federated_models_linear_regression

June 18, 2020

1 Federated learning: linear regression

We show how to set up a Linear Regression experiment in the Federated setting using *Sherpa.FL*. Results from the Federated Learning are compared to the (non-federated) centralized learning. Moreover, we also show how the addition of Differential Privacy affects the performance of the Federated model. Ultimately, an application of the composition theorems for adaptive differential privacy is given.

Dataset. In the present example, we will use the California Housing dataset from sklearn. We only make use of two features in order to reduce the variance in the prediction. *Sherpa.FL* allows to easily convert a generic dataset to interact with the platform:

```
[1]: import shfl
     from shfl.data_base.data_base import LabeledDatabase
     import sklearn.datasets
     import numpy as np
     from shfl.private.reproducibility import Reproducibility

     # Comment to turn off reproducibility:
     Reproducibility(1234)

     all_data = sklearn.datasets.fetch_california_housing()
     n_features = 2
     data = all_data["data"][:,0:n_features]
     labels = all_data["target"]

     # Retain part for DP sensitivity sampling:
     size = 2000
     sampling_data = data[-size:, ]
     sampling_labels = labels[-size:, ]

     # Create database:
     database = LabeledDatabase(data[0:-size, ], labels[0:-size])
     train_data, train_labels, test_data, test_labels = database.load_data()
```

```
[2]: print("Shape of train and test data: " + str(train_data.shape) + str(test_data.
     ↪shape))
     print("Total: " + str(train_data.shape[0] + test_data.shape[0]))
```

```
Shape of train and test data: (14912, 2)(3728, 2)
Total: 18640
```

We will simulate a FL scenario by distributing the train data over a collection of clients, assuming an IID setting:

```
[3]: iid_distribution = shfl.data_distribution.IidDataDistribution(database)
     federated_data, test_data, test_labels = iid_distribution.
       ↪get_federated_data(num_nodes=5)
```

At this stage, we need to define the linear regression model, and we choose the federated aggregator to be the average of the clients' models:

```
[4]: from shfl.model.linear_regression_model import LinearRegressionModel

     def model_builder():
         model = LinearRegressionModel(n_features=n_features, n_targets=1)
         return model

     aggregator = shfl.federated_aggregator.FedAvgAggregator()
```

Running the model in a Federated configuration We're now ready to run the FL model. The *Sherpa.FL* framework offers support for the Linear Regression model from scikit-learn. The user must specify in advance the number of features and targets. Note that in this case, we set the number of rounds `n=1` since no iterations are needed in the case of linear regression. The performance metrics used are the Root Mean Squared Error (RMSE) and the R^2 score. It can be observed that the performance of the *Global model* (i.e. the aggregated model) is in general superior with respect to the performance of each node, thus the federated learning approach proves to be beneficial:

```
[5]: federated_government = shfl.federated_government.
       ↪FederatedGovernment(model_builder, federated_data, aggregator)
     federated_government.run_rounds(n=1, test_data=test_data,␣
       ↪test_label=test_labels)
```

```
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8154147770321544, 0.5019079411109164)
```

And we can observe that the performance is comparable with the centralized learning model:

```
[6]:  # Comparison with centralized model:
      centralized_model = LinearRegressionModel(n_features=n_features, n_targets=1)
      centralized_model.train(data=train_data, labels=train_labels)
      print(centralized_model.evaluate(data=test_data, labels=test_labels))
```

```
(0.815401474919891, 0.5019241920365265)
```

Differential Privacy: sampling the model's sensitivity. We want to assess the impact of Differential Privacy (see The Algorithmic Foundations of Differential Privacy, Section 3.3) on the Federated model's performance. In the case of applying the Laplace privacy mechanism (see Laplace mechanism notebook), the noise added has to be of the order of the sensitivity of the model's output, i.e. the model parameters of our linear regression. In the general case, the model's sensitivity might be difficult to compute analytically. An alternative approach is to attain *random* differential privacy through a sampling over the data (e.g. see Rubinstein 2017. That is, instead of computing analytically the *global* sensitivity Δf, we compute an *empirical estimation* of it by sampling over the dataset. However, be advised that this would guarantee the *weaker* property of *random* differential privacy. This approach is convenient since allows for the sensitivity estimation of an arbitrary model or a black-box computer function. The *Sherpa.FL* framework provides this functionality in the class `SensitivitySampler`.

We need to specify a distribution of the data to sample from. This in general requires previous knowledge and/or model assumptions. In order not make any specific assumption on the distribution of the dataset, we can choose a *uniform* distribution. To the end, we define our class of `ProbabilityDistribution` that uniformly samples over a data-frame. We use the previously retained part of the dataset for sampling:

```
[7]:  class UniformDistribution(shfl.differential_privacy.ProbabilityDistribution):
          """
          Implement Uniform sampling over the data
          """
          def __init__(self, sample_data):
              self._sample_data = sample_data

          def sample(self, sample_size):
              row_indices = np.random.randint(low=0, high=self._sample_data.shape[0],␣
      ↪size=sample_size, dtype='l')

              return self._sample_data[row_indices, :]

      sample_data = np.hstack((sampling_data, sampling_labels.reshape(-1,1)))
```

The class `SensitivitySampler` implements the sampling given a *query*, i.e. the learning model itself in this case. We only need to add the method `get` to our model since it is required by the class `SensitivitySampler`. We choose the sensitivity norm to be the L_1 norm and we apply the sampling. The value of the sensitivity depends on the number of samples `n`: the more samples

we perform, the more accurate the sensitivity. Indeed, increasing the number of samples `n`, the sensitivity gets more accurate and typically decreases.

```
[8]: from shfl.differential_privacy import SensitivitySampler
     from shfl.differential_privacy import L1SensitivityNorm


     class LinearRegressionSample(LinearRegressionModel):

         def get(self, data_array):
             data = data_array[:, 0:-1]
             labels = data_array[:, -1]
             train_model = self.train(data, labels)

             return self.get_model_params()


     distribution = UniformDistribution(sample_data)
     sampler = SensitivitySampler()
     n_samples = 4000
     max_sensitivity, mean_sensitivity = sampler.sample_sensitivity(
         LinearRegressionSample(n_features=n_features, n_targets=1),
         L1SensitivityNorm(), distribution, n=n_samples, gamma=0.05)
     print("Max sensitivity from sampling: " + str(max_sensitivity))
     print("Mean sensitivity from sampling: " + str(mean_sensitivity))
```

```
Max sensitivity from sampling: 0.00829435406405371
Mean sensitivity from sampling: 0.0006633612087443461
```

Unfortunately, sampling over a dataset involves, at each sample, the training of the model on two datasets differing in one entry. Thus in general this procedure might be computationally expensive (e.g. in the case of training a deep neuronal network).

Running the model in a Federated configuration with Differential Privacy. At this stage we are ready to add a layer of DP to our federated learning model. We will apply the Laplace mechanism, employing the sensitivity obtained from the previous sampling. The Laplace mechanism provided by the *Sherpa.FL* Framework is then assigned as the *private* access type to the model's parameters of each client in a new `FederatedGovernment` object. This results into an ϵ-*differentially private FL model*. For example, picking the value $\epsilon = 0.5$, we can run the FL experiment with DP:

```
[9]: from shfl.differential_privacy import LaplaceMechanism

     params_access_definition = LaplaceMechanism(sensitivity=max_sensitivity,␣
      ↪epsilon=0.5)
     federated_governmentDP = shfl.federated_government.FederatedGovernment(
         model_builder, federated_data, aggregator,␣
      ↪model_params_access=params_access_definition)
     federated_governmentDP.run_rounds(n=1, test_data=test_data,␣
      ↪test_label=test_labels)
```

```
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8309024800913739, 0.4828070773551624)
```

In the above example we observed that the performance of the model has slightly deteriorated due to the addition of Differential Privacy. It must be noted that each run involves a different random noise added by the Differential Privacy mechanism. However, in general, the privacy increases at expenses of accuracy (i.e. for smaller values of ϵ). This can be observed by performing a mean on several runs, as explained in below.

1.1 Multiple queries: Composition of differential-private mechanisms using adaptivity

We here want to show the application of the composition theorems using adaptivity, as implemented in *Sherpa.FL* (see Composition concepts notebook). The idea is that we stop when the privacy budget is expended. This happens when the same query is executed on the clients' dataset, as this might disclose sensible information (see The Algorithmic Foundations of Differential Privacy, Section 3.5.2). Note that, when applying the composition theorems for privacy filters in the present example, we are assuming that the *estimated* sensitivity is a good enough approximation of the analytic sensitivity (see Rogers 2016).

In the below experiment, we fix a privacy budget (variable `global_epsilon_delta = (4, 0)`), and we consider different values of ϵ for the query (variable `epsilon_range = np.array([0.2,0.5,0.8])`). In each case, the execution automatically exits when the privacy budget is expended. Taking the average of the performance metrics, we can verify that the accuracy increases for larger values of ϵ, which is associated to lower privacy.

```
[10]:  # Run several runs with different levels of privacy: for fixed sensitivity, we
       ↪use different values of epsilon
       from shfl.differential_privacy.composition_dp import AdaptiveDifferentialPrivacy
       from shfl.differential_privacy.composition_dp import ExceededPrivacyBudgetError

       global_epsilon_delta = (4, 0)
       epsilon_range = np.array([0.2,0.5,0.8])
       gl_evaluationDP = np.zeros((epsilon_range.size, 2))

       for i_epsilon in range(epsilon_range.size):
```

```python
    print("---------------------------\n")
    print("epsilon = " + str(epsilon_range[i_epsilon]))

    dpm = LaplaceMechanism(sensitivity=max_sensitivity,
→epsilon=epsilon_range[i_epsilon])

    params_access_definition =
→AdaptiveDifferentialPrivacy(global_epsilon_delta,
→differentially_private_mechanism=dpm)
    federated_governmentDP = shfl.federated_government.FederatedGovernment(
        model_builder, federated_data, aggregator,
→model_params_access=params_access_definition)
    i_run = 0
    while True:
        try:
            # Queries are performed using the Laplace mechanism
            #print("i_run = " + str(i_run))
            federated_governmentDP.run_rounds(n=1, test_data=test_data,
→test_label=test_labels)
            print("Executed i_run = " + str(i_run))
            gl_evaluationDP[i_epsilon,:] += np.asarray(federated_governmentDP.
→_model.evaluate(data=test_data, labels=test_labels))
            print("\n")
            i_run += 1
        except ExceededPrivacyBudgetError:
            # At this point we have spent all our privacy budget
            print("Reached privacy budget at i_run = " + str(i_run))
            print("\n")
            gl_evaluationDP[i_epsilon,:] = np.divide(gl_evaluationDP[i_epsilon,:
→], i_run)
            break

print("Mean performance: \n" + str(gl_evaluationDP))
```

----

epsilon = 0.2
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode

object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8205501623431068, 0.4956143360079521)

Executed i_run = 0

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (1.074990065452661, 0.13431206623582514)

Executed i_run = 1

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (1.0838889051860943, 0.11992029841606633)

Executed i_run = 2

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode

object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.9006358505433762, 0.39235360792923857)

Executed i_run = 3

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.838060854345674, 0.47385727058244487)

Executed i_run = 4

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8683702560612934, 0.4351120099343775)

Executed i_run = 5

```
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8482794380318678, 0.460948395873121)



Executed i_run = 6


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (1.0813012878886836, 0.1241173910992629)



Executed i_run = 7


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
```

object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (1.0985162095688272, 0.09600630351063222)

Executed i_run = 8

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.9195076683374811, 0.36662170933854576)

Executed i_run = 9

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8683597755197254, 0.43512564535381504)

Executed i_run = 10

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode

object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (2.3822838037200094, -3.251473330111086)

Executed i_run = 11

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.9278091790950563, 0.355133535761471)

Executed i_run = 12

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.914542297913412, 0.3734437653233611)

Executed i_run = 13

```
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (2.002653017758975, -2.0044423947839456)




Executed i_run = 14


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.823579565862492, 0.4918831605645322)




Executed i_run = 15


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
```

object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8185593584459361, 0.4980588296831059)

Executed i_run = 16

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8275029974085175, 0.4870304171000157)

Executed i_run = 17

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.9534500621570825, 0.31899803419917483)

Executed i_run = 18

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode

```
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Reached privacy budget at i_run = 19


---------------------------

epsilon = 0.5
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8336216603409031, 0.47941644612415)



Executed i_run = 0


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8920666047172546, 0.40386169867150734)



Executed i_run = 1
```

```
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8493774573182328, 0.45955198779354733)


Executed i_run = 2


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8811876762168931, 0.4183130880290564)


Executed i_run = 3


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
```

Global model test performance : (0.8312531693273225, 0.4823704141596118)

Executed i_run = 4

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8384094969464859, 0.47341941711562974)

Executed i_run = 5

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8160208309802515, 0.5011672558807143)

Executed i_run = 6

Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)

```
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8182112409832832, 0.49848567064267824)




Executed i_run = 7


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Reached privacy budget at i_run = 8



--------------------------


epsilon = 0.8
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.816288579331826, 0.5008398535604772)




Executed i_run = 0
```

```
Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8170072801337276, 0.49996049609838666)




Executed i_run = 1


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8156196554752249, 0.5016576117115246)




Executed i_run = 2


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.8299556584011977, 0.4839850988098158)
```

```
Executed i_run = 3


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Global model test performance : (0.829710651570204, 0.48428971391414544)



Executed i_run = 4


Accuracy round 0
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc095d0>: (0.8161535463006577, 0.5010049851923566)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09a50>: (0.81637303674763, 0.5007365568636023)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b10>: (0.8155342443231007, 0.5017619784187598)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09b90>: (0.8158502097728686, 0.5013758352304258)
Test performance client <shfl.private.federated_operation.FederatedDataNode
object at 0x10fc09d90>: (0.8151607067608613, 0.5022182878756591)
Reached privacy budget at i_run = 5


Mean performance:
[[1.05541267 0.04224321]
 [0.84501852 0.46457325]
 [0.82171636 0.49414655]]
```