A PROJECT REPORT
on

"ENERGY AWARE: DECENTRALISED AGGREGATION MECHANISMS IN FEDERATED LEARNING"

Submitted to
KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

SREEJA SANYAL	**22051200**
SAKET KUMAR	**22054212**
ADITYA MAJUMDAR	**22054214**
ANANYA BISWAL	**2205874**

UNDER THE GUIDANCE
OF
PROFF. JHALAK HOTA

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024
April 2025

KIIT Deemed to be University
School of Computer Engineering
Bhubaneswar, ODISHA
751024

CERTIFICATE

This is certify that the project entitled
"ENERGY AWARE: DECENTRALISED AGGREGATION MECHANISMS IN FEDERATED LEARNING"

submitted by

SREEJA SANYAL	**22051200**
SAKET KUMAR	**22054212**
ADITYA MAJUMDAR	**22054214**
ANANYA BISWAL	**2205874**

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 09/04/2025

PROFF. JHALAK HOTA
(Project Guide)

Acknowledgement

We are profoundly grateful to Dr. Jhalak Hota for his expert guidance and constant support throughout, without which the completion of this project would not have been possible. His continuous encouragement has helped us in completing the project in the provided time.

SREEJA SANYAL
SAKET KUMAR
ADITYA MAJUMDAR
ANANYA BISWAL

ABSTRACT

This project presents an energy-aware approach to decentralized aggregation in Federated Learning (FL), aiming to enhance privacy, scalability, and efficiency in distributed machine learning. Traditional FL relies on a central server for model aggregation, which introduces risks such as single points of failure, latency, and energy inefficiency. Our work explores decentralized FL architectures where model updates are exchanged directly among clients through peer-to-peer communication.

We investigated multiple decentralized strategies—such as gossip-based protocols, blockchain consensus, and hierarchical clustering—with a focus on optimizing energy consumption. Techniques like adaptive client participation, energy-efficient node selection, and model sparsification were applied to reduce communication overhead. Additionally, we examined challenges such as synchronization issues, security threats, and lack of incentive mechanisms.

Our findings suggest that decentralized, energy-aware FL can offer a robust and scalable alternative to centralized systems, particularly for resource-constrained and privacy-sensitive environments. This work lays the foundation for future research into intelligent coordination and sustainable distributed learning.

Keywords: Federated Learning (FL), Decentralized Aggregation, Energy Efficiency, Gossip Protocols, Peer-to-Peer Learning, Edge Computing, Model Sparsification, Adaptive Participation, Privacy Preservation, Distributed Machine Learning, Client Selection.

Contents

Chapter 1: Introduction

1.1 WHAT IS FEDERATED LEARNING?

Federated Learning is defined as a distributed machine learning technique where a centralized server manages the training process across multiple decentralized devices or servers. In this framework, each client trains the same model architecture using only their local data, ensuring that raw information remains at its source and is never exchanged. Rather than sharing datasets, clients send model updates—such as parameters and gradients—to the central server. This server then aggregates these updates to enhance the performance of a global model, which is subsequently redistributed to all clients for the next round of training.
Decentralized Federated Learning goes beyond this conventional method by eliminating the need for a central coordinating server. Instead, it enables peer-to-peer communication among participating devices, which collectively improve the model through consensus-based mechanisms.
Compared to traditional centralized learning techniques, this fully decentralized approach enhances privacy protection and system resilience. By removing single points of failure and minimizing data transfer, it supports collaborative model development that maintains data confidentiality and consumes less bandwidth.

1.2 TYPES OF FEDERATED LEARNING

1.2.1 Based on distribution of data

Federated Learning is broadly classified into three types based on the distribution of data across devices or organization
1. Horizontal Federated Learning
2. Vertical Federated Learning
3. Federated Transfer Learning

1. **Horizontal Federated Learning:** Also known as s**ample based federated learning** .When data from various devices represent different users or samples (i.e., different rows), but share the same feature space (i.e., the same types of attributes or columns), horizontal federated learning is employed.
   For example a number of banks located in various areas that all keep client data with comparable characteristics, such as name, age, income, and credit score. They can work together to train a model using horizontal federated learning without sharing raw data because the data format is similar but the customers are different.

2. **Vertical Federated Learning:** Also known as **feature based federated learning**.When data from different parties refers to the same set of users (i.e., the same rows) but contains

different kinds of features or attributes (i.e., different columns), vertical federated learning is applicable.

For example when organisations such as a hospital and an insurance provider serve the same customers. While the insurance company has financial information, the hospital may have medical records. They can work together to create a model that integrates their knowledge without disclosing each other's private information by utilising vertical federated learning

3. **Federated Transfer Learning:** It is used when there is little to no overlap in the samples and features of the parties. Here knowledge is transferred from one domain to another using transfer learning techniques. For example, even though a healthcare provider in Europe and a retail company in Asia may have completely different datasets, they can still benefit from sharing high-level model representations or embeddings.

   This kind is helpful when cooperation is required even when the data is not very similar.

1.2.2 Based on Architecture

In addition federated learning systems can also differ based on how they are architecturally coordinated
1. Centralized Federated Learning
2. Decentralized Federated Learning

1. **Centralized Federated learning:** This process is managed by a central server. All participating devices or organisations (referred to as clients) receive the initial global model from the server. After using its own data to train the model locally, each client sends the updated model parameters and not the raw data back to the central server. The server then aggregates the updates to create a better global model, typically using an algorithm such as Federated Averaging (FedAvg). The process then iteratively continues with the redistribution of this updated model. Due to its ease of implementation and management, centralised FL is widely used. It does, however, come with the risk of a single point of failure and privacy issues in the event that the server is compromised.

2. **Decentralised federated learning:** It does not require a central server. Instead, there is peer-to-peer direct communication between devices or institutions. They might use pre-established communication patterns, such as a fully connected mesh, ring topology, or gossip protocols, to exchange model updates. Additionally, the clients work together to aggregate model updates. In addition to providing better fault tolerance and possibly increased privacy, this architecture increases robustness and lessens reliance on a central coordinator. Nevertheless, it adds complexity to the model's synchronisation, coordination, and convergence.

In conclusion, centralised and decentralised setups describe how the learning is coordinated among participants, whereas horizontal, vertical, and transfer federated learning describe how data is divided.

1.3 HOW IS FEDERATED LEARNING DIFFERENT FROM TRADITIONAL LEARNING APPROACH?

Criteria	Federated Learning	Traditional Learning
Data Collection	All user/device data is collected and transferred to a cebtral server	Data remains on user's sevice or local servers.Only model updates are shared.
Model Training Location	Model is trained centrally using all collected data	Model is trained locally on edge devices or decentralized nodes.
Data privacy	High risk of data leakage as raw data is transmitted and stored centrally	Enhanced privacy as the raw data never leaves the device and only gradients or parameters are shared.
Security	Vulnerable to data breaches especially with sensitove data like medical or financial records.	More secire by design especially when combined with techniques like differential privacy and secure aggregation.
Data Distribution	Assumes Independent and Identically distributed data which is often unrealistic	Handles nin-IID data(like users having different writing styles,photo etc)
Computation	High performance central servers or cloud are needed	Computation is offloaded to edge devices
Real life example	Google photos uploads all your pictures to the cloud to improve image recognition	Google keyboard learns your typing style locally to suggest better next words without uploading your messages

1.4 CENTRALISED vs DECENTRALISED FL

Features	Centralised FL	Decentralised FL
Architecture	Has a central server that maintains the model updates	There is no central server and all nodes communicate peer-to-peer
Model Aggregation	Central server collects and aggregates model updates from the clients	Clients themselves exchange updates and perform aggregation
Communication	One to many	Many to Many

Fault tolerance	Central point of failure as if the server goes down the whole process stops	More fault tolerant as there is no single point of failure
Scalability	Easier to scale with proper server infrastructure	More complex to scale due to peer to peer coordination
Security/Privacy	Data stays local but central server can be a risk	Better privacy but more complex to secure peer communication
Energy usage	Generally lower energy consumption due to coordinated communication and aggregation	Typically higher energy consumption due to increased peer to peer communicating and local processing

1.5 AGGREGATION IN FEDERATED LEARNING

1.5.1 What is aggregation in FL?

Aggregation in Federated Learning (FL) refers to the process of combining local model updates (parameters or gradients) from multiple distributed clients to produce an improved global model. This step is central to FL, as it enables learning from decentralized data while preserving privacy.

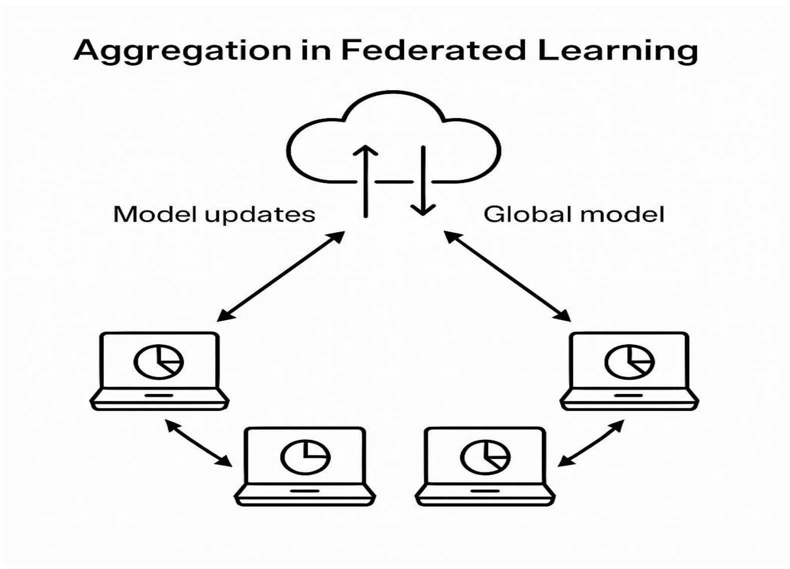

Fig 1.1: Aggregation in FL

1.5.2 Key Points

Process Overview:

- Each client trains a model using local data.
- The model's parameters or gradients are sent to an aggregator.
- The aggregator combines these updates to refine the global model.
- The updated model is sent back to clients for the next training round.

Common Aggregation Algorithms:

- **Federated Averaging (FedAvg)**: The most widely used method; averages local model weights weighted by dataset size.
- **FedProx**: An extension of FedAvg that adds a proximal term to handle data heterogeneity.
- **Krum / Bulyan**: Robust aggregation methods that resist Byzantine or malicious client updates.

Centralized vs. Decentralized Aggregation:

- **Centralized**: A server performs aggregation. Simple and efficient but vulnerable to failure and bottlenecks.
- **Decentralized**: Clients aggregate updates through peer-to-peer protocols (e.g., gossip, blockchain), improving fault tolerance and privacy.

Energy and Bandwidth Efficiency:

- Aggregation frequency and communication cost are key concerns.
- Techniques like sparse updates and quantization are used to reduce overhead.

Security and Trust Issues:

- Aggregation is vulnerable to **poisoning attacks** or **malicious updates**.
- Defense mechanisms include update validation, anomaly detection, and secure multiparty computation.

1.5.3 Aggregation Algorithms

1. Federated Averaging (FedAvg)

Overview: Federated Averaging (FedAvg) is the most foundational aggregation algorithm in Federated Learning (FL). Introduced by Google in 2017, FedAvg facilitates collaborative training without data centralization. It works by averaging model weights from multiple clients trained on their private datasets.

Steps:

1. The server initializes a global model.
2. In each round, it selects a subset of clients.
3. Each client trains the model on its local data for a few epochs.
4. The clients send updated model parameters to the server.
5. The server performs a weighted average of the client updates, where weights are proportional to the number of samples on each client.
6. The updated global model is sent back to clients for the next round.

Mathematical Formula: $\omega_t = \sum_{k=1}^{K} \frac{n_k}{n}\omega_t^k$

Where:

- ω_t^k : local model parameters from client **k**
- n_k : number of samples on client
- $n = \sum_k n_k$: total number of samples across selected clients

Advantages:

- Easy to implement
- Efficient in communication rounds
- Preserves user privacy by not sharing raw data

Limitations:

- Performance degrades under non-IID data distribution
- Client drift can occur with heterogeneous data

2. Federated Proximal (FedProx)

Overview: FedProx is an extension of FedAvg designed to handle system and statistical heterogeneity across clients. It adds a proximal term to the loss function that discourages local models from drifting too far from the global model.

Modified Objective Function: $min_{\omega} f_k(\omega) + \frac{\mu}{2}||\omega - \omega_t||^2$

Where:

- $f_k(\omega)$: local loss function on client **k**
- ω_t : current global model
- μ : regularization coefficient

Steps:

1. Server sends current global model ω_t to selected clients.
2. Clients perform local training with a proximal term.
3. Clients return the updated models to the server.
4. Server aggregates the models as in FedAvg.

Advantages:

- Improved stability for non-IID and unbalanced data
- Mitigates client drift

Limitations:

- Higher computational cost per client
- Sensitive to the value of μ

3. Krum Algorithm

Overview: Krum is a robust aggregation algorithm designed to protect against Byzantine failures, where some clients may send incorrect or malicious updates. It selects one client's update based on its similarity to others, assuming that most clients are honest.

Mechanism:

1. Each client sends its model update to the server.
2. For each update, compute the Euclidean distances to all other updates.
3. For each client, compute the sum of distances to its closest **n -f - 2** updates (where **f** is the number of possible Byzantine clients).
4. Select the update with the smallest total distance.
5. Use this update (or average a few selected ones in Multi-Krum) to update the global model.

Advantages:

- High robustness against adversarial behavior
- Does not require identifying which clients are malicious

Limitations:

- Computationally expensive for large numbers of clients
- Not communication-efficient for real-time FL scenarios

Chapter 2: Decentralised Federated Learning

2.1 WHAT IS DECENTRALISED FEDERATED LEARNING?

In federated learning, decentralized aggregation refers to the process where multiple client devices collaboratively train a global model without relying on a central server. Instead of sending local model updates to a central aggregator, each client shares updates directly with its peers. This peer-to-peer communication allows for the collective aggregation of model parameters, facilitating the training of a global model in a decentralized manner. Such an approach enhances privacy and reduces the risk of a single point of failure. (Source)
Federated learning was first proposed by Google researchers in 2016 and was applied to build a language model collaboration framework on Google Keyboard to learn whether people clicked on recommended suggestions and contextual information. In 2020, Google researchers expanded the concept of FL to federated analytics, extending from learning tasks to collaborative computing, data analysis, and inference, further deploying it within Google Keyboard. FL has demonstrated its excellent capabilities in various areas, including intelligent transportation, internet of things (IoT), healthcare, manufacturing, agriculture, energy, remote sensing, and more. FL also breaks geographical limitations allowing efficient collaboration worldwide.

2.2 HOW IT WORKS?

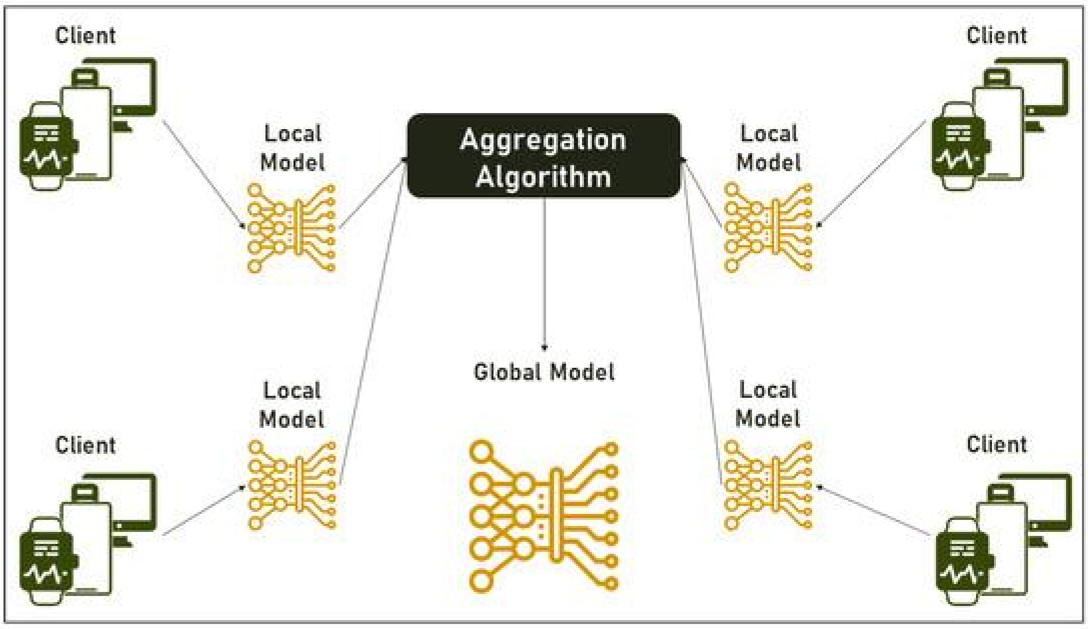

fig 2.1: Working of decentralised federated learning

1. Local Model Training (Client Side):

- Data Stays Local: Each client (e.g., your smartphone, smartwatch) has its own private dataset. This data never leaves the device.
- Model Initialization: A base machine learning model is distributed to each client from a central server or another client.
- Local Training: Each client trains the model on its local data. This involves updating the model's parameters (weights and biases) to improve its performance on the local data.
- Trained Model Updates: Instead of sending the raw data, each client only sends the updates to the model (i.e., the changes made to the model's parameters) to the central server. These updates are typically much smaller than the original dataset, preserving privacy.

2. Aggregation (Server Side):

- Receiving Updates: The central server (or a designated aggregator) receives the model updates from all participating clients.
- Aggregation Algorithm: The server uses an aggregation algorithm to combine these updates. A common algorithm is Federated Averaging (FedAvg), which calculates the weighted average of the model updates.
- Global Model Update: The aggregated updates are used to update the global model. This global model represents a collective understanding learned from all the clients' data.

3. Global Model Distribution (Server to Clients):

- Updated Global Model: The updated global model is then sent back to the clients.
- Local Model Update: Clients can either replace their local model with the new global model or use it to further refine their local model.

4. Iterative Process:

- Repeated Rounds: This process of local training, aggregation, and global model distribution is repeated for multiple rounds.
- Improved Global Model: With each round, the global model becomes more accurate as it learns from the collective data of all clients.

Key Aspects Highlighted in the Image:

- Decentralization: The image emphasizes the distributed nature of Federated Learning, with each client having its own local model and data.
- Privacy Preservation: The image implies that the raw data never leaves the client devices, as only model updates are shared.

- Aggregation: The central "Aggregation Algorithm" highlights the crucial role of combining model updates to create a global model.

2.3 KEY ISSUES

1. **High Communication Overhead**

The absence of a central server leads to direct peer-to-peer communication, which can significantly increase the volume and frequency of data transmission across the network. Reasons for high communication overhead include:

- **Peer-to-Peer Model Exchanges**
  Unlike centralized FL (where clients send updates to a single server), DFL clients must exchange model updates with multiple peers in each round—sometimes with all their neighbors. This increases the number of transmissions per round.

- **Redundancy in Model Updates**
  Without centralized filtering, the same model parameters may be exchanged repeatedly across paths, causing redundant communication and slower propagation efficiency.

- **Dense Communication Topologies**
  Topologies like fully connected graphs or dynamic peer networks result in exponential growth in communication as the number of clients increases.

- **Frequent Synchronization**
  Many DFL algorithms require frequent synchronization rounds to ensure convergence, especially in consensus-based or gossip protocols, increasing cumulative bandwidth demands.

- **Large Model Sizes**
  Modern machine learning models—particularly in deep learning—can have millions of parameters, making each peer-to-peer transmission costly in terms of bytes.

2. **Computational and Storage Burden**

Compared to the CFL, the Aggregate paradigm imposes significantly higher demands on client-side computational and storage resources. As there is no dedicated server in the Aggregate paradigm, clients are responsible for storing previous model parameters and performing aggregation computations alongside local model training. Consequently, the computational and storage burdens pose challenges for client hardware.

3. **Vulnerability in Cybersecurity**

Network security has always been a major challenge in FL, and this challenge is particularly prominent in DFL. In the traditional CFL setting, clients communicate with a central server, typically operated by a research institution or a large commercial organization. While there is still potential for attacks and data poisoning between clients and the server, communication is generally more regulated and protected compared to DFL. In DFL, the knowledge exchange occurs directly among users within a local area network, with free and unrestricted sharing agreements, which poses an increased risk of privacy exposure. Malicious attacks from clients, poisoned data, free-riding attacks, and other malicious behaviors are all possible in this decentralized setting.

4. Lack of Incentive Mechanism

In the absence of server management, the issue of fairness in aggregation has been effectively addressed in DFL. However, the lack of incentives and mutual distrust among clients can significantly impact their willingness to contribute knowledge. A key issue is the lack of incentives, which may lead to freeriding attacks where clients choose to benefit from the models without contributing their own knowledge.

5. Lack of Management

In DFL, the absence of a central server for managing all clients poses a significant challenge in receiving and sharing knowledge in an organized manner. The lack of central management can lead to confusion, particularly among clients with varying sample sizes, computational resources, and communication capabilities. In the ring variant, a client only needs to wait for the model parameters from the previous client, while in the mesh variant, a client needs to wait for model parameters from all other clients. Such dependencies on other clients for model transmission can result in deadlocks, causing the entire system to halt. Moreover, the communication among clients may not be robust, considering the possibility of SPoF. The lack of management can lead to reduced operational efficiency, confusion regarding model versions, and performance degradation.

Chapter 3: Future Work & Conclusion

3.1 FUTURE WORK

In order to solve the challenges indicated above, there are some possible future work directions worth studying:

3.1.1 Privacy restrictions

In fact, due to the heterogeneity of various devices in the network, their privacy restrictions have their own different characteristics, so it is necessary to define the privacy restrictions of batch devices at a more detailed level to ensure the privacy guarantee of specific samples, which can provide strong privacy. The development of privacy protection methods based on privacy restrictions of specific devices is an interesting and continuing direction for future work.

3.1.2 Trade-off between communication cost and computational pressure

We can mainly consider two aspects to improve the efficiency of communication: iteratively send small messages, or reduce the total number of communication rounds. For example, we can use model compression technology to reduce the data size communicated in federated learning. In terms of reducing communication rounds, the models that need to be communicated can be screened according to their importance. We can also combine these two methods, which can greatly reduce the cost of communication between mobile devices and servers, but it also increases some computational pressure. Finding the trade-off between communication cost and computational pressure is the main direction of future work.

3.1.3 Multi-center federated learning

The challenge of heterogeneity hinders federated learning. Some recent studies have shown that if the heterogeneity of the devices in the system can be obtained in advance, all mobile devices can be grouped according to the heterogeneity, and a local central server can be assigned to each group. We can first aggregate a group of similarly heterogeneous device models, and then send them to the server to aggregate into a global model. Studying multi-center federated learning to solve heterogeneous challenges is a promising direction in future work.

3.1.4 Reliable client selection

In federated learning, mobile devices may upload unreliable data, which could cause the server to fail to aggregate the global model. Therefore, it is crucial to find trustworthy and reliable clients in federated learning tasks.The concept of reputation was introduced as a metric to measure the reliability of the client. Therefore, we can select a highly reliable client during each round of model update to ensure the reliability of decentralised federated learning. The improvement of reliable decentralised federated learning based on this method is a far-reaching research direction in the future.

3.2 CONCLUSION

This project focused on designing energy-aware decentralized aggregation mechanisms in Federated Learning (FL) to overcome the limitations of traditional and centralized learning systems. By enabling peer-to-peer model updates without relying on a central server, we aimed to improve privacy, fault tolerance, and scalability.

We explored and compared various aggregation strategies like gossip-based protocols, energy-efficient node selection, and adaptive client participation. These approaches help reduce communication overhead and energy consumption while maintaining model accuracy in resource-constrained environments.

Despite its benefits, decentralized FL introduces challenges such as synchronization complexity, security risks, and lack of coordination. Our project addressed these through design considerations and proposed future directions including client reliability metrics, multi-center architectures, and model compression.

In summary, our work shows that decentralized, energy-aware FL is a viable and scalable approach for privacy-sensitive, distributed AI applications.

References

[1] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). *A survey on federated learning*. Knowledge-Based Systems, 216, 106775. https://doi.org/10.1016/j.knosys.2021.106775

[2] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), 50-60. https://doi.org/10.1109/MSP.2020.2975749

[3] Sun, T., Li, D., & Wang, B. (2023). Decentralized Federated Averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4), 4289-4303. https://doi.org/10.1109/TPAMI.2022.3196503

Appendix-I

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	SREEJA SANYAL
ROLL NO	22051200
PROJECT TITLE	ENERGY AWARE: DECENTRALISED AGGREGATION MECHANISMS IN FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project explores energy-aware decentralized aggregation in Federated Learning (FL), eliminating reliance on a central server. It analyzes peer-to-peer strategies like gossip and blockchain-based models for improved privacy, fault tolerance, and energy efficiency. Implementation challenges and future research directions for scalable, secure, and efficient decentralized FL are discussed.

CONTRIBUTIONS

- Conceptualized and developed a decentralized Federated Learning (FL) framework emphasizing energy-aware aggregation strategies.
- Implemented peer-to-peer communication protocols including gossip-based and blockchain-inspired models to eliminate central server dependency.
- Designed and tested energy optimization techniques like adaptive client selection, communication round tuning, and model sparsification.
- Evaluated trade-offs in decentralized FL in terms of fault tolerance, privacy, energy efficiency, and communication overhead.
- Identified key implementation challenges such as synchronization delays, Byzantine attacks, and incentive gaps among clients.
- Proposed future-oriented solutions including federated reinforcement learning for node selection and multi-center FL for heterogeneity management.
- Documented the complete development lifecycle in a structured technical report, aligning with real-world edge-device constraints.

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	SAKET KUMAR
ROLL NO	22054212
PROJECT TITLE	ENERGY AWARE: DECENTRALISED AGGREGATION MECHANISMS IN FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project explores energy-aware decentralized aggregation in Federated Learning (FL), eliminating reliance on a central server. It analyzes peer-to-peer strategies like gossip and blockchain-based models for improved privacy, fault tolerance, and energy efficiency. Implementation challenges and future research directions for scalable, secure, and efficient decentralized FL are discussed.

CONTRIBUTIONS

- Conceptualized and developed a decentralized Federated Learning (FL) framework emphasizing energy-aware aggregation strategies.
- Implemented peer-to-peer communication protocols including gossip-based and blockchain-inspired models to eliminate central server dependency.
- Designed and tested energy optimization techniques like adaptive client selection, communication round tuning, and model sparsification.
- Evaluated trade-offs in decentralized FL in terms of fault tolerance, privacy, energy efficiency, and communication overhead.
- Identified key implementation challenges such as synchronization delays, Byzantine attacks, and incentive gaps among clients.
- Proposed future-oriented solutions including federated reinforcement learning for node selection and multi-center FL for heterogeneity management.
- Documented the complete development lifecycle in a structured technical report, aligning with real-world edge-device constraints.

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	ADITYA MAJUMDAR
ROLL NO	22054214
PROJECT TITLE	ENERGY AWARE: DECENTRALISED AGGREGATION MECHANISMS IN FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project explores energy-aware decentralized aggregation in Federated Learning (FL), eliminating reliance on a central server. It analyzes peer-to-peer strategies like gossip and blockchain-based models for improved privacy, fault tolerance, and energy efficiency. Implementation challenges and future research directions for scalable, secure, and efficient decentralized FL are discussed.

CONTRIBUTIONS

- Conceptualized and developed a decentralized Federated Learning (FL) framework emphasizing energy-aware aggregation strategies.
- Implemented peer-to-peer communication protocols including gossip-based and blockchain-inspired models to eliminate central server dependency.
- Designed and tested energy optimization techniques like adaptive client selection, communication round tuning, and model sparsification.
- Evaluated trade-offs in decentralized FL in terms of fault tolerance, privacy, energy efficiency, and communication overhead.
- Identified key implementation challenges such as synchronization delays, Byzantine attacks, and incentive gaps among clients.
- Proposed future-oriented solutions including federated reinforcement learning for node selection and multi-center FL for heterogeneity management.
- Documented the complete development lifecycle in a structured technical report, aligning with real-world edge-device constraints.

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF THE STUDENT	ANANYA BISWAL
ROLL NO	2205874
PROJECT TITLE	ENERGY AWARE: DECENTRALISED AGGREGATION MECHANISMS IN FEDERATED LEARNING
ABSTRACT OF THE PROJECT (WITHIN 80 WORDS)	This project explores energy-aware decentralized aggregation in Federated Learning (FL), eliminating reliance on a central server. It analyzes peer-to-peer strategies like gossip and blockchain-based models for improved privacy, fault tolerance, and energy efficiency. Implementation challenges and future research directions for scalable, secure, and efficient decentralized FL are discussed.

CONTRIBUTIONS

- Conceptualized and developed a decentralized Federated Learning (FL) framework emphasizing energy-aware aggregation strategies.
- Implemented peer-to-peer communication protocols including gossip-based and blockchain-inspired models to eliminate central server dependency.
- Designed and tested energy optimization techniques like adaptive client selection, communication round tuning, and model sparsification.
- Evaluated trade-offs in decentralized FL in terms of fault tolerance, privacy, energy efficiency, and communication overhead.
- Identified key implementation challenges such as synchronization delays, Byzantine attacks, and incentive gaps among clients.
- Proposed future-oriented solutions including federated reinforcement learning for node selection and multi-center FL for heterogeneity management.
- Documented the complete development lifecycle in a structured technical report, aligning with real-world edge-device constraints.

SIGNATURE OF GUIDE **SIGNATURE OF STUDENT**