Federated Learning using PHE

Training a Machine Learning Model on Shared Data

Presenters -
Vibhav Agarwal (IMT2016003)
Aayush Grover (IMT2016005)

Content

- Motivation
- Related Work
- Approach
- Results

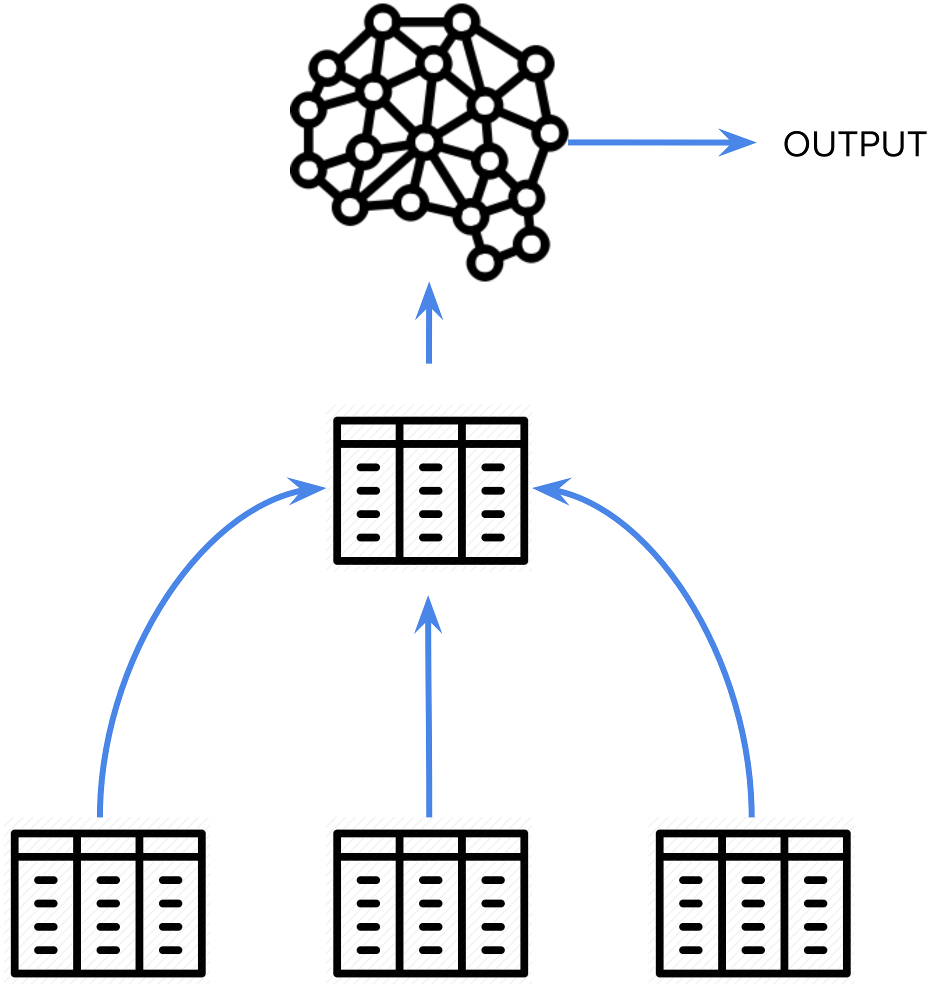
Motivation

Motivation

Constraints -

1. Data should not leave a hospital, even in the encrypted format.
2. Origin of data should not be inferred at any time.

Each party is “honest but curious”



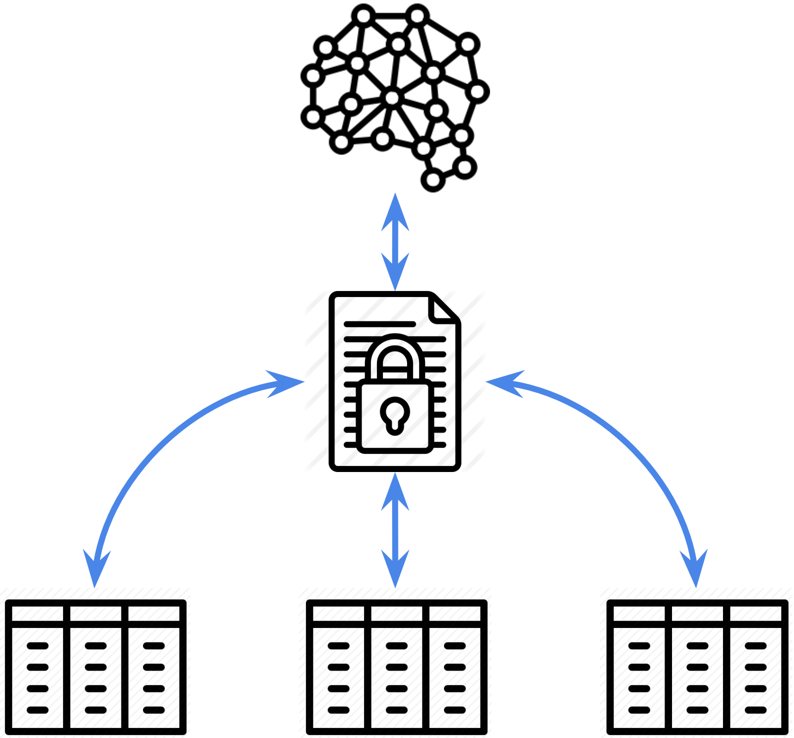
Related Work

Secure Model Fusion for Distributed Learning Using Partial Homomorphic Encryption - 2019

- Each client computes individual weights
- Sends encrypted weights to server
- Server sums them up and sends to one of the clients
- Client decrypts it and sends it to all clients

Privacy-Preserving Ridge Regression with only Linearly-Homomorphic Encryption - 2018

Takes about 40s to train on student dataset



Scalable and Secure Logistic Regression via Homomorphic Encryption - 2016

- Encrypted data is sent to the server by each client
- The model is computed by server by using mathematical approximations
- Outputs the model weights
- Use R-LWE

Approach

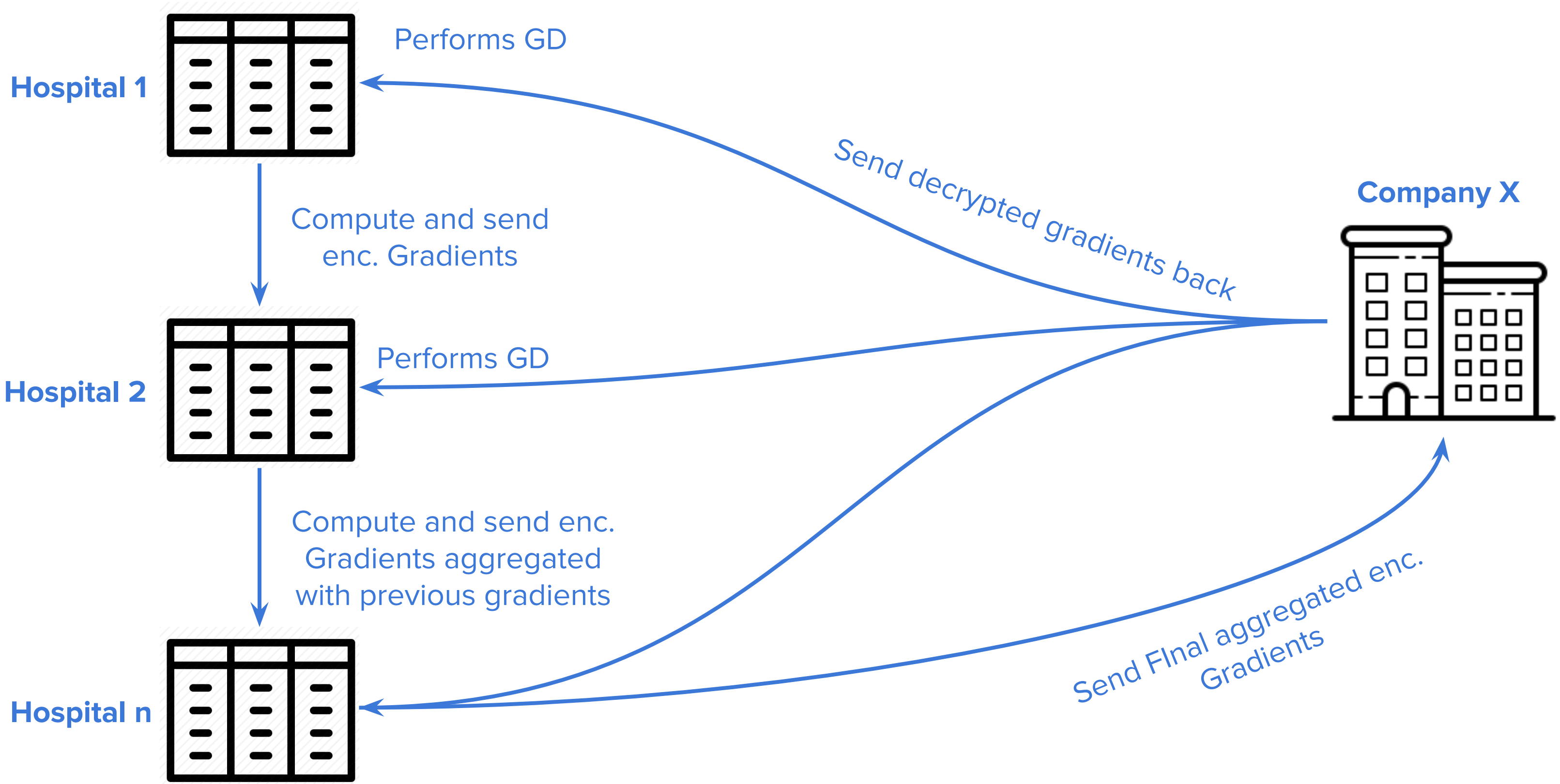
Approach: Data

- Use the breast_cancer dataset from scikit-learn
- The hospitals will have different patient records but with the same feature set.
- Data split into **test** and **train** set in 1:3 ratio.
- This train set is further split into **n** shares for **n** hospitals.
    - Represents the idea of horizontal split of the dataset.
- The **test** set is common for all parties and for all experiments.

Approach: Model

- Use Logistic Regression Classification model for classifying breast cancer into benign or malignant.
- Federated learning on the complete data of all the **n** hospitals.
- Our approach is to not share data in its true or encrypted format
  - Instead share the derived information i.e. gradients in the encrypted format.
- The company X creates the public-private key pair using Paillier scheme (PHE).
  - Public key is shared with the hospitals for the gradient encryption
  - Private key is kept with the company itself.

Approach: Training



Approach: Training

Algorithm 1 Pseudocode

for hospital h_i in n **do**
  $g_i \leftarrow$ compute gradient for h_i on data X_i
  $ag_i \leftarrow g_i + ag_{i-1}$ where aggregated gradient ag_{i-1} from h_{i-1}
end for
$d_{ag} \leftarrow$ Company decrypts ag
for hospital h_i in n **do**
  h_i performs gradient descent using d_{ag}
end for
$\therefore$ Model is trained on the whole data

Approach: Security

- We consider all parties to be **honest but curious**.
- No hospital will be able to point out where patients’ data originated.
    - True if the protocol is run by at least 3 hospitals preventing the reconstruction of each others’ gradients.
- The company X cannot learn anything about the underlying data from the total aggregated gradient, thereby preserving patients' privacy.
- **Centralized aggregation vs coupled:** O(1) vs O(logn) aggregation. Security issue in O(1) since Server can pinpoint patients’ data origins.

Results

Results

- 2 different datasets
- Test data is common in all scenarios (for a given dataset)
- Key length = 1024

Results - breast cancer

Number of Hospitals (n)	Avg Acc (LL) (in %)	Train Time (LL) (in s)	Avg Acc (FL) (in %)	Train Time (FL) (in s)	Avg Acc (sklearn) (in %)	Train Time (sklearn) (in s)
3	**96.27**	0.003	95.80	2.821	95.80	0.123
7	**96.00**	0.006	95.80	6.315	95.80	0.028
10	95.38	0.008	**95.80**	9.987	**95.80**	0.033
20	94.93	0.161	**95.80**	18.343	**95.80**	0.036
27	94.17	0.209	**95.80**	24.681	**95.80**	0.038

Table 1: Comparison of our model's performance on breast cancer data with the one without combining data and also, against scikit-learn's Logistic Regression. The best performances are indicated in bold. LL-Local Learning ; FL-Federated Learning

Results - grad admission

Number of Data Owners (n)	Avg Acc (LL) (in %)	Train Time (LL) (in s)	Avg Acc (FL) (in %)	Train Time (FL) (in s)	Avg Acc (sklearn) (in %)	Train Time (sklearn) (in s)
3	70.40	0.003	**86.40**	0.777	**86.40**	0.024
7	80.46	0.006	**86.40**	1.709	**86.40**	0.029
10	79.28	0.014	**86.40**	2.594	**86.40**	0.027
20	75.24	0.017	**86.40**	5.143	**86.40**	0.030
27	77.16	0.020	**86.40**	6.802	**86.40**	0.024

Table 2: Comparison of our model's performance on grad-admission data with the one without combining data and also, against scikit-learn's Logistic Regression.

Thank you!