

07-03-2024

## PROJECT DISCUSSION -2

Discussion was mainly about the dataset and existing framework such as **flower, pysyft, Fate, tensorflow federated** and how this framework works and the advantages and disadvantages of this framework.

### Key Points Discussed:

#### Advantages and Disadvantages of the Frameworks

Out of all this frameworks Flower Frameworks have more advantage compared to other frameworks

#### Deciding The Dataset:

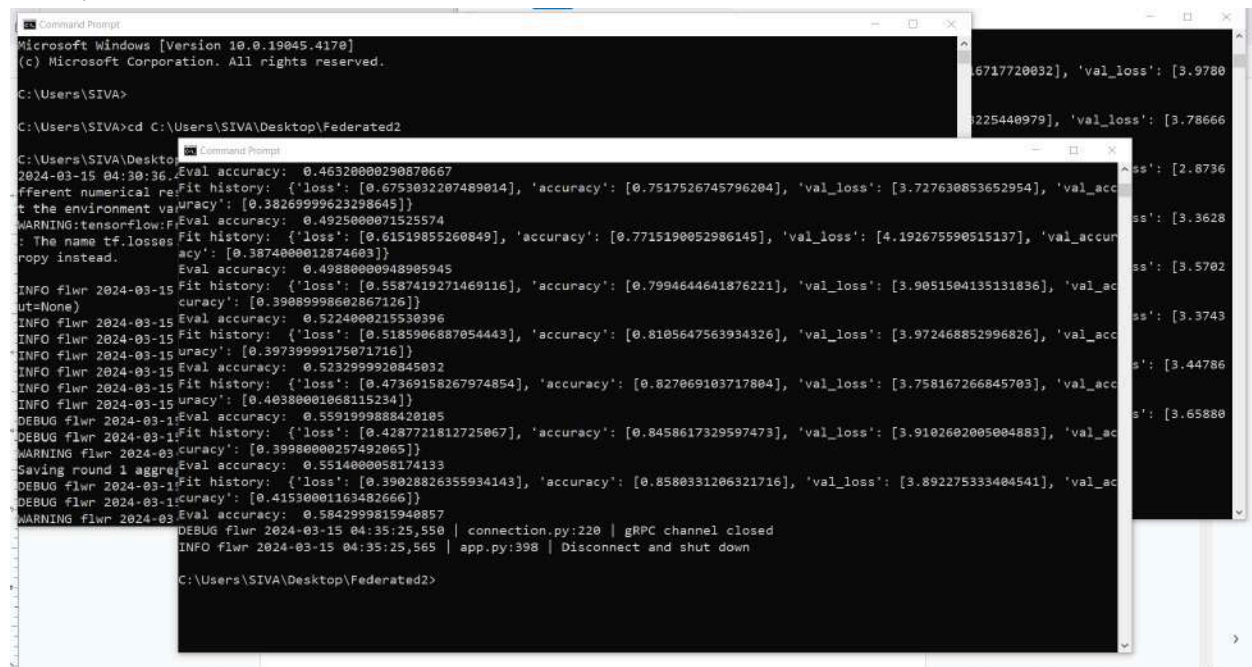
- Titanic Dataset(size: 994)
- Cancer Dataset(size:596)
- Mnist Dataset
- California House Prediction(Recommended by sir)

**Showed Some Existing Model**(based on class imbalance and horizontal federated learning)

```
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated
Saving round 9 aggregated_weights...
INFO flwr 2024-03-15 01:57:49,318 server.py:173 | evaluate_round 9: strategy sampled 1
ts (out of 2)
DEBUG flwr 2024-03-15 01:57:49,753 server.py:187 | evaluate_round 9 received 2 results
failures
DEBUG flwr 2024-03-15 01:57:49,753 server.py:222 | fit_round 10: strategy sampled 2
out of 2)
DEBUG flwr 2024-03-15 01:57:51,925 server.py:236 | fit_round 10 received 2 results
ures
Saving round 10 aggregated_weights.
DEBUG flwr 2024-03-15 01:57:51,929 nts (out of 2)
DEBUG flwr 2024-03-15 01:57:52,393 failures
INFO flwr 2024-03-15 01:57:52,393 INFO flwr 2024-03-15 01:57:52,393 s3016] (2, 0.3783248669486084), Eval accuracy: [0.938899977684021]
82217466831207], (6, 0.154402911663 Fit history: {'accuracy': [0.9871470332145691], 'loss': [0.03676206991074694],
0.14974884688854218), (10, 0.15134 Val accuracy: [0.649999761581421], 'val_loss': [1.7135902643203735]}
INFO flwr 2024-03-15 01:57:52,394 Eval accuracy: [0.9534000158309937]
INFO flwr 2024-03-15 01:57:52,394 Fit history: {'accuracy': [0.9898734092712402], 'loss': [0.029444887636470795],
INFO flwr 2024-03-15 01:57:52,394 'val_accuracy': [0.6620000004768372], 'val_loss': [2.1011500358581543]}
INFO flwr 2024-03-15 01:57:52,394 Eval accuracy: [0.9473999738693237]
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated Fit history: {'accuracy': [0.9913339614868164], 'loss': [0.02617462910717653],
'val_accuracy': [0.640699826431274], 'val_loss': [1.861505389213562]},
INFO flwr 2024-03-15 01:57:52,393 Eval accuracy: [0.952099978237976]
Fit history: {'accuracy': [0.9929406046867371], 'loss': [0.02007871819138527],
'val_accuracy': [0.6643000245894299], 'val_loss': [1.7174336910247803]},
Eval accuracy: [0.9546999931335449]
Fit history: {'accuracy': [0.992112934589386], 'loss': [0.021396998316049576],
'val_accuracy': [0.6766999959945679], 'val_loss': [1.8145411014556885]},
Eval accuracy: [0.9527999758720398]
DEBUG flwr 2024-03-15 01:57:52,398 | connection.py:220 | gRPC channel closed
INFO flwr 2024-03-15 01:57:52,398 | app.py:398 | Disconnect and shut down
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated$
```

```
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated1
DEBUG flwr 2024-03-15 02:25:44,665 | server.py:187 | evaluate_round 9 received 2 results and 0 failures
DEBUG flwr 2024-03-15 02:25:44,666 | server.py:222 | fit_round 10: strategy sampled 2 clients (out of 2)
DEBUG flwr 2024-03-15 02:25:46,729 | server.py:236 | fit_round 10 received 2 results and 0 failures
Saving round 10 aggregated weights...
DEBUG flwr 2024-03-15 02:25:46,731 | server.py:173 | evaluate_round 10: strategy sampled 2 clients (out of 2)
DEBUG flwr 2024-03-15 02:25:47,179 | server.py:187 | evaluate_round 10 received 2 results and 0 failures
INFO flwr 2024-03-15 02:25:47,179 | server.py:153 | FL finished in 36.2562194300012
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated1
INFO flwr 2024-03-15 02:25:47,179 | app.py:11 | (1, 0.898810876713562), (2, 0.69540184736, 5.9351634979248), (5, 0.5392398834228516, 0.21377563), (8, 0.48931998014450873), (1615852)}
INFO flwr 2024-03-15 02:25:47,179 | app.py:11 | Fit history: {'accuracy': [0.956599960809353], 'val_loss': [0.76419997215271]}
INFO flwr 2024-03-15 02:25:47,179 | app.py:11 | Val accuracy: [0.9271665215492], 'val_loss': [0.629180024700164], 'val_loss': [0.786499977118164]
INFO flwr 2024-03-15 02:25:47,180 | app.py:11 | Fit history: {'accuracy': [0.9318403005599], 'val_loss': [0.587190028916992], 'val_loss': [0.7803999781608582]}
INFO flwr 2024-03-15 02:25:47,180 | app.py:11 | Fit history: {'accuracy': [0.935589075088501], 'val_loss': [0.5968000292778015], 'val_loss': [0.79838002784729]}
INFO flwr 2024-03-15 02:25:47,180 | app.py:11 | Fit history: {'accuracy': [0.9352483153343201], 'val_loss': [0.6484000086784363], 'val_loss': [0.810400091552734]}
INFO flwr 2024-03-15 02:25:47,180 | app.py:11 | Fit history: {'accuracy': [0.9387536644935608], 'val_loss': [0.579299863624573], 'val_loss': [0.7509999871253967]}
INFO flwr 2024-03-15 02:25:47,180 | app.py:11 | Fit history: {'accuracy': [0.9414800485502319], 'val_loss': [0.5788000226028813], 'val_loss': [0.8256000280380249]}
DEBUG flwr 2024-03-15 02:25:47,183 | connection.py:220 | gRPC channel closed
INFO flwr 2024-03-15 02:25:47,183 | app.py:398 | Disconnect and shut down
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated1
Fit history: {'accuracy': [0.9441995024681891], 'loss': [0.17990176379680634], 'val_accuracy': [0.7114999890327454], 'val_loss': [1.9748142957687378]}
Eval accuracy: 0.76419997215271
Fit history: {'accuracy': [0.9476640224456787], 'loss': [0.16356860181222992], 'val_accuracy': [0.7278000116348267], 'val_loss': [1.5895325708389282]}
Eval accuracy: 0.786499977118164
Fit history: {'accuracy': [0.9518635272979736], 'loss': [0.15540309250354767], 'val_accuracy': [0.6815997474332632], 'val_loss': [1.926389455795288]}
Eval accuracy: 0.7803999781608582
Fit history: {'accuracy': [0.9536482691764832], 'loss': [0.1473647934951782], 'val_accuracy': [0.6883999705314636], 'val_loss': [1.9580696821227609]}
Eval accuracy: 0.79838002784729
Fit history: {'accuracy': [0.956179959297918], 'loss': [0.1377612054374992], 'val_accuracy': [0.660499989864197], 'val_loss': [1.83520853194397]}
Eval accuracy: 0.810400091552734
Fit history: {'accuracy': [0.9566929340362549], 'loss': [0.1377862244844366], 'val_accuracy': [0.596499979460022], 'val_loss': [2.0366592407725662]}
Eval accuracy: 0.7509999871253967
Fit history: {'accuracy': [0.9591600894927979], 'loss': [0.12615329027175903], 'val_accuracy': [0.7106800185012817], 'val_loss': [1.9507231712341389]}
Eval accuracy: 0.8256000280380249
DEBUG flwr 2024-03-15 02:25:47,183 | connection.py:220 | gRPC channel closed
INFO flwr 2024-03-15 02:25:47,184 | app.py:398 | Disconnect and shut down
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated1
Fit history: {'accuracy': [0.935589075088501], 'loss': [0.1052431002329000], 'val_accuracy': [0.5968000292778015], 'val_loss': [3.11248779296875]}
Eval accuracy: 0.79838002784729
Fit history: {'accuracy': [0.9352483153343201], 'loss': [0.17073878645896912], 'val_accuracy': [0.6484000086784363], 'val_loss': [2.4795634746551514]}
Eval accuracy: 0.810400091552734
Fit history: {'accuracy': [0.9387536644935608], 'loss': [0.1595805306596756], 'val_accuracy': [0.579299863624573], 'val_loss': [3.3580050441741943]}
Eval accuracy: 0.7509999871253967
Fit history: {'accuracy': [0.9414800485502319], 'loss': [0.15539324283599854], 'val_accuracy': [0.5788000226028813], 'val_loss': [3.385594367980957]}
Eval accuracy: 0.8256000280380249
DEBUG flwr 2024-03-15 02:25:47,183 | connection.py:220 | gRPC channel closed
INFO flwr 2024-03-15 02:25:47,183 | app.py:398 | Disconnect and shut down
siva@siva-Swift-SF314-SSG: ~/Desktop/Federated1
```

## Horizontal Federated Learning - CIFAR10 dataset (with different class distributions to each client)



```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SIVA>
C:\Users\SIVA>cd C:\Users\SIVA\Desktop\Federated2
C:\Users\SIVA\Desktop\Federated2>
2024-03-15 04:30:36, Eval accuracy: 0.46320000290870667
Different numerical representation of the environment variable
WARNING:tensorflow:From C:\Users\SIVA\Desktop\Federated2\app.py:398: tf.nn.conv2d (from tensorflow.nn.conv2d) is deprecated and will be removed in a future version. Use tf.nn.conv2d_v2 instead.
INFO flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.6753032207489014], 'accuracy': [0.7517526745796204], 'val_loss': [3.727630853652954], 'val_accuracy': [0.38269999623298645]}
INFO flwr 2024-03-15 04:30:36, Eval accuracy: 0.4925000071525574
INFO flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.61519855260849], 'accuracy': [0.7715190052986145], 'val_loss': [4.192675590515137], 'val_accuracy': [0.3874000012874603]}
INFO flwr 2024-03-15 04:30:36, Eval accuracy: 0.49880000948905945
INFO flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.5587419271469116], 'accuracy': [0.7994644641876221], 'val_loss': [3.9051504135131836], 'val_accuracy': [0.39089999602867126]}
INFO flwr 2024-03-15 04:30:36, Eval accuracy: 0.5224000215530396
INFO flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.5185906887054443], 'accuracy': [0.8105647563934326], 'val_loss': [3.972468852996826], 'val_accuracy': [0.39739999175071716]}
INFO flwr 2024-03-15 04:30:36, Eval accuracy: 0.5232999920845032
INFO flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.47369158267974854], 'accuracy': [0.827069103717804], 'val_loss': [3.758167266845703], 'val_accuracy': [0.40380001068115234]}
INFO flwr 2024-03-15 04:30:36, Eval accuracy: 0.5591999888420105
DEBUG flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.4287721812725067], 'accuracy': [0.8458617329597473], 'val_loss': [3.9102602005004883], 'val_accuracy': [0.39980000257492065]}
WARNING flwr 2024-03-15 04:30:36, Eval accuracy: 0.5514000058174133
Saving round 1 aggregated model
DEBUG flwr 2024-03-15 04:30:36, Fit history: {'loss': [0.39028826355934143], 'accuracy': [0.8580331206321716], 'val_loss': [3.892275333404541], 'val_accuracy': [0.41530001163482666]}
DEBUG flwr 2024-03-15 04:30:36, Eval accuracy: 0.5842999815940857
DEBUG flwr 2024-03-15 04:35:25,550 | connection.py:220 | gRPC channel closed
INFO flwr 2024-03-15 04:35:25,565 | app.py:398 | Disconnect and shut down
C:\Users\SIVA\Desktop\Federated2>
```

## BASIC MODEL OF FEDERATED LEARNING

### Comparing Performance Of Centralized And Decentralized Model

In this model, there is no server, client method instead the dataset is partitioned into three datasets each with different features and the datasets are modeled using a logistic regression for each dataset and the predicted outputs are sent to the FNN model which finally predicts the label for the dataset and this predicted output are compared with the actual output for calculating the accuracy for each dataset and for each model (centralized model, decentralized model)

FOR TITANIC DATASET:

```
Running centralised training...
Train accuracy: 84.248%
Test accuracy: 82.022%
Running decentralised training...
Iteration 1, loss = 0.64399825
Iteration 2, loss = 0.56226789
Iteration 3, loss = 0.49493030
Iteration 4, loss = 0.45459877
Iteration 5, loss = 0.43634872
Iteration 6, loss = 0.43703215
Iteration 7, loss = 0.43900626
Iteration 8, loss = 0.42954290
Iteration 9, loss = 0.42684795
Iteration 10, loss = 0.42327625
Iteration 11, loss = 0.42247988
Iteration 12, loss = 0.42048271
Iteration 13, loss = 0.41931750
Iteration 14, loss = 0.41798792
Iteration 15, loss = 0.41693441
```

```
Client 0 test accuracy: 73.034%
Client 1 test accuracy: 81.461%
Client 2 test accuracy: 71.348%
Combined test accuracy: 80.337%
```

**For Cancer Dataset :**

(since size of the dataset is very small we getting error related to not sufficient data in train and test set)

CENTRALIZED MODEL:

```
Running centralised training...
Train accuracy: 98.758%
Test accuracy: 95.349%
```

DECENTRALIZED MODEL:

```
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs.
Stopping.
Client 0 test accuracy: 96.512%
Client 1 test accuracy: 94.186%
Client 2 test accuracy: 94.186%
Combined Test accuracy: 96.512%
```

**FLOWER FRAMEWORK**

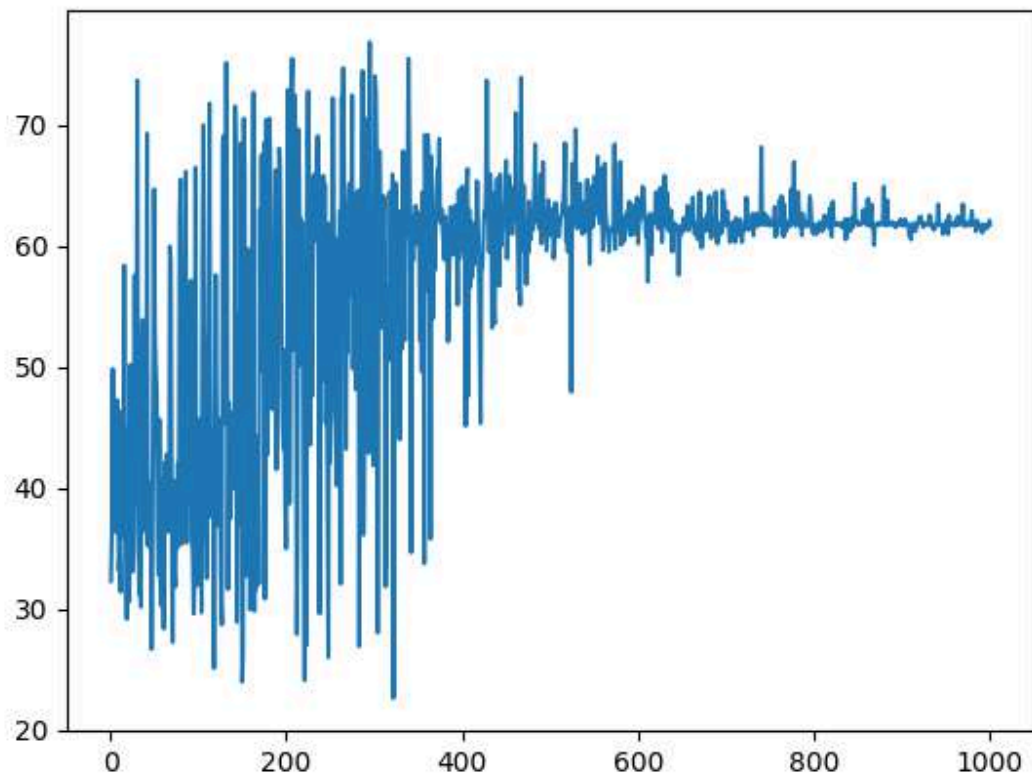
Vertical Federated Learning - Titanic data set (Features of the data set has been distributed to three clients)



```
siva@siva-Swift-SF314-S5G: ~/vertical-fl
siva@siva-Swift-SF314-S5G:~/vertical-fl$ poetry run python3 -c "import flwr"
siva@siva-Swift-SF314-S5G:~/vertical-fl$ poetry run python3 simulation.py
Command 'poetry' not found, did you mean:
  command 'poetry' from deb python3-poetry (1.1.12+dfsg-1ubuntu1)
Try: sudo apt install <deb name>
siva@siva-Swift-SF314-S5G:~/vertical-fl$ poetry shell
Spawning shell within /home/siva/.cache/pypoetry/virtualenvs/vertical-fl-XMIFJJjx-py3.10
siva@siva-Swift-SF314-S5G:~/vertical-fl$ ./home/siva/.cache/pypoetry/virtualenvs/vertical-fl-XMIFJJjx-py3.10/bin/activate
(vertical-fl-py3.10) siva@siva-Swift-SF314-S5G:~/vertical-fl$ poetry run python3 -c "import flwr"
(vertical-fl-py3.10) siva@siva-Swift-SF314-S5G:~/vertical-fl$ poetry run python3 simulation.py
INFO flwr 2024-03-15 02:51:27,944 | app.py:178 | Starting Flower simulation, config: ServerConfig(num_rounds=1000, round_timeout=None)
INFO flwr 2024-03-15 02:51:27,944 | app.py:213 | Flower VCE: Ray initialized with resources: {'CPU': 8.0, 'node:__internal_head__': 1.0, 'object_store_memory': 103458324
4.0, 'memory': 2069166491.0, 'node:10.32.2.114': 1.0}
INFO flwr 2024-03-15 02:51:27,944 | app.py:219 | Optimize your simulation with Flower VCE: https://flower.dev/docs/framework/how-to-run-simulations.html
INFO flwr 2024-03-15 02:51:27,944 | app.py:227 | No 'client_resources' specified. Using minimal resources for clients.
INFO flwr 2024-03-15 02:51:27,944 | app.py:242 | Flower VCE: Resources for each Virtual Client: {'num_cpus': 1, 'num_gpus': 0.0}
INFO flwr 2024-03-15 02:51:27,956 | app.py:288 | Flower VCE: Creating VirtualClientEngineActorPool with 8 actors
INFO flwr 2024-03-15 02:51:27,956 | server.py:89 | Initializing global parameters
INFO flwr 2024-03-15 02:51:27,956 | server.py:272 | Using initial parameters provided by strategy
INFO flwr 2024-03-15 02:51:27,956 | server.py:91 | Evaluating initial parameters
INFO flwr 2024-03-15 02:51:27,957 | server.py:104 | FL starting
DEBUG flwr 2024-03-15 02:51:27,957 | server.py:222 | fit_round 1: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,104 | server.py:236 | fit_round 1 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,123 | server.py:173 | evaluate_round 1: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,141 | server.py:187 | evaluate_round 1 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,141 | server.py:222 | fit_round 2: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,156 | server.py:236 | fit_round 2 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,159 | server.py:173 | evaluate_round 2: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,175 | server.py:187 | evaluate_round 2 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,176 | server.py:222 | fit_round 3: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,191 | server.py:236 | fit_round 3 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,193 | server.py:173 | evaluate_round 3: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,213 | server.py:187 | evaluate_round 3 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,213 | server.py:222 | fit_round 4: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,227 | server.py:236 | fit_round 4 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,230 | server.py:173 | evaluate_round 4: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,247 | server.py:187 | evaluate_round 4 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,247 | server.py:222 | fit_round 5: strategy sampled 3 clients (out of 3)
DEBUG flwr 2024-03-15 02:51:32,261 | server.py:236 | fit_round 5 received 3 results and 0 failures
DEBUG flwr 2024-03-15 02:51:32,264 | server.py:173 | evaluate_round 5: strategy sampled 3 clients (out of 3)

77390326), (771, 62.99212598425197), (772, 61.754780652418454), (773, 61.64229471316085), (774, 62.76715410573678), (775, 63.77952755905512), (776, 63.32958380202475), (
777, 62.20472440944882), (778, 61.754780652418454), (779, 61.754780652418454), (780, 60.967379077615306), (781, 62.20472440944882), (782, 62.5421822272216), (783, 60.404
949381327334), (784, 61.754780652418454), (785, 61.754780652418454), (786, 62.65466816647919), (787, 62.09223847019123), (788, 62.09223847019123), (789, 62.4296962879640
06), (790, 63.32958380202475), (791, 61.417322834645674), (792, 61.64229471316085), (793, 62.99212598425197), (794, 61.86726659167604), (795, 62.76715410573678), (796, 6
0.967379077615306), (797, 63.10461192350956), (798, 61.754780652418454), (799, 61.52980877390326), (800, 61.079865016872894), (801, 62.65466816647919), (802, 61.41732283
4645674), (803, 62.31721034870641), (804, 63.66704161979753), (805, 62.879640044994375), (806, 61.97975253093363), (807, 63.32958380202475), (808, 61.86726659167604), (8
09, 62.567121034870641), (810, 62.76715410573678), (811, 67.041619752531), (812, 59.84251968503938), (813, 62.76715410573678), (814, 62.879640044994375), (815, 61.41732
2834645674), (816, 61.754780652418454), (817, 65.12935883014623), (818, 61.754780652418454), (819, 61.52980877390326), (820, 59.5850618672666), (821, 62.20472440944882),
(822, 61.417322834645674), (823, 59.730033745701775), (824, 66.81664791901012), (825, 61.97975253093363), (826, 62.31721034870641), (827, 61.754780652418454), (828, 61.
19235095613048), (829, 61.754780652418454), (830, 61.86726659167604), (831, 62.76715410573678), (832, 62.65466816647919), (833, 61.19235095613048), (834, 61.642294713160
85), (835, 61.86726659167604), (836, 61.754780652418454), (837, 62.65466816647919), (838, 61.754780652418454), (839, 61.64229471316085), (840, 62.429696287964006), (841,
63.10461192350956), (842, 62.20472440944882), (843, 62.20472440944882), (844, 64.1169853768279), (845, 61.754780652418454), (846, 58.94263217097863), (847, 63.657041619
79753), (848, 62.65466816647919), (849, 61.97975253093363), (850, 62.09223847019123), (851, 61.30483689538808), (852, 61.754780652418454), (853, 60.742407199100114), (85
4, 64.90438695163104), (855, 62.09223847019123), (856, 62.5421822272216), (857, 62.879640044994375), (858, 61.86726659167604), (859, 61.417322834645674), (860, 60.503937
00787401), (861, 61.19235095613048), (862, 60.292463442069746), (863, 62.65466816647919), (864, 61.754780652418454), (865, 66.92913385826772), (866, 61.64229471316085), (
867, 68.2789651293582), (868, 61.97975253093363), (869, 62.76715410573678), (870, 61.417322834645674), (871, 61.97975253093363), (872, 63.10461192350956), (873, 59.730
033745781775), (874, 61.64229471316085), (875, 61.64229471316085), (876, 64.90438695163104), (877, 62.99212598425197), (878, 62.09223847019123), (879, 64.67941507311586),
(880, 60.292463442069746), (881, 61.879865016872894), (882, 59.392575928080995), (883, 61.754780652418454), (884, 61.754780652418454), (885, 62.09223847019123), (886,
63.66704161979753), (887, 62.09223847019123), (888, 62.879640044994375), (889, 60.51743532058492), (890, 62.429696287964006), (891, 61.754780652418454), (892, 61.1923509
5613048), (893, 61.754780652418454), (894, 61.64229471316085), (895, 64.34195725534309), (896, 61.97975253093363), (897, 63.66704161979753), (898, 61.754780652418454), (
899, 62.429696287964006), (900, 68.16647919010124), (901, 62.429696287964006), (902, 59.955005624296966), (903, 61.86726659167604), (904, 62.5421822272216), (905, 61.642
29471316085), (906, 60.292463442069746), (907, 61.64229471316085), (908, 62.76715410573678), (909, 61.97975253093363), (910, 62.09223847019123), (911, 61.754780652418454),
(912, 61.417322834645674), (913, 61.30483689538808), (914, 64.67941507311586), (915, 61.86726659167604), (916, 60.967379077615306), (917, 67.2665916760405), (918, 61.
64229471316085), (919, 63.55455680539934), (920, 61.64229471316085), (921, 61.30483689538808), (922, 61.417322834645674), (923, 61.97975253093363), (924, 62.76715410573
678), (925, 62.99212598425197), (926, 60.8548931383577), (927, 61.52980877390326), (928, 62.210797862767154), (929, 62.31721034870641), (930, 61.754780652418454), (931,
61.97975253093363), (932, 62.20472440944882), (933, 61.417322834645674), (934, 61.754780652418454), (935, 62.65466816647919), (936, 61.754780652418454), (937, 61.6422947
1316085), (938, 61.754780652418454), (939, 58.83014623172104), (940, 62.09223847019123), (941, 61.754780652418454), (942, 61.52980877390326), (943, 60.5399325084365), (
944, 61.64229471316085), (945, 61.754780652418454), (946, 61.30483689538808), (947, 59.955005624296966), (948, 60.51743532058492), (949, 61.86726659167604), (950, 61.754
780652418454), (951, 61.417322834645674), (952, 61.86726659167604), (953, 64.67941507311586), (954, 60.629921259842526), (955, 62.20472440944882), (956, 62.879640044994375),
(957, 61.754780652418454), (958, 61.52980877390326), (959, 61.86726659167604), (960, 61.754780652418454), (961, 61.417322834645674), (962, 63.210797862767154), (963,
61.52980877390326), (964, 61.52980877390326), (965, 60.967379077615306), (966, 62.76715410573678), (967, 61.86726659167604), (968, 61.079865016872894), (969, 62.0922384
7019123), (970, 61.86726659167604), (971, 61.64229471316085), (972, 61.64229471316085), (973, 60.742407199100114), (974, 61.86726659167604), (975, 61.754780652418454), (
976, 61.754780652418454), (977, 61.86726659167604), (978, 62.09223847019123), (979, 63.66704161979753), (980, 62.09223847019123), (981, 61.754780652418454), (982, 61.879
865016872894), (983, 59.392575928080995), (984, 67.2665916760405), (985, 61.754780652418454), (986, 61.86726659167604), (987, 61.64229471316085), (988, 61.97975253093363),
(989, 61.64229471316085), (990, 61.754780652418454), (991, 61.97975253093363), (992, 61.754780652418454), (993, 61.30483689538808), (994, 61.97975253093363), (995, 62
.65466816647919), (996, 61.86726659167604), (997, 60.967379077615306), (998, 61.754780652418454), (999, 61.754780652418454), (1000, 61.86726659167604)]]
INFO flwr 2024-03-15 02:51:58,003 | app.py:228 | app_fit: metrics_distributed {}
INFO flwr 2024-03-15 02:51:58,003 | app.py:229 | app_fit: losses_centralized {}
INFO flwr 2024-03-15 02:51:58,003 | app.py:230 | app_fit: metrics_centralized {}
(vertical-fl-py3.10) siva@siva-Swift-SF314-S5G:~/vertical-fl$
```

Accuracy plot vs Epochs(Vertical FL)



## References

1. “Communication-Efficient Learning of Deep Networks from Decentralized Data”  
[H. Brendan McMahan](#), [Eider Moore](#), [Daniel Ramage](#), [Seth Hampson](#), [Blaise Agüera y Arcas](#)  
Link: <https://arxiv.org/abs/1602.05629>
2. “A Comparative Evaluation of FedAvg and Per-FedAvg Algorithms for Dirichlet Distributed Heterogeneous Data”  
Hamza Reguieg1 , Mohammed El Hanjri2 , Mohamed El Kamili1 , Abdellatif Kobbane2  
Link: <https://arxiv.org/abs/2309.01275>
3. “Heterogeneous Federated Learning: State-of-the-art and Research Challenges”  
MANG YE, XIUWEN FANG, and BO DU, PONG C. YUEN, DACHENG TAO  
Link: <https://dl.acm.org/doi/10.1145/3625558>