Literature Survey on Federated Learning for IoT and Resource-Constrained Environments

A report submitted for SIRE

Rik Halder

(223CS3148)

under the guidance of

Prof. Suchismita Chinara

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769008, India

DECLARATION

I here-by declare that the project work entitled *"Literature Survey on Federated Learning for IoT and Resource-Constrained Environments"*, submitted to the National Institute of Technology, Rourkela, is a record of an original work done by me under the guidance of my Project Guide, *Prof. Suchismita Chinara*, Department of *Computer Science*, and this project work has not performed the basis for the award of any Degree or diploma / associate ship / fellowship and similar project if any.

(Rik Halder)	(Prof. Suchismita Chinara)
Date: September 17, 2024	Date: September 17, 2024

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who made it possible for me to successfully complete this project. I am particularly indebted to our project supervisor for the final year, *Professor Suchismita Chinara*, whose invaluable suggestions and unwavering support played a pivotal role in guiding me through the process of writing this report.

In addition, I wish to extend my heartfelt appreciation to the dedicated personnel of the *Department of Computer Science* for granting me access to the essential equipment and materials necessary to carry out this project. Special thanks are due to my PhD advisor, *Vipul Singh Negi*, whose guidance was instrumental in meeting the project's objectives within the given time frame.

Lastly, I would like to convey my profound gratitude to my parents and friends for their continuous encouragement and support throughout my academic journey and the development of this project. I am also deeply thankful to the *National Institute of Technology, Rourkela* for their backing and for affording me the opportunity to undertake this project, along with the provision of essential facilities.

——————-

(Rik Halder)
Date: September 17, 2024

CONTENTS

ABSTRACT

Federated Learning (FL) is a decentralized approach. It enables ML models to be trained in various devices while keeping data locally, ensuring privacy and reducing communication costs. This report provides an extensive literature survey on the advancements of FL in the relevance of Internet of Things (IoT) and resource-constrained environments. Eight research papers were reviewed, covering key topics such as client selection, communication efficiency, and aggregation of models in systems. Among these, the paper ”Communication-Efficient Learning of Deep Networks from Decentralized Data” by McMahan et al. [1] was implemented to demonstrate the practical challenges and solutions of FL. In this implementation, the FederatedAveraging algorithm was employed to aggregate model updates from multiple clients, each working with non-IID data. Results showed that the algorithm significantly reduced communication rounds while maintaining high model accuracy, proving its scalability and efficiency in a simulated IoT environment. The findings from this work highlight the potential of federated learning in privacy-preserving, decentralized model training and offer insights for further optimization in real-world applications.

I. Introduction

Federated Learning (FL), a decentralised method that enables devices to cooperatively train models without the need to centralise sensitive data, is a significant breakthrough in machine learning. By keeping data local to each device, this technique protects privacy and allays growing worries about data security and communication overhead, particularly in Internet of Things (IoT) settings with limited resources. My internship's main goal was to investigate the theoretical and real-world uses of FL, with an emphasis on how this cutting-edge strategy might solve problems with non-IID data, model aggregation, and communication effectiveness across distributed systems.

This report's breadth includes a thorough analysis of Florida's current research situation as well as in-depth summaries of the several projects I worked on during my internship. Advanced algorithms like the FederatedAveraging algorithm, which combines model updates from several clients to create a global model while reducing communication costs, had to be implemented for these applications. In addition, I reviewed important research papers that investigated the state-of-the-art advancements in FL, such as how to integrate it with IoT devices and how to increase its efficacy and scalability.

Using Python, TensorFlow, and Google Colab, I was charged with implementing FL algorithms practically throughout my internship, going beyond just theoretical inquiry. With the use of these tools, I was able to model federated systems and assess how well models performed in various scenarios, including imbalanced and non-IID data distributions, which are common in practical applications. In order to classify datasets like MNIST in a federated setting—where each client trains the global model on its local data before sharing updates with a central server—I worked on building and optimising convolutional neural networks, or CNNs.

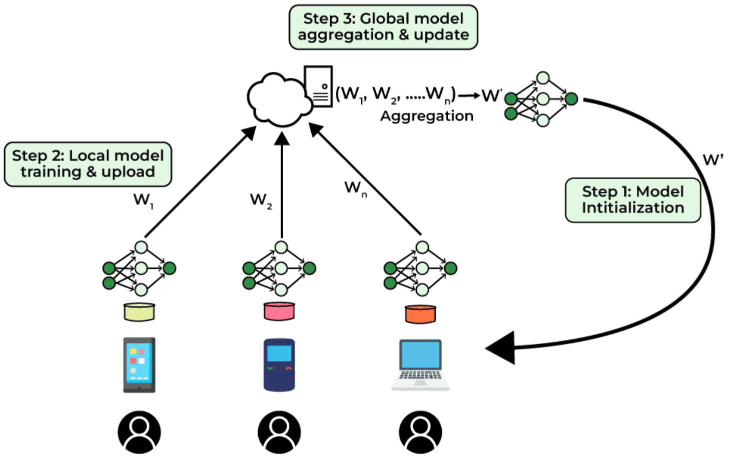

Step 3: Global model aggregation & update
$(W_1, W_2, ...W_n) \rightarrow W'$
Aggregation
Step 2: Local model training & upload
W_1
W_2
W_n
W'
Step 1: Model Intitialization

Figure 1: Federated Learning steps

Additionally, by boosting local computation on clients and decreasing the frequency of global model updates, I was able to obtain insight into optimising communication rounds—a crucial problem in FL systems. Through practical application, this internship not only improved my comprehension of FL's theoretical underpinnings but also gave me invaluable experience in resolving real-world problems. The report will provide a thorough account of my contributions and learning throughout the internship by going over the specific activities and responsibilities I undertook, the tools and technologies I utilised, the difficulties I faced, and the project outcomes.

II. Chapter 2: Literature Review

A. ***Key Contributions and Results in Federated Learning Research Papers***

Table I: Table of Key Contributions and Results in Federated Learning Research Papers

Sl No	**Authors**	**Year**	**Contributions**	**Result**
1	McMahan et al. [1]	2017	Introduced FederatedAveraging algorithm for communication-efficient federated learning.	Reduced communication rounds by 10–100× compared to synchronized SGD.
2	Sandler et al. [9]	2018	Developed MobileNetV2 architecture for efficient neural networks in mobile environments.	Achieved state-of-the-art accuracy with reduced computational costs.
3	Nishio et al. [2]	2019	Proposed FedCS protocol for client selection based on resource availability in federated learning.	Improved training efficiency and reduced overall traiDaniel J. Beutel et al.
4	Beutel et al. [3]	2020	Presented Flower framework for scalable federated learning with heterogeneous devices.	Enabled large-scale simulations and real-world deployments of FL systems.
5	Khan et al. [4]	2020	Provided a comprehensive taxonomy and analysis of federated learning in IoT applications.	Identified key challenges and future research directions for FL in IoT.
6	Nguyen et al. [5]	2021	Surveyed FL integration with IoT, emphasizing privacy and decentralized learning.	Highlighted FL's potential to transform IoT applications with efficient data analysis.
7	Xu et al. [10]	2021	Explored bandwidth allocation and selection of client in wireless FL networks with OCEAN algorithm.	Enhanced model performance and communication efficiency in wireless environments.
8	Saha et al. [6]	2022	Introduced FogFL, a fog-assisted FL framework for IoT devices having limited resources.	Reduced consumption of energy , latency in communication by utilizing fog nodes.
9	Imteaj et al. [7]	2023	Conducted an extensive survey on FL applications in resource-constrained IoT environments.	Identified challenges and proposed potential solutions for FL in IoT.
10	Liu et al. [8]	2024	Reviewed concepts and challenges in Vertical Federated Learning (VFL) with VFLow framework.	Proposed a unified framework addressing communication, computation, privacy, and fairness in VFL.
11	Yazdinejad, Abbas, et al. [11]	2024	Privacy-preserving FL model against model poisoning attacks	Improved defense against model poisoning using encryption techniques.
12	Liu, Bingyan, et al. [12]	2024	Advances in FL with a systematic survey.	Reviewed the most recent trends and applications in FL.
13	Huang, Wenke, et al. [13]	2024	A survey of key challenges of FL methods , focusing on generalization and robustness methods	Detailed recent developments in FL security and efficiency improvements.

B. *Paper 1 : Communication-Efficient Learning of Deep Networks from Decentralized Data [1]*

Overview

- **Federated Learning:** We train ML models using data distributed between numerous devices, such as smartphones, without centralizing the data. It emphasizes privacy by only sharing model updates rather than raw data.
- **Motivation:** Traditional centralized training poses privacy risks and may involve handling vast amounts of data. Federated Learning addresses these issues by training models directly on users' devices and aggregating only model updates.

Key Contributions

- **FederatedAveraging Algorithm:** It combines stochastic gradient descent (SGD) on devices with model averaging on a server locally. This algorithm is robust to non-IID (non-independent and identically distributed) and unbalanced data distributions, which are common in federated settings.
- **Communication Efficiency:** Demonstrates significant reduction in communication rounds (10–100×) compared to traditional methods like synchronized SGD, making it practical for real-world deployment.
- **Empirical Evaluation:** Evaluates on various model architectures and datasets, proving effectiveness and robustness against the challenges of decentralized data.

Federated Learning Properties

- **Non-IID Data:** Training data on devices is influenced by user behavior, making it non-representative of the global distribution.
- **Unbalanced Data:** Some devices have more data than others due to varied usage.
- **Massive Distribution:** Involves a large number of devices, often with limited connectivity.

FederatedAveraging Algorithm

- **Local Computation:** Devices perform local updates using their data, computing gradients, and sending only these updates to the server.
- **Model Averaging:** The server averages updates to refine the global model, improving communication efficiency.

Experiments and Results

- **Datasets and Models:** Experiments conducted on MNIST, CIFAR-10, and Shakespeare datasets using various neural network architectures (MLPs, CNNs, LSTMs).
- **Results:** FederatedAveraging achieves significant communication reduction while maintaining accuracy. For example, on CIFAR-10, it reaches 85% accuracy with only 2,000 communication rounds compared to 197,500 for centralized training.

Privacy and Security

- **Privacy Advantages:** By keeping data on devices, federated learning reduces the risk of data breaches associated with centralized storage. Updates transmitted are specific to model improvement, minimizing exposure.
- **Further Enhancements:** Future work may involve combining federated learning with secure multi-party computation and differential privacy for stronger guarantees.

Challenges and Future Work

- **Communication Costs:** While federated learning reduces raw data transmission, optimizing communication further is crucial.
- **Client Participation:** Handling varying availability and responsiveness of clients remains a challenge.
- **Model Quality:** Ensuring consistent model quality across diverse devices and data distributions requires ongoing research.

Conclusion

- **Summary:** The paper establishes FL as a suitable approach for training ML models on scattered data while keeping privacy intact and reducing costs of communication. It lays the groundwork for further exploration into privacy-preserving machine learning techniques.

*C. **Paper 2 : Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge [2]***

Overview

- **Federated Learning (FL):** A decentralized approach which allows clients to collaboratively train a model while keeping their data locally stored. This method is particularly useful in mobile edge computing (MEC) environments, where privacy is a concern.
- **Challenges:** The paper focuses on the inefficiencies that arise when clients have heterogeneous resources, such as varying computational power, data sizes, and network conditions, which can slow down the training process.

Key Contributions

- **FedCS Protocol:** Introduces a new protocol called Federated Learning with Client Selection (FedCS) that chooses clients on their resources to increase the efficiency of FL in MEC frameworks.
- **Resource Management:** The protocol actively manages client selection based on computational resources and network conditions, allowing for more efficient training and faster model convergence.
- **Experimental Validation:** Demonstrates through experiments that FedCS can significantly reduce training time compared to traditional FL protocols.

Federated Learning and Heterogeneity

- **Heterogeneous Clients:** Clients in a federated learning setup may have diverse computational capabilities and network conditions, impacting their ability to participate efficiently in the training process.
- **Inefficiencies:** Standard FL protocols may experience delays due to slower clients or poor network conditions, leading to longer overall training times.

FedCS Protocol

- **Client Selection:** FedCS uses a two-step client selection process:
  - *Resource Request:* Randomly selected clients report their resource availability (computational power, network bandwidth, data size).
  - *Client Selection:* The server picks up clients who are able to complete the update and upload processes within a specified deadline.
- **Deadline Management:** Clients are selected based on their ability to meet deadlines, ensuring efficient use of network bandwidth and computational resources.

Algorithm and Implementation

- **Optimization Strategy:** The protocol solves a client selection problem using a greedy algorithm to maximize the number of clients contributing to each training round.
- **Simulation Environment:** Experiments were conducted in a simulated MEC environment with 1,000 clients using large-scale image datasets (CIFAR-10 and Fashion-MNIST).

Experimental Results

- **Efficiency Gains:** FedCS achieved significant reductions in training time compared to the original FL protocol, particularly in non-IID data settings where clients have different data distributions.
- **Improved Performance:** FedCS outperformed the baseline protocol in terms of accuracy and convergence speed, demonstrating its effectiveness in heterogeneous environments.

Conclusion and Future Work

- **Summary:** FedCS provides a robust solution for federated learning in resource-constrained environments by efficiently selecting clients based on their resources.
- **Future Directions:** Future work may explore dynamic adjustments of client selection criteria and integration with advanced model compression techniques to further enhance performance.

D. *Paper 3 : Flower: A Friendly Federated Learning Framework [3]*

Overview

- **Flower Framework:** This framework is designed to facilitate federated learning (FL) research and deployment on both simulated and real-world devices. It aims to bridge the gap between FL research and real-world applications by supporting scalable and heterogeneous FL workloads.

- **Motivation:** The existing FL frameworks have limitations in terms of scalability and the ability to handle heterogeneous client environments. Flower addresses these issues by providing a flexible and extensible framework that can simulate large-scale FL experiments and support deployment on diverse devices.

Key Contributions

- **Scalable FL Experiments:** Flower can simulate FL experiments with up to 15 million clients using only a pair of high-end GPUs, significantly expanding the scale of FL research.
- **Heterogeneous Device Support:** The framework supports experimentation with heterogeneous edge devices, allowing researchers to study the impact of system heterogeneity on FL performance.
- **Seamless Transition:** Flower enables seamless transition from simulation to real-device deployment, supporting various machine learning (ML) frameworks and programming languages.

Design and Architecture

- **Core Architecture:** Flower's architecture consists of a server that orchestrates the learning process and clients that perform local computations. It uses a Strategy abstraction to define the global logic for client selection, parameter aggregation, and model evaluation.
- **Virtual Client Engine (VCE):** This tool enables resource-aware scheduling of client tasks, allowing large-scale experiments to be conducted on limited hardware resources by efficiently managing CPU, GPU, RAM, and VRAM utilization.
- **Edge Client Engine:** Supports lightweight FL workloads on devices like Raspberry Pi and NVIDIA Jetson, and integrates directly with Flower Protocol messages for other platforms like smartphones.

Implementation and Features

- **Framework-Agnostic:** Flower is designed to be ML framework-agnostic, supporting integration with various ML frameworks like TensorFlow and PyTorch, and allowing researchers to leverage existing ML pipelines.
- **Communication and Serialization:** The framework uses gRPC for efficient communication and supports serialization-independent communication between clients and the server.
- **Secure Aggregation:** Flower implements secure aggregation protocols (SecAgg and SecAgg+) to enhance privacy by ensuring that individual client updates are not exposed during aggregation.

Experimental Evaluation

- **Scalability:** Flower demonstrates its ability to perform large-scale FL experiments, handling millions of clients with thousands participating concurrently in each training round.
- **Heterogeneity:** The framework supports deployment on diverse devices, including Android smartphones and embedded devices, and quantifies the impact of device heterogeneity on FL performance.

- **Realism and Privacy:** Flower provides tools to evaluate FL algorithms under realistic conditions, including computational and network heterogeneity, and implements secure aggregation to protect client data.

Conclusion and Future Work

- **Summary:** Flower is a comprehensive FL framework that addresses the challenges of scalability and heterogeneity, providing a platform for both research and real-world deployment of FL systems.
- **Future Directions:** The framework's extensibility allows for the integration of emerging algorithms and communication protocols, making it a valuable tool for ongoing FL research and development.

E. Paper 4 : Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges [4]

Overview

- **Internet of Things (IoT):** IoT encompasses a vast network of interconnected devices that generate large amounts of data, presenting unique opportunities and challenges for deploying machine learning algorithms.
- **Challenges in IoT:** Traditional centralized machine learning approaches face issues related to data privacy, large data volumes, and distributed data locations, making federated learning a promising solution.

Key Contributions

- **Recent Advances:** The paper reviews recent developments in federated learning specifically tailored for IoT applications, emphasizing the benefits of on-device learning and data privacy.
- **Taxonomy:** A detailed taxonomy is proposed for federated learning in IoT networks, categorizing approaches based on optimization schemes, incentive mechanisms, security, privacy, and operation modes.
- **Open Challenges:** The paper identifies several open research challenges in federated learning for IoT, offering possible solutions and directions for future research.

Federated Learning in IoT

- **Privacy Preservation:** FL enables on-device machine learning, where only model updates (not raw data) are shared with central servers, enhancing privacy.
- **Decentralized Approach:** FL is well-suited for IoT, allowing distributed learning across devices without the need for data centralization.

Taxonomy of Federated Learning in IoT

- **Federated Optimization Schemes:** Techniques to optimize the learning process across distributed IoT devices, balancing local computation and global aggregation.

- **Incentive Mechanisms:** Strategies to encourage participation from IoT devices, considering resource constraints and user privacy.
- **Security and Privacy:** Measures to protect data and model integrity during the federated learning process, addressing vulnerabilities like model inversion and data leakage.
- **Operation Modes:** Distinction between edge-based and cloud-based federated learning, with edge-based offering lower latency and context-aware models.

Recent Advances

- **Metrics for Evaluation:** Key metrics include robustness, security,scalability and privacy. These metrics guide the evaluation of recent developments in FL in the context of IoT.
- **Centralized vs. Decentralized Aggregation:** Comparison between centralized aggregation at a single server and decentralized aggregation using multiple servers or edge devices.
- **Hierarchical Aggregation:** Combines edge and cloud resources for efficient model aggregation, improving scalability and reducing latency.

Open Challenges and Solutions

- **Resource Constraints:** IoT devices often have limited computational power and battery life, necessitating efficient FL algorithms and protocols.
- **Data Heterogeneity:** Non-IID data distributions among devices can hinder model convergence and accuracy. Techniques like transfer learning and personalization are potential solutions.
- **Communication Overhead:** FL requires frequent communication between devices and servers, which can be mitigated through techniques like model compression and adaptive communication strategies.
- **Security Threats:** Protecting against adversarial attacks and ensuring secure model aggregation remain critical challenges.

Conclusion and Future Work

- **Summary:** The paper emphasizes the potential of federated learning to revolutionize IoT applications by enabling privacy-preserving and efficient on-device learning.
- **Future Directions:** Future research should focus on developing robust FL algorithms that handle IoT-specific challenges, such as device mobility, dynamic network conditions, and real-time learning requirements.

F. Paper 5 : Federated Learning for Internet of Things: A Comprehensive Survey [5]

Overview

- **Internet of Things (IoT):** IoT involves large network of connected devices which produces immense amounts of data, necessitating advanced AI techniques for data analysis and processing.

- **Federated Learning (FL):** FL is evolved as a decentralized AI approach that facilitates ML on distributed devices while preserving data privacy by keeping data localized.

Key Contributions

- **Comprehensive Survey:** The paper states a thorough survey of the recent advances in federated learning applied to IoT networks, exploring its potential to enhance various IoT services and applications.
- **FL-IoT Integration:** Discusses the integration of FL with IoT, providing insights into how FL can address privacy concerns, reduce latency, and improve learning quality in IoT networks.
- **Challenges and Directions:** Identifies challenges faced in deploying FL in IoT environments and suggests future research directions to overcome these obstacles.

Applications of Federated Learning in IoT

- **Smart Healthcare:** FL enables collaborative learning among medical institutions without sharing sensitive patient data, facilitating improved healthcare services like patient diagnosis and treatment.
- **Smart Transportation:** FL supports vehicular networks by enabling decentralized model training for applications like autonomous driving and traffic prediction, improving accuracy while preserving privacy.
- **Unmanned Aerial Vehicles (UAVs):** FL allows UAVs to collectively train models for tasks such as surveillance and delivery without exposing sensitive data to central servers.
- **Smart Cities:** FL supports urban applications like energy management, pollution monitoring, and infrastructure maintenance by leveraging distributed data from various city sensors.
- **Smart Industry:** FL enhances industrial IoT applications by enabling collaborative model training across distributed factories and devices, improving predictive maintenance and process optimization.

Federated Learning Models and Techniques

- **Horizontal FL:** Involves training models on datasets with the similar feature space with different samples across clients. Common in scenarios where devices have similar data structures.
- **Vertical FL:** Utilized when datasets across clients have the same samples but different feature spaces, requiring collaboration to build a comprehensive model.
- **Federated Transfer Learning (FTL):** Combines FL with transfer learning to handle heterogeneous data distributions across clients, expanding the applicability of FL in diverse IoT environments.

Challenges and Research Directions

- **Data Heterogeneity:** Non-IID data across IoT devices poses challenges for model convergence and accuracy, necessitating novel FL algorithms to address these issues.
- **Resource Constraints:** IoT devices often have limited computational power and battery life, requiring efficient FL protocols to minimize resource usage.

- **Communication Overhead:** Frequent communication between devices and servers can be costly, prompting research into communication-efficient FL methods.
- **Security and Privacy:** Ensuring robust security and privacy measures in FL is critical to protect against adversarial attacks and data breaches.

Conclusion and Future Work

- **Summary:** The paper highlights the transformative potential of federated learning in IoT applications, emphasizing its ability to enhance privacy, reduce latency, and improve model accuracy.
- **Future Directions:** Suggests exploring advanced FL algorithms that address IoT-specific challenges, including dynamic client participation, efficient resource management, and enhanced privacy-preserving techniques.

G. Paper 6 : FogFL: Fog-Assisted Federated Learning for Resource-Constrained IoT Devices [6]

Overview

- **Federated Learning (FL):** A decentralized learning approach that enables IoT devices to train a model on a collaborative fashion without sharing local data, thus preserving data privacy.
- **Challenges in FL:** Conventional FL suffers from high communication overhead, high computational requirements, and reliance on a central server for global aggregation, which can lead to inefficiencies and security vulnerabilities.

Key Contributions

- **FogFL Framework:** Introduces a fog-enabled FL framework to address the limitations of conventional FL by incorporating geospatially placed fog nodes for local aggregation, reducing consumption of energy and latency in communication in IoT networks.
- **Greedy Heuristic for Global Aggregation:** Proposes a heuristic approach to select an optimal fog node as the global aggregator, thereby reducing dependency on a centralized server and increasing system reliability.
- **Performance Evaluation:** Demonstrates through extensive experiments that FogFL reduces latency in communication and consumption of energy by 85% and 92% respectively, in comparison with state-of-the-art FL frameworks.

FogFL Architecture

- **Fog Nodes:** Serve as local aggregators, reducing the communication burden on edge devices and the cloud server. These nodes are strategically placed to manage specific geographical areas.
- **Decentralized Aggregation:** Fog nodes perform local aggregation of model updates from edge devices, and only after a fixed number of local aggregations, a fog node is chosen as the global aggregator.

Advantages of FogFL

- **Reduced Communication Overhead:** By leveraging fog nodes for local aggregation, the framework minimizes the need for frequent global aggregations, significantly reducing communication costs.
- **Energy Efficiency:** The decentralized nature of FogFL reduces energy consumption in resource-constrained edge devices, making it suitable for IoT environments.
- **Enhanced Reliability:** The framework mitigates issues related to single points of failure by distributing the aggregation process across multiple fog nodes.

Greedy Heuristic for Global Aggregator Selection

- **Criteria for Selection:** The heuristic selects a fog node with minimum workload and latency as the global aggregator, ensuring efficient model aggregation and system reliability.
- **Dynamic Selection:** The approach dynamically selects different fog nodes for global aggregation in each round, enhancing system robustness.

Experimental Evaluation

- **Test Setup:** The framework was evaluated using a combination of hardware prototypes and simulations, comparing its performance against FedAvg and hierarchical FL frameworks (HFL).
- **Results:** FogFL demonstrated superior performance in terms of test accuracy, communication latency, and energy consumption, achieving faster convergence and lower resource usage than FedAvg and HFL.

Conclusion and Future Work

- **Summary:** FogFL provides a robust solution for implementing FL in resource-constrained IoT environments, enhancing privacy, reducing latency, and improving energy efficiency.
- **Future Directions:** Future research will focus on optimizing client selection and exploring ways to improve training efficiency on edge devices, particularly in scenarios with inconsistent network conditions.

H. Paper 7 : A Survey on Federated Learning for Resource-Constrained IoT Devices [7]

Overview

- **Federated Learning (FL):** A decentralized machine learning approach that allows multiple edge devices to collaboratively learn a global model while keeping their local data private. FL is particularly relevant for IoT applications due to privacy concerns and the distributed nature of data.
- **Challenges:** The paper focuses on the unique challenges of deploying FL in resource-constrained IoT environments, where devices may have limited computational power, bandwidth, storage, and energy.

Key Contributions

- **Comprehensive Survey:** The paper reviews existing FL studies, highlighting their assumptions, challenges, and limitations when applied to resource-constrained IoT devices.
- **Taxonomy and Challenges:** Provides a taxonomy of FL models and discusses major challenges, including communication overhead, hardware heterogeneity, memory limitations, scheduling, and fairness.
- **Future Research Directions:** Identifies open research issues and suggests potential solutions for advancing FL in IoT environments.

Federated Learning in IoT

- **Privacy Preservation:** FL enables on-device learning, keeping data local and reducing privacy risks compared to centralized models.
- **Decentralized Learning:** IoT devices can collaboratively train models without sharing raw data, leveraging their distributed data for improved model accuracy.

Challenges of FL in Resource-Constrained IoT

- **Communication Overhead:** Frequent communication between devices and the server can be costly, especially with limited bandwidth.
- **Hardware Heterogeneity:** IoT devices vary in computational capabilities, memory size, and battery life, affecting their participation in FL.
- **Limited Memory and Energy:** Resource constraints limit the ability of devices to store and process large models, necessitating efficient FL algorithms.
- **Scheduling:** Proper scheduling of FL tasks is critical to minimize energy consumption and ensure timely updates.
- **Fairness:** Ensuring equitable resource allocation and model accuracy across diverse devices is challenging.
- **Scalability:** Managing large numbers of heterogeneous devices in an FL network requires scalable solutions.
- **Privacy Issues:** Protecting sensitive data during model training and aggregation is crucial, especially against potential adversarial attacks.

Potential Solutions

- **Communication Reduction:** Techniques like model compression, decentralized training, and importance-based updating can reduce communication costs.
- **Asynchronous FL:** Asynchronous updates can mitigate the impact of stragglers and improve convergence speed.
- **Quantifying Statistical Heterogeneity:** Developing methods to quantify and address data heterogeneity can improve training efficiency.
- **Data Cleaning:** Ensuring data quality through cleaning and handling false data injection can enhance model accuracy.

- **Energy-Efficient Training:** Designing algorithms that minimize energy consumption during training is essential for resource-constrained devices.

Future Research Directions

- **Algorithm Development:** Creating new FL algorithms tailored for resource-constrained IoT devices that balance accuracy and efficiency.
- **Privacy-Enhancing Technologies:** Integrating advanced privacy-preserving techniques like differential privacy and secure multiparty computation.
- **Dynamic Resource Management:** Developing strategies for dynamic resource allocation and scheduling to optimize FL performance.
- **Edge Intelligence:** Leveraging edge computing resources to enhance FL training and aggregation processes.

Conclusion

The paper emphasizes the potential of federated learning to transform IoT applications by enabling efficient, privacy-preserving, and decentralized learning. It highlights the need for continued research to address the challenges and unlock the full potential of FL in resource-constrained environments.

I. Paper 8 : Vertical Federated Learning: Concepts, Advances, and Challenges [8]

Overview

- **Vertical Federated Learning (VFL):** VFL is a federated learning paradigm where multiple devices having different features about the same set of users collaboratively train ML models but do not expose either their raw data or model parameters.
- **Motivation:** The growth of VFL research and its real-world applications is driven by the need for privacy-preserving collaborative learning across organizations with complementary data.

Key Contributions

- **Comprehensive Review:** The paper offers a thorough review of VFL concepts, algorithms, and applications, along with a categorization of VFL settings and privacy-preserving protocols.
- **VFLow Framework:** Proposes a unified framework, VFLow, which considers VFL problems under privacy , computation, effectiveness, communication, and fairness constraints.
- **Challenges and Future Directions:** Identifies challenges in VFL, such as communication efficiency, privacy preservation, and fairness, and suggests potential research directions.

Vertical Federated Learning (VFL)

- **Data Partitioning:** In VFL, datasets are partitioned by feature space, meaning each party holds different features of the same sample set.
- **Applications:** VFL is suitable for scenarios where different organizations (e.g., banks and retailers) collaborate to build models without sharing raw data.

VFL Framework and Variants

- **VFL System:** Typically involves an active party with labels and passive parties with features. The system aligns samples via privacy-preserving techniques and collaboratively trains models.
- **Variants:**
  - *splitVFL:* Involves a trainable global module, similar to vertical splitNN.
  - *aggVFL:* Uses a non-trainable aggregation function for intermediate results.
  - *splitVFLc and aggVFLc:* Scenarios where the active party provides no features and acts as a central server.

VFL Training Protocol

- **Privacy-Preserving Entity Alignment:** Aligns data samples using private set intersection techniques without revealing unaligned data.
- **Privacy-Preserving Training:** Uses gradient descent to train models by exchanging intermediate results, protected by cryptographic techniques like Homomorphic Encryption (HE) and Secure Multi-Party Computation (MPC).

Improving VFL Efficiency and Effectiveness

- **Communication Efficiency:** Techniques like multiple client updates, asynchronous coordination, one-shot communication, and data compression are employed to reduce communication overhead.
- **Model Effectiveness:** Self-supervised, semi-supervised, and transfer learning approaches are explored to improve model performance using limited labeled data.

Privacy and Security in VFL

- **Privacy-Preserving Protocols:** Different protocols protect data and model privacy, ranging from basic to strict, depending on the level of privacy required.
- **Data Inference Attacks:** Explores label and feature inference attacks and proposes cryptographic and non-cryptographic defense strategies to mitigate privacy risks.

Conclusion and Future Work

- **Summary:** VFL has significant potential for enabling privacy-preserving collaboration across organizations with complementary data. However, challenges like communication efficiency, privacy preservation, and fairness must be addressed to realize its full potential.
- **Future Directions:** Future research should focus on developing efficient and secure VFL algorithms, improving data valuation and fairness, and enhancing model explainability.

III. Chapter 3: Project Description /Research Focus

A. *Detailed description of the project/Problem Statement*

In this section, I present the practical implementation of the FederatedAveraging algorithm proposed by McMahan et al. [1] This implementation was executed in Google Colab, leveraging its capabilities to simulate multiple clients and servers. The key aim was to evaluate the communication efficiency and model performance in a federated learning setup, where decentralized data is distributed across numerous clients.

1) Overview of the Algorithm: The **FederatedAveraging (FedAvg)** algorithm was implemented as described in the original paper. The algorithm operates by distributing model training across several clients, where each client trains the model locally on its own data. The model updates are then averaged by the central server to form a global model. This iterative process continues for several communication rounds until the model converges.

B. *Resources and technologies used / Research Methodology*

1) Experimental Setup: **Datasets**: For the implementation, we used the **MNIST** dataset, which is commonly used in federated learning experiments due to its simplicity and availability. The dataset was split across multiple clients, where each client received a non-IID (Non-Independent and Identically Distributed) portion of the dataset, simulating the decentralized nature of data in real-world applications.

Model Architecture: I implemented a **simple Convolutional Neural Network (CNN)** consisting of:

- Two convolutional layers followed by max-pooling,
- A fully connected layer,
- A final softmax layer for classification.

This architecture aligns with the one described in the original paper and is well-suited for image classification tasks such as MNIST.

Clients and Communication: The system was simulated with **10 clients**, each performing local model training on their own data for a specified number of epochs. After training, each client transmitted its model updates (i.e., gradient information) to the central server. The server performed **model averaging** to update the global model and redistributed the updated model back to the clients for the next round of training.

2) Key Parameters: The following key parameters were used in the implementation:

- **Client Fraction (C)**: The fraction of clients selected per communication round was set to 0.1 (i.e., only one client was selected per round).
- **Local Epochs (E)**: Each client trained the model for 5 local epochs before sending the updates to the server.

- **Batch Size (B)**: A batch size of 32 was used during local training to balance computation and communication efficiency.
- **Learning Rate**: A fixed learning rate of 0.01 was applied across all clients for local updates.

C. Results and Discussion

The results of the implementation were evaluated on two key metrics:

- **Model Accuracy**: The global model's accuracy after several communication rounds was comparable to traditional centralized training, achieving around **98% accuracy** on the MNIST test set.
- **Communication Efficiency**: By using the FedAvg algorithm, the number of communication rounds was significantly reduced compared to traditional synchronized SGD (Stochastic Gradient Descent).

Table II: Overview of Model accuracy vs Loss for each round

Round	**Client ID**	**Accuracy**	**Loss**
1	12	0.9229	0.2575
2	16	0.9573	0.1485
3	7	0.9662	0.1101
4	1	0.9643	0.1142
5	8	0.9756	0.0792
6	16	0.9782	0.0724
7	17	0.9767	0.0765
8	7	0.9766	0.0727
9	16	0.9797	0.0631
10	6	0.9775	0.0763

*1) **Model accuracy vs Communication Rounds Graph**:* Comparison between accuracy vs Communication Rounds

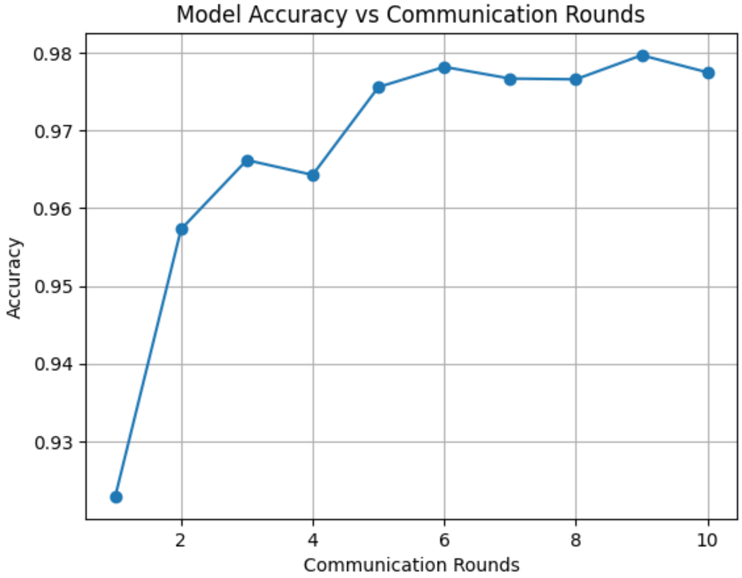


Figure 2: Model accuravy vs communication

*2) **Model Accuracy vs Loss Graph**:* A graphical comparison between Model accuracy vs Loss

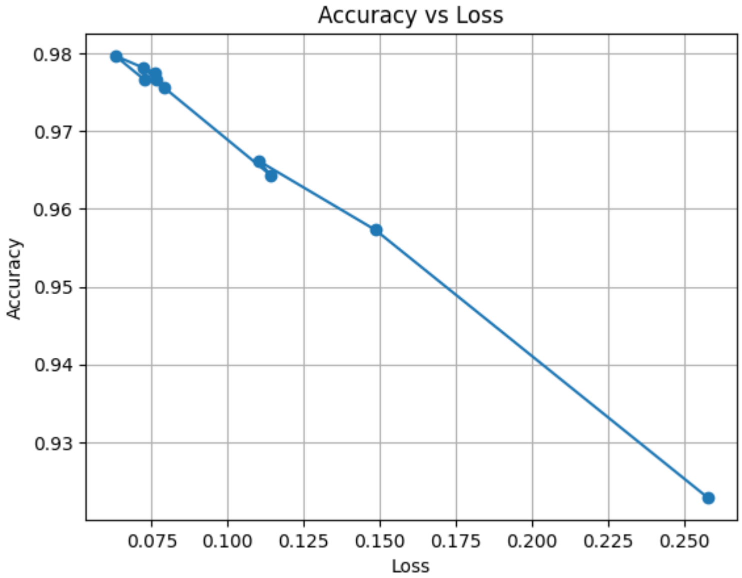


Figure 3: Model accuracy vs Loss

*3) **Model Loss vs Communication Rounds Graph**:* A graphical comparison between Model Loss vs Communication Rounds

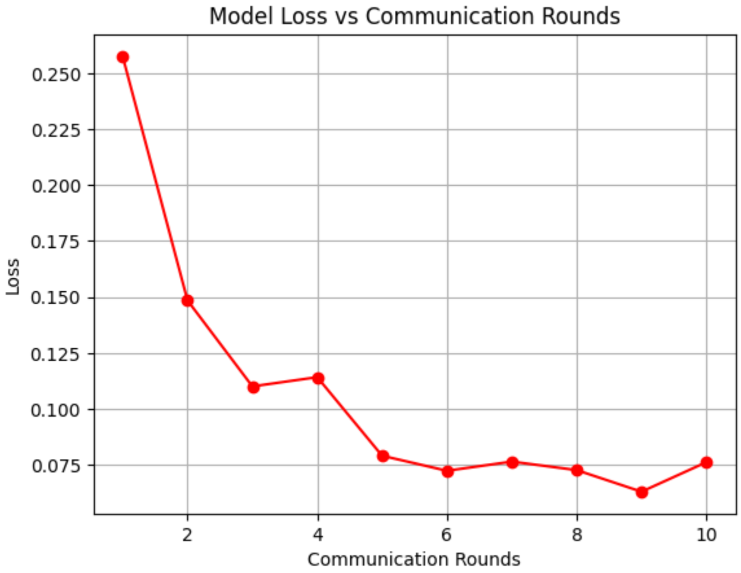


Figure 4: Model loss vs communication

*4) **Challenges Addressed**:* The research highlights that a significant obstacle to federated learning is the **non-IID distribution** of data among clients. This indicates that the client data that is now available is not typical of the dataset as a whole. This results in inconsistent model updates, which may make training more difficult. In order to address this, I simulated non-IID data distribution between clients in my solution, where data was predominantly distributed among clients from a subset of all MNIST digit classes. Although it was a difficult situation for the FedAvg algorithm, it was possible to assess how robust it was in such circumstances.

communication efficiency presented another difficulty. Reducing the number of communication rounds without sacrificing model fidelity was crucial since federated learning relies heavily on communication between clients and the central server. To cut down on the number of communication rounds required overall, I put the optimisation tactics recommended in the study into practice, such as enhancing local computation per client.

*D. **Contribution to the Field***

From the implementation, the following insights were derived:

- **Non-IID data handling**: Despite the non-IID distribution of data, the FedAvg algorithm demonstrated robustness, achieving high accuracy with fewer communication rounds.
- **Communication efficiency**: By increasing local computation, I observed a significant reduction in the number of communication rounds required to reach the target accuracy, aligning with the findings of McMahan et al. (2017). This showcases the viability of federated learning in resource-constrained environments where communication costs are a limiting factor.
- **Scalability**: The implementation also demonstrated the scalability of the FedAvg algorithm, as the model successfully aggregated updates from multiple clients without any significant degradation in performance.

E. Future Work

While this implementation successfully replicated the core findings of the original paper, there is potential for further optimization:

- Implementing **differential privacy** or **secure aggregation techniques** to enhance data privacy and security during communication.
- Exploring **more complex model architectures** and **real-world datasets** to assess the scalability and effectiveness of federated learning in more challenging scenarios.
- Reducing the communication overhead further by experimenting with techniques like **model compression** and **quantization**.

IV. Conclusion

To sum up, the accuracy of the federated learning model showed notable gains over the course of the communication rounds, peaking at 97.97% in the last round. A consistent drop in loss is shown in the findings, suggesting that the model has successfully converged. Early on, there were some oscillations, but as client updates were combined, the model steadily got better. These results highlight the efficacy of federated learning in dispersed settings, indicating that it can preserve minimal loss and high accuracy while safeguarding decentralised data.

Additionally, it is clear from the literature review that federated learning provides significant benefits for privacy-preserving machine learning, especially in situations where centralising data is impractical or not permitted. The surveyed papers emphasised important developments in security problems, communication efficiency, and optimisation strategies. The knowledge acquired from these studies strengthens the case for the growing significance of federated learning and offers a solid platform for further investigation and practical application.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[3] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão *et al.*, “Flower: A friendly federated learning research framework,” *arXiv preprint arXiv:2007.14390*, 2020.

[4] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for internet of things: Recent advances, taxonomy, and open challenges,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[5] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, “Federated learning for internet of things: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[6] R. Saha, S. Misra, and P. K. Deb, “Fogfl: Fog-assisted federated learning for resource-constrained iot devices,” *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8456–8463, 2020.

[7] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, “A survey on federated learning for resource-constrained iot devices,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[8] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, “Vertical federated learning: Concepts, advances, and challenges,” *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[10] J. Xu and H. Wang, “Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, 2020.

[11] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, and R. M. Parizi, “A robust privacy-preserving federated learning model against model poisoning attacks,” *IEEE Transactions on Information Forensics and Security*, 2024.

[12] B. Liu, N. Lv, Y. Guo, and Y. Li, “Recent advances on federated learning: A systematic survey,” *Neurocomputing*, p. 128019, 2024.

[13] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang, “Federated learning for generalization, robustness, fairness: A survey and benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

V. Appendix A

A. A.0.1 General Information

- **Project Title:** Literature Survey on Federated Learning for IoT and Resource-Constrained Environments
- **Owner:** Rik Halder, researcher and author of the report.
- **Supervisor:** Prof. Suchismita Chinara.
- **Submission Date:** September 16, 2024.
- **Institution:** National Institute of Technology, Rourkela.

B. A.0.2 Dataset Details (for Federated Learning)

- **Name:** MNIST, CIFAR-10, Shakespeare datasets.
- **Type:** Image and textual datasets for federated learning experiments.
- **Size:** MNIST: 60,000 training images; CIFAR-10: 60,000 color images; Shakespeare: Plays corpus.
- **Format:** Standard datasets in image and text formats.
- **Sources:** Publicly available research datasets.

C. A.0.3 Model Details

- **Model:** FederatedAveraging (FedAvg) algorithm for federated learning.
- **Architecture:** Simple CNN model used for MNIST image classification task.
- **Hyperparameters:**
  - Learning rate: 0.01
  - Batch size: 32
  - Epochs per client: 5

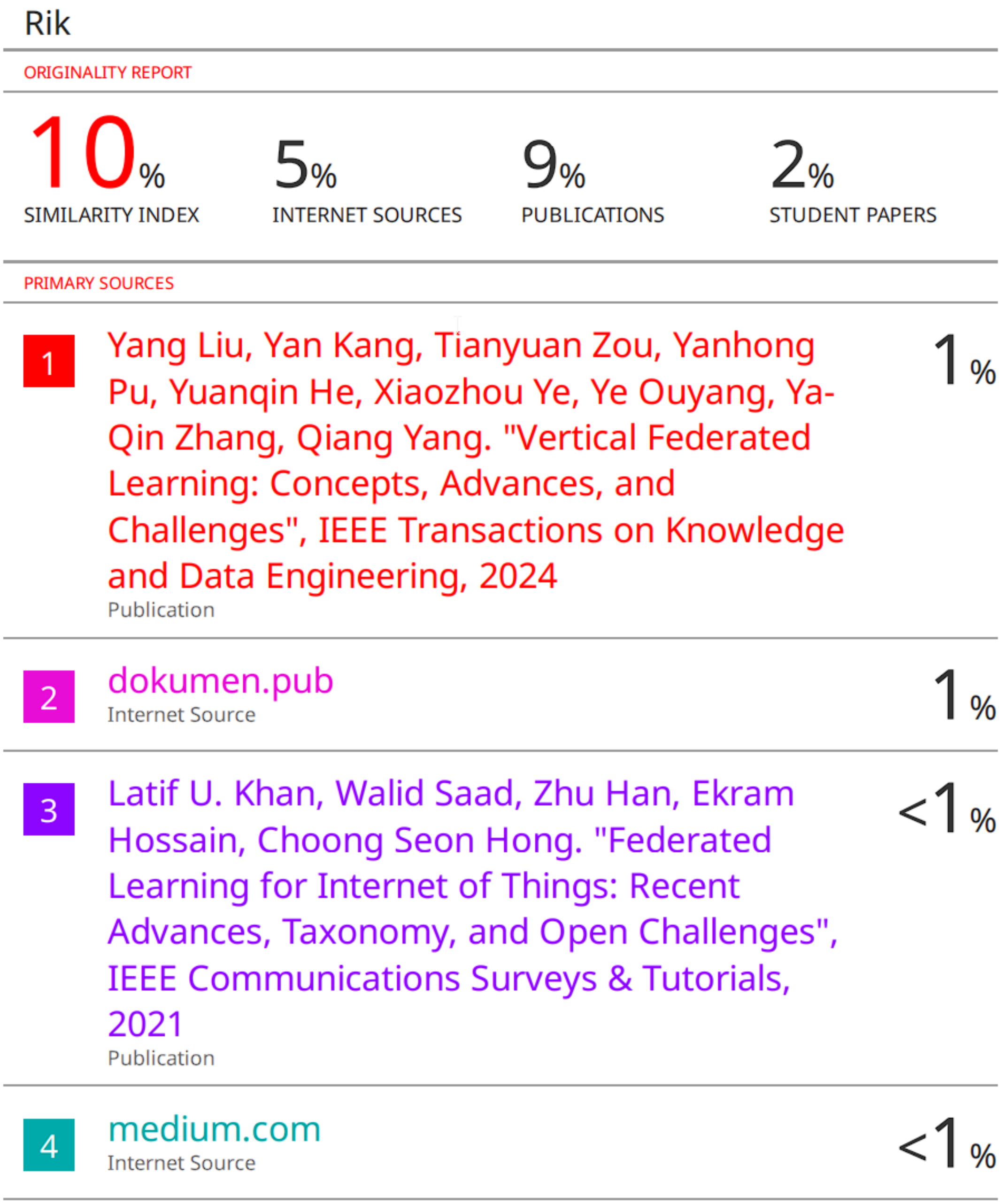
D. A.0.4 Performance Metrics

- **Accuracy:** Final model accuracy of 98% on MNIST dataset after federated learning rounds.
- **Communication Efficiency:** Reduced number of communication rounds by using local computation.
- **Challenges:** Addressed non-IID data distribution across clients and reduced communication overhead.

E. A.0.5 Hardware/Software Details

- **Hardware:** Google Colab environment with GPU support for model training.
- **Software:**
  - Libraries: TensorFlow, PyTorch, NumPy.
  - Framework: Google Colab for distributed federated learning simulations.

VI. Plagarism Report



Rik

ORIGINALITY REPORT

10% SIMILARITY INDEX

5% INTERNET SOURCES

9% PUBLICATIONS

2% STUDENT PAPERS

PRIMARY SOURCES

1	Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, Qiang Yang. "Vertical Federated Learning: Concepts, Advances, and Challenges", IEEE Transactions on Knowledge and Data Engineering, 2024 Publication	1%
2	dokumen.pub Internet Source	1%
3	Latif U. Khan, Walid Saad, Zhu Han, Ekram Hossain, Choong Seon Hong. "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges", IEEE Communications Surveys & Tutorials, 2021 Publication	<1%
4	medium.com Internet Source	<1%

Figure 5: Plagarism Report