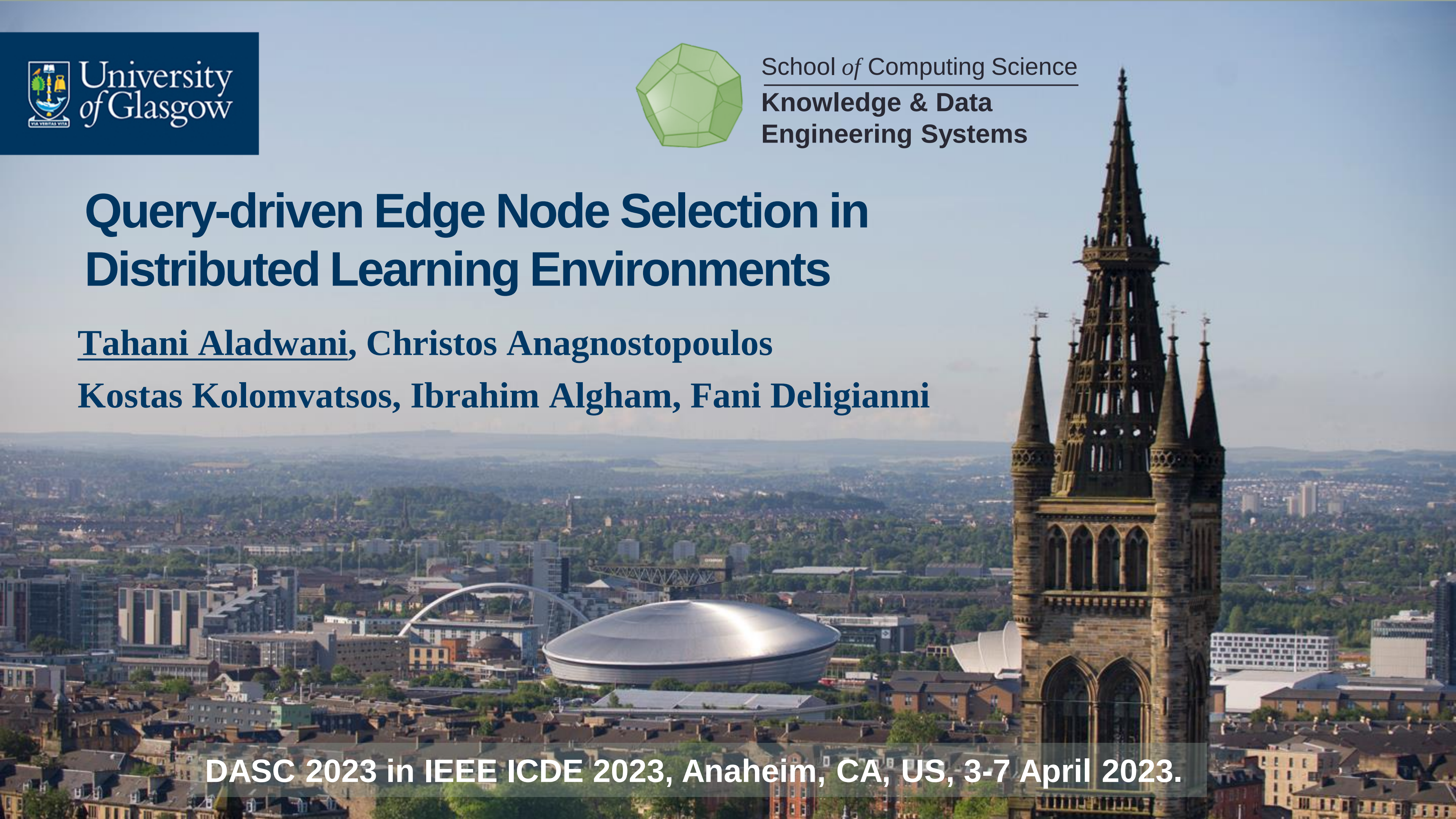


University of Glasgow

School *of* Computing Science

Knowledge & Data Engineering Systems

Query-driven Edge Node Selection in Distributed Learning Environments

Tahani Aladwani, Christos Anagnostopoulos

Kostas Kolomvatsos, Ibrahim Algham, Fani Deligianni

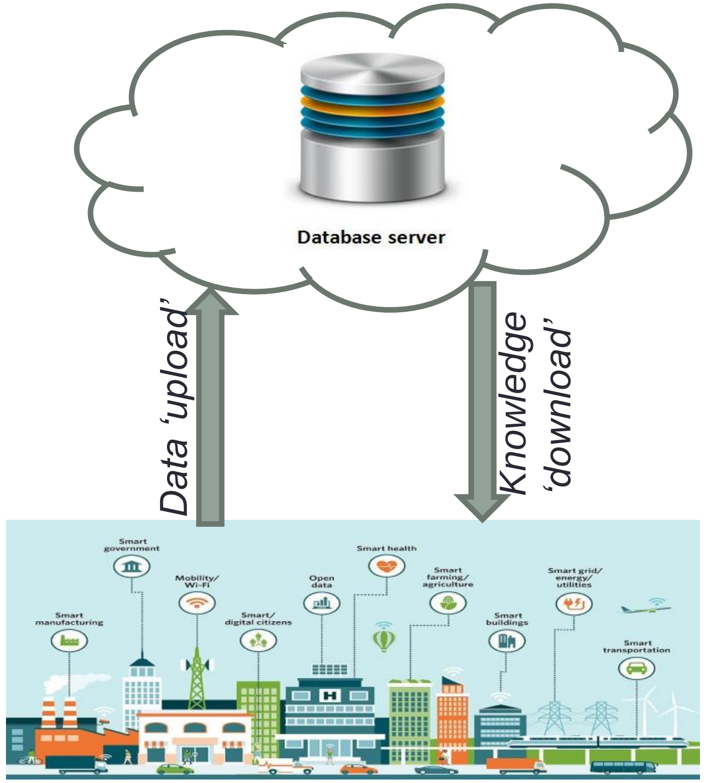
DASC 2023 in IEEE ICDE 2023, Anaheim, CA, US, 3-7 April 2023.

Introduction

Machine & Deep Learning (ML/DL) models are typically trained using centralized data.

However, bringing **all** data into a centralized server is no longer practical:

- **Violation of data privacy**
- **Communication burden due to data transfer**



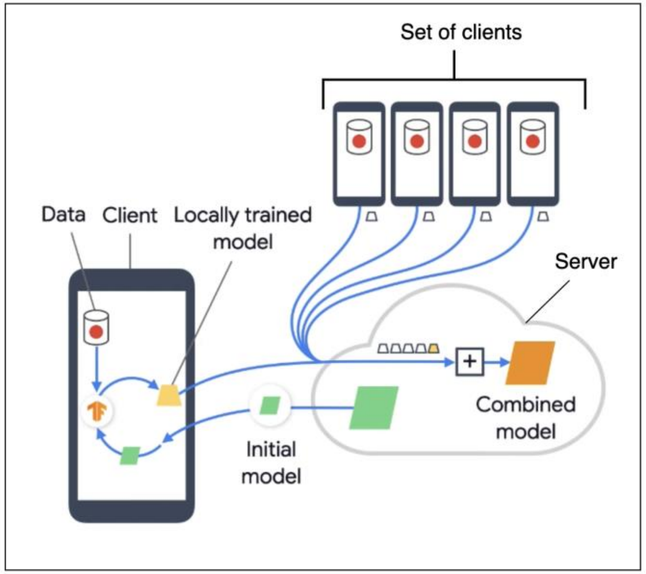
Solution: Distributed ML Model Training

Distributed Learning facilitates access to distributed data by training a ML model over *disjoint* data spaces by leveraging **nodes' local data and computational resources**.

Aim: Train a ML model efficiently requires training over a set of nodes, a.k.a., participants.

However, not all participants play the same role.
This is determined by:

- **Amount** of available data in each participant.
- **Quality** of the data in each participant.
- Percentage of **data overlap** between query's data requirement (analytics task) and participant's available data.



Problem Fundamentals

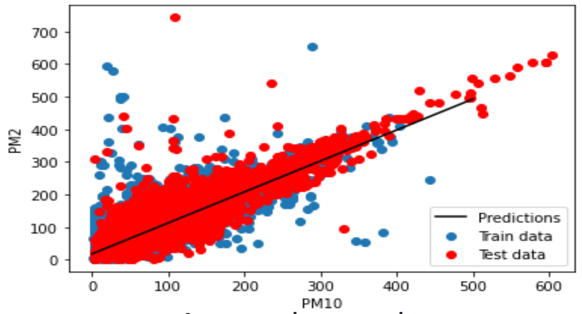
A network of communicating nodes $\mathbb{N}$ being **heterogeneous** in terms of data distributions and spaces.

Set of analytics queries $\mathbb{Q} = \{q_1, q_2, q_3, \ldots, q_n\}$

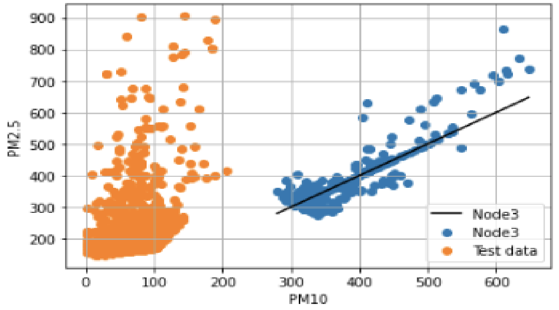
Each query q in $\mathbb{Q}$ is an analytic/ML learning task that **requires access** to d-dim. data to be executed.

Given a query q, we engage nodes for the corresponding task. **However**, by selecting not appropriate nodes given a query, it might degrade the effectiveness of distributed ML learning.

Problem: Given a query q, **find** and **engage** the most appropriate subset of participants in the ML training task.



Appropriate node



Not appropriate node

Data Overlapping

Objective: Determine the **appropriateness** of a node for a query by exploring the node's available data overlapping between node's data and query.

Given a **query *q_k*** and **node *n_k***, estimate the **percentage of data** (out of the whole **node's dataset *D_k***) required for executing that query.

Step 1: The node ***n_k*** quantizes its own data space ***D_k***, by e.g., adopting *k*-means into **K clusters**.

$$\min_{\{u_1,\ldots,u_K\}} \sum_{k=1}^{K} \sum_{j=1}^{m} \|\xi_j - u_k\|^2.$$

Goal: find the number of clusters (out of K clusters) that have **high** overlapping with query ***q_k***

Data Overlapping per Cluster

Step 2: Extract the **boundaries of each cluster** across all data dimensions, e.g., taking the minimum and maximum for each data dimension per cluster, obtaining the vector:

$$\boldsymbol{\mathcal{K}} = [\boldsymbol{k_1^{min}}, \boldsymbol{k_1^{max}}, \boldsymbol{k_2^{min}}, \boldsymbol{k_2^{max}}, \dots, \boldsymbol{k_d^{min}}, \boldsymbol{k_d^{max}}]$$

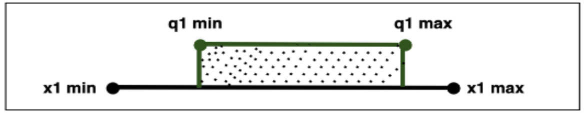
Step 3: Obtain the query $\boldsymbol{q_k}$ boundaries (e.g., hyper-rectangle) to calculate the **overlapping** between query’s rectangle and each cluster's rectangle per dimension. The query is represented by the vector:

$$\boldsymbol{q_k} = [\boldsymbol{q_1^{min}}, \boldsymbol{q_1^{max}}, \boldsymbol{q_2^{min}}, \boldsymbol{q_2^{max}}, \dots, \boldsymbol{q_d^{min}}, \boldsymbol{q_d^{max}}]$$

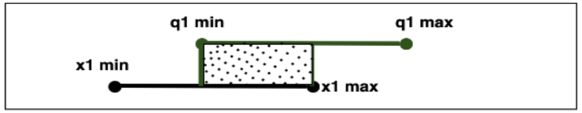
The **data overlapping** $\boldsymbol{h_k}$ **of cluster** $\boldsymbol{k}$ is estimated through the overlapping of the intervals of each dimension in cluster *k* with these of query $\boldsymbol{q_k}$ at node $\boldsymbol{n_k}$.

Data Overlapping per Cluster per dimension

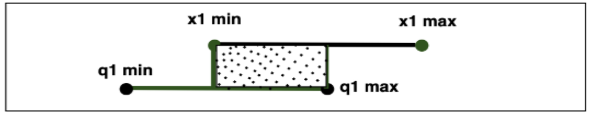
Five overlapping cases between cluster *k* and query q_k per dimension



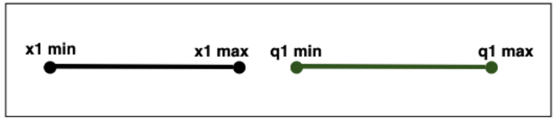
(a) Both of the query boundaries belong inside the cluster boundaries.



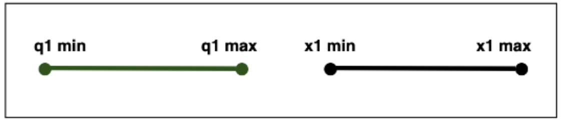
(b) Only the minimum boundary of the query boundaries belongs to cluster boundaries.



(c) Only the maximum boundary of query belongs to Cluster boundaries.

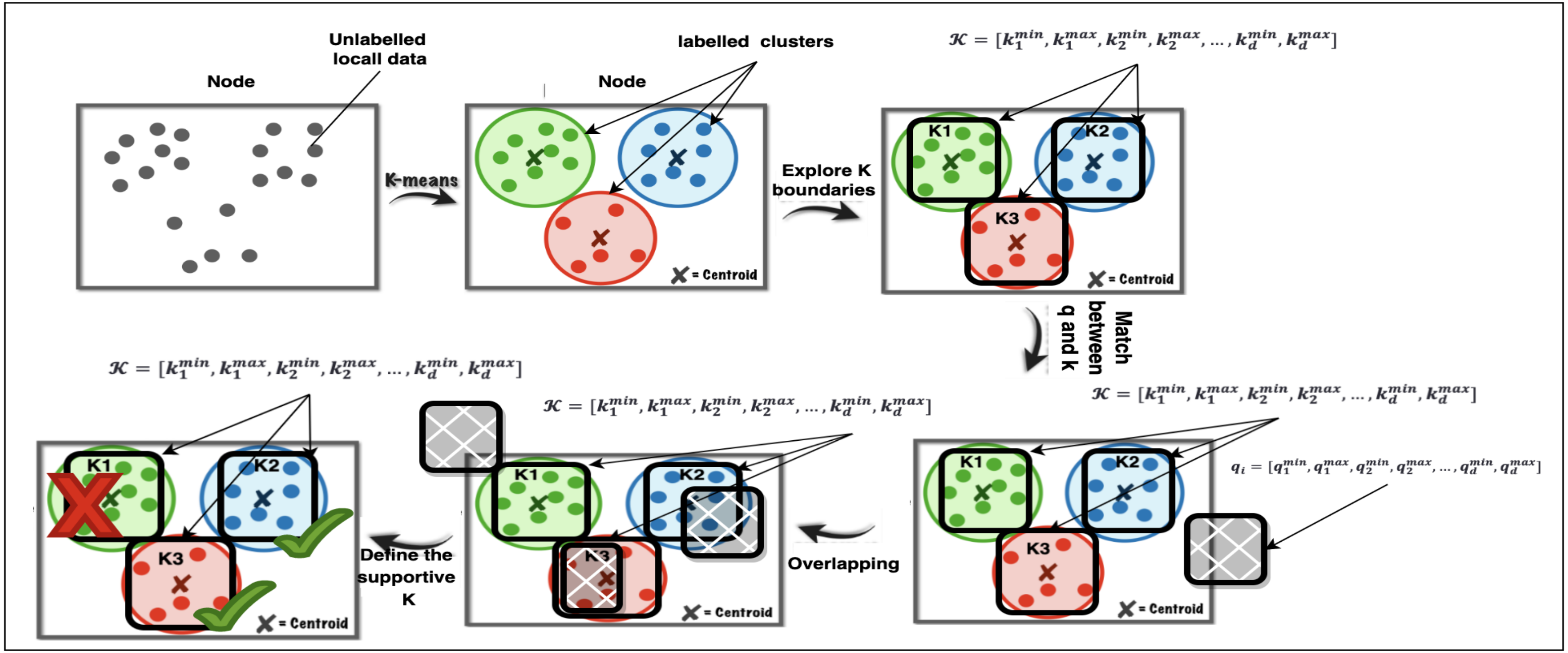


(d) Zero overlapping: both of cluster boundaries are lower than the query boundaries.



(e) (Opposite) Zero overlapping: both of the query boundaries are lower than cluster boulders.

Visualization


Unlabelled
locall data
Node
K-means
labelled clusters
Node
= Centroid
Explore K boundaries
$\mathcal{K} = [k_1^{min}, k_1^{max}, k_2^{min}, k_2^{max}, \dots, k_d^{min}, k_d^{max}]$
K1
K2
K3
= Centroid
Match between q and k
$\mathcal{K} = [k_1^{min}, k_1^{max}, k_2^{min}, k_2^{max}, \dots, k_d^{min}, k_d^{max}]$
$q_i = [q_1^{min}, q_1^{max}, q_2^{min}, q_2^{max}, \dots, q_d^{min}, q_d^{max}]$
Overlapping
Define the supportive K
= Centroid

Node's Potential & Ranking

Step 4: Define the **potential** $\boldsymbol{p_k}$ for node $\boldsymbol{n_k}$ according to the number of **supporting clusters** $\boldsymbol{K'} \leq \boldsymbol{K}$ whose overlapping $h_k \geq \epsilon$, i.e.,

$$p_k = \sum_{k=1}^{K'} h_k$$

Given the **potential** and the **supporting** clusters $\boldsymbol{K'}$, we define the **ranking** for node $\boldsymbol{n_k}$ w.r.t. query $\boldsymbol{q_k}$

$$r_k(q_k) = p_k \frac{K'}{K}$$

Step 5: The nodes **are sorted w.r.t. their rankings** $\{r_1, \dots, r_n\}$ w.r.t. query.

Leader node (with the highest rank) selects the subset of **the top-ℓ ranked nodes** to be acting as the participants for the query q.

Experimental Evaluation

Target: Set up a suitable environment for ML models requested to be built over **the data subspaces** as specified by the **incoming queries**.

We assess how **we correctly select the nodes to be engaged per query to build models over the right data.**

Dataset: Beijing Multi-Site Air-Quality Data.

N = 10 nodes. Node’s local data are collected from a different geographical region.

Number of Clusters: K=5 for all nodes to avoid biases.

Q = 200 queries randomly created over the **whole data space** (based on the dynamic query workload method described in [1])

Implementation: Keras to train local models **incrementally** on each node **over the supporting clusters' data per query**.

ML Models: Linear Regression (LR) and Neural Network (NN)

Learning Mechanisms Under Comparison

Assess our **top-ranked selected nodes** per query with:

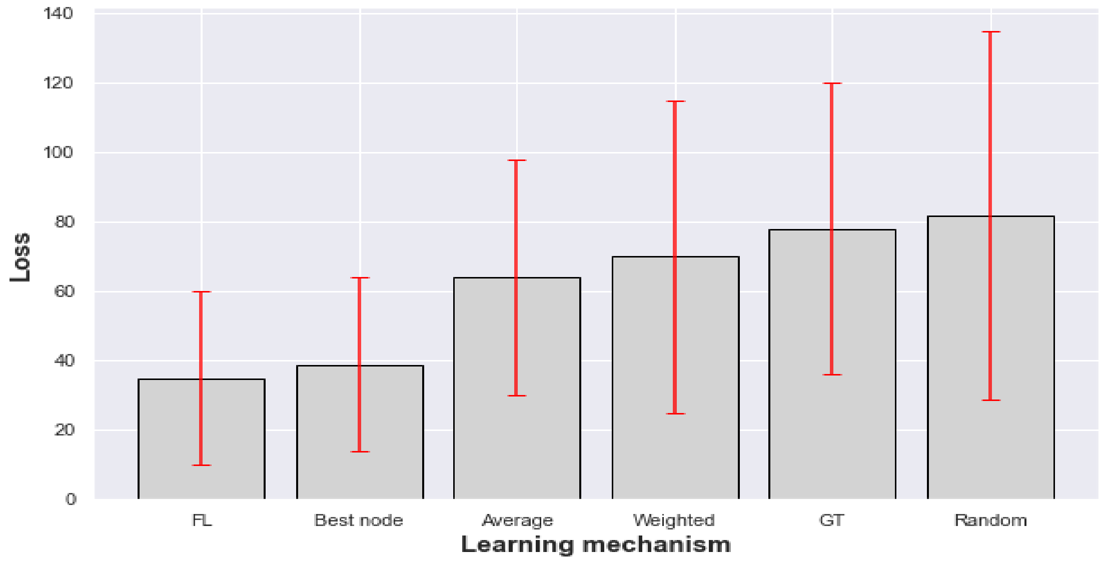
- **Model Averaging**: each **selected** node builds its own local model given a query. The final model is aggregated at the Leader node.
- **Weighted Averaging**: each **selected** node builds its own local model given a query. The final model is aggregated at the Leader node weighted by the nodes' ranking.

<u>Comparative Assessment</u>

- **Random selection**: a node (or a subset of nodes) are randomly selected per query.
- **Game Theory (GT) selection mechanism** [2]: nodes are selected based on their pre-trained models performances, i.e., models are built independently of the queries.

Results

ML Model Loss (over all queries)

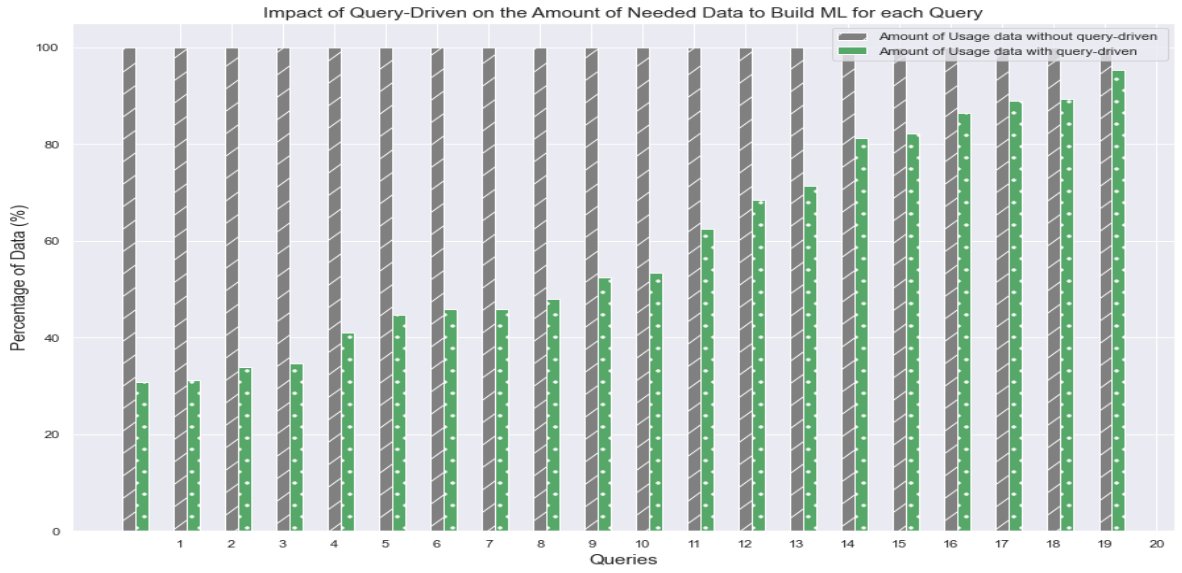


School *of* Computing Science

Knowledge & Data Engineering Systems

Results

Percentage/Amount of local data accessed per query



Conclusion

- ✓ We propose a **query-driven node selection mechanism** in distributed learning environments.
- ✓ We contributed with a mechanism **to determine the most appropriate subset of nodes** to be engaged in a model building **per analytics query**.
- ✓
- ✓ Our mechanism **minimizes the data needed by each participant** to train the model **only** over the query-driven supporting clusters' data.
- ✓ Our experimental results and comparative assessment showcase that our selection mechanism is deemed appropriate in distributed edge learning environments.

University of Glasgow

School *of* Computing Science
Knowledge & Data Engineering Systems

Thank you!