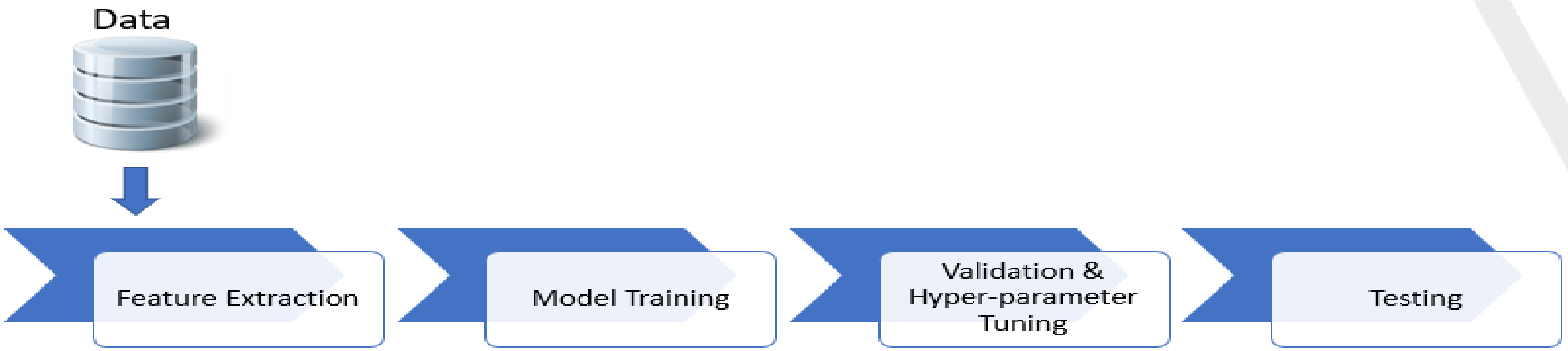
Is Your Data Relevant?: Dynamic Selection of Relevant Data for Federated Learning

Paper ID: 5983

Lokesh Nagalapatti (*)	-	IIT Bombay	-	nlokesh@cse.iitb.ac.in
Ruhi Sharma Mittal (*)	-	IBM Research	-	ruhi.sharma@in.ibm.com
Ramasuri Narayanam	-	Adobe Research India	-	rnarayanam@adobe.com

(*) - Equal Contribution

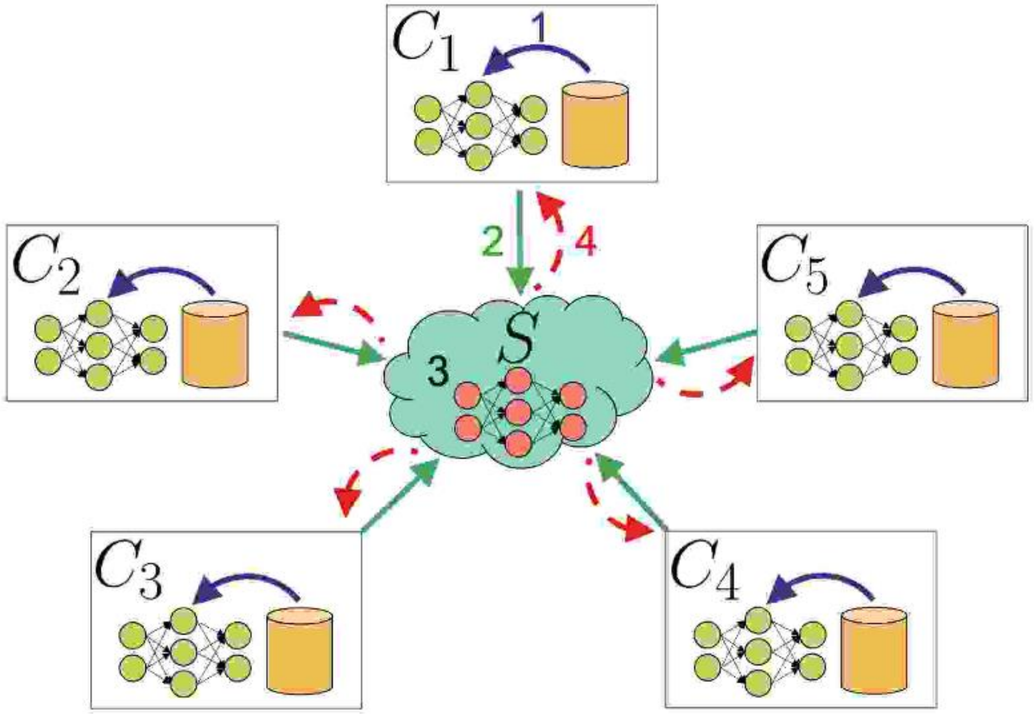
Standard Machine Learning (ML) Setting



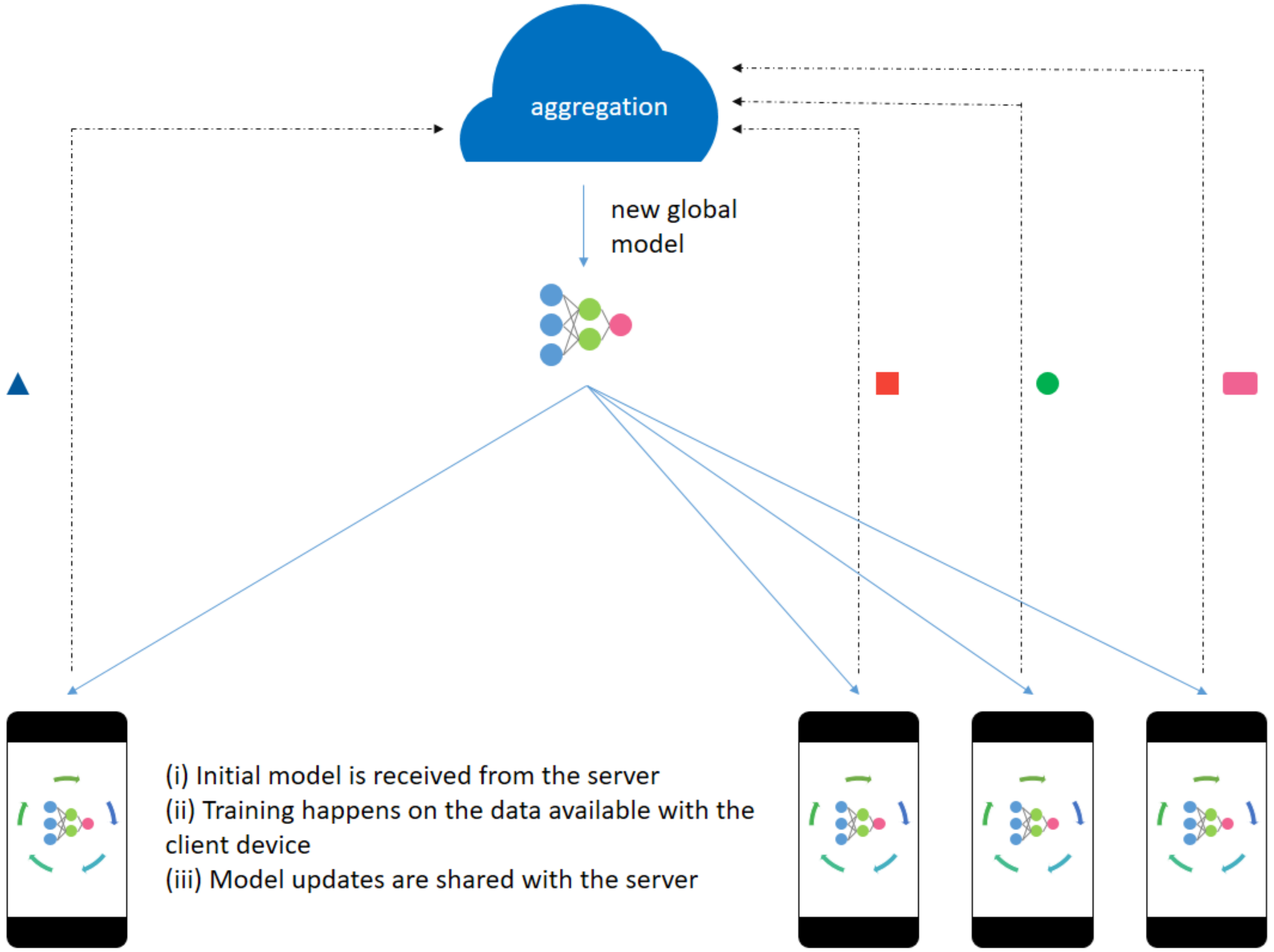
Standard machine learning algorithms assume the whole Data to be available at one central place (machine or data centre)

Federated Learning (FL) Setting

[McMahan et.al. AISTATS 2017] The paradigm of *Federated learning (FL)* deals with multiple clients (owning private data) participate in collaborative training of a machine learning model under the orchestration of a central server.



Federated Learning architecture



Reference: https://towardsdatascience.com/introduction-to-federated-learning-and-privacy-preservation-75644686b559

One round of Federated Learning

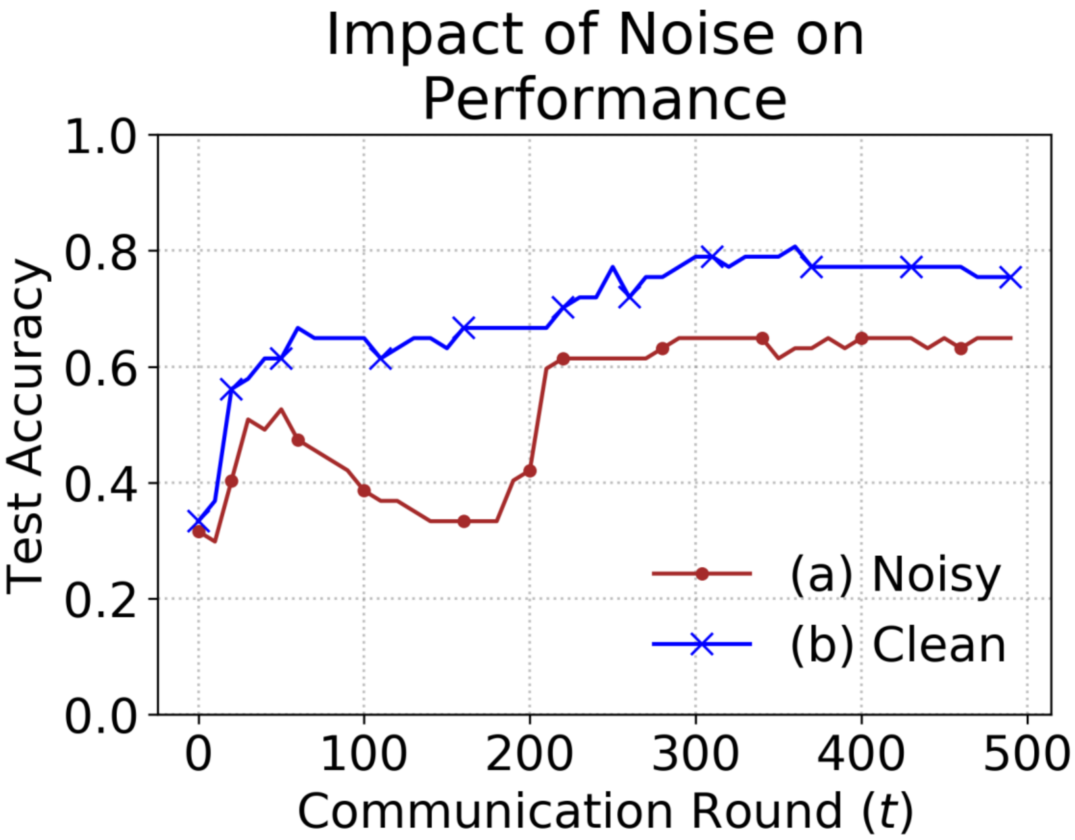
- Client Selection
- Parameter broadcast
- Local client update compute
- Aggregation
- Model update

Federated Learning Setup

- Server needs to learn a global Learning model (***GLM***) $f_\theta: X \rightarrow Y$
- The training data $D = D_1 \cup \cdots \cup D_N$ is partitioned across N clients
- Server possess a small validation dataset D_v that had iid samples from ground truth distribution
- In each round, all clients are sampled
- Each client i derives gradients δ_i from a subset of dataset for K steps and sends it back to server
- Server averages the gradients and applies it to *GLM*

Motivating Experiment

- Partitioned Iris dataset across 2 clients
- Injected 20% closed-set Label noise
- Ran Federated Averaging for 500 rounds



Problem Statement

- We observe that there is value in each client deriving update only from clean data points
- Thus, each client i needs to learn a Relevant Data Selector (RDS_i) $g_{\phi_i}: (X, Y) \rightarrow [0,1]$
- In each round client can thus sample useful points thereby sharing useful updates
- The training objective thus is:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{(x,y)\in D_i} g_{i\phi_i}(x, y) \cdot l(y, f_\theta(x))$$

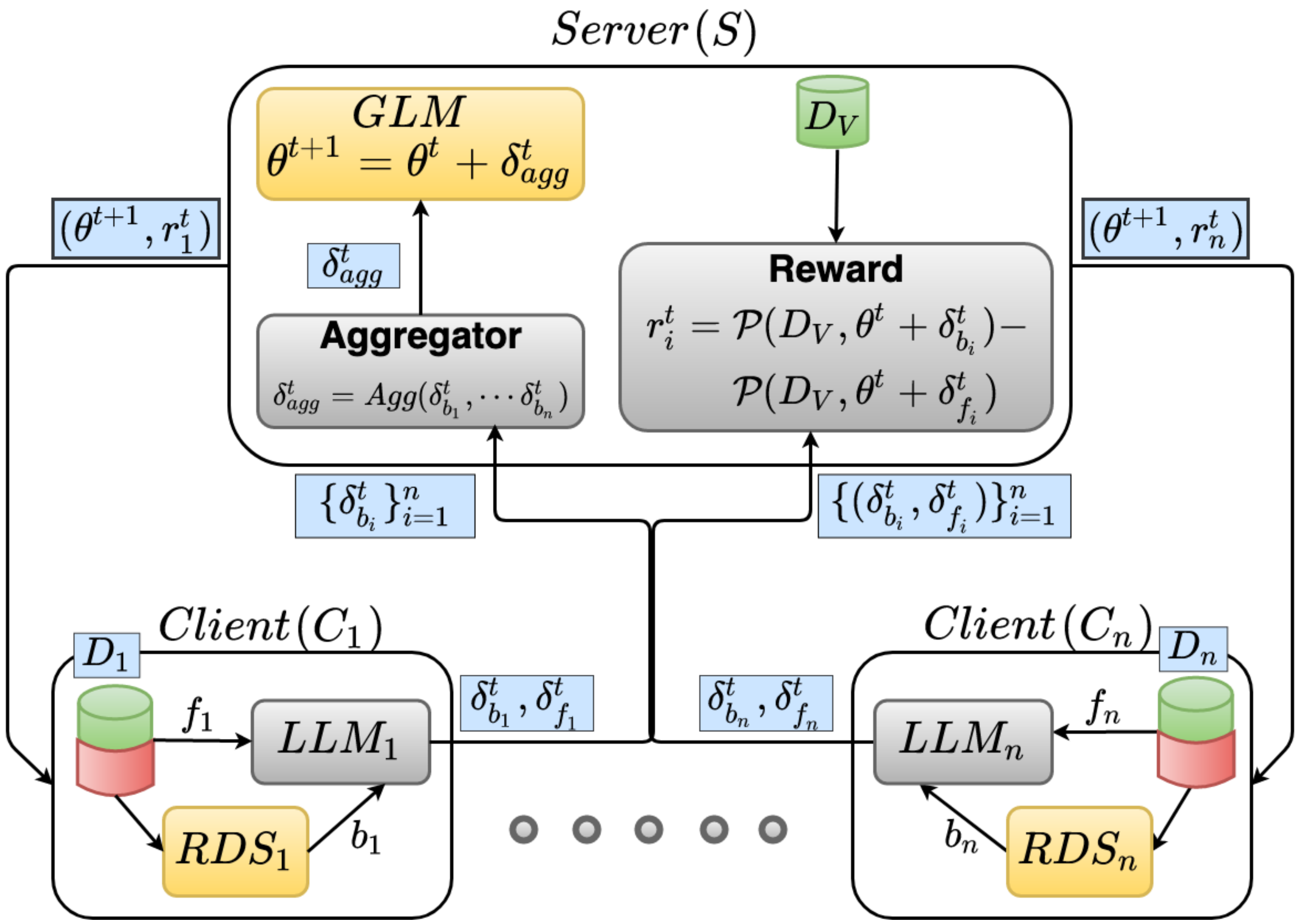
Proposed Approach

- We train RDS using Policy Gradients Algorithms
- Assumption:
  - At each round, client sends two updates
  - δ_f^t gradient computed from full data
  - δ_b^t: gradient computed using subset sampled according to RDS
- With this protocol, reward for client i at time t is computed as:

$$r_i^t(b_i) = \mathcal{P}(\theta^t + \delta_{b_i}^t) - \mathcal{P}(\theta^t + \delta_{f_i}^t)$$

$$\mathcal{P}(\theta) = \frac{1}{|D_V|} \sum_{(x,y) \in D_V} \mathcal{I}(y == f_\theta(x))$$

Proposed Architecture of FLRD



Experiment: Irrelevant Data Samples Detection

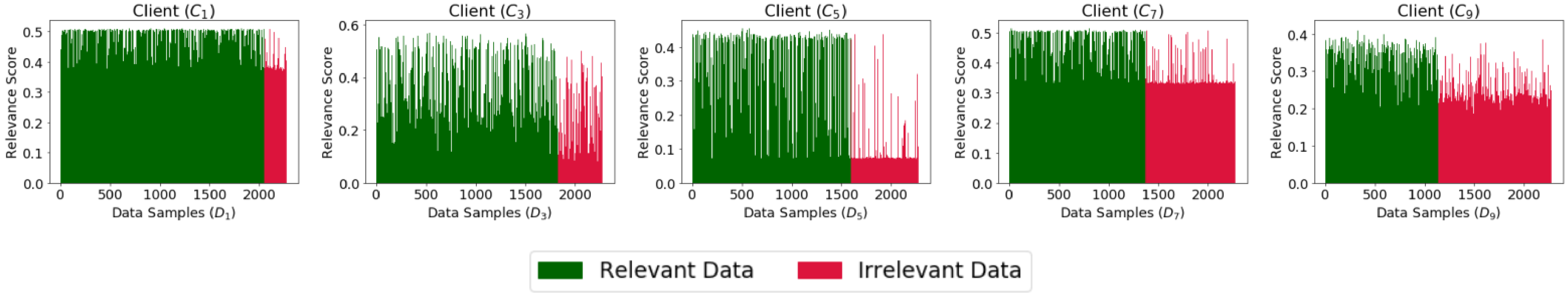


Figure 3: Relevance scores of clients using Adult dataset with closed-set label noise obtained after 100 communication rounds. The noise percentage in client C_1, C_2 is 10%; C_3, C_4 is 20%; C_5, C_6 is 30%; C_7, C_8 is 40%; C_9, C_{10} is 50%. The data samples are sorted for the representational purpose only; however, in the training data the samples are shuffled.

Impact of removing High valued Data

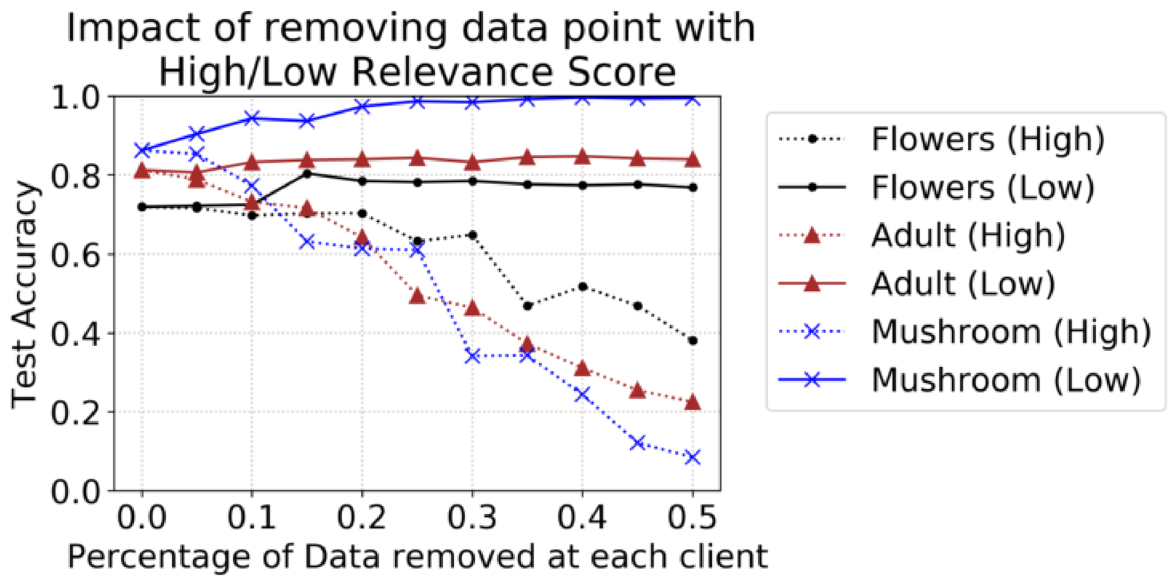


Figure 6: Performance of GLM on D_{Test} after removing data samples with the high/low relevance score at each client.

Experiment: Closed set Label Noise

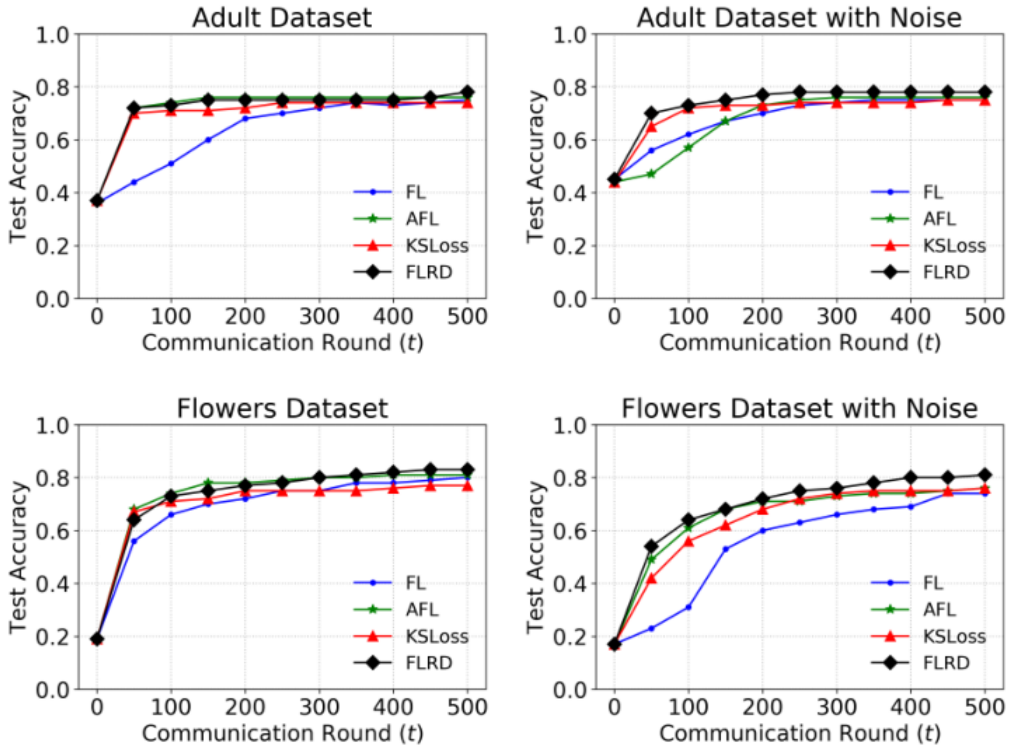


Figure 5: Performance of GLM on D_{Test} using $FLRD$ and other baselines across multiple communication rounds with the original dataset (without noise) and noisy dataset (with noise).

Experiment: Attribute Noise

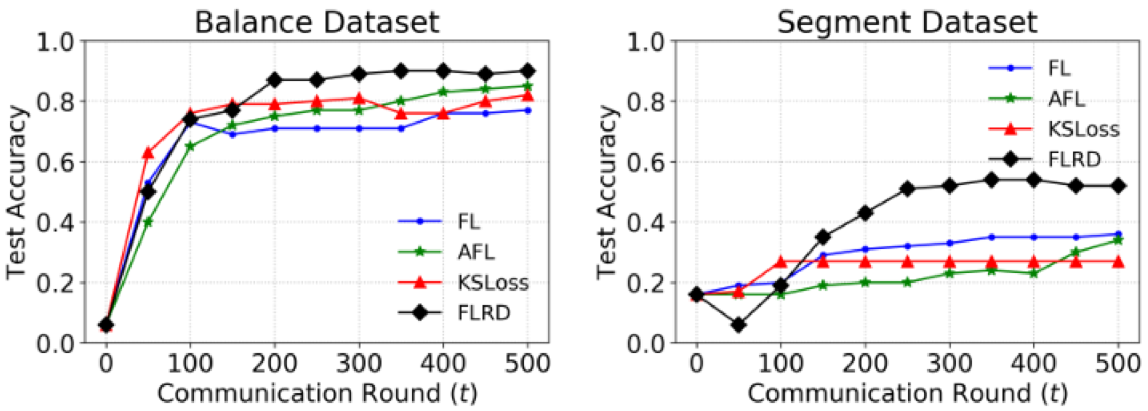


Figure 4: Performance of GLM on test data using $FLRD$ and other baselines across multiple communication rounds with datasets having 5% attribute noise.

Experiment: Robustness to Noise

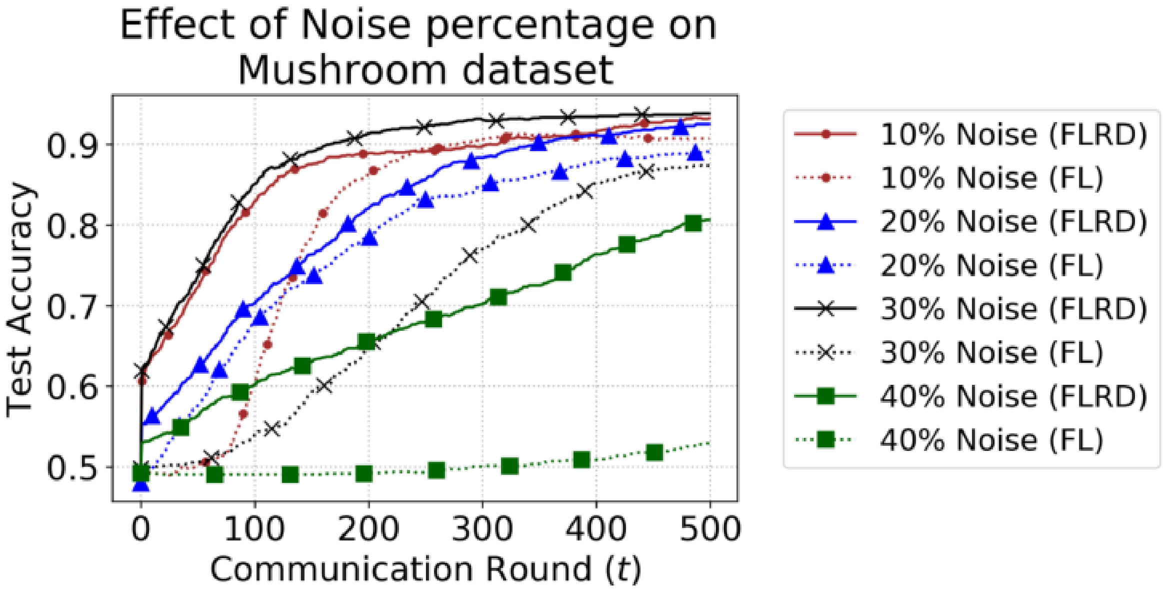


Figure 7: Parameter Sensitivity: noise percentage

Effect of Size of D_v

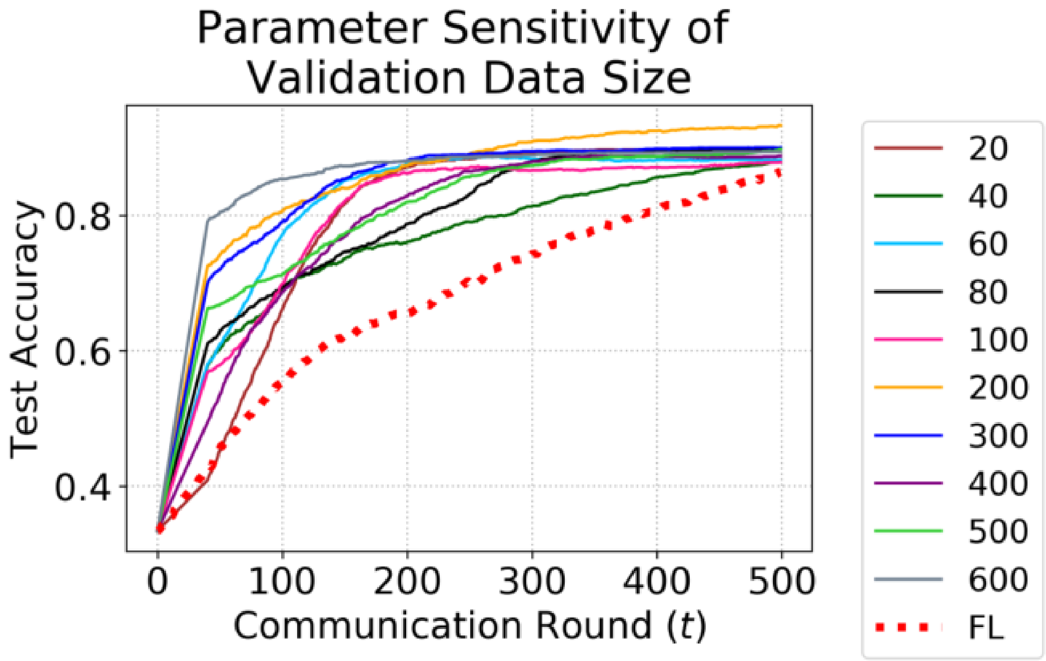


Figure 8: Parameter Sensitivity: validation dataset size

Conclusion

- We proposed an approach called FLRD that is instrumental in selecting relevant data at each client
- The proposed approach can tackle various types of noise in data
- In future, we like to extent RDS_i to Active Learning settings
- The proposed Policy gradients-based method to train RDS_i does not take cost of exploration into account which is substantial in Active Learning