accepted 18 September 2019
Available online 24 September 2019

Table 1
Examples of *drug features*.

Patient unit stay ID	2 ML - METOCLOPRAMIDE HCL 5 MG/ML IJ SOLN	3 ML VIAL: INSULIN REGULAR HUMAN 100 UNIT/ML IJ SOLN	METOPROLOL SUCCINATE ER 50 MG PO TB24	…
141,194	1	1	1	…
141,203	0	0	1	…
141,208	0	0	0	…
141,229	0	0	0	…
141,233	0	0	0	…
…	…	…	…	…

a desirable framework for developing machine learning applications on privacy-sensitive EMRs.

However, FL may underperform when data is non-identically independently distributed [25,27–29], as EMRs usually are [30]. To tackle this non-IID challenge and inspired by deep embedding clustering [31], we proposed a community-based federated learning (CBFL) algorithm that clustered EMR data into several communities and simultaneously trained one model per community, so that the learning process became markedly more efficient than FL. Success of data clustering (albeit being centralized analyses) has been reported in previous medical studies such as quality assessment of diabetes physician groups [32], identification of cancer symptom clusters to benefit therapeutics [33], and delineation of chronic pain patient subgroups for improving treatment [34]. In this study, by presenting the development and evaluation of CBFL, we demonstrate the application of decentralized clustering together with federated machine learning to make predictions on ICU EMRs.

2. Materials and methods

2.1. eICU data

CBFL was developed based on the eICU collaborative research database [35], which contains highly granular critical care data of 200,859 patients admitted to 208 hospitals from across the United States. Our study mainly concerned with three variables.

- *Mortality*: unit discharge status that specifies patients' survival/mortality (0 for alive and 1 for expired). This would be the dependent variable in the mortality prediction task.
- *ICU stay time*: unit discharge offset that records the number of days from unit admission to discharge (with an average of 64 h, that is, 2.7 days). This would be the dependent variable in the unit-stay-time prediction task.
- *drug features*: drugs administered on patients during the first 48 h of ICU stay (1399 binary *drug features* in total). Table 1 shows the first three *drug features*: 2 mL of Metoclopramide HCL 5 mg/ml given in injection solution, 3 mL vial of insulin regular human 100 unit/ml given in injection solution, and Metoprolol Succinate ER 50 mg taken orally once per 24 h. If Drug i was prescribed to Patient j, Cell (i, j) in the table would become 1, and 0 otherwise. For instance, Patient 141,194 who received all three drugs had a feature vector of [1, 1, 1], whereas Patient 141,203 who took Metoprolol Succinate ER only had a feature vector of [0, 0, 1]. These *drug features* would be used as independent variables in the prediction tasks of our study.

The proposed CBFL algorithm and the baseline FL algorithm would use *drug features* as predictors to forecast *mortality* and *ICU stay time* of critical care patients. We chose medication as predictors rather than other variables (such as age, gender and diagnosis) because the eICU database contained highly dimensional drug information. In contrast, both age and gender were a single dimensional variable and diagnosis had only dozens of dimensions. Prediction on medication most closely resemble the real case scenario of federated learning, since the technology was devised to tackle the challenge of big data in large volume and with high dimensionality.

Extracting these three variables out of the database yielded a smaller dataset of 126,490 patients coming from 58 hospitals. Furthermore, we selected 50 hospitals whose patient count was over 600 and, from each of them, randomly sampled 560 patients to form the final dataset of 28,000 examples. This data was split into a training set of 20,000 examples and a test set of 8000 examples so that the training-test ratio was about 7:3. Splitting data in this ratio is a common practice in fitting machine learning models to make biomedical predictions [36–39] and solve problems in other fields [40–43]. Results of cohort analysis on these 28,000 patients will be presented in Section 3.1.

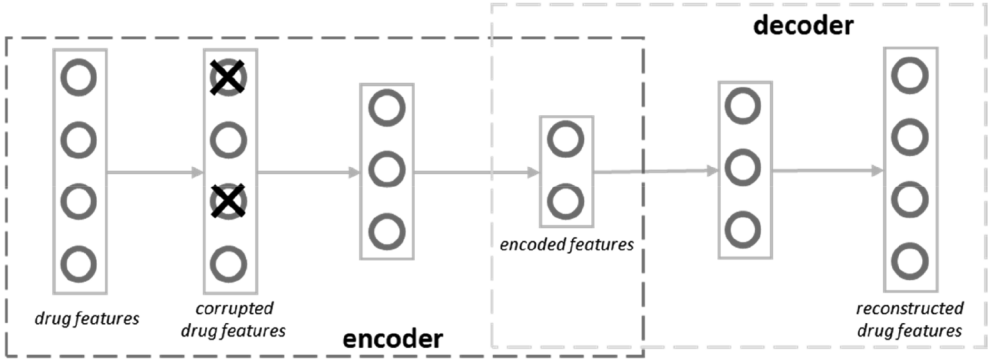


Fig. 1. A small denoising autoencoder of three hidden layers with 3, 2 and 3 units, respectively.

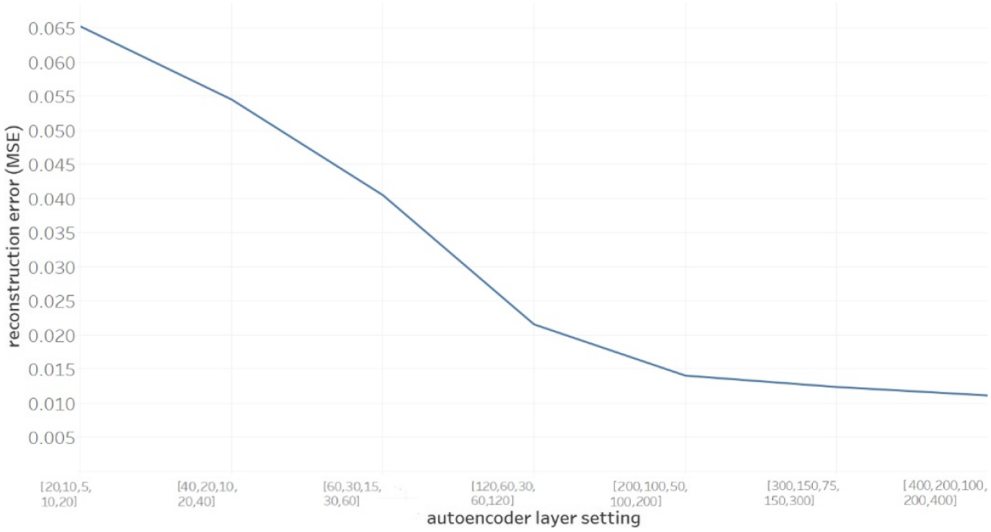


Fig. 2. Grid hyperparameter search for defining the structure of the denoising autoencoder.

2.2. CBFL

Algorithm 1 displays the three procedures involved in CBFL. During encoder training, each client (that is, hospital) learnt a denoising autoencoder $f_{autoencoder}$ initialized with $w_{0,autoencoder}$ for E_1 epochs and returned only the trained weights of encoder $w^c_{1,encoder}$ to the server for average. Here, N, c, n^c, and $f_{encoder}$ denoted the total number of examples, the client index, the size of each client, and the averaged encoder, respectively. Fig. 1 shows the structure of a small denoising autoencoder that contained three hidden layers with 3, 2, and 3 units, respectively. The binary *drug features* would be fed into the input layer with 4 units, and then be stochastically corrupted [44]: each feature would have 50% chance of being forced to zero, regardless of what its original value was. Subsequently, the corrupted *drug features* would be used to train the encoder and the decoder. The former served as a dimensionality reduction function by transforming the highly dimensional features into lower dimensional encodings (known as representation), whereas the latter would reconstruct the original *drug features* from the representation. Performance of the denoising autoencoder was measured by reconstruction error, that is, mean squared error (MSE) between the original *drug features* and the reconstructed ones.

During k-means clustering, each client used $f_{encoder}$ to transform its data into representations X^c and sent the average representation $\bar{X}^c$ to the sever. Then, the server learnt a k-means clustering model f_{kmeans} with K centroids (that is, communities) on $\bar{X}^c$s from all clients.

During community-based learning, the server initialized a series of K neural network models $f_1, f_2, \cdots, f_K$ with the same weights w_0; each client received all K models from the server and learnt each model on its full data for E_2. Meanwhile, $f_{encoder}$ and f_{kmeans} were used to determine which cluster each example belonged to. The size of clusters was denoted $m^c_1, m^c_2, \cdots, m^c_K$ and returned together with the learnt weights to the server, where each model was updated by taking the weighted average of s based on $m^c_1, m^c_2, \cdots, m^c_K$. The updated models were sent to each client for the next round of training. This community-based learning process was repeated until the algorithm converged. The convergence condition was that the weights of the server-side global model converged to specific values, or that the number of maximum communication rounds was reached. Given a test example, CBFL would firstly convert its features into encodings by $f_{encoder}$, then define its community by f_{kmeans} and finally use the corresponding community model to make prediction. These three procedures of CBFL are visualized in Fig. 3.

Algorithm 1. Community-Based Federated Learning

procedure Encoder Training (C, E_1) ▷ 1st procedure
  initialize weights $w_{0,autoencoder}$ of denoising autoencoder $f_{autoencoder}$
  for each client $c = 1, 2, \cdots, C$ **in parallel do**
    train $f_{autoencoder}$ for E_1 epochs to obtain $w^c_{1,encoder}$
    return $w^c_{1,encoder}$ to server ▷ return weights of encoder layers only
  $w_{1,encoder} \leftarrow \sum_{c=1}^{C} \frac{n^c}{N} w^c_{1,encoder}$ ▷ perform one update to obtain $f_{encoder}$
procedure K-means Clustering(K, C) ▷ 2nd procedure
  for each client $c = 1, 2, \cdots, C$ **in parallel do**
    use $f_{encoder}$ to obtain encoded features X^c of each example
    return $\bar{X}^c = \sum_{i=1}^{n^c} \frac{X^i}{n_c}$ to server ▷ return average features
  initialize K cluster centroids from $\{\bar{X}^1, \bar{X}^2, \cdots, \bar{X}^C\}$
  train k-means clustering model f_{kmeans}
  for each client $c = 1, 2, \cdots, C$ **in parallel do**
    use f_{kmeans} on X^c to determine cluster of each example
    count examples in each cluster $\{m^c_1, m^c_2, \cdots, m^c_K\}$
    return $\{m^c_1, m^c_2, \cdots, m^c_K\}$ to server ▷ return example count
procedure Community-based Learning (K, C, E_2) ▷ 3rd procedure
  initialize K community NN models $\{f_1, f_2, \cdots, f_K\}$ with same weights w_0
  while not converged **do**
    for k in 1. ..K **in parallel do** ▷ simultaneously train K models
      for each client $c = 1, 2, \cdots, C$ **in parallel do**
        train f^c_k for E_2 epochs to obtain w^c_k
        return w^c_k to server
    $w_k \leftarrow \frac{\sum_{c=1}^{C} m^c_k w^c_k}{\sum_{c=1}^{C} m^c_k}$ ▷ update weights of each community model
  end while

2.3. Parameter set of CBFL

The denoising autoencoder $f_{autoencoder}$ had a structure of five fully connected hidden layers with 200, 100, 50, 100 and 200 units, respectively, using the rectified linear unit (ReLu) activation function. Given $f_{autoencoder}$ of such structure, an individual hospital holding 560 patients would seem to overfit the autoencoder because of insufficient data size. However, each of the 50 hospitals would learn its own model and send the parameters to the server for averaging. As indicated by McMahan et al. [25], this model-averaging mechanism prevented

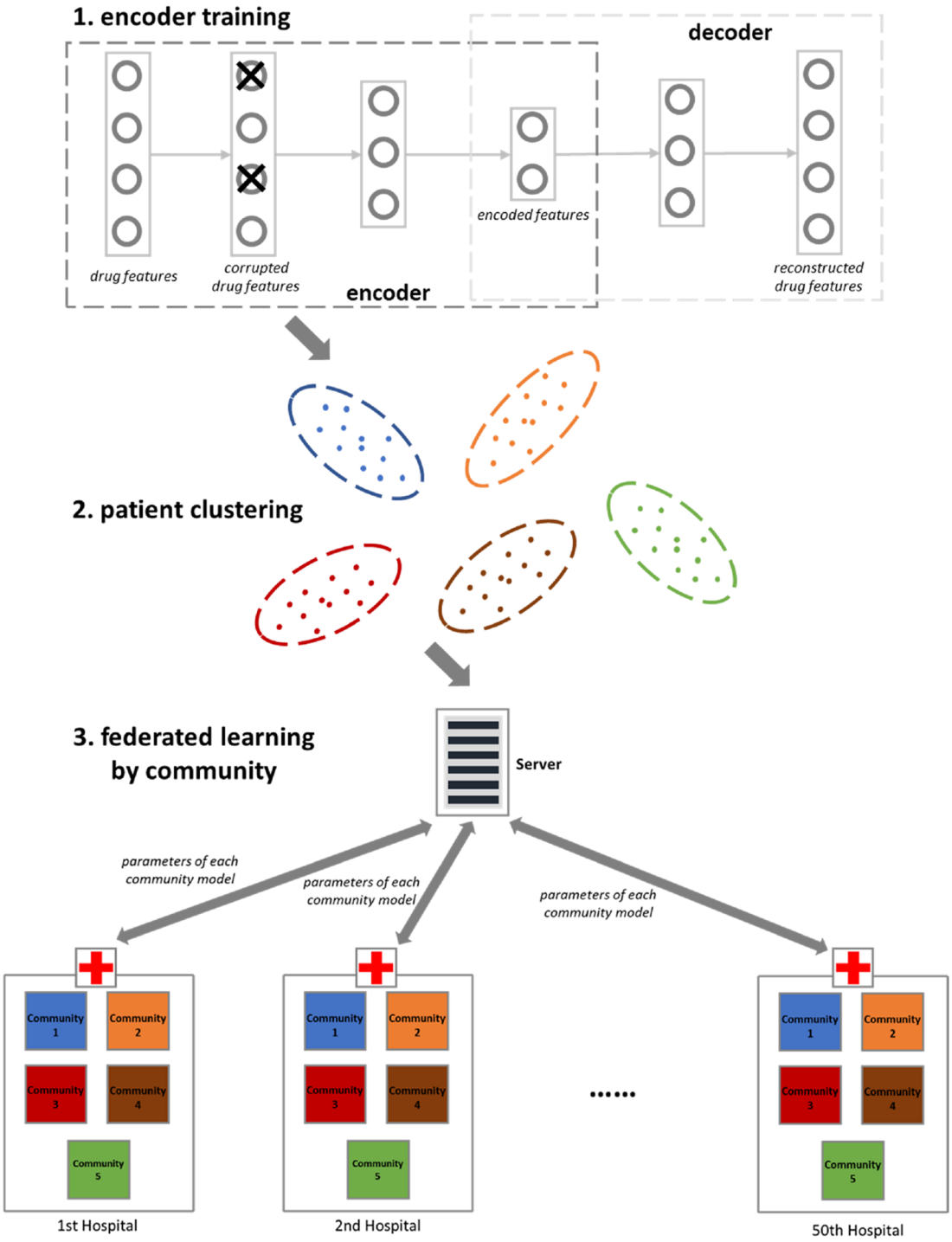


Fig. 3. Flowchart of CBFL. A denoising autoencoder was trained on each hospital's data and averaged at the server. Subsequently, encoder layers were used to convert patients' *drug features* into privacy-preserving representations that were in turn used for patient clustering by *k*-means. In this figure, patients were clustered into five communities as an example. Lastly, each hospital individually learnt five community models and sent them to the server for weighted average based on community size.

overfitting since it produced a beneficial regularization effect similar to that of dropout. In fact, the number of units in each layer of $f_{autoencoder}$ was chosen based on grid hyperparameter search. We fitted denoising autoencoders with various number of units in each of the five hidden layers, including [20, 10, 5, 10, 20], [40, 20, 10, 20, 40], [60, 30, 15, 30, 60], [120, 60, 30, 60, 120], [200, 100, 50, 100, 200], [300, 150, 75, 150, 300], and [400, 200, 100, 200, 400]. For each denoising autoencoder, we calculated the reconstruction error (mean squared error, MSE) between the original drug features and the reconstructed features (see Fig. 2). MSE decreased as the structure became larger, though the slope was flatter. After the point of [200, 100, 50, 100, 200], further increasing the number of units in each layer resulted in only marginal MSE reduction. Therefore, we decided to adopt this structure of denoising autoencoder in the CBFL algorithm.

The output layer of $f_{autoencoder}$ used the sigmoid function because of binary input *drug features*. The number of epochs E_1 was set to five, meaning that each hospital would train $f_{autoencoder}$ on every example for five times. We chose Adaptive Moment Estimation (Adam) as the

stochastic optimizer with default parameters (the learning rate $\eta = 0.001$ and the exponential decay rates for the moment estimates $\beta_1 = 0.9$ and $\beta_2 = 0.999$) on the categorical cross-entropy loss function. For the k-means model f_{kmeans}, we used various numbers of centroids (five, 10, 15 and the extreme case of one centroid per hospital) to evaluate CBFL. Each one of the community models $f_1, f_2, \cdots, f_K$ (and the baseline FL model) consisted of three hidden layers with 20, 10 and five hidden units respectively and activated by ReLu. Here, grid search was not carried out to select the best structure. This is because, unlike $f_{autoencoder}$ that only needed to be fitted once, $f_1, f_2, \cdots, f_K$ would be trained iteratively for many communication rounds. Grid search on them was computationally costly. So, we chose a small structure of 20, 10 and five units for simplicity of calculation. Moreover, because the FL model had the same structure as community models and the purpose of experiments was to compare CBFL and FL rather than fine-tuning models, we think that [20,10,5] would suffice in our study. Same as $f_{autoencoder}$, we used sigmoid as the output layer activation function and Adam with default setting as the optimizer for $f_1, f_2, \cdots, f_K$.

Because both *mortality* and *ICU stay time* were binary response variables, we chose binary cross-entropy as the objective function in our study:

$$\underset{f}{\operatorname{argmin}} - \sum_{i=1}^{N} [y_i \log f(x_i) + (1 - y_i)\log(1 - f(x_i))] \tag{1}$$

where i was the patient index, N was the total number of patients, x_i was the input *drug features* for Patient i, y_i was the binary label for Patient i, and f was the CBFL model. Binary cross-entropy measured the deviation of the prediction $f(x_i)$ from the true value y_i (which was either 0 or 1). We optimized the objective function for the parameters of f. The fine-tuned f should achieve the minimum overall cross-entropy loss, which was calculated by summing up the loss for individual patients' predictions made by f.

2.4. Evaluation metrics

The proposed algorithm was evaluated mainly using Area Under the Receiver Operating Characteristic Curve (ROC AUC). The ROC curve was produced by plotting the true positive rate (TPR or sensitivity) and the false positive rate (FPR or 1-specificity) at thresholds ranging from 0 to 1. The prediction scores (that is, the predicted probabilities of patient mortality) were compared with each threshold: if a patient's score is above the threshold, he/she is predicted to be mortal, or otherwise alive. The confusion matrix is used to summarize the comparison results (see Table 2). Among predicted mortalities, those patients who died fall into the true mortality category and those who survived are false mortalities. Among predicted survivals, those who died are false survivals and those who survived are true survivals.

The two measures that constitute the ROC curve, TPR and FPR, are calculated based on the confusion matrix

$$\text{TPR} = \frac{\text{the number of true motalities}}{\text{the number of actual mortalities}} \tag{2}$$

$$\text{FPR} = \frac{\text{the number of false mortalities}}{\text{the number of actual survivals}} \tag{3}$$

Each point on the ROC curve corresponds to a pair of TPR and FPR for a threshold. Fig. 4 shows five example points for thresholds of 0, 0.01, 0.05, 0.1, and 1. The values of TPR and FPR are firstly computed from the confusion matrix for each threshold and then visualized on the ROC plot. The area under the ROC curve, ROC AUC, is the probability that the classifier will rank a random positive example over a random negative one in terms of prediction scores and regardless of what threshold is chosen. A perfect classifier has an AUC of 1.0, meaning that at any threshold the classifier will always distinguish positive examples from negative ones; a random classifier has an AUC of 0.5, indicating a random guess on which example is positive.

Table 2
The structure of a confusion matrix.

	Actual mortality	Actual survival
Predicted mortality	true mortality	false mortality
Predicted survival	false survival	true survival

Another evaluation metric was Area Under the Precision-Recall Curve (PR AUC). PR AUC was calculated in a similar manner to ROC AUC. The PR curve was generated by plotting *precision* and *recall* (same as TPR) at thresholds ranging from 0 to 1. *Precision* was given by

$$\textit{precision} = \frac{\text{the number of true mortalities}}{\text{the number of predicted survivals}} \tag{4}$$

Unlike ROC AUC, PR AUC did not consider the true survivals in its calculation, and was another measure of predictive accuracy. Lastly, we further used communication cost to assess the convergence speed of the algorithm, that is, how many communication rounds between hospitals and the server were required by the algorithm to converge.

3. Results

3.1. Cohort analysis

This study involved EMRs from 50 hospitals, each containing 560 critical care patients. Table 3 summarizes the study cohort's information, including patient count, gender, age, mortality/survival rate, prolonged stay time rate, and the most frequent diagnosed diseases for ICU patients. The cohort contained more males than females (54.95% versus 45.03%) and most patients (61.82%) were aged above 60 years old. The mortality rate was 4.98%, and 6.12% of the study cohort experienced a prolonged unit stay time. In our study, patients had a prolonged unit stay time if they experienced greater than or equal to eight days of stay, and non-prolonged otherwise. As for the top frequent diagnoses upon ICU admission, patients most likely suffered from diseases related to burns/trauma, cardiovascular, endocrine, gastrointestinal, other general conditions, hematology, infectious diseases, musculoskeletal, neurologic, obstetrics/gynecology, and oncology.

3.2. Community analysis

Patient clustering was a key step in our algorithm: since patients with similar features were grouped together, community-based learning (that is, learning an independent model on each community) would be easier than learning one whole model on all patients. To illustrate what common features were shared among patients in the same community, we clustered the 28,000 patients into five communities and carried out enrichment analysis of diagnoses in them. Table 4 lists the number of patients and overrepresented diagnoses with adjusted p-values within each community. It can be noted that every community exhibited a different focus: for instance, *Community* 1 tended to primarily capture neurologic, endocrine and burns/trauma diseases, whereas *Community* 3 concerned more with pulmonary, cardiovascular and gastrointestinal diseases.

In addition to the above cohort and community analyses considering the characteristics of patients, we further performed clustering at the hospital level to reveal distinctions between hospital communities. Fig. 5 visualizes the 50 hospitals (labeled with their eICU IDs) clustered into five communities on a PCA plot. Separation between communities can be easily recognized, and *Communities* 1 and 5 had a larger size than the rest three. Moreover, geographical bias could be found: *Community* 1 had 15 hospitals located in the Midwest (nine), the South (five) and the West (one) of the United States; *Community* 2 had seven hospitals, all situated in the South; *Community* 3 had eight hospitals, seven of which came from the West and one with unknown

	actual mortality	actual survival
predicted mortality	411	7589
predicted survival	0	0

$$TPR = \frac{351}{351+60} = 0.8540$$
$$FPR = \frac{4764}{4764+2825} = 0.6278$$

a) threshold = 0

	actual mortality	actual survival
predicted mortality	351	4764
predicted survival	60	2825

$$TPR = \frac{351}{351+60} = 0.8540$$
$$FPR = \frac{4764}{4764+2825} = 0.6278$$

b) threshold = 0.01

	actual mortality	actual survival
predicted mortality	240	2268
predicted survival	171	5321

$$TPR = \frac{240}{240+171} = 0.5839$$
$$FPR = \frac{2268}{2268+5321} = 0.2989$$

c) threshold = 0.05

	actual mortality	actual survival
predicted mortality	145	887
predicted survival	266	6702

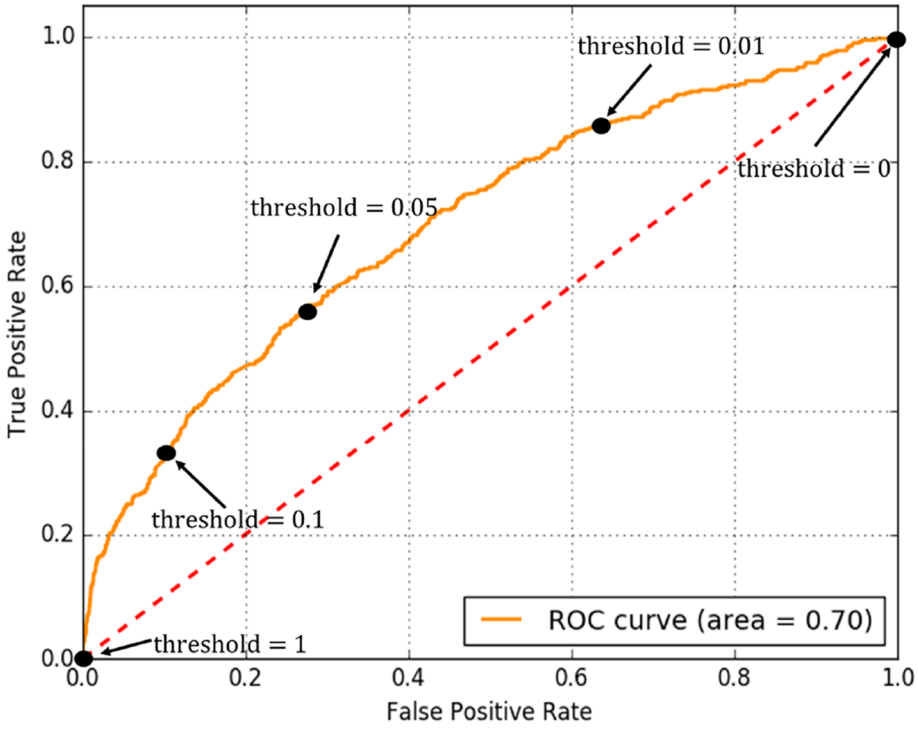
$$TPR = \frac{145}{145+266} = 0.3528$$
$$FPR = \frac{887}{887+6702} = 0.1169$$

d) threshold = 0.1

	actual mortality	actual survival
predicted mortality	0	0
predicted survival	411	7589

$$TPR = \frac{351}{351+60} = 0.8540$$
$$FPR = \frac{4764}{4764+2825} = 0.6278$$

e) threshold = 1



f) ROC curve

Fig. 4. Example points on the ROC curve for CBFL in the mortality prediction task using the same hospitals in the training and test sets. (a) to (e) illustrate calculation of TPR and FPR for thresholds of 0, 0.01, 0.05, 0.1 and 1, respectively; (f) shows the five thresholds on the ROC plot.

location; *Community* 4 had three Midwestern hospitals and two Western hospitals; *Community* 5 had 15 hospitals, one with unknown location and the others residing in the Northeast (three), the Midwest (five), the South (four), and the West (two). In summary, *Community* 2 seemed to capture Southern hospitals only and *Community*3 tended to accommodate hospitals from the West, while no notable bias was observed in the other communities. Supplementary Table 1 contains full information of each hospital.

Combining patient clusters from individual hospitals could be achieved at the server level to further enhance the performance of CBFL. For instance, if Cluster 1 from Hospital 66 was closer to Cluster 2 from Hospital A than to other Hospital B's clusters, then the two clusters could be combined into a single cluster; the new cluster would be used to train the server-side Community Models 1 and 2 so that the training data size was increased for both models. This could be implemented by comparing the centroids of each hospital's clusters, which would require each hospital to send to the server not only the parameters of its community models but also its cluster centroids. Although additional hospital-to-server communication cost and extra computational cost at the server would be incurred, we regard this optimized clustering

Table 3
Cohort analysis of 28,000 patients from 50 hospitals.

	Count	Percentage
Patients	28,000	–
Male	15,386	54.95%
Female	12,609	45.03%
Unknown gender	5	0.02%
age ≤ 20	267	0.95%
20 < age ≤ 40	2552	9.11%
40 < age ≤ 60	7943	28.37%
60 < age ≤ 80	12,630	45.11%
80 < age	4680	16.71%
Death	1395	4.98%
Alive	26,605	95.02%
Patients with prolonged unit stay time	1713	6.12%
Top 10 frequent diagnoses upon admission	●burns/trauma ●cardiovascular ●endocrine ●gastrointestinal ● general ●hematology ●infectious diseases ●musculoskeletal ● neurologic ●obstetrics/gynecology ●oncology	

Table 4
Enrichment analysis of diagnoses in five patient communities.

	Patient count	Overrepresented diagnoses (with adjusted *p*-values)
Community 1	5027	● neurologic (1.10e-20) ● endocrine (4.95e-14) ● burns/trauma (1.11e-11) ● hematology (5.23e-09) ● infectious diseases (1.57e-06) ● renal (7.22e-06)
Community 2	6726	● cardiovascular (1.27e-29) ● transplant (0.00105) ● hematology (0.00667) ● oncology (0.0249)
Community 3	2322	● pulmonary (3.00e-26) ● cardiovascular (9.99e-14) ● gastrointestinal (1.19e-10)
Community 4	6247	● pulmonary (3.97e-28) ● cardiovascular (1.05e-25) ● toxicology (0.00201)
Community 5	7678	● endocrine (1.75e-24) ● burns/trauma (5.90e-24) ● hematology (9.77e-12) ● infectious diseases (9.81e-11) ● gastrointestinal (1.72e-10) ● toxicology (1.37e-06) ● oncology (4.08e-05) ● general (0.00347) ● transplant (0.0138) ● surgery (0.0172)

method as a valuable continuation of our study in the future.

3.3. Mortality prediction

3.3.1. Same hospitals in training and test sets

Mortality was predicted based on patients' prescribed *drug features*. The training dataset was formed by randomly selecting 400 patients from each of the 50 hospitals, and thus had a size of 20,000 examples; the test dataset contained the remaining 160 patients from each hospital, totaling 8000 examples. All patients were labeled with their unit discharge status (1 for mortality and 0 for alive). Evaluation metrics included not only predictive accuracy on the mortality prediction task (that is, forecasting the probability of mortality base on drugs prescribed to patients) in terms of ROC AUC and PR AUC, but also the number of communication rounds between the server and hospitals to complete the training process. The ROC curve was generated by plotting true positive rate (TPR) versus false positive rate (FPR), while the PR curve was produced by plotting positive predictive value (PPV) against TPR. In our study, ROC AUC referred to the probability that CBFL would rank a randomly chosen patient as experiencing mortality over survival, while PR AUC indicated the average precision across the recall range between 0 and 1.

Fig. 6 illustrates the curves of ROC AUC versus communication rounds for FL, CBFL with five communities, 10 communities, 15 communities and the extreme case of 50 communities (that is, one expected community per hospital). Two major messages are conveyed by the plots. First, CBFL consistently outperformed FL by converging to higher ROC AUCs with fewer communication rounds. FL achieved a final ROC AUC of 0.6895 and a PR AUC of 0.1107 in 101 rounds, whereas CBFL with five communities obtained a ROC AUC of 0.6984 and a PR AUC of 0.1430 in 75 rounds (see Table 5). Second, clustering patients into more communities tended to cause CBFL to overfit, yielding slightly reduced ROC AUCs and PR AUCs, but also caused the algorithm to converge faster. For instance, when the number of communities was increased to 15, the ROC AUC, the PR AUC and communication rounds decreased to 0.6935, 0.1339 and 46, respectively. Neither FL nor CBFL performed better than centralized learning with a ROC AUC of 0.7368 and a PR AUC of 0.1449. This superiority of centralized learning was in line with reported in the literature of FL [25].

3.3.2. Different hospitals in training and test sets

To evaluate the robustness of our model given different training/test data distributions, we prepared a training set of 19,600 examples from randomly chosen 35 hospitals and a test set of 8400 from the remaining 15 hospitals. Unlike Section 3.3.1, no random split was performed in individual hospitals and patients in each hospital were used together. ROC AUC, PR AUC and communication rounds were used as evaluation metrics.

Fig. 7 and Table 6 depict performance comparison between FL and CBFL with 5, 10, 15 and 35 communities. There was a drop in ROC AUC for centralizing learning (from 0.7368 to 0.6811), FL (from 0.6895 to 0.6520), and CBFL (from 0.69+ to 0.65+), resonating with the fact that inconsistent training/set data distributions lead to less effective learning. Nonetheless, both FL and CBFL converged faster, which we speculate resulted from having data from fewer hospitals (35, compared with 50 in previous evaluation) in the training dataset. Specifically, FL reached its peak (a ROC AUC of 0.6520 and a PR AUC of 0.0871) in 66 communication rounds, and CBFL with 10 communities performed better than CBFL with any other community number, converging to a ROC AUC of 0.6628 and a PR AUC of 0.0912 in 27 rounds. In addition, overfitting was observed: the more communities were clustered, the lower the ROC AUC score, albeit with fewer communication rounds.

3.4. Stay time prediction

3.4.1. Same hospitals in training and test sets

Like *mortality*, prediction of prolonged ICU *stay time* was based on *prescribed drug features*, with patients split in the same way as in Section 3.3.1 to form the training and test datasets and assessed by the same evaluation metrics. Performance comparison of FL and CBFL is demonstrated in Fig. 8 and Table 7. Here the AUC ROC gap (0.02) was wider than that of the *mortality* prediction task (0.01). FL achieved a ROC AUC of 0.6360 and a PR AUC of 0.0816 in 123 communication rounds, whereas the most performant CBFL with five communities obtained a ROC AUC of 0.6512 and a PR AUC of 0.0910 in 87 rounds. Again, the effect of overfitting became more severe as the number of communities rose.

3.4.2. Different hospitals in training and test sets

When the training and test datasets were prepared in the same manner as in Section 3.3.2 such that they came from different

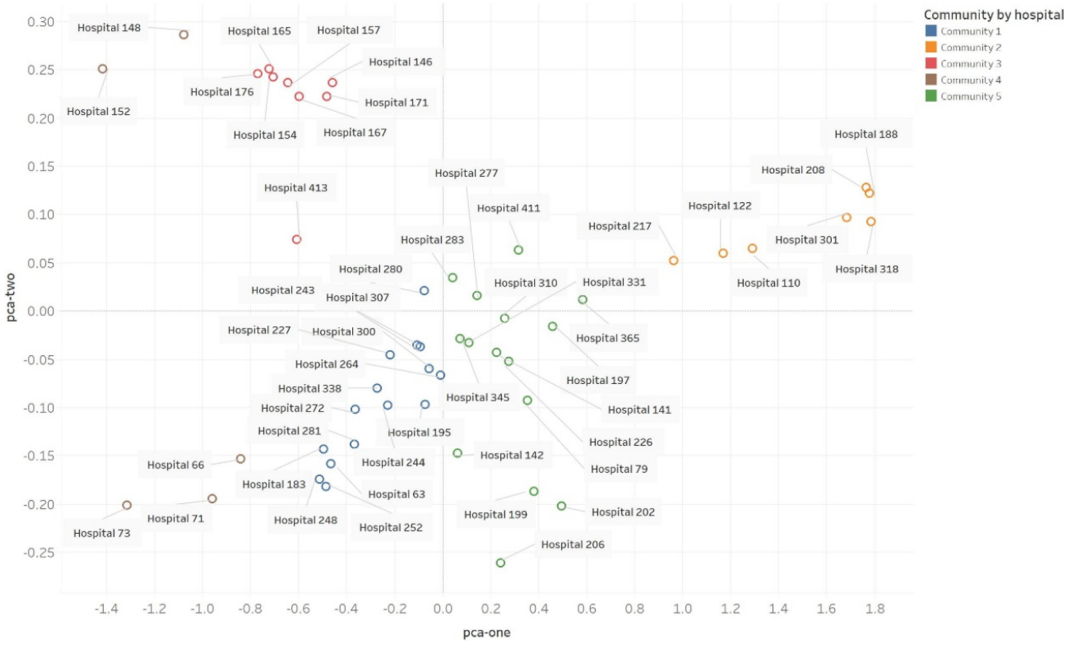


Fig. 5. Communities by hospital (50 hospitals clustered into five communities and visualized on a PCA scatterplot).

distributions, ROC AUCs reduced significantly from 0.7083 to 0.6189 for centralized learning, from 0.6360 to 0.6212 for FL, and from 0.63+ to 0.62+ for CBFL, despite faster convergence (see Fig. 9 and Table 8). It is worth noting that FL and CBFL outperformed centralized learning, and we conjecture the reason to be that in this particular task of predicting prolonged *stay time* on different training and test data, the beneficial effect of regularization [25] in federated machine learning outweighed the adverse effect of decentralized information loss during distributed communication. With five communities, CBFL exhibited the highest ROC AUC of 0.6400 and a PR AUC of 0.0822 in 23 communication rounds. The impact of overfitting on communication cost was less observable than that in previous sections, since raising the number of communities from five to 10 or 15 resulted in more communication rounds (31) rather than fewer.

Table 5
Summary of ROC AUCs, PR AUCs, and communication rounds after convergence in the *mortality* prediction task; centralized learning, FL and CBFL with 5, 10, 15 and 50 communities were compared; training and test data came from the same 50 hospitals.

	ROC AUC	PR AUC	Communication rounds
Centralized learning	0.7368	0.1449	–
FL	0.6895	0.1107	101
CBFL: 5 communities	0.6984	0.1430	75
CBFL: 10 communities	0.6989	0.1070	57
CBFL: 15 communities	0.6935	0.1339	46
CBFL: 50 communities	0.6913	0.1168	33

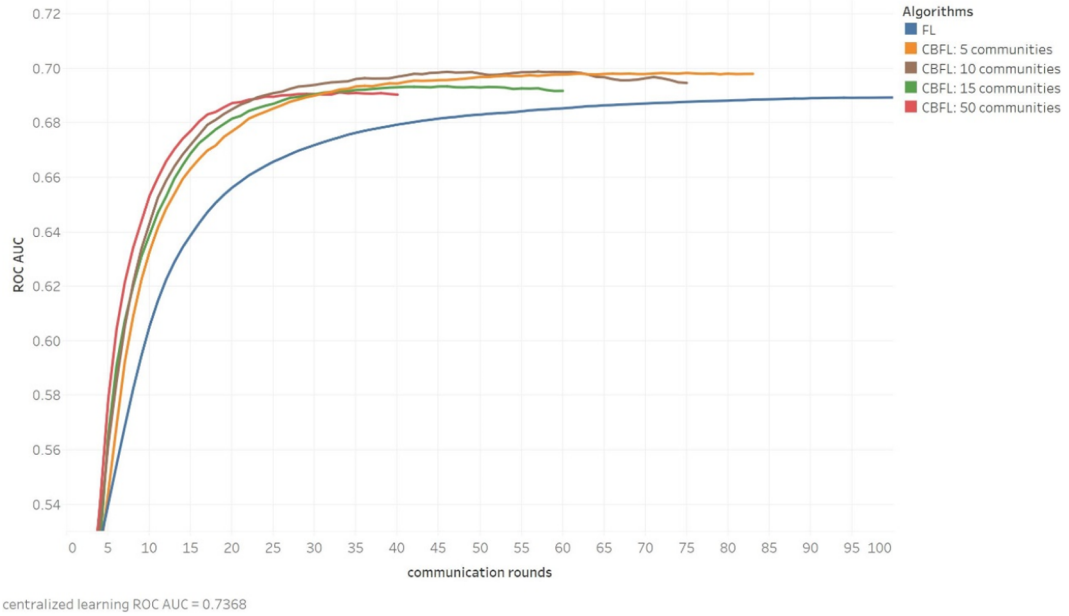


Fig. 6. Plot of ROC AUC against communication rounds in the *mortality* prediction task; FL and CBFL with 5, 10, 15 and 50 communities were compared; training and test data came from the same 50 hospitals.

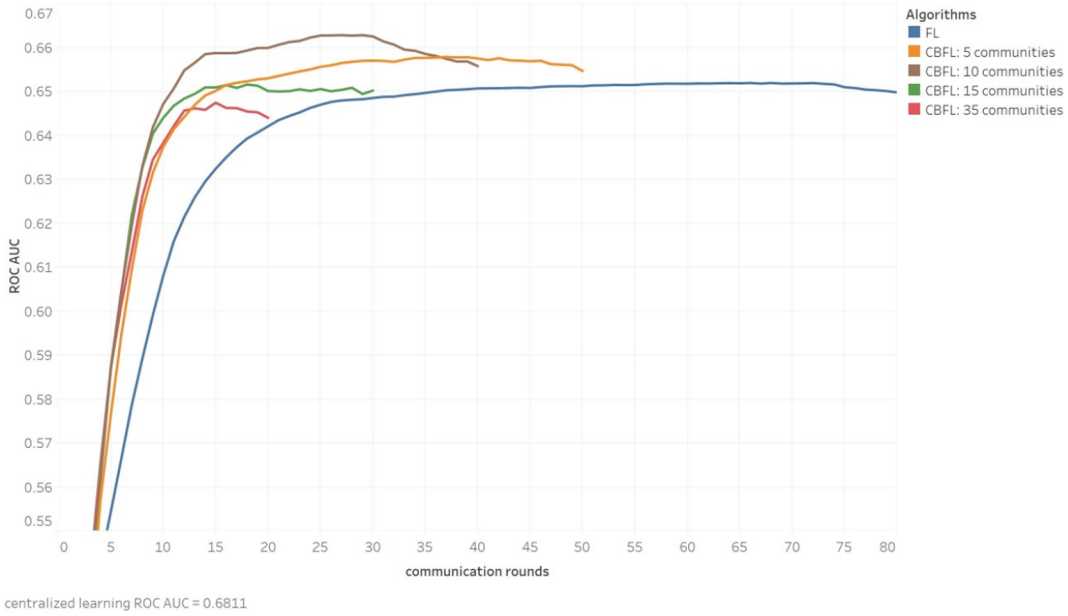


Fig. 7. Plot of ROC AUC against communication rounds in the *mortality* prediction task; FL and CBFL with 5, 10, 15 and 35 communities were compared; training data came from randomly chosen 35 hospitals and test data from the remaining 15 ones.

Table 6
Summary of ROC AUCs, PR AUCs, and communication rounds after convergence in the *mortality* prediction task; centralized learning, FL and CBFL with 5, 10, 15 and 35 communities were compared; training data came from randomly chosen 35 hospitals and test data from the remaining 15 ones.

	ROC AUC	PR AUC	Communication rounds
Centralized learning	0.6811	0.0947	–
FL	0.6520	0.0871	66
CBFL: 5 communities	0.6579	0.0893	37
CBFL: 10 communities	0.6628	0.0912	27
CBFL: 15 communities	0.6516	0.1145	18
CBFL: 35 communities	0.6475	0.0920	15

Table 7
Summary of ROC AUCs, PR AUCs, and communication rounds after convergence in the *stay time* prediction task; centralized learning, FL and CBFL with 5, 10, 15 and 50 communities were compared; training and test data came from the same 50 hospitals.

	ROC AUC	PR AUC	Communication rounds
Centralized learning	0.7083	0.1145	–
FL	0.6360	0.0816	123
CBFL: 5 communities	0.6512	0.0910	87
CBFL: 10 communities	0.6527	0.0607	89
CBFL: 15 communities	0.6449	0.0549	61
CBFL: 50 communities	0.6353	0.0840	28

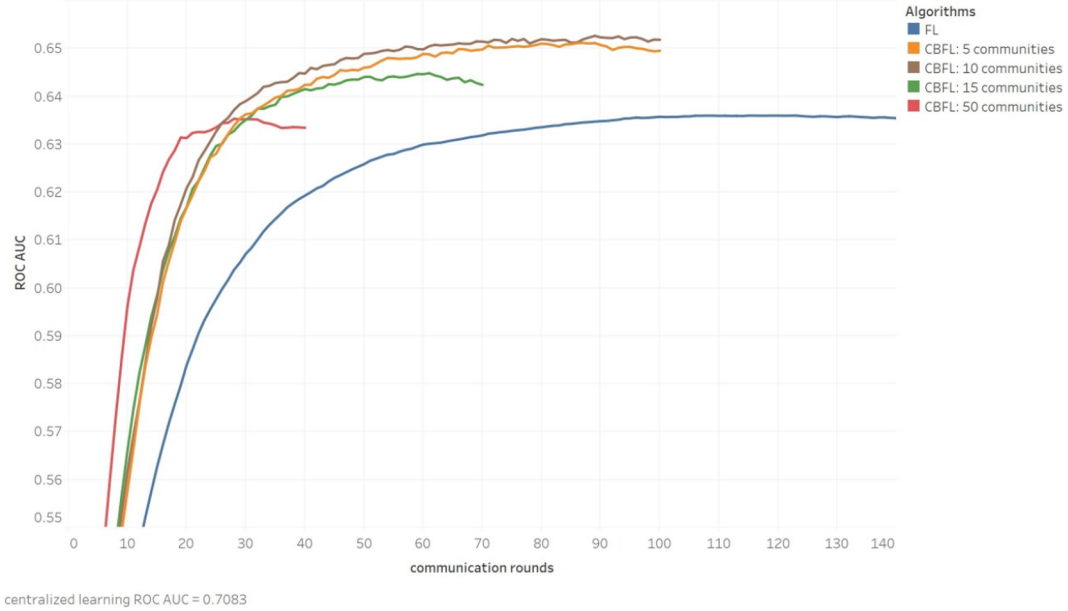


Fig. 8. Plot of ROC AUC against communication rounds in the *stay time* prediction task; FL and CBFL with 5, 10, 15 and 50 communities were compared; training and test data came from the same 50 hospitals.

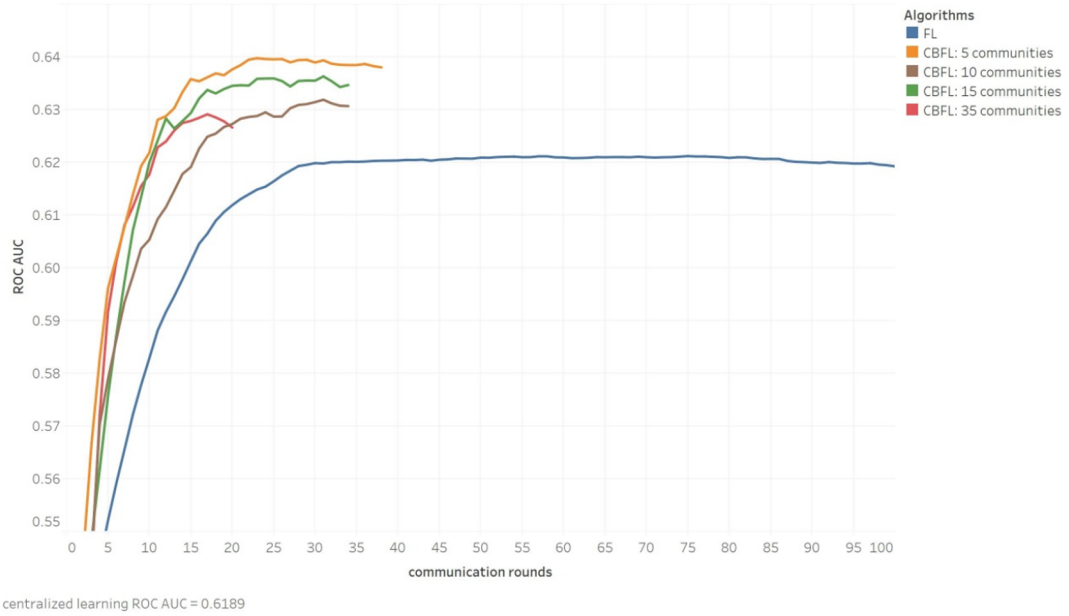


Fig. 9. Plot of ROC AUC against communication rounds in the *stay time* prediction task; FL and CBFL with 5, 10, 15 and 35 communities were compared; training data came from randomly chosen 35 hospitals and test data from the remaining 15 ones.

Table 8
Summary of ROC AUCs, PR AUCs, and communication rounds after convergence in the *stay time* prediction task; centralized learning, FL and CBFL with 5, 10, 15 and 35 communities were compared; training data came from randomly chosen 35 hospitals and test data from the remaining 15 ones.

	ROC AUC	PR AUC	Communication rounds
Centralized learning	0.6189	0.0620	–
FL	0.6212	0.0720	75
CBFL: 5 communities	0.6400	0.0822	23
CBFL: 10 communities	0.6319	0.0786	31
CBFL: 15 communities	0.6364	0.0836	31
CBFL: 35 communities	0.6292	0.0694	17

3.5. Statistical significance of experimental results

To examine statistical significance of results in the four prediction tasks, we repeated experiments on FL for five times (with different random seeds for partitioning training/test data). As for CBFL, because the five-community setting stroke a good balance between predictive performance and computational cost, we decided to carry out repeated experiments for CBFL with five communities only. Fig. 10 shows the 95% confidence intervals (dashed lines) and the average (solid lines) of ROC AUC values for FL and CBFL in the mortality and ICU-stay-time prediction tasks. CBFL consistently converged to higher AUCs at fewer communication rounds than FL. In addition, we performed paired *t*-test for equal means on the average AUCs of CBFL and FL, and obtained *p*-values less than 2.2e-16 for all four plots. By observing the figure and from the statistical tests, it can be concluded that CBFL statistically significantly outperformed the baseline.

3.6. Community distribution analysis

The abovementioned evaluation results reveal that CBFL had better predictive accuracy in fewer communication rounds than FL in both mortality and stay time prediction tasks. Communities tended to accommodate patients of similar diagnoses and geographical locations, making individual community models on average easier to learn than one model for all patients. In this section, we took CBFL with five communities for mortality prediction as an example to investigate and illustrate the performance differences of each community model. As shown in Table 9, *Community* 1 exhibited the highest ROC AUC of 0.7561 and *Community* 4 yielded the highest PR AUC of 0.2155, while *Community* 2, the only one underperforming FL, obtained the worst performance with a ROC AUC of 0.6179 and a PR AUC of 0.0773. This can be explained by the average distance of each community centroid to other community centroids on the PCA plot (see the third column of Table 9): *Community* 2 was the furthest apart from the rest, with an average distance of 2.562. In addition, ROC curves for the community models were plotted in Fig. 11.

3.7. Clinical use of CBFL

The aforementioned evaluation was mainly based on ROC AUCs that required computation of TPR and FPR at various thresholds between 0 and 1. To enable clinical use of CBFL, a single threshold should be chosen and two methods of defining the appropriate value are recommended. One is related to prevalence of mortality (p) in the population. Firstly, p should be estimated from training examples, on which the CBFL model is learnt. Then, CBFL is used to generate the training examples' prediction scores, the $[(1-p)\times 100]$th percentile of which should be selected as the threshold. In our data, 1395 out of the 28,000 critical care patients expired and so $1-p$ approximately equaled 95%. The 95th percentile of the prediction scores given by CBFL was 0.1741. If a patient's score surpassed this threshold, he/she would have a mortality prediction, or otherwise a survival prediction. Table 10 shows the confusion matrix that summarizes predictions of the 8000 test examples.

The other method concerns with the Youden Index J [45] that measures the diagnostic performance of a binary classifier. By estimating J, the optimal cut-off may be found

$$J = \underset{c}{\text{argmax}} \text{TPR}_c - \text{FPR}_c \tag{5}$$

where c was the cut-off for the algorithm and became optimal when the difference between TPR and FPR reached its maximum. Applying this method to our test data, J was estimated to be 0.2918 at the optimal threshold of 0.0550, which resulted in a confusion matrix in Table 11.

Using which of the two threshold methods depends on what metrics clinicians care more about. By observing Tables 10 and 11, a major

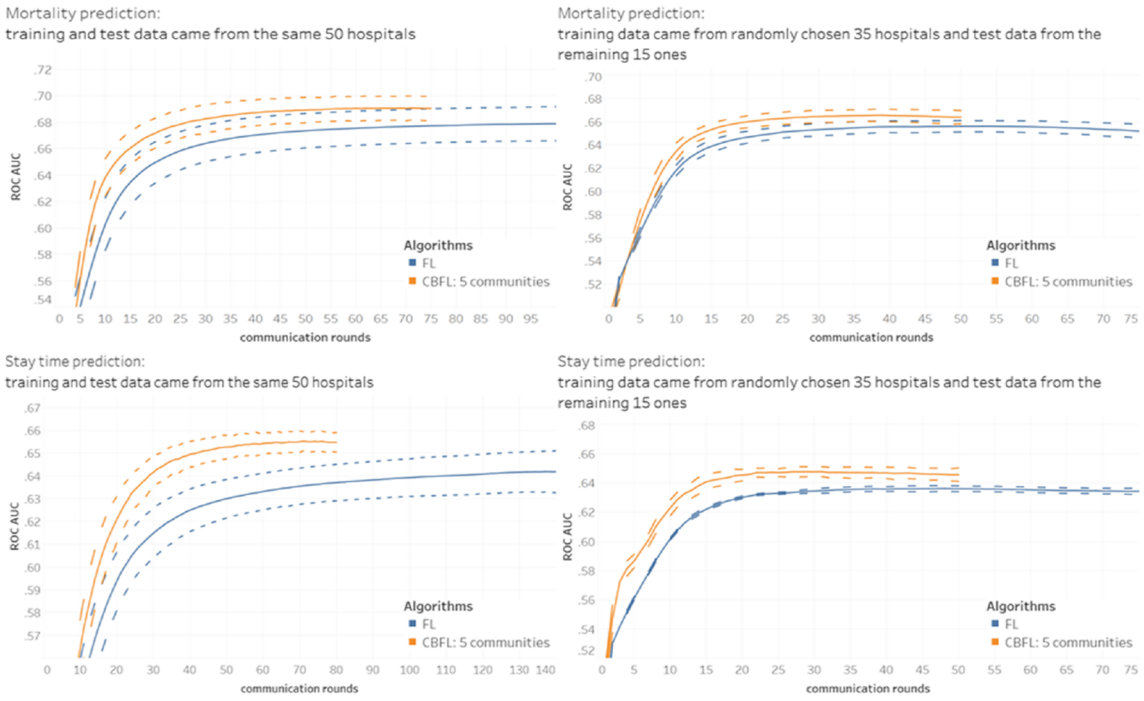


Fig. 10. 95% confidence intervals for FL and CBFL with five communities in the mortality and ICU-stay-time prediction tasks. The upper left shows AUCs of mortality prediction for training and testing data from the same 50 hospitals; the upper right shows AUCs of mortality prediction for training data from randomly chosen 35 hospitals and test data from the remaining 15 ones; the lower left shows AUCs of stay time prediction for training and testing data from the same 50 hospitals; the lower right shows AUCs of stay time prediction for training data from randomly chosen 35 hospitals and test data from the remaining 15 ones. All results indicate that CBFL significantly outperformed FL by converging to higher AUCs at fewer communication rounds.

Table 9
CBFL with 5 communities in the *mortality* prediction task; community comparison in terms of ROC AUCs, PR AUCs, and average distance to other communities, with FL's ROC AUC and PR AUC as a reference.

	ROC AUC	PR AUC	Average distance to other communities
FL	0.6895	0.1107	–
CBFL Community 1	0.7561	0.1291	1.253
CBFL Community 2	0.6179	0.0773	2.562
CBFL Community 3	0.7168	0.1321	1.420
CBFL Community 4	0.7454	0.2155	2.101
CBFL Community 5	0.6967	0.1548	1.397

difference between the two thresholds is that 0.1741 led to a larger number of correct predictions than 0.0550 (7388 versus 5717), but yielded lower positive predictive value (PPV, 0.2068 versus 0.5693) and negative predictive value (NPV, 0.9573 versus 0.9687). PPV represented the probability that a patient with predicted mortality truly dies, and NPV denoted the probability of dying when survival was predicted.

To evaluate performance in clinical settings, using only ROC AUC may not be enough because of its inability to describe clinical gain or loss of a classifier [46,47]. Therefore, we adopted net benefit proposed by Halligan et al. [47] as another performance measure in our study

$$net\ \ benefit = \Delta sensitivity + \Delta specificity \times \frac{1}{W} \times \frac{1-p}{p} \tag{6}$$

where

$$\Delta sensitivity = sensitivity_{\text{CBFL}} - sensitivity_{\text{baseline}} \tag{7}$$

$$\Delta specificity = specificity_{\text{CBFL}} - specificity_{\text{baseline}} \tag{8}$$

The weighting factor W was used to control the effect of specificity reduction, and p was the prevalence of abnormality (mortality in our study) in the population. A positive net benefit indicates that the proposed method performed better than the baseline, zero means no difference, and a negative value implies worse performance. We used the mortality prediction task with the same hospitals in training and test sets to demonstrate the superiority of CBFL over the baseline FL algorithm. W was set to the consensus value of 3 according to Halligan et al. and p was estimated to 0.0498 based on cohort analysis in Section 3.1. Sensitivity and specificity of CBFL were 0.5693 and 0.7225 respectively at the optimal threshold of 0.0550. Sensitivity and specificity of FL were 0.6399 and 0.6454 respectively at the optimal threshold of 0.0449 (determined by the Youden Index method). Given these values, net benefit of CBFL was calculated as

$$\begin{aligned} net\ \ benefit &= (0.5693 - 0.6399) + (0.7225 - 0.6454) \times \frac{1}{3} \times \frac{1-0.0498}{0.0498} \\ &= 0.4197 \end{aligned} \tag{9}$$

A positive net benefit of 0.4197 indicated that CBFL outperformed the baseline.

4. Discussion

Patients admitted to ICUs come from diverse ethnic and age groups, exhibit various levels of vital sign measurements and illness severity, and receive different diagnoses and treatment [35]. Among these dimensions, CBFL focused primarily on admission diagnoses for patients' unit stay and also on geographical locations of hospitals. By clustering patients of common features into the same community and learning separate models for individual communities, the algorithm converged to higher predictive accuracy in fewer communication rounds than the baseline FL model in both mortality and stay time prediction tasks. Clustering also made prediction results interpretable: analyzing the distances between communities could help explain why prediction on some examples was more reliable than on others (refer to Table 8 for an example). Moreover, unlike other optimization algorithms for federated learning on non-IID data [27–29] that needed a fraction of all data to be

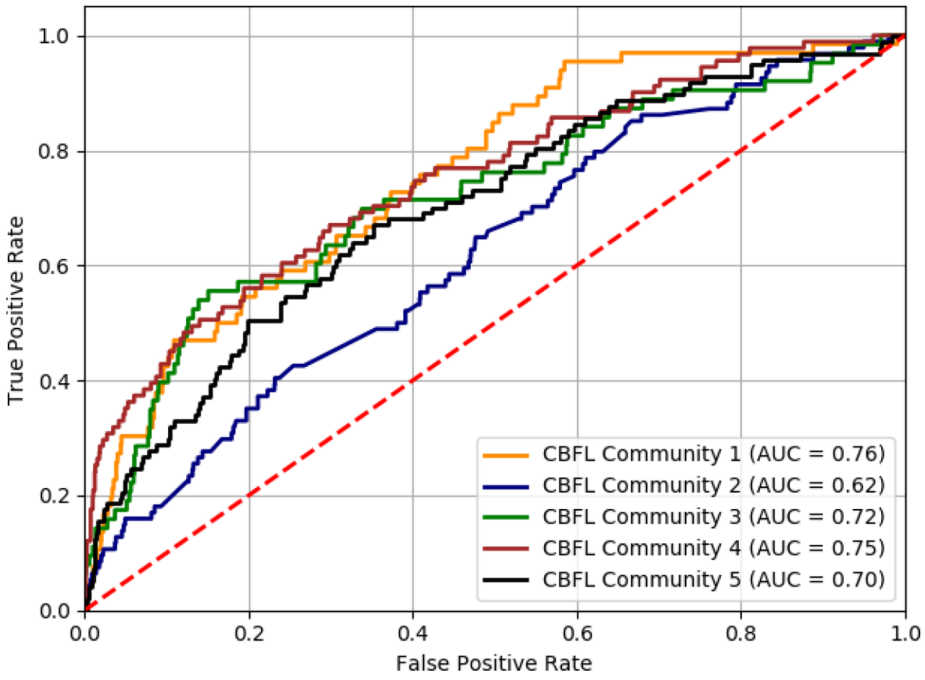


Fig. 11. ROC curves for individual community models.

Table 10
Confusion matrix for CBFL with a threshold of 0.1741.

	Actual mortality	Actual survival
Predicted mortality	85	286
Predicted survival	326	7303

Table 11
Confusion matrix for CBFL with the optimal threshold of 0.0550.

	Actual mortality	Actual survival
Predicted mortality	234	2106
Predicted survival	177	5483

shared across the clients, CBFL did not require hospitals to share their patient data with each other or the server at all, thereby keeping privacy intact. Any data sent to the server for fitting the clustering model f_{kmeans} was firstly encoded by $f_{encoder}$ and, since the decoder was discarded at the end of training $f_{autoencoder}$ on each client, recovering original *drug features* from encoded representation was nearly impossible. On the other hand, a limitation of CBFL was that, if K community models were trained on each client, then $K-1$ times more model parameters would be transferred between the clients and the server than those of FL. Such additional communication load would escalate with increasing training samples and communities. Fortunately, experimental results show that CBFL performed the best with five or at most ten communities. Nevertheless, there is no guarantee that applications of CBFL on other biomedical datasets will also be the most performant with fewer communities.

Future research directions may include optimization of communication load by devising more efficient community-based learning schemes and better clustering methods. For instance, as mentioned in Section 3.2, combining clusters of patients from different hospitals can be implemented at the server level to increase the training set size for community models. The extent of performance improvement by this method is worth investigating.

In addition, grouping patients based on their diagnoses could be implemented in the future, though we reckon that the performance would probably be similar to that of clustering on medication. This is because diagnoses and drugs prescribed to patients for curing the diagnosed diseases are highly correlated. However, we expect that combining diagnoses and drug features as the basis for patient clustering would yield a better performance than using only one of the two variables. We did not carry out experiments to verify this because using medication was a proof of concept, indicating that privacy-preserving clustering on distributed medical data was feasible. Including more variables such as gender, age group and diagnosis in clustering and finding out the best combination can be another future work.

Apart from clustering, these additional features may also be included in prediction. In this study, we used medication as predictors to demonstrate that federated learning was applicable to the medical field and that CBFL would exhibit better performance than the baseline. Age, gender, diagnosis and any other variable related to critical care patients could all be included in the prediction tasks and would expectedly yield a higher predictive accuracy.

5. Conclusions

This study presents a novel federated machine learning model CBFL that seeks to tackle the challenge of non-IID ICU patient data that complicated decentralized learning, cluster patients into clinically meaningful communities, and optimize performance of predicting *mortality* and *ICU stay time*. Although it is self-evident that learning on communities of similar patients was easier than on dissimilar ones, finding out similar patients and grouping them together required sharing individual patients' information, which is undesirable in federated learning on decentralized medical data. The value of our work lies in that we devised a novel method that clustered patients without privacy breach and a learning strategy that trained multiple community models to harness the clustered data. Our model was evaluated against the baseline FL model on three metrics, namely, ROC AUC that quantifies the likelihood of a model ranking a positive example over a negative one, PR AUC that measures prediction success of a model given datasets with imbalanced labels, and communication rounds that indicate a model's learning speed. Moreover, net benefit was introduced to measure clinical gain of CBFL at the optimal threshold defined by the Youden Index method. Experimental results show that CBFL had

predictive accuracy close to that of centralized learning, hence alleviating the non-IID problem, and that it outperformed FL in terms of all three metrics and in every prediction task, whether it be mortality or stay time prediction, and with or without same training/test data distributions. Patient communities formed by CBFL contained different overrepresented diagnoses and seemed to accommodate hospitals from diverse geographical locations. In addition, performance differences in communities could be attributed to Euclidean distances on the PCA plot. A last point to make is that, while this study concerned with machine learning on ICU EMRs, CBFL can be extended to other data types as long as the data is distributed across different silos and privacy sensitive, to other prediction tasks in the biomedical field such as forecasting disease incidence rate, recognizing medical images or inferring gene-disease associations, and to tasks in other fields such as finance and social science.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jbi.2019.103291.

References

[1] R.D. Cebul, T.E. Love, A.K. Jain, C.J. Hebert, Electronic health records and quality of diabetes care, N. Engl. J. Med. 365 (9) (2011) 825–833.

[2] A. Neumann, E. Kalenderian, R. Ramoni, A. Yansane, B. Tokede, J. Etolue, R. Vaderhobli, K. Simmons, J. Even, J. Mullins, S. Kumar, S. Bangar, K. Kookal, J. White, M. Walji, Evaluating quality of dental care among patients with diabetes: adaptation and testing of a dental quality measure in electronic health records, J. Am. Dent. Assoc. 148 (9) (2017) 634–643 e1.

[3] D. Dorr, L.M. Bonner, A.N. Cohen, R.S. Shoai, R. Perrin, E. Chaney, A.S. Young, Informatics systems to promote improved care for chronic illness: a literature review, J. Am. Med. Inform. Assoc. 14 (2) (2007) 156–163.

[4] T. Bodenheimer, E.H. Wagner, K. Grumbach, Improving primary care for patients with chronic illness, JAMA 288 (14) (2002) 1775–1779.

[5] T. Bodenheimer, E.H. Wagner, K. Grumbach, Improving primary care for patients with chronic illness: the chronic care model, part 2, JAMA 288 (15) (2002) 1909–1914.

[6] V. Podichetty, D. Penn, The progressive roles of electronic medicine: benefits, concerns, and costs, Am. J. Med. Sci. 328 (2) (2004) 94–99.

[7] S.J. Simon, S.J. Simon, An examination of the financial feasibility of electronic medical records (EMRs): a case study of tangible and intangible benefits, Int. J. Electron. Healthc. 2 (2) (2006) 185–200.

[8] M.J. Tierney, N.M. Pageler, M. Kahana, J.L. Pantaleoni, C.A. Longhurst, Medical education in the electronic medical record (EMR) era: benefits, challenges, and future directions, Acad. Med. 88 (6) (2013) 748–752.

[9] S. Mani, Y. Chen, T. Elasy, W. Clayton, J. Denny, Type 2 diabetes risk forecasting from EMR data using machine learning, AMIA Ann. Symp. Proc. 2012 (2012) 606–615.

[10] T. Tran, T.D. Nguyen, D. Phung, S. Venkatesh, Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM), J. Biomed. Inform. 54 (2015) 96–105.

[11] C.S. Lee, D.M. Baughman, A.Y. Lee, Deep learning is effective for classifying normal versus age-related macular degeneration OCT images, Ophthalmol. Retina 1 (4) (2017) 322–327.

[12] K. Shameer, K.W. Johnson, A. Yahi, R. Miotto, L.I. Li, D. Ricks, J. Jebakaran, P. Kovatch, P.P. Sengupta, S. Gelijns, A. Moskovitz, B. Darrow, D.L. David, A. Kasarskis, N.P. Tatonetti, S. Pinney, J.T. Dudley, Predictive modeling of hospital readmission rates using electronic medical record-wide machine learning: a case-study using Mount Sinai heart failure cohort, Pac. Symp. Biocomput. 22 (2017) 276–287.

[13] R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Brief. Bioinform. 19 (6) (2018) 1236–1246.

[14] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, F. Wang, Secure and trustable electronic medical records sharing using blockchain, AMIA Ann. Symp. Proc. 2017 (2017) 650–659.

[15] J.-J. Yang, J.-Q. Li, Y. Niu, A hybrid solution for privacy preserving medical data sharing in the cloud environment, Fut. Gener. Comput. Syst. 43 (2015) 74–86.

[16] R. Wu, G.-J. Ahn, H. Hu, Secure sharing of electronic health records in clouds, 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), IEEE, 2012, pp. 711–718.

[17] G. Perera, A. Holbrook, L. Thabane, G. Foster, D.J. Willison, Views on health information sharing and privacy from primary care practices using electronic medical records, Int. J. Med. Inform. 80 (2) (2011) 94–101.

[18] S.B. Trinidad, S.M. Fullerton, J.M. Bares, G.P. Jarvik, E.B. Larson, W. Burke, Genomic research and wide data sharing: views of prospective participants, Genet. Med. 12 (8) (2010) 486–495.

[19] R. Basavegowda, S. Seenappa, Electronic medical report security using visual secret sharing scheme, 2013 UKSim 15th International Conference on Computer Modelling and Simulation, IEEE, 2013, pp. 78–83.

[20] B. Middleton, J. Anderson, J. Fletcher, F.E. Masarie Jr., M.K. Leavitt, Use of the WWW for distributed knowledge engineering for an EMR: the KnowledgeBank concept, Proc. AMIA Symp. (1998) 126–130.

[21] S.C. Russell, S.A. Spooner, Barriers to EMR adoption in internal medicine and pediatric outpatient practices, Tennessee Med.: J. Tennessee Med. Assoc. 97 (10) (2004) 457–460.

[22] K. Caine, R. Hanania, Patients want granular privacy control over health information in electronic medical records, J. Am. Med. Inform. Assoc. 20 (1) (2013) 7–15.

[23] G. Loukides, J.C. Denny, B. Malin, The disclosure of diagnosis codes can breach research participants' privacy, J. Am. Med. Inform. Assoc. 17 (3) (2010) 322–327.

[24] M.J. Rantz, L. Hicks, G.F. Petroski, R.W. Madsen, G. Alexander, C. Galambos, V. Conn, J. Scott-Cawiezell, M. Zwygart-Stauffacher, L. Greenwald, Cost, staffing and quality impact of bedside electronic medical record (EMR) in nursing homes, J. Am. Med. Dir. Assoc. 11 (7) (2010) 485–493.

[25] .B. McMahan, E. Moore, D. Ramage, S. Hampson, Communication-efficient learning of deep networks from decentralized data, arXiv preprint arXiv:1602.05629; 2016.

[26] D. Liu, T. Miller, R. Sayeed, K. Mandl, FADL: Federated-Autonomous Deep Learning for Distributed Electronic Health Record, arXiv preprint arXiv:1811.11400; 2018.

[27] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated Learning with Non-IID Data, arXiv preprint arXiv:1806.00582; 2018.

[28] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, S.-L. Kim, Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data, arXiv preprint arXiv:1811.11479; 2018.

[29] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, D. Liu, LoAdaBoost: Loss-Based AdaBoost Federated Machine Learning on medical Data, arXiv preprint arXiv:1811.12629; 2018.

[30] R.E. Rosales, R.B. Rao, Guest editorial: special issue on impacting patient care by mining medical data, Data Mining Knowled. Discov. 20 (3) (2010) 325–327.

[31] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, Int. Conf. Machine Learn. (2016) 478–487.

[32] S. Greenfield, S.H. Kaplan, R. Kahn, J. Ninomiya, J.L. Griffith, Profiling care provided by different groups of physicians: effects of patient case-mix (bias) and physician-level clustering on quality assessment results, Ann. Intern. Med. 136 (2) (2002) 111–121.

[33] D. Walsh, L. Rybicki, Symptom clustering in advanced cancer, Support. Care Cancer 14 (8) (2006) 831–836.

[34] S.H. Sanders, S.F. Brena, Empirically derived chronic pain patient subgroups: the utility of multidimensional clustering to identify differential treatment effects, Pain 54 (1) (1993) 51–56.

[35] T.J. Pollard, A.E.W. Johnson, J.D. Raffa, L.A. Celi, R.G. Mark, O. Badawi, The eICU collaborative research database, a freely available multi-center database for critical care research, Sci. Data 5 (2018) 180178.

[36] K. Polat, B. Akdemir, S. Güneş, Computer aided diagnosis of ECG data on the least square support vector machine, Digital Signal Process. 18 (1) (2008) 25–32.

[37] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, N. Elhadad, Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1721–1730.

[38] E.K. Hashi, M.S.U. Zaman, M.R. Hasan, An expert clinical decision support system to predict disease using classification techniques, 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, 2017, pp. 396–400.

[39] J. Cogley, N. Stokes, J. Carthy, J. Dunnion, Analyzing patient records to establish if and when a patient suffered from a medical condition, Proceedings of the 2012 Workshop on Biomedical Natural Language Processing, Association for Computational Linguistics, 2012, pp. 38–46.

[40] H. Wang, C. Schmid, Action recognition with improved trajectories, Proceedings of the IEEE international conference on computer vision, 2013, pp. 3551–3558.

[41] M. Bansal, D. Klein, Coreference semantics from web features, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 389–398.

[42] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Dynamic integration with random forests, European conference on machine learning, Springer, 2006, pp. 801–808.

[43] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, Pacific-Asia conference on knowledge discovery and data mining, Springer, 2004, pp. 22–30.

[44] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. (2010) 3371–3408.

[45] W.J.J.C. Youden, Index Rating Diagnostic Tests 3 (1) (1950) 32–35.

[46] D. Berrar, P. Flach, Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them), Briefings Bioinf. 13 (1) (2011) 83–97.

[47] S. Halligan, D.G. Altman, S. Mallett, Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: a discussion and proposal for an alternative approach, Eur. Radiol. 25 (4) (2015) 932–939.