Federated Unlearning using Diffusive Noise Injection

Dr Muhammad Atif Tahir
Professor in Computer Science
Institute of Business Administration (IBA), Karachi

About me

- Professor, IBA, Karachi
- Director of Post Graduate and Graduate Program, IBA, Karachi

NED UNIVERSITY OF ENGINEERING & TECHNOLOGY
KARACHI

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
1963

QUEEN'S UNIVERSITY BELFAST

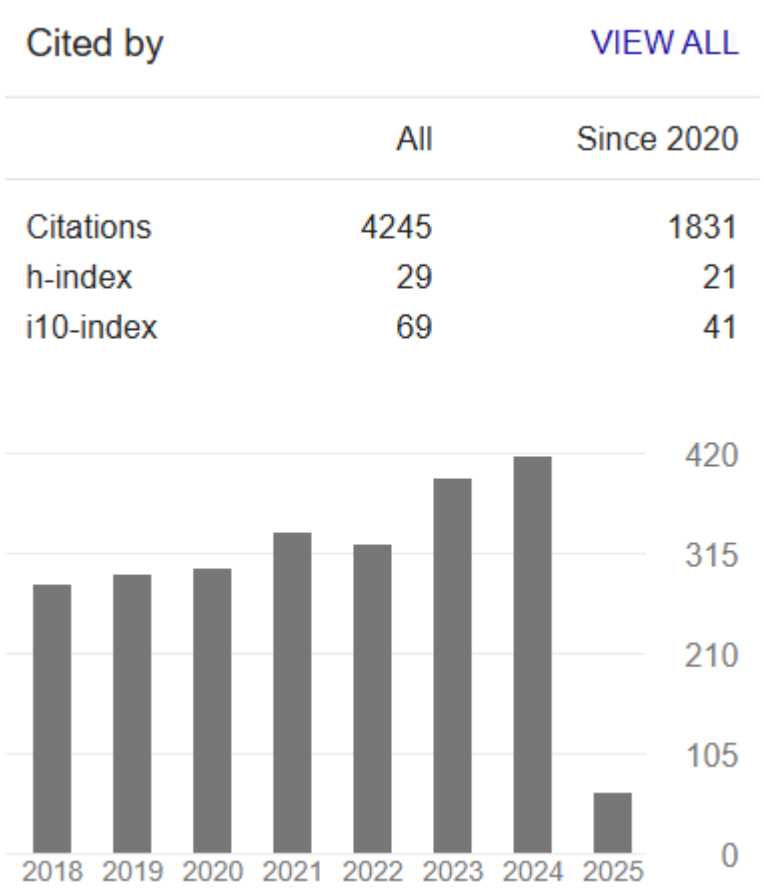
Summary of Research

- More than 100 High Quality Journal and Conference Publications

- W category (Platinum and Gold) with high impact factor including but not limited to
  - PAMI (IF: 24.3)
  - Pattern Recognition (IF 7.5)
  - Expert System and Applications (IF: 8.5)
  - Neural Computing & Applications (IF: 6.0)
  - Medical Image Analysis (IF: 11.3)

- More than 100 million of research funding

- Several Researchers under my supervision



Digital Footprint

What does it actually mean to delete your data in an AI-driven world?

Is pressing ‘delete’ enough or …….

Is that just an illusion?

Digital Footprint

But somewhere, deep inside a machine learning model, their data was still alive

Still influencing predictions, decisions… outcomes

This is where Federated Unlearning comes in; a new frontier in making AI systems truly forget

Part A

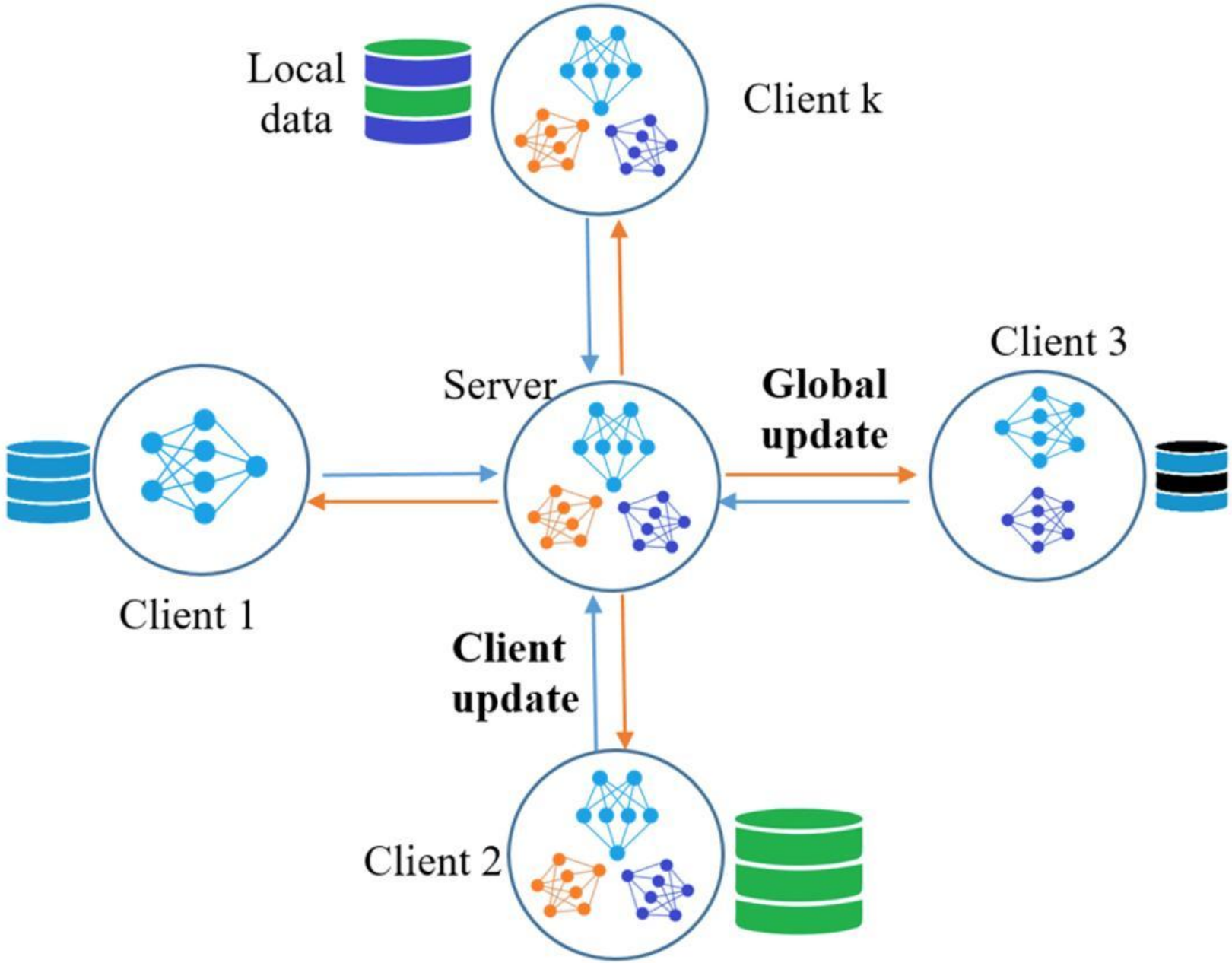
Federated Unlearning

Federated Learning

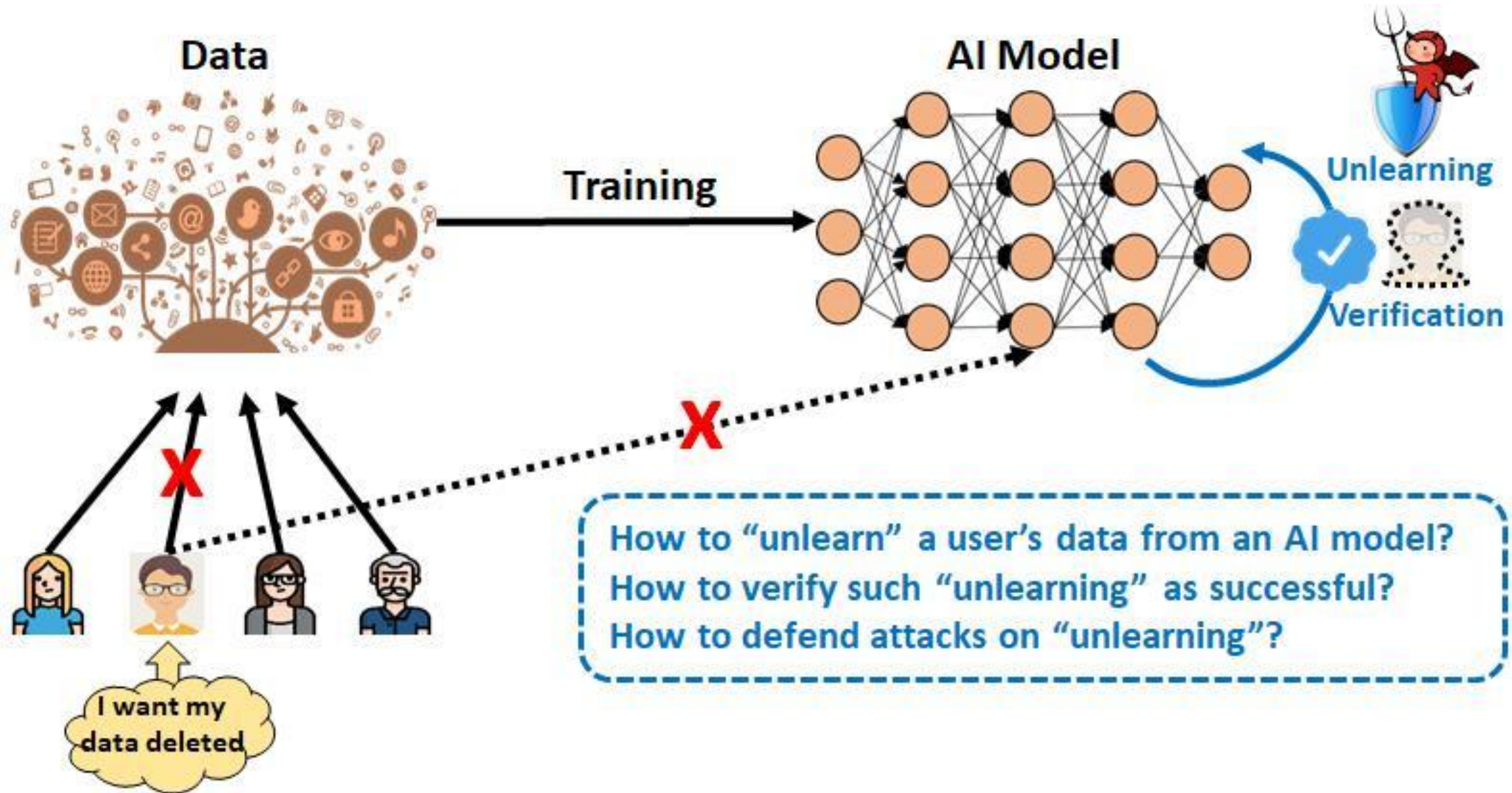
Key Idea: Learn from decentralized data without moving the data

Main Steps:

- ✓ Multiple clients (phones, hospitals, etc.) train a shared model locally
- ✓ Only model updates (gradients/weights) are sent to a central server
- ✓ The server aggregates updates (e.g., FedAvg)
- ✓ A global model is redistributed to clients


Local
data
Client k
Server
Global
update
Client 3
Client 1
Client
update
Client 2

Unlearning


Data
Training
AI Model
Unlearning
Verification
I want my data deleted
How to “unlearn” a user’s data from an AI model?
How to verify such “unlearning” as successful?
How to defend attacks on “unlearning”?

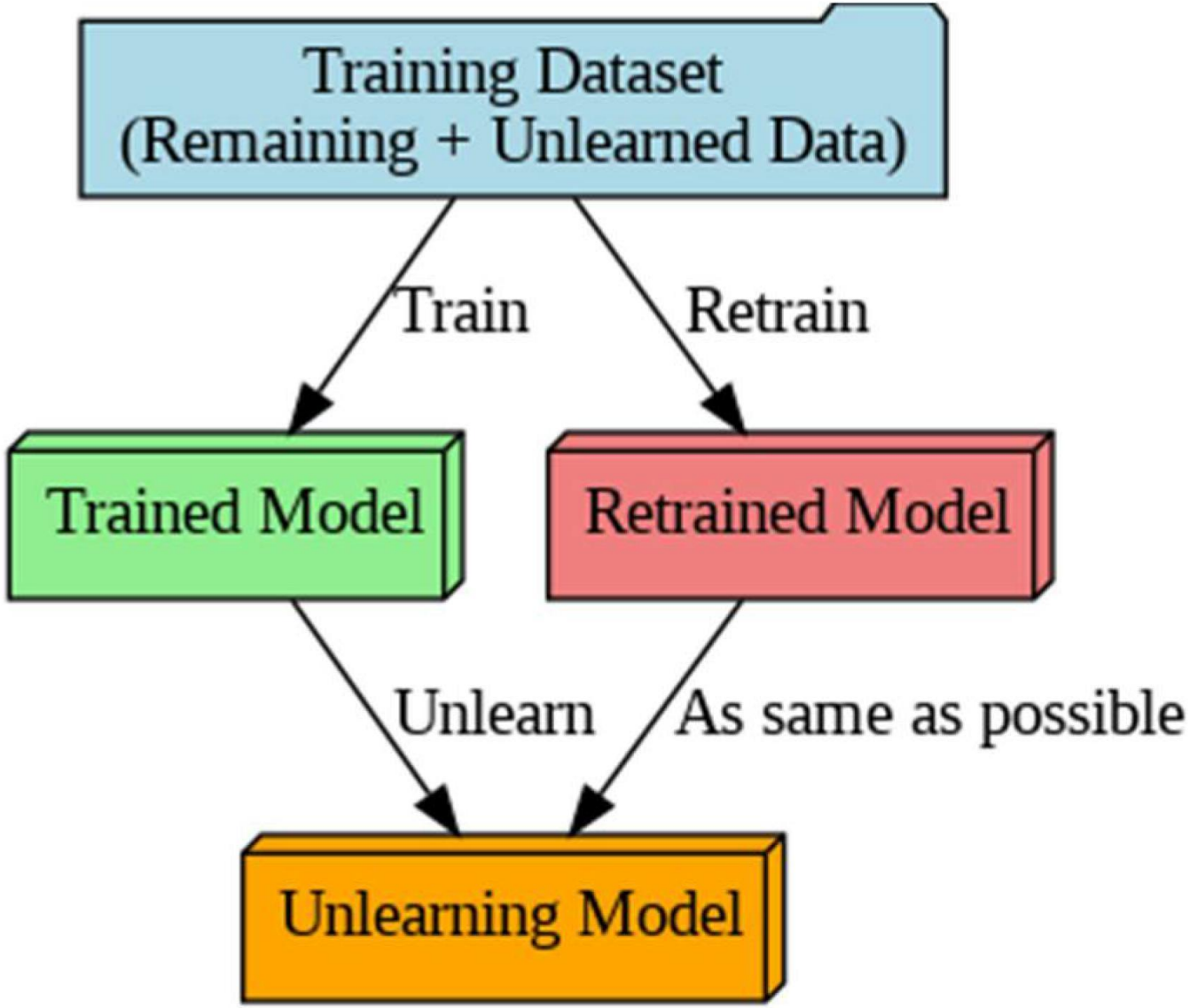
Federated Unlearning

Key Idea: Erase specific knowledge from a trained federated model

Main Steps:

- ✓ After training, a client (or its data) requests removal
- ✓ The system identifies that client's contribution to the global model
- ✓ It reverses or subtracts that influence (via retraining, approximation, or stored updates)
- ✓ A new global model is produced as if that client never participated

Federated Unlearning



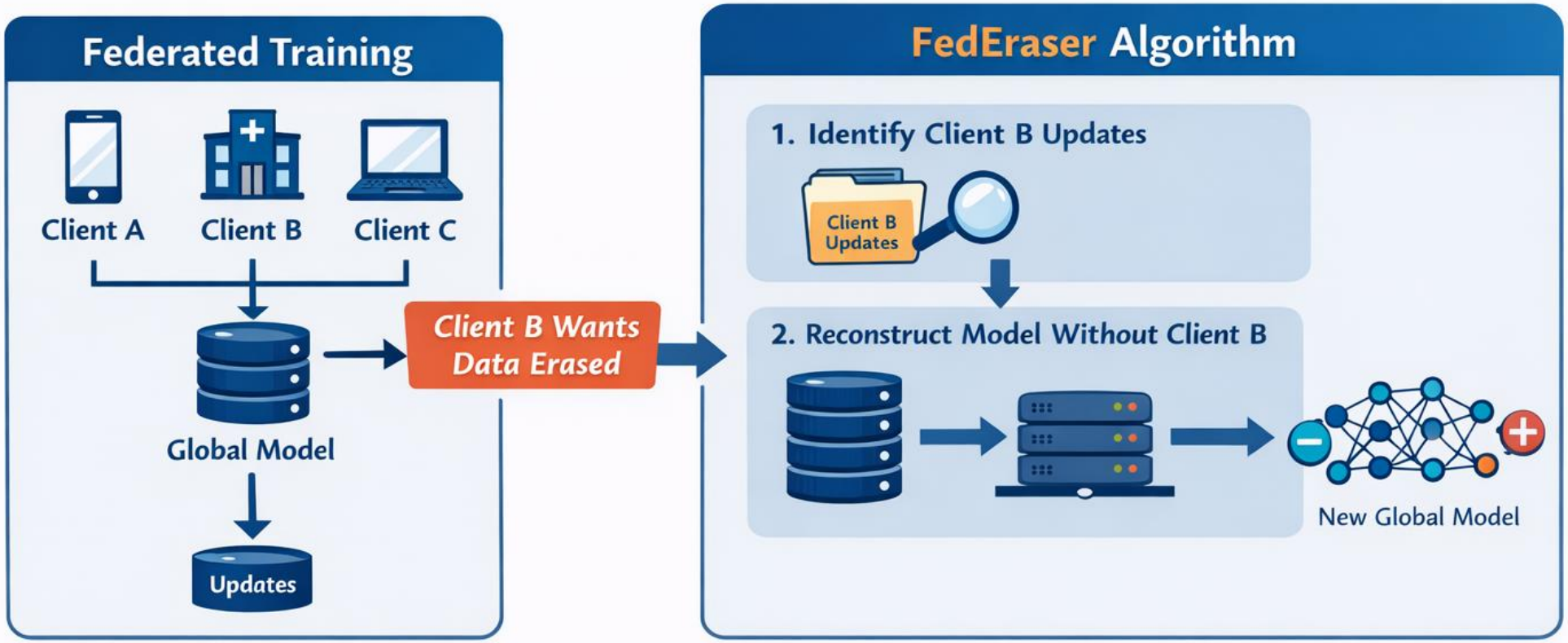
Part B

Key Research on Federated Unlearning

Algorithm #1: FedEraser

- ✓An early and foundational method in **federated unlearning**
- ✓Key Idea: Trade storage for speed
- ✓During training, the server stores historical updates (gradients/weights) from each client
- ✓When a client requests deletion:
  - The system reconstructs the model as if that client never participated
  - Uses stored updates instead of retraining from scratch

FedEraser: Federated Unlearning



Limitations

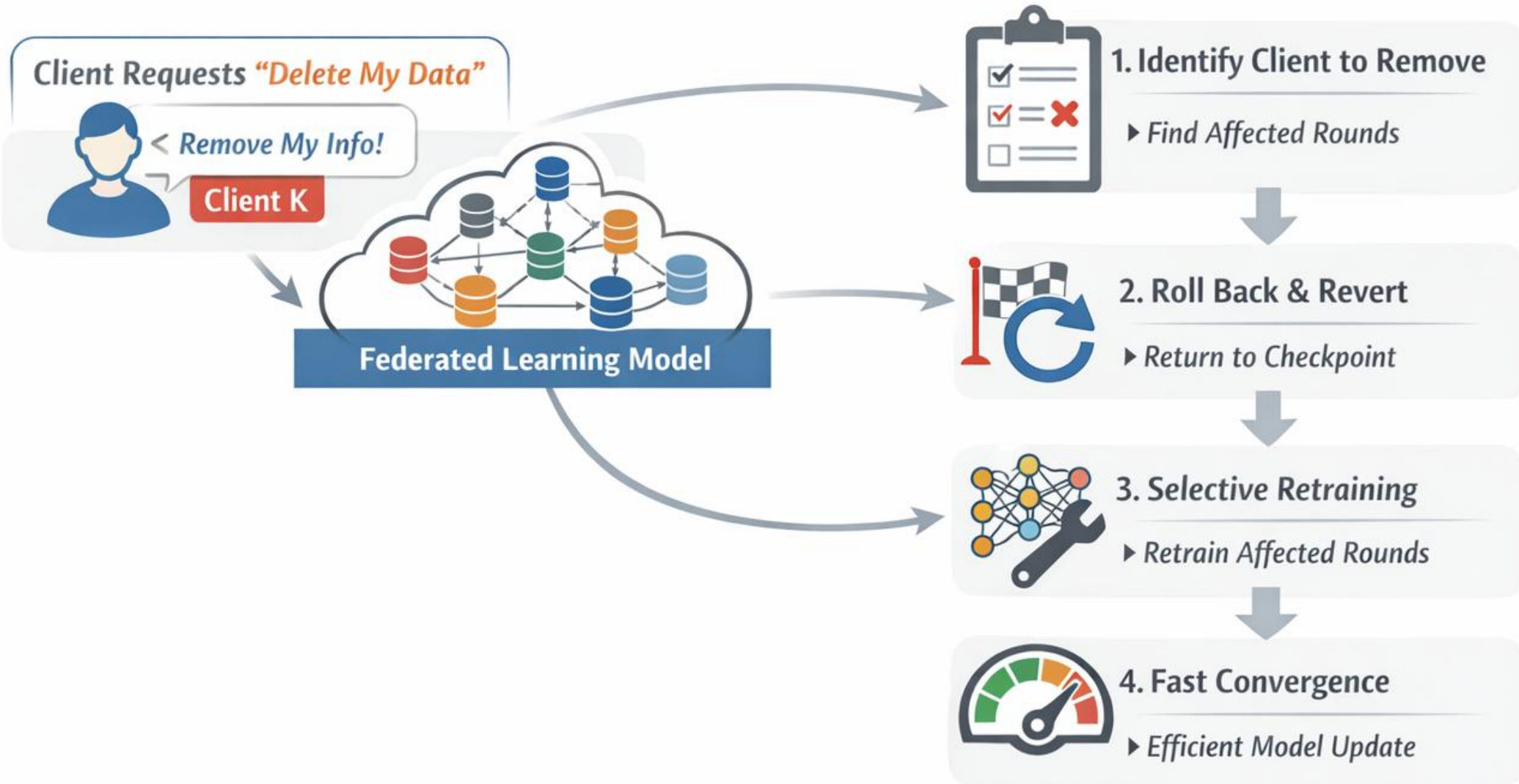
- Requires Storing Checkpoints
- Approximation Errors
- Non-IID & Large Scale Data

Algorithm #2: Forgotten in FL

- ✓Key Idea: Don't relearn everything; only fix the part affected by the deleted client
- ✓Introduces efficient unlearning via partial retraining
- ✓Reduces Training Time and Communication Cost
- ✓Cons:
  - ➢Still requires some retraining
  - ➢Needs checkpoint management

Efficient “Right to be Forgotten” in Federated Learning

Rapid Retraining Approach



Only Retrain What’s Needed!

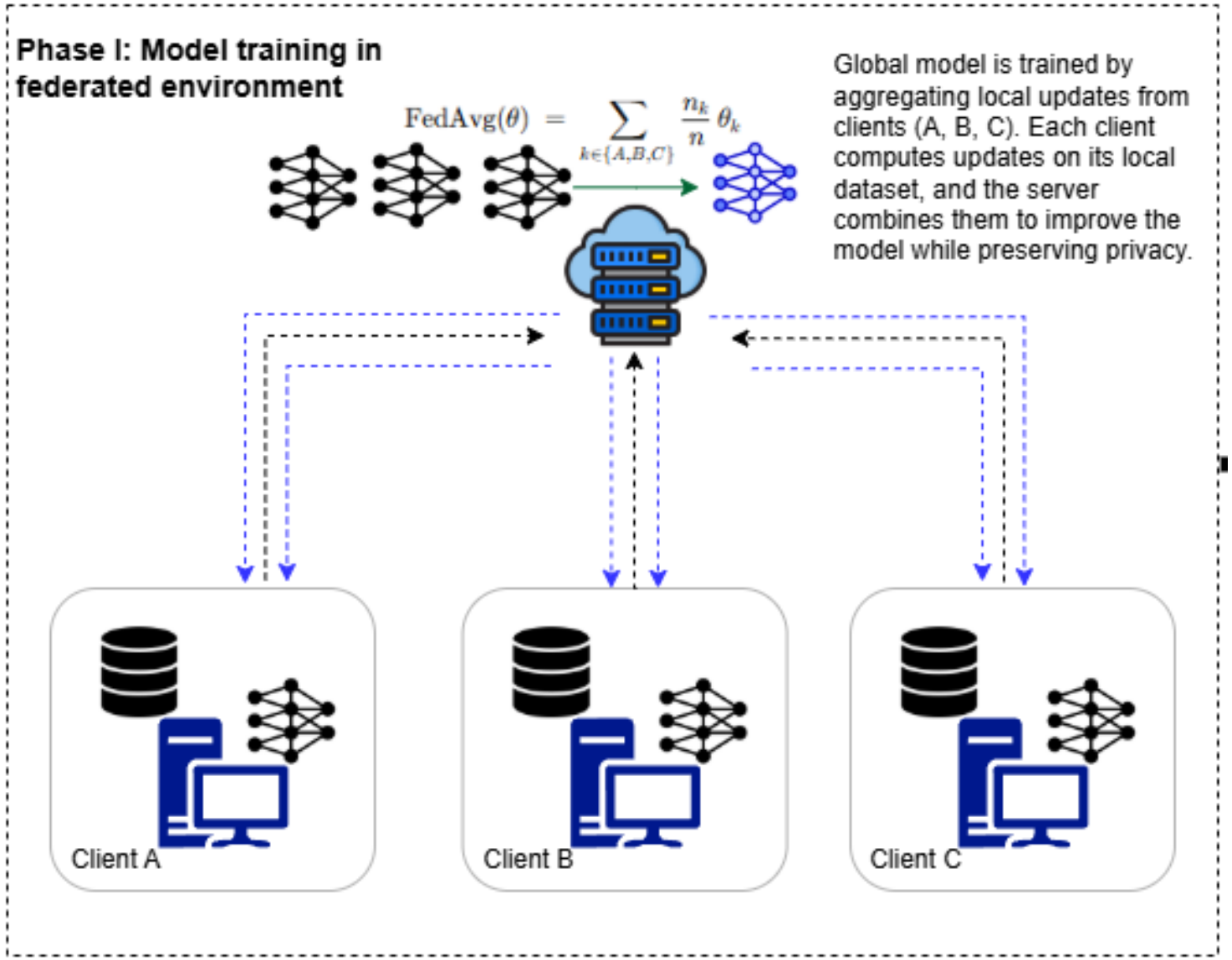
Part C

Our Contribution

Federated Unlearning using Diffusive Noise Injection

March 2026

Impact Factor: 15.5


Phase I: Model training in federated environment
$\text{FedAvg}(\theta) = \sum_{k \in \{A,B,C\}} \frac{n_k}{n} \theta_k$
Global model is trained by aggregating local updates from clients (A, B, C). Each client computes updates on its local dataset, and the server combines them to improve the model while preserving privacy.
Client A
Client B
Client C

Federated Unlearning

- Main Idea: Removes client-specific data influence without server-side retraining
- Client has three options

  ✓Quit the system and remove data
  ✓Remove all the data for a specific class
  ✓Remove selected data from a class

Federated Unlearning

- Three main steps

  ✓Selection of samples from the specified class or classes for removal (forget-set)

  ✓Generation of anti-samples through diffusive noise injection

  ✓Global Healing

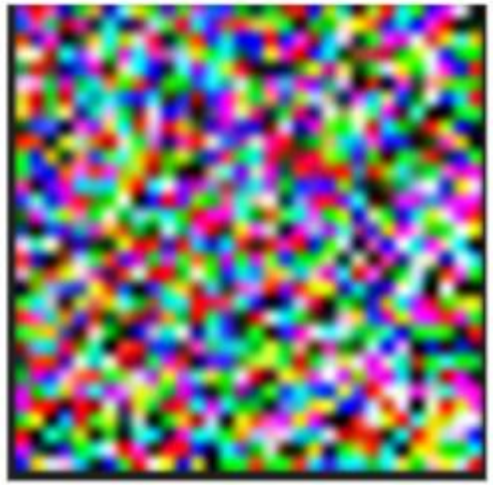
Generation of Anti-Samples

- The generation of anti-samples aims to maximize the loss for the forget-set, effectively overwriting the model's memory of the corresponding class
- A learnable noise matrix N is injected directly into the input
- The following objective function is optimized

$$\left[-\mathcal{L}(f(\mathbf{x}_0 + N), y) + \lambda \|N\|_2^2\right]$$

- Here loss function is cross entropy loss, N is injected noise matrix

Generation of Anti-Samples



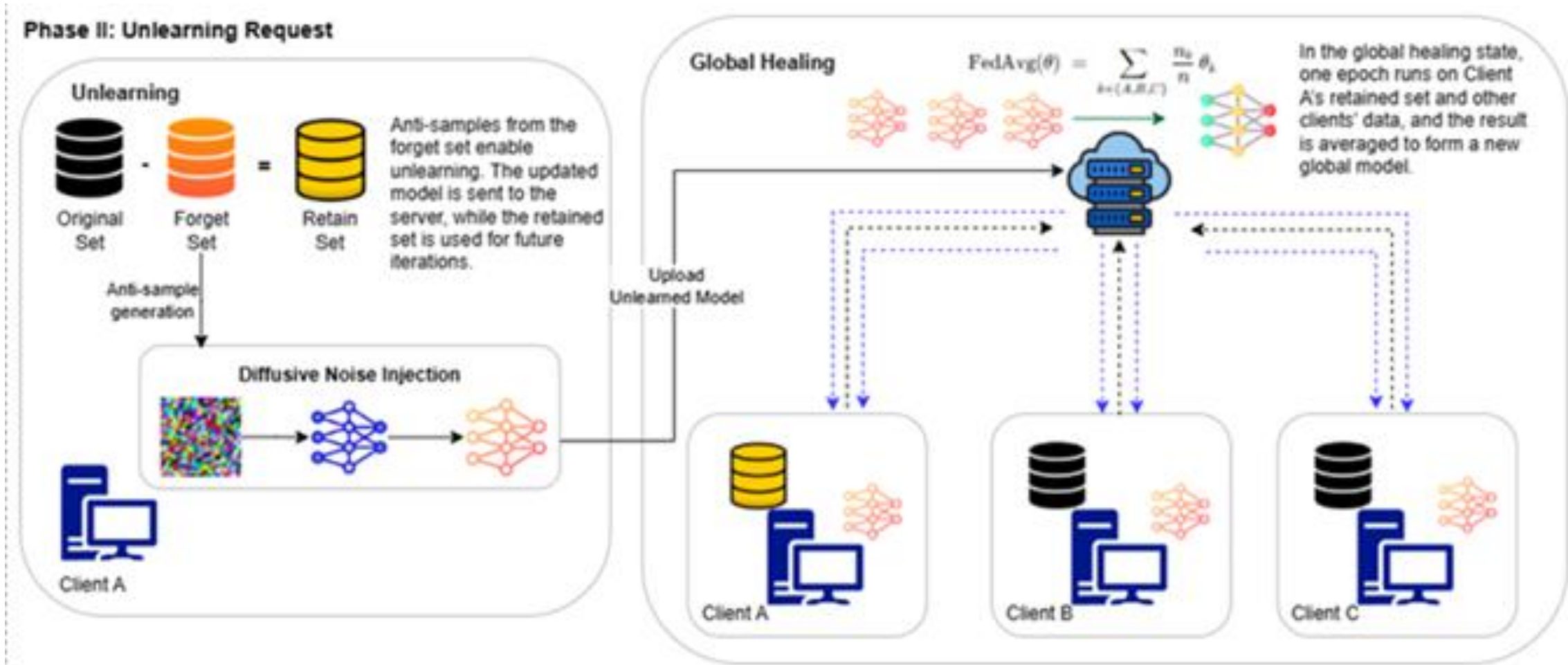
Global Healing

- Three main steps

  ✓The newly unlearned model is distributed to all clients

  ✓Each client performs one training round on its retained local dataset

  ✓The updated models are aggregated via averaging, producing a healed global model


Phase II: Unlearning Request
Unlearning
Original Set
Forget Set
Retain Set
Anti-samples from the forget set enable unlearning. The updated model is sent to the server, while the retained set is used for future iterations.
Anti-sample generation
Diffusive Noise Injection
Client A
Upload Unlearned Model
Global Healing
$\mathrm{FedAvg}(\theta) = \sum_{k=\{A,B,C\}} \frac{n_k}{n} \theta_k$
In the global healing state, one epoch runs on Client A's retained set and other clients' data, and the result is averaged to form a new global model.
Client A
Client B
Client C

Experimental Setup

Dataset	DataPoints / Classes	Train / Test Split
CIFAR10	60000 / 10	50000 / 10000
CIFAR100	60000 / 100	50000 / 10000
MINST	70000 / 10	60000 / 10000
KVASIR	8000 / 8	6400 / 1600

Evaluation Measures	Summary
Accuracy on Forget Set	Close to Zero indicating that the model has effectively forgotten the specified data
Accuracy on Retain Set	Close to the performance of the original model
Time	This metric serves as a proxy for measuring the amount of time taken for the unlearning process

Class Unlearning Evaluation on CIFAR-10 with ResNet18.

Class	Init. Acc.	6 Rnds.	Unlearn C1	Heal	Retrain
airplane	0.437	0.958	0.020	0.2500	0.280
automobile	0.455	0.812	0.000	0.1500	0.260
bird	0.401	0.804	0.752	0.8125	0.820
cat	0.383	0.916	0.770	0.9375	0.920
deer	0.254	0.979	0.791	1.0000	0.989
dog	0.159	0.895	0.895	0.9583	0.950
frog	0.439	0.979	0.812	0.9583	0.979
horse	0.440	0.875	0.937	0.9375	0.950
ship	0.580	0.895	0.958	0.9792	1.000
truck	0.490	0.895	0.916	0.9375	0.950

Results on CIFAR 100 dataset

- Accuracy using FL on whole dataset was 87.0 ± 0.18 using ResNET 18

5 clients	Time (Minutes)	Accuracy on Forget Set	Accuracy on Retain Set
Federated	70	-	86.0 ± 0.2
Retrain	65	15.0 ± 0.05	87.5 ± 1.1
Proposed	15	16.5 ± 0.9	88.0 ± 1.0

Results on MINST dataset

- Accuracy using FL on whole dataset was 95.0 using ViT B16

5 clients	Time (Minutes)	Accuracy on Forget Set	Accuracy on Retain Set
Federated	145	-	96.0
Retrain	130	25.5	93.5
Proposed	45	25.0	95.0

Results on KVASIR dataset

- Accuracy using FL on whole dataset was 82.0 using ResNET 18

5 clients	Time (Minutes)	Accuracy on Forget Set	Accuracy on Retain Set
Federated	90	-	83.0
Retrain	85	11.0	83.5
Proposed	20	12.5	82.5

Comparison with State of the Art

Paper	Comasison to Retriaing
FedEraser	4x Faster, 39% hit in accuracy
RFUL	5x Faster, 9% hit in accuracy
FedWiper	12x Faster, 3% hit in accuracy
FedCF	1.5x Faster, 34% hit in accuracy

Conclusion

- The proposed approach facilitates unlearning within a federated setup by decentralizing the unlearning process to the client side
- Proposed system eliminates the need for the global server to store client instances
- Proposed method enables selective removal of client data, such as specific records or classes, and does not assume that a client will be removed completely
- This makes it practical for real world federated learning scenarios

Thank you

atiftahir@iba.edu.pk