BTP

HDFL - Hierarchical Decentralized Federated Learning

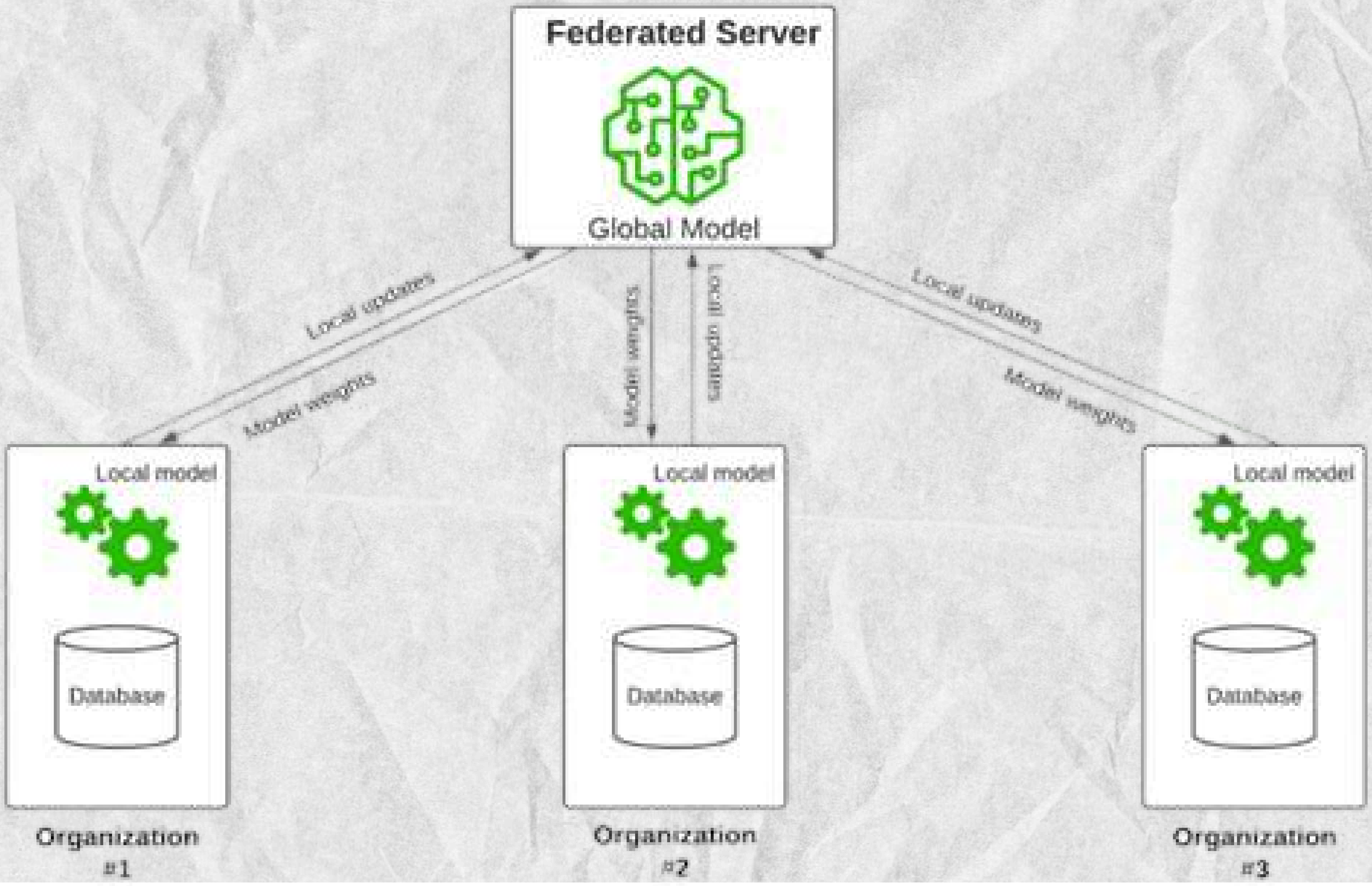
Our Team Members

Kritika Bansal - 2021CSB1184

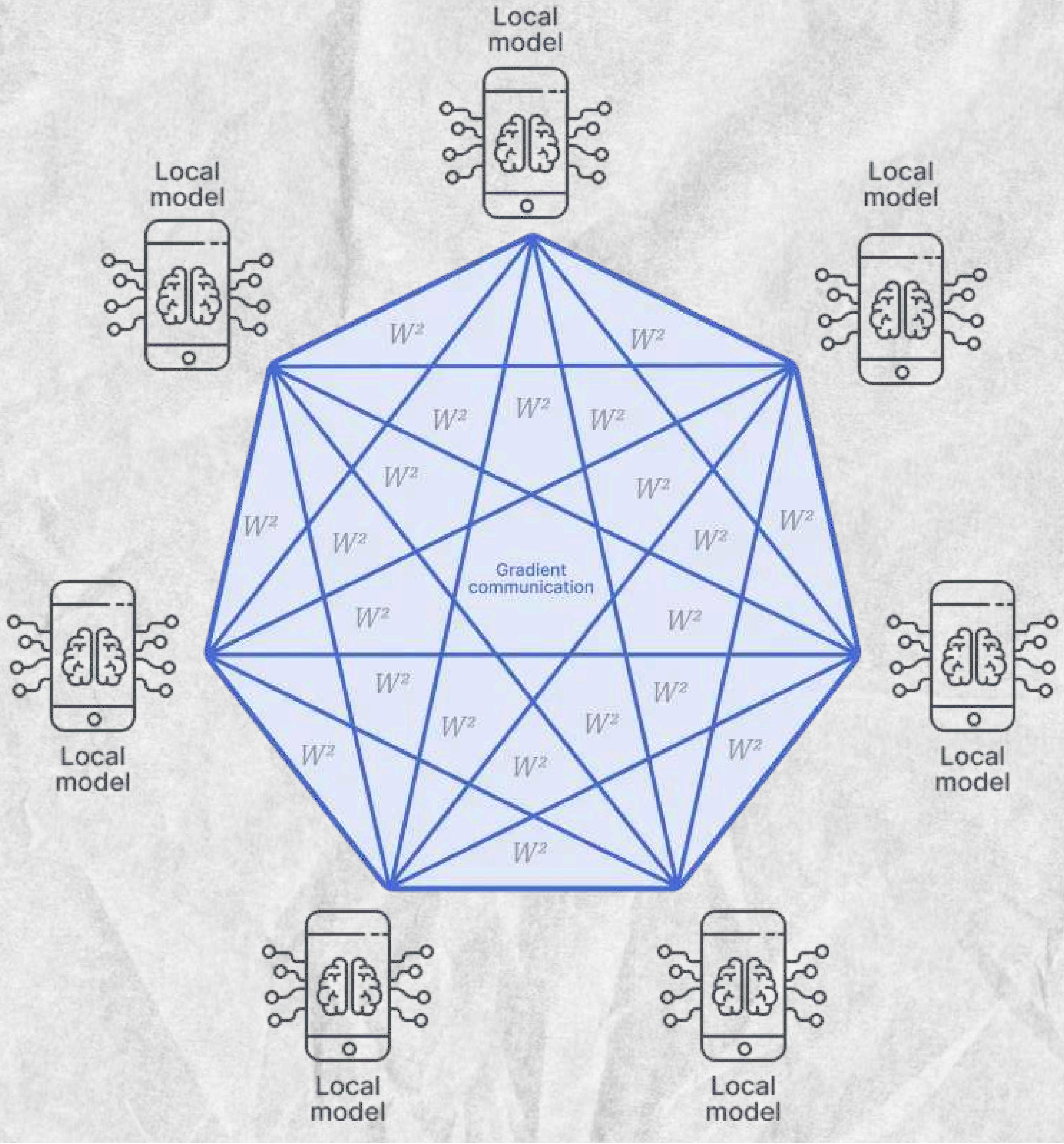
Virat Jain - 2021CSB1184

Supervisor - Dr. Shweta Jain

What is Federated Learning ?



What is Decentralized FL ?



Problem Statement

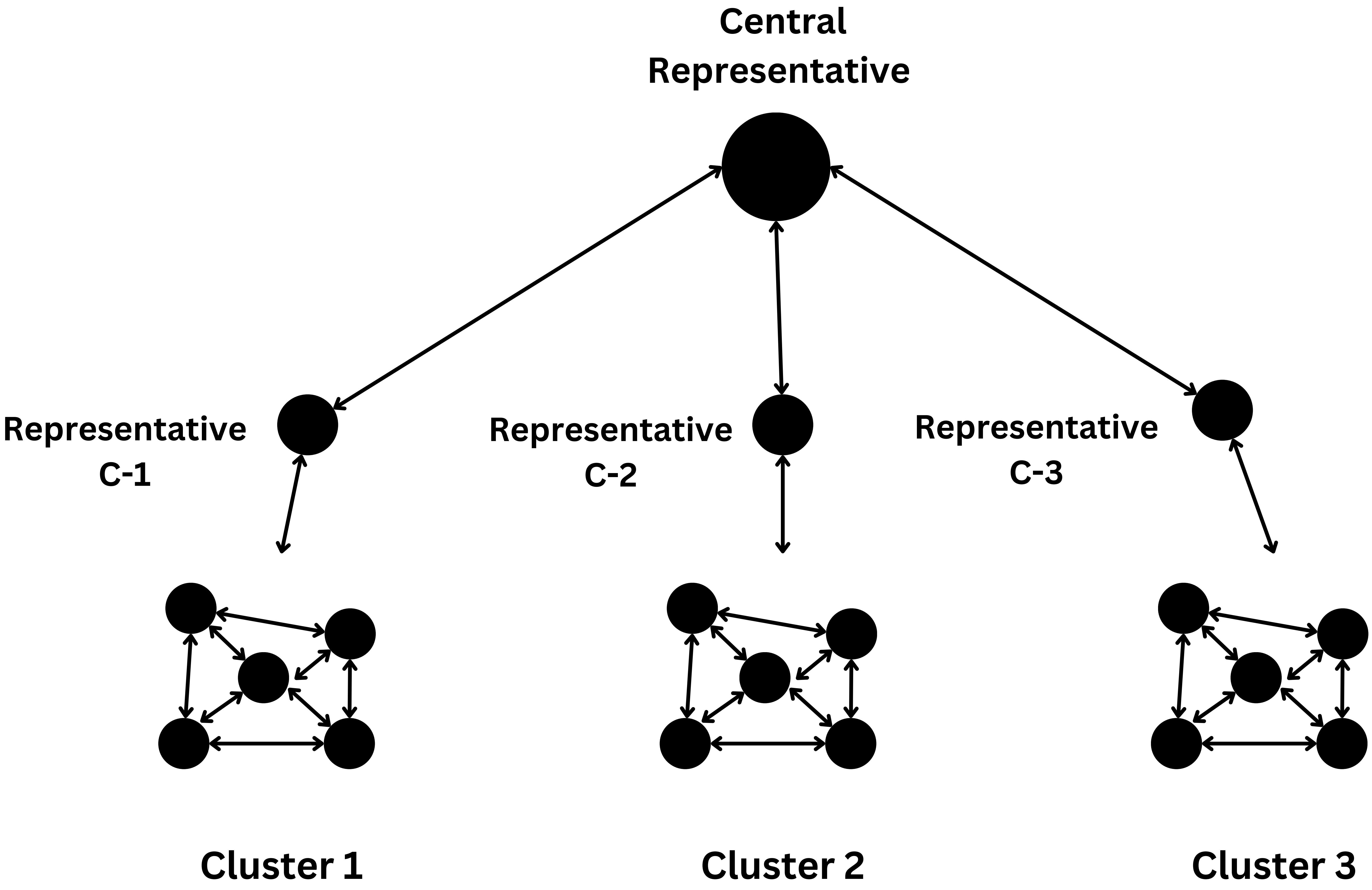
- FL ensures privacy but has a single point of failure due to its central server.
- DFL removes this risk but increases time complexity and communication overhead.
- The challenge is to design a system that eliminates failure points while reducing complexity.

Motivation

- HDFL merges FL and DFL, eliminating the single point of failure with a decentralized hierarchy.
- It reduces time complexity through co-operative grouping of devices.

Methodology

Grouping Algorithm → Intra-group Learning → Inter-group Aggregation

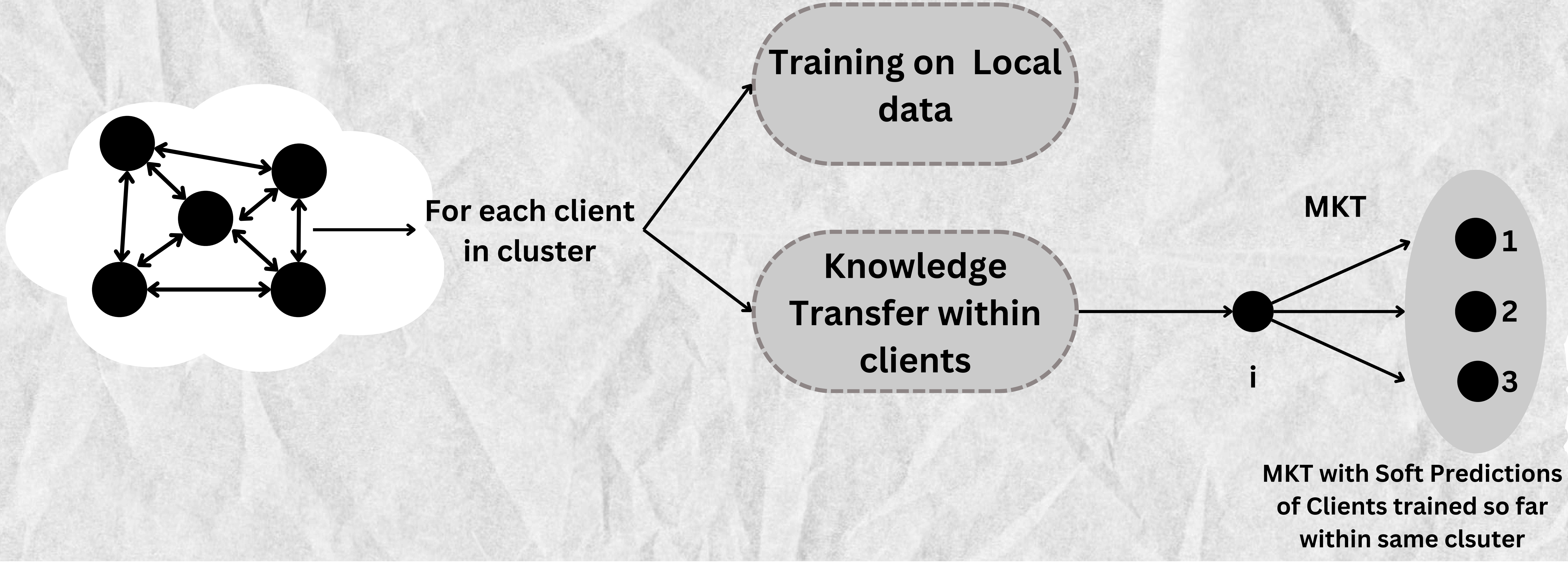

Central
Representative
Representative
C-1
Representative
C-2
Representative
C-3
Cluster 1
Cluster 2
Cluster 3

Work Done Before

- HDFL model by Nitin Sir combined a hierarchical approach with decentralization.
- Used Cosine Similarity to cluster clients based on data distribution.
- Performed well on some datasets and not others.
- Some observations lacked theoretical reasoning, providing scope for improvement.

Our Contribution (7th Sem)

1. Model Architecture Enhancements
2. Learning Rate Optimization
3. Incorporation of Mutual Knowledge Transfer (MKT)



Reference: Decentralized Federated Learning via Mutual Knowledge Transfer

Observations on Mutual Knowledge Transfer:

- **Minimal Impact on Simple Datasets:** MKT shows negligible improvement on MNIST and FashionMNIST due to their low complexity.
- **Significant Gains on CIFAR-10:** This is attributed to the dataset's higher complexity, where MKT facilitates better generalization and knowledge sharing across clients with heterogeneous data distributions.

Method	FL	DFL	HDFL
With MKT	NA	95.8	95.2
W/o MKT	96.8	96.93	97.7

MNIST

Method	FL	DFL	HDFL
With MKT	NA	80	79.5
W/o MKT	83.2	83.5	84.9

FashionMNIST

Method	FL	DFL	HDFL
With MKT	50.8	53.6	51.1
W/o MKT	50.8	51.2	47.7

CIFAR10

4. Clustering Algorithm Analysis:

- Explored different clustering algorithms to identify the most effective approach for client clustering.

Clustering	Random	Cosine Similarity	Kmeans
HDFL	97.43 %	97.7 %	97 %

MNIST

Clustering	Random	Cosine Similarity	Kmeans
HDFL	82.95 %	84.9 %	84.5 %

FashionMNIST

Clustering	Random	Cosine Similarity	Kmeans
HDFL	47.41 %	51.2 %	49.5 %

CIFAR10

5. Data Partitioning Experiments:

- Investigated the effects of various data partitioning strategies on the performance of the HDFL model.

Distribution	FL	DFL	HDFL
2-class	96.8	96.93	97.7
3-class	98	98	98
Dirichlet	98.6	98.25	98.7
IID	98.91	98.86	98.95

MNIST

Distribution	FL	DFL	HDFL
2-class	83.2	83.5	84.9
3-class	85.1	86	85.8
Dirichlet	83.1	83.7	84.5
IID	90.09	90.42	89.27

FashionMNIST

Distribution	FL	DFL	HDFL
2-class	53	52.1	51.2
3-class	61.5	62.8	60.23
Dirichlet	62.5	63.9	63.7
IID	70.11	70.63	64.91

CIFAR10

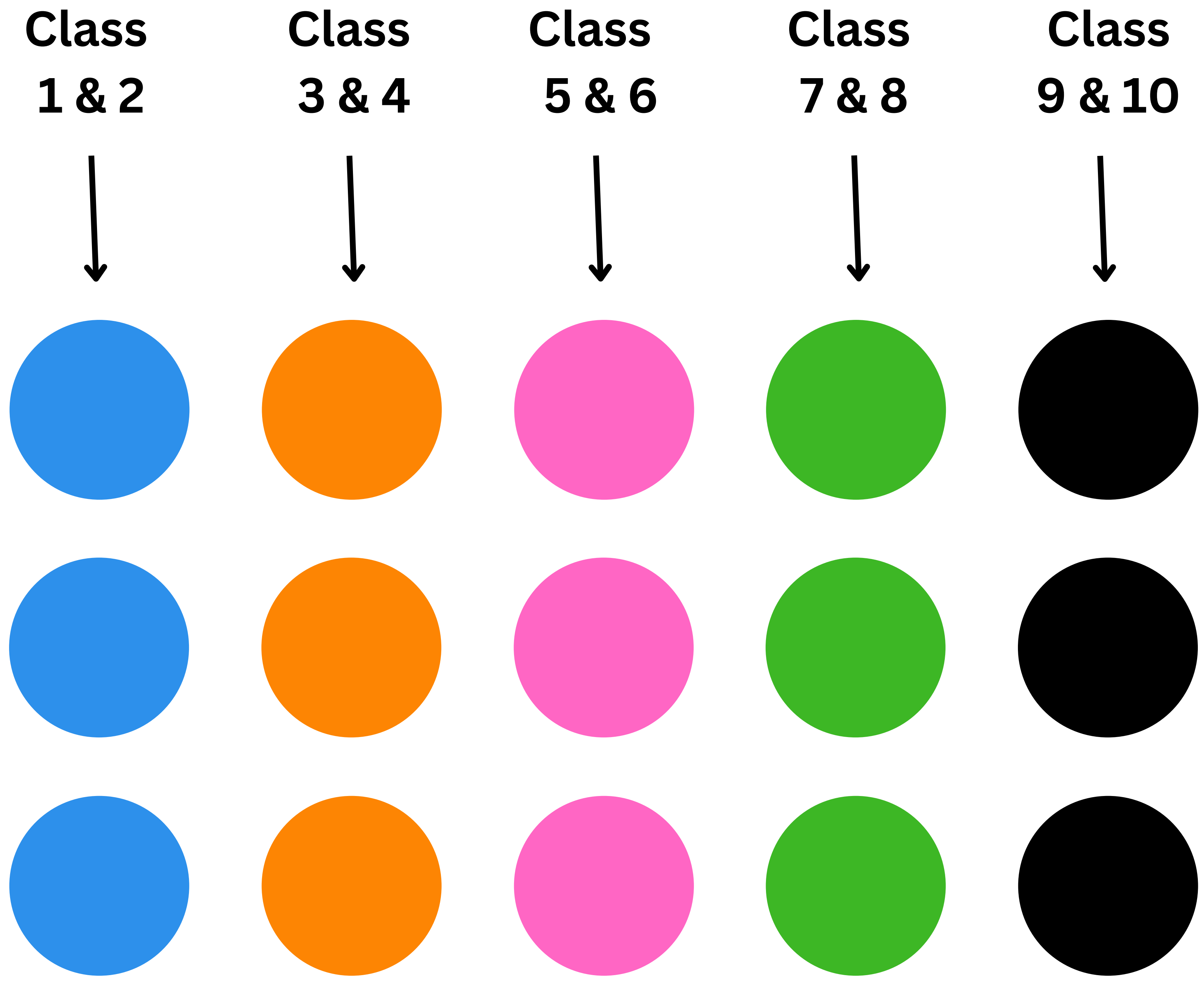
Observations on data partitioning:

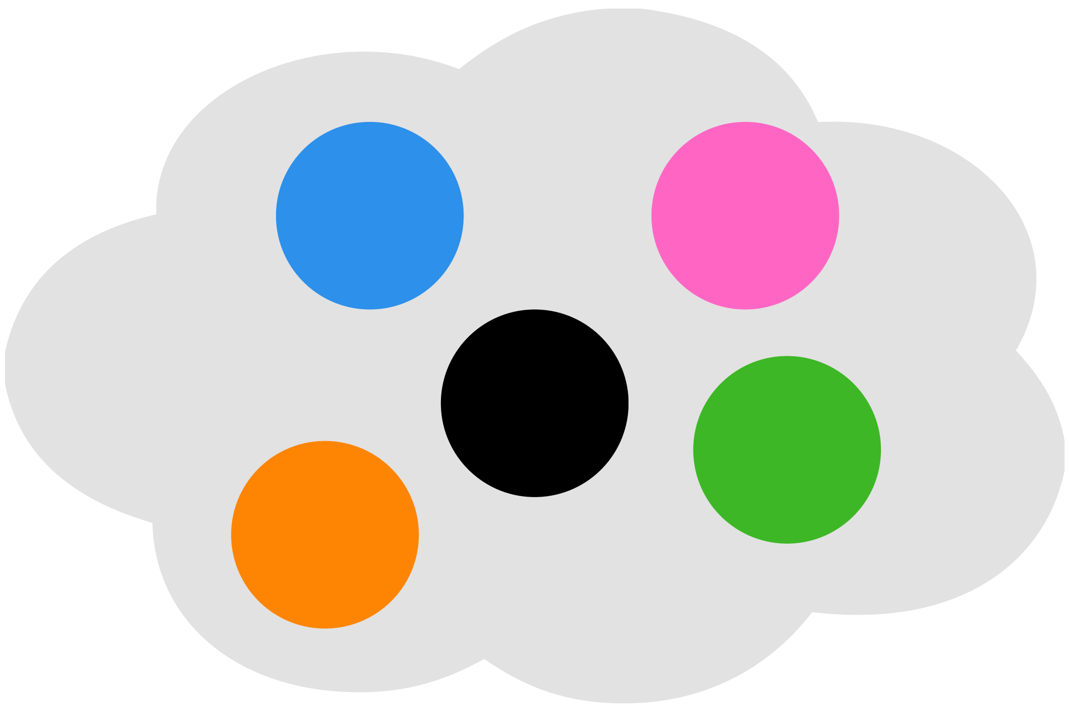
- Dirichlet Partitioning: Achieved the best performance by preserving diverse feature distributions, enhancing generalization.
- 3 Classes per Client: Performed moderately well; improved diversity over the 2-class setup but still limited generalization.
- 2 Classes per Client: Performed the worst due to restricted diversity, leading to overfitting and poor generalization.

Our Contribution (8th Sem)

1. <u>**Heterogeneity of data:**</u>

   - **Manual Clustering Experiment:**
     - 25 clients, each assigned 2 classes.
     - Formed clusters: [1,2,3,4,5], [6,7,8,9,10], [11,12,13,14,15].
     - Each cluster collectively covered all 10 classes.
     - Results showed that intra-cluster heterogeneity helps in learning balanced representations.



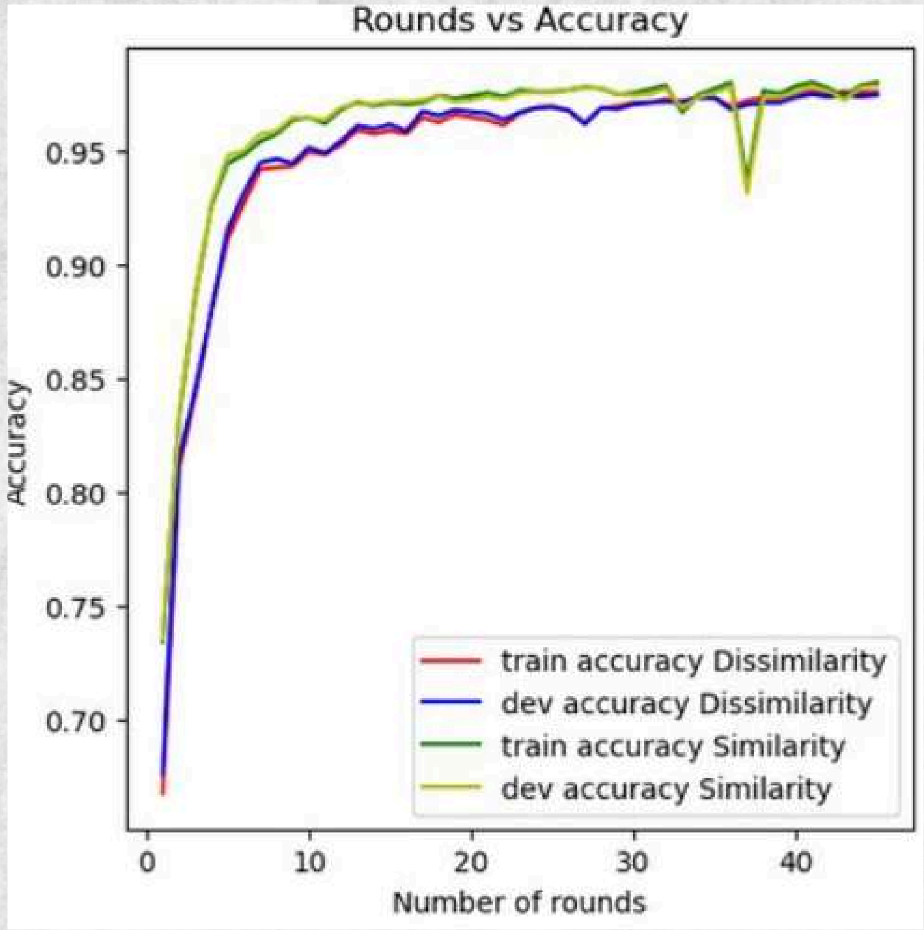


As all classes are covered by clients in this cluster, the weights shared by this cluster will not be biased towards any class and should improve performance of overall model

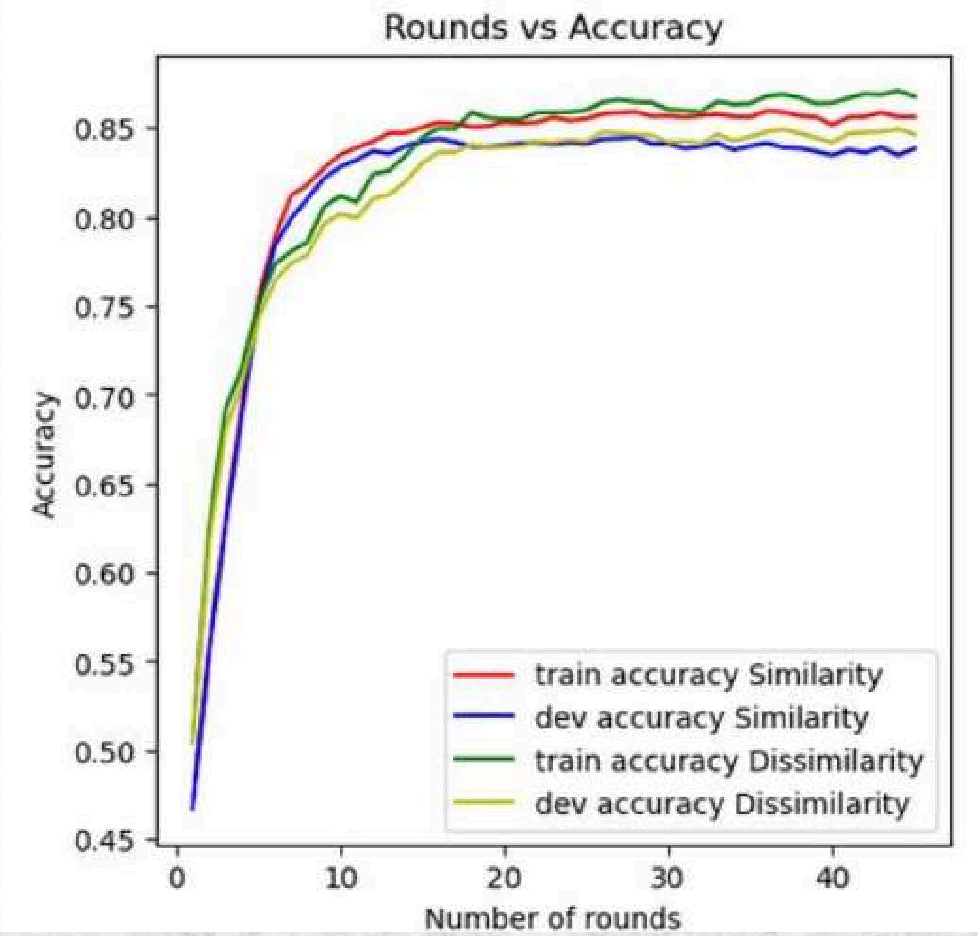
- **Cosine Dissimilarity-Based Clustering:**
  - To exploit heterogeneity, used cosine dissimiliarity in HDFL to form clusters
  - Built clusters using cosine dissimilarity for better feature separation

Results on HDFL

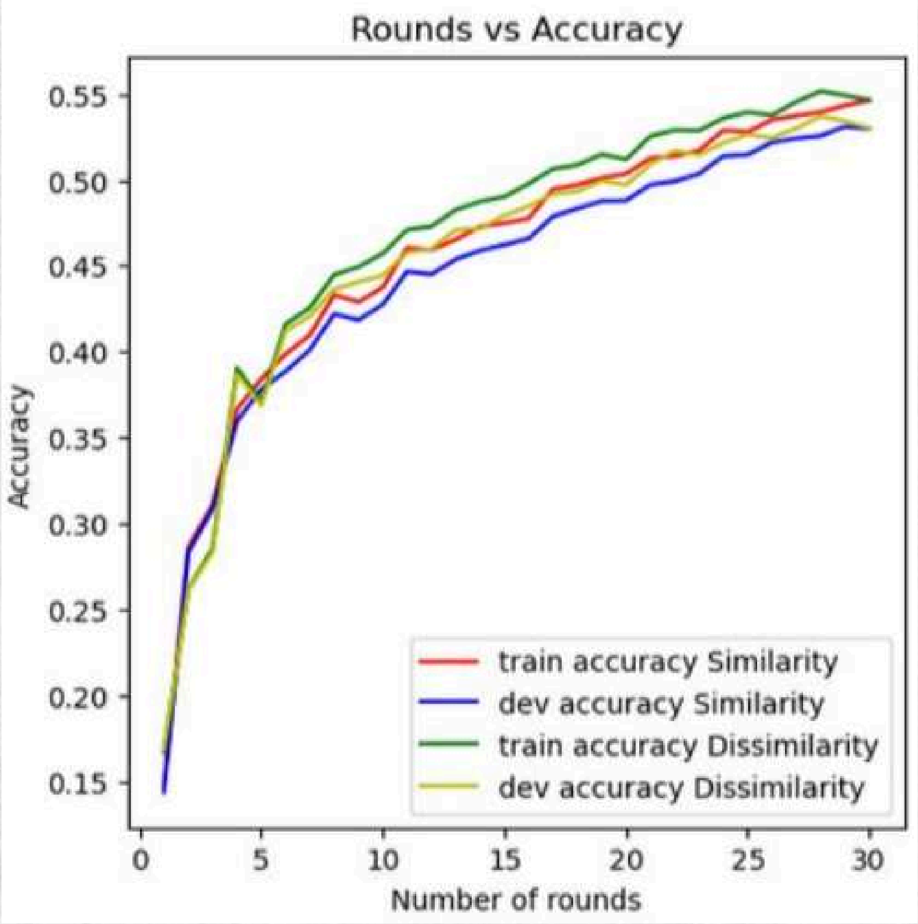
Method	Similarity	Dissimilarity
Mnist	97.45	97.8
FashionMnist	83.6	84.5
Cifar10	53.1	53.8



Mnist



FashionMnist



Cifar10

2. Subset Selection of Clusters in HDFL:

- **Challenge:** Not all clusters have well-balanced weights, as some lack representation across all classes. Biased clusters can negatively impact accuracy.

- **Goal:** Select a subset of clusters in each round that best represent all classes.

Approaches:

1. **Top-K Cosine Dissimilarity:**
   - Select the top k clusters with the highest pairwise average cosine dissimilarity.

   - Ensures diverse representation by prioritizing clusters with distinct feature distributions.

Approaches:

2. **DivFL:**
    - **Random Sampling** → The server picks a small random subset of clients instead of evaluating all.
    - **Gradient-Based Selection** → From this subset, clients with diverse and representative gradients are chosen.
    - **Greedy Optimization** → A stochastic greedy algorithm selects clients one by one, maximizing diversity.

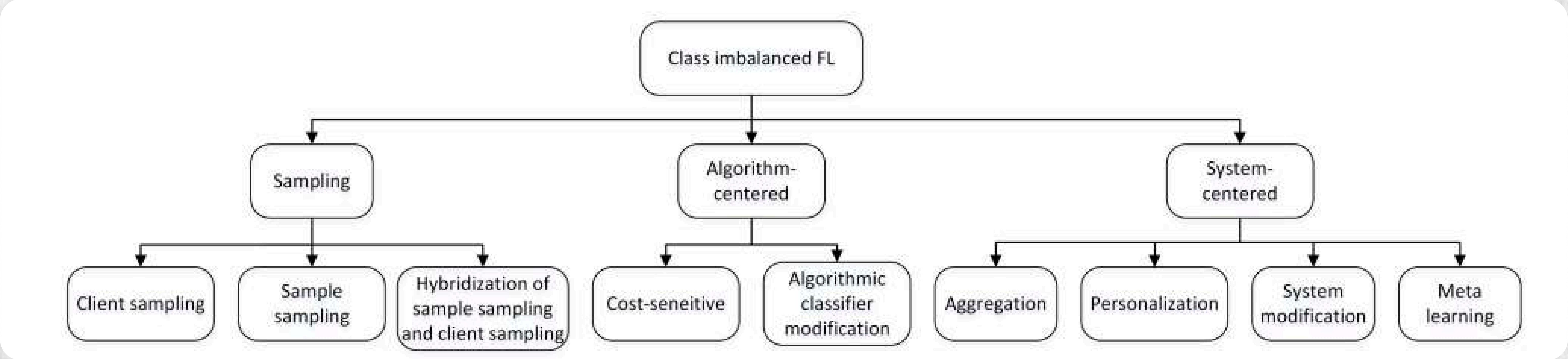
Reference: https://openreview.net/pdf?id=nwKXyFvaUm

3. <u>Unbalanceness in data:</u>

- **Challenge:** Datasets have an unequal number of samples per class, leading to biased model weights favoring majority classes.

- **Solution:** Address imbalance to obtain fairer weight distributions.

Various methods to tackle Unbalanceness



Reference: https://arxiv.org/pdf/2303.11673

Future Plans

- Work subset selection of clusters to improve performance of model
- **Cost-sensitivity**: modifies the loss function by assigning higher penalties to misclassified instances from the minority class. This ensures that the model pays more attention to correctly classifying these underrepresented samples.
- Test model on other datasets

Thank you