Blekinge Institute of Technology
Doctoral Dissertation Series No. 2024:04
ISSN 1653-2090
ISBN 978-91-7295-478-6

Resource-Aware and Personalized Federated Learning via Clustering Analysis

Ahmed Abbas Mohsin Al-Saedi

DOCTORAL DISSERTATION
for the degree of Doctor of Philosophy at Blekinge Institute of Technology to be publicly defended on May 17th, 2024, at 10:00 in room C413A, Campus Gräsvik

Supervisors
Prof. Veselka Boeva, Blekinge Institute of Technology, Sweden
Prof. Emiliano Casalicchio, Sapienza University of Rome and Blekinge Institute of Technology

Faculty Opponent
Prof. György Dán, KTH Royal Institute of Technology, Sweden
Grading Committee
Prof. Volker Markl, Technical University of Berlin, Germany
Prof. Paul Davidsson, Malmö University, Sweden
Assoc. Prof. Eva Cernadas Garsia, University of Santiago de Compostela, Spain

Abstract

Today's advancement in Artificial Intelligence (AI) enables training Machine Learning (ML) models on the daily-produced data by connected edge devices. To make the most of the data stored on the device, conventional ML approaches require gathering all individual data sets and transferring them to a central location to train a common model. However, centralizing data incurs significant costs related to communication, network resource utilization, high volume of traffic, and privacy issues. To address the aforementioned challenges, Federated Learning (FL) is employed as a novel approach to train a shared model on decentralized edge devices while preserving privacy. Despite the significant potential of FL, it still requires considerable resources such as time, computational power, energy, and bandwidth availability. More importantly, the computational capabilities of the training devices may vary over time. Furthermore, the devices involved in the training process of FL may have distinct training datasets that differ in terms of their size, quality and distribution. As a result of this, the convergence of the FL models may become unstable and slow. These differences can influence the FL process and ultimately lead to sub-optimal model performance within a heterogeneous federated network.

In this thesis, we have tackled a number of the aforementioned challenges. Initially, a resource-aware FL algorithm is proposed that utilizes cluster analysis to address the problem of communication overhead. This issue poses a major bottleneck in FL, particularly for complex models, large-scale applications, and frequent updates. The subsequent step in this thesis involved extending the previous study to include wireless networks (WNs). In WNs, achieving energy-efficient transmission is a significant challenge due to their limited resources. This has motivated us to continue with a comprehensive overview and classification of the latest advancements in context-aware edge-based AI models, with a specific emphasis on sensor networks. The review has also investigated the associated challenges and motivations for adopting AI techniques, along with an evaluation of current areas of research that need further investigation. To optimize the aggregation of the FL model and alleviate communication expenses, the resource-aware FL algorithm is extended with cluster optimization approach. Furthermore, to reduce the detrimental effect caused by data heterogeneity between edge devices on FL, a new study of group-personalized FL models is conducted. We have also studied and proposed resource-aware techniques for analyzing clients' contributions by assessing their behavior during training.

The proposed FL algorithms are assessed on a range of real-world datasets. The extensive experiments have demonstrated their effectiveness and robustness. They improve communication efficiency, resource utilization, model convergence speed, and aggregation efficiency in comparison with similar state-of-the-art methods.

Keywords: Federated Learning, Clustering Analysis, Eccentricity Analysis, Non-IID Data, Model Personalization

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2024:04

Resource-Aware and Personalized Federated Learning via Clustering Analysis

Ahmed Abbas Mohsin Al-Saedi

Doctoral Dissertation in Computer Science

Department of Computer Science
Blekinge Institute of Technology
SWEDEN

Blekinge Institute of Technology
Department of Computer Science

Blekinge Institute of Technology Doctoral Dissertation Series No. 2024:04
ISBN 978-91-7295-478-6
ISSN 1653-2090
urn:nbn:se:bth-26081

Printed in Sweden by Media-Tryck, Lund University, Lund 2024

Dedication

To Zaynab, Mohamed, and Maryam. I love you more than words could ever say.

“There is no wealth like knowledge, no poverty like ignorance.”

Ali ibn Abi Talib

Acknowledgements

To take a quote from Alfred North Whitehead, “No one who achieves success does so without acknowledging the help of others.” Firstly, I would like to express my utmost gratitude to *Prof. Veselka Boeva*, for her constant guidance, insightful suggestions, and kind support over the last four years. I will always consider myself very privileged to have been one of her students. Also, I extend my sincere gratitude to *Prof. Emiliano Casalicchio*, whose indispensable guidance and valuable feedback were continuous throughout this endeavor.

My sincere appreciation goes to my parents and brothers. I owe immense gratitude to my father. I really wish my father were still alive to share this with us. Extremely grateful to my mother, whose prayers are always with me. I am also so indebted to my father-in-law for his support and encouragement, who has always been there for me.

This would never have been possible without my beloved wife, *Zaynab*, who put her successful academic career and dreams on hold to help me achieve mine. Your unlimited support and unwavering patience have been a constant motivation and strength source through the challenges of my PhD studies. My son *Mohamed* and my daughter *Maryam*, you were refreshing me in all my ways. Words would never express how grateful I am to have you all.

What remains to express is my heartfelt gratitude to all staff in BTH and colleagues in the computer science department for the friendly environment and permanent willingness to help at any time during the entire PhD journey.

April 2024, Karlskrona, Sweden
Ahmed A. Al-Saedi

List of Papers

This thesis is a compilation of the six papers found below. The formatting of the included papers has been changed to conform to a common style, no other changes have been performed.

Paper I

Ahmed A. Al-Saedi, Veselka Boeva and Emiliano Casalicchio. ”Reducing Communication Overhead of Federated Learning through Clustering Analysis”. 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 2021, pp. 1-7. DOI: 10.1109/ISCC53001.2021.9631391

Paper II

Ahmed A. Al-Saedi, Emiliano Casalicchio and Veselka Boeva. ”An Energy-Aware Multi-Criteria Federated Learning Model for Edge Computing”. 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, 2021, pp. 134-143. DOI: 10.1109/FiCloud49777.2021.00027

Paper III

Ahmed A. Al-Saedi, Veselka Boeva, Emiliano Casalicchio and Peter Exner. ”Context-Aware Edge-Based AI Models for Wireless Sensor Networks—An Overview”. In: Emerging Sensor Communication Network-Based AI/ML Driven Intelligent IoT, Sensors 2022, 22(15). ISSN: 1424-8220. DOI: 10.3390/s22155544

Paper IV

Ahmed A. Al-Saedi, Veselka Boeva and Emiliano Casalicchio. ”FedCO: Communication-Efficient Federated Learning via Clustering Optimization”. In: Edge-Cloud Computing and Federated-Split Learning in the Internet of Things, Future Internet 2022,

14(12). ISSN: 1999-5903. DOI: 10.3390/fi14120377. The paper is an extension of Paper I.

Paper V

Ahmed A. Al-Saedi and Veselka Boeva. "Group-Personalized Federated Learning for Human Activity Recognition Through Cluster Eccentricity Analysis". In Engineering Applications of Neural Networks, June, 2023, pp. 522- 536, Springer, León, Spain. DOI: $10.1007/978-3-031-34204-2_{41}$

Paper VI

Ahmed A. Al-Saedi, Veselka Boeva and Emiliano Casalicchio. "Contribution Prediction in Federated Learning via Client Behavior Evaluation", submitted for journal publication (under review).

Other research contributions that are related to this thesis but are not included:

Paper VII

Boeva, Veselka, Emiliano Casalicchio, Shahrooz Abghari, Ahmed A. Al-Saedi, Vishnu Manasa Devagiri, Andrej Petef, Peter Exner, Anders Isberg and Mirza Jasarevic. "Distributed and Adaptive Edge-based AI Models for Sensor Networks (DAISeN)". Position Papers of the 17th Conference on Computer Science and Intelligence Systems, Annals of Computer Science and Information Systems 31 (2022): 71-78. DOI: 10.15439/2022F267

Paper VIII

Emiliano Casalicchio, Simone Esposito and Ahmed A. Al-Saedi. "FLWB: a Workbench Platform for Performance Evaluation of Federated Learning Algorithms". 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), Rome, Italy, 2023, pp. 401-405. DOI: 10.1109/TechDefense59795.2023.10380832

Funding

- The author is supported by the Iraq Ministry of Higher Education and Scientific Research PhD Scholarship.
- Part of the research work presented in this thesis was partially funded by
  - ”Distributed and Adaptive Edge-based AI Models for Sensor Networks”, Sony Research Award Program 2020 Project.
  - ”Human-centered Intelligent Realities (HINTS)”, a project funded by the Swedish Knowledge Foundation (grant: 20220068).

Author's contribution to the papers

Paper I

Co-defined the research problem. Designed and executed the experimental evaluation. Conducted the main part of the analysis of the results. Wrote and edited most of the paper.

Paper II

Defined the research problem. Designed and implemented the proposed algorithm. Designed and executed the experimental evaluation. Conducted the main part of the analysis of the results. Wrote and edited most of the paper.

Paper III

Co-defined the research problem. Performed the literature search and the main part of the data analysis and synthesis. Wrote and edited the majority of the paper.

Paper IV

Defined the research problem. Designed and implemented the proposed algorithm. Designed and executed the experimental evaluation. Conducted the main part of the analysis of the results. Wrote and edited most of the paper.

Paper V

Defined the research problem. Designed and implemented the proposed algorithm. Designed and executed the experimental evaluation. Conducted the main part of the analysis of the results. Wrote and edited most of the paper.

Paper VI

Defined the research problem. Designed and implemented the proposed algorithm. Designed and executed the experimental evaluation. Conducted the main part of the analysis of the results. Wrote and edited most of the paper.

Abbreviations

AI	Artificial Intelligence.
CFL	Clustered Federated Learning.
CMFL	Communication-Mitigated Federated Learning.
CNN	Convolutional Neural Network.
DL	Deep Learning.
ED	Euclidean Distance.
FedAvg	Federated Averaging.
FL	Federated Learning.
HAR	Human Activity Recognition.
IID	Independently and Identically Distributed.
IoT	Internet of Things.
KD	Knowledge Distillation.
MCL	Markov Clustering.
ML	Machine Learning.
MTL	Multi-Task Learning.
NN	Neural Network.
Non-IID	Non-Independently and Identically Distributed.
RL	Reinforcement learning.
SGD	Stochastic Gradient Descent.
SI	Silhouette Index.
SNs	Sensor Networks.
TEDA	Typicality and Eccentricity Data Analytics.
WNs	Wireless Networks.

Table of Contents

1 Introduction

Today, recent developments in Artificial Intelligence (AI), including the use of Machine Learning (ML) and Deep Learning (DL) techniques, in particular, have resulted in remarkable advances of the Internet of Things (IoT) applications. Much of this success is mainly due to the presence of large-scale training infrastructures and large amounts of training data [1]. The widely used approach involves gathering data in a central location, processing them centrally, and creating a unified model based on the processed data. This increases data transmission costs and raises privacy issues. In addition to preserving privacy, the concept of learning on the edge, which involves moving computing to the location where data was initially captured and stored, is becoming increasingly attractive due to its energy efficiency and considerations for climate change [2]. Although the idea of transferring computation to distributed edge devices has been presented for a long time, its application was mainly limited to basic tasks such as querying in sensor networks [3] and fog computing [4]. However, with AI chipsets and available computing resources on edge devices, the training of AI models has gradually shifted from the central server to edge devices.

In light of this context, Google has presented the concept of Federated Learning (FL) [5] as an emerging ML paradigm for decentralized data to tackle these challenges. It enables multiple edge devices to collaboratively train a central AI model without direct access to their private data. Learning occurs locally on the devices, orchestrated by a central server. In this paradigm, model updates are exchanged instead of sharing raw data. Collaboration among these devices can lead to better generalization, offering advantages in terms of privacy and distributed computation [6]. This is especially advantageous in sectors such as healthcare care, where the utmost importance is placed on data privacy. McMahan et al. [5] has introduced the Federated Averaging (FedAvg) algorithm to implement this concept, originally intended for a cross-device scenario often observed in mobile phones. The field of FL offers a promising approach, as it addresses the issues of centralized learning while upholding data privacy when applied in the real world [7]. However, despite its potential, this approach still confronts considerable challenges in the domain. These include the elevated communication cost required for model updates transferred between the central server and client devices, the energy consumption required for client devices, and the diversity of potentially large amounts of data involved in such a process. Some of these challenges are addressed in the papers collected in this thesis; in the

following section, we will discuss these challenges in more detail.

1.1 Federated Learning Challenges

FL continues to gain attention, and researchers are actively working to address the associated challenges and improve system performance. However, despite FL having several advantages, such as preservation of privacy, collaborative learning, and decentralization. The more benefits it offers, the more challenges it presents that need to be paid attention to. In this section, we discuss specific challenges that can be further explored to improve the performance of the FL system.

- **Resource Limitations**: FL process requires iterative transfer of data (e.g., model parameters, weights, etc.) between a central server and edge devices. For example, when it comes to Neural Network (NN) [8], these models contain a large number of parameters, reaching millions, and require frequent updates to reach the desired convergence. Consequently, the requirement for communication bandwidth is exceptionally high. On the other hand, in such environments, participating devices are typically small in size and have a constrained nature in terms of connectivity and computing resources [9, 10]. In addition, the number of participant devices can range from hundreds to millions. Thus, FL requires substantial communication resources and energy overhead before reaching the desired accuracy in such a scenario [11, 12]. To reap the advantages of FL, these challenges must be addressed to allow for the wider adoption of FL systems [13]. These challenges have been addressed in **Papers I, II, IV,** and **VI**. More specifically, different FL models are proposed to address the problem of communication overhead, minimize energy usage, and evaluate the individual contribution of each client participating in FL. Each paper focuses on these aspects independently.

- **Expensive Communication**: As stated previously, the models for which participating devices train locally and exchange with the server may be quite large. For instance, VGG-16, a NN used for image recognition, contains 138 million parameters [14] and requires 526 megabytes of storage when encoded with 32 bits. In addition, the number of devices involved in the FL training could range from hundreds of thousands to millions, Figure 1.1 illustrates the run-time cost of FedAvg, with the red and blue blocks denoting the local computation time and communication delay, respectively. To demonstrate the impact of different numbers of clients W_t in round (t) on the FedAvg algorithm for a fixed number of training rounds (e.g., $T = 3$), we generate a graph of two global averaging steps. Specifically, we set $W_t = 3$ for the first step and $W_t = 5$ for the second step. In particular, the run-time cost of each global averaging step can be observed in Figure 1.1, which consists mainly of two components: the time taken

to calculate the local model and the delay in communication with the central server. The computation time per global averaging step in synchronous FedAvg is determined by the slowest client device, while the shared bandwidth among all client devices influences the communication delay. Therefore, although numerous client devices can help accelerate model training, the computation and transmission times will increase significantly when W_t is large. Similarly, a shorter communication period T helps in the model's convergence, but the communication delay cost will increase compared to the computation time. Hence, optimizing the FedAvg algorithm involves a complex trade-off between

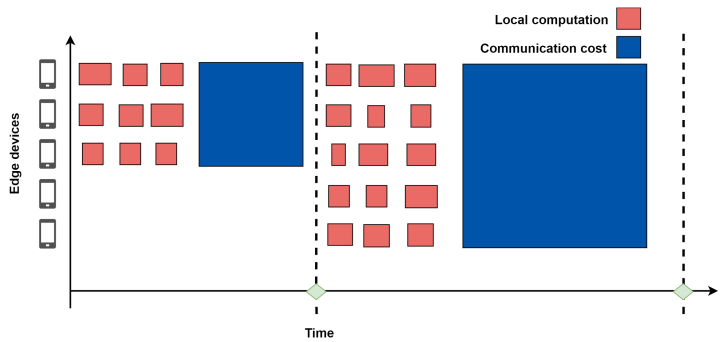


Figure 1.1: An example of the runtime expense of FedAvg algorithm, where the red and blue blocks indicate the time taken for local computation and transmission time, respectively.

the number of client devices and the transmission time. Furthermore, FL systems are frequently characterized by high dynamic due to the participation of new client devices, as well as the continuous generation of data by existing devices with limited network bandwidth. These limitations pose a considerable communication cost challenge in FL. It directly impacts the effectiveness, scalability, and overall performance of the FL process, making it a crucial area of concern.

In recent years, several techniques have been proposed to improve communication efficiency in such a context. For instance, one potential way is data compression, like quantization and sparsification methods, which are used to directly decrease part of the data size. More details about compression methods are introduced in Section 3.1.2. However, these methods often cause heavy performance, especially when a high compression ratio is required. Furthermore, compression of global model updates could also degrade the model's ability to handle the diversity of decentralized data [15].

Another method commonly proposed is to minimize the number of transferred updates to the server [11]. Since the exchanging of large model updates requires significant communication resources, it is vital to reduce the volume of data that has to be sent to the server [11, 12, 16]. Our proposed FL models fall

into this category. These have been explored in **Papers I** and **IV** focused on decreasing the volume of transferred data.

- **Data Heterogeneity**: In ML settings, the data is assumed to be independently drawn from the same joint distribution. This is known as the data is Independently and Identically Distributed (IID). However, when we move to FL, we quickly face violations of this assumption. To be more precise, FL involves training a model on a global scale using multiple distributed devices that might collect unique dataset and possess different data distributions. These distributions, known as Non-Independently and Identically Distributed (Non-IID), reflect real-world applications [17, 18]. Non-IID data represents one of the key challenges in FL [19, 20].Particularly, Non-IID data implies that the datasets may vary in size and distribution, making it hard to fit all local datasets with one global model. Moreover, the presence of Non-IID data may lead to client drift [21]. This phenomenon can considerably undermine the performance of FL [22, 23]. The impact of client drift on IID and Non-IID data is shown in Figure 1.2. In the FedAvg approach, the server updates gradually converge toward the average of client optima. In the case of IID data, the average of client updates (e.g., in the case of two clients) $\mathcal{M}^1$ and $\mathcal{M}^2$ is close to the global optimum $\mathcal{M}$. Therefore, the direction of the average model is also similar to that of the global model. However, in the case of Non-IID data, the global optimum $\mathcal{M}$ is far from the true local optima. In this example, $\mathcal{M}$ is closer to $\mathcal{M}^2$. Therefore, the global model deviates from its true global optimum direction for Non-IID data. In other words, the averaged model (global model) $\mathcal{M}_{t+1}$ in round $(t+1)$ will be far from the true global optimum. Furthermore, the divergence of the model may increase with successive communication rounds (t).

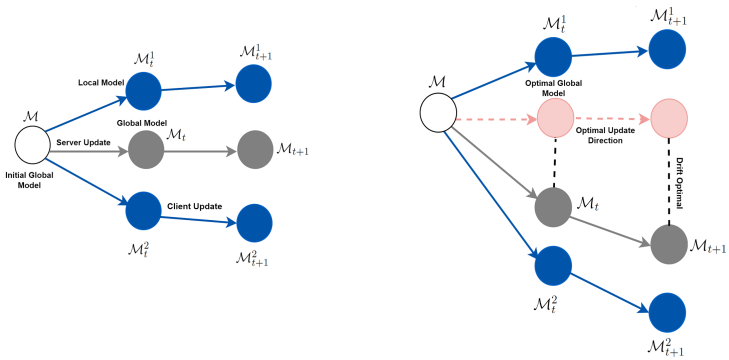


Figure 1.2: Visualization of client drift in FedAvg algorithm for two clients with two local epochs on different structured data: (a) IID data setting; (b) Non-IID data setting.

Recent works [22–25] have shown that such data heterogeneity in FL approaches

can significantly degrade the performance of the global model, which could eliminate the main motivation of FL training. While tradtional FL approaches pursue a global optima of all client devices, the concept of personalized FL seeks to learn personalized models for each task or specific device [26, 27] that fit the diverse local datasets. The challenge of data heterogeneity is addressed in **Paper V**.

- **Client Selection**: In the FL training process, the naive FedAvg algorithm employs a strategy of randomly choosing clients in each training round, which negatively affects the performance of the federated model, and causes failure to fully utilize local updates from heterogeneous clients [5]. Meanwhile, the computing capacity and communication resources of the participant devices in FL vary and they may be reluctant to take part in FL training. Therefore, it is essential to have an effective mechanism to select participants in the FL systems [28, 29]. This challenge has been addressed in **Papers I, II**, and **IV**.

- **Personalization**: The general FL approach creates a single global shared model by taking the average of all local models of client devices [30]. It assumes that all client devices have the same learning task. Furthermore, certain clients may show poor performance while others perform well, inevitably eliminating client personalization and decreasing the ability to represent client characteristics [31]. However, in real-world applications, different devices involved may encounter distinct ML tasks and possess diverse data distributions that need to tailor specialized/ personalized approaches to meet their specific requirements. For example, vanilla FL does not personalize the model for each client device which is essential for each user on platforms such as YouTube and Netflix. On the other hand, participant devices may have different models with completely different architectures. These models can include Convolutional Neural Network (CNN) with 5 layers, CNN with 10 layers, ResNet, Random Forest, etc., as a result of diverse computational resources or distinct requirements [32]. Therefore, while there are benefits to using personalized FL for personal learning, it does not fully take advantage of the potential of collaborative learning. The limitation poses a challenge for two main reasons: First, most client devices have limited data size. Second, even though there are variations among clients or tasks, it is reasonable to assume that there is some level of similarity among them. The challenge of FL personalization has been studied in **Paper V**.

- **Security and Privacy**: Traditional FL addresses the data security issues that arise from the need to centralize client datasets on a central server for model training. Although traditional FL ensures significant data privacy compared to centralized learning. Recent works indicate that FL applications are still vulnerable to numerous attacks. These attacks can have adverse effects on

various aspects, such as the precision of the learned model, confidentiality, integrity, and availability of the data used [33, 34]. In FL scenarios, the privacy of the input data is ensured by transferring only trained parameters instead of the original raw data. However, model updates in training can accurately infer valuable information [35]. Hence, even though there have been enhancements in comparison to centralized methods, ensuring data security and privacy in FL remains crucial issues that need to be addressed. It is important to note that the privacy aspect will remain beyond the scope of this thesis.

1.2 Thesis Scope and Objectives

This thesis aims *to develop new resource-aware and personalized FL models by using clustering analysis*. These new solutions pursue to improve the resource efficiency and robustness of personalized FL when dealing with heterogeneous and dynamic data.

The thesis goal is achieved by addressing the following objectives:

1. *To develop FL resource-aware solutions based on clustering analysis*.
2. *To develop groups' personalized FL models to address data heterogeneity*.
3. *To investigate the recent advances in context awareness for sensor networks using AI*.

1.3 Research Questions

This thesis investigates three distinct directions: resource-conscious FL solutions, personalized FL solutions, and the exploration of edge-based AI for Sensor Networks (SNs). Based on the scope and objectives established, the following research questions are formulated and addressed in this thesis. The visualization of the included studies and their connection with the aim, objectives, and research questions is given in Figure 1.3.

RQ 1: *How can we develop FL models that reduce resource consumption without sacrificing the model performance?*
Motivation: In practical federated systems, the participating devices often have limited resources such as cache memory, storage, network bandwidth, and processing capabilities. Reducing resources is crucial to achieving cost-effective training in FL. In particular, when dealing with a fleet of diverse client devices that differ in data quality, computational capacity, and battery lifetime levels. Several studies [36–38] have been conducted to minimize the resources of client devices in FL. Therefore, it

is prudent to develop effective methods in a FL system that considers the constraints of the device resource.

Papers: In **Paper I**, an approach is proposed that applies clustering analysis to bring efficiency to FL communication. Only representative updates of each cluster are uploaded to the server to mitigate communication overhead. The main contribution in **Paper II** is the multi-criteria selection of the cluster representative into the Wireless Networks (WNs) environment. In this study, various factors such as energy consumption, bandwidth usage, and accuracy are taken into account when choosing a representative of a cluster to communicate with a central server. This study examined how these factors affect FL performance. In **Paper IV**, we have extended the work presented in **Paper I** by improving the optimization of model aggregation and minimizing communication overhead. This is achieved by implementing clustering optimization to select representatives. Additionally, the split optimization technique is utilized to update and enhance the overall clustering solution.

RQ 2: *How the clients' behaviour can be efficiently evaluated during the FL process?*

Motivation: Numerous applications illustrate that FL is an effective solution for making collaborative decisions while maintaining data privacy. Studies in FL aim to explore different approaches to achieve performance levels similar to centralized models. One such approach involves measuring the client's contribution. Identifying high-quality data for frequent use and removing low-quality data are other essential pre-tasks to achieve a high-performing FL [39]. Similarly, in federated environments, not all clients have the same level of significance; for example, clients with highly biased datasets [21] or clients with noisy labeled datasets [40] might disrupt the optimization procedure of the federated model. However, most of the current solutions that measure the contribution of clients require considerable computational resources and time [41]. Therefore, a robust and effective approach is needed to assess the contribution of the client in FL without excessive use of time and resources.

Papers: As stated above, most of the existing methods that assess client contribution require substantial resources and are usually carried out as an additional evaluation procedure. This results in a high computational burden for data owners with large datasets. **Paper VI** answers this research question using the FL models proposed in **Papers I** and **V**. Although the main focus of **Papers I** and **V** are not on clients' behavior evaluation, we have noticed that these models could be used to assess the contribution of the client. In **Paper I**, a CA-FL approach was introduced, which can identify clients with reliable behavior throughout the training process. In contrast, **Paper V** presented a GP-FL technique that can detect unreliable behavior in clients during training. As a result, the evaluation of client behavior can be used to measure the contribution of clients.

RQ 3: *How we can personalize FL models to achieve robust model performance?*

Motivation: Despite the significant success of FL, it is built upon the assumption that all participants have similar data distributions [33]. However, in real-world scenarios, there may be considerable differences between data distributions (e.g., Non-IID), which makes it challenging to achieve federated collaboration. In FedAvg (the first FL method), all participants jointly train a global model [5, 33] while protecting their privacy. However, a single global model may not generalize well to new participants. Furthermore, given the wide range of application scenarios, each client device may need to build a dedicated local model based on specific characteristics and data, which cannot be satisfied in the existing FL setting. Although personalized FL greatly benefits personalized learning, it does not take advantage of collaborative learning between devices. This limitation raises an issue due to the possibility of a small amount of data in each client device and not leveraging the similarity between devices in tasks or data. Therefore, its existing limitation requires developing a new personalized FL model.
Papers: This research question is addressed in **Paper V**, which presents a clustering-based approach for group-personalized FL in the context of Human Activity Recognition (HAR) applications. The FL model proposed in this study tries to cope with a statistically heterogeneous FL setting by introducing a group-personalized FL (GP-FL) solution. The proposed GP-FL algorithm generates several global ML models, each of which is trained iteratively on a dynamic set of clients with similar class probability estimations.

RQ 4: *What AI-based solutions are underrepresented in the recent state-of-the-art of context-aware edge intelligence systems?*
Motivation: Context-aware systems must have a refined understanding of the environment surrounding them and be able to make appropriate updates to adapt to different contexts. From this perspective, applying AI techniques to context-aware systems effectively enables such systems to process complex behaviors and adapt to rapidly changing situations in real time. This research question aims to identify AI-based solutions in context-aware systems, particularly within SNs. Furthermore, the aim is to determine any research gaps in current state-of-the-art AI-based solutions.
Papers: In **Paper III**, we carried out a comprehensive literature review on the application of AI methods in context awareness, specifically in WNs. This study investigated the existing literature to provide an overview of various domains, highlight the main challenges within each field, outline the research's motivation, and identify gaps in current studies.

1.4 Thesis Outline

The remaining sections of this thesis are organized as follows:

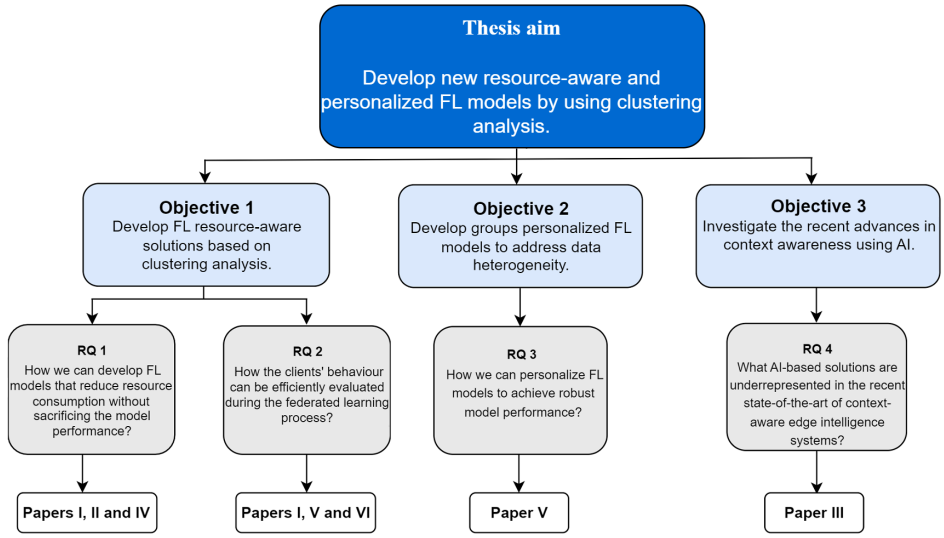


Figure 1.3: A visualization of the relations among the thesis aim, objectives, research questions, and included studies.

Chapter 3 - Background: We elaborate on the relevant background information that serves as the foundation for our thesis.
Chapter 3 - Related Work: This chapter discusses the relevant studies in previous research.
Chapter 4 - Research Methodology: The datasets and baselines used in our studies are presented in this chapter. In addition, we provide details about the evaluation measures used. We proceed with the examination of the research methodology used to conduct our studies. Lastly, we address potential threats to research validity.
Chapter 5 - Results and Discussion: This chapter discusses and analysis the results of the thesis. Each research question is addressed with the corresponding papers.
Chapter 7 - Conclusions and Future Directions: We present the conclusions of the thesis and discuss future research directions.

2 Background

This chapter provides the essential background information necessary to understand the remainder of the thesis. In particular, we introduce ML and focus on the FL method, specifically highlighting the FedAvg method and Non-IID Data. Following that, we discuss cluster analysis and cluster validation measures and conclude the chapter with typicality and eccentricity data analytics.

2.1 Machine Learning

In recent years, there has been significant interest from various fields in the rapid advancement of AI and, in particular, ML. ML tools can accurately analyze large datasets, revealing insights that humans might find difficult to discover. ML algorithms can be broadly classified into three main groups: Supervised learning, Unsupervised learning, and Reinforcement learning (RL). The advantages and disadvantages of each category vary depending on the available data and the purpose of the application. Furthermore, the advancement in IoT devices and ML have led to the generation of large amounts of data, commonly known as big data [42]. In order to gain insight from big data, it is necessary to collect, store, process and analyze the data in a central location, where large datasets from multiple devices are aggregated on a central server [43].

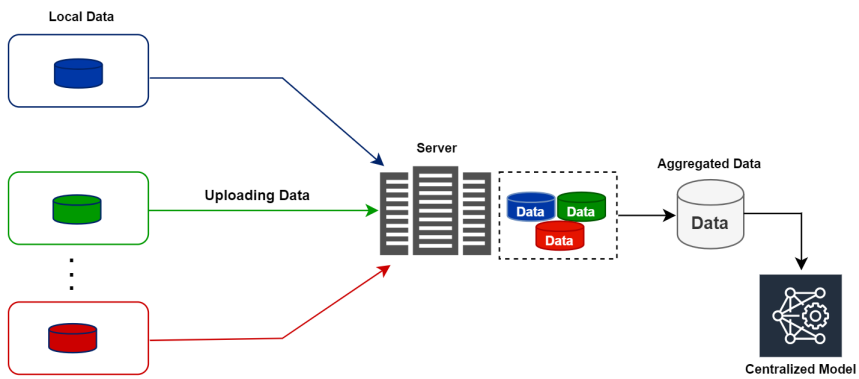


Figure 2.1: General framework for a centralized training approach.

Figure 2.1 clearly shows that data must be collected from different clients to train a joint model. Despite its impressive success, several issues must be highlighted. In fact, data privacy is violated when sensitive data is transferred to a central server. Moreover, uploading large chunks of data can also create a huge load on the centralized network and put a huge processing load on a single service provider during joint model training [44].

2.2 Federated Learning

Distributed learning algorithms are designed to address computational challenges that arise when dealing with complex algorithms on large-scale datasets. In contrast to centralized machine learning, distributed machine learning algorithms offer improved effectiveness and scalability. In the distributed learning scenario, a model's training occurs on multiple devices rather than being centralized in a single location, using a dataset. During training in a distributed algorithm, participants independently train their models and send updates to the central server, where they are averaged [45]. The concept of FL was first proposed by Google in 2017 [5, 46] as a type of distributed machine learning approach that allows training a joint model by cooperating with clients without revealing training data under centralized server supervision. The main idea of FL is to cooperatively train ML models among numerous independent clients (e.g. mobile phones, wearables, computers, sensors, and IoT devices) under the constraint that training data must remain stored and processed locally in an agreed setting. Instead, the training of the shared model is performed by the clients on their local datasets. Updated models are then sent to the central server, which aggregates these trained local models to produce a unified model, in contrast to traditional centralized machine learning methods [47]. The updated model is then returned to the clients for another communication period. This process continues until a stopping criterion is met.

FL ensures data privacy and offers greater scalability compared to centralized learning methods, as it does not involve exchanging raw data between clients. In FL, multiple number of clients share a global ML model. Each edge device receives a replica of the shared model and improves it through local learning using its private dataset. The edge device then sends the updated model to the server, which aggregates them to produce the global model. By utilizing the resources of clients, FL introduces a transition from expensive centralized ML training to a distributed approach [48].

- **Federated Averaging (FedAvg)**: The FedAvg is a foundational algorithm in FL widely used for federated aggregation. FedAvg is built around Stochastic Gradient Descent (SGD) to solve problems involving Non-IID data. FedAvg involves a central server and a randomly selected subset of clients W_t, (i.e.

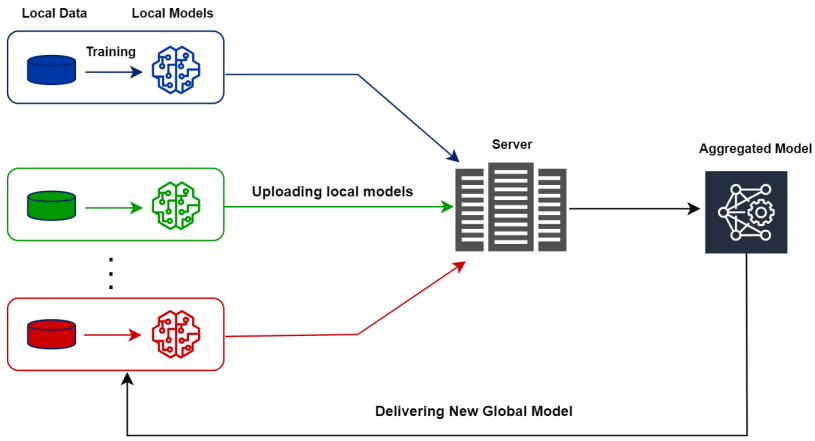


Figure 2.2: General working process of standard FL.

$W_t \subset W$). Each client device in the federation uses its local dataset $\mathcal{D}_i$, and n_i is the size of the data set $\mathcal{D}_i$ (i.e., $\mid \mathcal{D}_i \mid = n_i$), where $i = 1, 2, \ldots, N$ denotes the index of the client device involved in FL. Figure 2.2 illustrates a flow diagram of the standard FL approach. To provide background for the proposed methods, this section presents FedAvg algorithm [5]. In addition, FedAvg will serve as a reference for the experimental analysis that was conducted. Generally, it can be summarized as follows:

Initialization:

Step 1: The central server and edge device $W_t \subset W$ are initialized, and the server generates an initial global model $\mathcal{M}_0$ based on the small amount of available data.
Step 2: The central server distributes the global model $\mathcal{M}_0$ to all participating devices $w_i \in W$.

Local Training:

Step 3: Each edge device $w_i \in W_t$ performs mini-batch SGD with a local training dataset $\mathcal{D}_i$ in parallel and updates the model for a total of E epochs as follows:

$$\mathcal{M}_{t+1}^i = \mathcal{M}_t^i - \eta g(\mathcal{M}_t^i), \tag{2.1}$$

where t is the index of communication round, η is the learning rate, and

$g(\mathcal{M}_t^i)$ refers to the stochastic gradient, which is calculated as follows:

$$g(\mathcal{M}_t^i) = \frac{1}{N_{w_i}} \sum_{\mathcal{D}_i} \nabla \ell(\mathcal{D}_i; \mathcal{M}_t^i) \cdot \tag{2.2}$$

Global Aggregation:

Step 4: Each edge device $w_i \in W_t$ uploads the updated model $\mathcal{M}_t^i$ (called the local model) to the central server.
Step 5: Once all local updates have been received, the central server proceeds to initiate the global aggregation process. The updated global model is obtained using the average weighted aggregation method, as specified by the following:

$$\mathcal{M}_{t+1} = \mathcal{M}_t + \frac{\sum\limits_{w_i \in W_t} p_i \mathcal{M}_{t+1}^i}{\sum\limits_{w_i \in W_t} p_i}, \tag{2.3}$$

where p_i is the relative weight of client w_i. This updated global model is also used as a starting point for the next communication round. However, the model weights are averaged in the traditional FedAvg framework. Steps 2-5 are repeated until the entire FL process stops after t rounds.

The Algorithm 1 provides pseudo-codes for the FedAvg algorithm, which involves the participation of a set of client devices denoted as $W_t \subseteq W$, as stated in [49]. The structure of the model of both the global model $\mathcal{M}_t$ and all local models $\mathcal{M}_t^i$ is identical, with different values of the model parameters (where t represents the communication round). Under this assumption, direct model aggregation can be implemented as described in line 7 of Algorithm 1, in which each local model uploaded $\mathcal{M}_t^i$ in E local epochs using a learning rate η. Generally, to improve the performance of the FL system, it is often important to select an appropriate setting of four hyperparameters, $W_t \subseteq W$, E, $\mathcal{B}$, and η, based on the specific task that can be determined by metaheuristic search methods [50, 51].

However, Algorithm 1 highlights that the global model $\mathcal{M}$ and local updates $\mathcal{M}^i$ need to be frequently downloaded and uploaded, as indicated in lines 5 and 6. This process consumes a significant amount of communication resources compared to those typically required for standard centralized learning. Furthermore, FedAvg has established that the more heterogeneous the data, the longer FedAvg takes to converge [21, 22]. Accordingly, a robust and efficient aggregation strategy is crucial to the success of FL.

Algorithm 1 FedAvg. W is the total number of client devices; T is the total number of global rounds, E is the total number of local training epochs, $\mathcal{B}$ the local mini-batch size and η is the learning rate.

Input: Initial shared model $\mathcal{M}_0$, set of clients $W_t \subseteq W$, the number of iterations T
Output: The FedAvg procedure global model $\mathcal{M}_t$ for T iterations

procedure FEDAVG($\mathcal{M}_0, W_t \subseteq W, T$)
  $t \leftarrow 0$
  while $t \leq T$ **do**
    $t \leftarrow t + 1$
    $\forall\, w_i \in W_t$, the server exec SEND($w_i, \mathcal{M}_t$)
    Each $w_i \in W_t$ exec CLIENTUPDATE($w_i, \mathcal{M}_t$)
    $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$ ▷ global update, (Eq. 2.3)
  end while ▷ *S*topping criteria is met
end procedure
function CLIENTUPDATE($(w_i, \mathcal{M}_t)$)
  B = (split $\mathcal{D}_i$ into batches of size $\mathcal{B}$)
  RECEIVE($w_i, \mathcal{M}_t$)
  for each local epoch i from 1 to E **do**
    for each batch $b \in B$ **do**
      $\mathcal{M}^i_{t+1} \leftarrow \mathcal{M}^i_t - \eta g^i_t$ ▷ *Local update*, (Eq. 2.1)
    end for
  end for
  SEND($i, \mathcal{M}^i_{t+1}$)
end function

- **Non-IID Data**: In most of ML and data science scenarios, it is generally assumed that the data are independently sampled from a similar joint distribution. This is known as assuming that the data are IID. To generalize the population from which the data is drawn, it is necessary to consider that each data point is independent of the others and that the population remains unchanged as data points are gathered (identically distributed). In other terms, the data is more uniform [52]. However, in a FL scenario, the devices involved in training are typically IoT devices that produce data distribution that is unstructured and highly random with each other. More specifically, client devices may have different label distributions, and some labels may be more available on some devices than others.

  This phenomenon is referred to as Non-IID, which can lead to significant model divergence [53, 54]. In the context of supervised learning on a specific device i, let's consider a data sample (x, y), where x represents the features and y denotes the labels, follows a local data distribution $P_i(x, y)$. Non-IID

refers to the situation where P_i varies from one device to another. Although McMahan et al. [5] argue that FedAvg can handle Non-IID data to some extent, numerous studies have suggested that a deterioration in FL accuracy is almost inevitable when dealing with Non-IID [23]. Different types of Non-IID Partitions have been introduced based on features x, labels y, and other more complex FL scenarios. The attribute skew, the label skew, and the temporal skew represent the main Non-IID Data categories. The *label skew* is investigated in our thesis studies; specifically, different degrees of *label distribution skew* are studied in our papers, where the label distributions $P_i(y)$ on the clients are different. We control the skewness by controlling the fraction of data that is Non-IID. For example, 0% non-iid would mean that the data labels are uniformly/evenly distributed between clients. 30% of the data is Non-IID with 2 clients and 2 labels would imply that one client has at least 30% of one label, while the rest is evenly distributed.

2.3 Clustering Analysis

Clustering analysis plays a crucial role in the research and application of data mining. It is an active research topic that has been applied in various fields, including data science, and statistics [55–57]. Clustering analysis is a traditional unsupervised classification method that aims to uncover the inherent structural characteristics and patterns of the data and label the data to reveal potential information [58]. Clustering analysis divides datasets into multiple categories, reducing the dissimilarities between data in the same group and increasing those between data in different groups. This section introduces the various components necessary for conducting a clustering analysis approach. In this section, we begin by discussing clustering techniques, specifically focusing on partitioning algorithms. Towards the end of the section, we also introduce similarity measures.

Nowadays, the real world is full of a huge amount of Big Data as a result of the continuous increase in the volume of data every day. Thus, clustering techniques can be employed to uncover interesting patterns within these massive datasets, even with little or no background knowledge [59]. The clustering of client devices is an essential part of our proposed FL models, which involves grouping devices with similar characteristics. This enables the participating devices to take advantage of collaboration with other devices that exhibit similar learning traits [60, 61]. This is advantageous in contrast to naive FL training, where irrelevant devices contribute to each other, potentially harming their respective datasets' performance. Hierarchical clustering, centroid-based clustering, and density-based clustering are the three most widely used clustering techniques. These algorithms are represented by agglomerative clustering, k-medoids clustering, k-means clustering, DBSCAN, density peak clustering, etc. [62].

Two clustering algorithms, k-medoids [63] and Markov Clustering (MCL) [64], are used in our studies due to their relevance to FL settings. k-medoids algorithm is a partitioning algorithm that has an input parameter, the number of clusters, that should be determined in advance. In contrast, MCL does not need to determine the number of clusters ahead, but it has a parameter called "inflation" that indirectly affects the precision of the clustering. For example, increasing the inflation parameter results in a higher number of clusters.

- **k-medoids Clustering**: k-medoids clustering is a modified version of the k-means-based clustering method that is more robust to noise and outliers. Instead of selecting the mean data point as the cluster center, k-medoids clustering chooses an actual data point within the cluster to represent it. The k-medoids is the data point within a cluster that is located at the center and has the lowest sum of distances to all other points [65]. In this algorithm, we initially select k data points and iteratively move towards the points in the best cluster. We then examine all possible combinations of data points and assess the clustering quality for each pair of points. If a data point is found with the most enhanced distortion function value, it will replace the current best data point. The newly created optimal data points form the enhanced medoids. This algorithm aims to minimize the dissimilarities between data points and their reference points.

  Given a finite set of initial data points $P = \{p_1, p_2, \ldots, p_n\}$, $i = 1, \ldots, n$, we need to split into k disjoint clusters. k-medoids selects k medoids (data objects) $C = \{ob_1, ob_2, \ldots, ob_k\}$ from the given set P, to minimize the objective function known as the absolute error function (E) given in Eq. 2.4:

$$E = \sum_{j=0}^{k} \sum_{p \in c_j} \mid p - ob_j \mid, \tag{2.4}$$

  where E represents the sum of the absolute error. The variable p, which belongs to the set P, represents a data point corresponding to a point in the cluster C_j from the set of clusters C. Additionally, ob_j denotes the representative data point (object) of the cluster C_j. Therefore, medoids (also referred to as client devices in our studies) are chosen from actual data point to serve as cluster representatives. It is important to note that the Euclidean Distance (ED) is used in k-medoids in our studies.

- **MCL**: It is an efficient graph-based clustering algorithm. Unlike the k mean and k medoids, which are partitioning algorithms, this algorithm does not require prior knowledge of the number of clusters. This clustering algorithm is widely used in bioinformatics for clustering protein sequences and co-expression

data of genes. In addition, this algorithm is well suited for distributed computing [64]. The MCL procedure involves two operations performed on stochastic matrices, namely *Expand* and *Inflate*. The expansion of matrix M is defined as the result of multiplying M by itself, that is, $M * M$. On the other hand, the inflation operation Inflate (M, r) involves raising each entry in the matrix M to the power of the inflation parameter r (where r is greater than 1, typically set to 2) and then normalizing the columns so that they sum up to 1. This operation is expressed as follows:

$$M_{inf}(i,j) = \frac{M(i,j)^{rM}}{\sum_{k=1}^{n} M(k,j)^{rM}}. \tag{2.5}$$

Next, we assign the matrix M_{inf} to M. MCL was used in this thesis to divide client devices with similar empirical probability vectors into similar groups.

In data analysis, similarity measurements are used to discover similarities or dissimilarities between data points [66, 67], generating valuable findings within large datasets. Moreover, these terms are often used in clustering techniques when data points are split into clusters or to determine the similarity between points within a given cluster [68]. Centroid-based algorithms represent a notable example of such methods. The selection of a distance metric (dissimilarity) significantly impacts the effectiveness of the ML clustering. Therefore, how distances are calculated between date points is a critical factor in determining the performance of the clustering algorithm. The distance measures used in our studies are formulated in Table 2.1.

Table 2.1: Distance measures.

Measure	Equation
Euclidean [69]	$D_{Euc}(p,q) = \sqrt{\sum_{i=0}^{n}(p_i - q_i)^2}$ where p, q is two data points in the Euclidean n-space, q_i, p_i are Euclidean vectors, and n is n-space.
Jaccard [70]	$Sim_{jac}(A,B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$ where A and B are two finite sets, and $\lvert A \cap B \rvert$ is the size of the intersection and $\lvert A \cup B \rvert$ size of the union of the sample sets.
Wasserstein [71]	$D_{was}(X,Y) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}$ where X and Y are probability distributions, m and n denote the points of X and Y, respectively, d_{ij} denotes the distance from the i point of X to the j point of Y and f_{ij} denotes the number of moves from i to j, $f_{ij} \geq 0, i = 1, \ldots, m, j = 1, \ldots, n$.

Identifying the appropriate number of compact and well-separated clusters is among the most challenging tasks in cluster analysis. Typically, cluster validity techniques are utilized to assess the quality of clustering solutions, with a specific focus on the compactness and separability of clusters.

2.4 Cluster Validation Measures

The cluster validity measures serve as the evaluation criteria for assessing the quality of the clustering results. To determine the best clustering strategy, the Silhouette Index (SI) [72] is used to evaluate the clustering results. SI assess clustering validity and detect compact and well-separated clusters [73, 74].

The quality of a clustering solution $C = \{C_1, C_2, \ldots, C_k\}$ can be evaluated using SI. Let a_i denote the average distance between the data point i and all other points to its own cluster, and let b_i denote the minimum average distance between the data point i and the points in all to another cluster. The value of $s(i)$ for item i can be calculated using the following formula:

$$s(i) = (b_i - a_i)/\max\{a_i, b_i\}\cdot \tag{2.6}$$

According to its definition, the value of $s(i)$ falls within the range of $[-1, 1]$. If $s(i)$ is close to 1, it indicates that the data point i is assigned to be 'well-clustered' On the other hand, if $s(i)$ is equal to 0 or close to 0, it suggests that the data point i lies between two clusters, making it unclear to which cluster it should belong. In this scenario, the data point can be considered as an 'intermediate case'. Lastly, when $s(i)$ is close to -1, it results in 'misclassification' of the data point i. The SI indicates which data points belong to their respective clusters and whether they are located closer to one cluster or in between clusters. In other words, it can provide information on the degree of separation between a specific cluster and the others.

The SI can also be computed for each cluster C_j ($j = 1, 2, \ldots, k$) of n_j data points using the following formula:

$$s(C_j) = \frac{1}{n_j} \sum_{i=1}^{n_j} s(i)\cdot \tag{2.7}$$

Furthermore, SI for the entire clustering solution C containing n items is calculated as

$$s(C) = \frac{1}{n} \sum_{i=1}^{n} \frac{(b_i - a_i)}{\max\{a_i, b_i\}}\cdot \tag{2.8}$$

2.5 Typicality and Eccentricity Data Analytics

TEDA, which stands for Typicality and Eccentricity Data Analytics (TEDA), is a statistical approach that utilizes the principles of typicality and eccentricity to categorize similar data observations. Instead of using the conventional concept of clusters, the data is organized into granularities known as data clouds. These data clouds are structures that exist within predefined shapes or boundaries [75]. In [76], novel principles for anomaly detection analysis have been presented, focusing on eccentricity. Building upon these principles, a new algorithm named AutoCloud is proposed in [77].

Eccentricity refers to the degree to which a specific data point differs from other points and from its cluster. In this context, the calculation of the eccentricity ξ^j for the data point i for a cluster of data C_j can be computed as [77]:

$$\xi^j(i) = \frac{1}{n_j} + \frac{(\mu_i^j - \hat{p}_i)^T(\mu_i^j - \hat{p}_i)}{\sigma_i^j}, \tag{2.9}$$

where n_j represents the size of C_j, $\hat{p}_i$ denotes the empirical probability vector corresponding to the data point i, μ_i^j is the mean and σ_i^j indicates the variance, assuming that i belongs to C_j. Eq. 2.10 demonstrates the utilization of eccentricity to determine the membership of a data point in a specific cluster.

In addition, the Chebyshev inequality has been used to apply a threshold to verify whether a data point remains part of a current cluster [78]. A specific data point i is considered to be a member of the group C_j if the following condition is met:

$$\xi^j(i) \leq \upsilon_j \text{ and } \upsilon_j = (m^2 + 1)/2n_j, \tag{2.10}$$

where the parameter m ($m > 0$) is defined by the user and directly affects the evaluation of the cluster, and υ_j is the threshold associated with the cluster C_j. Although it can be defined using multiple criteria, $m = 3$ is commonly used as a standard value, leading to satisfactory results for different datasets and configurations [79]. We utilize eccentricity analysis, similar to the approach used in AutoCloud, to maintain the clustering solution of client devices.

3 Related Work

This section focuses on previous work that tackles the reduction of resource consumption and personalization in the FL setting. For this reason, we categorize the study of previous research into two main categories, namely: resource-aware FL and Personalized FL. Figure 3.1 illustrates our proposed taxonomy of related work and the corresponding subgroups according to the approach used in the solutions.

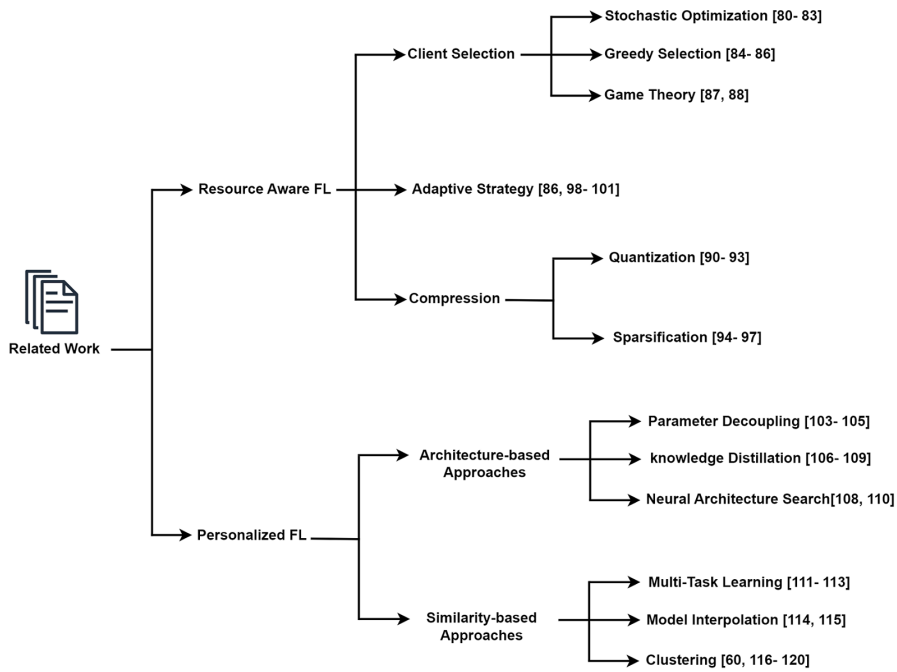


Figure 3.1: Classification of related work in the thesis.

3.1 Resource Aware Federated Learning

In the following, we will examine current research efforts closely related to reducing the resource consumption of participants in FL. As transmission models require a significant amount of communication resources, this aspect will be investigated

within existing research aimed at minimizing the communication resources needed by FL clients. Current research solutions to minimize communication costs in the field of FL can be classified into the following categories.

3.1.1 Client Selection

In FL, during the training phase, the communication bottleneck is exacerbated by the exchange of model updates between many participating clients. Using a random selection method, such as FedAvg, to choose a subset of clients is a viable approach. However, this randomness can lead to a significant number of missed potentials. In most FL implementations, the clients differ in terms of their design and capability. This diversity also extends to the quality of the communication mediums used. By selecting clients with the most favorable communication conditions in each round, it is possible to increase the average data rate and reduce communication costs.

- Stochastic Optimization: Chen et al. [80] select clients that achieved the most optimal probability sampling for clients during each round of communication. In [81], the proposed algorithm selects a subset of clients with a higher global loss value that have a higher chance of being selected. FLOB framework identifies a subset of clients that achieve the minimum global loss using biased stochastic optimization [82]. AdaFL assigns more importance to clients with a higher probability value, which is determined by the difference between the local updates tensor and the global one [83].

- Greedy Selection: The problem addressed by Balakrishnan et al. in [84] introduces FedAvg with Diverse Client Selection (DivFL), which is a greedy approach. FedMCCS [85] takes into account various criteria when selecting clients, including training time, memory size, CPU capacity, and energy consumption during training. To achieve early convergence without increasing communication costs, Wang et al. [86] introduce a method called Communication-Mitigated Federated Learning (CMFL).

- Game Theory: Le et al. [87] defined the problem of FL using the concept of an auction game. The clients act as bidders and the central server acts as the auctioneer. The objective is to select a client whose payoff is the client's selection. The authors in [88] developed a similar auction-based incentive strategy to efficiently choose clients in the FL context.

3.1.2 Compression

Compression techniques play a crucial role in efficient utilization of edge resources in FL. Compression techniques aim to reduce the size of the data and facilitate the

exchange of models between client devices and the central server [89]. In the context of FL, compression is particularly advantageous for handling DL models due to the typically large size of the transferred updates. Listed below are a variety of techniques that can be employed to compress FL models:

- Quantization: Quantization can be used to represent models as integers with reduced precision, instead of transmitting high-precision floating points [90]. QSGD algorithm in [91] balances the trade-off between convergence and quantization levels to minimize the communication cost. TernGrad algorithm in [92] involves quantizing floating-point numbers. CosSGD algorithm in [93] utilizes the cosine function to allocate a finer quantization space for values with more significant gradients.

- Sparsification: Sparsification refers to the selective transmission of partial gradients, which reduces communication costs by discarding gradients with small contributions [94]. The authors use a constant compression rate to select the transmitted gradients in [95]. The work in [96] introduces Sparsifying gradients using fixed proportions of positive and negative gradients. LAG sparse communication algorithm adaptively computes a threshold in each round of communication to minimize part of the transmission of gradients [97]. However, existing methods usually apply compression operations on the client side, which requires additional processing for both encoding and decoding. Furthermore, these compression strategies usually lead to a significant decrease in performance when the compression ratio needs to be very large [15].

3.1.3 Adaptive Strategy

Several communication strategies can be adapted to reduce communication and resources in the FL settings such as those induced by [86]. Communication-mitigating FL framework(CMFL) was proposed, which enables the transmission of only relevant local updates to the server. This method not only speeds up convergence but also reduces the number of communications needed. In [98], the authors proposed a method to improve convergence analysis by advocating the use of an optimal and unbiased sampling technique. The authors in [99] proposed a method to adaptively aggregate partial models in FL using RL. This strategy aims to optimize the selection of the client devices involved. Wu et al. [100] introduced an adaptive aggregation using the topK strategy to identify the top clients with minimal losses to update the model parameters in each communication round. Similarly, Asad et al. [101] introduced a filtering method on each local update that allows the transmission of only the important gradients. Although the aforementioned methods reduce communication costs, they also lead to inefficient utilization of computational resources for clients that perform local training without involvement in model aggregation. Our FL proposed

models in **Papers I, II, IV,** and **VI** lie in this category, which introduces a new line of approaches for efficient communication FL that are orthogonal to most of the current FL methods.

3.2 Personalized Federated Learning

In this section, we examine personalized FL methods that focus on developing personalized models. Inspired by the classification of personalized FL in [102], the strategies were categorized into Architecture-based Approaches and Similarity-based Approaches in the following manner.

3.2.1 Architecture-Based Approaches

Architecture-based personalized FL approaches for personalized learning aim to achieve customization by tailoring a model for each client. The following sections will explore these techniques to achieve personalized FL:

- Parameter Decoupling: Parameter decoupling is the process of decoupling the specific parameters of the private model from the parameters of the global FL model. The Private parameters are trained on the clients' devices without being transmitted to the FL server. This allows the learning of task-specific representations to improve personalization. The authors in [103] introduced an approach that involves splitting model layers into global and personalized parts. These parts are updated separately, with more frequent exchanges of parameters in important layers. Furthermore, in a related work [104], authors presented a layer-wise personalized FL method. It enhances the aggregation of personalized models by taking into account the significance of individual layers from various clients. Arivazhagan et al. in [105] introduced the concept of "base layers + personalized layers", in which clients keep their personalized layers to learn task-specific representations while sharing the base layers with the server.

- knowledge Distillation: The technique of Knowledge Distillation (KD) allows the transfer of knowledge from one model to another. This procedure involves training the local model to copy the behavior of the more complex model. The authors in [106] introduced a distillation term to the local objective function to train the local model using the output of the global model. In the work of Jeong et al., [107], they utilize the a KD technique to allow individual clients to determine statistical differences between their local models, leading to better personalization and increased effectiveness within the FL setting. Jin et al., in [108] proposed a method that allows the retrieval of personalized knowledge for new clients by enabling them to distill the knowledge derived from past per-

sonalized models into their current local models. The authors proposed a decentralized FL that involves mutual knowledge sharing among clients in [109].

- Neural Architecture Search: Neural Architecture Search is an approach employed to autonomously discover efficient NN structures for a specific task. This method includes exploring a range of potential network architectures and choosing the optimal design according to a predefined objective, such as improving accuracy or reducing computational resources. The authors introduced a new approach named federated classifier averaging (FedClassAvg) for personalized FL in [108]. FedClassAvg allows clients with heterogeneous NN architectures to participate in collaborative training without sharing sensitive data, all while maintaining communication efficiency. Wan et al. [110] present a new approach known as Federated Modular Network (FedMN) for personalized FL. The FedMN technique involves assembling diverse neural architectures by selecting sub-components from a module pool, specifically tailored to the distinct characteristics and requirements of each client.

3.2.2 Similarity-Based Approaches

Similarity-based approaches focus on achieving personalization by representing client relationships. A personalized model is trained for every client and clients with similarities learn similar models.

- Multi-Task Learning: The objective of Multi-Task Learning (MTL) is to develop a model that can jointly perform multiple related tasks. This enhances generalization by utilizing domain-specific knowledge across the various learning tasks. By considering every FL client as a task within MTL, it is possible to capture the relationships between clients based on their heterogeneous local data. The MOCHA algorithm [111] was introduced to extend the distributed MTL to the FL context. It employs a primal-dual formulation to enhance the optimization of the learned models. MOCHA develops a personalized model for each FL client. [112] introduced the VIRTUAL federated MTL algorithm, which conducts variational inference through a Bayesian method. Huang et al. [113] proposed a method for pairwise collaboration among FL clients with similar data distributions was introduced.

- Model Interpolation: In the work by Hanzely et al., [114], a novel approach has been introduced to train personalized models using a mixture of global and local models to balance generalization and personalization. A penalty parameter is used to incentivize private models to be similar to the average model. Adaptive personalized FL algorithm was introduced by the authors in [115] to find the best combination of global and private models. A mixing parameter was introduced for each client, which is adaptively learned to control the weights of

the global and private models. This allows the optimal level of personalization for each client.

- Clustering: Clustering involves grouping similar participants into groups to enhance the efficiency of FL. In FL, the data are distributed among various clients, and each client trains a local model using its own data. Clustering can help by clustering participants that have similar data or similar private models. Several recent studies focus on clustering for FL personalization. Sattler et al. [116] group FL clients based on the similarity measures of local models to train multiple global models instead of a single model. Clients with similar models are grouped together in the same cluster, and only models within the same cluster are aggregated on the global server. The authors in [60] apply a hierarchical clustering-based FL method to gather clients based on the similarity of their local updates. The approach forms a set of groups, each comprising a group of clients with similar data.

  Ghosh et al. [117] introduce an iterative Federated Clustering Algorithm (IFCA), which groups clients based on the similarity of data distribution. This approach allows them to collaboratively train a shared model within their group. The authors in [118] propose a FL cluster approach named FedGroup to achieve personalization by modeling the similarity of different clients. The CFedPer approach in [119] includes a pre-start phase for grouping clients and an in-training phase comprising a base layer and a personalization layer. The authors in [120] introduce a personalized FL framework, which identified clients who share similar data distributions for clustering, followed by a co-distillation within the cluster to allow personalized FL.

  However, incorrect clustering can lead to degradation of the efficiency of the system in FL algorithms. In addition, clustering involves high computational and communication expenses that limit the practical applicability in large-scale settings [60, 116]. The proposed FLmodels in **Papers V** and **VI** fall into this type, paving the way for a new line of strategies for group-personalized models.

4 Methodology

This thesis introduces new resource-aware and personalized FL models through clustering analysis. These new solutions aim to improve the efficiency and robustness of FL resources in the face of diverse and evolving data. This chapter introduces the datasets and baselines used in the papers collected in this thesis. In addition, the evaluation measures used to assess and validate the proposed FL solutions are described. The research methodology used in this thesis is then elaborated. Finally, the chapter concludes by presenting the validity threats of the conducted studies.

4.1 Datasets

The datasets used to evaluate the FL models in this thesis are outlined in this section. A combination of synthetic data and publicly available real-world data has been used. In particular, we have evaluated the proposed FL models on ten different datasets: MHealth [121], PAMAP2 [122], MNIST [123], FashionMNIST [124], CIFAR-10 [125], FEMNIST [126], CelebA [127], REALWORLD [128], HHAR [129] and Synthetic [130].

Table 4.1: Summary of the datasets and base ML models used for the evaluation of the proposed FL models in the papers collected in this thesis.

Task	Model	Dataset	Classes	Papers
HAR	Logistic Regression	MHealth [121]	12	I, II
		PAMAP2 [122]	17	I, II
		REALWORLD [128]	8	V
		HHAR [129]	6	V
Image Classification	CNN	MNIST [123]	10	IV
		FashionMNIST [124]	10	IV
		CIFAR-10 [125]	10	IV
		FEMNIST [126]	62	IV, VI
		CelebA [127]	-	IV, VI
Cluster Identification	Logistic Regression	Synthetic [130]	-	VI

Table 4.1 provides details on the datasets used in our studies, including the model applied and the available classes. In **Papers I** and **II**, we have used two HAR datasets containing physical activity monitoring data. The MHealth and PAMAP2 datasets are used to monitor physical activity. Both datasets contain motion sensor data for various physical activities. In **Paper IV**, we have evaluated our FL

model on five datasets, namely the MNIST, Fashion MNIST, CIFAR-10, FEMNIST, and CelebA datasets. The MNIST, FashionMNIST, and CIFAR-10 are commonly served as benchmark datasets for image classification. Furthermore, we use LEAF datasets [130] that are more realistic than the simulated datasets. **Paper V** introduces two realistic datasets available online, REALWORLD and HHAR from the HAR domain. Finally, in **Paper VI**, we have validated our solutions on three LEAF datasets. FEMNIST, CelebA, and Synthetic Dataset are used to show the robustness of our proposed FL models.

4.2 Baseline Algorithms

In order to show that our FL proposed models can bring a better training performance and save communication costs, various existing FL methods are used for comparison. Those are listed below.

- **FedAvg**: Federated averaging is the first published FL algorithm proposed by McMaham et al. [5]. The approach consists of simply averaging the local updates of the different models communicated by the client devices, as described in Section 2.2.

- **FedProx** [131]: FedProx addresses the challenges posed by heterogeneous networks by exploring the limitations of FedAvg algorithm in Non-IID settings. FedProx controls the deviation of local updates from the most recent global model. The devices that participate in the FL process utilize a proximal update technique to ensure that the client model does not deviate from the global model.

- **CMFL** [132]: CMFL improves the efficiency of communication in FL, guaranteeing the achievement of learning convergence. In the FL scenario, CMFL aims to decrease communication overhead by eliminating the need to transmit irrelevant client updates. This approach effectively reduces network usage and minimizes overhead.

- **Clustered Federated Learning (CFL)** [116]: CFL aims to mitigate the detrimental impact of Non-IID data in FL scenarios where the data distribution of individual clients varies. The CFL method divides client populations into clusters that have similar data distributions. This allows for training the same model on each cluster, which alleviates the effects of data heterogeneity on the overall performance of the FL approach.

- **Deletion Approach**: A technique based on deletion diagnostics [41] calculates the contributions of each client in FL. This ensures that each party's contribu-

tions are correctly appreciated, and motivates high-quality ones to join as early as possible.

- **FL-Cohort**: The proposed algorithm [133] calculates the contribution for each party in FL. Instead of removing a single client at a time, the FL-Cohort removes multiple similar clients from FL training at a time.

Table 4.2 presents an overview of the baseline methods used in our thesis, along with details of the datasets utilized.

Table 4.2: Summary of the Baselines used in this thesis.

Method	Datasets	Papers
FedAvg [5]	MHealth, PAMAP2, MNIST, FashionMNIST, CIFAR-10, FEMNIST, CelebA, REALWORLD, HHAR	I, II, IV, V
FedProx [131]	MNIST, FashionMNIST, CIFAR-10	IV
CMFL [132]	MNIST, FashionMNIST, CIFAR-10	IV
CFL [116]	REALWORLD, HHAR	V
Deletion Approach [41]	Synthetic, FEMNIST, CelebA	VI
FL-Cohort [133]	Synthetic, FEMNIST, CelebA	VI

In **Papers I, II, IV** and **V**, naive FedAvg method is used as a benchmark to compare with our proposed FL models and with other FL methods (e.g., FedAvg, Prox, CMFL, etc.) in terms of communication overhead, accuracy, and the measure F. FedProx is the baseline algorithm used in **Paper IV** to compare the communication cost and accuracy of different FL methods. In **Paper IV**, CMFL, a method aimed at mitigating communication overhead in FL, is served as a baseline to study and compare performance in terms of communication cost and accuracy. In **Paper V**, we evaluate the performance of two FL aggregation algorithms, i.e., FedAvg and CFL against the proposed FL algorithm, namely (GP-FL). Finally, the deletion approach and the FL-Cohort, which are used to measure the client contribution in FL, are used to compare the performance achieved with our proposed FL models in **Paper VI**. It is important to note that all our proposed FL models are considered an optimized and efficient version of FedAvg.

4.3 Evaluation Measures

The fundamental aspect of any evaluation involves determining what performance means. However, establishing a clear definition of performance is not straightforward due to the numerous measures proposed to evaluate performance found in the literature [134–136]. When a new algorithm is introduced, it is typical to demonstrate its enhancement over other existing algorithms in a certain aspect. The fundamental question is whether algorithm A is better than algorithm B, or how probable is

that algorithm A yields better results in contrast to algorithm B. In the context of FL tasks, an improved algorithm is often understood as one that achieves more accurate results in a few iterations and/or reduces resource consumption compared to other cutting-edge FL approaches. Our evaluation focuses on the performance of our proposed FL algorithm against some other baseline methods in terms of communication cost, the model's accuracy, energy consumption, battery lifetime of devices, etc. Table 4.3 lists the evaluation measures that have been used primarily in the studies of this thesis.

Table 4.3: Evaluation measures used across studies.

Evaluation measures	Papers
Communication Overhead	I, IV
F1	I, II, V
Accuracy	IV, VI
Energy Consumption	II
Battery Lifetime	II
Kendall's Tau Rank Correlation	VI

In FL, computational tasks are distributed among several resource-limited devices like smartphones, wearables, autonomous vehicles, and others. Given that communication in FL is more resource intensive than computation, minimizing communication is a highly desirable concern. Therefore, the performance in FL is characterized by the highest accuracy achieved after a given number of communications. The FL communication overhead is described by the average calculation performed by all participants during the global training. In the first round of FL, the central server sends the initial global model (N bytes) to several participants. Let us denote by W_s the set of all involved participants (clients). At the end of the round, it downloads the model updates of size N bytes from each client. This is repeated for T global rounds. The total communication overhead is specified as:

$$(2 \times N \times \mid W_s \mid) \times T + N \times \mid W_s \mid, \tag{4.1}$$

where N represents the size of the trained model in bytes, $\mid W_s \mid$ indicates the number of participants, and T is the overall number of training rounds. It is assumed that the size of the model updates and the number of training iterations are constant [137]. **Papers I** and **IV** have proposed FL models that improve the performance of the FL scenario in terms of reducing communication overhead while achieving better F-score/ accuracy values. In these studies, communication overhead is calculated using Eq. 4.1 to compare the efficiency against other state-of-the-art FL algorithms (FedAvg, FedProx, and CMFL).

In order to assess the effectiveness of a ML model, it is important to have a robust performance measurement that can evaluate the model's accuracy and correct-

ness. There are several appropriate measurements available, the most straightforward being a confusion matrix that can be applied to this problem. A confusion matrix displays four different prediction outcomes: (i) True Positive, (ii) False Positive, (iii) False Negative, and (iv) True Negative. These outcomes can be used to define different evaluation measures such as the F1 and the accuracy. The F1 (see Eq. 4.4) is calculated based on the precision and recall scores, which are defined in Eq. 4.2 and Eq. 4.3, respectively.

$$Precision = \frac{TP}{TP + FP}. \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN}. \tag{4.3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{4.4}$$

F1 varies between 0 and 1, with higher performance represented by higher values. F1 defined is used in **Papers I, II,** and **V** as an evaluation measure to compare the performance of our proposed FL methods with existing FL methods, such as FedAvg and CFL methods. Similarly, the accuracy given by Eq. 4.5 is used to compare and evaluate the FL performance in **Papers IV** and **VI** against different FL methods such as FedAvg, FedProx, CMFL, and CFL. Accuracy can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.5}$$

Energy consumption, defined in Eq. 4.6, has been used in **Paper II** as part of the multi-criteria evaluation to calculate the score of client (sensor nodes in the WNs) settings. We have assumed that the power required to transmit forth and back the model parameters is P_i, the size of the model parameters is s bytes, the bandwidth of network for client i is b_i bytes per second, and the round trip network latency experienced in communication with the server is l_i seconds. Assuming that $P_i^s = P_i^r = P_i$, energy E_i^t consumed by the client i in a time interval t can be expressed as:

$$E_i^t = P_i \times \left(2 \times \frac{s}{b_i} + l_i\right). \tag{4.6}$$

Moreover, the client's battery lifetime computed by Eq. 4.7 is also considered in **Paper II** as a criterion for selecting a suitable client for transmitting model parameters. At one point in time T it is useful to represent the battery lifetime as follows:

$$L_i^a(T) = \frac{1}{B_i}\left(B_i - \sum_{\tau=0}^{T} E_i^t(\tau)\right), \tag{4.7}$$

where the battery capacity of a client i is B_i Joule (J). Assuming $\sum_{\tau=0}^{T} E_i^t(\tau)$ is always less than or equal to B_i.

Kendall's Tau Rank Correlation is a commonly employed correlation technique for measuring the consistency between ranked characteristics. If two rankings x and y are generated for the specified set of clients, Kendall's tau $\mathcal{T}$ can be computed as follows:

$$\mathcal{T} = \frac{A - D}{\sqrt{(A + D + T_x)(A + D + T_y)}}, \tag{4.8}$$

where, A is the number of pairs in agreement, D is the number of pairs in disagreement, T_x is the number of pairs tied w.r.t. x, and T_y is the number of pairs tied w.r.t. y. We assess and compare four methods (CA-FL, GP-FL, Deletion approach, and FL-Cohort) by calculating Kendall's tau correlations among the clients' ranking scores produced by those methods. In **Paper VI**, Kendall's tau rank correlation given by Eq. 11 has been applied as a measure to compare client rankings produced by two different clients' contribution evaluation approaches.

4.4 Research Methodology

Two main research methodologies have been employed to obtain scientific results that address the research questions described. Firstly, in **Paper III**, the research methodology used is a *literature review*, an approach to gather information to improve the understanding of the topic being studied [138, 139]. Our study reviews current scientific results related to context-aware AI models, particularly in the context of SNs. Additionally, in **Papers I, II, IV, V** and **VI**, we use a research methodology based on *implementation and experimentation* [140]. In each of the studies presented in this thesis, new FL models are studied and evaluated through experimentation. A range of experiments are designed and carried out to verify the effectiveness of the algorithms using diverse datasets and benchmarking their performance to baseline methods. This research methodology involves performing experiments to explore particular research questions. Furthermore, this method assesses algorithms in a controlled experimental setting to measure a specific variable.

In **Paper I**, the proposed algorithm, namely the *Cluster Analysis-based FL (CA-FL)* model, has been compared with a state-of-the-art algorithm (FedAvg [5]) on HAR datasets (MHealth [121] and PAMAP2 [122]), with respect to the reduction of communication overhead between the central server and participants under IID and Non-IID data settings. The selection of representatives is based on the evaluation of the performance of the local model of each participant. Various experiments are designed and conducted to demonstrate the algorithm's ability to minimize communication overhead in system performance.

Similarly, in **Paper II**, the proposed *Energy-aware Multi-Criteria Federated Learning (EaMC-FL)* model is compared with FedAvg on the same HAR datasets used in **Paper I**, to select only one representative of a cluster for communication with the server with IID and Non-IID data distributions. In contrast to **Paper I**, the selection of the representatives is based on a multi-criteria evaluation for each sensor node (e.g., the local model performance, consumed energy, and battery lifetime). Several experiments are carried out to show that the EaMC-FL model can decrease the energy used by the edge nodes by reducing the amount of transmitted data in various use cases.

Paper IV is an extension study of **Paper I** in which the proposed algorithm, entitled *Federated Learning via Clustering Optimization (FedCO)* is extensively evaluated on publicly available datasets (MNIST [123], FashionMNIST [124] and CIFAR 10 [125]) and also on LEAF datasets (FEMNIST [126] and CelebA [127]) under IID and Non-IID data scenarios. The proposed algorithm is also benchmarked to various state-of-the-art FL methods, namely FedAvg [5], FedProx [131], and CMFL [132]. The results of several experiments demonstrated that the proposed *FedCO* technique outperforms the state-of-the-art FL approaches (i.e., FedAvg, FedProx and CMFL), in minimizing communication overhead and attaining higher accuracy in both IID and Non-IID scenarios.

In **Paper V**, a *group-personalized FL (GP-FL)* has been proposed and evaluated on two real-world HAR data (REALWORLD [128] and HHAR [129]). The performance of GP-FL has been compared to that of FedAvg and CFL. The experiments show that our method outperforms two baseline FL algorithms in terms of both model performance and convergence speed.

Paper VI introduces straightforward and efficient FL approaches that illustrate the evaluation of client behavior during the training phase. This is demonstrated using two established FL models published in **Papers I** and **V**, respectively. Our proposed FL models have been compared to the Deletion Approach [41] and FL-Cohort [133] on three LEAF datasets. These LEAF datasets used to validate the approaches are Synthetic [130], FEMNIST [126] and CelebA [127].

4.5 Validity Threats

In this section, we present different types of validity threats that may have arisen for the results of the thesis in four dimensions, including internal, external, construct and conclusion, along with the strategies implemented to address them.

4.5.1 Internal Validity

Internal validity refers to the impact of the experimental setup on the results [141, 142]. In this thesis, the threat of selection bias is presented, which can often be reme-

died by random sampling [143]. We split the experimental dataset into different sets. Specifically, in **Papers I** and **II**, we have used 10 different test sets (cross-validation) generated from each dataset used in the conducted experiments to avoid selection bias. 3-fold cross-validation on each training set is performed in **Paper IV**. In addition, in **Paper V**, we have performed 3-fold cross-validation on each experimental dataset. Finally, 3- and 5-fold cross-validations are performed on each experimental dataset for several communication rounds in **Paper VI**. Selection bias is not seen as a concern in **Paper III** as it is a literature review.

4.5.2 External Validity

External validity refers to the extent to which the results of the experiment can be applied or generalized [141, 142] in a different scenario. The experiments carried out in all the included studies are carefully designed to reduce such threats. Although many studies in thesis focus on different FL tasks, such as HAR, image classification, and/or cluster identification, they often use multiple datasets to evaluate the effectiveness of the proposed FL algorithm and prevent results that are specific to a particular scenario. Nevertheless, the limited number of datasets may not suffice to ensure generalizability/applicability to all real-world settings. All of our studies have used at least two types of datasets for evaluation, except for **Paper III**, which is a review of the literature.

4.5.3 Construct Validity

Construct validity deals with issues surrounding the extent to which the outcomes align with the intended conceptual goals [144]. If the algorithm fails to generate results comparable to the one created during the development stage, these threats could materialize. In order to mitigate these threats, the research group has discussed the proposed algorithms and setups prior to commencing the implementation phase. Throughout the implementation stage, regular tests are carried out to verify that the code functions correctly. This practice is essential to prevent the occurrence of runtime errors that are harder to detect than compile-time errors.

Papers I, II and **IV** employ the partitioning technique k-medoids to group participants into similar groups according to model parameters, necessitating the predefinition of the parameter k. The SI cluster validation measure has been used to determine the optimal number of clusters k. **Paper III** applies the DBSCAN algorithm to categorize keywords from the analyzed studies into clusters of keywords that are semantically related. Although DBSCAN does not necessitate prior knowledge of the number of clusters, it does require the specification of a parameter (eps). Through our experimentation with various eps values, we found that 0.3 resulted in the most well-balanced clustering without any outliers. **Paper V** applies the Markov

clustering technique to divide participants with similar empirical probability vectors into similar clusters. In order to assess the quality of clustering, we have conducted clustering with varying inflation values. Modularity is computed for each clustering iteration to determine the optimal inflation value for the given graph. In **Paper VI**, SI and modularity computation methods are used to determine the number of clusters for CA-FL and GP-FL, respectively. Similar to commonly used methods in **Papers I, II, IV** and **V**.

4.5.4 Conclusion Validity

Conclusion validity pertains to the efficiency of the study in the handling of the data, the experimental procedures, the evaluation, and the results [144]. In order to mitigate validity threats and limitations in our thesis, we provide a detailed description of the procedures followed to obtain the research results. A detailed description of the design, implementation, setup, and evaluation is given to provide the necessary understanding. Furthermore, **Papers I-V** have been through a peer-review procedure and have been published (except **Paper VI**, which is presently under review) in different conferences and journals, which confirms the validity of the experimental approach, analysis, and conclusions utilized in the studies.

5 Results and Analysis

This chapter provides an overview of the results of the thesis. Those have been considered and analysis in four main research directions: resource-aware FL, personalized FL, evaluation of client behavior in FL, and context-aware edge-based AI for SNs. The latter are the areas where our studies contribute to by providing new FL models, clients' behavior evaluation techniques and identifying future challenges and gaps in edge-based AI field.

5.1 Resource-aware Federated Learning

The results presented in this thesis, which are relevant to the resource-aware FL, are published in **Papers I, II, IV, V** and **VI**. Mitigating communication overheads through FL directly ties into resources-aware, hindering the scalability and efficiency of FL systems (see Section 1.1). **Papers I** and **IV** propose novel FL models to reduce communication overhead for the FL process. Namely, CA-FL introduced in **Paper I**, reduces communication overhead of FL through clustering analysis, and its optimized version FedCO, published in **Paper IV**, provides a communication-efficient FL via applying clustering optimization. In **Paper I**, the proposed CA-FL model has applied clustering analysis to reduce FL communication overhead by only sending the most representative updates to the central server. Particularly, we have developed a regression model ML and assessed this model against FedAvg under different experimental scenarios. CA-FL model has been evaluated on two HAR datasets in terms of accuracy and communication costs. The results indicate that CA-FL can significantly reduce communication costs compared to the conventional FL-baseline while maintaining the accuracy of the learning process. On the other hand, in **Paper IV**, we have improved the initial CA-FL framework by incorporating a dynamic clustering method that decreases communication overhead and speeds up the convergence of the global model. Our proposed FedCO evaluates the local updates of the clusters' representatives during each communication round and consequently moves certain clients to different clusters. The result of this cluster-updating process is the possibility of the appearance of new clusters or the disappearance of some of the existing ones. Our approach is capable of detecting and managing such situations. Furthermore, it implements a splitting procedure that conducts additional fine calibration of

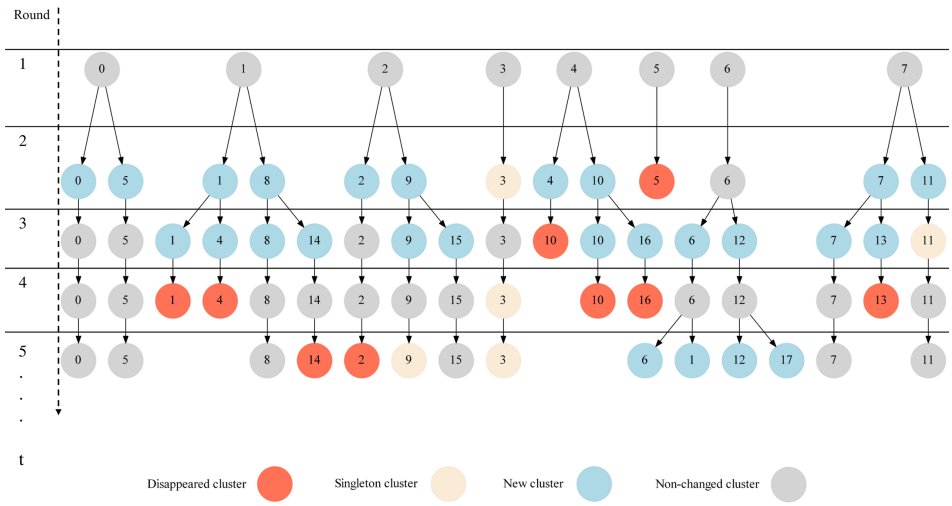


Figure 5.1: The clustering updates in the first five global communication rounds of the proposed FedCO algorithm applied on the Non-IID FashionMNIST dataset. Notice that the number in the circle represents the cluster label. The figure is copied from **Paper IV**.

the current clustering solution with respect to recently uploaded updates. The improvements have led to the development of a new version of a DL-based framework called FedCO. Figure 5.1 demonstrates the properties of the clustering optimization approach during the initial five global communication rounds of the FedCO algorithm. The cluster optimization operations mentioned above and introduced in **Paper IV** continue in a similar fashion for the next communication rounds. clients' partitions are dynamically adapted in each round of communication to reflect the new local updates from the clusters' representatives.

The proposed FedCO method is evaluated and compared with three other state-of-the-art FL algorithms such as FedAvg, FedProx and CMFL on five publicly available and widely exploited datasets, including the MNIST, CIFAR-10, Fashion-MNIST and LEAF datasets. The experimental results have shown that the proposed FedCO algorithm significantly reduces the number of communication rounds without sacrificing accuracy. Furthermore, experimental evaluations have shown that our FedCO algorithm outperforms the other three FL algorithms under IID and Non-IID, respectively. We have also shown that the FedCO algorithm can dynamically adapt the clients' partitioning in each communication round by relocating representative workers and performing the cluster splitting needed for the improvement of the clustering.

In FL, the client's computing resources are directly involved in the local training. For this reason, **Paper II** has examined the issue of energy consumption and the energy budget of FL edge nodes in SNs. In this regard, a proposed algorithm, namely EaMC-FL, takes into account the trade-off between the performance of the ML model trained and the energy usage of the respective sensor node. The performance of the

EaMC-FL algorithm is compared to the conventional FedAvg algorithm. The EaMC-FL algorithm has been evaluated under six use cases on the same HAR datasets used in **Paper I**. The experimental results indicate that EaMC-FL surpasses FedAvg in terms of total energy consumption, energy budget, and model precision.

Traditional approaches to assessing client contributions require an independent evaluation of each client's contribution outside the initial FL procedure. This leads to increased use of computational resources and longer processing times, as stated above. Furthermore, these methods do not take into account the dynamic nature of the data during the training process. **Paper VI** has introduced an indirect way of measuring clients' contribution in FL through evaluation of clients' behavior through the whole training process using our already published FL models. Those are CA-FL and GP-FL models presented in **Papers I** and **V**, respectively. During the training process, CA-FL can assess how often each client can serve as a cluster's representative. The score calculated could be considered as an indirect indicator of the client's reliability, i.e., the evaluation of the client's data quality. Typically, the CA-FL selects the highest-performing (most reliable) clients in each training round to participate in the global training of the joint model in a FL setting. Moreover, GP-FL, introduced in **Paper V** and proposing a group personalized FL model, counts the frequency of cluster changes made by the client during the FL training procedure. Practically, if the client changes the cluster to which it is assigned frequently, this indicates varying data quality for that client, as its position within the clustering structure is not stable. This could also be interpreted as an indirect evaluation of the quality of the client data.

The results of the experiments have demonstrated that our approaches (CA-FL and GP-FL) can provide with realistic evaluate of the clients' contributions to the overall FL model without significant communication and computation costs.

5.2 Personalized Federated Learning

Although customized FL is promising, it does not benefit from the potential for collaborative learning between participants, which presents a great problem due to two aspects. On one hand, personalized FL cannot learn effectively with a small amount of data on each client device. On the other hand, it fails to leverage device similarity with respect to tasks or data.

Paper V introduces a group personalized FL approach using cluster eccentricity analysis evaluated in the context of HAR applications. Figure 5.2 shows the various ways to model FL. The conventional FL scenario, presented in the central plot, assumes a federation of decentralized clients, each of which has its own private data. These clients participate in FL global training with the aim of enhancing the performance of the model. Consequently, the global model is created by aggregating local models with various features derived from various data sources. In the context of

private FL, each client represents a distinct ML task characterized by a unique data distribution. A dedicated model will be trained privately for each task to effectively address the specific characteristics of the data. As a result, a distinct personal model is generated for each client. The situation is depicted in the plot on the left side of Figure 5.2. Despite the fact that different clients have different tasks, it is reasonable to assume that there is a degree of similarity between the various tasks. The objective of the GP-FL model proposed in the mentioned study is to address the challenge of data heterogeneity. Also, it is worth pointing out that the proposed model is considered a general solution that is evaluated on the HAR data.

Clearly, GP-FL can train multiple global models simultaneously, each corresponding to a group of clients with homogeneous class probability estimations. During every training iteration, the empirical probability vector of each client is updated to reflect the information in its new batch. The eccentricity analysis introduced in [76] is used in GP-FL to maintain a dynamic grouping of clients. In particular, eccentricity is applied to determine whether a client is a member of a given cluster. In addition, the Chebyshev inequality is used as a criterion to determine whether a client continues to be a member of an existing cluster [78]. As a result, in the subsequent rounds, some clients may change groups, or entirely new singleton groups could appear.

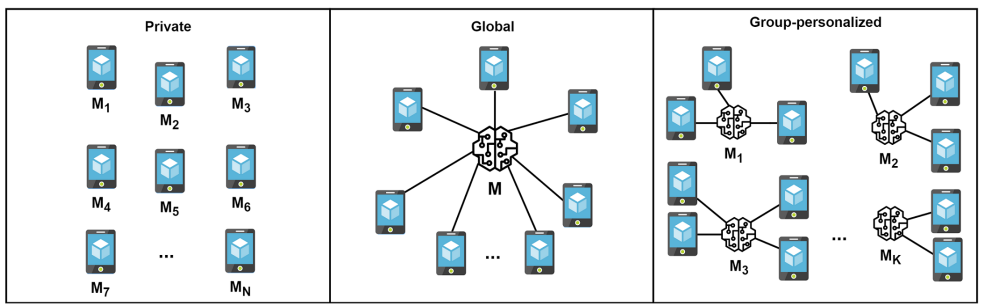


Figure 5.2: Comparison of three distinct FL scenarios: (i) The plot on the left illustrates a setting where every model is trained using the private data of the client; (ii) The middle plot depicts a situation in which a global model is generated using the models trained by different clients; (iii) The plot on the right demonstrates a setting that considers the similarity among participants and create a global model based on the local models of each group of similar devices. The figure is copied from **Paper V**.

The HAR problem is well suited for studying and evaluating our proposed group personalized FL technique, since different activities often exhibit common patterns while also being highly unique [145, 146]. The GP-FL algorithm has been evaluated through a set of experiments conducted in the HAR domain. The experimental results show that GP-FL surpasses two baseline FL algorithms (FedAvg and CFL) in terms of both the performance of the model and the speed of convergence.

5.3 Evaluation of Client Behavior

Although the main emphasis of **Papers I** and **V** is differed from clients' contribution evaluation in FL, it has been noticed that their proposed FL models are capable of assessing the clients' behavior through the training process. The latter can be indirectly used to assess and predict clients' contribution to the shared model.

Paper VI has proposed to assess and predict client contribution through clients' behavior evaluation by demonstrating the idea using our FL models CA-FL and GP-FL. These FL algorithms exhibit that the evaluation of the clients' behavior can be used to measure the contribution of the clients to the built shared model. The primary idea involves training a global model with the participation of a specific group of clients while also assessing the clients' behavior (e.g., reliable versus unreliable) simultaneously during the training phase.

The CA-FL algorithm, outlined in **Paper I**, computes the frequency of each client to be selected as a cluster representative throughout the training phase. During each training iteration of the CA-FL, one client with the best performing model is selected for each cluster to be its representative. Consequently, each client assigns a score that reflects its frequency of being selected as a cluster's representative, and can be interpreted as its reliability evaluation. In contrast to the CA-FL approach, the GP-FL technique (introduced in **Paper V**) provides an opportunity to detect clients exhibiting unstable behavior while undergoing training. The GP-FL algorithm initially divides the clients into several clusters based on the similarity between their class distributions. This clustering is continuously updated throughout the training process by assessing at each training round the clients' assignment among the clusters, and potentially reassigning some to new clusters. Specifically, we calculate how many times each client changes a cluster it belongs to during FL training. This can be seen as an indicator of the instability of the client's behavior (data), i.e., it can be interpreted as reflecting the client's lack of unreliability. Hence, these FL algorithms have shown the ability to practically evaluate the client's contribution without significant computational costs and time. The experimental results obtained demonstrate that our proposed approaches can efficiently evaluate client contributions in a robust manner.

Our proposed FL clients' behavior evaluation techniques have been evaluated against the Deletion approach and FL-Cohort technique using three LEAF datasets. In addition, the performance of our two methods, CA-FL and GP-FL, have been compared with two baselines, the Deletion approach, and FL-Cohort, in different experimental settings in terms of their ability to evaluate the clients' contributions to the global model. In our experiments, the Non-IID setting was used. It is important to note that various criteria can be applied to categorize clients into three (or more) groups (such as highly reliable clients, clients with a good degree of reliability, and unreliable clients) based on the scores generated by CA-FL or GP-FL algorithms.

Figure 5.3 illustrates the heatmaps displaying Kendall's tau correlation ranking

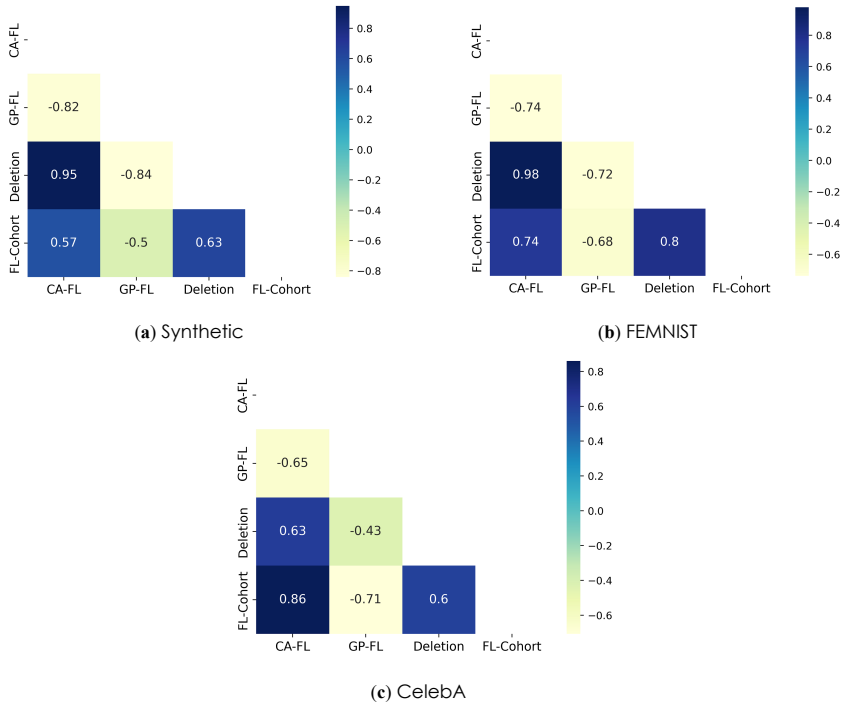


Figure 5.3: Heatmaps of Kendall's tau ranking scores of CA-FL, GP-FL, Deletion approach, and FL-Cohort on Synthetic, FEMNIST and CelebA datasets. The figure is copied from **Paper VI**.

scores for the rankings generated by the four studied methods: CA-FL, GP-FL, Deletion approach, and FL-Cohort. Those are evaluated across three different datasets. The stronger the correlation between the two methods, the darker the color, and conversely. These correlations help us to determine the similarity between the rankings produced by our two FL algorithms and the two baseline methods.

In case of the Synthetic dataset (see Figure 5.3(a)) it is evident that there is a highly positive correlation of 0.95 between CA-FL and the Deletion approach. This indicates that our method (CA-FL), which demands significantly fewer computational resources in comparison with the Deletion approach, can assess the clients' influence in a comparable manner to the latter method. In contrast, let us remind that GP-FL generates a score that indicates the level at which client behavior is considered unreliable. This leads to negative correlation scores (-0.84 and -0.5) between the GP-FL and the two baseline methods (Deletion approach and FL-Cohort, respectively). The latter approaches assess how the global model performance will be impacted if data of a client or a group of clients is not used during training.

In the FEMNIST dataset, as depicted in Figure 5.3(b), as one can see the CA-FL exhibits stronger correlation with the Deletion approach compared to its correlation with the FL-Cohort, similar to the Synthetic dataset. On the other hand, GP-FL correlated with lower scores (-0.72 and -0.68, respectively) with the Deletion approach

and FL-Cohort, respectively. In case of the CelebA dataset, CA-FL exhibits a higher correlation score with the FL-Cohort compared to the Deletion method, as shown in Figure 5.3(c). Similar to Synthetic and FEMNIST, GP-FL also exhibits a lower correlation score in the CelebA dataset with the Deletion method and FL-Cohort, as shown in Figure 5.3(c) with (-0.43 and -0.71, respectively).

5.4 Edge-based Artificial Intelligence for Sensor Networks

Paper III has conducted an in-depth analysis of the literature on context-aware edge-based AI models that leverage sensor technology, uncovering their applications, related challenges, and motivations for adopting AI solutions, along with identifying existing research gaps. Another interesting aspect of this research is the use of a semantic-based method to identify subjects relevant to the survey topic. In particular, the method is based on the examination of the keywords collected from all the studied articles. Initially, all unique keywords from the extracted articles are gathered. The total count of unique keywords is 637. Subsequently, this count is reduced to focus on the keywords that appear most frequently. Each keyword is assigned a score based on how often it appears among the keywords gathered from the articles. Then, all keywords with scores lower than the defined threshold value are excluded, resulting in only the top 82 most frequent keywords being retained. This method relies on assessing the semantic similarity among keywords to identify the main research or application topics covered by the articles studied in the survey.

Figure 5.4 shows the flowchart of the procedure to identify the primary survey subjects. The latter identifies eleven primary research topics supported by the articles included in the study. Furthermore, the relative percentage of cluster size generated by using DBSCAN with eps value of 0.3 on the 82 most frequent keywords. The parameter eps determines how close keywords should be to each other to be classified as part of a cluster, i.e., defines the neighborhood. Different values of eps have been experimented with, and 0.3 has produced the most evenly distributed clusters with no outliers. The selected articles are analyzed regarding identified subjects to get a deeper understanding of the limitations and gaps. These aspects are examined from various perspectives to address five main research questions. Potential future research directions are also deliberated.

AI, ML and DL, edge computing and smart monitoring, smart healthcare, and smart and wearable devices are the most popular topics that have been recognized. In the analysis carried out, we have also discovered that healthcare, smart cities, autonomous driving, environmental monitoring, and transportation are the top five domains of application. Enhancing recognition quality, optimizing management effectiveness, improving quality of service (QoS) and efficiency, and guaranteeing

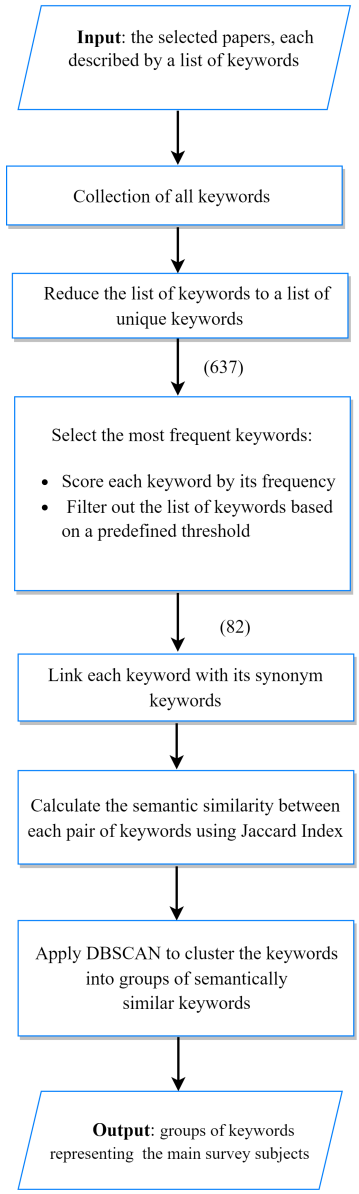


Figure 5.4: Flowchart describing the different steps of the semantic-aware approach applied to identify the main subjects covered by the included papers. The figure is copied from **Paper III**.

higher security are the primary motivations for implementing intelligent applications in context-aware AI-based systems.

The included papers have explored a range of AI-based solutions. Unsupervised and semi-supervised algorithms, along with transfer learning techniques, are highlighted as areas that have not received significant attention from researchers in many context-aware scenarios. Additionally, a promising collaborative framework,

such as FL has not been thoroughly investigated. The reviewed studies also lack research on location-based services, indicating a need for more studies that focus more on these issues.

5.5 Summary

This section summarizes how the research results achieved answer the research questions we formulated.

RQ 1: *How can we develop FL models that reduce resource consumption without sacrificing the model performance?*
The reduction of resource consumption we have shown to be tackled by decreasing the number of data transferred to the server while maintaining the model performance. In **Paper I**, a new FL model, entitled CA-FL, based on clustering analysis is introduced, aiming to minimize communication overhead by interacting only with the clusters' representatives. The latter leads to improvement in the speed of learning convergence. Specifically, we first cluster clients into groups based on the similarity between their local model parameters based on the ED similarity. Then, we select one representative from each cluster that realizes higher performance in terms of F1-score to communicate with the server.

Paper II has proposed a multi-criteria client evaluation FL model in WNs, called EaMC-FL. In particular, we first split the clients into groups in a similar way used in **Paper I**, but then we select representatives of groups according to the multi-criteria evaluation of clients. We define a client selection metric integrating several criteria, such as the client's resources and model performance. In the same way, as described in **Paper I**, only the selected representative from each cluster interacts with the server to train a unified model. At each subsequent iteration, the SI is also used to update the grouping of clients by evaluating whether representatives are still closely tied to their respective clusters.

In **Paper IV**, the CA-FL method is further improved by introducing a clustering optimization technique to improve model aggregation and reduce communication expenses. This has been achieved by optimizing the clustering of the representatives. The proposed FedCO method has applied regular updating of the clusters by iteratively assessing their quality and applying splitting procedure when is needed to enhance the clients' partitioning. The FL algorithms presented in **Papers I, II** and **IV** use only group representatives during their FL process, which reduces communication overhead and results in resource-efficient procedures. **Paper VI** has used the FL models introduced in **Papers I** and **V** to propose resource-efficient techniques for evaluation of the clients' contribution. The proposed techniques are capable of evaluating the clients' contributions similarly to two baseline approaches, but use a decreased expenses in terms of computational resources and time. Namely, in this

study (**Paper VI**) we have introduced a resource-efficient way to measure clients' contribution by evaluating clients' behavior during FL the training.

RQ 2: *How the clients' behaviour can be efficiently evaluated during the FL process?*
We have addressed this research question in **Paper VI** by proposing a way to measure contribution in FL by evaluating clients' behavior. The proposed FL clients' behavior evaluation techniques are based on the FL models presented in **Papers I** and **V**. Even though **Papers I** and **V** have different main focus, it has been observed that these algorithms (CA-FL and GP-FL) could be applied to assess the clients' contribution. In the CA-FL, the evaluation involves determining the frequency of each client to be selected as a cluster representative to participate in the construction of the shared model. This process can be considered as an indicator of the reliability of the client's behavior, i.e., of its data. In the GP-FL, we count how many times each client changes its cluster during FL training, interpreted as an indicator of the client's instability behavior and eventually suggesting unreliability. More specifically, if the client frequently changes the cluster to which it belongs, it indicates the varying quality of the data, because it does not have a stable position in the clustering structure. This behavior can also be seen as an indirect evaluation of the quality of the client's data.

RQ 3: *How can we personalize FL models to achieve robust model performance?*
A group-personalized FL (GP-FL) model is proposed for investigating this research question in **Paper V**. The proposed GP-FL aims to address the challenge of data heterogeneity and achieves a trade-off between the global model and local models performance. To study and evaluate the performance of GP-FL, we have computed and compared three different evaluation scenarios. A single *global performance* is assessed by computing the accuracy or F1 score generated by the overall model on the individual device's data. Then, the *personal performance* is evaluated by calculating the accuracy or F1 score achieved by the client's local model with its private data. Finally, *group performance* is evaluated by the accuracy or F1 score attained by each group's global model. Thus, three different client models are evaluated: the model trained locally by the client, the global model that averages the parameters of all clients' local models, and the group-personalized model, which relies only on the local models of clients having similar class distributions, i.e., activity patterns in HAR case. This study proposed GP-FL algorithm that constructs multiple global ML models, each iteratively trained on a dynamic group of clients with homogeneous class probability estimations.

RQ 4: *What AI-based solutions are underrepresented in the recent state-of-the-art of context-aware edge intelligence systems?*
In this thesis, we have presented a literature review of the recent development of

context-aware edge-based AI methods in SNs. In **Paper III**, an extensive review of the literature is carried out to explore the applications, associated challenges, and motivations to implement context-aware AI solutions in SNs. Furthermore, our main motivation in this study has been to identify current research gaps. Many AI-based solutions have been studied in the included research papers. Interestingly, we found that a collaborative AI framework such as FL has not been adequately examined in the reviewed research studies. Another aspect of this research involves employing a semantic-based method to identify subjects relevant to the survey main topic. In particular, the method is based on semantic-aware analysis of the keywords of the studied articles.

6 Conclusion and Future Directions

6.1 Conclusion

This thesis has introduced new resource-aware and personalized FL models that employ cluster analysis to enhance the effectiveness and robustness of FL in the context of diverse and dynamic data. In particular, the research results reported in this thesis have been identified to contribute to four research areas: resource-aware FL, personalized FL, evaluation of client behavior in FL, and exploration of edge-based AI for SNs. The main contributions of this thesis are outlined below:

1. We have proposed FL algorithms to bring communication efficiency into the FL environment. An FL approach (CA-FL, **Paper I**) is proposed to reduce communication overhead through clustering analysis. The study is also extended to control and optimize FL communication in the wireless network (EaMC-FL, **Paper II**). Clustering optimization is further introduced for client selection to minimize communication overhead (FedCo, **Paper IV**). Furthermore, indirect ways of measuring the contribution of clients in FL are proposed (**Paper VI**) through the evaluation of clients' behavior during the training process. The obtained results have shown the ability of the proposed FL algorithms to minimize the consumption of computational resources of FL clients while maintaining the performance of the model.

2. We have developed a group-personalized FL approach (GP-FL, **Paper V**) using cluster eccentricity analysis to achieve robust model performance and effectively manage heterogeneity in the data. The proposed FL model achieves a trade-off between the global and the local models performance. The experimental results have demonstrated that our group-personalized FL approach outperforms two conventional FL algorithms in terms of both model performance and convergence speed.

3. We have proposed two techniques (**Paper VI**) that are capable of effectively analyzing the clients' behavior during the FL process to support the evaluation of the clients' contribution to the shared model. The proposed techniques

require fewer resources compared to traditional FL clients' contribution evaluation techniques and are typically performed as a component of FL training procedures.

4. We have conducted a thorough literature review to identify AI edge-based solutions focusing on context-aware systems, particularly within SNs (**Paper III**). In addition, underrepresented AI-based solutions in recent context-aware edge intelligence systems are determined.

6.2 Future Directions

Our future interests are in applications within domains such as industrial IoT, underwater IoT, etc., to further validate the proposed in this thesis FL solutions and introduce novel resource-aware FL models. The focus will be to exploit horizontal FL, vertical FL, and transfer FL in such environments. In particular, we will study different resource-aware distributed (federated) AI methods to develop accurate and reliable asset fault detection and diagnosis solutions. Several potential directions for future research are described in the following:

- We are interested in studying energy-efficient vertical FL models applicable to different types of inaccurate data collected from various IoT contexts, e.g., under water or in harsh industrial environments. Settings that contain labeled and unlabeled data to detect and diagnose faults will be explored.

- Another interesting direction for future work is to investigate energy-efficient horizontal FL models based on multi-source data to identify and detect potential faults of critical assets.

- Transferring knowledge and pre-trained FL models from IoT devices to new devices with different data domains is also an interesting problem that could be studied further.

Bibliography

[1] K. M. H. et al. “Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective”. In: *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)* (2018), pp. 620–629.

[2] X. Q. et al. “A first look into the carbon footprint of federated learning”. In: *ArXiv* abs/2010.06537 (2020).

[3] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. “Model-based approximate querying in sensor networks”. In: *The VLDB Journal* 14 (2005), pp. 417–443.

[4] F. Bonomi, R. A. Milito, J. Zhu, and S. Addepalli. “Fog computing and its role in the internet of things”. In: *MCC ’12*. 2012.

[5] H. B. M. et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *International Conference on Artificial Intelligence and Statistics*. 2016.

[6] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik. “Federated Optimization: Distributed Machine Learning for On-Device Intelligence”. In: *ArXiv* abs/1610.02527 (2016).

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong. “Federated Machine Learning: Concept and Applications”. In: *arXiv: Artificial Intelligence* (2019).

[8] B. S. Guendouzi, S. Ouchani, H. E. Assaad, and M. E. Zaher. “A systematic review of federated learning: Challenges, aggregation methods, and development tools”. In: *J. Netw. Comput. Appl.* 220 (2023), p. 103714.

[9] M. hany mahmoud, A. Albaseer, M. M. Abdallah, and N. Al-Dhahir. “Federated Learning Resource Optimization and Client Selection for Total Energy Minimization Under Outage, Latency, and Bandwidth Constraints With Partial or No CSI”. In: *IEEE Open Journal of the Communications Society* 4 (2023), pp. 936–953.

[10] P. e. a. Kairouz. “Advances and Open Problems in Federated Learning”. In: *Found. Trends Mach. Learn.* 14 (2019), pp. 1–210. URL: https://api.semanticscholar.org/CorpusID:209202606.

[11] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao. "Communication Efficiency in Federated Learning: Achievements and Challenges". In: *ArXiv* abs/2107.10996 (2021).

[12] S. Huang, W. Shi, Z. Xu, I. W.-H. Tsang, and J. Lv. "Efficient federated multi-view learning". In: *Pattern Recognit.* 131 (2022), p. 108817.

[13] M. Xu, J. Liu, Y. Liu, F. X. Lin, Y. Liu, and X. Liu. "A First Look at Deep Learning Apps on Smartphones". In: *The World Wide Web Conference* (2018).

[14] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014).

[15] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie. "Communication-efficient federated learning via knowledge distillation". In: *Nature Communications* 13 (2021).

[16] J. Mills, J. Hu, and G. Min. "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT". In: *IEEE Internet of Things Journal* 7 (2020), pp. 5986–5994.

[17] H. M. et al. "Federated Learning of Deep Networks using Model Averaging". In: *arXiv preprint arXiv:1602.05629* (2016).

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. "Practical Secure Aggregation for Privacy-Preserving Machine Learning". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017).

[19] Q. Li, Y. Diao, Q. Chen, and B. He. "Federated Learning on Non-IID Data Silos: An Experimental Study". In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (2021), pp. 965–978.

[20] M. F. Criado, F. E. Casado, R. Iglesias, C. V. Regueiro, and S. Barro. "Non-IID data and Continual Learning processes in Federated Learning: A long road ahead". In: *Inf. Fusion* 88 (2021), pp. 263–280.

[21] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning". In: *International Conference on Machine Learning*. 2019.

[22] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. "On the Convergence of FedAvg on Non-IID Data". In: *ArXiv* abs/1907.02189 (2019).

[23] Y. Zhao et al. "Federated Learning with Non-IID Data". In: *ArXiv* 1806.00582 (2018).

[24] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. "Federated Optimization in Heterogeneous Networks". In: *arXiv: Learning* (2018).

[25] V. Kulkarni, M. Kulkarni, and A. Pant. “Survey of Personalization Techniques for Federated Learning”. In: *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)* (2020), pp. 794–797.

[26] C. T. Dinh, N. H. Tran, and T. D. Nguyen. “Personalized Federated Learning with Moreau Envelopes”. In: *ArXiv* abs/2006.08848 (2020).

[27] A. Fallah, A. Mokhtari, and A. E. Ozdaglar. “Personalized Federated Learning: A Meta-Learning Approach”. In: *ArXiv* abs/2002.07948 (2020).

[28] W. Bao, C. Wu, S. Guleng, J. Zhang, K.-l. A. Yau, and Y. Ji. “Edge computing-based joint client selection and networking scheme for federated learning in vehicular IoT”. In: *China Communications* 18 (2021), pp. 39–52.

[29] M. Hu, D. Wu, Y. Zhou, X. Chen, and M. Chen. “Incentive-Aware Autonomous Client Participation in Federated Learning”. In: *IEEE Transactions on Parallel and Distributed Systems* PP (2022), pp. 1–1.

[30] Q. Wu, K. He, and X. Chen. “Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework”. In: *IEEE Open Journal of the Computer Society* 1 (2020), pp. 35–44.

[31] H. Ren, J. Deng, and X. Xie. “Privacy Preserving Text Recognition with Gradient-Boosting for Federated Learning”. In: *ArXiv* abs/2007.07296 (2020).

[32] Z. Iqbal and H. Y. Chan. “Concepts, Key Challenges and Open Problems of Federated Learning”. In: *International Journal of Engineering* (2021).

[33] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. “Federated Learning: Challenges, Methods, and Future Directions”. In: *IEEE Signal Processing Magazine* 37 (2019), pp. 50–60.

[34] R. Gosselin, L. Vieu, F. Loukil, and A. Benoit. “Privacy and Security in Federated Learning: A Survey”. In: *Applied Sciences* (2022).

[35] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov. “Exploiting Unintended Feature Leakage in Collaborative Learning”. In: *2019 IEEE Symposium on Security and Privacy (SP)* (2018), pp. 691–706.

[36] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu. “HFEL: Joint Edge Association and Resource Allocation for Cost-Efficient Hierarchical Federated Edge Learning”. In: *IEEE Transactions on Wireless Communications* 19 (2020), pp. 6535–6548.

[37] X. Zhang, Z. Chang, T. Hu, W. Chen, X. Zhang, and G. Min. “Vehicle Selection and Resource Allocation for Federated Learning-Assisted Vehicular Network”. In: *IEEE Transactions on Mobile Computing* (2023).

[38] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton. “Cell-Free Massive MIMO for Wireless Federated Learning”. In: *IEEE Transactions on Wireless Communications* 19 (2019), pp. 6377–6392.

[39] S. K. Shyn, D. Kim, and K. Kim. “FedCCEA : A Practical Approach of Client Contribution Evaluation for Federated Learning”. In: *ArXiv* abs/2106.02310 (2021).

[40] T. Tuor, S. Wang, B. Ko, C. Liu, and K. K. Leung. “Data Selection for Federated Learning with Relevant and Irrelevant Data at Clients”. In: *ArXiv* abs/2001.08300 (2020).

[41] G. Wang, C. X. Dang, and Z. Zhou. “Measure Contribution of Participants in Federated Learning”. In: *2019 IEEE International Conference on Big Data (Big Data)* (2019), pp. 2597–2604.

[42] D. Jatain, V. Singh, and N. Dahiya. “A contemplative perspective on federated machine learning: Taxonomy, threats & vulnerability assessment and challenges”. In: *J. King Saud Univ. Comput. Inf. Sci.* 34 (2021), pp. 6681–6698.

[43] L. Yang, Z. Meng, and L. Wang. “A multi-layer two-dimensional convolutional neural network for sentiment analysis”. In: *Int. J. Bio Inspired Comput.* 19 (2022), pp. 97–107.

[44] G. Drainakis, K. V. Katsaros, P. Pantazopoulos, V. Sourlas, and A. J. Amditis. “Federated vs. Centralized Machine Learning under Privacy-elastic Users: A Comparative Analysis”. In: *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)* (2020), pp. 1–8.

[45] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan. “MLbase: A Distributed Machine-learning System”. In: *Conference on Innovative Data Systems Research*. 2013.

[46] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. “Federated Learning: Strategies for Improving Communication Efficiency”. In: *ArXiv* abs/1610.05492 (2016).

[47] A. Li, L. Zhang, J. Wang, F. Han, and X. Li. “Privacy-Preserving Efficient Federated-Learning Model Debugging”. In: *IEEE Transactions on Parallel and Distributed Systems* 33 (2022), pp. 2291–2303.

[48] M. N. Fekri, K. Grolinger, and S. Mir. “Distributed load forecasting using smart meter data: Federated learning with Recurrent Neural Networks”. In: *International Journal of Electrical Power & Energy Systems* (2021).

[49] J. X. et al. “Ternary Compression for Communication-Efficient Federated Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33 (2020), pp. 1162–1176.

[50] J. Liu and Y. Jin. “Multi-objective Search of Robust Neural Architectures against Multiple Types of Adversarial Attacks”. In: *Neurocomputing* 453 (2021), pp. 73–84.

[51] N. Zeng, D. Song, H. Li, Y. You, Y. Liu, and F. E. Alsaadi. “A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution”. In: *Neurocomputing* 432 (2021), pp. 170–182.

[52] H. Zhu, J. Xu, S. Liu, and Y. Jin. “Federated Learning on Non-IID Data: A Survey”. In: *ArXiv* abs/2106.06843 (2021).

[53] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang. “Client Selection for Federated Learning With Non-IID Data in Mobile Edge Computing”. In: *IEEE Access* 9 (2021), pp. 24462–24474.

[54] T.-C. Chiu, Y.-Y. Shih, A.-C. Pang, C.-S. Wang, W. Weng, and C.-T. Chou. “Semisupervised Distributed Learning With Non-IID Data for AIoT Service Platform”. In: *IEEE Internet of Things Journal* 7 (2020), pp. 9266–9277.

[55] E. E. A. et al. “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects”. In: *Eng. Appl. Artif. Intell.* 110 (2022), p. 104743.

[56] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. “An Efficient k-Means Clustering Algorithm: Analysis and Implementation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (2002), pp. 881–892.

[57] W. Xiao and J. Hu. “A Survey of Parallel Clustering Algorithms Based on Spark”. In: *Sci. Program.* 2020 (2020), 8884926:1–8884926:12.

[58] A. K. Jain, M. N. Murty, and P. J. Flynn. “Data clustering: a review”. In: *ACM Comput. Surv.* 31 (1999), pp. 264–323.

[59] P. Arora and S. Varshney. “Analysis of K-Means and K-Medoids Algorithm For Big Data”. In: *Procedia Computer Science* 78 (2016), pp. 507–512.

[60] C. Briggs, Z. Fan, and P. András. “Federated learning with hierarchical clustering of local updates to improve training on non-IID data”. In: *2020 International Joint Conference on Neural Networks (IJCNN)* (2020), pp. 1–9.

[61] Y. Kim, E. A. Hakim, J. Haraldson, H. Eriksson, J. M. B. da Silva, and C. Fischione. “Dynamic Clustering in Federated Learning”. In: *ICC 2021 - IEEE International Conference on Communications* (2020), pp. 1–6.

[62] Y. Xiao, H.-B. Li, and Y.-p. Zhang. “DBGSA: A Novel Data Adaptive Bregman Clustering Algorithm”. In: *ArXiv* abs/2307.14375 (2023).

[63] J. B. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *In Lucien M. Le Cam and Jerzy Neyman, editors, Proceedings of the Berkley symposium on mathematical statistics and probability* 1 (1967), pp. 281–297.

[64] S. van Dongen. “Graph clustering by flow simulation”. In: 2000.

[65] H.-S. Park and C.-H. Jun. “A simple and fast algorithm for K-medoids clustering”. In: *Expert Syst. Appl.* 36 (2009), pp. 3336–3341.

[66] A. Gordon. “Measures of similarity and dissimilarity”. In: 1999.

[67] D. B. Bisandu, R. Prasad, and M. M. Liman. “Data clustering using efficient similarity measures”. In: *Journal of Statistics and Management Systems* 22 (2019), pp. 901–922.

[68] S. kiran Vangipuram and R. Appusamy. “A SURVEY ON SIMILARITY MEASURES AND MACHINE LEARNING ALGORITHMS FOR CLASSIFICATION AND PREDICTION”. In: *International Conference on Data Science, E-learning and Information Systems 2021* (2021).

[69] X. Wang, A. A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. J. Keogh. “Experimental comparison of representation methods and distance measures for time series data”. In: *Data Mining and Knowledge Discovery* 26 (2010), pp. 275–309.

[70] P. Jaccard. “Étude comparative de la distribution florale dans une portion des Alpes et du Jura”. In: *Bulletin del la Société Vaudoise des Sciences Naturelles* (1901).

[71] S. Kolouri et al. “Optimal Mass Transport: Signal processing and machine-learning applications”. In: *IEEE Signal Processing Magazine* 34 (2017), pp. 43–59.

[72] P. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[73] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. “An extensive comparative study of cluster validity indices”. In: *Pattern Recognit.* 46 (2013), pp. 243–256.

[74] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. R. Dougherty. “Model-based evaluation of clustering validation measures”. In: *Pattern Recognit.* 40 (2007), pp. 807–824.

[75] C. G. Bezerra, B. S. J. Costa, L. A. Guedes, and P. P. Angelov. “A new evolving clustering algorithm for online data streams”. In: *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)* (2016), pp. 162–168.

[76] P. Angelov. “Anomaly detection based on eccentricity analysis”. In: *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*. 2014, pp. 1–8. DOI: 10.1109/EALS.2014.7009497.

[77] C. G. Bezerra et al. “An evolving approach to data streams clustering based on typicality and eccentricity data analytics”. In: *Information Sciences* 518 (2020), pp. 13–28. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.12.022. URL: https://www.sciencedirect.com/science/article/pii/S0020025519311363.

[78] J. G. Saw et al. “Chebyshev Inequality With Estimated Mean and Variance”. In: *The American Statistician* 38 (1984), pp. 130–132.

[79] I. Škrjanc et al. “Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey”. In: *Inf. Sci.* 490 (2019), pp. 344–368.

[80] W. Chen, S. Horváth, and P. Richtárik. “Optimal Client Sampling for Federated Learning”. In: *Trans. Mach. Learn. Res.* 2022 (2020).

[81] M. Mitzenmacher. “The Power of Two Choices in Randomized Load Balancing”. In: *IEEE Trans. Parallel Distributed Syst.* 12 (2001), pp. 1094–1104.

[82] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor. “Fast-Convergent Federated Learning”. In: *IEEE Journal on Selected Areas in Communications* 39 (2020), pp. 201–218.

[83] Z. C. et al. “Dynamic Attention-based Communication-Efficient Federated Learning”. In: *ArXiv* abs/2108.05765 (2021).

[84] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. A. Bilmes. “Diverse Client Selection for Federated Learning via Submodular Maximization”. In: *International Conference on Learning Representations*. 2022.

[85] S. Abdulrahman, H. Tout, A. Mourad, and C. Talhi. “FedMCCS: Multicriteria Client Selection Model for Optimal IoT Federated Learning”. In: *IEEE Internet of Things Journal* 8 (2021), pp. 4723–4735.

[86] L. Wang, W. Wang, and B. Li. “CMFL: Mitigating Communication Overhead for Federated Learning”. In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[87] T. H. T. Le, N. H. Tran, Y. K. Tun, M. N. H. Nguyen, S. R. Pandey, Z. Han, and C. S. Hong. “An Incentive Mechanism for Federated Learning in Wireless Cellular Networks: An Auction Approach”. In: *IEEE Transactions on Wireless Communications* 20 (2020), pp. 4874–4887.

[88] J. Zhang, Y. Wu, and R. Pan. “Incentive Mechanism for Horizontal Federated Learning Based on Reputation and Reverse Auction”. In: *Proceedings of the Web Conference 2021* (2021).

[89] F. S. et al. “Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31 (2019), pp. 3400–3413.

[90] K. Ozkara, N. Singh, D. Data, and S. N. Diggavi. “QuPeD: Quantized Personalization via Distillation with Applications to Federated Learning”. In: *Neural Information Processing Systems*. 2021.

[91] D. A. et al. “QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding”. In: *Neural Information Processing Systems*. 2016.

[92] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li. “TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning”. In: *NIPS*. 2017.

[93] Y. H. et al. “CosSGD: Nonlinear Quantization for Communication-efficient Federated Learning”. In: *ArXiv* abs/2012.08241 (2020).

[94] Y. Ren, Y. Cao, C. Ye, and X. Cheng. “Two-layer accumulated quantized compression for communication-efficient federated learning: TLAQC”. In: *Scientific Reports* 13 (2023).

[95] A. F. Aji and K. Heafield. “Sparse Communication for Distributed Gradient Descent”. In: *arXiv preprint arXiv:1704.05021* (2017).

[96] N. Dryden, T. Moon, S. A. Jacobs, and B. C. V. Essen. “Communication Quantization for Data-Parallel Training of Deep Neural Networks”. In: *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)* (2016), pp. 1–8.

[97] T. Chen, G. B. Giannakis, T. Sun, and W. Yin. “LAG: Lazily Aggregated Gradient for Communication-Efficient Distributed Learning”. In: *Neural Information Processing Systems*. 2018.

[98] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni. “Federated Learning with Matched Averaging”. In: *ArXiv* abs/2002.06440 (2020).

[99] J. Liu, J. H. Wang, C. Rong, Y. Xu, T. Yu, and J. Wang. “FedPA: An adaptively partial model aggregation strategy in Federated Learning”. In: *Comput. Networks* 199 (2021), p. 108468.

[100] X. Wu, Z. Liang, and J. Wang. “FedMed: A Federated Learning Framework for Language Modeling”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[101] M. Asad, A. Moustafa, and M. Aslam. “CEEP-FL: A comprehensive approach for communication efficiency and enhanced privacy in federated learning”. In: *Appl. Soft Comput.* 104 (2021), p. 107235.

[102] A. Z. Tan et al. “Towards Personalized Federated Learning”. In: *IEEE transactions on neural networks and learning systems* PP (2021).

[103] Y. Mei, B. Guo, D. Xiao, and W. Wu. “FedVF: Personalized Federated Learning Based on Layer-wise Parameter Updates with Variable Frequency”. In: *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)* (2021), pp. 1–9.

[104] X. Ma, J. Zhang, S. Guo, and W. Xu. “Layer-wised Model Aggregation for Personalized Federated Learning”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 10082–10091.

[105] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. “Federated Learning with Personalization Layers”. In: *ArXiv* abs/1912.00818 (2019).

[106] X. Ni, X. Shen, and H. Zhao. “Federated optimization via knowledge codistillation”. In: *Expert Syst. Appl.* 191 (2021), p. 116310.

[107] E. Jeong and M. Kountouris. “Personalized Decentralized Federated Learning with Knowledge Distillation”. In: *ICC 2023 - IEEE International Conference on Communications* (2023), pp. 1982–1987.

[108] J. Jang, H. Ha, D. Jung, and S. Yoon. “FedClassAvg: Local Representation Learning for Personalized Federated Learning on Heterogeneous Neural Networks”. In: *Proceedings of the 51st International Conference on Parallel Processing* (2022).

[109] C. Li, G. Li, and P. K. Varshney. “Decentralized Federated Learning via Mutual Knowledge Transfer”. In: *IEEE Internet of Things Journal* 9 (2020), pp. 1136–1147.

[110] T. Wan, W. Cheng, D. Luo, W. Yu, J. Ni, L. Tong, H. Chen, and X. Zhang. “Personalized Federated Learning via Heterogeneous Modular Networks”. In: *2022 IEEE International Conference on Data Mining (ICDM)* (2022), pp. 1197–1202.

[111] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. “Federated Multi-Task Learning”. In: *Neural Information Processing Systems*. 2017.

[112] L. Corinzia and J. M. Buhmann. “Variational Federated Multi-Task Learning”. In: *ArXiv* abs/1906.06268 (2019).

[113] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang. “Personalized Cross-Silo Federated Learning on Non-IID Data”. In: *AAAI Conference on Artificial Intelligence*. 2020.

[114] F. Hanzely and P. Richtárik. “Federated Learning of a Mixture of Global and Local Models”. In: *ArXiv* abs/2002.05516 (2020).

[115] Y. Deng, M. M. Kamani, and M. Mahdavi. “Adaptive Personalized Federated Learning”. In: *ArXiv* abs/2003.13461 (2020).

[116] F. Sattler, K.-R. Müller, and W. Samek. “Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2019), pp. 3710–3722.

[117] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran. “An Efficient Framework for Clustered Federated Learning”. In: *IEEE Transactions on Information Theory* 68 (2020), pp. 8076–8091.

[118] M. Duan, D. Liu, X. Ji, R. Liu, L. Liang, X. Chen, and Y. Tan. “FedGroup: Efficient Federated Learning via Decomposed Similarity-Based Clustering”. In: *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)* (2020), pp. 228–237.

[119] Z. Gao, Y. Yang, C. Zhao, and Z. Mo. “CFedPer: Clustered Federated Learning with Two-Stages Optimization for Personalization”. In: *2022 18th International Conference on Mobility, Sensing and Networking (MSN)* (2022), pp. 171–177.

[120] Y. J. Cho, J. Wang, T. Chirvolu, and G. Joshi. “Communication-Efficient and Model-Heterogeneous Personalized Federated Learning via Clustered Knowledge Transfer”. In: *IEEE Journal of Selected Topics in Signal Processing* 17 (2023), pp. 234–247.

[121] O. Baños, R. García, J. A. H. Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga. “mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications”. In: *IWAAL*. 2014.

[122] A. Reiss and D. Stricker. “Introducing a New Benchmarked Dataset for Activity Monitoring”. In: *2012 16th International Symposium on Wearable Computers* (2012), pp. 108–109.

[123] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[124] H. Xiao, K. Rasul, and R. Vollgraf. “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: *ArXiv* abs/1708.07747 (2017).

[125] A. Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. In: 2009.

[126] G. Cohen, S. Afshar, J. C. Tapson, and A. van Schaik. “EMNIST: Extending MNIST to handwritten letters”. In: *2017 International Joint Conference on Neural Networks (IJCNN)* (2017), pp. 2921–2926.

[127] Z. Liu, P. Luo, X. Wang, and X. Tang. “Deep Learning Face Attributes in the Wild”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3730–3738. DOI: 10.1109/ICCV.2015.425.

[128] T. Sztyler and H. Stuckenschmidt. “On-body localization of wearable devices: An investigation of position-aware activity recognition”. In: *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (2016), pp. 1–9.

[129] A. Stisen et al. “Smart Devices are Different: Assessing and MitigatingMobile Sensing Heterogeneities for Activity Recognition”. In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems* (2015).

[130] S. e. a. Caldas. “LEAF: A Benchmark for Federated Settings”. In: (2018).

[131] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. “Federated Optimization in Heterogeneous Networks”. In: *arXiv: Learning* (2018).

[132] L. Wang, W. Wang, and B. Li. “CMFL: Mitigating Communication Overhead for Federated Learning”. In: *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[133] C. Düsing and P. Cimiano. “Towards predicting client benefit and contribution in federated learning from data imbalance”. In: *Proceedings of the 3rd International Workshop on Distributed Machine Learning* (2022).

[134] D. J. Hand. “Assessing the Performance of Classification Methods”. In: *International Statistical Review* 80 (2012).

[135] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand. “A Performance Evaluation of Federated Learning Algorithms”. In: *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning* (2018).

[136] S. Divi, Y.-S. Lin, H. Farrukh, and Z. B. Celik. “New Metrics to Evaluate the Performance and Fairness of Personalized Federated Learning”. In: *ArXiv* abs/2107.13173 (2021).

[137] E. Babu, D. Rueckert, and A. Davison. “Federated Deep Learning for Healthcare Data”. In: *Department of Computing, Imperial College London* (June. 13, 2020).

[138] H. Snyder. “Literature review as a research methodology: An overview and guidelines”. In: *Journal of Business Research* (2019).

[139] J. Paul and A. R. Criado. “The art of writing literature review: What do we know and what do we need to know?” In: *International Business Review* 29 (2020), p. 101717.

[140] M. Berndtsson, J. Hansson, B. Olsson, and B. Lundell. “Developing your Objectives and Choosing Methods”. In: 2002.

[141] R. Feldt and A. Magazinius. “Validity Threats in Empirical Software Engineering Research - An Initial Survey”. In: *International Conference on Software Engineering and Knowledge Engineering*. 2010.

[142] V. M. Erthal, B. P. de Souza, P. Santos, and G. H. Travassos. “Characterization of continuous experimentation in software engineering: Expressions, models, and strategies”. In: *Sci. Comput. Program.* 229 (2023), p. 102961.

[143] E. C. Weyant. “Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, 5th Edition”. In: *Journal of Electronic Resources in Medical Libraries* 19 (2022), pp. 54–55.

[144] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, and B. Regnell. “Experimentation in Software Engineering”. In: *Springer Berlin Heidelberg*. 2012.

[145] E. Sannara et al. “Evaluation and comparison of federated learning algorithms for Human Activity Recognition on smartphones”. In: *Pervasive Mob. Comput.* 87 (2022), p. 101714.

[146] E. Sannara et al. “Evaluation of federated learning aggregation algorithms: application to human activity recognition”. In: *In ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiCompISWC ’20)* (2020).

Paper I

Reducing Communication Overhead of Federated Learning through Clustering Analysis

Ahmed A. Al-Saedi, Veselka Boeva, Emiliano Casalicchio
In: 2021 IEEE Symposium on Computers and Communications (ISCC), 2021, pp. 1-7, Athens, Greece.

Abstract

Training of machine learning models in a Datacenter, with data originated from edge nodes, incurs high communication overheads and violates a user's privacy. These challenges may be tackled by employing Federated Learning (FL) machine learning technique to train a model across multiple decentralized edge devices (workers) using local data. In this paper, we explore an approach that identifies the most representative updates made by workers and those are only uploaded to the central server for reducing network communication costs. Based on this idea, we propose a FL model that can mitigate communication overheads via clustering analysis of the worker local updates. The Cluster Analysis-based Federated Learning (CA-FL) model is studied and evaluated in human activity recognition (HAR) datasets. Our evaluation results show the robustness of CA-FL in comparison with traditional FL in terms of accuracy and communication costs on both IID and non-IID cases.

1 Introduction

Today, low-end edge devices, such as smart-phones and Internet of Things (IoT) devices, are equipped with powerful and energy-efficient processing units. This unique landscape enables the execution of complex Machine Learning (ML) tasks at the edge of the Internet. Federated Learning (FL) [1] has been proposed by Google in 2017 to enable learning collaboratively over a large number of edge devices (workers hereafter) without the need to centrally collect training data and to train centralized models.

In FL, a global model is initialized in a central location (e.g., a cloud datacenter) and is shared with all participating workers for a set of iterative training round. At each training round, the workers train the model locally (using their local data), and then each worker sends its model parameter update back to the central node. The global model is then updated averaging the local model parameters received by all the workers and shared again with them [1]. These operations are repeated at each iteration round.

The iterative nature of FL, however, does not eliminate the network congestion problem completely due to data offloading from the edge to the cloud. Indeed, for complex models, large scale applications (with hundred thousand or millions of workers), and high frequency update, the communication overhead at each iteration is not negligible, and becomes a challenge to be addressed [2–4]. Several research studies have proposed decreasing the overall number of bits transferred for each worker update which is mostly used by means of data compression. However, none of those works considers the impact of some loss of data compression results that could bring harm to the learning accuracy, resulting in no convergence guarantees [4]. The other direction is to reduce the total number of worker updates transferred during the model training [5]. In this paper, we propose a new federated learning algorithm, entitled Cluster Analysis-based Federated Learning (CA-FL), that is reducing the communication overhead without losing accuracy rate by minimizing the number of worker updates transferred during each FL iteration phase. CA-FL consists of an initialization phase and a sequence of iteration steps (an iterative training phase). At the initialization phase the local updates of the available workers are analysed and partitioned into a number of clusters. Then at each training round (iteration phase), representatives (one or more) are selected for each cluster based on some task specific evaluation criteria. Only the data supplied by the representative workers are used for training during the current round in order to build the global model. As a final operation, at each round the worker partitioning is adapted based on the analysis of the newly arrived information (the local updates of the selected workers). In that way new representatives will be selected at the next training round using the adapted partitioning of the workers. The training process is iterative and continues until a predefined stopping criteria is met. The rationale behind the above idea is that it is not necessary to calculate and upload the local updates that are similar. Instead, we can use representatives for each group of similar updates (model local parameters) and reflect the importance of each group into the aggregation of the global model by weighting its cardinality.

The proposed CA-FL algorithm can be considered as a communication-efficient version of Federated Averaging (FedAvg) algorithm introduced in [2]. Hence in this study we compare CA-FL to FedAvg on two publicly available physical activity monitoring datasets in terms of model accuracy and communication costs. These datasets have been selected since human activity recognition is one of smart monitoring applications that will benefit of federated learning models that reduce communication

costs.

The rest of this paper is organized as follows. Section 2 contains a brief description of the related works. This is followed by Section 3 which introduces required background and methods. The proposed approach is explained in Section 4. Details about the data, experimental setting and discussion of the results obtained from initial evaluation of the proposed approach are presented in Section 6. Section 7 is devoted to conclusions and future work.

2 Related Work

Asad et al. [6] have evaluated the communication efficiency of FL approaches and additionally provided a detailed review of the state-of-the-art FL frameworks. Solutions that address the problem of reducing network overhead in FL can be classified in two broad categories: works that reduce the communication traffic by means of data compression [3, 7–11] and studies that aim at reducing the number of local updates and syncs with the central model [5, 12].

Notice that our proposed FL model falls into the second category. For instance, Gaia approach, introduced in [12], proposes a communication architecture for minimizing the communication cost for geo-distributed ML systems. The Gaia measures the importance of a local update with a predefined threshold. Thus, the insignificant local update will be excluded from uploading to the server. A framework called Communication-Mitigated Federated Learning (CMFL) has been proposed in [5] to specify the relevance of a worker update by calculating the proportion of parameters having different signs of the local update and global update then excluding the irrelevant updates. Chen et al. have proposed a communication-dynamic method for FL where a layer-wise asynchronous update method is used [13]. Jeong et al. have introduced a FL scheme to reduce communication overhead with few losses on the accuracy with a feature fusion technique for aggregating the features [14]. Wu et al. have introduced in [15] a framework (FedMed) for the adaptive aggregation, mediation incentive scheme, and topK strategy to address the model aggregation and communication overhead. Elbir [16] proposes a hybrid FL and centralized learning (HFCL) mechanism that performs gradient computation on only active devices that have enough computational power during model training. Caldas et al. have presented strategies to alleviate communication costs by sending a lossy compressed model to the client and permitting users to train small subsets of the global model (Federated Dropout) [17]. In [18] a practical Robust, and Communication-efficient Semi-supervised FL (RC-SSFL) system that allows the clients to jointly learn a high-quality model is used. We have been inspired by the above discussed studies and explored an approach that applies clustering analysis to identify the most representative updates made by workers and those are only uploaded to the central server for reducing network communication costs.

3 Background and Methods

In this section, we introduce the concept of federated learning upon which our solution is based [4]. In addition, we discuss techniques used for conducting clustering analysis on the workers' local updates.

3.1 Federated Learning

In Federated Learning [4], the learning task is conducted by having a coordinator (also referred as server and usually running in a cloud Datacenter) that, throughout a set of iterations, collaborates with all the participating workers (also referred as clients) to learn and build a shared, privacy preserving ML model. Note that private data is kept by the workers and not shared with the Server.

In detail, a model M is learned iteratively by using a randomly selected subset, denoted by W_s, of all available workers. The workers in W_s participate at each round and compute the gradient of the loss over all the data held by them. Each worker $w \in W_s$ at round t has its own row of data D_w^t and a local model m_w^t. At each round t, each worker trains its local model by iterating the local update multiple times of Stochastic Gradient Descent (SGD) before sending the next local model m_w^{t+1} to the server which holds the global model. The server, after collecting all the local models computed at round t, performs a synchronous update of the global model M_{t+1}. The global model update can be computed using different criteria. In this paper, we assume it is evaluated by means of federated averaging, that is the local model m_w^t, $w \in W_s$ and global model M_t are updated by the following equations: [19]:

$$m_w^{t+1} = m_w^t - \eta g_w^t; \tag{1}$$

$$M_{t+1} = \sum_{w \in W_s} \frac{n_w}{n} m_w^{t+1}, \tag{2}$$

where m_w^{t+1} is the local update, g_w^t are the updated weights on its local data at the current model m_w^t, M_{t+1} is the next global model, η is a learning rate computed by each worker, W_s is the set of workers that participate in the training, n is the sum (total number) of all data points and n_w is the number of local data points. The server then distributes the global model M_{t+1} to the workers that can perform another iteration of local training and model update. We refer to this baseline algorithm as FedAvg (FL-baseline hereafter).

3.2 Partitioning Algorithms

Three partitioning algorithms are well known and generally used for data analysis to partition the data points into k disjoint clusters [20]: k-means, k-medians, and k-medoids clustering. All these methods start by initializing a set of k cluster centers,

where k is preliminarily determined. Then, each data point is assigned to the cluster whose center is the nearest, and the cluster centers are recomputed. This process is repeated until the points inside every cluster become as close to the center as possible and no further item reassignments take place. The three partitioning techniques vary in how the cluster center is located.

In our CA-FL algorithm we use k-medoids for partitioning the available workers into groups of similar workers w.r.t. their local updates. The medoid is the most centrally located point in a given cluster. Due to this in the CA-FL, the medoid of each cluster can be used as the representative of the cluster in a case the evaluation of the workers' updates is missing.

3.2.1 Silhouette Index

Silhouette Index (SI) is a well-known and widely used internal cluster validation technique [21]. The cluster validation techniques are designed to be used for evaluating the quality of data partitions. Internal cluster validation techniques base their evaluation on the same information used to derive the clusters themselves and can be split with respect to the specific clustering property they assess. A detailed and comparative overview of different types of validation measures can be found in [22]. Suppose a_i represents the average distance of object i from all the other objects in the cluster to which the object i is assigned, and b_i represents the minimum of the average distances of object i from objects of the other clusters. Then the Silhouette Index $s(i)$ of object i can be calculated as

$$s(i) = (b_i - a_i)/\max\{a_i, b_i\}. \tag{3}$$

$s(i)$ measures how well object i matches the clustering at hand. $s(i) \in [-1, 1]$ and higher value points out a better quality of the clustering results. For example, when $s(i)$ is close to 1 this means that object i is assigned to a very appropriate cluster. A situation is different when $s(i)$ is about zero. Namely, object i lies between two clusters. The worst case is when $s(i)$ is close to -1. Evidently, this object has been misclassified.

The proposed CA-FL algorithm applies the Silhouette Index at each iteration round for assessing whether the worker representatives are still well tied to their current clusters (see Section 4).

3.2.2 Estimation of the number of clusters

The partitioning algorithms, such as k-medoids used in our study, require k, the number of clusters to be known in advance. However, determining the optimal number of clusters k may be challenging in solving problems involving real-world data sets. One solution that may apply in such cases is to build a clustering model with a range of values for k and then evaluate the quality of the generated clustering partitions. For example, different internal cluster validation indices (such as SI) can be applied to recognize the optimal clustering solution.

In this study, we take the advantage of Silhouette Index method for identifying the optimal k. Namely, k-medoids is conducted for each value of k in a given interval. The corresponding scores of SI generated by the different k are depicted as a function of k. The optimal k is the value at which a significant local change in the value of this method observed.

4 Proposed CA-FL algorithm

The proposed CA-FL algorithm foresee two distinctive phases, *initialization* and *iteration*, described in what follows.

Let us define $W = \{w_1, w_2, \ldots, w_n\}$ the set of all available workers and W^t is a subset of W that contains workers selected at round t. The workers in W^t can be the clusters' representatives or a set of randomly selected workers and $\mid W^t \mid < n$.

Initialization Phase:

1. The Server initializes model M_t, $t = 0$.
2. Server sends the initial global model M_t to a set of workers W^t ($W^t \subset W$). These are selected to be used for a initial training of FL at round t.
3. Each worker $w \in W^t$ receives the global model M_t and optimizes its parameters locally, i.e. m_w^t initial update is produced and sent back to the server (eq. 1).
4. Server aggregates the parameters $\{m_w^t \mid w \in W^t\}$ uploaded by the selected workers W^t to update the global model M_t through the FedAVG algorithm (eq. 2).
5. The local updates $\{m_w^t \mid w \in W^t\}$ of the workers in W^t are analyzed by using k-medoids algorithm and the workers are partitioned into k clusters of similar updates, i.e. an initial clustering $C^t = \{C_1^t, C_2^t, \ldots, C_k^t\}$ of the workers in W^t is produced.

Iteration Phase:

1. At each iteration round t ($t \geq 0$) server evaluates each local update m_w^t, $w \in W^t$ by using an evaluation measure suitable for the task under consideration.
2. Server ranks the workers in each cluster $C_i^t, i = 1, 2, \ldots, k$ and selects the top-ranked worker (the representative) or the first p top-ranked workers, where p is defined to be proportional to the cluster size. The selected representatives form a new set of workers $W^{t+1} = \{w_1^{t+1}, w_2^{t+1}, \ldots, w_k^{t+1}\}$, where $k << \mid W^0 \mid$. Each worker $w \in W^{t+1}$ will check-in with the server.

3. Server sends the global model M_t to each representative $w \in W^{t+1}$.

4. Each representative $w \in W^{t+1}$ receives the global model M_t and optimizes its parameters locally, i.e. m_w^{t+1} update is produced (eq. 1) and sent back to the server.

5. Server aggregates the received parameters $\{m_w^{t+1} \mid w \in W^{t+1}\}$ uploaded by the representatives to update the global model through the FedAVG algorithm, i.e. an updated global model M_{t+1} is produced.

6. Server adapts C^t to the newly arrived local updates, i.e. SI is applied for assessing whether each representative $w \in W^{t+1}$ is still well tied to its current cluster (eq. 3). It may happen that some workers will change their clusters, i.e. the updated clustering C^{t+1} of the workers in W^0 is produced.

Then the steps 1 – 6 of the *iteration* phase are repeated until the maximum number T of training rounds is executed. The CA-FL algorithm iteration phase is illustrated in Figure 1.

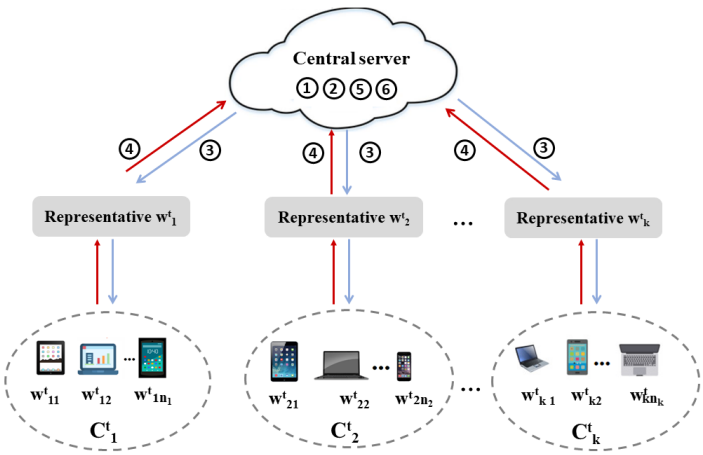


Figure 1: A schematic illustration of the CA-FL algorithm iteration phase. Circled numbers correspond to the iteration phase.

5 Evaluation

5.1 Data and Experimental Setup

For initial evaluation of the proposed CA-FL algorithm, we have used publicly available real-world datasets [23, 24] from UCI Machine Learning repository. We use mHealth and Pamap2 physical activity monitoring datasets. Both datasets involve motion sensor data of several physical activities. Each worker trains its local model

for a number of epochs on the local dataset using sklearn as the ML library. The updated local model is sent back to the Server. When all workers performed e number of epochs, the Server updates the global model and sends it again to the workers. The process continues and conducts until max number of training rounds T is reached. In order to simulate a distributed scenario each dataset is used to generate 10 experimental datasets by randomly separating the data points into a number of groups. Each group is supposed to represent data supplied by one worker. In addition, we have studied two different experimental data distribution scenarios: IID (Independent and Identically Distributed) and non-IID. In that way for each studied number of workers (20, 30, 40, 50 and 60) and each used dataset (mHealth and Pamap2) we have built 20 experimental datasets, i.e. 200 experimental datasets in total. We compare the CA-FL algorithm against the FL-baseline algorithm [2] on the built experimental datasets.

5.2 Evaluation Metrics

The FL communication overhead metrics are defined by the average calculation performed by all workers during the model training. In the first FL round of communication, the server sends the initial global weights (N bytes) to set of selected workers W_s. At the end of the round, it downloads the updates of size N bytes from each participating worker and carries out the aggregation. This is iterated for T communication rounds. The overall communication cost of the server is given by:

$$(2 \times N \times \mid W_s \mid) \times T + N \times \mid W_s \mid,$$

where N is the size of the trained model in bytes, $\mid W_s \mid$ is the number of selected workers and T is the total number of training rounds. We assume the size of the model updates and the number of training iterations to be fixed [25].

5.3 Implementation and Availability

We have compared the proposed CA-FL algorithm against the FL-baseline, briefly explained in Section 3.1, using Stochastic Gradient Descent (SGD) classifier as a training model in two data distribution scenarios, i.e. IID versus non-IID. The FL-baseline and CA-FL algorithms are implemented in Python using Scikit-learn library. The machine learning task is a multi-class classification problem of human activity recognition. Silhouette Index is used from the Python library Scikit-learn. The scikit-learn implementation of the F-measure (*micro-average* F_1) has been used to evaluate the accuracy of the two studied algorithms.

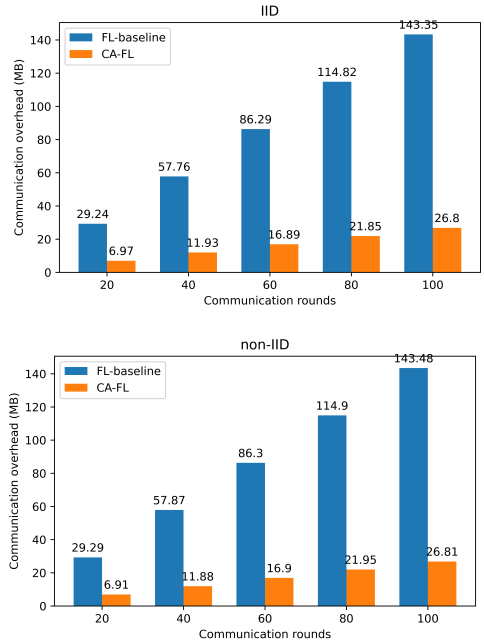


Figure 2: Communication overhead accumulated by FL-baseline and CA-FL algorithms on mHealth datasets under IID (top) and non-IID (bottom) scenarios for different communication rounds.

5.4 Results and Discussion

We initially evaluate the communication overhead by applying FL-baseline and CA-FL algorithms on the experimental datasets of mHealth and Pamap2 datasets built for 20 numbers of workers under two data distribution scenarios: IID versus non-IID. The results obtained on mHealth dataset are reported in Figure 2. The results generated on the experimental datasets of Pamap2 are similar. We have also studied how the number of communication rounds affects the communication cost and the experiments have been conducted for different numbers of rounds (20, 40, 60, 80 and 100). We can observe that in both data distribution scenarios, CA-FL substantially reduces the communication costs. As it will be shown and discussed later in this section this is achieved without losing the learning correctness. In addition, it is interesting to notice that the communication costs accumulated by FL-baseline increase faster with the higher number of rounds in comparison to the ones generated by CA-FL.

The classification performance of the two compared algorithms has been evaluated by running 10-fold cross-validation on each experimental dataset of mHealth and Pamap2 datasets for 20 communication rounds and different number of workers. Figure 3 shows the average F1 scores produced by FL-baseline and CA-FL algorithms on Pamap2 experimental datasets. We can see that the lost in the learning accuracy of CA-FL algorithm in comparison with the FL-baseline algorithm is insignificant.

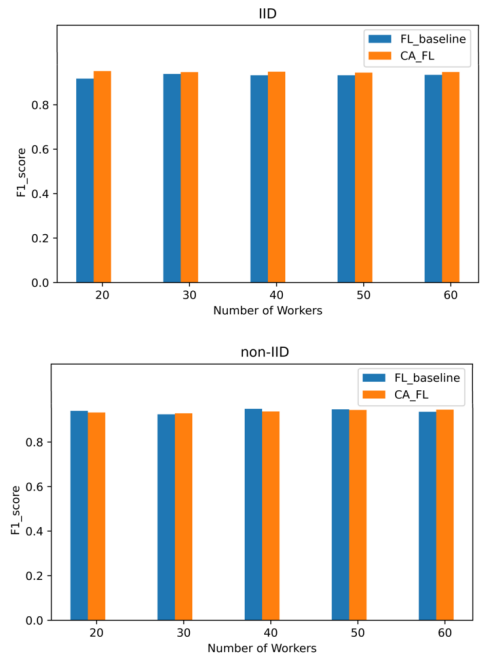


Figure 3: F1 scores generated by FL-baseline and CA-FL algorithms on Pamap2 datasets under IID (top) and non-IID (bottom) scenarios for different number of workers.

The results generated on the experimental datasets of mHealth are similar.

We have additionally studied how the number of workers used to train the global model affect the communication cost and classification performance of FL-baseline and CA-FL. Figures 3 and 4 display F1 scores and the generated communication costs of FL-baseline and CA-FL algorithms on Pamap2 experimental datasets in IID and non-IID data scenarios, respectively. The obtained results confirm the discussed above observations. Namely, CA-FL reduces the communication costs significantly and these are not highly dependent on the number of workers in contradiction to the FL-baseline algorithm behaviour in the same context. The classification performance of both algorithms is not affected by the number of workers. Notice that the similar results under both evaluation criteria are generated on the experimental datasets of mHealth. The propose CA-FL approach also supplies with an opportunity to conduct a post-analysis step by evaluating the quality of the workers' data. For example, the number of rounds at which a worker has been used as a representative can eventually be considered as a measure for its data quality.

Table 1 presents the workers' data quality scores (frequency of being representatives) obtained by averaging over the scores generated on the experimental datasets built on each of the two datasets (mHealth and Pamap2) in the two studied data distribution scenarios (IID versus non-IID) using 20 workers, i.e. four different scores are generated for each worker.

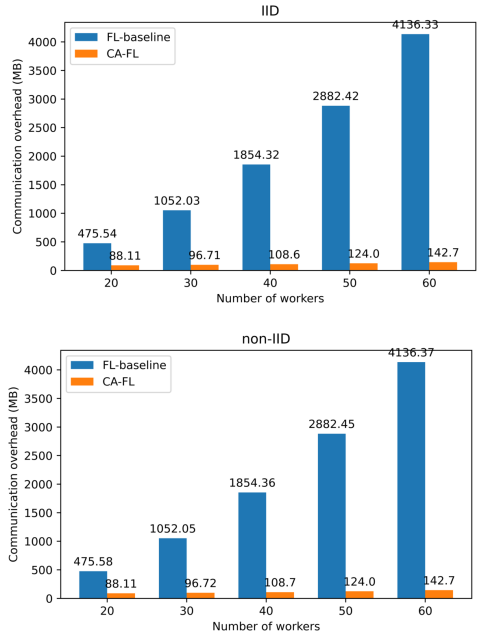


Figure 4: Communication overhead accumulated by FL-baseline and CA-FL algorithms on Pamap2 datasets under IID (top) and non-IID (bottom) scenarios for different number of workers.

Table 1: Evaluation of the workers' data quality on the datasets of mHealth and Pamap2 in the two data distribution settings (IID vs non-IID). The last column contains the average scores calculated over the studied scenarios.

	mHealth		**Pamap2**		
worker	IID	Non-IID	IID	Non-IID	Average Score
0	3.5	2.7	**4.9**	2	3.275
1	3.8	2.8	2.8	3.6	3.25
2	2.3	4	2.7	3.1	3.025
3	3.3	3.3	3.4	2.2	3.05
4	2.7	2.3	2.4	3.7	2.775
5	2.8	2.1	4.6	2.3	2.95
6	3.5	3.9	3.6	2.1	3.275
7	1.4	3.5	2.5	3.7	2.775
8	3	2.6	2.9	3.7	3.05
9	2.7	3.4	2.8	3.5	3.1
10	4.2	1.4	1.7	3.2	2.625
11	2.1	2.5	2.5	2.6	2.425
12	3.2	3.5	3.5	4.3	3.625
13	**4.5**	3.3	3.7	4	**3.875**
14	3.4	3.2	3.3	**5.1**	**3.75**
15	1.9	3.8	3.3	3.7	3.175
16	4.3	2.6	4.3	3.5	**3.675**
17	2.7	3.4	4.2	3.4	3.425
18	3.5	3.6	4.3	4.3	**3.925**
19	3.8	**4.1**	3.1	3.5	**3.625**

It is interesting to notice that the workers' performance is influenced by the used dataset and the data distribution scenario. For example, worker 0 has produced the

highest score on Pamap2 data set in IID scenario, but the lowest one in non-IID for the same dataset. In addition, one can see that worker 17 has a high score (4.2) for Pamap2 dataset under IID scenario, but a significantly lower one under the same scenario for mHealth dataset. However, five workers (13, 14, 16, 18 and 19) perform comparatively well under the four studied cases and can be considered more reliable then the others. This is supported by the last column presenting the average score over all four scores for each worker. Such an evaluation procedure may be applied as an initial step for the exploration of the workers in order to be able to select the top ranked (the most reliable) workers for the real training of the model. This is initially studied by the experiments discussed at the end of this section.

We have studied how the number of selected representatives varies with respect to the studied experimental scenarios. The identified optimal number of clusters (representatives) in different experimental scenarios of the two datasets are presented in Table 2. Interestingly, the non-IID scenario of mHealth dataset uses less number of representatives under all studied numbers of workers while the rest three experimental scenarios demonstrate very similar patterns. In addition, we investigate the

Table 2: Identified optimal number of clusters (representatives) in different experimental scenarios (IID and non-IID)

	mHealth		**Pamap2**	
No. workers	IID	Non-IID	IID	Non-IID
20	10	8	10	8
30	16	10	17	16
40	24	17	23	25
50	27	20	28	27
60	38	23	34	36

communication costs and F1-score values in a scenario when the number of workers selected by traditional federated learning is equal to the number of representatives (k) used by the CA-FA algorithm. The results obtained on the experimental non-IID datasets of Pamap2 data are given in Table 4. It can be observed that under different number of rounds, CA-FL outperforms FL-baseline in term of accuracy. However, since CA-FL communicates with all participating workers in the initialization phase rather than only k number of workers this costs 7.0974 MB for one round of training when 20 workers involved in federated learning scenario. The latter, as can be seen in Table 4, affects the overall communication cost of CA-FL. We can also notice that the server overhead increases linearly with the number of rounds. Furthermore, the results of the FL-baseline and the ones accumulated during the iteration phase of CA-FL are comparable since both use k number of workers. Note that the overall communication costs of the CA-FL are calculated by aggregating the communication costs of the initialization and iteration phases. The results generated on Pamap2 experimental IID dataset are similar. Note that the above discussed scenario is artificial, since in our definition of CA-FL algorithm the representative workers are identified

from a set of workers (W^0) which are randomly selected from all available workers (W) during the initialization phase (see Step 2). This means that CA-FL can always use less number of workers that the FL-baseline in training of FL, i.e. k is always smaller than $\mid W^0 \mid$ (see Step 2 in the Iteration Phase). Finally, we conduct experi-

Table 3: F1 scores and communication costs by FL-baseline (top) and CA-FL (bottom) on Pamap2 dataset under non-IID data

FL-baseline		
No. Rounds	F1 Score	Communication Overhead (MB)
20	0.943	16.606
40	0.944	32.839
60	0.940	49.149
80	0.943	65.439
100	0.953	81.640

CA-FL			
		Communication Overhead (MB)	
No. Rounds	F1 Score	Iteration Phase	Overall Costs
20	0.947	15.575	22.673
40	0.950	31.490	38.587
60	0.951	48.336	55.433
80	0.954	63.900	70.998
100	0.961	81.324	88.421

ments on both datasets for the two studied experimental scenarios (IID and non-IID) in which the FL-baseline algorithm (FedAvg) uses in the training process the five most frequently selected representatives pointed out by our CA-FL algorithm. These are given in bold in Table 1 (13, 14, 16, 18 and 19). Notice that the used five workers have produced higher F1 scores for the built ML model than those randomly selected eight workers used to generate the results given in Table 4. For example, one can compare the scores produced on Pamap2 non-IID experimental datasets (the last column in Table 5) with the F1 scores in Table 4. Therefore, we believe the quality of data of the five most frequently selected workers by CA-FL algorithm is better than of those that are randomly selected by FedAvg for the different number of rounds. These initial results are interesting and worth to be further studied in our future work.

Table 4: F1 scores of FL-baseline on mHealth and Pamap2 datasets, where the server uses only the most frequently selected representatives to train the global model in different experimental scenarios (IID and non-IID)

	mHealth		**Pamap2**	
No. Rounds	IID	Non-IID	IID	Non-IID
20	0.556	0.574	0.962	0.952
40	0.553	0.551	0.958	0.951
60	0.561	0.568	0.943	0.944
80	0.556	0.553	0.951	0.966
100	0.567	0.569	0.945	0.957

6 Conclusion and Future Work

In this paper, we have proposed a FL algorithm (entitled CA-FL), that can mitigate communication overheads via clustering analysis of the worker local updates. The key idea is to cluster the participating workers into groups based on the similarity of their local model parameters. The most representative workers from each cluster are then selected and used in each training round to optimize the global model. CA-FL model has been evaluated on two human activity recognition datasets in terms of accuracy and communication costs. The CA-FL algorithm has been benchmarked against FL-baseline algorithm under different experimental scenarios. We have studied how the classification performance and communication costs of the two algorithms are affected by the number of workers under two data distribution settings (IID and non-IID). The obtained results have shown that CA-FL can achieve substantial reduction of communication costs in comparison with the traditional FL-baseline without losing the learning correctness. Our future plans are to pursue further study and evaluate CA-FL algorithm potential for different ML tasks on richer real-world data sets in different application scenarios. We also plan to compare the proposed FL algorithm with some other state-of-the-art communication-efficient FL algorithms.

Acknowledgments

The first author is supported by an Iraq Ministry of Higher Education and Scientific Research PhD Scholarship.

References

[1] B. McMahan and D. Ramage. “Federated learning: Collaborative machine learning without centralized training data”. In: *Google Research Blog* (Apr. 2017). URL: https://ai.googleblog.com/2017/04/federated-learning-collaborative.htKonecnml.

[2] H. B. M. et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *International Conference on Artificial Intelligence and Statistics*. 2016.

[3] W.-T. Chang and R. Tandon. “Communication Efficient Federated Learning over Multiple Access Channels”. In: *arXiv preprint arxiv:2001.08737* (2020).

[4] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. “Federated Learning: Strategies for Improving Communication Efficiency”. In: *ArXiv* abs/1610.05492 (2016).

[5] L. Wang, W. Wang, and B. Li. “CMFL: Mitigating Communication Overhead for Federated Learning”. In: *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[6] M. Asad, A. Moustafa, T. Ito, and A. Muhammad. “Evaluating the Communication Efficiency in Federated Learning Algorithms”. In: *arXiv preprint arXiv:2004.02738* (2020).

[7] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. “1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs”. In: *INTERSPEECH*. 2014.

[8] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li. “TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning”. In: *NIPS*. 2017.

[9] S. Zhou, Z. Ni, X. Zhou, H. Wen, Y. Wu, and Y. Zou. “DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients”. In: *arXiv preprint arXiv:1606.06160* (2016).

[10] A. F. Aji and K. Heafield. “Sparse Communication for Distributed Gradient Descent”. In: *arXiv preprint arXiv:1704.05021* (2017).

[11] M. Asad, A. Moustafa, and T. Ito. “FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning”. In: *Applied Sciences* 10 (2020).

[12] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. Ganger, P. B. Gibbons, and O. Mutlu. “Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds”. In: *NSDI*. 2017.

[13] Y. Chen, X. Sun, and Y. Jin. “Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31 (2020), pp. 4229–4238.

[14] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim. “Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data”. In: *arXiv preprint arXiv:1811.11479* (2018).

[15] X. Wu, Z. Liang, and J. Wang. “FedMed: A Federated Learning Framework for Language Modeling”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[16] A. M. Elbir. “Hybrid Federated and Centralized Learning”. In: *arXiv preprint arXiv:2011.06892* (2020).

[17] S. Caldas, J. Konecný, H. McMahan, and A. Talwalkar. “Expanding the Reach of Federated Learning by Reducing Client Resource Requirements”. In: *arXiv preprint arXiv:1812.07210* (2018).

[18] Y. Liu, X. Yuan, R. Zhao, Y. Zheng, and Y. Zheng. “RC-SSFL: Towards Robust and Communication-efficient Semi-supervised Federated Learning System”. In: *arXiv preprint arXiv:2012.04432* (2020).

[19] H. M. et al. “Federated Learning of Deep Networks using Model Averaging”. In: *arXiv preprint arXiv:1602.05629* (2016).

[20] J. B. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *In Lucien M. Le Cam and Jerzy Neyman, editors, Proceedings of the Berkley symposium on mathematical statistics and probability* 1 (1967), pp. 281–297.

[21] P. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[22] L. Vendramin, R. Campello, and E. Hruschka. “Relative clustering validity criteria: A comparative overview”. In: *Statistical Analysis and Data Mining* 3 (2010), pp. 209–235.

[23] O. Baños, R. García, J. A. H. Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga. “mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications”. In: *IWAAL*. 2014.

[24] A. Reiss and D. Stricker. “Introducing a New Benchmarked Dataset for Activity Monitoring”. In: *2012 16th International Symposium on Wearable Computers* (2012), pp. 108–109.

[25] E. Babu, D. Rueckert, and A. Davison. “Federated Deep Learning for Healthcare Data”. In: *Department of Computing, Imperial College London* (June. 13, 2020).

Paper II

An Energy-aware Multi-Criteria Federated Learning Model for Edge Computing

Ahmed A. Al-Saedi, Emiliano Casalicchio, Veselka Boeva
In: 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud), 2021, pp. 134-143, Rome, Italy.

Abstract

The successful convergence of Internet of Things (IoT) technology and distributed machine learning have leveraged to realise the concept of Federated Learning (FL) with the collaborative efforts of a large number of low-powered and small-sized edge nodes. In Wireless Networks (WN), an energy-efficient transmission is a fundamental challenge since the energy resource of edge nodes is restricted. In this paper, we propose an Energy-aware Multi-Criteria Federated Learning (EaMC-FL) model for edge computing. The proposed model enables to collaboratively train a shared global model by aggregating locally trained models in selected representative edge nodes (workers). The involved workers are initially partitioned into a number of clusters with respect to the similarity of their local model parameters. At each training round a small set of representative workers is selected on the based of multi-criteria evaluation that scores each node representativeness (importance) by taking into account the trade-off among the node local model performance, consumed energy and battery lifetime. We have demonstrated through experimental results the proposed EaMC-FL model is capable of reducing the energy consumed by the edge nodes by lowering the transmitted data.

1 Introduction

The development of Machine Learning (ML) techniques has shown great success in many tasks and applications, especially when powered by other technologies like

IoT [1]. Unfortunately, training ML models in central cloud produces a high communication cost, which is not suitable for many resource-constrained IoT devices, and introduce privacy concerns as well. A distributed ML framework has been proposed by Google to move beyond the cloud-centric ML, which is called Federated Learning (FL), provides collaborative learning of a shared ML model, with the help of a large number of edge devices (workers hereafter). Meanwhile, Edge computing is a natural computational paradigm to run FL algorithms, specifically when used to deploy IoT applications. In FL context, edge devices such as smart phones, tablets and sensors' boards download a shared ML model, improve it by learning using their local data, and then upload only their updated models through the base station (BS) to the cloud for global model aggregation [2, 3]. Image classification in Vehicular Edge Computing [4], End-to-End Autonomous Driving [5], the Virtual Keyboard [6], out-of-vocabulary word learning [7] and Human Activity Recognition (HAR) [8] are examples of edge computing applications that use FL to improve model accuracy and reduce latency in decision making. Furthermore, Artificial intelligence (AI) and FL are also proposed as innovative approaches for edge computing platform management tasks like privacy-preserving optimized service placement at the edge [9], or to optimize caching and communication in edge computing [10], [11]. In practice, to employ FL over IoT networks, edge nodes must repeatedly transmit their local FL models to a BS via wireless links [12]. Due to the constrained resources of IoT devices compared to resources in the cloud, reduction of the energy consumed by edge nodes while preserving the quality of the decision making is one of the main challenges in AI-based edge computing applications (e.g., [13]) Let us consider the edge computing scenario represented in Figure 1 with heterogeneous edge nodes connected to a cloud datacenter by means of (BSs). Each edge node has its own physical characteristics in terms of networking capabilities and energy consumption. We assume that the edge nodes run an application that leverages FL model to make decisions (e.g. image classification or HAR using a large amount of sensor data). Motivated by the aforementioned challenges, we propose a FL model, entitled Energy-aware Multi-Criteria Federated Learning (EaMC-FL), that reduces the energy consumed by the edge node and transmitted data and additionally tries to improve the quality of the local ML model. The proposed FL model updates the global model only with local model parameters from few edge nodes that are considered to be representative. Such nodes are selected at each training round by identifying a trade-off between the quality of the local model produced and the physical characteristics of the node. The proposed EaMC-FL algorithm can be considered as an energy-aware improvement of the FL algorithm proposed in [2], FL-baseline hereafter. Hence, the EaMC-FL algorithm is benchmarked against the FL-baseline on two publicly available physical activity monitoring datasets under different experimental scenarios. These datasets have been selected since human activity monitoring and recognition is one of the IoT application field that benefits of federated learning algorithms that reduce energy consumption at the edge. Experimental results demon-

strate that our model is capable of reducing the consumed energy by a factor between 5 and 12 with respect the FL-baseline model without compromising the quality of the built ML model. The paper is organized as follows. Related works are analyzed in Section 2. Backgrounds on Federated Learning and edge node (worker) selection policy are presented in Section 3. The problem addressed is stated in Section 4 along with the description of the energy model adopted. The proposed EaMC-FL approach is described in Section 5. Finally, the experimental settings and the obtained evaluation results are presented and analysed in Section 6.

2 Related Work

The communication overhead of FL mostly arises from FL's iterative nature and from the global model aggregation, which can be alleviated by effective resource allocation [14]. Recent research works address the FL challenges of reducing communication overhead (e.g., [13], [15], [16], [17], [18], [19]) and energy consumption (e.g., [13], [16], [20], [21]). In [15] the authors propose to use analog aggregation to reduce energy consumption, that is the summation of local models can be carried out over-the-air if workers synchronize with each other and align the transmit power. That solution requires to properly schedule workers and to leverage network channel transmission properties. The authors do not compare their results against other energy aware solutions neither against the FL-baseline model [2]. A different approach to minimize energy consumption is proposed in [16]. In that work the authors have formulated an optimization problem whose goal is to minimize the total energy consumption of the system under a latency constraint. Results of the proposed solution are compared with a traditional federated learning schema over wireless channels. CMFL [17] is a FL model that let clients to send updates only if relevant enough to model improvement. That reduce the network footprint. The obtained results are

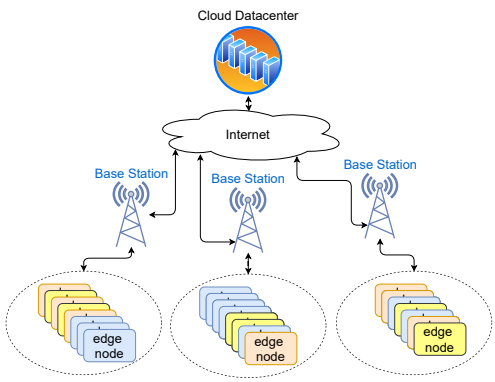


Figure 1: The Edge computing scenario considered. Edge node with different colors have different hardware features/capabilities

compared with the FL-baseline. Gaia [18] is a geo-distributed ML system that employs an intelligent communication mechanism to efficiently utilize the WAN bandwidth, while retaining the accuracy and correctness guarantees of an ML algorithm. Error accumulation method is considered to decrease the communication cost in [19]. A neural-structure-aware resource management approach with module-based federated learning is proposed in [13]. That solution assigns mobile clients with different subnetworks of the global model according to the status of their local resources with the goal of using more efficiently client resources, included network bandwidth. The proposed solution is compared against the traditional FL approach. A new approach for energy-efficient radio resource management (RRM) for federated learning is proposed in [20]. RMM performs energy-efficient bandwidth allocation and scheduling while preserving learning performance. RMM is benchmarked against the FL-baseline algorithm. HybridFL [21] adopts model aggregation at edge level and cloud level. Their client selection strategy for model update improves the FL training process significantly in terms of shortening the federated round length, speeding up the global model's convergence and reducing end device energy consumption. Similarly to the solutions proposed in [13], [17], [21], we achieve energy efficiency by proposing an new edge node selection policy that splits edge nodes in optimal clusters based on local model similarity and that selects few representatives of the clusters trading-off model accuracy, battery lifetime and energy consumption. The selection of representative is heuristic, that has the advantage of not requiring the solution of complex optimization problems. The drawback is the achievement of a sub-optimal energy efficiency, that however is enough to outperform the FL-baseline model.

3 Background

In this section, we first introduce the FL-baseline model [2]. We then provide a definition of k-medoids clustering algorithm used for initial grouping of the workers. Finally, we explain the Silhouette Index validation measure that is applied at each round for updating the worker clustering by assessing whether the representatives are still well tied to their current clusters.

3.1 Federated learning

As it has already been mentioned, the proposed EaMC-FL algorithm can be considered as an energy-aware improvement of the federated learning model introduced in [2]. Hence the latter FL-baseline algorithm (also abbreviated as FedAvg) is used as benchmark for EaMC-FL.

In a FL scenario, the learning task is conducted by having a coordinator (also referred as server and usually running in a cloud datacenter) that, throughout a set of iterations, collaborates with the participating workers to learn and build a shared ML

model [22]. In detail, a model M is learned iteratively by using a randomly selected subset of workers $V_s \subset V$. The workers in V_s participate at each round and compute the gradient of the loss over the data held by them. Each worker $v \in V_s$ at round t has its own data D_v^t and a local model m_v^t. At each round t, each worker trains its local model by iterating the local update multiple times of Stochastic Gradient Descent (SGD) before sending the next local model m_v^{t+1} to the server. The server, after aggregating the local models computed at round t, performs a synchronous update of the global model M_{t+1}. The global model update can be computed using different criteria. In this paper, we assume it is evaluated by means of federated averaging, that is the local model m_v^t, $v \in V_s$ and global model M_t are updated by the following equations: [23]:

$$m_v^{t+1} = m_v^t - \eta g_v^t \tag{1}$$

$$M_{t+1} = \sum_{v \in V_s} \frac{n_v}{n} m_v^{t+1}, \tag{2}$$

where m_v^{t+1} is the local update, g_v^t are the updated weights on its local data at the current model m_v^t, M_{t+1} is the next global model, η is a learning rate computed by each worker, n is the sum of all data points and n_v is the number of local data points. The server then distributes the global model M_{t+1} to the workers that can perform another iteration of local training and model update.

3.2 K-medoids clustering

K-medoids clustering [24] allows to split the available workers (edge nodes) into groups of similar workers with respect to their local updates. The medoid is the most centrally located point in a given cluster. Due to this in the EaMC-FL algorithm, the medoid of each cluster can be used as the representative of the cluster. A challenge in the partitioning algorithms like k-medoids is the identification of the number of clusters k in advance. One solution that may apply in such cases is to build a clustering model with a range of values for k and then evaluate the quality of the generated clustering partitions. For example, different internal cluster validation indices [25] (such as Silhouette Index introduced below) can be applied to recognize the optimal number of clusters.

3.3 Silhouette Index

The proposed EaMC-FL algorithm uses Silhouette Index (SI) [26] at each iteration round for assessing whether the worker representatives are still well tied to their current clusters. The Silhouette Index is a well-known and widely used internal cluster validation technique that can be applied for assessing compactness and separation properties of a given partitioning [26]. Suppose a_i represents the average distance of object i from all the other objects in the cluster to which the object i is assigned,

and b_i represents the minimum of the average distances of object i from objects of the other clusters. Then the Silhouette Index $s(i)$ of object i can be calculated as

$$s(i) = (b_i - a_i)/\max\{a_i, b_i\}\cdot \tag{3}$$

$s(i)$ measures how well object i matches the clustering at hand. $s(i) \in [-1, 1]$ and higher value points out a better quality of the clustering results. For example, when $s(i)$ is close to 1 this means that object i is assigned to a very appropriate cluster. A situation is different when $s(i)$ is about zero. Namely, object i lies between two clusters. The worst case is when $s(i)$ is close to -1. Evidently, this object has been misclassified.

4 Problem statement

As mentioned earlier, in a FL setting when workers are edge nodes with limited battery lifetime, a reduction of the amount of transferred data to the master node in the cloud will decrease the energy consumed. Indeed, as outlined by [27] data transfer over radio channels is the process with higher energy consumption impact.

In this paper, we address the problem of reducing the quantity of data transferred by the workers to the master node and vice-versa trading-off the worker local model performance, the energy consumed and the battery lifetime.

4.1 Energy model

For an edge node i, acting as a worker in a FL scenario, the power required to transmit forth and back the model parameters is $P_i^t = P_i^s + P_i^r$ Watt (W), where superscript t, s and r stand for transmit, send and receive, respectively. The energy E_i^t consumed by the edge node i in a time interval t is the power used in t, hence

$$E_i^t = E_i^s + E_i^r = P_i^s \times t^s + P_i^r \times t^r, \tag{4}$$

where t^s and t^r are respectively the time needed to transmit the data and to receive the data, and E_i^s (E_i^r) is the energy consumed for sending (receiving) data. The battery capacity, or energy budget, of an edge node is B_i Joule (J). Then, the battery lifetime (in seconds) is

$$L_i = B_i/P_i^t\cdot \tag{5}$$

We recall that $1W = 1J/sec$. Each time an edge device sends/receives data for a certain amount of time t^s (t^r) the battery is drained of E_i^s (E_i^r) Joule. At one point in time T it is useful to express the available energy budget (or battery lifetime) as a percentage

$$L_i^a(T) = \frac{1}{B_i}\left(B_i - \sum_{\tau=0}^{T} E_i^t(\tau)\right), \tag{6}$$

assuming $\sum_{\tau=0}^{T} E_i^t(\tau)$ is always less or equal than B_i. For example, if $B_i = 100J$, $P_i^s = P_i^r = 335.5\mu W$ then $L_i = 0.13 \times 10^6 secs$ that is approximately 36 hours. If the edge node at time $t = 0$ starts transmitting for 50 seconds and receiving for other 50 seconds, the energy drained at time $T \geq 100$ sec is $33550\mu J$ and $L_i^a(T) = 0.99\%$. Equation 6 allows to decide whether a node is suitable to transmit the model parameters. Indeed if $\sum_{\tau=0}^{T'} E_i^t(\tau) \geq B_i$ the node cannot be selected to send (receive) data at time T'.

The final step is to define E_i^t as a function of the size of the model parameters transmitted, and of the edge node characteristics like network bandwidth and latency. Let assume the size of the model parameters is s bytes, the network bandwidth for node i is b_i bytes per second, and the round trip network latency experienced in the communication with the master node is l_i seconds, the transmission time is

$$\tau_i = 2 \times \frac{s}{b_i} + l_i \cdot \tag{7}$$

Assuming that $P_i^s = P_i^r = P_i$, Eq. 4 can be expressed as:

$$E_i^t = P_i \times \left(2 \times \frac{s}{b_i} + l_i\right) \cdot \tag{8}$$

In summary, Eq. 6, Eq. 8 and F-measure (see Section 6.1) are evaluation criteria that can be used to select an appropriate node for transmitting model parameters to the master node.

5 Multi-Criteria Federated Learning

5.1 Evaluation Criteria and Selection Policy

As introduced above, an edge node i is described by E_i^t, $L_i^a(T)$ (or L_i), along with a metric I_i that evaluates the model performance, e.g. F-measure can be used for this purpose. This three criteria are used to select an edge node as worker representative for a cluster of workers. Considering that for a node i the transmission power P_i, the size s of the model and the bandwidth b_i are constant where the latency l_i slightly varies from node to node in a given time frame, since there can be several factors such as edge node characteristics that can lead to a variation in the latency value in a realistic environment, the worker i score can be calculated as follows:

$$S_i = w_1 \times I_i + w_2 \times L_i^a(T) + w_3 \times \frac{E_i^* - E_i^t}{E_i^*}, \tag{9}$$

where $w_j \in [0, 1]$, $j \in \{1, 2, 3\}$ and $\sum_{j=1}^{3} w_j = 1$, are relative weights that define the importance of the evaluation criteria. I_i and $L_i^a(T)$ are adimensional and vary in the range $[0, 1]$. Note that higher values mean a better quality of the model and more

battery budget available to transmit model parameters to the master coordinator. The third term is an adimentional and normalization (in $[0, 1]$) factor accounting for the energy consumed to transfer the model parameters to the master coordinator in the cloud. Higher is its value lower is the energy consumed: $E_i^* = P_i(2(s/b_i) + 1)$ is an upper bound of the energy consumed by the edge node i when the round trip network latency is equal to 1 second, thas is the worst case, because the round trip latency between the edge and the cloud is usually of the order of few hundred of milliseconds [28] and values of the order of seconds means the master coordinator is not reachable (we assume it is always reachable). Hence S_i is an adimentional number in the range $[0, 1]$ used as an evaluation criteria for the workers' selection. The selection policy is assumed to be implemented in a cloud, i.e. a central node computes the S_i scores.

Based on the above assumptions, the **cluster representative selection policy** at round t is described in what follows. Let $C^t = \{C_1^t, C_2^t, \ldots, C_k^t\}$ is the clustering at round t, k is the number of clusters and $\{v_{j1}^t, v_{j2}^t, \ldots, v_{jk_j}^t\}$ is the set of edge nodes belonging to cluster C_j^t, $j = 1, 2, \ldots, k$. Then

1. For each $C_j^t \in C^t$
   - compute $\bar{C}_j^t = C_j^t \setminus \{v_{ji}^t \mid l_{ji}^t \geq 1\}$
   - for each $v_{ji}^t \in \bar{C}_j^t$ compute the edge node scores S_i by applying Eq. 9
2. For each $\bar{C}_j^t$ select the highest scored edge node $v_{ji^*}^t$ for which $L_{i^*}^a(T) > 0$ (to be conservative could be selected a node with $L_{i^*}^a(T) > X\%$).

5.2 Energy-aware multi-criteria federated learning algorithm

Let us assume N edge nodes are involved in training of a FL model. Initially, the three evaluation criteria defined in Section 5.1 (see also Section 6.1) are analyzed and relatively weighted with respect to the studied applied scenario, i.e. weights w_1, w_2 and w_3 are defined. We denote with $V = \{v_1, v_2, \ldots, v_N\}$ the set of available workers. A set V^t ($t = 0$ and $\mid V^t \mid < N$) of randomly selected workers is initially used to build the initial global model M_t. The local updates received at the initialization round $\{m_v^t \mid v \in V^t\}$ are analyzed and partitioned into k groups of similar updates by applying k-medoids. The basic operations conducted by our EaMC-FL algorithm at each round t ($t > 0$) are explained below:

1. **Multi-criteria evaluation of workers:** Each worker $v \in V^{t-1}$ is evaluated under the predefined criteria by calculating its edge score S_v (see Eq. 9);
2. **Selection of representative workers:** The workers in each cluster C_i^{t-1}, $i = 1, 2, \ldots, k$ are ranked in descent order with respect to the calculated edge

scores. Using the selection policy defined in Section 5.1 the admissible top-ranked worker is selected as a representative of each cluster $C_i^{t-1}, i = 1, 2, \ldots, k$. In that way, the selected representatives form a new set of workers $V^t = \{v_1^t, v_2^t, \ldots, v_k^t\}$, where $k << \mid V^0 \mid$.

3. **Updating of local models:** Each worker $v \in V^t$ receives the global model M_{t-1} and optimizes its parameters locally (Eq. 1), i.e. m_v^t update is produced and sent back to the server;

4. **Building of global model:** M_t is built by aggregating the local updates $\{m_v^t \mid v \in V^t\}$ received at round t (Eq. 2);

5. **Updating of clustering model:** C^{t-1} is updated by evaluating the local update of each $v \in V^t$ and whether it is still tied to its current cluster or must be assigned to an other (Eq. 3). As a result a new updated clustering C^t is produced.

Then steps 1 – 5 are repeated until reaching either convergence or the maximum number of rounds (T). The proposed EaMC-FL algorithm is illustrated in Figure 2.

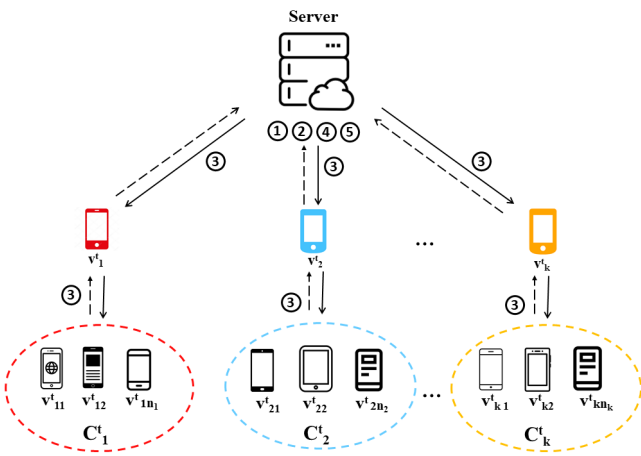


Figure 2: A schematic illustration of EaMC-FL algorithm. A global model is trained on each representative's data and averaged at the server of each iteration phase. Circled numbers correspond to the basic operations.

6 Experimental Evaluation

6.1 Data

For initial evaluation of the proposed EaMC-FL algorithm, we have used publicly available real-world datasets from UCI Machine Learning repository. We use mHealth [29]

and Pamap2 [30] physical activity monitoring datasets. The both datasets involve motion sensor data of several physical activities. These provide an opportunity to simulate a multi-class classification problem of HAR, which is a suitable machine learning task for studying FL scenarios.

The mHealth dataset has 215040 instances distributed in 7 classes, while Pamap2 dataset has 16819 instances that contain information about 18 different physical activities. subsectionEvaluation metrics The evaluation metrics that are used to score the importance of each node and then select a representative of each group of workers are F1-score, consumed energy, and energy budget. The evaluation measures used for the definition and evaluation of the EaMC-FL algorithm are explained hereafter.

- **Sum of energy consumed per round:** We calculate the aggregation of energy consumed by all workers per round. In order to measure the energy consumption in the FL-baseline we sum the energy consumed of all workers that have participated in training of the shared model at each round, while in EaMC-FL, the sum of energy consumed by all representative workers is calculated.

- **Average of energy budget per round:** The energy budget or battery lifetime in the FL-baseline is calculated by the average energy budget of all workers in each round until a global model is converged. Similarly, the energy budget in EaMC-FL is calculated by averaging the energy budget of all representative workers in each round.

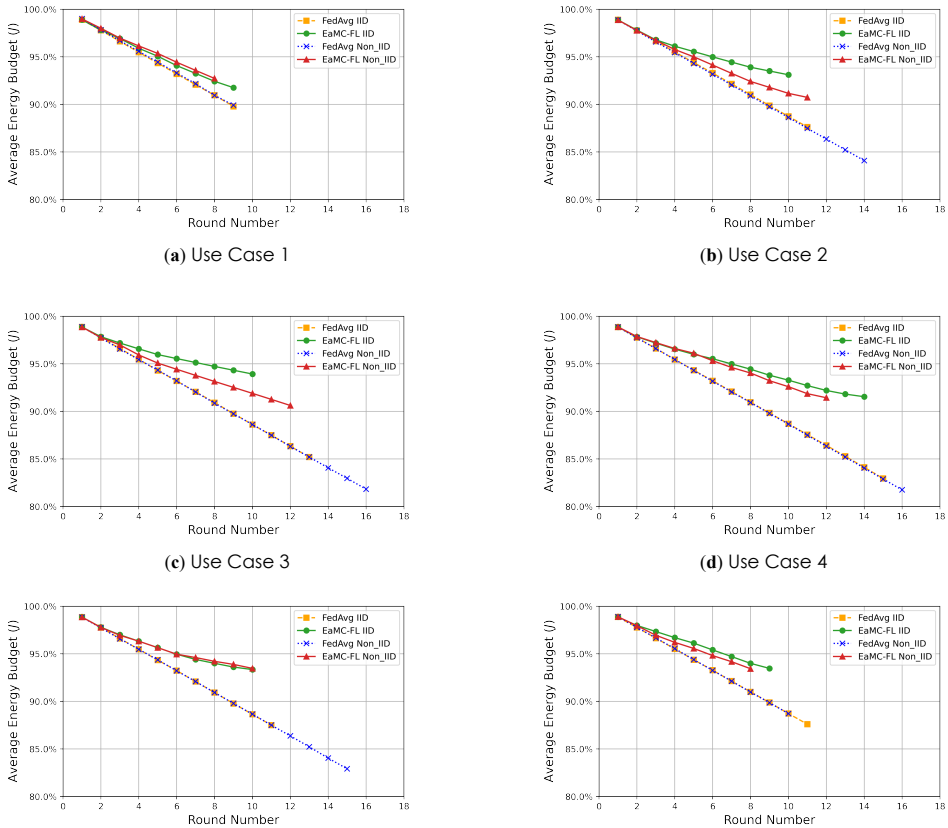
- **F1-score:** F-measure is the harmonic mean of precision and recall values for each class. The scikit-learn of the F-measure (*micro-average* F_1) has been used to evaluate the accuracy of EaMC-FL and FedAvg models.

- **Number of rounds to converge:** In FL context, the training process consists of iterative rounds. It is repeated until a stopping criteria is met. In our experimental scenarios, the training process stops when the difference between the accuracy scores produced by two consecutive training rounds is less than 0.001 or when number of iterations exceeds 100.

- **Number of representatives per round:** The initial number of representatives (edge nodes) is equal to the optimal number of clusters identified for each dataset. At each training round this number can change due to the clustering update.

6.2 Experimental setting

In order to simulate a distributed scenario each dataset is used to generate 10 experimental datasets by randomly separating the data points into a number of groups



(**a**) Use Case 1

(**b**) Use Case 2

(**c**) Use Case 3

(**d**) Use Case 4

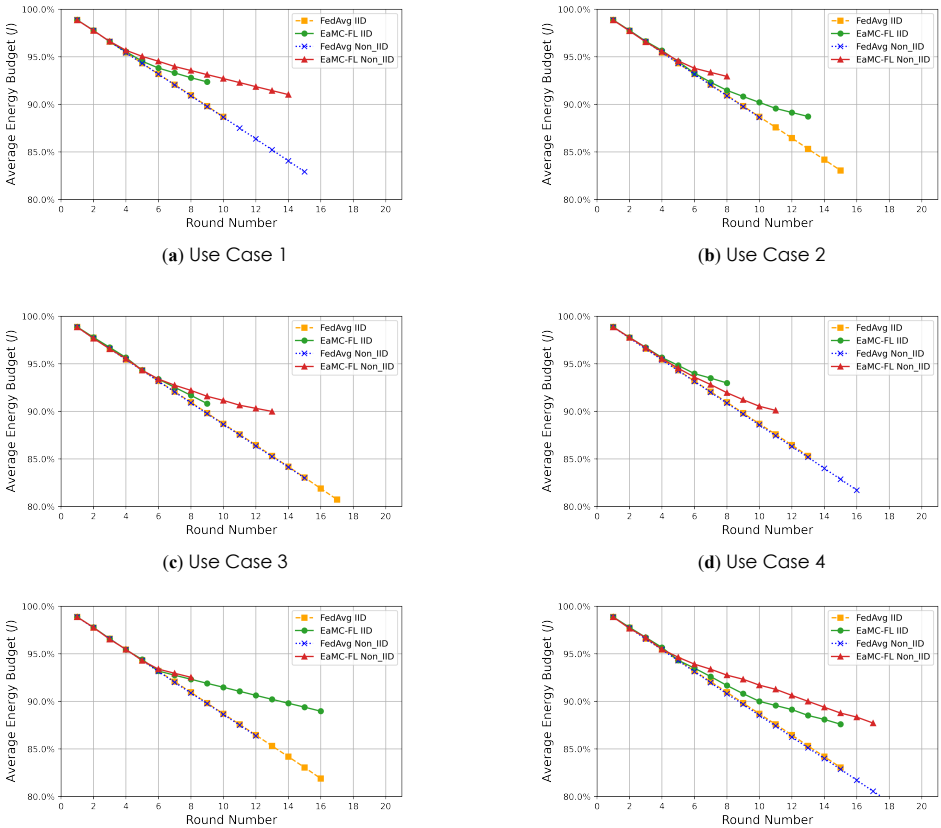
(**e**) Use Case 5

(**f**) Use Case 6

Figure 3: Average energy with the total number of rounds to achieve the convergence on Pamap2 dataset.

(workers). Each group is supposed to represent data supplied by one worker. In addition, we have studied two different experimental data distribution scenarios: IID (Independent and Identically Distributed) and non-IID. In that way for each studied number of workers (20, 30, 40, 50 and 60) and each used dataset (mHealth and Pamap2) we have built 20 experimental datasets, i.e. 200 experimental datasets in total.

We benchmark the EaMC-FL algorithm against the FL-baseline [2] on the built experimental datasets under six different use cases (see Table 1). Each use case presents a different multi-criteria evaluation scenario. For example, in the first use case the three evaluation criteria (model accuracy, energy consumption and battery lifetime) are equally important. In the second use case, the battery lifetime is prioritized ($w_2 = 0.5$), i.e. we care more about the consumed energy and battery lifetime than accuracy. While the accuracy is more important in use case 3 ($w_1 = 0.5$). Use cases 4, 5 and 6 present more extreme scenarios in which one of the three evaluation



(a) Use Case 1 (b) Use Case 2

(c) Use Case 3 (d) Use Case 4

(e) Use Case 5 (f) Use Case 6

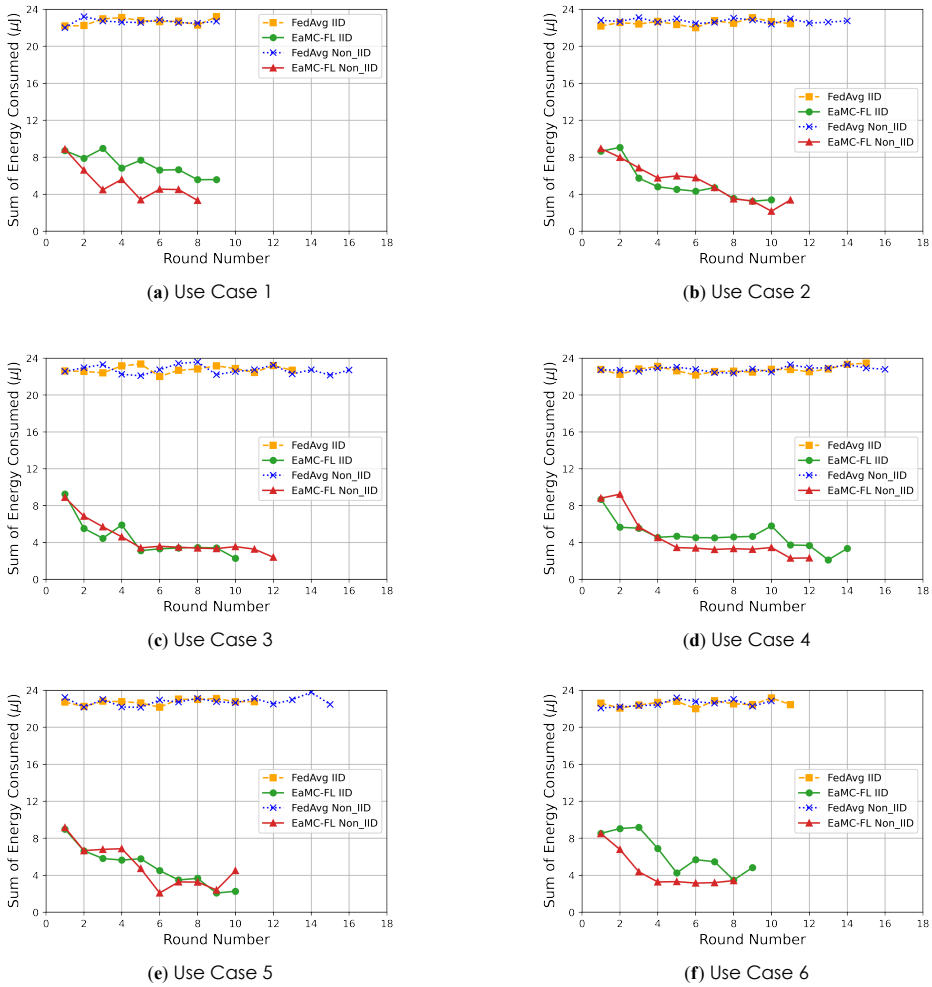
Figure 4: Average energy budget with the total number of rounds to achieve the convergence on mHealth dataset.

criteria is neglected, i.e. the corresponding weight is zero.

Table 1: Multi-criteria evaluation scenarios

Use case	w_1	w_2	w_3
1	0.33	0.33	0.33
2	0.2	0.5	0.3
3	0.5	0.3	0.2
4	0	0.5	0.5
5	0.9	0.1	0
6	0	0.9	0.1

The FL-baseline and EaMC-FL algorithms are implemented in Python using Scikit-learn library. Stochastic Gradient Descent (SGD) classifier is used as a training model in the two compared FL algorithms. The machine learning task is a multi-class classification problem of human activity recognition. Each worker trains its local model for a number of epochs e on its local dataset using sklearn as the ML library. The updated local model is then sent back to the coordinator, which updates



(**a**) Use Case 1 (**b**) Use Case 2

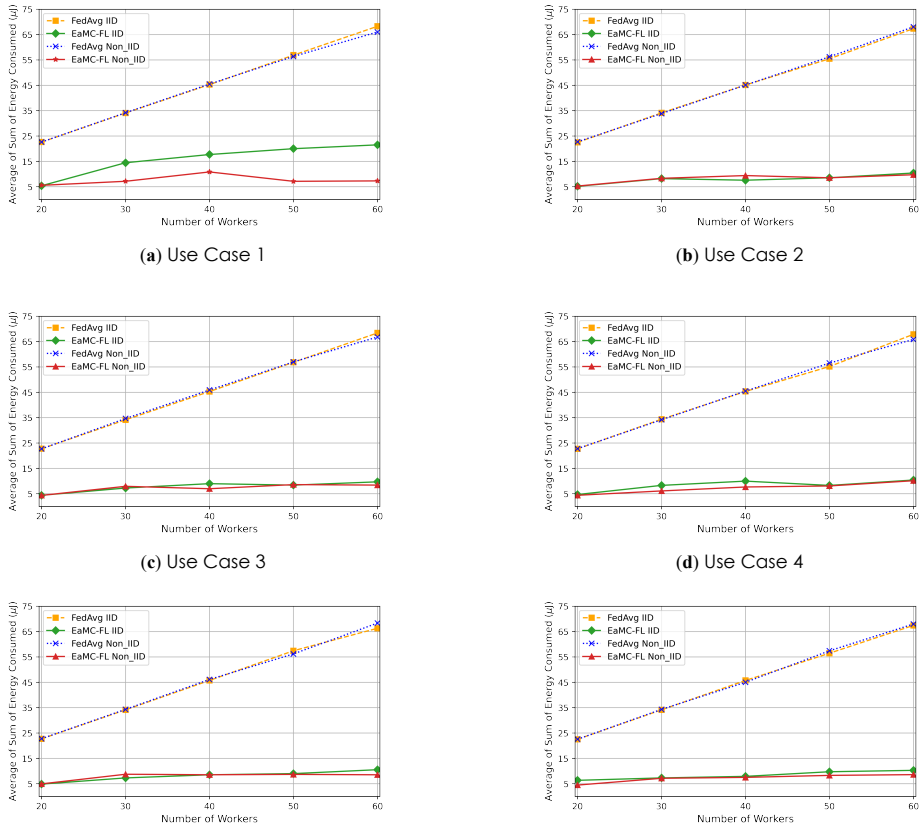
(**c**) Use Case 3 (**d**) Use Case 4

(**e**) Use Case 5 (**f**) Use Case 6

Figure 5: Sum of energy consumed with the total number of rounds to achieve the convergence on Pamap2 dataset.

the global model and sends it again to the workers. The process continues and conducts until the model is converged.

In addition, in our experimental scenarios, the size of local models the edge nodes send to the server at each iteration round is constant, 2.447 kB and 6.575 kB for mHealth and Pamap2 datasets, respectively. Furthermore, the energy consumed by the edge node of a model transmission on average is 1.17888 (J), whereby a random value between 43 ms and 54 ms is assigned for the latency per worker during transmission. Finally, we assume the simulated edge devices have homogeneous battery power.



(**a**) Use Case 1

(**b**) Use Case 2

(**c**) Use Case 3

(**d**) Use Case 4

(**e**) Use Case 5

(**f**) Use Case 6

Figure 6: Average of energy consumed with the total number of rounds to achieve the convergence on Pamap2 dataset.

6.3 Results and analysis

We initially compare FL-baseline (FedAvg) and EaMC-FL algorithms with respect to the average energy budget under the six studied evaluation scenarios (see Table 1) by applying the algorithms on the experimental datasets of Pamap2 [30] and mHealth [29] for 20 numbers of workers under two data distribution scenarios: IID and non-IID. The obtained results are plotted in Figures 3 and 4, respectively. One can notice that for both datasets EaMC-FL outperforms FedAvg in all studied experimental scenarios. It is interesting to observe that EaMC-FL demonstrates different behaviour with respect to the two data distribution scenarios (IID versus non-IID) in two datasets. Namely, EaMC-FL IID has generated higher values than EaMC-FL Non IID in 2/3 of the use cases of Pamap2 dataset (see Figure 3), while EaMC-FL Non IID outperforms EaMC-FL IID on mHealth dataset again in 2/3 of the use cases (see Figure 4). We have also compared the FedAvg and EaMC-FL algorithms un-

der the same experimental scenarios as ones studied in Figures 3 and 4, but with respect to the sum of energy consumed of the workers over a number of rounds until the model is converged. The results produced by EaMC-FL for both datasets in all studied experimental scenarios are significantly better than the ones of FedAvg. In addition, in comparison with the FedAvg algorithm the EaMC-FL algorithm in most of the evaluated use cases needs less or at most equal number of rounds for training the shared model. This is expected to reduce further the total energy consumed for the ML model training. Due to the space constraint we present in Figure 5 only the results generated on the Pamap2 experimental datasets. The results obtained on mHealth data are similar. Table 2 presents the accuracy scores (Accuracy) of the FedAvg and EaMC-FL models, the average of energy consumed over the total number of rounds (Energy) and the corresponding number of rounds (Rounds) needed for the model convergence generated in the six evaluation scenarios on Pamap2 experimental datasets for IID and Non-IID data distribution settings, respectively. Note that in case of FedAvg, the coordinator randomly chooses 20 workers out of 40 workers for each training round. In both data distribution settings, the EaMC-FL model produces higher accuracy scores on the whole six evaluation scenarios than the FedAvg and in most of the use cases for less number of rounds. It is also interesting to analyse and benchmark the results (in Table 2) generated in evaluation scenarios 2, 4 and 6 against those produced by use cases 3 and 5. The former ones assign higher importance to energy consumption and battery lifetime than the latter ones, which prioritise the model performance. Logically, the highest accuracy scores generated by the EaMC-FL model in both data distribution settings are for use cases 3 and 5, i.e. the ones that assign higher importance to the model accuracy. It is also interesting to notice that in both data distribution settings the lowest amount of consumed energy is in use cases 3 and 4. The former use case balances between the model accuracy and energy consumption and budget while the latter one neglects the model accuracy. Curiously, the smallest number of rounds is needed in two use cases that are again the same for both data distribution scenarios, namely 1 and 6. Remind that in use case 1 the three evaluation criteria are equally important while use case 6 prioritises the battery lifetime and neglects the model accuracy. Note that different trends are observed on mHealth experimental datasets in the different multi-criteria evaluation scenarios (use cases in Table 1). Hence this is worth to be further studied maybe in a larger scale real-world data context, which is part of our future work.

Figure 6 shows the average of total energy consumed over the needed training rounds with respect to different number of workers for the Pamap2 dataset. The results produced on the mHealth experimental dataset are similar. One can clearly see that for the FL-baseline algorithm (FedAvg), the total energy consumed increases with the number of workers that participating in federated learning, which is not as much as that of the EaMC-FL algorithm. This is due to the fact that more energy must be used by the workers for model parameters transmission. In summary, EaMC-FL outperforms FL-baseline in terms of total energy consumption with respect to

Table 2: EaMC-FL and FedAvg accuracy scores, average of energy consumed with the total number of rounds and number of rounds to achieve the convergence in the six evaluation scenarios on Pamap2 experimental datasets

IID Setting						
	FedAvg			**EaMC-FL**		
Case	Accuracy	Energy	Rounds	Accuracy	Energy	Rounds
1	0.950	22.633	9	0.965	5.419	**9**
2	0.949	22.516	11	0.963	5.205	10
3	0.946	22.769	13	**0.968**	**4.413**	10
4	0.948	22.737	15	0.964	**4.723**	14
5	0.952	22.732	11	**0.966**	4.886	10
6	0.944	22.550	11	0.963	6.375	**9**

Non-IID Setting						
	FedAvg			**EaMC-FL**		
Case	Accuracy	Energy	Rounds	Accuracy	Energy	Rounds
1	0.940	22.640	9	0.964	5.588	**8**
2	0.942	22.736	14	0.963	5.307	11
3	0.923	22.724	16	**0.967**	**4.379**	12
4	0.941	22.815	16	0.964	**4.419**	12
5	0.935	22.787	15	**0.965**	4.974	10
6	0.950	22.574	10	0.963	4.505	**8**

different number of workers in both data distribution settings.

7 Conclusion and Future Work

In this paper, we have investigated the problem of energy consumed and energy budget of edge nodes of FL over wireless networks. The presented work is an approach to control and optimize communication of FL in the wireless network. We have proposed an Energy-aware Multi-Criteria Federated Learning (EaMC-FL) algorithm for edge computing. The EaMC-FL algorithm has been benchmarked against the FL-baseline (FedAvg) algorithm on two publicly available physical activity monitoring datasets under different experimental scenarios. The obtained experimental results have shown that EaMC-FL outperforms FedAvg in terms of total energy consumption, energy budget, and model accuracy.

Our future aim is to carry out further evaluation of the proposed EaMC-FL algorithm by studying its potential on much larger scale real-world datasets and other applied FL scenarios. In addition, we plan to compare the proposed EaMC-FL algorithm with some other state-of-the-art energy-aware FL models.

Acknowledgments

The first author is supported by an Iraq Ministry of Higher Education and Scientific Research PhD Scholarship. The work of E.Casalicchio is partially supported by the

SmartDefense Project (Ricerca Ateneo 2019). The work of V. Boeva is partially supported by the MIRAI ITEA3 (19034) 2020 Project.

References

[1] I. J. Goodfellow, Y. Bengio, and A. C. Courville. “Deep Learning”. In: *Nature* 521 (2015), pp. 436–444.

[2] H. B. M. et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *International Conference on Artificial Intelligence and Statistics*. 2016.

[3] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik. “Federated Optimization: Distributed Machine Learning for On-Device Intelligence”. In: *ArXiv* abs/1610.02527 (2016).

[4] D. Ye, R. Yu, M. Pan, and Z. Han. “Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach”. In: *IEEE Access* 8 (2020), pp. 23920–23935.

[5] Z. Chen and X. Huang. “End-to-end learning for lane keeping of self-driving cars”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 1856–1860. DOI: 10.1109/IVS.2017.7995975.

[6] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. “Federated Learning for Mobile Keyboard Prediction”. In: *CoRR* abs/1811.03604 (2018). arXiv: 1811.03604.

[7] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays. “Federated Learning Of Out-Of-Vocabulary Words”. In: *CoRR* abs/1903.10635 (2019). arXiv: 1903.10635.

[8] K. Sozinov, V. Vlassov, and S. Girdzijauskas. “Human Activity Recognition Using Federated Learning”. In: *2018 IEEE ISPA*. 2018, pp. 1103–1111.

[9] Y. Qian, L. Hu, J. Chen, X. Guan, M. M. Hassan, and A. Alelaiwi. “Privacy-aware service placement for mobile edge computing via federated learning”. In: *Information Sciences* 505 (2019), pp. 562–570.

[10] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen. “In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning”. In: *IEEE Network* 33.5 (2019), pp. 156–165.

[11] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas. “Federated Learning Based Proactive Content Caching in Edge Computing”. In: *2018 IEEE Global Communications Conference*. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647616.

[12] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang. “Toward an Intelligent Edge: Wireless Communication Meets Machine Learning”. In: *IEEE Communications Magazine* 58 (2020), pp. 19–25.

[13] R. Yu and P. Li. “Toward Resource-Efficient Federated Learning in Mobile Edge Computing”. In: *IEEE Network* 35.1 (2021), pp. 148–155. DOI: 10.1109/MNET.011.2000295.

[14] H. H. Yang, Z. Liu, T. Quek, and H. Poor. “Scheduling Policies for Federated Learning in Wireless Networks”. In: *IEEE Transactions on Communications* 68 (2020), pp. 317–333.

[15] Y. Sun, S. Zhou, and D. Gündüz. “Energy-Aware Analog Aggregation for Federated Learning with Redundant Data”. In: *2020 IEEE ICC* (), pp. 1–7.

[16] Z. Yang, M. Chen, W. Saad, C. Hong, and M. Shikh-Bahaei. “Energy Efficient Federated Learning Over Wireless Communication Networks”. In: *IEEE Trans. on Wireless Communications* 20 (2021), pp. 1935–1949.

[17] L. Wang, W. Wang, and B. Li. “CMFL: Mitigating Communication Overhead for Federated Learning”. In: *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[18] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. Ganger, P. B. Gibbons, and O. Mutlu. “Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds”. In: *NSDI*. 2017.

[19] M. Amiri and D. Gündüz. “Federated Learning Over Wireless Fading Channels”. In: *IEEE Transactions on Wireless Communications* 19 (2020), pp. 3546–3557.

[20] Q. Zeng, Y. Du, K. Leung, and K. Huang. “Energy-Efficient Radio Resource Allocation for Federated Edge Learning”. In: *2020 IEEE International Conference on Communications Workshops* (2020), pp. 1–6.

[21] W. Wu, L. He, W. Lin, and R. Mao. “Accelerating Federated Learning Over Reliability-Agnostic Clients in Mobile Edge Computing Systems”. In: *IEEE TPDS* 32.7 (2021), pp. 1539–1551. DOI: 10.1109/TPDS.2020.3040867.

[22] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. “Federated Learning: Strategies for Improving Communication Efficiency”. In: *ArXiv* abs/1610.05492 (2016).

[23] H. M. et al. “Federated Learning of Deep Networks using Model Averaging”. In: *arXiv preprint arXiv:1602.05629* (2016).

[24] J. B. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *In Lucien M. Le Cam and Jerzy Neyman, editors, Proceedings of the Berkley symposium on mathematical statistics and probability* 1 (1967), pp. 281–297.

[25] L. Vendramin, R. Campello, and E. Hruschka. “Relative clustering validity criteria: A comparative overview”. In: *Statistical Analysis and Data Mining* 3 (2010), pp. 209–235.

[26] P. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[27] X. Fafoutis and et al. “Extending the battery lifetime of wearable sensors with embedded machine learning”. In: *2018 IEEE 4th WF-IoT* (), pp. 269–274.

[28] S. Maheshwari, D. Raychaudhuri, I. Seskar, and F. Bronzino. “Scalability and Performance Evaluation of Edge Cloud Systems for Latency Constrained Applications”. In: *2018 IEEE/ACM SEC* (), pp. 286–299.

[29] O. Baños, R. García, J. A. H. Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga. “mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications”. In: *IWAAL*. 2014.

[30] A. Reiss and D. Stricker. “Introducing a New Benchmarked Dataset for Activity Monitoring”. In: *2012 16th International Symposium on Wearable Computers* (2012), pp. 108–109.

Paper III

Context-Aware Edge-Based AI Models for Wireless Sensor Networks- An Overview

Ahmed A. Al-Saedi, Veselka Boeva, Emiliano Casalicchio and Peter Exner
In: Emerging Sensor Communication Network-Based AI/ML Driven Intelligent IoT, Sensors, 2022.

Abstract

Recent advances in sensor technology are expected to lead to a greater use of wireless sensor networks (WSNs) in industry, logistics, healthcare, etc. On the other hand, advances in artificial intelligence (AI), machine learning (ML), and deep learning (DL) are becoming dominant solutions for processing large amounts of data from edge-synthesized heterogeneous sensors and drawing accurate conclusions with better understanding of the situation. Integration of the two areas WSN and AI has resulted in more accurate measurements, context-aware analysis and prediction useful for smart sensing applications. In this paper, a comprehensive overview of the latest developments in context-aware intelligent systems using sensor technology is provided. In addition, it also discusses the areas in which they are used, related challenges, motivations for adopting AI solutions, focusing on edge computing, i.e., sensor and AI techniques, along with analysis of existing research gaps. Another contribution of this study is the use of a semantic-aware approach to extract survey-relevant subjects. The latter specifically identifies eleven main research topics supported by the articles included in the work. These are analyzed from various angles to answer five main research questions. Finally, potential future research directions are also discussed.

1 Introduction

Recent advances in technology have had a great impact on today's digital world, surrounded by billions of intelligent sensors integrated with the Internet of Things

(IoT) [1, 2]. In such a complex dynamic environment, IoT devices, which usually have limited computing power and small memory capacity, can constantly generate huge amounts of data that can be analyzed in remote cloud data centers. Wireless Sensor Networks (WSN) are considered to be one of the key technologies used for data generation in IoT components.

However, transferring data from where it is generated to a data center increases communication overhead and bandwidth consumption, and also raises privacy concerns. Thus, the use of cloud processing alone is clearly not the most efficient approach for real time systems (e.g., health monitoring, autonomous driving, smart city) [3]. Thus, it is necessary to conduct the computation of the data collected by sensors as locally as possible, incorporating intelligence from edge devices, to move computation from cloud to edge [4]. This means placing a kind of artificial intelligence close to edge devices capable of processing complex behaviors and adapting to rapidly changing situations. In addition, more than ever before, microcontrollers are powerful enough to make intelligent decisions without any external help based on data collected from various sensors [5]. These devices can also analyze data, transmit data to low-latency actuators, and only transfer summarized information to the cloud [6]. In short, adding some sort of intelligence to the sensor nodes represents the next step in fulfilling an "awareness" level to the edge [7–10]. If data from different sensors are appropriately combined, the integrated data can be more precise, more reliable or simply provide a better understanding of the context in which the data was obtained [11]. Thus, sensor fusion will continue advancing in almost all applications, including security, logistics, voice recognition, object detection, etc. In such a sensor environment, most of these applications focus on performance metrics such as latency, reliability and even security [12–14].

As new wireless technologies such as WiFi Direct, 5G, Zigbee, LoRa, NB-IoT and LiFi are rapidly being developed, Edge Computing (EC) will soon help transition processing and analysis from the cloud to the edge [15–17].

1.1 Our Contribution

The purpose of this study is to provide an overview and understanding of the theoretical background, challenges, approaches, motivations, and gaps for the implementation of intelligent context awareness in wireless sensor networks. This document presents a literature review that covers the analysis of research documents published in the period between 2015 and January 2022 and available from the Scopus and Web of Science databases. In previous related reviews, we have not found an exhaustive work that explores deeply intelligent solutions using artificial intelligence (AI), machine learning (ML), or deep learning (DL) algorithms and their contributions to building the context-awareness in various WSN applications. Moreover, this study can be distinguished from the existing related surveys through the following key con-

tributions.

- Apply a semantic-aware approach to identify survey-relevant subjects.
- Identify and explore various AI/ML and DL methods that can be used in the establishment of a context awareness setting of sensor networks.
- Analyze key challenges and the research gaps found in the literature that need to be solved.
- Discuss the motivations for integration of intelligent context awareness in wireless sensor networks.
- Outline future research directions.

This investigation will fill the research gaps by comparing the included papers in terms of their challenges, strengths, limitations, motivations, and the way forward in the field. Furthermore, this review can help future researchers in identifying and exploring new perspectives in the field of sensor network context awareness field.

1.2 Organization of This Work

The rest of the paper is organized as follows. Section 2 introduces the technological context and discusses other surveys related to the topic of this study. Section 3 explains the methodology of this work. Section 4 analyzes the data extracted from included papers and discusses the results related to the defined research questions. Section 5 provides a discussion about industrial perspectives, context-aware challenges and corresponding intelligent solutions in a logistics use case. Finally, we conclude the paper with some open issues in Section 6. Figure 1 exhibits the organization of this paper.

2 Background and Related Work

2.1 Background

This section introduces the technological context necessary to facilitate the understanding of the context-aware AI modeling challenges in edge environments. Firstly, in Section 2.1.1, we describe the main features of context-aware computing as a specific paradigm within the EC environment. Then, in Section 2.1.2, we present the basics of EC. Section 2.1.3 briefly discusses main concepts of AL, ML and DL. Finally, in Section 2.1.4, we focus on the role of sensors and present examples of applications that use sensors in dynamic environments.

Paper organization

- Section 2 Background and Related Work
  - Section 2.1. Background
    - Section 2.1.1. Context awareness
    - Section 2.1.2. Edge computing
    - Section 2.1.3. AI disciplines
    - Section 2.1.4. Wireless Sensor Network
  - Section 2.2. Related Work
- Section 3 Methodology
  - Section 3.1 Preparation of the data
  - Section 3.2 Search Conducting
  - Section 3.3 Data extraction and analysis
  - Section 3.4 A semantic-aware approach for identifying the survey main subjects
- Section 4 Results
  - Section 4.1 Q1: How much literature activity has there been between 2015 and January 2022?
  - Section 4.2 Q2: What are the challenges in context-aware edge-based AI for sensor networks?
  - Section 4.3 Q3: What are the state-of-the-art solutions used to address the challenges depending on the specific application field?
  - Section 4.4 Q4: What are the motivations to adopt AI solutions to context awareness scenario?
  - Section 4.5 Q5: What are the limitations of current literature or what are gaps existing in the current research about applying AI technologies to context awareness that future researchers can investigate?
- Section 5 Logistics use case: industrial perspectives, challenges and intelligent techniques
- Section 6 Conclusions and open issues

Figure 1: A schematic illustration of the paper organization.

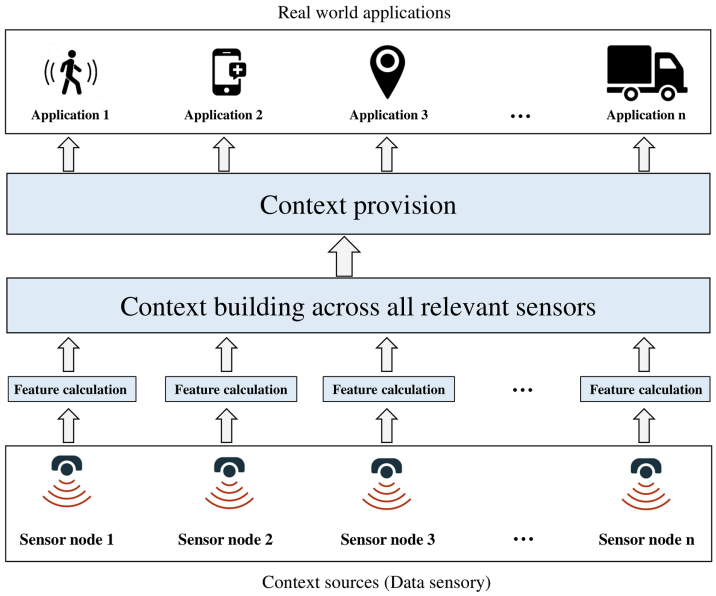


Figure 2: Context-aware framework layers.

2.1.1 Context Awareness

In recent years, context-aware applications have captured a lot of attention as they extract user context, such as location, activity, time, health status, physical environmental state, etc. Various types of special sensors are used. These can be physical sensors, such as the Global Positioning System (GPS) sensor and accelerometer, or virtual sensors, such as user calendar, weather web service and weather radar [18, 19]. However, in a consensus definition, context awareness is defined as "systems that adjust according to conditions: environmental (e.g., the level of pollution), physical (e.g., one's current location), social (e.g., one's family and colleagues), or temporal (e.g., the time of the day), as well as changes in these things over time" [20, 21]. As part of this article, context is defined as a situation and environment of sensors in WSN. Therefore, contextual information use includes interactions between sensor nodes and the reaction of sensor nodes to environmental changes to discover information of interest [21, 22]. A context-aware system architecture is exhibited in Figure 2.

Sensor nodes typically have specific context metrics. Some examples of these metrics are location, energy level, connectivity, speed, temperature, pressure, and link quality.

2.1.2 Edge Computing

EC, a computing paradigm which extends cloud computing, enables all computing outside the cloud to happen at the edge of the network [23], and more specifically

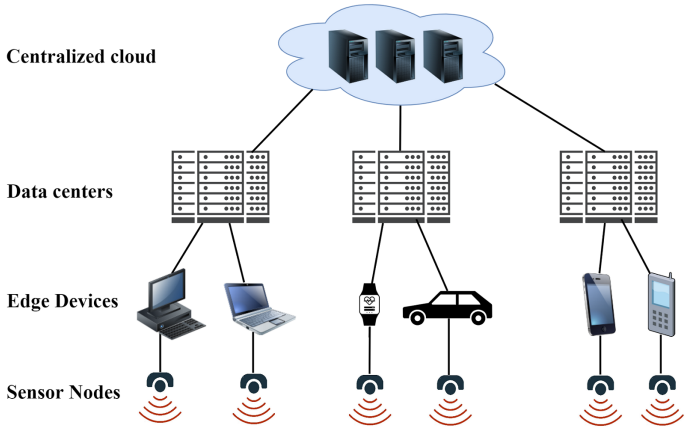


Figure 3: EC ecosystem.

in applications where real-time processing of data is required. Lately, the "Edge" defines the point where sensor nodes and IoT devices are located in the local network [24]. EC works on a huge quantity of data generated by sensor nodes or users in the edge network [25]. However, with respect to context awareness, data generated either from a single sensor node or by multiple sensor nodes represent unprocessed data, while context information represents processed raw data [26]. EC system architecture consists of four primary components shown in Figure 3: centralized cloud, a centralized data processing system, operates on a massive amount of data that can be accessed at anytime. Edge data centers, are specialized data centers located closer to the edge than the cloud that deliver faster processing than edge devices, as well as minimal latency and data transmission costs compared to the centralized cloud in real time. Edge devices are pieces of physical hardware that send data between the local network and the central cloud. Traditional edge devices can include many different things, such as edge sensors, routers, firewalls, and chips. Sensor nodes are data accumulation sources. These technologies include edge sensors and chips which are capable of gathering, sensing, and processing data—to an extent.

Three types of EC architectures have been introduced, namely: Mobile-Edge Computing (MEC), fog computing, and cloudlets computing [24]. MEC extends EC by providing compute and storage resources near to low energy, low resource mobile devices. While fog computing seeks to realize a seamless continuum of computing services from the cloud to the things rather than treating the network edges as isolated computing platforms. Cloudlets are small data centers that are typically one hop away from mobile devices [24]. These paradigms differ in terms of software architecture, context awareness, and location of nodes.

2.1.3 AI Disciplines

The definition of Artificial Intelligence (AI) was first coined by McCarthy in the 1950s, where the field of AI refers to the capability of a machine to imitate human intelligence processes [27]. The overall goal of AI research is to let machines perform some advanced decisions that require intelligent humans to complete. The main concern of AI was and still is to do tasks that are typically hard to characterize formally in terms of mathematical rules [28]. The difficulty of explaining this type of task showed that AI approaches needed the ability to find patterns and gain knowledge [28, 29]. This ability is defined as ML, which allows computer applications to learn and act on data without explicitly programming it [30]. However, mapping the knowledge gained from learning to final prediction requires the implementation of methods classified as representation learning, in which features are converted into representations including useful information [31]. For complex concepts, if a representation is indicated in terms of other representations, DL needs to be used. DL allows computational models to learn representations with different levels of abstraction. Thus, DL can be seen as representation learning that can imitate human thinking and gain knowledge [28]. These days, AI, ML , and DL are three popular terms that are often used interchangeably to characterize intelligent systems. Their relations are shown in [32], in which, DL is part of ML and is also a part of the broad field of AI while ML is considered a part of the AI umbrella.

2.1.4 Wireless Sensor Network

The WSN, which is the backbone of IoT, consists of dedicated, small, resource-constrained, and low-cost sensor nodes that are randomly deployed in a monitoring environment to perform certain specific tasks over a period of time [33]. Current WSN are widely used in various applications, such as healthcare [34], smart homes [35], environment monitoring [36], etc. Each sensor node has a processing power, radio, and electrical storage device that converts a physical phenomenon of a heterogeneous environment into an electrical signal. The main task is to cooperatively sense, gather and process data about devices in the coverage region, and then transfer it to remote servers for deriving the information [37, 38]. Figure 4 displays the typical WSN architecture which contains sensor nodes, fog nodes, and a central cloud.

A number of factors play a role in determining node failures such as harsh environments, restricted energy, and device faults. It is also necessary that the sensor network is able to support the task for a minimum specified period of time [39].

2.2 Related Work

With the aim of outlining the contribution of the present study over the existing related surveys, we provide herein a brief overview of these works. There are many

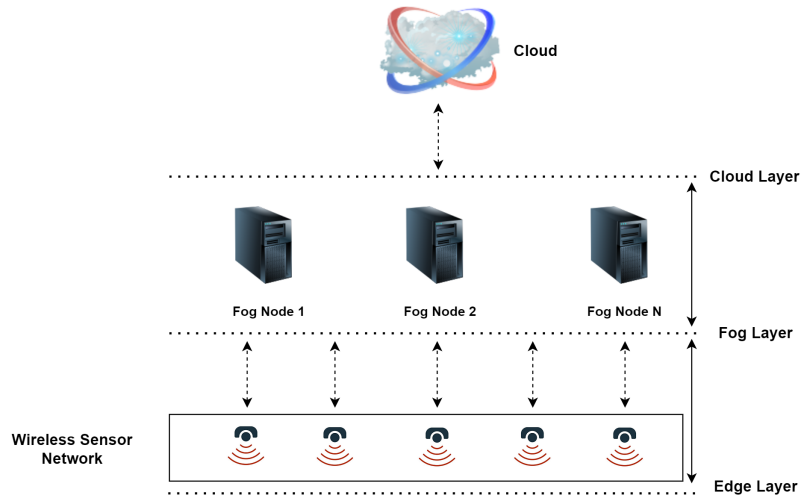


Figure 4: A schematic presentation of the general WSN architecture.

surveys on the subject of context-aware computing, context-aware sensor networks and context-aware intelligence related to the subject of our study. Peraraet et al. [40] provide a framework for an overview of context-aware IoT that briefly describes how ML models work, but does not deepen this point. Furthermore, their proposed solutions are yet to be implemented in real time. In [41] , a literature review focusing on the most common techniques in the development of context-aware systems is presented. However, they show that all methods have disadvantages and do not dive into a discussion of ML methods. Vahdat et al. [42] expose a survey study about specific application domains, namely Mobile Crowd Sensing (MCS) in smart environments. In [43], authors have aimed to understand the state-of-the-art in the development of context-aware middlewares (CAMs) for aiding the construction of HAR applications when using ML. However, they do not consider explicitly the sensor networks and they have focused only on HAR applications. Sezer et al. [44] focus on Data Analytics in Edge-to-Cloud environments. However, they do not deepen the discussion of ML for HAR. Bogale et al. [45] consider the AI approaches in the context of fog (edge) computing architecture, but the authors do not present a deep discussion of the various ML algorithms that are used. In Preeja and Krishnamoorthy's study [46], authors have outlined the context characteristics, context organization, and context-aware systems, such as context modeling and the use of a middleware approach to simplify the development considering the heterogeneity of technologies. However, their survey has been developed in the context-aware middleware domain. The work in [47] brings an overview of significant concepts and related applications in various fields of context-aware systems. Although this work has presented a review of the latest development of context-aware systems during the period from 2008 through 2019, the authors present only a few discussions. Chatterjee et al.'s study [48], on the

other hand, is the most relevant paper to our review. This paper has focused on identifying the current trends, foundations, and components of the envisioned IoT devices to enable the design of more efficient connected intelligent systems in the future. However, the authors do not deepen in a discussion of the type of AI, ML, or DL solutions that have been used to address the predefined challenges or dive deeper into a discussion of application domains. In [49], authors have focused only on the algorithms and modeling techniques used in context-aware recommenders. The purpose of our study is to fill the gap by considering recent advances in the field of context-aware edge-based AI models for sensor networks and by identifying application domain-independent challenges. Moreover, our work differentiates from the above-mentioned studies by applying a semantic-aware approach for identifying the main subjects supported by the survey included papers. Namely, the articles' keywords are analyzed by clustering them into groups of semantically similar terms. Thus, in our survey we have managed to extract the major covered topics in its subject framework. Table 1 lists a comparison of our study with the various related reviews conducted in the period between 2015 and January 2022.

3 Methodology

Given the latest changes that occurred on advances in edge computing with advances in artificial intelligence, the attention of academia/industries is predominantly focused on the state of the art in context-awareness systems, which are considered crucial for the realization of intelligent IoT and sensor network applications. Therefore, it is necessary to identify the state of the literature on the relationship between AI fields and edge computing to support context-aware sensing systems. To achieve this goal, the authors formulate research questions to define the scope of work.

Leaden by research questions listed in Table 2, this research was carried out by defining the below listed search criteria to gather all relevant publications. We divided this review of the literature into four main phases outlined in Figure 5: data preparation, search conducting, data extraction and analysis, identification of survey topics.

3.1 Preparation of the Data

Based on the above research questions (Table 2), this study is focused on the following meanings: "context awareness", "edge computing", "artificial intelligence", "machine learning", "deep learning", "sensors" and "wireless sensor network". In addition to the main concepts, their synonymous were defined. The indexing databases considered for this study were Web of Science and Scopus, which were recommended for conducting a literature review by multiple researches in [50, 51]. In line with the setup of the study, a list of inclusion and exclusion criteria were set to improve the se-

Table 1: Overview of previous related surveys and comparison with our study with respect to their contributions and discussed intelligent techniques (classical AI, ML and DL).

Reference	Year	Main Focus	AI Techniques
[40]	2015	Evaluation of different available resources communication mediums, and, frameworks for industrial market perspective.	×
[41]	2016	A context-aware review for recognizing emerging fields from a software development point of view.	×
[44]	2018	A survey on context awareness for IoT big data analysis.	AI, ML and DL
[45]	2018	A comprehensive survey on the utilization of AI integrating ML, data analytics, and NLP techniques for enhancing the efficiency of wireless networks.	AI, ML and DL
[46]	2019	A literature analysis of various context-aware systems (modelling, organization, and middleware).	×
[47]	2019	A short survey of the latest development of context-aware systems.	AI, ML and DL
[42]	2019	A survey study about context-aware crowd sensing systems for urban environments.	×
[48]	2019	A survey of recent advances in intelligent sensing, computation, communication, and energy management for resource-constrained IoT sensor nodes.	AI, ML and DL
[49]	2021	An extensive survey of AI-based mobile context-aware recommender systems.	AI, ML and DL
[43]	2022	A survey on the use of ML methods in context-aware middlewares for HAR.	AI, ML and DL
Our Paper	2022	A broad study of the adoption of edge-based AI solutions for context-awareness in WSNs.	AI, ML and DL

Table 2: Research questions (RQs).

ID	Question
Q1	How much literature activity has there been between 2015 and January 2022?
Q2	What are the challenges in context-aware edge-based AI for sensor networks?
Q3	What are the state-of-the-art solutions used to address the challenges depending on the specific application field?
Q4	What are the motivations to adopt AI solutions to context awareness scenario?
Q5	What are the limitations of current literature or what are gaps existing in the current research about applying AI technologies to context awareness that future researchers can investigate?

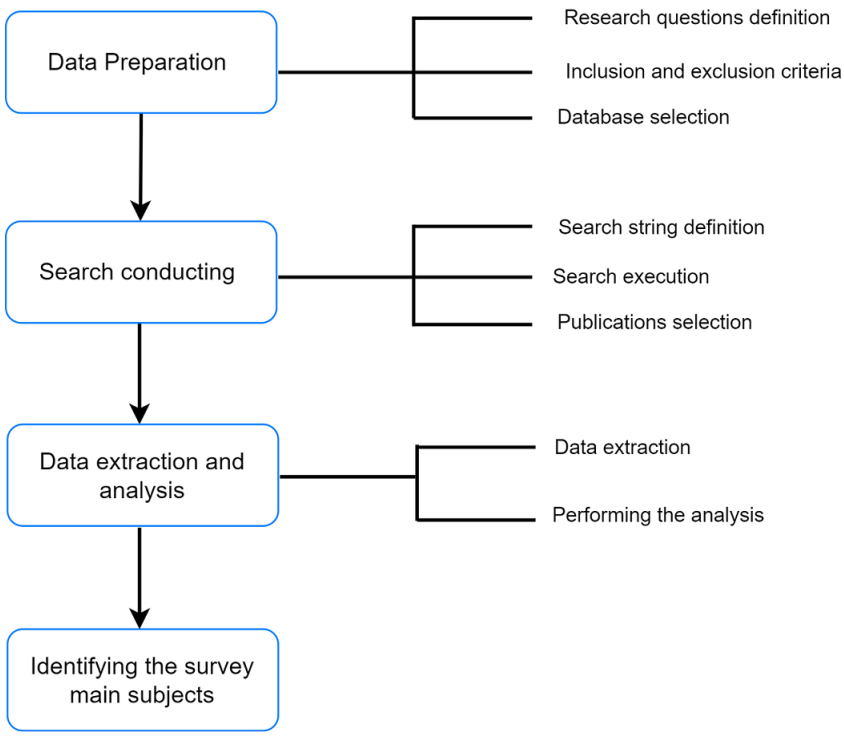


Figure 5: The main methodological phases of the study.

lection of publications and to guarantee a successful analysis process. The inclusion criteria are:

IC1. Journal and conference papers that address the intersection between context-aware, artificial intelligence methods, and sensor network domain, containing the terms in the title, or keywords. Papers with the terms just in the abstract are excluded in this study.

IC2. Papers available in electronic form published between 2015 and January 2022.

IC3. Journal and conference papers written in English.

The defined exclusion criteria are:

EC1. Articles without access to the electronic file.

EC2. Bibliographic, conference reviews, works of non-indexed or gray literature, and master thesis.

EC3. Duplicate studies after reading the title.

EC4. No relevance after reading title and abstract.

Then, the identified studies were sieved according to five defined filters, explained below.

Filter 1 allows the retention of papers related to context-awareness and AI fields such as ML and DL for sensor networks. The search takes the TITLE + ABSTRACT + KEYWORDS fields as a whole, making those 3 fields into just one and then running a text search (IC1).

Filter 2 allows the retention of publications available in electronic form and published between 2015 and January 2022 (IC2). Articles without access to its electronic file are discarded (EC1).

Filter 3 includes only publications journal and conference papers written in English (IC3). It also allows the removal of bibliographic, conference reviews, works of non-indexed or gray literature, and no research thesis (EC2).

Filter 4 allows the removal of duplicate or redundant publications (EC3).

Filter 5 allows the removal of irrelevant papers. The authors of the current survey have conducted this task by reviewing the title and abstract of each paper and selecting only papers that are related to the topic of the survey (EC4).

3.2 Search Conducting

Considering the above-defined research questions, the main focus was papers from the most reputed scientific databases, namely Web of Science and Scopus, during 2015–January 2022.

Table 3: Search strings considering the search process strategy with inclusion and exclusion criteria.

Scientific Database	Search String
Scopus	TITLE-ABS-KEY ((“Context*” OR “aware*”) AND (*edge OR device) AND (“artificial intelligence” OR “machine learning” OR “deep learning”) AND (“sensor*”))
Web of Science	TS = ((“Context*“ OR “aware*”) AND (*edge OR device) AND (“artificial intelligence” OR “machine learning” OR “deep learning”) AND (“sensor*”))

3.3 Data Extraction and Analysis

Initially, a search of scientific papers from the Web of Science and Scopus databases was performed to extract the publications from the selected sources. The selection criteria were divided into the five filters discussed above in order to collect more relevant articles. Therefore, the selection process comprehended the phases depicted in Figure 6 and followed the procedures outlined below:

- A total of 2760 publications related to context-awareness and AI fields were retrieved, of which, 1841 were obtained from Scopus and 919 from Web of Science.
- As a result of the second filter, 515 publications from Scopus and 380 publications from Web of Science were retrieved, available in electronic form and published between 2015 and January 2022 with access to its electronic file.
- Only publications in journals and conference papers written in English were retained. Bibliographic, conference reviews, etc., were excluded. Thus, 435 documents for Scopus and 328 for Web of Science were retrieved as a result of the third filter.
- After merging the publications of Scopus and Web of Science, 763 duplicate publications were removed in the fourth filter, and 490 left.
- After reading the title, abstract, and keywords of these publications, 349 were eliminated because they were not related to the topic of the survey. At the end

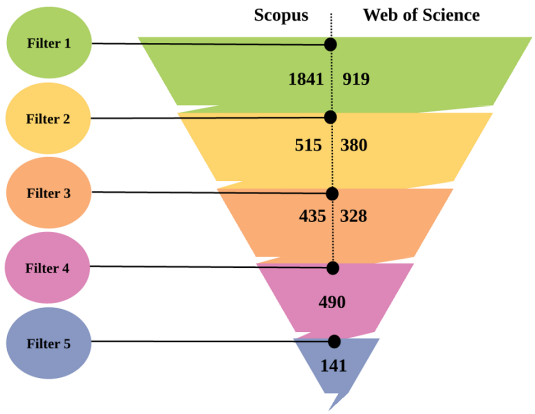


Figure 6: The number of papers selected after applying each filter of the survey's related papers is given for WoS and Scopus databases, respectively.

of the fifth filter, 141 papers were left. These are included and examined in this work.

After we managed to extract and classify the data, the aggregated data were then analyzed to be used to respond to the research questions in Section 4.

3.4 A Semantic-Aware Approach for Identifying the Survey Main Subjects

We have applied a semantic-aware approach for identifying the main subjects supported by the survey-related articles. The approach is built upon the analysis of the articles' keywords. Initially, all different (unique) keywords of the extracted articles are gathered together. The number of all unique keywords is equal to 637. Then, this number is reduced to the most frequent keywords. Namely, a score is assigned to each keyword reflecting its frequency of appearing among the articles' keywords. Then, all keywords which scores are below the preliminary defined threshold (2 in our consideration) are filtered out, i.e., only the most frequent 82 keywords are left. The applied approach uses the semantic similarity between keywords to identify the main research/application subjects covered by the survey. It is based on the idea published in [52]. In order to be able to apply this approach we have manually associated each keyword (from the most frequent ones) with its synonym keywords. This assists us in calculating the semantic similarity between keywords based on the common synonymy between two keywords by using the Jaccard coefficient [53]. Thus, the semantic similarity between two keywords w_i and w_j can be computed as follows:

$$SemSim(w_i, w_j) = \frac{n_i + n_j - n_{ij}}{n_{ij}}, \quad (1)$$

where n_i and n_j are the synonymy numbers of w_i and w_j, respectively, and n_{ij} is the synonymy common number between w_i and w_j. The keywords then can be partitioned into groups of semantically similar keywords by applying a selected clustering algorithm (e.g., DBSCAN). The obtained clusters of keywords represent the main research/application subjects supported by the selected articles. The flowchart in Figure 5.4 illustrates the process of identifying main survey subjects. We have identified 11 of these subjects presented in Figure 7. The 11 keyword groups and the title of their subjects are shown in Table 4. In addition, the relative percentage of cluster size produced by applying DBSCAN with eps = 0.3 on the 82 most frequent keywords are also elaborated in Figure 7. The parameter eps specifies how close data points (keywords) should be to each other to be considered a part of a cluster. We have experimented with different values of eps and 0.3 has produced the most balanced grouping without any outliers.

The selected articles can be further analyzed with respect to identified subjects to obtain deeper insight into the limitations and gaps in the current research related to the survey theme (Q5). For example, each article can be represented by a vector of membership degrees of the article to the different subjects (clusters of keywords). The membership degree of article i to subject S_j can be calculated as k_{ij}/n_i, where k_{ij} is the number of keywords from the keyword list of article i that belong to cluster S_j and n_i is the total number of keywords describing article i. In this way, a fuzzy distribution of the articles among the identified subjects is obtained.

The fuzzy grouping of the articles can easily be transferred into a non-fuzzy clustering by associating each article to only subject(s), for which it has the highest membership degree. This allows us to evaluate the popularity of each subject quantified by taking into account the number of articles belonging to it (see Figure 8). Each group (research/application subject) can be associated with specific AI/ML techniques and domains of application by further analysis of the challenges and application domains addressed by the articles assigned to it. The knowledge extracted due to this analysis can be used to answer Q2, Q3 and Q5 by facilitating the identification of under/over-represented topics in the current research along with the challenges shared among different application fields. Each cluster of articles can also be studied with respect to the state-of-the-art solutions used to address the issues in the research/application subject presented by this cluster (Q3). The articles can also be grouped with respect to the identified subjects by using their membership degree vectors to measure the similarity between each pair of articles. In comparison with the grouping produced by the first approach where each group of articles is related to one concrete subject, in the current clustering the articles that are grouped together will be similar with respect to more than one research/application subject, e.g., we can identify articles that use the similar AI/ML techniques and at the same time deal with issues in the same application fields. As a result of this grouping the studied articles have been distributed in 15 disjoint clusters. We have experimented with different values for the parameter eps. However, all of those have produced clustering solutions where

some of the articles are considered to be outliers. This is due to the scatter of articles in terms of topics, i.e., most of the articles are related to no more than two topics. The value 0.4 for the parameter eps is chosen, since it has produced the less number of outlying articles.

The identified eleven clusters of keywords along with the titles of the subjects (cluster labels) they represent.

Table 4: The identified eleven clusters of keywords along with the titles of the subjects (cluster labels) they represent.

Cluster Label	Size	Keywords
AI, ML and DL	27	active learning, AI, ANN, attention mechanism, big data, classification, CNN, data mining, data models, DL, DNN, feature extraction, feature selection, inference, intelligent systems, LSTM, ML, prediction, predictive models, RF, RNN, regression, RL, supervised learning, SVM, time-series classification, training.
Edge Computing and Smart Monitoring	11	EC, pervasive computing, biomedical monitoring, ECG, electrocardiography, health monitoring, heart rate, monitoring, pervasive healthcare, physiological signals, physiology.
Smart Healthcare	10	accelerometer, action recognition, activity recognition, gait recognition, HAR, mhealth, mobile computing, mobile health, mobile sensing, smart healthcare.
Smart and Wearable Devices	8	on-device computation smart devices, smartphone, wearable computing, wearable devices, wearable sensors, wearable system, wearables.
Anticipatory Computing and SSL	4	anticipatory computing, recommendation system, semi-supervised learning, transfer learning.
Context-Awareness	4	context modeling, context-aware systems, context-awareness, context-awareness services.
Energy Consumption and Saving	4	energy consumption, energy efficiency, energy saving, power consumption.

Continued on next page

Table 4 – *Continued from previous page*

IoT	4	industry 4.0, IoMT, IoT, smart home.
Sensors and WSN	4	WSN, sensor data, sensor fusion, sensors.
Mental Health	3	mental health, stress, stress monitoring.
Computer Vision	3	computer vision, object recognition, pattern recognition.

Note. The clustering is produced by applying DBSCAN clustering with eps= 0.3.

As one can see in Table 5, the top ten keywords that appear the most in the articles included in this review are "ML", "DL", "IoT", "activity recognition", "sensors", "HAR", "wearable sensors", "CNN", "classification" and "context-awareness", i.e., the review perimeter is well outlined by those.

Figure 8 presents the percentage of papers of sample studied per the eleven major subjects identified. The references to the papers related to each subject are given in Table 6. In addition, as it was discussed before the popularity of each identified subject is assessed relatively with respect to the others by taking into account the frequency of the keywords assigned to its cluster. This is represented by a pie chart in Figure 7. Interestingly, the four most popular subjects (see Figure 7) identified based on the keywords' frequency coincide with the four subjects supported by the quantity of the published papers (see Figure 8). These subjects are AI, ML and DL, Smart Healthcare, Smart and Wearable Devices and Edge Computing and Smart. However, AI, ML and DL has a much higher percentage than Smart Healthcare with respect to the keywords' frequency while these subjects are equally represented with respect to the published papers.

Table 5: Top ten most frequently used keywords.

Keyword	**Occurrences**
ML	55
DL	27
IoT	22
activity recognition	17
sensors	12
HAR	10
wearable sensors	10
CNN	8
classification	7
context-awareness	7

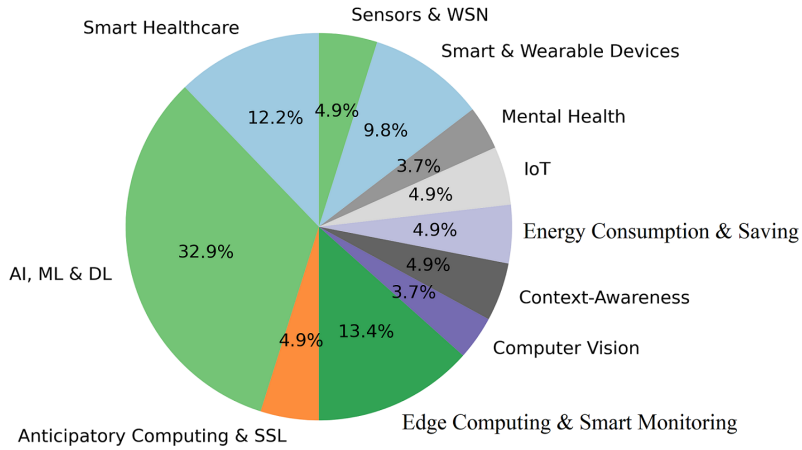


Figure 7: Relative popularity of the identified subjects assessed on the based of the keywords' frequency. The most popular subject is AI, ML and DL followed by Edge Computing and Smart Monitoring, Smart Healthcare and Smart and Wearable Devices.

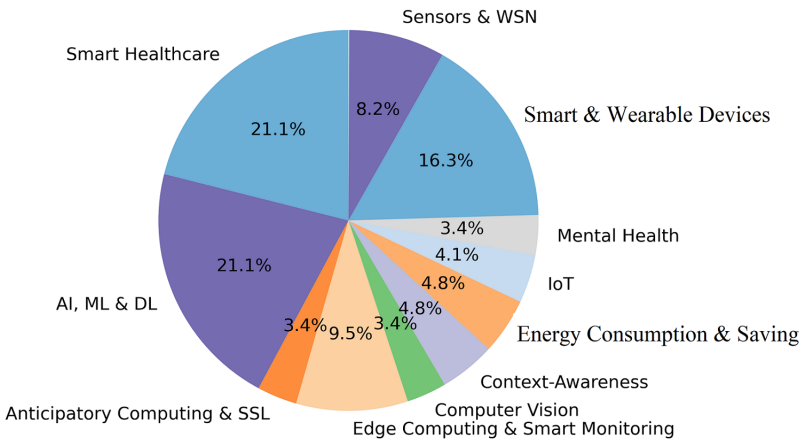


Figure 8: Percentage of papers of sample studied per the main identified subjects. The most represented subjects are AI, ML and DL and Smart Healthcare followed by Smart and Wearable Devices, Edge Computing and Smart Monitoring and Sensors and WSN. These well reflect the survey theme.

This may be due to the fact that in the case of Figure 8, some papers are cross-disciplinary, i.e., they have the same highest membership degree to more than one subject and in that way, they are counted for all those subjects. One can also notice that Context-Awareness and Energy Consumption and Saving have the same representativeness in both Figures 7 and 8. In addition, the two least popular subjects (Mental Health and Computer Vision) are identical in both figures.

As it was mentioned above, Table 6 exhibits the paper references belonging to each subject and how many belong to these primary subjects. Smart Healthcare and

AI, ML and DL stand out, (both with 34 papers), followed by the Smart and Wearable Devices with 30 documents. In the third position, it can be found the Edge Computing and Smart Monitoring with 16 papers. The number of included papers for Computer Vision and Mental Health is equal, i.e., only 5 papers are assigned to each one. It is worth noting that most of these studies, as can be seen in the table, belong to more than one subject, since their keywords are distributed among various clusters of keywords (main subjects).

Table 6: The main subjects along with the references to their related papers.

Main Subject	References	# of Studies
Smart Healthcare	[54–87]	34
AI, ML and DL	[54, 57, 72, 76, 77, 81, 87–113]	34
Smart and Wearable Devices	[54, 56, 62, 68, 73, 74, 76, 77, 79, 82, 85–87, 89, 100, 104, 107, 114–126]	30
Sensors and WSN	[68, 72, 73, 75, 79, 82, 96, 101, 107, 110, 112, 113, 120, 126–129]	17
Edge Computing and Smart Monitoring	[56, 73, 76, 84, 89, 94, 104, 113, 122, 130–136]	16
Context-Awareness	[58, 61, 63, 64, 72, 80, 95, 110, 130, 137–140]	13
Energy Consumption and Saving	[55, 57, 109, 129, 132, 141–143]	8
Anticipatory Computing and SSL	[55, 58, 64, 72, 82, 125, 144]	7
IoT	[75, 103, 145–148]	6
Computer Vision	[86, 90, 149–151]	5
Mental Health	[68, 79, 94, 122, 131]	5

4 Result Analysis

We have analyzed the data extracted by the selected publications (see Section 3.3) to answer each research question presented in Table 2. The research questions are addressed one by one in the following subsections.

4.1 Q1: How Much Literature Activity Has There Been between 2015 and January 2022?

We have reviewed the significant research papers in the field published from 2015 to January 2022. Figure 9 presents the details of the year-wise publications (publishing

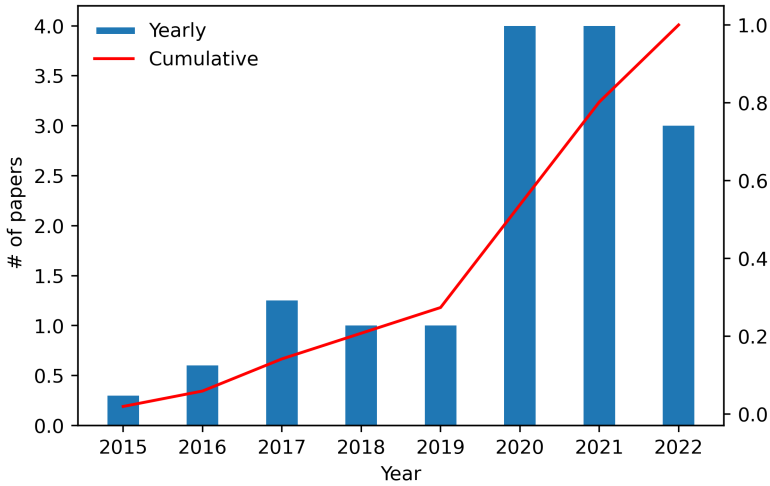


Figure 9: Included papers per year (publishing trend) normalized on monthly base. There was a significant increase in the number of included papers published after 2019.

trend). A clear increasing interest in the recent years can be seen from that figure. For each year, we show the total number of papers normalized on monthly base. The highest number of papers published per year are after 2020. This demonstrates not only highly increased interest, but also the high need of research in intelligent context-aware WSNs.

Moreover, the included papers per year are analyzed and distributed in four groups based on the used computational techniques, i.e., ML, DL, ML and DL and AI. These are presented in Figure 10, showing a significant increase in the use of DL in the studies published after 2019 with the expectation that this will continue to flourish in the following years. In addition, one can notice after 2019 the appearance of studies using AI modelling and reasoning techniques such as fuzzy logic.

4.2 Q2: What Are the Challenges in Context-Aware Edge-Based AI for Sensor Networks?

In order to answer the question Q2, the main challenges have been identified and are shown in Figure 11. These are Human Activity Recognition (HAR), monitoring, Quality of Service (QoS), energy saving, activity recognition, object detection and location-based service (LBS). They have been addressed by various AI, ML and DL approaches under different application domains as this will be discussed in the answers of the next research question. According to Figure 11, HAR is the top-addressed challenge, namely in 28.5% of the sample. HAR includes recognizing daily performed locomotion modes [64, 74, 76, 152–154], analyzing the behavior of the elderly in daily life [60, 130, 133], gait analysis [54, 155–157], etc. Monitor-

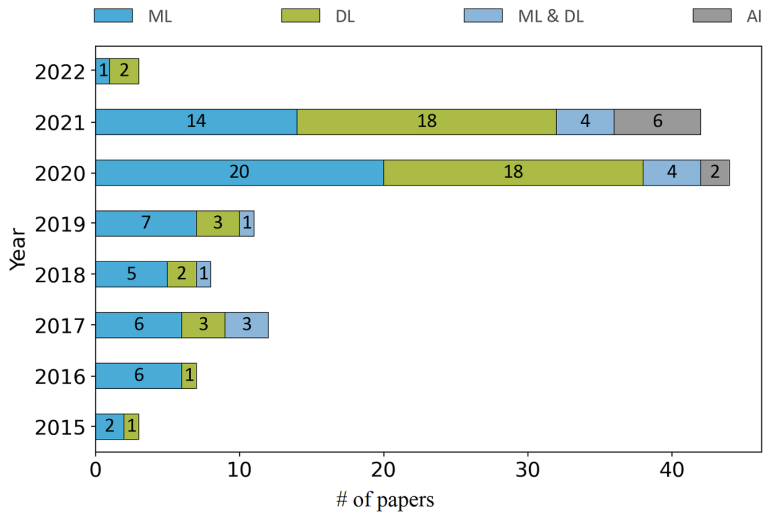


Figure 10: Included papers per year are distributed in four categories based on the used computational techniques, i.e., ML, DL, ML and DL and AI.

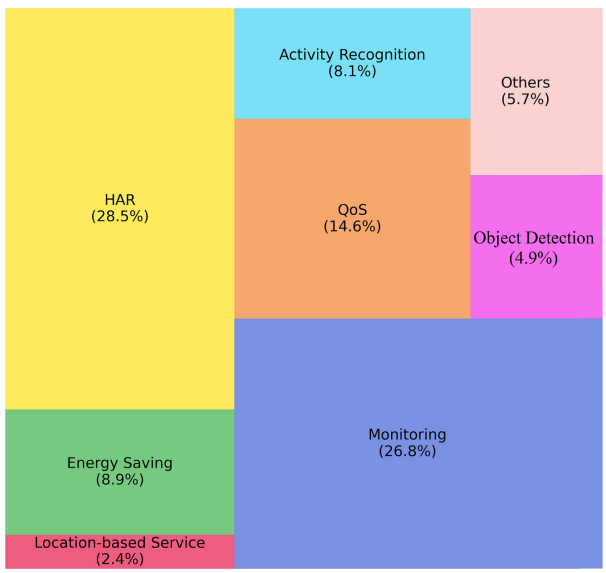


Figure 11: Main challenges addressed by the papers included in the survey.

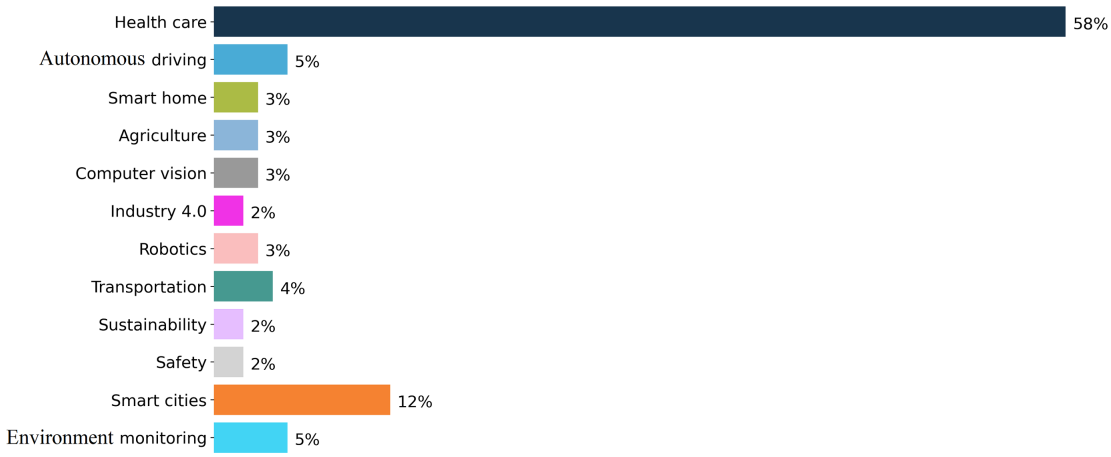


Figure 12: Percentage of papers of sample studies per domain of applications. The most popular category is healthcare followed by smart cities, autonomous driving, environment monitoring and transportation (logistics).

ing is the second most studied issue, namely in 26.8% of the included papers. Not surprisingly, in the context of monitoring, various applications have been identified, e.g., health [73, 100, 120, 158, 159], smart buildings [102, 121, 160], agriculture [161, 162], stress [68, 122, 131, 163], transportation [110], military defense [164], etc. Other challenges comparatively highly studied in the included papers are QoS [95, 109, 116, 128, 132, 138, 140, 149, 165–174] with 14.6%, and energy saving [55, 57, 88, 92, 94, 98, 125, 141–143, 175] with 8.9%.

4.3 Q3: What Are the State-of-the-Art Solutions Used to Address the Challenges Depending on the Specific Application Field?

As it was already discussed in the answer of Q2, the two most studied challenges are HAR (28.5%) and monitoring (26.8%), followed by QoS (14.6%), energy saving (8.9%) and activity recognition (8.1%). In addition, as one can notice in Figure 11, LBS is investigated only in 2.4% of the sample. In order to answer Q3, we have initially explored the relationships of these challenges with the application fields addressed in the included papers. The studied application domains are presented in Figure 12.

The top explored category is healthcare studied in 58% of the included papers. It is followed by smart cities (12%), autonomous driving (5%), environment monitoring (5%) and transportation/logistics (4%). All the other categories are below 3%. It is interesting to study the relationships among the five most studied application domains, the top addressed challenges and used intelligent techniques. In that way, two types of connections will be revealed: one between the state-of-the-art solutions

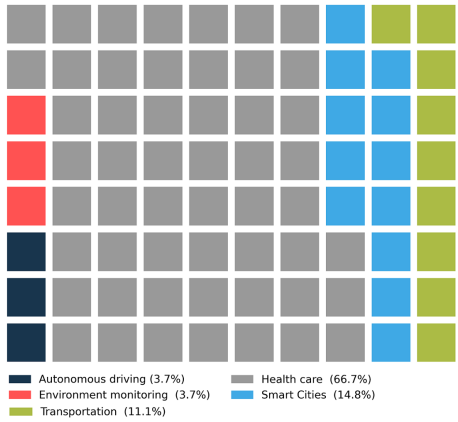


Figure 13: The relationship between HAR and top five most studied application domains.

used to address the identified main challenges and the other between the addressed challenges and corresponding application fields used to evaluate the proposed intelligent solutions. This will outline the technological and application perimeter of the context-aware intelligent systems for sensor networks. In addition, Section 5 discusses the identified challenges along with the intelligent techniques used in the logistic use case that has inspired this study.

Figure 13 illustrates the relationship between the HAR challenge and top five application domains. As can be seen in the figures, 66.7% of health care studies have addressed the HAR challenge. Smart cities and transportation are the second and third, with 14.8% and 11.1% of the papers studying this challenge, respectively. HAR is logically less explored in environmental monitoring and autonomous driving applications.

Furthermore, Figure 14 depicts the relationship between QoS with the top five application domains. Healthcare is again the most studied application domain (38.5%), followed by smart cities, transportation, autonomous driving and environmental monitoring sharing the equal interest (15.4%) in the included papers.

In addition, we study the relationships of the identified challenges against AI techniques applied. Figure 15 illustrates the relationship between the HAR challenge and intelligent techniques used to address it in the included papers. According to this figure, 41.4% of the papers studying HAR challenge use traditional ML approaches for handling it, in 44.8% of the papers DL techniques are applied, while in 10.3% of the studies ML and DL approaches are applied together to address this challenge, and only in 3.4% of the papers addressing HAR, AI methods have been employed.

Furthermore, Figure 16 depicts the relationship between the QoS challenge and intelligent methods used to address it. We can observe that in 61.5% of the studies ML techniques have been applied to solve this challenge, in 30.8% of the papers DL methods have been preferred, and AI techniques are only 7.7% of the included papers

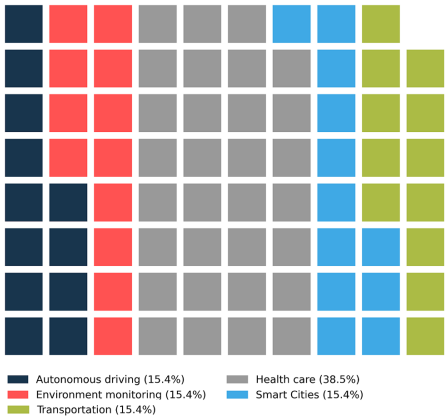


Figure 14: The relationship between QoS and top five most studied application domains.

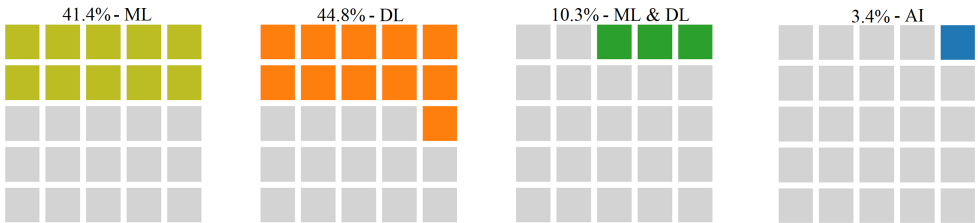


Figure 15: The relationship between HAR challenge and AI techniques categories used to address it.

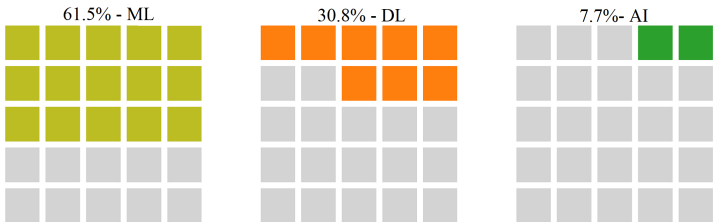


Figure 16: The relationship between QoS challenge and corresponding AI techniques categories used to deal with it.

devoted to QoS.

Figure 17 visualizes, respectively, the relationships between the energy saving challenge and intelligent approaches used to address it in the studied papers. It is worth mentioning that the same trend is observed for the activity recognition challenge.

Figure 18 presents the more frequently used ML/DL algorithms in addressing HAR challenge. We can observe that Decision Tree (DT) (ML) and Convolutional Neural Networks (CNN) (DL) are equally used to address this challenge, namely in 17% of the papers studied the challenge. In addition, 17% of the studies have applied various other approaches such as Linear Regression (LR), active learning, fuzzy logic,

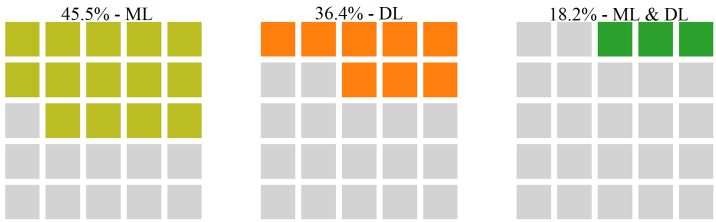


Figure 17: The relationship between Energy Saving challenge and AI techniques categories applied to address it.

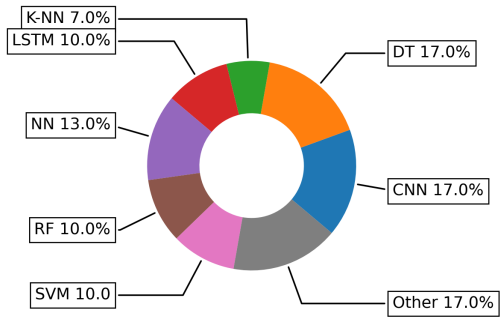


Figure 18: Specific ML and DL algorithms more frequently used in addressing the HAR challenge.

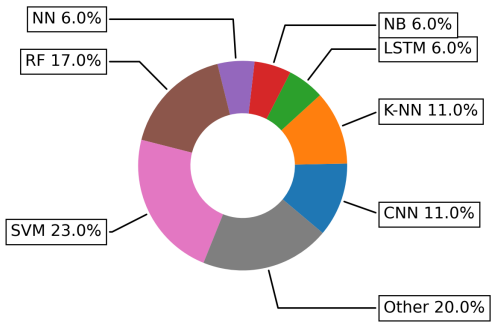


Figure 19: Specific ML and DL algorithms more frequently used in addressing the Monitoring challenge.

etc. Neural Networks (NN) are mentioned only in 13% of approaches, while Random Forest (RF) and Support Vector Machine (SVM) have been implemented in 10% of proposed methods addressing HAR challenge. Figure 19 depicting specific ML/DL techniques used to address Monitoring challenge shows that in contrast to HAR challenge, a lion's share (23%) of techniques used is for SVM, followed by 17% for RF and then K-Nearest Neighbor (K-NN) and CNN taking the equal percentage (11%).

Figure 20 depicts the trend donut chart of the ML and DL approaches that have been applied in addressing the Activity Recognition challenge. The results show that two DL techniques (CNN and Recurrent Neural Network (RNN)) and traditional LR

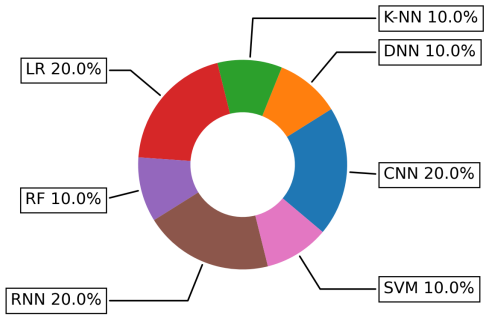


Figure 20: Specific ML and DL algorithms more frequently used in addressing Activity Recognition challenge.

are sharing the same percentage of usage, namely 20% of each one, while Deep Neural Network (DNN), K-NN, RF and SVM have also shared the equal usage percentage, but twice lower (10%). It is worth mentioning that 25% of methods used to address the QoS challenge apply DT while techniques such as CNN, RF, Naive Bayes (NB), K-means and Q-learning have been used only by 12% of the approaches. The analysis of the Energy Saving challenge reveals that 38% of approaches utilized are based on Long Short-Term Memory (LSTM), while each of the techniques K-means, RF, CNN, RNN and DNN has been used in 12% of the studies devoted to this challenge.

With regard to the state-of-the-art solutions used to solve the identified main challenges, we initially analyzed the sample of selected studies from the view of AI. Studies using fuzzy logic techniques are 37.5% of articles, while studies using various other approaches to ML/DL approaches have the attention of 62.5% of articles. In the discussed papers, see Figure 21, the review of the sample studied through the ML lens shows that the most used ML techniques are SVM, RF, DT and K-NN. For example, SVM and RF approaches are used by 17.2% and 14.9% of the selected articles, respectively. DT and K-NN are identified in 12.6% and 10.3% of the studies, respectively. While the clustering techniques are applied only by 5.7% of the selected papers. Regarding other ML techniques used, 28.7% are applied by sample studies. From the sample studied of DL discipline, illustrated in Figure 22, papers considering CNN are the most numerous, representing 29.3%. NN are discussed in 24.1% of the studies, while LSTM is used by 19.0% of the sample papers. The use of Reinforcement Learning (RL) and DNN is found in 3.4% of the articles. Figure 23 illustrates the percentage of usage of the ML and DL approaches in the selected studies. The diagram shows that 52% of selected papers have used ML techniques such as SVM, RF, K-NN, clustering, etc., while 39% addressed the edge-based AI challenges by DL approaches such as CNN, NN, LSTM etc. In addition, we have found that 10% of the papers use ML and DL together to address different challenges in context-aware scenarios. Table 7 presents an overview of selected studies that used ML and

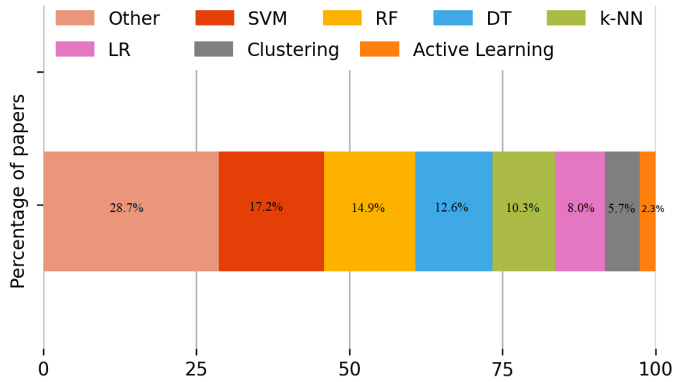


Figure 21: Percentage of papers per ML category of algorithms found in the sample studied. The most used ML techniques are SVM, RF, DT and K-NN.

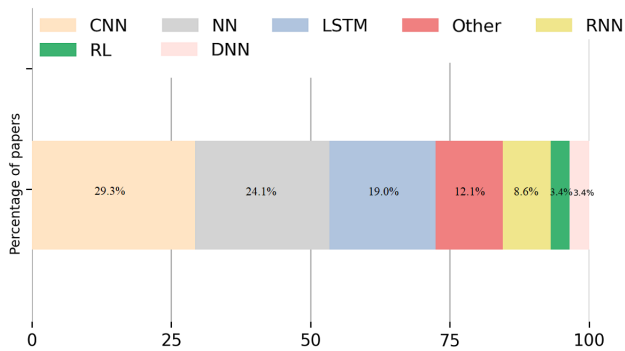


Figure 22: Percentage of papers per DL category of algorithms found in the sample studied. The three most applied DL techniques are CNN, NN and LSTM.

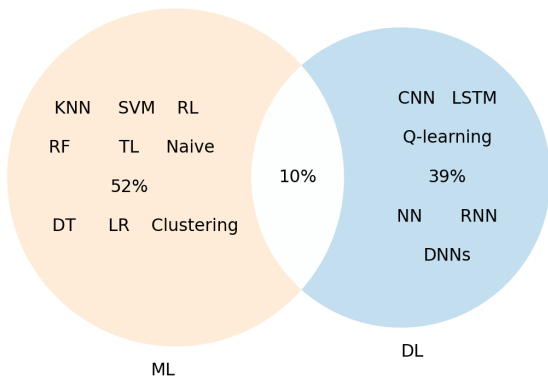


Figure 23: Overview of ML and/or DL techniques that have been used in the included papers.

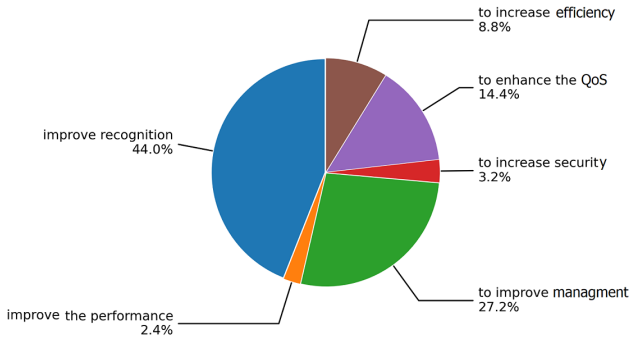


Figure 24: Motivations of adopting AI solutions to context awareness.

DL techniques to address challenges in context-aware scenarios and highlights the techniques used in each of them.

Table 7: Various ML and DL techniques used in context-aware scenarios for sensor networks.

Reference	ML	DL
[115]	DT, Discriminant Analysis, SVM, K-NN, NB	NN
[176]	Gaussian mixture models	DNN, RNN
[71]	SVM	NN
[131]	SVM, J48, RF, NB	NN
[63]	RF, DT	NN
[73]	SVR, RF, GP, LR, K-NN	ANN
[125]	semi-supervised k-means	DNN
[62]	LR	RNN
[117]	RF, SVM, K-NN, SGD, LR, NB, ET	DF
[93]	SVM	NN
[177]	SVM	MLP, LSTM, CNN

4.4 Q4: What Are the Motivations to Adopt AI Solutions to Context Awareness Scenario?

The motivations for applying AI/ML approaches to context-aware scenarios identified in the included studies are shown in Figure 24.

The results show that 44% of the studies have a motivation to improve the recognition. Some of these are proposed to recognize human activity with the wearable devices [54, 66, 81, 84, 152, 153] or to ease the finding of objects [86, 127, 151], or to

recognize the emotion [89, 113]. Another motivation is related to management which refers to the configuration, maintenance, and monitoring (27.2%), such as monitoring the elderly [130, 133] or detection of health-related problems [59, 60, 79, 120, 158], detection of abnormal driving behavior [71], etc. QoS is also an important motivation with 14.4%, as optimization of resource-constrained IoT devices [132, 169, 173], or forecast the connectivity and bandwidth of mobile devices [140]. In addition, data privacy and security are another significant motivation, as to address the data privacy concern in healthcare applications [109, 166], or in autonomous driving [165], or to improve access control techniques of smart devices [116]. Improving the performance of location-aware applications is the motivation used in 2.4% of the selected studies, e.g., in location tracking applications [96, 178] or in autonomous driving technology [107].

4.5 Q5: What Are the Limitations of Current Literature or What Are Gaps Existing in the Current Research about Applying AI Technologies to Context Awareness That Future Researchers Can Investigate?

The papers included in the survey are analyzed from three main perspectives: used state-of-the-art (AI, ML, and DL) techniques, application domains, and addressed challenges. The identified limitations and gaps in the study are discussed in light of these three perspectives.

As a result of the analysis conducted in Section 4.3, we have identified the lack of unsupervised and semi-supervised approaches that allow for dealing with the cases of not enough or entirely missing labeled data as well as transfer learning techniques in the reviewed state-of-the-art solutions. These can be considered as a particular gap calling for future research and development of techniques dealing with those challenges typical for most context-aware real-world scenarios. In addition, the current state-of-the-art research in the context-aware intelligent systems is lacking solutions in the framework of collaborative learning where several smart devices share insights from the local training, without sharing the raw data, namely decentralized and distributed learning schemes such as Federated Learning [179] and Swarm Learning [180].

The second perspective that has been studied in the included papers reviews the application domains used to evaluate the proposed intelligent context-aware solutions. Figure 12 exhibits that healthcare domain is studied in more than half of the papers (58%). The percentages of the other identified application domains are very low in comparison to that of healthcare, see the discussion presented in Section 4.3. For example, logistics/transportation domain which is in the focus of our special interest (see Section 5) is studied in only 4% of the reviewed papers. More than the half of the application domains (e.g., smart homes, agriculture, computer vision, In-

dustry 4.0, robotics, sustainability and safety) are even below this percentage, only smart cities show a higher representation, namely the domain is mentioned in 12% of the papers.

Finally, from the perspective of the challenges identified (see Section 4.2), we can observe in Figure 11 that location-based services are understudied, mentioned in only 2.4% of the reviewed articles. The interest in studying energy saving, activity recognition and object detection is also not very high, below 10%, which is quite lower in comparison with HAR and monitoring, each one explored in more than 25% of the studies included.

5 Logistics Use Case: Industrial Perspectives, Challenges and Intelligent Techniques

The current survey is inspired by an industrial use case in smart logistics. This and the fact that this application domain is less studied in comparison to the other top four fields motivated us to provide with an additional discussion of the industrial perspectives in logistics, the challenges identified along with the corresponding intelligent solutions used to addressed them.

Logistics is the backbone of global trade with global logistics expenditures making up between 10 to 15 percent of the total world GDP [181]. It is a high volume and low-margin market with many actors, turning supply chains into very complex operations with numerous logistics partners involved in each shipment. Because of this complexity, visibility into where goods are at a given moment, if they have been handled correctly, and if they are going to be delivered on time is a difficult goal to achieve. Most transportations lack visibility and traceability of what happens during the journey, making it difficult to answer when goods will arrive.

With the emergence of IoT trackers, tracking of individual goods has become possible by attaching a tracker on goods instead of manually tracking supply chain segments, such as individual lorries or containers. The decreasing footprint and price point of IoT devices have additionally enabled the ubiquitous deployment to not only high value goods and entire pallets, but to also individual items.

Due to the mobile nature of trackers, they are essentially battery-operated and function in environments where charging is often limited or nonexistent. Trackers often undergo shipment in demanding environments, such as containers and warehouses, in which wireless communication is unattainable and incur a high energy cost. In addition, trackers are often equipped with other sensors, such as temperature and accelerometer, each with their own energy profile. Altogether, trackers need to operate during the entire length of a transportation, while sensing and reporting significant events along a route and maintaining a sufficiently low energy profile.

With the ubiquity of small form factor AI computing and individual goods level

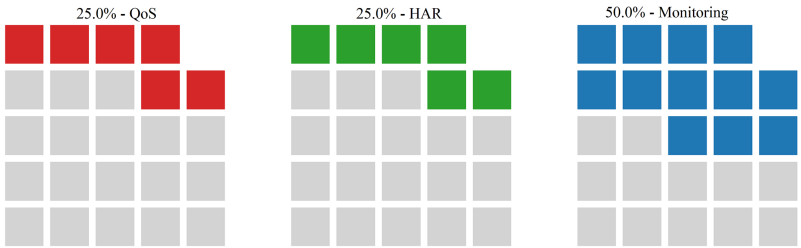


Figure 25: Main challenges in logistics addressed by the papers included in the survey.

tracking, the possibility for trackers to sense their environment and adapt behavior accordingly, both individually and collaboratively together with other devices, has become attainable. As an example, trackers can uncover relations between the device and its operating environment in order to adjust their sensing and operating profiles. For example, detecting indoor/outdoor and providing this context-aware information in various environments may be helpful and lead to battery-saving solutions [182]. In addition, multiple trackers can work in unison to make distributed decisions and utilize sensor sharing for improved power efficiency [183].

The study published in [184] identifies research in the context of intelligent transport logistics as performance enhancing approaches that combine multiple modalities of information technology and sensing into a real-time transportation management system using AI and ML. A central challenge posed by the authors stemming from the black-box nature of AI systems, lies in identifying and finding solutions for mitigating the effects of biased decisions taken by artificial intelligence. Given the advent of continuously learning AI systems wherein decision-making is updated given new input, we believe that this challenge will receive an increasing importance and focus. It is also important to note that the main challenges addressed in the studied papers in logistics domain are HAR, QoS, and monitoring as it is shown in Figure 25. The monitoring challenge is in the focus of 50% of the sampled papers, while each of HAR and QoS is studied in 25% of the logistics devoted papers.

In addition, AI categories identified from the included studies in logistics are shown in Figure 26. Interestingly half of the papers in logistics have used DL techniques, namely CNN, while one-quarter of the papers included have utilized ML and AI equally. This once more confirms the finding identified in [184] concerning the black-box nature of the most existing intelligent solutions in logistics and the need for new more transparent approaches.

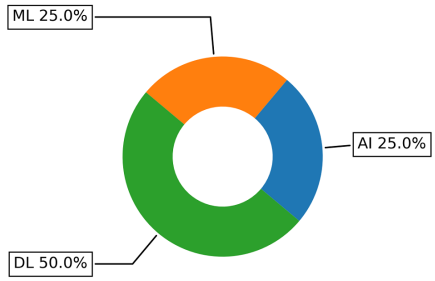


Figure 26: Main AI techniques in logistics addressed by the papers included in the survey.

6 Conclusions And Open Issues

In this paper, we have provided an extensive survey of context-aware edge-based AI methods for WSN technology. Five research questions have been addressed by analyzing 141 research articles used as primary papers. Initially, we have applied a semantic-aware approach for analyzing the keywords of the included papers in order to extract the survey main subjects. Eleven such topics have been identified, e.g., the most popular are AI, ML and DL, Edge Computing and Smart Monitoring, Smart Healthcare and Smart and Wearable Devices. In the analysis carried out, we have also discovered that healthcare, smart cities, autonomous driving, environmental monitoring, and transportation are the top five application domains. Improving the quality of recognition, efficient management, enhancing QoS and efficiency, and ensuring higher security are the top five motivations for enabling intelligent applications in context-aware systems.

Various AI-based solutions have been studied in the included papers. Unsupervised and semi-supervised algorithms, as well as transfer learning techniques are identified as ones that have not grabbed much attention of researchers in most context-aware scenarios. Moreover, other promising collaborative frameworks such as federated learning and swarm learning have not been adequately explored. There is also a lack of research covering the location-based services in the reviewed articles, further studies with more focus on these challenges is highly suggested.

Our future research plans include deeper investigation of the challenges and gaps identified due to the conducted survey in order to expand further the knowledge gained and use those to develop new efficient intelligent edge-based solutions. For example, we are particularly interested in developing decentralized resource-efficient unsupervised or semi-supervised learning frameworks. Our short run goal is the implementation of such a federated framework and its initial study and evaluation in a logistics use case.

References

[1] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. M. Barnaghi, and A. Sheth. “Machine learning for Internet of Things data analysis: A survey”. In: *ArXiv* abs/1802.06305 (2017).

[2] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. “Middleware for Internet of Things: A Survey”. In: *IEEE Internet of Things Journal* 3 (2016), pp. 70–95.

[3] A. V. Dastjerdi and R. Buyya. “Fog Computing: Helping the Internet of Things Realize Its Potential”. In: *Computer* 49 (2016), pp. 112–116.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. S. Palaniswami. “Internet of Things (IoT): A vision, architectural elements, and future directions”. In: *Future Gener. Comput. Syst.* 29 (2012), pp. 1645–1660.

[5] F. Sakr, F. Bellotti, R. Berta, and A. D. Gloria. “Machine Learning on Mainstream Microcontrollers †”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[6] M. Merenda, C. Porcaro, and D. Iero. “Edge Machine Learning for AI-Enabled IoT Devices: A Review”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[7] Y. Li and S. Wang. “An Energy-Aware Edge Server Placement Algorithm in Mobile Edge Computing”. In: *2018 IEEE International Conference on Edge Computing (EDGE)* (2018), pp. 66–73.

[8] Z. Wu, L. Su, and Q. Huang. “Stacked Cross Refinement Network for Edge-Aware Salient Object Detection”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 7263–7272.

[9] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. P. C. Valentin, and S. Izadi. “StereoNet: Guided Hierarchical Refinement for Real-Time Edge-Aware Depth Prediction”. In: *ArXiv* abs/1807.08865 (2018).

[10] G. Yang, Q. Zhang, and G. Zhang. “EANet: Edge-Aware Network for the Extraction of Buildings from Aerial Images”. In: *Remote. Sens.* 12 (2020), p. 2161.

[11] J. Kim, D. S. Han, and B. Senouci. “Radar and Vision Sensor Fusion for Object Detection in Autonomous Vehicle Surroundings”. In: *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)* (2018), pp. 76–78.

[12] S. Buzura, B. Iancu, V. T. Dadarlat, A. Peculea, and E. Cebuc. “Optimizations for Energy Efficiency in Software-Defined Wireless Sensor Networks”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[13] R. Wang, Z. Zhang, Z. Zhang, and Z. Jia. “ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs”. In: *Comput. Networks* 139 (2018), pp. 119–135.

[14] F. Junli, W. Yawen, and S. Haibin. “An improved energy-efficient routing algorithm in software define wireless sensor network”. In: *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)* (2017), pp. 1–5.

[15] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen. “A Survey and Future Directions on Clustering: From WSNs to IoT and Modern Networking Paradigms”. In: *IEEE Transactions on Network and Service Management* 18 (2021), pp. 2242–2274.

[16] T. Buckley, B. Ghosh, and V. Pakrashi. “Edge Structural Health Monitoring (E-SHM) Using Low-Power Wireless Sensing”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[17] J. L. Álvarez, J. D. Mozo, and E. Durán. “Analysis of Single Board Architectures Integrating Sensors Technologies”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[18] A. K. Dey. “Understanding and Using Context”. In: *Personal and Ubiquitous Computing* 5 (2001), pp. 4–7.

[19] A. Al-alshuhai and F. Siewe. “An extension of the use case diagram to model context-aware applications”. In: *2015 SAI Intelligent Systems Conference (IntelliSys)* (2015), pp. 884–888.

[20] D. Salber, A. K. Dey, and G. D. Abowd. “The context toolkit: aiding the development of context-enabled applications”. In: *International Conference on Human Factors in Computing Systems*. 1999.

[21] C. Hoareau and I. Satoh. “Modeling and Processing Information for Context-Aware Computing: A Survey”. In: *New Generation Computing* 27 (2009), pp. 177–196.

[22] A. E. Ghazi, Z. Aarab, and B. Ahiod. “Context-aware routing protocol based on PSO for mobile WSN”. In: *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)* (2017), pp. 1–6.

[23] K. Bajaj, B. Sharma, and R. Singh. “Implementation analysis of IoT-based offloading frameworks on cloud/edge computing for sensor generated big data”. In: *Complex & Intelligent Systems* 8 (2021), pp. 3641–3658.

[24] A. Yousefpour, C. Fung, T. Nguyen, K. P. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue. “All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey”. In: *ArXiv* abs/1808.05283 (2018).

[25] J. Pan and J. McElhannon. “Future Edge Cloud and Edge Computing for Internet of Things Applications”. In: *IEEE Internet of Things Journal* 5 (2018), pp. 439–449.

[26] L. Sánchez, J. Lanza, R. L. Olsen, M. P. Bauer, and M. Girod-Genet. “A Generic Context Management Framework for Personal Networking Environments”. In: *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services* (2006), pp. 1–8.

[27] S. J. Russell and P. Norvig. “Artificial intelligence - a modern approach, 2nd Edition”. In: *Prentice Hall series in artificial intelligence*. 2003.

[28] J. Kelleher. *Deep Learning*. MIT Press essential knowledge series. MIT Press, 2019. ISBN: 9780262354899. URL: `https://books.google.se/books?id=p%5C_fTxQEACAAJ`.

[29] R. Off. “The time scale of artificial intelligence: Reflections on social effects”. In: 1985.

[30] K. P. Murphy. “Machine learning - a probabilistic perspective”. In: *Adaptive computation and machine learning series*. 2012.

[31] I. H. Witten, E. Frank, and M. A. Hall. “Data Mining: Practical Machine Learning Tools and Techniques, 3/E”. In: 2011.

[32] J. Patterson and A. Gibson. *Deep Learning: A Practitioner’s Approach*. 1st. O’Reilly Media, Inc., 2017. ISBN: 1491914254.

[33] S. Li and J. G. Kim. “Maximizing the lifetime of wireless sensor networks with random forwarding”. In: *Aeu-international Journal of Electronics and Communications* 69 (2015), pp. 455–457.

[34] A. Noel, A. Abdaoui, T. M. Elfouly, M. H. Ahmed, A. A. Badawy, and M. S. Shehata. “Structural Health Monitoring Using Wireless Sensor Networks: A Comprehensive Survey”. In: *IEEE Communications Surveys & Tutorials* 19 (2017), pp. 1403–1423.

[35] S. Chitnis, N. Deshpande, and A. Shaligram. “An Investigative Study for Smart Home Security: Issues, Challenges and Countermeasures”. In: *Wireless Sensor Network* 08 (2016), pp. 61–68.

[36] B. Rashid and M. H. Rehmani. “Applications of wireless sensor networks for urban areas: A survey”. In: *J. Netw. Comput. Appl.* 60 (2016), pp. 192–219.

[37] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen. “Clustering objectives in wireless sensor networks: A survey and research direction analysis”. In: *Comput. Networks* 180 (2020), p. 107376.

[38] K. Bajaj, B. Sharma, and R. Singh. “Integration of WSN with IoT Applications: A Vision, Architecture, and Future Challenges”. In: *Integration of WSN and IoT for Smart Cities*. 2020.

[39] P. Jiang. “A New Method for Node Fault Detection in Wireless Sensor Networks”. In: *Sensors (Basel, Switzerland)* 9 (2009), pp. 1282–1294.

[40] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen. “A Survey on Internet of Things From Industrial Market Perspective”. In: *IEEE Access* 2 (2014), pp. 1660–1679.

[41] U. A. Ibarra, J. C. Augusto, and T. Clark. “Engineering context-aware systems and applications: A survey”. In: *J. Syst. Softw.* 117 (2016), pp. 55–83.

[42] H. Vahdat-Nejad, E. Asani, Z. Mahmoodian, and M. H. Mohseni. “Context-aware computing for mobile crowd sensing: A survey”. In: *Future Generation Computer Systems* 99 (2019), pp. 321–332. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2019.04.052. URL: https://www.sciencedirect.com/science/article/pii/S0167739X18329583.

[43] L. Miranda, J. Viterbo, and F. C. Bernardini. “A survey on the use of machine learning methods in context-aware middlewares for human activity recognition”. In: *Artif. Intell. Rev.* 55 (2022), pp. 3369–3400.

[44] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu. “Context-Aware Computing, Learning, and Big Data in Internet of Things: A Survey”. In: *IEEE Internet of Things Journal* 5.1 (2018), pp. 1–27. DOI: 10.1109/JIOT.2017.2773600.

[45] T. E. Bogale, X. Wang, and L. B. Le. *Machine Intelligence Techniques for Next-Generation Context-Aware Wireless Networks*. 2018.

[46] P. Pradeep and S. Krishnamoorthy. “The MOM of context-aware systems: A survey”. In: *Computer Communications* 137 (2019), pp. 44–69. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0140366418309472.

[47] L. Shuai, Z. Xueyan, S. Xiaodong, Y. Xiaohan, T. Rui-chun, and J. Qingyun. “Survey on Context-aware Systems and Their Applications”. In: *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)* (2019), pp. 234–237.

[48] B. Chatterjee, S. Sen, N. Cao, and A. Raychowdhury. “Context-Aware Intelligence in Resource-Constrained IoT Nodes: Opportunities and Challenges”. In: *IEEE Design & Test* 36 (2019), pp. 7–40.

[49] M. del Carmen Rodríguez-Hernández and S. Ilarri. “AI-based mobile context-aware recommender systems from an information management perspective: Progress and directions”. In: *Knowl. Based Syst.* 215 (2021), p. 106740.

[50] R. Rekik, I. Kallel, J. Casillas, and A. M. Alimi. “Assessing web sites quality: A systematic literature review by text and association rules mining”. In: *Int. J. Inf. Manag.* 38 (2018), pp. 201–216.

[51] S. Gupta, A. K. Kar, A. M. Baabdullah, and W. Al-Khowaiter. “Big data with cognitive computing: A review for the future”. In: *Int. J. Inf. Manag.* 42 (2018), pp. 78–89.

[52] V. Boeva, L. Boneva, and E. Tsiporkova. “Semantic-Aware Expert Partitioning”. In: *Artificial Intelligence: Methodology, Systems, and Applications*. Ed. by G. Agre, P. Hitzler, A. A. Krisnadhi, and S. O. Kuznetsov. Cham: Springer International Publishing, 2014, pp. 13–24.

[53] P. Jaccard. “Étude comparative de la distribution florale dans une portion des Alpes et du Jura”. In: *Bulletin del la Société Vaudoise des Sciences Naturelles* (1901).

[54] H. Huang, P. Zhou, Y. Li, and F. Sun. “A Lightweight Attention-Based CNN Model for Efficient Gait Recognition with Wearable IMU Sensors”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[55] P. Paudel, S. Kim, S. Park, and K. Choi. “A Context-Aware IoT and Deep-Learning-Based Smart Classroom for Controlling Demand and Supply of Power Load”. In: *Electronics* 9 (2020), p. 1039.

[56] Y. Lu, S. Zhang, Z. Zhang, W. Xiao, and S. Yu. “A Framework for Learning Analytics Using Commodity Wearable Devices”. In: *Sensors (Basel, Switzerland)* 17 (2017).

[57] Z. Chen, J. Chen, and X. Huang. “An Activity-Aware Sampling Scheme for Mobile Phones in Activity Recognition”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[58] V. Pejovic and M. Musolesi. “Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges”. In: *ACM Comput. Surv.* 47.3 (2015). ISSN: 0360-0300.

[59] M. Peleg, Y. Shahar, S. Quaglini, T. H. F. Broens, R. I. Budasu, N. L. S. Fung, A. Fux, G. García-Sáez, A. Goldstein, A. González-Ferrer, H. J. Hermens, M. E. Hernando, V. Jones, G. Klebanov, D. Klimov, D. Knoppel, N. Larburu, C. M. Lagunar, I. Martínez-Sarriegui, C. Napolitano, A. Pallàs, E. Parimbelli, and et al. “Assessment of a personalized and distributed patient guidance system”. In: *International journal of medical informatics* 101 (2017), pp. 108–130.

[60] M. A. U. Alam, N. Roy, S. D. Holmes, A. Gangopadhyay, and E. Galik. “AutoCogniSys: IoT Assisted Context-Aware Automatic Cognitive Health Assessment”. In: *ArXiv* 2003.07492 (2020).

[61] M. Rabbi, A. F. Pfammatter, M. Zhang, B. J. Spring, and T. Choudhury. “Automated Personalized Feedback for Physical Activity and Dietary Behavior Change With Mobile Phones: A Randomized Controlled Trial on Adults”. In: *JMIR mHealth and uHealth* 3 (2015).

[62] J. Hauth, S. Jabri, F. Kamran, E. W. Feleke, K. Nigusie, L. V. Ojeda, S. Handelzalts, L. V. Nyquist, N. B. Alexander, X. Huan, J. Wiens, and K. H. Sienko. "Automated Loss-of-Balance Event Identification in Older Adults at Risk of Falls during Real-World Walking Using Wearable Inertial Measurement Units". In: *Sensors (Basel, Switzerland)* 21 (2021).

[63] M. Ehatisham-ul-Haq, M. A. Azam, Y. Amin, and U. Naeem. "C2FHAR: Coarse-to-Fine Human Activity Recognition With Behavioral Context Modeling Using Smart Inertial Sensors". In: *IEEE Access* 8 (2020), pp. 7731–7747.

[64] C. Bettini, G. Civitarese, and R. Presotto. "CAVIAR: Context-driven Active and Incremental Activity Recognition". In: *Knowl. Based Syst.* 196 (2020), p. 105816.

[65] N. D. Lane and P. Georgiev. "Can Deep Learning Revolutionize Mobile Sensing?" In: *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (2015).

[66] M. Ehatisham-ul-Haq, M. A. Azam, U. Naeem, Y. Amin, and J. K.-K. Loo. "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing". In: *J. Netw. Comput. Appl.* 109 (2018), pp. 24–35.

[67] X. Zhou, W. Liang, K. I.-K. Wang, H. Wang, L. T. Yang, and Q. Jin. "Deep-Learning-Enhanced Human Activity Recognition for Internet of Healthcare Things". In: *IEEE Internet of Things Journal* 7 (2020), pp. 6429–6438.

[68] R. Shaukat-Jali, N. van Zalk, and D. E. Boyle. "Detecting Subclinical Social Anxiety Using Physiological Data From a Wrist-Worn Wearable: Small-Scale Feasibility Study". In: *JMIR Formative Research* 5 (2021).

[69] A. I. Petrenko, R. Kyslyi, and I. Pysmennyi. "Detection of human respiration patterns using deep convolution neural networks". In: *Eastern-European Journal of Enterprise Technologies* (2018).

[70] C. Culman, S. Aminikhanghahi, and D. J. Cook. "Easing Power Consumption of Wearable Activity Monitoring with Change Point Detection". In: *Sensors (Basel, Switzerland)* 20 (2020).

[71] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong, and M. Li. "Fine-Grained Abnormal Driving Behaviors Detection and Identification with Smartphones". In: *IEEE Transactions on Mobile Computing* 16 (2017), pp. 2198–2212.

[72] A. Mehrotra, V. Pejović, and M. Musolesi. "FutureWare: Designing a Middleware for Anticipatory Mobile Computing". In: *IEEE Transactions on Software Engineering* 47 (2021), pp. 2107–2124.

[73] T. Choksatchawathi, P. Ponglertnapakorn, A. Ditthapron, P. Leelaarporn, T. Wisutthisen, M. Piriyajitakonkij, and T. Wilaiprasitporn. “Improving Heart Rate Estimation on Consumer Grade Wrist-Worn Device Using Post-Calibration Approach”. In: *IEEE Sensors Journal* 20 (2020), pp. 7433–7446.

[74] A. Filippoupolitis, W. Oliff, B. Takand, and G. Loukas. “Location-Enhanced Activity Recognition in Indoor Environments Using Off the Shelf Smart Watch Technology and BLE Beacons”. In: *Sensors (Basel, Switzerland)* 17 (2017).

[75] J. Jansson and I. Hakala. “Managing sensor data streams in a smart home application”. In: *Int. J. Sens. Networks* 32 (2020), pp. 247–258.

[76] Z. E. Ashari, N. S. Chaytor, D. J. Cook, and H. Ghasemzadeh. “Memory-Aware Active Learning in Mobile Sensing Systems”. In: *IEEE Transactions on Mobile Computing* 21 (2022), pp. 181–195.

[77] K. Fujinami. “On-Body Smartphone Localization with an Accelerometer”. In: *Inf.* 7 (2016), p. 21.

[78] Y. Liang, H.-W. Fan, Z. Fang, L. Miao, W. Li, X. Zhang, W. Sun, K. Wang, L. He, and X. ’. Chen. “OralCam: Enabling Self-Examination and Awareness of Oral Health Using a Smartphone Camera”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020).

[79] D. C. Mohr, M. Zhang, and S. M. Schueller. “Personal Sensing: Understanding Mental Health Using Ubiquitous Sensors and Machine Learning.” In: *Annual review of clinical psychology* 13 (2017), pp. 23–47.

[80] C. Bettini, G. Civitarese, D. Giancane, and R. Presotto. “ProCAVIAR: Hybrid Data-Driven and Probabilistic Knowledge-Based Activity Recognition”. In: *IEEE Access* 8 (2020), pp. 146876–146886.

[81] K. Peppas, A. C. Tsolakis, S. Krinidis, and D. Tzovaras. “Real-Time Physical Activity Recognition on Smart Mobile Devices Using Convolutional Neural Networks”. In: *Applied Sciences* 10 (2020), p. 8482.

[82] Y. Liu, G. Li, and L. Lin. “Semantics-Aware Adaptive Knowledge Distillation for Sensor-to-Vision Action Recognition”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5573–5588.

[83] S. Chung, J. Lim, K. J. Noh, G. Kim, and H. Jeong. “Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning”. In: *Sensors (Basel, Switzerland)* 19 (2019).

[84] U. R. Alo, H. F. Nweke, T. Y. Wah, and G. Murtaza. “Smartphone Motion Sensor-Based Complex Human Activity Identification Using Deep Stacked Autoencoder Algorithm for Enhanced Smart Healthcare System”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[85] R. Jackermeier and B. Ludwig. “Smartphone-Based Activity Recognition in a Pedestrian Navigation Context”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[86] B. Calabrese, R. Velázquez, C. Del-Valle-Soto, R. de Fazio, N. I. Giannoccaro, and P. Visconti. “Solar-Powered Deep Learning-Based Recognition System of Daily Used Objects and Human Faces for Assistance of the Visually Impaired”. In: *Energies* 13 (2020), p. 6104.

[87] J. Ranjan and K. Whitehouse. “Towards recognizing person-object interactions using a single wrist wearable device”. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (2016).

[88] T. Zhang, A. H. Sodhro, Z. Luo, N. Zahid, M. I. Nawaz, S. Pirbhulal, and M. Muzammal. “A Joint Deep Learning and Internet of Medical Things Driven Framework for Elderly Patients”. In: *IEEE Access* 8 (2020), pp. 75822–75832.

[89] C. Dobbins, S. H. Fairclough, P. J. G. Lisboa, and F. F. González-Navarro. “A Lifelogging Platform Towards Detecting Negative Emotions in Everyday Life using Wearable Devices”. In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (2018), pp. 306–311.

[90] Y. Wang, J. Yan, Q. Sun, J. Li, and Z. Yang. “A MobileNets Convolutional Neural Network for GIS Partial Discharge Pattern Recognition in the Ubiquitous Power Internet of Things Context: Optimization, Comparison, and Application”. In: *IEEE Access* 7 (2019), pp. 150226–150236.

[91] A. Thiebault, C. Huetz, P. A. Pistorius, T. Aubin, and I. Charrier. “Animal-borne acoustic data alone can provide high accuracy classification of activity budgets”. In: *Animal Biotelemetry* 9 (2021), pp. 1–16.

[92] V. Marinakis. “Big Data for Energy Management and Energy-Efficient Buildings”. In: *Energies* 13 (2020), p. 1555.

[93] U. Saeed, Y.-D. Lee, S. U. Jan, and I. Koo. “CAFD: Context-Aware Fault Diagnostic Scheme towards Sensor Faults Utilizing Machine Learning”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[94] N. Momeni, A. Arza, J. Rodrigues, C. Sandi, and D. A. Alonso. “CAFS: Cost-Aware Features Selection Method for Multimodal Stress Monitoring on Wearable Devices”. In: *IEEE Transactions on Biomedical Engineering* 69 (2022), pp. 1072–1084.

[95] B. Sliwa, R. Adam, and C. Wietfeld. “Client-Based Intelligence for Resource Efficient Vehicular Big Data Transfer in Future 6G Networks”. In: *IEEE Transactions on Vehicular Technology* 70 (2021), pp. 5332–5346.

[96] T. Yang, P. Guo, W. Liu, X. Liu, and T. Hao. “DeepPIRATES: A Training-Light PIR-Based Localization Method With High Generalization Ability”. In: *IEEE Access* 9 (2021), pp. 86054–86061.

[97] P. G. Madoery, R. Detke, L. Blanco, S. Comerci, J. A. Fraire, A. M. González-Montoro, J. C. Bellassai, G. M. Britos, S. M. Ojeda, and J. M. Finochietto. “Feature selection for proximity estimation in COVID-19 contact tracing apps based on Bluetooth Low Energy (BLE)”. In: *Pervasive and Mobile Computing* 77 (2021), pp. 101474–101474.

[98] M. E. Gendy, A. Al-Kabbany, and E. F. Badran. “Green CrowdSensing With Comprehensive Reputation Awareness and Predictive Device-Application Matching Using a New Real-Life Dataset”. In: *IEEE Access* 8 (2020), pp. 225757–225776.

[99] L. Ferrari, F. Dell’acqua, P. Zhang, and P. Du. “Integrating EfficientNet into an HAFNet Structure for Building Mapping in High-Resolution Optical Earth Observation Data”. In: *Remote. Sens.* 13 (2021), p. 4361.

[100] O. Boursalie, R. Samavi, and T. E. Doyle. “M4CVD: Mobile Machine Learning Model for Monitoring Cardiovascular Disease”. In: *EUSPN/ICTH*. 2015.

[101] G. Healey and S. Zhao. “Measurement Space Partitioning for Estimation and Prediction”. In: *IEEE Access* 9 (2021), pp. 137419–137429.

[102] F. Rossier, P. Lang, and J. Hennebert. “Near Real-Time Appliance Recognition Using Low Frequency Monitoring and Active Learning Methods”. In: *Energy Procedia* 122 (2017), pp. 691–696.

[103] S. Alghamdi, E. A. Fadel, and N. Alowidi. “Recognizing Activities of Daily Living using 1D Convolutional Neural Networks for Efficient Smart Homes”. In: *International Journal of Advanced Computer Science and Applications* 12 (2021).

[104] S. Sukreep, K. Elgazzar, C. H. Chu, C. Nukoolkit, and P. Mongkolnam. “Recognizing Falls, Daily Activities, and Health Monitoring by Smart Devices”. In: *Sensors and Materials* (2019).

[105] W. Seo, S. Cha, Y. Kim, J. Huh, and J. Park. “SLO-Aware Inference Scheduler for Heterogeneous Processors in Edge Platforms”. In: *ACM Transactions on Architecture and Code Optimization (TACO)* 18 (2021), pp. 1–26.

[106] S. Saeb, T. R. Cybulski, S. M. Schueller, K. P. Kording, and D. C. Mohr. “Scalable Passive Sleep Monitoring Using Mobile Phones: Opportunities and Obstacles”. In: *Journal of Medical Internet Research* 19 (2017).

[107] R. Roor, J. Hess, M. Saveriano, M. Karg, and A. Kirsch. “Sensor Fusion for Semantic Place Labeling”. In: *VEHITS*. 2017.

[108] J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gasmann, J. Klucken, and B. Eskofier. “Sensor-Based Gait Parameter Extraction With Deep Convolutional Neural Networks”. In: *IEEE Journal of Biomedical and Health Informatics* 21 (2017), pp. 85–93.

[109] A. O. Akmandor, H. Yin, and N. K. Jha. “Smart, Secure, Yet Energy-Efficient, Internet-of-Things Sensors”. In: *IEEE Transactions on Multi-Scale Computing Systems* 4 (2018), pp. 914–930.

[110] B. Alotaibi. “Transportation Mode Detection by Embedded Sensors Based on Ensemble Learning”. In: *IEEE Access* 8 (2020), pp. 145552–145563.

[111] M. Zappatore, C. Loglisci, A. Longo, M. A. Bochicchio, L. Vaira, and D. Malerba. “Trustworthiness of Context-Aware Urban Pollution Data in Mobile Crowd Sensing”. In: *IEEE Access* 7 (2019), pp. 154141–154156.

[112] M. Magno, L. Cavigelli, R. Andri, and L. Benini. “Ultra-Low Power Context Recognition Fusing Sensor Data from an Energy-Neutral Smart Watch”. In: *IoT 360*. 2015.

[113] F. Alqahtani, S. Katsigiannis, and N. Ramzan. “Using Wearable Physiological Sensors for Affect-Aware Intelligent Tutoring Systems”. In: *IEEE Sensors Journal* 21 (2021), pp. 3366–3378.

[114] N. Dipsis and K. Stathis. “A RESTful middleware for AI controlled sensors, actuators and smart devices”. In: *Journal of Ambient Intelligence and Humanized Computing* 11 (2020), pp. 2963–2986.

[115] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic. “A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition”. In: *Sensors (Basel, Switzerland)* 17 (2017).

[116] A. K. Sikder, H. Aksu, and A. S. Uluagac. “A Context-Aware Framework for Detecting Sensor-Based Threats on Smart Devices”. In: *IEEE Transactions on Mobile Computing* 19 (2020), pp. 245–261.

[117] L. Zhang, Y. Zhu, M. Jiang, Y. Wu, K. Deng, and Q. Ni. “Body Temperature Monitoring for Regular COVID-19 Prevention Based on Human Daily Activity Recognition”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[118] S. A. Khowaja, B. N. Yahya, and S.-L. Lee. “CAPHAR: context-aware personalized human activity recognition using associative learning in smart environments”. In: *Human-centric Computing and Information Sciences* 10 (2020), pp. 1–35.

[119] M. Xu, F. Qian, M. Zhu, F. Huang, S. Pushp, and X. Liu. “DeepWear: Adaptive Local Offloading for On-Wearable Deep Learning”. In: *IEEE Transactions on Mobile Computing* 19 (2020), pp. 314–330.

[120] T. G. Stavropoulos, G. Meditskos, I. Lazarou, L. Mpaltadoros, S. Papagiannopoulos, M. Tsolaki, and Y. Kompatsiaris. “Detection of Health-Related Events and Behaviours from Wearable Sensor Lifestyle Data Using Symbolic Intelligence: A Proof-of-Concept Application in the Care of Multiple Sclerosis”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[121] E. D. Corso, T. Cerquitelli, and D. Apiletti. “METATECH: METeorological Data Analysis for Thermal Energy CHaracterization by Means of Self-Learning Transparent Models”. In: *Energies* 11 (2018), p. 1336.

[122] H. J. Han, S. Labbaf, J. L. Borelli, N. D. Dutt, and A.-M. Rahmani. “Objective stress monitoring based on wearable sensors in everyday settings”. In: *Journal of Medical Engineering & Technology* 44 (2020), pp. 177–189.

[123] M. Muñoz-Organero. “Outlier Detection in Wearable Sensor Data for Human Activity Recognition (HAR) Based on DRNNs”. In: *IEEE Access* 7 (2019), pp. 74422–74436.

[124] G. Aldaz, S. Puria, and L. J. Leifer. “Smartphone-Based System for Learning and Inferring Hearing Aid Settings.” In: *Journal of the American Academy of Audiology* 27 9 (2016), pp. 732–749.

[125] B. Islam and S. M. S. Nirjon. “Zygarde: Time-Sensitive On-Device Deep Inference and Adaptation on Intermittently-Powered Systems”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4 (2020), 82:1–82:29.

[126] S. Samyoun, S. S. Shubha, M. A. S. Mondol, and J. A. Stankovic. “iWash: A smartwatch handwashing quality assessment and reminder system with real-time feedback in the context of infectious disease”. In: *Smart Health (Amsterdam, Netherlands)* 19 (2021), pp. 100171–100171.

[127] J. Mendez, M. Molina, N. Rodríguez, M. P. Cuéllar, and D. P. Morales. “Camera-LiDAR Multi-Level Sensor Fusion for Target Detection at the Network Edge”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[128] M. Alharbi and H. A. Karimi. “Context-Aware Sensor Uncertainty Estimation for Autonomous Vehicles”. In: *Vehicles* (2021).

[129] K. L. Mugumya, J.-Y. Wong, A. Chan, C.-C. Yip, and S. Ghazy. “Indoor haze particulate control using knowledge graphs within self-optimizing HVAC control systems”. In: 2020.

[130] K. Qian, T. Koike, K. Yoshiuchi, B. W. Schuller, and Y. Yamamoto. “Can Appliances Understand the Behavior of Elderly Via Machine Learning? A Feasibility Study”. In: *IEEE Internet of Things Journal* 8 (2021), pp. 8343–8355.

[131] S. Koldijk, M. A. Neerincx, and W. Kraaij. “Detecting Work Stress in Offices by Combining Unobtrusive Sensors”. In: *IEEE Transactions on Affective Computing* 9 (2018), pp. 227–239.

[132] G. Schiboni, J. C. Suarez, R. Zhang, and O. Amft. “DynDSE: Automated Multi-Objective Design Space Exploration for Context-Adaptive Wearable IoT Edge Devices”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[133] E. Khodabandehloo, D. Riboni, and A. Alimohammadi. “HealthXAI: Collaborative and explainable AI for supporting early diagnosis of cognitive decline”. In: *Future Gener. Comput. Syst.* 116 (2021), pp. 168–189.

[134] S. Chakkor, M. Baghouri, Z. Cheker, A. E. Oualkadi, J. A. el Hangouche, and J. Laamech. “Intelligent Network for Proactive Detection of COVID-19 Disease”. In: *2020 6th IEEE Congress on Information Science and Technology (CiSt)* (2020), pp. 472–478.

[135] L. Santamaria-Granados, J. F. Mendoza-Moreno, Á. C. Astaiza, M. M. Organero, and G. Ramírez-González. “Tourist Experiences Recommender System Based on Emotion Recognition with Wearable Data”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[136] M. D. Brouwer, F. Ongenae, P. Bonte, and F. D. Turck. “Towards a Cascading Reasoning Framework to Support Responsive Ambient-Intelligent Healthcare Interventions”. In: *Sensors (Basel, Switzerland)* 18 (2018).

[137] N. M. do Nascimento, P. S. C. Alencar, C. J. P. Lucena, and D. D. Cowan. “An IoT Analytics Embodied Agent Model based on Context-Aware Machine Learning”. In: *2018 IEEE International Conference on Big Data (Big Data)* (2018), pp. 5170–5175.

[138] D. Dzemydienė and A. Burinskienė. “Integration of Context Awareness in Smart Service Provision System Based on Wireless Sensor Networks for Sustainable Cargo Transportation”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[139] A. E. Kelishomi, A. H. S. Garmabaki, M. Bahaghighat, and J. Dong. “Mobile User Indoor-Outdoor Detection through Physical Daily Activities”. In: *Sensors (Basel, Switzerland)* 19 (2019).

[140] G. Orsini, W. Posdorfer, and W. Lamersdorf. “Saving bandwidth and energy of mobile and IoT devices with link predictions”. In: *J. Ambient Intell. Humaniz. Comput.* 12 (2021), pp. 8229–8240.

[141] E. L. Lydia, A. A. Jovith, A. F. S. Devaraj, C. Seo, and G. P. Joshi. “Green Energy Efficient Routing with Deep Learning Based Anomaly Detection for Internet of Things (IoT) Communications”. In: 2021.

[142] Y. Zhang, A. K. Srivastava, and D. J. Cook. “Machine learning algorithm for activity-aware demand response considering energy savings and comfort requirements”. In: 2020.

[143] I. C. Draa, S. Niar, J. Tayeb, E. G.-L. Strugeon, and M. Desertot. “Sensing user context and habits for run-time energy optimization”. In: *EURASIP Journal on Embedded Systems* 2017 (2017), pp. 1–19.

[144] S. Rivera, O. Mendoza-Schrock, and A. Diehl. “Transfer Learning for Aided Target Recognition: Comparing Deep Learning to other Machine Learning Approaches”. In: *ArXiv* 2011.12762 (2020).

[145] G. Fenza, M. Gallo, V. Loia, D. Marino, and F. J. Orciuoli. “A Cognitive Approach based on the Actionable Knowledge Graph for supporting Maintenance Operations”. In: *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)* (2020), pp. 1–7.

[146] C. Turner, C. Emmanouilidis, T. Tomiyama, A. Tiwari, and R. Roy. “Intelligent decision support for maintenance: an overview and future trends”. In: *International Journal of Computer Integrated Manufacturing* 32 (2019), pp. 936–959.

[147] R. Xu, W. Jin, Y. Hong, and D. Kim. “Intelligent Optimization Mechanism Based on an Objective Function for Efficient Home Appliances Control in an Embedded Edge Platform”. In: *Electronics* (2021).

[148] A. R. Ruiz, D. Villa, R. C. Navarro, M. J. S. Romero, J. D. Chaparro, and J. Lopez. “Leveraging commonsense reasoning towards a smarter Smart Home”. In: *KES*. 2021.

[149] F. Waldner and F. I. Diakogiannis. “Deep learning on edge: extracting field boundaries from satellite images with a convolutional neural network”. In: *ArXiv* abs/1910.12023 (2020).

[150] A. D’Eusanio, A. Simoni, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara. “Multimodal Hand Gesture Classification for the Human-Car Interaction”. In: *Informatics* 7 (2020), p. 31.

[151] B. C. Mocanu, R. Tapu, and T. B. Zaharia. “When Ultrasonic Sensors and Computer Vision Join Forces for Efficient Obstacle Detection and Recognition”. In: *Sensors (Basel, Switzerland)* 16 (2016).

[152] P. Augustyniak and G. Ślusarczyk. “Graph-based representation of behavior in detection and prediction of daily living activities”. In: *Computers in biology and medicine* 95 (2018), pp. 261–270.

[153] D. Dalmazzo and R. Ramírez. “Air violin: a machine learning approach to fingering gesture recognition”. In: *Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education* (2017).

[154] Y. Kim, M. Imani, and T. Simunic. “Efficient human activity recognition using hyperdimensional computing”. In: *Proceedings of the 8th International Conference on the Internet of Things* (2018).

[155] Y. Bhatia, A. H. Bari, G.-S. J. Hsu, and M. Gavrilova. “Motion Capture Sensor-Based Emotion Recognition Using a Bi-Modular Sequential Neural Network”. In: *Sensors (Basel, Switzerland)* 22 (2022).

[156] B. Ao, Y. Wang, H. Liu, D. Li, L. Song, and J. Li. “Context Impacts in Accelerometer-Based Walk Detection and Step Counting”. In: *Sensors (Basel, Switzerland)* 18 (2018).

[157] A. Angrisano, M. L. Bernardi, M. Cimitile, S. Gaglione, and M. Vultaggio. “Identification of Walker Identity Using Smartphone Sensors: An Experiment Using Ensemble Learning”. In: *IEEE Access* 8 (2020), pp. 27435–27447.

[158] R. Zhao, R. Yan, J. Wang, and K. Mao. “Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks”. In: *Sensors (Basel, Switzerland)* 17 (2017).

[159] Y. Zhou, S. Hong, J. Shang, M. Wu, Q. Wang, H. Li, and J. Xie. “Addressing Noise and Skewness in Interpretable Health-Condition Assessment by Learning Model Confidence †”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[160] E. Hitimana, G. Bajpai, R. Musabe, L. Sibomana, and K. Jayavel. “Implementation of IoT Framework with Data Analysis Using Deep Learning Methods for Occupancy Prediction in a Building”. In: *Future Internet* 13 (2021), p. 67.

[161] J. Giménez-Gallego, J. D. González-Teruel, F. Soto-Vallés, M. J. Buendía, H. Navarro-Hellín, and R. Torres-Sánchez. “Intelligent thermal image-based sensor for affordable measurement of crop canopy temperature”. In: *Comput. Electron. Agric.* 188 (2021), p. 106319.

[162] A. Chadwick, N. C. Coops, C. W. Bater, L. A. Martens, and B. White. “Species Classification of Automatically Delineated Regenerating Conifer Crowns Using RGB and Near-Infrared UAV Imagery”. In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5.

[163] A. Tazarv, S. Labbaf, S. M. Reich, N. Dutt, A.-M. Rahmani, and M. Levorato. “Personalized Stress Monitoring using Wearable Sensors in Everyday Settings”. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2021), pp. 7332–7335.

[164] W. Xie, J. S. Ide, D. Izadi, S. Banger, T. T. Walker, R. Ceresani, D. S. Spagnuolo, C. Guagliano, H. Diaz, and J. C. Twedt. “Multi-object tracking with deep learning ensemble for unmanned aerial system applications”. In: *Security + Defence*. 2021.

[165] Z. Wang, Y. Li, D. Li, M. Li, B. Zhang, S. Huang, and W. He. “Enabling Fairness-Aware and Privacy-Preserving for Quality Evaluation in Vehicular Crowdsensing: A Decentralized Approach”. In: *Security and Communication Networks* (2021).

[166] C. Stach, C. Giebler, M. Wagner, C. Weber, and B. Mitschang. “AMNESIA: A Technical Solution towards GDPR-compliant Machine Learning”. In: *ICISSP*. 2020.

[167] K. S.-H. Ong, D. T. Niyato, and C. Yuen. “Predictive Maintenance for Edge-Based Sensor Networks: A Deep Reinforcement Learning Approach”. In: *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)* (2020), pp. 1–6.

[168] Y. Zhou, Y. Chen, Y. Ma, and H. Liu. “A Real-Time Dual-Microphone Speech Enhancement Algorithm Assisted by Bone Conduction Sensor”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[169] R. Ali, I. Ashraf, A. K. Bashir, and Y. B. Zikria. “Reinforcement-Learning-Enabled Massive Internet of Things for 6G Wireless Communications”. In: *IEEE Communications Standards Magazine* 5 (2021), pp. 126–131.

[170] A. Musaddiq, Z. Nain, Y. A. Qadri, R. Ali, and S. W. Kim. “Reinforcement Learning-Enabled Cross-Layer Optimization for Low-Power and Lossy Networks under Heterogeneous Traffic Patterns”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[171] A. Khan, A. I. Umar, A. Munir, S. H. Shirazi, M. A. Khan, and M. Adnan. “A QoS-Aware Machine Learning-Based Framework for AMI Applications in Smart Grids”. In: *Energies* (2021).

[172] L. Galindez, K. M. H. Badami, J. Vlasselaer, W. Meert, and M. Verhelst. “Dynamic Sensor-Frontend Tuning for Resource Efficient Embedded Classification”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8 (2018), pp. 858–872.

[173] Y. Zhang, T. Gu, and X. Zhang. “MDLdroid: a ChainSGD-reduce Approach to Mobile Deep Learning for Personal Mobile Sensing”. In: *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (2020), pp. 73–84.

[174] P. Fabian and A. Rachedi. “Dynamic selection of relays based on classification of mobility profile in a highly mobile context”. In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)* (2020), pp. 1–6.

[175] G. Masinelli, F. Forooghifar, A. Arza, D. A. Alonso, and A. Aminifar. “Self-Aware Machine Learning for Multimodal Workload Monitoring during Manual Labor on Edge Wearable Sensors”. In: *IEEE Design & Test* 37 (2020), pp. 58–66.

[176] J. A. González, L. A. Cheah, A. M. Gómez, P. D. Green, J. M. Gilbert, S. R. Ell, R. K. Moore, and E. Holdsworth. “Direct Speech Reconstruction From Articulatory Sensor Data by Machine Learning”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (2017), pp. 2362–2374.

[177] A. D. Singh, S. S. Sandha, L. Garcia, and M. B. Srivastava. “RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-wave Radar”. In: *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems* (2019).

[178] Y. Yin, Y. Zhang, Z. Liu, Y. Liang, S. Wang, R. R. Shah, and R. Zimmermann. "Learning Multi-context Aware Location Representations from Large-scale Geotagged Images". In: *Proceedings of the 29th ACM International Conference on Multimedia* (2021).

[179] A. A. M. Al-Saedi, V. Boeva, and E. Casalicchio. "Reducing Communication Overhead of Federated Learning through Clustering Analysis". In: *2021 IEEE Symposium on Computers and Communications (ISCC)* (2021), pp. 1–7.

[180] Warnat-Herresthal and S. et al. "Swarm Learning for decentralized and confidential clinical machine learning". In: *Nature* 594 (2021), pp. 265–270.

[181] J.-P. Rodrigue. *The geography of transport systems*. Routledge, 2020.

[182] Y. Zhu, H. Luo, Q. Wang, F. Zhao, B. Ning, Q. Ke, and C. Zhang. "A Fast Indoor/Outdoor Transition Detection Algorithm Based on Machine Learning". In: *Sensors* 19.4 (2019).

[183] J. Z. Hare, S. Gupta, J. Song, and T. A. Wettergren. "Classification induced distributed sensor scheduling for energy-efficiency in underwater target tracking sensor networks". In: *OCEANS 2017 – Anchorage* (2017), pp. 1–7.

[184] M. Woschank, E. Rauch, and H. Zsifkovits. "A Review of Further Directions for Artificial Intelligence, Machine Learning, and Deep Learning in Smart Logistics". In: *Sustainability* 12.9 (2020). ISSN: 2071-1050. DOI: 10.3390/su12093760. URL: https://www.mdpi.com/2071-1050/12/9/3760.

Paper IV

FedCO: Communication-Efficient Federated Learning via Clustering Optimization †

Ahmed A. Al-Saedi, Veselka Boeva, and Emiliano Casalicchio
In: Edge-Cloud Computing and Federated-Split Learning in the Internet of Things, Future Internet, 2022.

Abstract

Federated Learning (FL) provides a promising solution for preserving privacy in learning shared models on distributed devices without sharing local data on a central server. However, most existing work shows that FL incurs high communication costs. To address this challenge, we propose a clustering-based federated solution, entitled Federated Learning via Clustering Optimization (FedCO), which optimizes model aggregation and reduces communication costs. In order to reduce the communication costs, we first divide the participating workers into groups based on the similarity of their model parameters and then select only one representative, the best performing worker, from each group to communicate with the central server. Then, in each successive round, we apply the Silhouette validation technique to check whether each representative is still made tight with its current cluster. If not, the representative is either moved into a more appropriate cluster or forms a cluster singleton. Finally, we use split optimization to update and improve the whole clustering solution. The updated clustering is used to select new cluster representatives. In that way, the proposed FedCO approach updates clusters by repeatedly evaluating and splitting clusters if doing so is necessary to improve the workers' partitioning. The potential of the proposed method is demonstrated on publicly available datasets and LEAF datasets under the IID and Non-IID data distribution settings. The experimental results indicate that our proposed FedCO approach is superior to the state-of-the-art FL approaches, i.e., FedAvg, FedProx, and CMFL, in reducing communication costs and achieving a better accuracy in both the IID and Non-IID cases.

1 Introduction

With recent advances in Internet of Things (IoT) devices and the fast growth of high-speed networks, the need to collect and process vast amounts of distributed data generated by these devices is significantly increasing. Furthermore, Artificial Intelligence (AI) has concurrently transformed the discovery of knowledge methods with cutting-edge success in several applications, including text prediction, facial recognition, natural language processing, document identification, and other tasks [1, 2]. However, those applications require IoT devices to send sensitive information to a remote cloud server for centralized model training, which raises data privacy concerns [3, 4]. These privacy concerns of IoT devices are supposed to be reduced by introducing an alternative setting, i.e., Federated Learning (FL). The main idea of FL is to collaboratively train a shared machine learning model across distributed devices, where the data are stored locally on devices [5, 6]. However, a naive implementation of the FL setting requires that each participant has to upload a full model update to a central server during each iteration. For large updates with millions of parameters for deep learning models and thousands of iterations [7], this step is likely to be a major hindrance in FL when the network bandwidth is limited. Thus, Federated Learning can become completely impractical [8].

Over the past few years, there has been a growing consensus that the more data that can be guaranteed, the better and higher accuracy that will be achieved. It should not be assumed, however, that blindly introducing more data into a model will improve its accuracy, but only that ensuring high-quality data will guarantee a higher degree of accuracy.

Our Contributions: In this paper, we propose a novel FL framework, entitled Federated Learning via Clustering Optimization (FedCO), to lessen the challenges described above during the training process. In particular, FedCO draws inspiration from our previous work, Cluster Analysis-Based Federated Learning (CA-FL), presented in [9]. In the CA-FL framework, the server only communicates with the representative who achieved a higher level of accuracy in each cluster.We implemented a regression model in machine learning and evaluated and compared the CA-FL model using only the federated average (FedAvg) [5] for human activity recognition (HAR) datasets. In the current work, we have enhanced the original CA-FL framework with a dynamic clustering scheme that reduces communication costs and more quickly ensures global model convergence. The result of the improvements is a new version of a deep learning-based framework called FedCO. In contrast to the original framework and compared to related work studies, discussed in Section 2, FedCO incorporates the following amendments.

- We propose a deep learning-based FL framework, FedCO for short, that employs a dynamic adaptation procedure to new data, which evaluates representatives tied to their clusters at each learning round and redistributes them among the clusters if necessary. In addition, the quality of the obtained adapted clus-

tering is evaluated at each round, and over-represented clusters of workers undergo a splitting procedure if this improves the whole clustering (Section 4).

- We provide a convergence analysis for our proposed FedCO algorithm (Section 6.2).
- We initially evaluate the proposed FedCO by comparing its performance with that of three baseline FL methods—FedAvg [5], FedProx [10], and CMFL [11]—on MNIST, CIFAR-10, and Fashion-MNIST under two different data-distribution scenarios, Independent and Identically Distributed (IID), and Non-IID.
- In addition, since our proposed FedCO algorithm is intended as a communication-mitigated version of FedAvg, we further study and assess the robustness of the FedCO with respect to FedAvg on two LEAF datasets under IID and Non-IID data.
- The conducted experiments have demonstrated the efficiency of FedCO over the FedAvg, FedProx, and CMFL algorithms in terms of convergence rate and communication overhead (Section 6).

The rest of the paper is structured as follows. Section 2 reviews the previous studies related to our work. The methodology used in our paper is presented in Section 3. Section 4 is devoted to the proposed FedCO and its strategy. The practical applications of those experimental settings are discussed in Section 5. The conducted experiments and the obtained results are analyzed and discussed in Section 6. The conclusions of our study and potential future works are presented in Section 7.

2 Related Work

This section mainly reviews the published research works aimed at reducing communication overheads in FL. In general, Federated Learning requires massive communication between the central server and the workers to train a global model [5]. Such an overhead is imputed to the size of the model exchanged and to the number of rounds to converge. Many works aim at reducing communication costs; e.g., HeteroFL [12] utilizes models of different sizes to address heterogeneous clients equipped with different computation and communication capabilities, while the work in [13] uses decentralized collaborative learning in combination with the master–slave model. Among many of the published FL solutions, there are few existing FL works that use clustering techniques [14–18]. For example, in [14] the study proposes clustering algorithms based on clients' similarities. The authors have tried to find a cluster structure of data to collect clients with similar data distributions and to perform baseline FedAvg training per cluster. In [15], the authors introduce clustering techniques to partition the clients with similar data distribution using a measure

of distance between the weight updates of the clients. A dynamic clustering through generative adversarial network-based clustering (GAN) is designed to obtain a partition of the data distributed on FL clients in [16]. The authors in [17] introduced a new framework, namely the Iterative Federated Clustering Algorithm (IFCA), in which clusters of users also aggregate their data with others in the same cluster (the same learning task) and optimize model parameters for the user clusters via gradient descent. Finally, Ouyang et al. [18] present clustering algorithms to cluster the heterogeneous data across clients into various clusters to participate in global model learning. The authors grouped the data after reducing its dimensions using PCA, and they measured the similarity of local updates. Although the studies discussed above [14–18] have applied clustering techniques to FL scenarios, all of them have clustered the clients based on the distribution of their own data, while our proposed technique partitions the clients based on their training model parameters, i.e., in a way that ensures that each cluster will contribute to the model by learning different aspects (different model parameters' values) of the studied phenomenon. Evidently, our solution for mitigating communication costs of FL is conceptually different from the approaches discussed above, despite it also being based on clustering. The majority of the studies in the field of resource-aware FL can be distributed into two main categories: a reduction in the total number of bits transferred, and a reduction in the number of local updates. Table 1 summarizes the techniques proposed by the research community, classifying them according to the categorization mentioned above.

Table 1: Summary of recent studies to minimize communication overhead in FL.

Categories	Existing Studies	ML Model Used	Datasets
First category	Chen et al. [19]	CNN, LSTM	MNIST, HAR
	Fed-Dropout [20]	DNN	CIFAR-10, MNIST, EMNIST
	Lin et al. [21]	CNNs, RNNs	Cifar10, ImageNet, Penn Treebank
	STC [22]	VGG11, CNN	CIFAR-10, MNIST
	PowerSGD [23]	ResNet-18, LSTM	CIFAR10, WIKITEXT-2
	FedOpt [24]	NN, LM	CIFAR10, MNIST
	FEDZIP [25]	CNN, VGG16	MNIST, EMNIST
	FetchSGD [26]	NN	CIFAR-100, CIFAR-10, FEMNIST
	T-FedAvg [27]	MLP, ResNet-18	MNIST, CIFAR-10
	FedAT [28]	CNN, Logistic	CIFAR-10, Fashion-MNIST, Sentiment140, FEMNIST, Reddit
Second category	CMFL [11]	CNN, LSTM	MNIST, NWP
	FedMed [29]	LSTM	PTB, WikiText-2, Yelp
	CEEP-FL [30]	CNN	MNIST, CIFAR-10
	FedCS [31]	NN	CIFAR-10, FashionMNIST
	FedPSO [32]	CNN	MNIST, CIFAR-10
	AdaFL [33]	MLP, CNN	MNIST, CIFAR-10
	MAB [34]	NN, CNN	MNIST, Video QoE
	FedAtt [35]	GRU	WikiText-2, PTB, Reddit
	FedPAQ [13]	CNN, Logistic	MNIST, CIFAR-10
	Ribero et al. [36]	CNN, Logistic, RNN	Synthetic, EMNIST, Shakespeare
	CA-FL [9]	SGD	mHealth, Pamap2
	Proposed (FedCO)	CNN	MNIST, Fashion-MNIST, CIFAR-10, FEMNIST, CelebA

2.1 Reduction of the Total Number of Bits

The first category incorporates works that reduce the total number of bits transferred for each local update through data compression. Chen et al. [19] propose an enhanced Federated Learning technique by introducing an asynchronous learning strategy on the clients and a temporally weighted aggregation of the local models on the server. Different layers of the deep neural networks are categorized into shallow and deep layers, and the parameters of the deep layers are updated less frequently than those of the shallow layers. In addition, a temporally weighted aggregation strategy is applied on the server to make use of the previously trained local models, thereby enhancing the accuracy and convergence of the central model. Caldas et al. [20] design two novel strategies to reduce communication costs. The first relies on lossy compression on the global model sent from the server to the client. The second strategy uses Federated Dropout, which allows users to efficiently train locally on smaller subsets of the global model and reduces client-to-server communication and local computation. Lin et al. [21] propose Deep Gradient Compression (DGC) to significantly reduce the communication bandwidth. Sattler et al. [22] introduce a new compression framework, entitled Sparse Ternary Compression, that is specifically designed to meet the requirements of the Federated Learning environment. Asad et al. [24] implement a Federated Optimization (FedOpt) approach by designing a novel compression algorithm, entitled Sparse Compression Algorithm (SCA), for efficient communication, and then they integrate the additively homomorphic encryption with differential privacy to prevent data from being leaked. Malekijoo et al. [25] develop a novel framework that significantly decreases the size of updates while transferring weights from the deep learning model between the clients and their servers. A novel algorithm, namely FetchSGD, that compresses model updates using a Count Sketch and takes advantage of the mergeability of sketches to combine model updates from many workers, is proposed in [26]. Xu et al. [27] present a federated trained ternary quantization (FTTQ) algorithm, which optimizes the quantized networks on the clients through a self-learning quantization factor. Vogel et al. [23] design a PowerSGD algorithm that computes a low-rank approximation of the gradient using a generalized power iteration. A novel Federated Learning method, entitled FedAT, with asynchronous tiers under Non-IID data, is presented in [28]. FedAT synergistically combines synchronous intra-tier training and asynchronous cross-tier training. By bridging the synchronous and asynchronous training through tiering, FedAT minimizes the straggler effect with improved convergence speed and test accuracy. Our research does not consider methods that leverage data compression techniques because of reduced scalability in scenarios such as edge and fog computing, and 5G networks, where hundreds of thousands of nodes cooperate in updating global models on the central server. Moreover, these approaches strictly depend on the application field.

2.2 Reduction of the Number of Local Updates

The second category includes studies that aim at reducing the number of local updates during the training process. For example, Wu et al. [29] have proposed a novel FedMed method with adaptive aggregation using the topK strategy to select the top workers who have lower losses to update the model parameters in each round. Likewise, Asad et al. [30] have provided a novel filtering procedure on each local update that allows transferring only the significant gradients to the server. The authors in [11] identify the relevant updates of the participants and upload only them to the server. In particular, at each round, the participants receive the global tendency and check the relevancy of their local updates with the global model, and only upload them if they align. Nishio and Yonetani in [31] propose an FL protocol of two-step client selection based on their resource constraints instead of the random client selection. In addition, a global model update algorithm, namely FedPSO, proposed transmitting the model weights only for the client that has provided the best score (such as accuracy or loss) to the cloud server [32].

Notice that our proposed FL model falls into the second category. We have been inspired by the studies discussed above, especially by CMFL [11] and FedProx [10], and we explored an approach that applies clustering optimization to bring efficiency and robustness in FL's communication. The most representative updates are uploaded only to the central server to reduce network communication costs.

The state-of-the-art solutions analyzed mainly conduct experiments considering a CNN model, except for FedMed, which uses an LSTM model, and FedCS, which uses an NN model (cf. Table 1, second category). Hence, we have chosen to assess the performance of our approach (FedCO) by using a CNN model. While there are many datasets used for the evaluation of FL solutions in the literature, the recurrent ones are MNIST, FashionMNIST, and CIFAR-10. Hence, we have evaluated the performance of FedCO training the FL model on the three datasets mentioned above. Additionally, we used datasets from the LEAF FL repository (FEMNIST and CelebA) to benchmark the performance of our FL algorithm against FedAvg [5] and FedProx [10].

3 Preliminaries and Definitions

In this section, we first briefly present the communication model and describe some preliminaries of a naive method of FL [37]. We then describe three state-of-the-art FL algorithms used for the comparison of our solution. Finally, we introduce the techniques used to conduct clustering optimization, i.e., the k-medoids clustering algorithm, and the Silhouette Index validation method. Table 1 summarizes the main notations used in the paper.

Table 2: Main notations.

Notation	Description
W	Set of available workers
W_t	Set of selected workers at tth communication round
w_i	A worker, i.e., $w_i \in W$
$\mathcal{D}_i$	The local data in worker w_i
n_i	The size of data in worker w_i
n	Total size of data
k_t	The number of clusters in round t
$C = \{C_1, \ldots, C_{k_t}\}$	The clustering solution in round t
$\mathcal{M}$	The global model
$\mathcal{M}^*$	The optimal global model
$\mathcal{M}_t$	The global model at tth round
$\mathcal{M}_t^i$	The local model of worker w_i at round t
$F(.)$	The objective function of the global model
$F_i(.)$	The objective function of the local model of worker w_i
T	Maximal number of communication rounds
E	The number of local epochs
η	Learning rate
g_t^i	The gradients computed using back-propagation
$s(.)$	Silhouette Index score

3.1 Communication Model

In the proposed FL environment, FL is split into two major parts: workers and the central server. Our work aims to reduce communication overhead without sacrificing accuracy value during the training process. In this setting, the server coordinates a network of workers, controls the training progress of the model, broadcasts the original model to all participating workers, and then executes all the aggregation processes of the model updates. All workers are share model updates instead of sending their private data to a central server for global model aggregation. Figure 1 outlines the overall operations of the Federated Learning procedure. Data are protected, with private access for each worker. Thus, model training occurs locally on each worker's

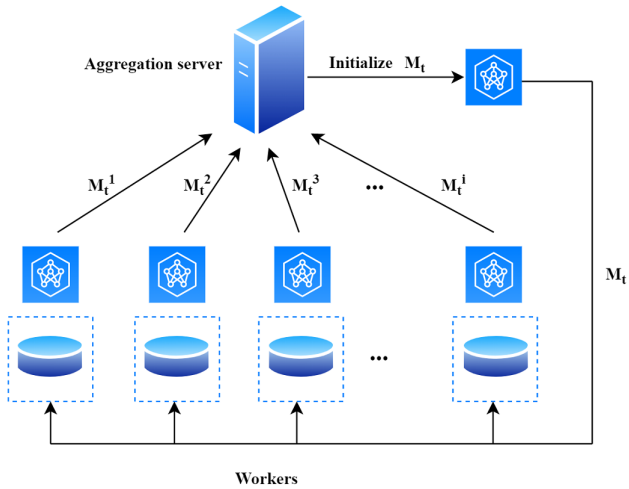


Figure 1: The general operations of the Federated Learning process.

side. In this context, we assume that each worker agrees on the same learning task and the model parameters throughout the training process. In particular, the proposed FL model updates the global model only with local model parameters from a few workers that are considered representative. Such workers are selected at each training round by identifying the highest quality of the local model produced of the worker. The selection policy is assumed to be implemented in a server, i.e., a central node selects a representative of the cluster with the highest accuracy. Furthermore, we assume that the server is always reachable by the workers. Finally, our proposed technique works by following this iterative collaboration between the central server and the workers.

3.2 Problem Description

In this work, we mainly concentrate on synchronous Federated Learning algorithms. A Federated Learning system consists of a global model $\mathcal{M}$ and a set of workers W. At each communication round t, the server deploys the current model $\mathcal{M}_t$ to a subset of workers $W_t \subset W$ that dynamically participate in the global aggregation at round t. Each worker $w_i \in W_t$ locally keeps its personal data $\mathcal{D}_i = \{x_{ij}\}_{j=1}^{n_i}$, $(j = 1, 2, \ldots, n_i)$, where x_{ij} is the jth training sample in $\mathcal{D}_i$. The size of the local dataset $\mathcal{D}_i$ varies with different real-world applications.

In standard centralized Stochastic Gradient Descent (SGD), the local updates of each w_i are calculated according to Equation (1) to optimize $\mathcal{M}_t^i$, where η is the learning rate and g_t^i refers to the gradients computed:

$$\mathcal{M}_{t+1}^i = \mathcal{M}_t^i - \eta g_t^i. \tag{1}$$

Then, each worker w_i sends the local model changes $\mathcal{M}_{t+1}^i$ to the central server

after the number of E local step, where p_i is the relative weight of worker w_i, and the global model is computed by applying Equation (3):

$$\mathcal{M}_{t+1} = \mathcal{M}_t + \frac{\sum\limits_{w_i \in W_t} p_i \mathcal{M}_{t+1}^i}{\sum\limits_{w_i \in W_t} p_i}. \tag{2}$$

These are iterated until a certain stop criterion is met.

The corresponding local loss function of $\mathcal{M}^i$ of each worker w_i is defined as

$$F_i(\mathcal{M}^i) = \frac{1}{|\mathcal{D}_i|} \sum_{x_{ij} \in \mathcal{D}_i} f(\mathcal{M}^i, x_{ij}), \tag{3}$$

where $f(\mathcal{M}^i, x_{ij})$ is the loss function for data point x_{ij} using 1. Each worker w_i independently updates the model over its own data $\mathcal{D}_i$ to optimize its local loss function $F_i(\mathcal{M}^i)$. The aim of improving the communication efficiency of Federated Learning is to minimize the cost of sending $\mathcal{M}_t^i$ to a central server while learning from the data distributed over a large number of decentralized edge devices. Similarly, the global loss function on all the distributed datasets is defined as:

$$F(\mathcal{M}) = \frac{1}{\mid W \mid} \sum_{w_i \in W_t} F_i(\mathcal{M}^i), \tag{4}$$

where $\mathcal{M}$ is the aggregated global model, and the overall goal is to decrease the global loss function $F(\mathcal{M})$, namely,

$$\mathcal{M}^* = \arg\min F(\mathcal{M}). \tag{5}$$

Other issues related to Federated Learning problems, such as system heterogeneity or privacy, are beyond the scope of this paper. Specifically, the proposed FedCO algorithm does not account for heterogeneity, which for example could affect the selection of workers that have enough power to transmit the model parameters. In the worst case, heterogeneity could increase the convergence time or reduce the accuracy, if for example, workers that achieve a higher accuracy cannot be selected because they have short battery lifetimes.

3.3 FL State-of-the-Art Algorithms

Most of the work on the convergence of compared FL algorithms such as FedAvg, CMFL, and FedProx centers around minimizing (4). We compare the proposed FedCO with the following state-of-the-art algorithms in the FL setting:

3.3.1 *FedAvg*

FedAvg, proposed by McMahan et al. in [5] can be viewed as a communication-light implementation of the standard centralized SGD, wherein the local updates are aggregated in the server after E local steps, where $E \geq 1$.

3.3.2 *FedProx*

FedProx [10] is a distributed algorithm, wherein a round-varying proximal term is introduced to control the deviation of the local updates from the most recent global model. A participating worker uses a proximal update that involves solving a minimization problem.

3.3.3 *CMFL*

Communication-Mitigated Federated Learning (CMFL) [11] improves the communication efficiency of Federated Learning while at the same time providing guaranteed learning convergence.

3.4 K-Medoids Clustering Algorithm

K-medoids is a robust clustering algorithm. It is used to partition a given set of data points into k disjoint clusters [38]. In contrast to the k-means, which use the mean value of the data points in each cluster as a cluster centroid, k-medoids chooses an actual data point, called a medoid. The medoid is the most centrally located point in a given cluster. Therefore, k-medoids are more robust to outliers and noise than other points. The algorithm works by arbitrarily choosing a set of k initial cluster medoids from a given set of data points, where k is preliminarily specified. Then, each data point is assigned to the cluster whose center is the nearest, and the cluster centers (medoids) are recomputed. This process is repeated until the points inside every cluster become as close to the center as possible, and no further item reassignments take place.

In our FedCO algorithm, we use k-medoids for partitioning the available workers into groups of similar workers with respect to their local updates. Furthermore, 2-medoids are used in the iteration phase of the algorithm for conducting cluster splitting.

3.5 Silhouette Index

The Silhouette Index (SI) is a widely used internal cluster validation technique, introduced in [39]. SI can be used to judge the quality of any clustering solution $C = \{C_1, C_2, \ldots, C_k\}$. It assesses the separation and compactness between the clusters. Suppose that a_i represents the average distance of item i from all the other items in the cluster to which item i is assigned, and b_i represents the minimum of the

average distances of item i from the items of the other clusters. Then, the Silhouette score $s(i)$ of item i can be calculated as

$$s(i) = (b_i - a_i)/\max\{a_i, b_i\}\cdot \tag{6}$$

$s(i)$ measures how well item i matches the clustering at hand. $s(i) \in [-1, 1]$, and if $s(i)$ is close to 1, this means that item i is assigned to a very appropriate cluster. The situation is different when $s(i)$ is near zero. Specifically, item i lies between two clusters. The worst case is when $s(i)$ is close to -1. Evidently, this item has been misclassified.

In addition, the overall Silhouette score for the whole clustering solution C of n items is determined as

$$s(C) = \frac{1}{n}\sum_{i=1}^{n}\frac{(b_i - a_i)}{\max\{a_i, b_i\}}\cdot \tag{7}$$

The SI can also be calculated for each cluster C_j ($j = 1, 2, \ldots, k$) of n_j objects as follows:

$$s(C_j) = \frac{1}{n_j}\sum_{i=1}^{n_j} s(i)\cdot \tag{8}$$

The FedCO algorithm proposed in this study uses the Silhouette Index at each iteration round for assessing the current workers' partitioning and, based on this assessment, selects what optimizing actions to conduct. For example, we used SI to check whether a representative is still firmly tied to its current cluster of workers. It may happen that some representatives will change their clusters. If we have a worker that produces a negative SI value for all clusters, this means that this worker cannot be assigned to any of the existing clusters, and it will form a new singleton cluster; i.e., a new concept appears. In addition, SI is applied to assess whether an intended splitting of a cluster will improve the quality of the whole clustering solution, i.e., whether it should be conducted. For more details, see Section 4. Note also that in the implemented version of our FedCO algorithm, we use Euclidean distance to measure the similarity between each pair of workers. In particular, the Euclidean distance between the worker (the representative) and the cluster centers (medoids) has been computed.

4 Proposed Approach

Our proposed FedCO algorithm foresees two distinctive phases: *initialization* and *iteration*. These phases are described in what follows, along with cluster optimization algorithms. In addition, the algorithm pseudo-code is reported in Algorithms 1 and 5. Let $W = \{w_1, w_2, \ldots, w_n\}$ be the set of all available workers, and W_t is a subset

of W that contains the workers selected at round t. The workers in W_t can be the representatives of the clusters $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk_t}\}$ obtained by applying a clustering algorithm to W, or a set of randomly selected workers, and $\mid W_t \mid < n$.

Algorithm 1 Federated Learning Using Clustering Optimization (FedCO)

Output: The FEDCO procedure updates the global model $\mathcal{M}_t$ for T iterations

1: **procedure** FEDCO($\mathcal{M}_0$, $W_t \subseteq W$, k_t, T)
Initialization Phase
2: $t \leftarrow 0$
3: $\forall\, w_i \in W_t$, SEND(w_i, $\mathcal{M}_t$)
4: **for** each worker $w_i \in W_t \;\; in \;\; parallel$ **do**
5: $\mathcal{M}^i_{t+1} \leftarrow$ WORKERUPDATE($i, \mathcal{M}_t$)
6: **end for**
7: $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$ following 3
8: $C_t \leftarrow$ KMEDOIDS($k_t, \{\mathcal{M}^i_{t+1} \mid w_i \in W_t\}, W_t$)
Iteration Phase
9: **while** $t \leq T$ **do**
10: $t \leftarrow t + 1$
11: $W_t \leftarrow$ SELECTTOPRANKED(p, C_t)
12: $\forall\, w_i \in W_t$, SEND(w_i, $\mathcal{M}_t$)
13: **for** each worker $w_i \in W_t \;\; in \;\; parallel$ **do**
14: $\mathcal{M}^i_{t+1} \leftarrow$ WORKERUPDATE($w_i, \mathcal{M}_t$)
15: **end for**
16: $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$
17: $C_{t+1} \leftarrow$ SILHOUETTE(k_t, C_t, W_t)
18: **while** $\mid C_{t+1} \mid < \mid C_t \mid$ **do**
19: $C_{t+1} \leftarrow$ CLUSTERINGOPTIMIZATION(k_{t+1}, C_{t+1})
20: **end while**
21: **end while**
22: **end procedure**

Algorithm 1 Federated Learning Using Clustering Optimization (FedCO) (continued)

function SILHOUETTE((k_t, C_t, W_t)) ▷ *C*heck whether each cluster representative still belongs to its cluster
  for $w_i \in W_t$ **do**
    for $j = 1, 2, \ldots, k$ **do**
      compute $s(w_i)$ ▷ *A*ccording to Equation (8)
    end for
    if $s(w_i) < 0, \forall j \in \{1, 2, \ldots, k\}$ **then**
      $k_t \leftarrow k_t + 1$
      $C_{tk_t} \leftarrow w_i$
    else
      Assign w_i to the nearest cluster C_{tj}
    end if
  end for
  $\forall\, C_{tj}\ (j = 1, 2, \ldots, k)$ recompute the cluster center
  return C_{t+1} ▷ *T*he new set of clusters
end function

function WORKERUPDATE($(w_i, \mathcal{M}_t)$) ▷ *L*ocal update
  while True **do**
    RECEIVE(w_i, $\mathcal{M}_t$)
    LOCALTRAINING(w_i, $\mathcal{M}_t$)
    $\mathcal{M}^i_{t+1} \leftarrow \mathcal{M}^i_t - \eta g^i_t$ ▷ *L*ocal update, (1)
    SEND(i, $\mathcal{M}^i_{t+1}$)
  end while
end function

Algorithm 2 ClusteringOptimization

Output: updated k_t and C_t

procedure ClusteringOptimization(k_t, C_t)
  $s(C_t) \leftarrow$ SilhouetteScore(k_t, C_t, W_t)
  $C'_t \leftarrow \emptyset$
  for $C_{tj} \in C_t \; s.t. \mid C_{tj} \mid > 1$ **do**
    $s(C_{tj}) \leftarrow$ SilhouetteCluster(C_{tj}, W_t)
    while $s(C_{tj}) < 0$ **do**
      $(C^1_{tj}, C^2_{tj}) \leftarrow$ Kmedoids$(\{\mathcal{M}^i_{t+1} \mid w_i \in C_{tj}\}, k = 2)$ ▷ *r*un 2-medoids to generate two new clusters
      $\bar{C}_t \leftarrow \{C_t \setminus \{C_{tj}\}\} \cup \{C^1_{tj}, C^2_{tj}\}$
      $s(\bar{C}_t) \leftarrow$ SilhouetteScore(k_t, C_t, W_t)
      if $s(\bar{C}_t) > s(C_t)$ **then**
        $C'_t \leftarrow C'_t \cup \bar{C}_t$
        $k_t \leftarrow k_t + 1$
      end if
    end while
  end for
  return (C'_t, k_t)
end procedure

function SilhouetteScore(k_t, C_t, W_t) ▷ *S*ilhouette score of whole cluster solution C_t
  Compute $s(w_i)$ between each $w_i \in W_t$ and each medoid $c_{tj} \in C_{tj}$ $(j = 1, 2, \ldots, k_t)$ according to (8)
  Compute the average Silhouette score over all representatives $w_i \in W_t$ according to (7)
  return $s(C_t)$
end function

function SilhouetteCluster(C_{tj}, W_t) ▷ *S*ilhouette Score of cluster $C_{tj} \in C_t$ $(j = 1, 2, \ldots, k)$
  Calculate Silhouette score $s(w_i)$ for each $w_i \in C_{tj}$ according to (8)
  Compute the mean over Silhouette scores of all cluster members $\{s(w_i) \mid w_i \in C_{tj}\}$ according to (8)
  return $s(C_{tj})$
end function

4.1 Initialization Phase

1. At time $t = 0$, the Server initializes the inputs for the FedCO algorithm (Algorithm 1). These are the model $\mathcal{M}_0$, the set of representative workers W_t, the number of clusters k_t, and the number of iterations T. $t = 0$ (line 1 in Algorithm 1).

2. A central Server transmits the initial global model $\mathcal{M}_t$ to a set of workers W_t ($W_t \subset W$). These are selected to be used for initial training in round $t = 0$ of Federated Learning (lines 3 in Algorithm 1).

3. Each worker $w_i \in W_t$ receives the global model $\mathcal{M}_t$ and optimizes its parameters locally; i.e., the $\mathcal{M}_t^i$ initial update is produced and sent back to the Server (Equation (1)) (lines 4–6 and lines 38–45 in Algorithm 1).

4. The Server aggregates the parameters $\{\mathcal{M}_t^i \mid w_i \in W_t\}$ uploaded by the selected workers W_t to update the global model $\mathcal{M}_t$ through the FedAvg algorithm (Equation (3)) (line 7 in Algorithm 1).

5. The local updates $\{\mathcal{M}_t^i \mid w_i \in W_t\}$ of the workers in W_t are analyzed by using the k-medoids clustering algorithm (function KMEDOIDS, line 8 in Algorithm 1)). As a result, k_t clusters of workers with similar updates are obtained; i.e., an initial clustering $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk_t}\}$ of the workers in W_t is produced.

4.2 Iteration Phase

1. At each iteration round t ($t \geq 0$), the Server evaluates each local update $\mathcal{M}_t^i$, $w_i \in W_t$ by using an evaluation measure that is suitable for the task under consideration. It ranks the workers in each cluster $C_{tj}, j = 1, 2, \ldots, k_t$ with respect to their evaluation scores and selects the top-ranked worker, i.e., the representative (function SELECTTOPRANKED, line 11 in Algorithm 1). The selected representatives form a new set of workers W_{t+1}, where $\mid W_{t+1} \mid = k_t$ and $k_t << \mid W_0 \mid$. Each selected worker $w_i \in W_{t+1}$ will check in with the Server.

2. The Server sends the global model $\mathcal{M}_t$ to each representative $w_i \in W_{t+1}$ (line 12 in Algorithm 1).

3. Each representative $w_i \in W_{t+1}$ receives the global model $\mathcal{M}_t$ and optimizes its parameters locally; i.e., the $\mathcal{M}_{t+1}^i$ update is produced (Equation 1) and sent back to the Server (lines 13 and 15 in Algorithm 1).

4. The Server aggregates the received local models $\{\mathcal{M}_{t+1}^i \mid w_i \in W_{t+1}\}$ uploaded by the representatives to update the global model through the FedAvg

algorithm; i.e., an updated global model $\mathcal{M}_{t+1}$ is produced (Equation (3)) (line 16 in Algorithm 1).

5. The Server adapts C_t to the newly arrived local updates by conducting the following operations:

    a) SI invokes the SILHOUTTE function (lines 17, 23–37 in Algorithm 1), which assesses whether each representative $w_i \in W_{t+1}$ is still adequately tight with its current cluster (Equation (8)). The updated clustering C_{t+1} of W_t is produced, and the clusters in C_{t+1} may contains a set of workers different from C_t. Note that $k_{(t+1)} \geq k_t$, where $k_{(t+1)} =\mid C_{t+1} \mid$, since new singleton clusters may appear due to the updating operation. This happens when the Silhouette coefficient $s(w_i)$ of a representative for all clusters gives a negative value (lines 28–30 in Algorithm 1), which means that this representative cannot be assigned to any existing cluster. Hence, this representative could be considered as a new cluster with a single item (singleton).

    b) If there is a cluster $C_{(t+1)j} \in C_{t+1}$, such that $C_{(t+1)j} = \emptyset$, then $C_{t+1} = C_{t+1} \setminus \{C_{(t+1)j}\}$, and therefore, $\mid C_{t+1} \mid < \mid C_t \mid$. This condition/event triggers the optimization of the number of clusters by invoking the CLUSTEROPTIMIZATION function (lines 18-20 in Algorithm 1). This operation is repeated for each empty cluster of C_{t+1}.

A schematic illustration (flowchart) of the overall processes of the proposed FedCO algorithm is given in Figure 1.

4.3 Cluster Optimization

The CLUSTEROPTIMIZATION algorithm works in what follows (cf. Algorithm 5):

1. Firstly, the SI score of the whole clustering solution C_{t+1} is computed. This score is used to check whether the splitting operation really improves the quality of the clustering solution (line 2 in Algorithm 5).

2. Then, the SI score is calculated for each cluster $C_{(t+1)j} \in C_{t+1}$, such that $\mid C_{(t+1)j} \mid > 1$ using Equation (8). If $s(C_{(t+1)j}) < 0$, then this cluster is a candidate to be split into two clusters, and the following operations are performed (lines 4–6 in Algorithm 5):

    a) The two most distant points in the cluster $C_{(t+1)j}$ are found. They are used to seed 2-medoids clustering, which is applied to split the cluster $C_{(t+1)j}$ into two clusters (function KMEDOIDS at line 7 in Algorithm 5).

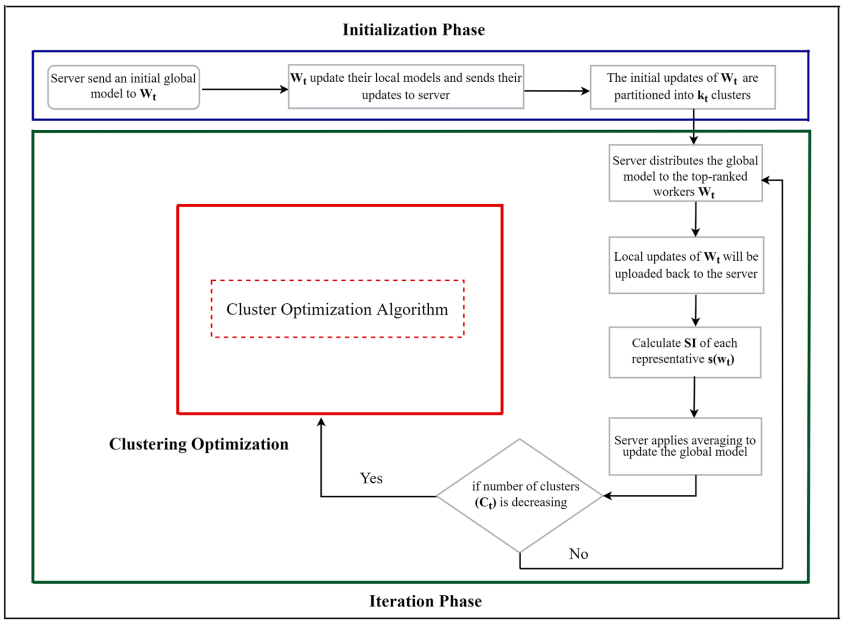


Figure 2: A schematic illustration of the entire process of the FedCO algorithm in two global communication rounds: Initialization Phase and Iteration Phase.

b) The clustering solution C_{t+1} is updated by replacing cluster $C_{(t+1)j}$ with the two clusters obtained due to the splitting operation (line 8 in Algorithm 5), and stored in the set $\bar{C}_{t+1}$.

c) The SI score of the updated clustering solution $\bar{C}_{t+1}$ is computed (line 9 in Algorithm 5 (7)).

d) If the SI score of the new clustering solution $\bar{C}_{t+1}$ is higher than the one before splitting, the new clustering solution is adopted and stored in the set C'_{t+1}; otherwise, the clustering solution C_{t+1} is kept (lines 10–13 in Algorithm 5).

Steps 1–5 of the *iteration* phase are repeated until a certain number of training rounds T is reached. Figure 3 shows a flowchart depicting the cluster optimization algorithm in a single round of communication. The proposed FedCO implementation, at each training round, always selects the top performing representative; i.e., the size of the clusters is not reflected in the aggregated global model, and the size of the cluster does not impact the selection/importance of the representative. The FedCO design, however, allows from each cluster the selection of several top-ranked representatives, i.e., more than one, proportionally to the cluster size. In that way, the bigger clusters will have more weight in the building of the global model. It is also possible to assign explicit weights to the clusters representing their relative importance, and calculated based on their size. Our future plans include the investigation

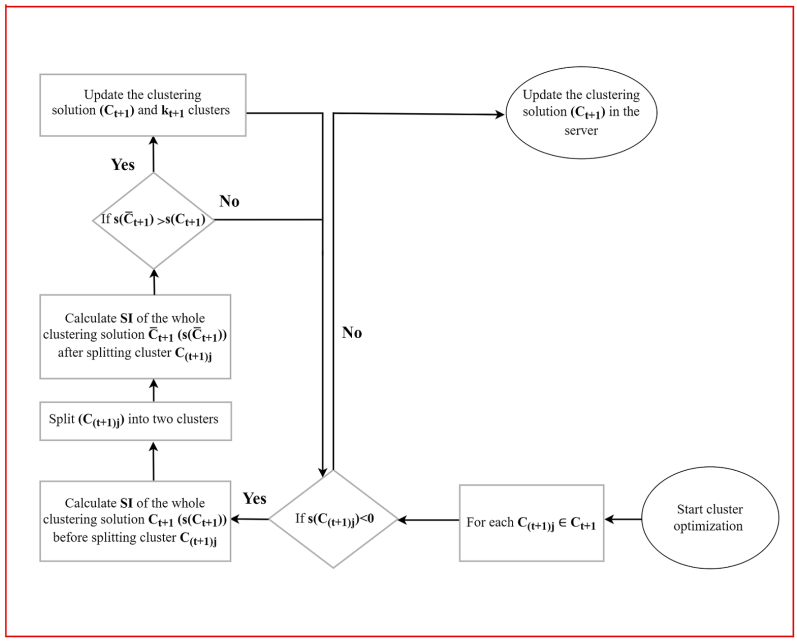


Figure 3: Flowchart depicting Cluster optimization algorithm.

of an optimized version of the FedCO algorithm where the importance of clusters will be considered in the aggregated model.

5 Datasets and Experimental Setup

This section describes the datasets, the distribution of the data across the edge nodes, the model selected and related parameters, and the performance metrics used for evaluating FedCO.

5.1 Datasets

We conducted experiments using a wide range of datasets. Firstly, we selected three benchmark datasets widely used for image classification: MNIST [40], Fashion MNIST [41], and CIFAR-10 [42].

- The MNIST dataset contains a 60,000-point training set and a 10,000 point test set with 10 classes. Each sample is based on a grayscale image of handwritten digits with a size of 28×28 pixels.

- The Fashion MNIST dataset comprises a 60,000-point training set and a 10,000 point testing set of images of fashion items with 10 different classes. Each

image has dimensions of 28×28 in grayscale.

- The CIFAR-10 dataset consists of a 50,000-point training set and a 10,000-point testing set with images of objects from frogs to planes, where each image is 32×32 pixels in 10 classes.

Secondly, we considered two LEAF datasets [43], an open-source benchmark for Federated Learning.

- FEMNIST for 62-class image classification, which serves as a more complex version of the popular MNIST dataset [44].

- CelebA for determining whether the celebrity in the image is smiling, which is based on the Large-scale CelebFaces Attributes Dataset.

5.2 Data Distribution

In an FL context, the performance is affected by the distribution of the training data stored on the various workers. Interestingly, unlike other FL studies using clustering techniques, different degrees of non-IID data do not affect the clustering results, as FedCO clustering occurs based on the model parameters and not on the data themselves. In order to assess the impact of different data distribution scenarios, we generated two experimental datasets for each dataset introduced above:

- The IID dataset: Each worker holds the local data equal in size and label distribution.

- The Non-IID dataset: Each worker holds different data distributions in size and label distribution compared to the global dataset.

5.3 Model Selection and Parameters

We have compared the proposed FedCO algorithm against the FedAvg, CMFL, and FedProx algorithms using the Convolutional Neural Network (CNN) classifier as a training model. The CNN model we used consists of two 5×5 convolution layers with a ReLU activation and a final softmax output layer.

The baseline configuration parameters' values listed in Table 2 are shared among the four compared algorithms.

Table 3: Hyper-parameter configuration.

Hyper-Parameter	Value
Workers	100
Optimizer	SGD
Classes	10
Batch Size	50
Learning rate	0.15
Local epochs	10
Global rounds	200
Clusters	8
Non-IID degree	0.5

5.4 Performance Metrics

FL typically relies on a large number of edge devices, sometimes in the magnitude of millions, and due to the limited computing capabilities of those devices, decreasing the communication rounds or communication overhead is crucial during the training process. Hence, the performance metrics selected are the *Number of Communication Rounds*, the *Communication Overhead*, and the model *Accuracy*. The *Communication Overhead* is defined in [9] as

$$(N\times \mid W_s \mid) \times (2 \times T + 1),$$

where N is the size of the trained model in bytes, $\mid W_s \mid$ is the number of selected workers, and T is the total number of training rounds. We assume the size of the model updates to be fixed. However, other communication costs are negligible.

It is worth mentioning that the total communication overhead of FedCO can be calculated as the summation of the communication costs of the initialization stage and the iteration stage together.

6 FedCO Performance Evaluation and Analysis

In this section, we first study the clustering optimization scheme used for the dynamic adaptation of partitioning of workers' updates at each communication round. This adaptive behavior contributes to achieving robust communication in FL. The performance of the proposed FedCO is then evaluated and compared to three other existing FL approaches (FedAvg, FedProx, and CMFL) in terms of accuracy, communication rounds, and communication overhead.

Our proposed FedCO algorithm is a communication-optimized version of FedAvg. Therefore, we further evaluate these two algorithms by benchmarking them

on two datasets from the LEAF Federated Learning repository, namely FEMNIST and CelebA. In addition, we further study our FedCO algorithm for two different scenarios for selecting cluster representatives: a performance threshold-based worker selection versus the single (top-performer) cluster representative selection, explained in Algorithm 1.

6.1 Clustering Optimization Behavior

Our clustering optimization algorithm assesses the local updates of clusters' representatives at each communication round, and as a result, it assigns some workers to different clusters. An output of this cluster-updating procedure is that clusters may appear or disappear. Our solution is capable of catching and handling these scenarios. In addition, it implements a splitting procedure that performs a further fine calibration of the clustering for the newly uploaded updates.

In order to illustrate the properties of the clustering optimization scheme discussed above, we show in Figure 4 the clustering updates in the first five global communication rounds of the FedCO algorithm applied to the Non-IID FashionMNIST dataset. In the example, in round 2, cluster 5 has disappeared and cluster 3 is a singleton, i.e., it cannot be a candidate for splitting. Almost all of the remaining clusters (except cluster 6) have negative SI scores. The remaining clusters (0, 1, 2, 4, and 7) have been split into two new clusters and their cluster labels are replaced. Interestingly, in round 3, the unique number of clusters is 17. However, in round 4, five clusters have turned out empty and have disappeared (1, 4, 10, 13, and 16). Furthermore, eight clusters have positive SI scores (0, 2, 5, 6, 9, 12, 14, and 15), while four have negative SI scores (3, 7, 8, and 11). The algorithm did not split the clusters 7, 8, and 11 because this did not improve the quality of the clustering solution; i.e., it did not increase its SI score. Cluster 3 is still a singleton. The worker belonging to this cluster may be considered as one that provides unique model parameters due to its training data.

Consequently, in round 5, we have only 12 clusters. Two clusters disappeared (2 and 14), and four new clusters appeared (1, 6, 12, and 17), while clusters 3 and 9 were singletons.

The cluster optimizations discussed above will continue in the same fashion for the upcoming communication rounds. The workers' partitioning is dynamically adapted at each communication round to reflect the new local updates of the representatives.

6.2 Convergence Analysis

In this section, we provide a convergence analysis of the proposed FedCO algorithm and theoretically show that it ensures a faster convergence than the baseline FedAvg

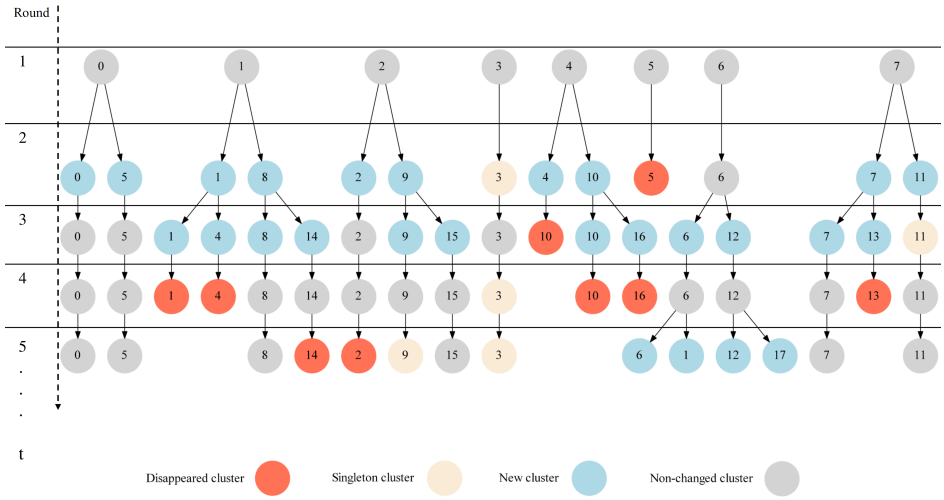


Figure 4: The clustering updates in the first five global communication rounds of the proposed FedCO algorithm applied on the Non-IID FashionMNIST dataset. Notice that the number in the circle represents the cluster label.

algorithm.

Our analysis is based on two assumptions. The first one supposes that the data are non-IID. Secondly, we assume that there is a partial involvement of workers; this strategy is much more realistic as it does not require all of the worker output. Therefore, at each iteration, we can calculate the global update by aggregating the local updates by using those cluster representatives, which have reached a high accuracy level at this iteration phase. Two scenarios are considered to this end: (i) a global model is trained by FedAvg based on updates made by randomly selected workers, regardless of their accuracy value; (ii) a global model is trained by applying FedCO, and in that way, at each training round, only workers (cluster representatives) that have achieved the highest accuracy values are used. Let us briefly summarize the working mechanism of the proposed FedCO algorithm. In the tth global training iteration, each worker involved ($w_i \in W_t$) calculates the average g_t^i gradients using the optimization algorithms in the local dataset in the current global model $\mathcal{M}_t$. Note that according to Equation (5), high-quality data and a high accuracy of the workers' models can lead to a faster convergence of the local loss functions (Equation (1)) and the global loss function (Equation (4)) [45]. Both the local model update $\mathcal{M}_t^i$ of the worker in Equation (1) and the shared global model update $\mathcal{M}_{t+1}$ in Equation (5) can be more quick to converge to the target value with fewer iterations. Consequently, the training time of a worker in a global iteration is decreasing. Therefore, highly accurate workers' models can significantly improve the learning efficiency of Federated Learning; e.g., it can ensure less training time [31, 46]. This process is iterative until a global accuracy ϵ ($0 \leq \epsilon \leq 1$) is achieved. Specifically, each update of the local model has a local accuracy w_i^ϵ that corresponds to the local quality of the worker

w_i data. A higher local accuracy leads to fewer local and global iterations [46, 47]. FedCO uses an iterative approach that requires a series of communication rounds to achieve a level of global accuracy ϵ. Server and representative communications occur during each global round of the iteration phase. Specifically, each representative minimizes its objective $F_i(\mathcal{M}^i)$ in Equation (1) using the local training data n_i. Minimizing $F(\mathcal{M})$ in Equation (4) also requires multiple local iterations up to a target accuracy. Then, the global rounds will be bounded as follows:

$$\frac{\mathcal{O}(\log(\frac{1}{\epsilon}))}{1 - w_i^{\epsilon}}$$

Thus, the global rounds are affected by both the global accuracy ϵ and the local accuracy w_i^{ϵ}. When ϵ and w_i^{ϵ} are high, FedCO needs to run a few global rounds. On the other hand, each global round consists of both computation and transmission time. Our primary motivation in this work is to consider the communication overhead, discussed and analyzed in detail in Section 6.3. The computation time (w_i^{cmp}), however, depends on the number of local iterations. When the global accuracy ϵ is fixed, the computation time is bound by $\log(\frac{1}{w_i^{\epsilon}})$ for an iterative algorithm to solve Equation 1; here, (SGD) is used [46]. Therefore, the total time of one global communication round for a set of representatives is denoted as

$$T^{com} = \sum_{w_i \in W_t} \log(\frac{1}{w_i^{\epsilon}}) w_i^{cmp} + w_i^{com},$$

where w_i^{com} represents the transmission time of a local model update. As a result, a high local accuracy value of w_i^{ϵ} leads to fewer local iterations w_i^{cmp} and eventually to lower global communication rounds T^{com}. Unlike FedCO's convergence rate, FedAvg does not necessarily guarantee a faster convergence speed. This is because FedAvg uses a much larger number of workers compared to the FedCO model. Therefore, if there are more workers with poor data quality, the convergence will be reached at a slower rate than when much fewer workers with high data quality are used. However, at each global round, FedCO may have selected a different set of workers. Those, however, are not selected randomly, but each one is a representative of a cluster of workers having modeled similar parameters, and in addition, it achieves the highest accuracy among the cluster members. Let T_{FedAvg} and T_{FedCO} represent the number of global rounds for which convergence has been reached by FedAvg and FedCO, respectively. Then, Tables 4 and 5 demonstrate that the inequality $T_{FedCO} < T_{FedAvg}$ is valid in the experiments aiming to reach the same accuracy using the two algorithms.

Table 4: The number of communication rounds to reach a target accuracy for the three compared FL algorithms.

	IID					
	MNIST		**FashionMNIST**		**CIFAR-10**	
	Rounds	**Saving**	**Rounds**	**Saving**	**Rounds**	**Saving**
FedAvg	190	(ref)	200	(ref)	200	(ref)
FedProx	185	26%	190	5%	188	6%
CMFL	50	73%	60	70%	80	60%
FedCO	25	86%	50	75%	30	85%
	Non-IID					
	MNIST		**FashionMNIST**		**CIFAR-10**	
	Rounds	**Saving**	**Rounds**	**Saving**	**Rounds**	**Saving**
FedAvg	170	(ref)	200	(ref)	200	(ref)
FedProx	167	17%	186	7%	>200	-
CMFL	60	64%	150	25%	160	20%
FedCO	30	82%	70	65%	60	70%

Table 5: The number of communication rounds to reach a certain accuracy level for the two compared FL algorithms on each LEAF dataset.

	IID			
	FEMNIST		CelebA	
	Rounds	Saving	Rounds	Saving
FedAvg	140	(ref)	110	(ref)
FedCO	12	91%	30	72%
	Non-IID			
	FEMNIST		CelebA	
	Rounds	Saving	Rounds	Saving
FedAvg	100	(ref)	150	(ref)
FedCO	14	86%	10	93%

6.3 Communication Rounds versus Accuracy

In this subsection, we present the results related to the evaluation of the accuracy of our distributed deep learning (DL) model. Figures 5–7 show how the compared FL (FedAvg, FedProx, CMFL, and FedCO) algorithms perform in terms of Accuracy versus the Number of Communication Rounds. For the MNIST dataset (see Figure 5), we can observe that the FedCO algorithm converges faster than with the state-of-the-

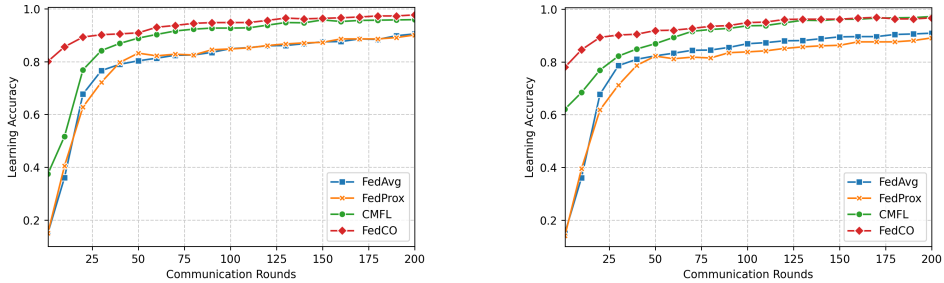


Figure 5: Learning accuracy versus the number of communication rounds for MNIST data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in the case of the Non-IID data distribution scenario. (**a**) IID; (**b**) Non-IID.

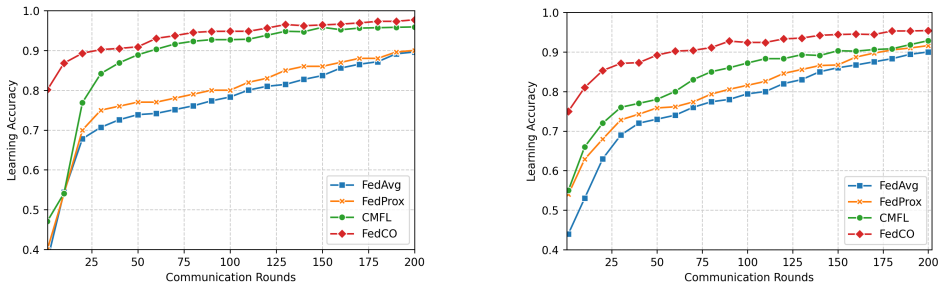


Figure 6: Learning accuracy versus communication rounds for FashionMNIST data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in the case of the Non-IID data distribution scenario. (**a**) IID; (**b**) Non-IID.

art approaches. As is shown in Figure 5a (IID data distribution setting), FedAvg and FedProx use 100 rounds to obtain an accuracy of 85%. The CMFL reaches the same accuracy in 30 rounds, while our FedCO algorithm achieves this result with only 10 rounds. Furthermore, in Figure 5b, FedCO dramatically decreases the communication rounds with respect to FedAvg, FedProx, and CMFL. Indeed, in Non-IID data, a learning accuracy of 90% is achieved by FedCO in 40 rounds, FedAvg has conducted 160 rounds, FedProx requires 200 rounds, and CMFL needs 60.

In Figure 6a, we compare the accuracy of the four FL approaches in the case of the IID data distribution scenario of FashionMNIST. The FedCO outperforms FedAvg, FedProx, and CMFL in this experimental setting. Within 25 communication rounds, CMFL, FedAvg, and FedProx reach 81%, 69%, and 74% accuracy, respectively, while our FedCO algorithm achieves an accuracy of 90% with the same number of communication rounds. Notice that under the Non-IID data distribution setting, our FedCO algorithm outperforms the other, reaching an accuracy of nearly 79% with only 11 rounds; this costs 100 communication rounds for FedAvg and 80 rounds for FedProx. CMFL considerably minimizes this cost to 60; see Figure 6b.

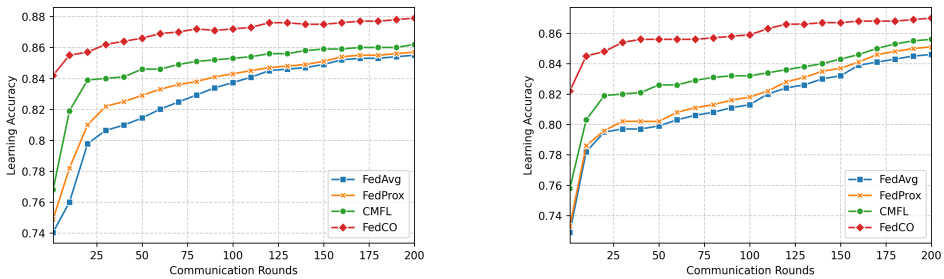


Figure 7: Learning accuracy versus communication rounds for the CIFAR-10 data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in case of the Non-IID data distribution scenario. (**a**) IID; (**b**) Non-IID.

Finally, for the CIFAR-10 IID data, the required communication costs of the FedAvg and FedProx to achieve 85% accuracy is 150 rounds, while CMFL obtains the same result for 75 rounds. Our FedCO algorithm outperforms the others, needing only nine rounds to reach this accuracy value (cf. Figure 7a). In the case of the CIFAR-10 Non-IID data (see Figure 7b), in 25 communication rounds, FedCO obtains an accuracy of 85%, while FedAvg and FedProx reach 79% and 80%, respectively. On the other hand, CMFL achieves a close result 82% of accuracy in the same number of rounds.

FedCO differs from the Federated Learning baseline FedAvg as follows: our algorithm uses a much smaller number of nodes while the aggregation procedure is the same. Thus, if we have a smaller number of workers, convergence is reached faster than in FedAvg, where all the available workers are used. At each round, FedCO selects and uses a different set of workers, and each worker is a representative that achieves the highest accuracy in each cluster. Hence, the accuracy is not sacrificed.

Table 4 shows the number of communication rounds to achieve the maximum model accuracy (i.e., to converge) for the datasets considered. Specifically, the target accuracy values are 90% for MNIST and FashionMNIST and 85% for CIFAR-10. FedAvg is the baseline benchmark, and the iterations saved for algorithm X (X = FedProx, CMFL, or FedCO) is computed as

$$1 - \frac{num\_of\_iteration\_X}{num\_of\_iteration\_FedAvg}.$$

FedCO saves from 75% to 86% of iterations to converge with respect to FedAvg for IID data distribution setting, and it saves from 65% to 82% iterations for the Non-IID data distribution scenario. Moreover, FedCO always converges with at least half of the iteration rounds needed by CMFL. In more detail, one can observe that the model on the MNIST IID data distribution setting converges to an accuracy of 90% in 190 rounds with the FedAvg algorithm, and in 25 rounds for our FedCO algorithm, providing savings of 86%, and in 185 rounds for FedProx, and in 50 rounds for CMFL,

providing savings of 26% and 73%, respectively. The model trained on the FashionMNIST IID data distribution scenario converges to a target accuracy of 90% in 200 rounds for FedAvg, and in 50 rounds for FedCO, saving 75% of communication rounds, while it requires 190 and 60 rounds for FedProx (5% saving) and CMFL (70% saving), respectively. Furthermore, in the FashionMNIST Non-IID data distribution scenario, the model converges to an accuracy of 90% in 200 rounds for the FedAvg algorithm, and in 186 rounds for FedProx, saving only 7%. In contrast, it requires 70 and 150 communication rounds for FedCO and CMFL, with savings of 65% and 25%, respectively. The experimental results on the CIFAR-10 data show that the model trained in the IID and Non-IID data settings need 200 rounds for FedAvg to reach 85% of the accuracy, while it requires 188 rounds for FedProx to reach 85% in IID, and more than 200 rounds in Non-IID to obtain target accuracy. On the other hand, FedCO and CMFL require 30 and 80 rounds, respectively, to converge under the IID data distribution scenario. Furthermore, within the Non-IID data distribution setting, the model converges to an accuracy of 85% in 200 rounds for the FedAvg, while it requires 60 and 160 communication rounds for FedCO and CMFL, respectively. Similarly, in the Non-IID data distribution setting, the FedCO communication costs are reduced to 82% with the MNIST data, 65% with the FashionMNIST data, and up to 70% with the CIFAR-10 dataset, compared to the FedAvg.

Although FedProx is considered to be an optimized version of FedAvg, we can observe from the results discussed above that FedProx behaves very similarly to FedAvg and shows only a slightly better performance than FedAvg in the conducted experiments. In addition, as we mentioned earlier, our FedCO algorithm can also be interpreted as an optimized version of FedAvg. Therefore, we further study these two algorithms (FedCO and FedAvg) by conducting experiments and benchmarking their performance on two datasets from the LEAF repository, namely FEMNIST and CelebA. Figure 8 shows the final accuracy scores after several rounds of communication for the FEMNIST dataset. Comparing the results produced by the two methods, it is evident that FedCO performs significantly better than FedAvg, on both the IID and Non-IID data scenarios. Specifically, FedCO ensures a higher accuracy than that of FedAvg within a smaller number of communication rounds. For example, in Figure 8a, FedCO can reach 90% in only 110 iterations, while the FedAvg never reaches that level within 200 iterations. Analyzing the results in Figure 9, we can observe the following: (1) FedCO consistently outperforms FedAvg in both data distribution scenarios; (2) FedCO generally achieves better accuracies than FedAvg in most cases (see Figure 9b), considering that both of them have been trained with only 200 rounds. Table 5 reports the number of communication rounds that the FedAvg and FedCO algorithms need in order to converge, for the considered datasets. Specifically, the target accuracy values are 70% for FEMNIST and 65% for CelebA, respectively. In addition, FedAvg is considered as the baseline.

Note that these results again verify the faster convergence of FedCO compared to that of FedAvg. Notice that we have also studied and compared FedProx and

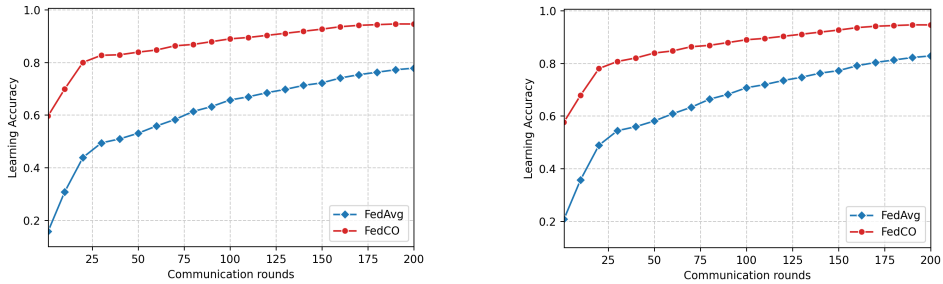


Figure 8: Learning accuracy versus number of communication rounds for FEMNIST data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in the case of the Non-IID data distribution scenario. (**a**) IID; (**b**) Non-IID.

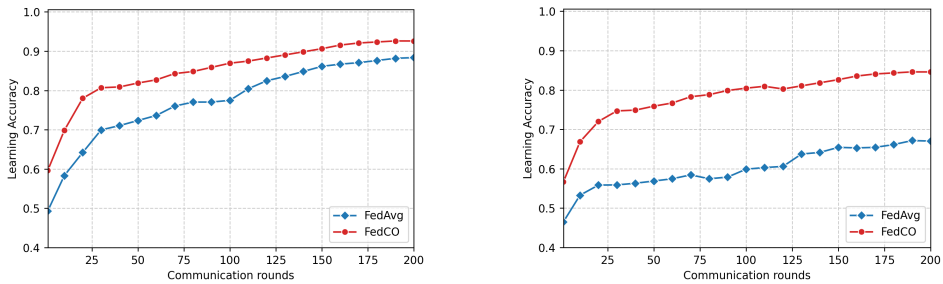


Figure 9: Learning accuracy versus number of communication rounds for CelebA data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in case of the Non-IID data distribution scenario. (**a**) IID; (**b**) Non-IID.

FedAvg on the same LEAF datasets, and they again have demonstrated very similar behaviors.

6.4 Communication Overhead Analysis

In this section, we compare the efficiencies of the two compared FL algorithms for 100 communication rounds with respect to different numbers of workers on the CIFAR-10 and the MNIST datasets, under the IID and Non-IID data distribution scenarios. The obtained results are reported in Figures 10 and 11, respectively. As one can notice, the FedCO algorithm has performed significantly better than the FedAvg, FedProx, and CMFL. The reader can also observe that the communication overhead increases linearly with the number of workers. Hence, to scale in a real scenario with thousands of workers, a FL algorithm should be capable of reducing the communication cost as much as possible, and reducing the number of rounds to converge, as with the proposed FedCO algorithm. Finally, the communication overhead in the IID and Non-IID cases is very close or identical. The results produced on the FashionMNIST dataset are similar to the other two datasets.

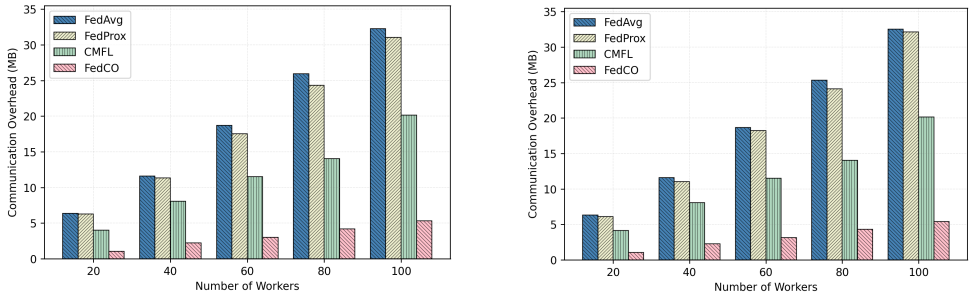


Figure 10: The communication overhead for 100 rounds for the CIFAR-10 data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in the case of the Non-IID data distribution scenario. (**a**) IID. (**b**) Non-IID.

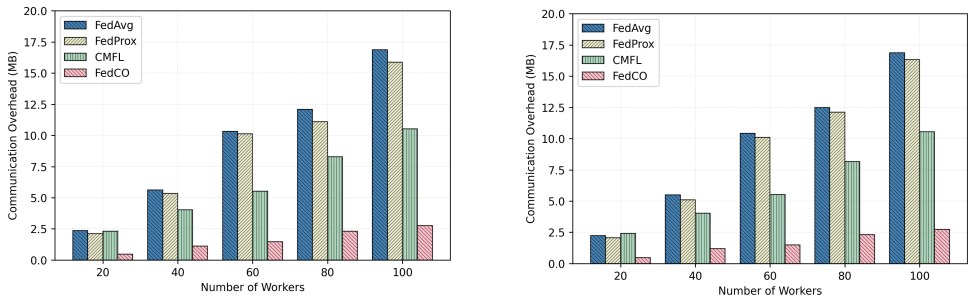


Figure 11: The communication overhead for 100 rounds for the MNIST data. The top plot presents the results produced in the case of the IID data distribution scenario, while the bottom plot depicts the results generated in the case of the Non-IID data distribution scenario. (**a**) IID. (**b**) Non-IID.

The communication cost savings for algorithm X (X = FedProx, CMFL, or FedCO) is computed as

$$1 - \frac{Communication\_overhead\_X}{Communication\_overhead\_FedAvg}.$$

As can be seen in Figure 10a,b, the FedCO costs on the CIFAR-10 IID data are 1 MB for 20 workers, which is a reduction in communication costs by 83% in comparison with FedAvg, while FedProx and CMFL are allowed to save only 12% and 36% in communication costs, respectively. In an experiment involving 100 workers on the CIFAR-10 dataset, FedAvg, FedProx, and CMFL exchange 32.5, 31.04, and 20 MB of data, while the proposed FedCO consumes only 5.4 MB, which means that FedCO reduces the communication overhead by 84% with respect to FedAvg, and CMFL reduces the communication overhead by 38%, while FedProx saves only 3% in communication costs. Figure 11a,b report the communication costs under the MINIST IID and Non-IID data distribution scenarios, respectively. The trend is similar to the results of the CIFAR-10 data experiments. Both the IID and Non-IID data distribution

settings confirm that our FedCO algorithm ensures a significantly smaller communication overhead in comparison with FedAvg, FedProx, and CMFL, by substantially reducing the required number of bytes exchanged. As can be noticed, FedCO allows a saving of between 80% and 85% with respect to FedAvg. In Figures 10 and 11, the communication costs increase linearly with the increasing number of workers for all of the compared algorithms. It is obvious that FedCO consistently outperforms FedAvg, FedProx, and CMFL in terms of reducing communication costs.

6.5 Threshold-Based Worker Selection

We also study scenarios in which we use an accuracy threshold to select the number of workers. The threshold is the specified cut-off of accuracy value for the selection of representatives of a cluster of workers. where the local update ensures an accuracy of greater than or equal to the predefined threshold as a representative of a cluster. In this section, we report the results produced by testing four different threshold values for FedCO, namely 70%, 75%, 80%, and 85%. The network threshold for the selection of workers varies from bandwidth, transmission speed, or packet loss [48].

Table 6 reports the number of the top-ranked workers that the FedCO algorithm has selected to communicate with the server when the predefined threshold is met within 100 communication rounds.

Table 6: The number of selected representatives with respect to four different threshold values on the LEAF datasets CelebA (top) and FEMNIST (bottom) for 100 global rounds.

CelebA		
Threshold Accuracy	IID	Non-IID
≥70%	853	826
≥75%	673	515
≥80%	600	245
≥85%	257	226
FEMNIST		
Threshold Accuracy	IID	Non-IID
≥70%	912	844
≥75%	806	694
≥80%	730	604
≥85%	408	380

In the case of the CelebA data, the highest number of representatives has been selected when the accuracy of the local models is equal to or above 70%, namely 853 and 826 workers under IID and Non-IID, respectively. In the experiments conducted on the FEMNIST data, when the threshold of the local models was greater than or

equal to 70%, 912 workers were selected as representatives for the IID scenario, and 844 workers for the Non-IID one. Similarly, these two values represent the highest numbers of selected workers. It is obvious from the number of representatives reported in Table 6 that the low threshold value implies the greater number of representatives to be selected for global training in FL and vice versa. Thus, we can observe that the proposed algorithm substantially reduces the accumulated communication overhead when FedCO selects only k representatives (i.e., one per cluster), rather than selecting a variable number of representatives based on a predefined threshold to train a global model.

Table 7 presents how many workers per round have been selected as representatives when various thresholds are applied for CelebA under the IID and Non-IID data scenarios, respectively. We can see that until 10 communication rounds, the FedCO selects only k representatives, since there are no local models where the accuracy has reached 70% at 10 rounds. Thus, the number of representatives increases from 10 to 97 at round 12 due to the selection of all the clusters' workers, ensuring an accuracy that is equal to or above the given threshold. Notice that there are 97 workers of different clusters that reach the value of accuracy of their local models of 70% or above. We can see that FedCO needs 30 rounds to have a number of workers whose accuracy is greater than or equal to 80% and to meet this condition under IID data. Furthermore, to meet the threshold of 85%, FedCO requires 100 rounds to have a number of workers (98) such that their accuracy of the local models meets this condition under IID. On the other hand, for Non-IID, FedCO never meets this condition, since no local models have a accuracy value of higher than or equal to 85%; thus, FedCO selects only 36 workers to represent the different clusters.

Table 7: Total number of selected representatives when the given accuracy threshold is reached in CelebA dataset at different rounds.

	IID				**Non-IID**			
Round	**≥70%**	**≥75%**	**≥80%**	**≥85%**	**≥70%**	**≥75%**	**≥80%**	**≥85%**
1	10	10	10	10	10	10	10	10
10	10	12	16	14	14	16	18	16
12	97	13	16	16	14	18	22	18
20	102	103	20	20	104	26	25	24
30	105	104	95	18	104	30	28	24
40	102	108	98	20	110	32	30	26
50	106	108	100	22	112	96	32	28
60	106	110	104	24	114	99	34	28
70	107	111	106	26	116	102	36	32
80	108	114	108	22	120	104	38	35
90	110	116	110	26	118	106	38	34
100	112	116	112	98	122	108	97	36

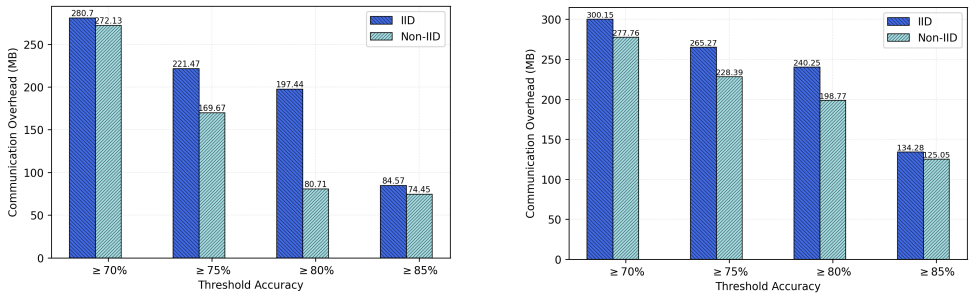


Figure 12: The communication overhead for 100 rounds for the two LEAF datasets. The top plot presents the results produced for CelebA dataset, while the bottom plot depicts the results generated on FEMNIST dataset. (**a**) CelebA; (**b**) FEMNIST.

Figure 12 provides communication overheads for various thresholds. It is obvious to the reader that a higher number of selected representatives implies a higher values of communication costs to the server.

The above results suggest that our proposed FedCO algorithm can substantially reduce the communication overhead by using a higher accuracy threshold. In general, FedCO can be considered as being robust to different application scenarios by being able to tune its parameters (e.g., the accuracy threshold or the number of top-ranked representatives per cluster) to find a trade-off between the application-specific resource constraints and the accuracy requirements.

7 Conclusions

This paper proposes a clustering-based FL approach, entitled Federated Learning using Clustering Optimization (FedCO). The proposed FedCO approach partially builds upon our previous work and extending further towards proposing a dynamic clustering scheme that improves global accuracy and that reduces the communication overhead in a Federated Learning context. The proposed approach dynamically identifies worker participants in each communication round by initially clustering the workers' local updates and selecting a representative from each cluster to communicate with the central server, thus minimizing the communication cost. The proposed FedCO method is evaluated and benchmarked to three other state-of-the-art FL algorithms (FedAvg, FedProx, and CMFL) on five publicly available and widely exploited datasets for studying distributed ML algorithms. The experimental results have shown that the proposed FedCO algorithm significantly reduces communication rounds without sacrificing accuracy. In addition, the experimental evaluation has demonstrated that our FedCO algorithm outperforms the three other FL algorithms under the two studied data distribution scenarios. We have also shown that the FedCO algorithm can dynamically adapt the workers' partitioning at each com-

munication round by relocating the representative workers and conducting the cluster splitting needed for the clustering improvement.

Our future plans include the enhancement of the FedCO approach through using other data distillation techniques; e.g., an interesting future direction could be made by applying computational topology methods for studying data topology and selecting representatives based on this. Another direction is the translation of the FedCO concept to unsupervised learning settings, i.e., developing a resource-efficient FL algorithm based on the unsupervised ML model.

References

[1] W. G. Hatcher and W. Yu. "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends". In: *IEEE Access* 6 (2018), pp. 24411–24432.

[2] I. J. Goodfellow, Y. Bengio, and A. C. Courville. "Deep Learning". In: *Nature* 521 (2015), pp. 436–444.

[3] N. Papernot, P. Mcdaniel, A. Sinha, and M. P. Wellman. "SoK: Security and Privacy in Machine Learning". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (2018), pp. 399–414.

[4] F. Liang, W. G. Hatcher, W. Liao, W. Gao, and W. Yu. "Machine Learning for Security and the Internet of Things: The Good, the Bad, and the Ugly". In: *IEEE Access* 7 (2019), pp. 158126–158147.

[5] H. B. M. et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *International Conference on Artificial Intelligence and Statistics*. 2016.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. "Federated Learning: Challenges, Methods, and Future Directions". In: *IEEE Signal Processing Magazine* 37 (2019), pp. 50–60.

[7] G. Huang, Z. Liu, and K. Q. Weinberger. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2261–2269.

[8] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. "Sparse Binary Compression: Towards Distributed Deep Learning with minimal Communication". In: *2019 International Joint Conference on Neural Networks (IJCNN)* (2018), pp. 1–8.

[9] A. A. M. Al-Saedi, V. Boeva, and E. Casalicchio. "Reducing Communication Overhead of Federated Learning through Clustering Analysis". In: *2021 IEEE Symposium on Computers and Communications (ISCC)* (2021), pp. 1–7.

[10] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. “Federated Optimization in Heterogeneous Networks”. In: *arXiv: Learning* (2018).

[11] L. Wang, W. Wang, and B. Li. “CMFL: Mitigating Communication Overhead for Federated Learning”. In: *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[12] E. Diao, J. Ding, and V. Tarokh. “HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients”. In: *ArXiv* abs/2010.01264 (2020).

[13] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani. “FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization”. In: *ArXiv* abs/1909.13014 (2019).

[14] F. Sattler, K.-R. Müller, and W. Samek. “Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2019), pp. 3710–3722.

[15] N. Shlezinger, S. Rini, and Y. C. Eldar. “The Communication-Aware Clustered Federated Learning Problem”. In: *2020 IEEE International Symposium on Information Theory (ISIT)* (2020), pp. 2610–2615.

[16] Y. Kim, E. A. Hakim, J. Haraldson, H. Eriksson, J. M. B. da Silva, and C. Fischione. “Dynamic Clustering in Federated Learning”. In: *ICC 2021 - IEEE International Conference on Communications* (2020), pp. 1–6.

[17] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran. “An Efficient Framework for Clustered Federated Learning”. In: *IEEE Transactions on Information Theory* 68 (2020), pp. 8076–8091.

[18] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing. “ClusterFL: a similarity-aware federated learning system for human activity recognition”. In: *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (2021).

[19] Y. Chen, X. Sun, and Y. Jin. “Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31 (2020), pp. 4229–4238.

[20] S. Caldas, J. Konecný, H. McMahan, and A. Talwalkar. “Expanding the Reach of Federated Learning by Reducing Client Resource Requirements”. In: *arXiv preprint arXiv:1812.07210* (2018).

[21] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally. “Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training”. In: *ArXiv* abs/1712.01887 (2017).

[22] F. S. et al. “Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31 (2019), pp. 3400–3413.

[23] T. Vogels, S. P. Karimireddy, and M. Jaggi. “PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization”. In: *Neural Information Processing Systems*. 2019.

[24] M. Asad, A. Moustafa, and T. Ito. “FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning”. In: *Applied Sciences* 10 (2020).

[25] A. M. et al. “FEDZIP: A Compression Framework for Communication-Efficient Federated Learning”. In: *ArXiv* abs/2102.01593 (2021).

[26] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. E. Gonzalez, and R. Arora. “FetchSGD: Communication-Efficient Federated Learning with Sketching”. In: *ArXiv* abs/2007.07682 (2020).

[27] J. X. et al. “Ternary Compression for Communication-Efficient Federated Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33 (2020), pp. 1162–1176.

[28] Z. Chai, Y. Chen, A. Anwar, L. Zhao, Y. Cheng, and H. Rangwala. “FedAT: A High-Performance and Communication-Efficient Federated Learning System with Asynchronous Tiers”. In: *SC21: International Conference for High Performance Computing, Networking, Storage and Analysis* (2020), pp. 1–17.

[29] X. Wu, Z. Liang, and J. Wang. “FedMed: A Federated Learning Framework for Language Modeling”. In: *Sensors (Basel, Switzerland)* 20 (2020).

[30] M. Asad, A. Moustafa, and M. Aslam. “CEEP-FL: A comprehensive approach for communication efficiency and enhanced privacy in federated learning”. In: *Appl. Soft Comput.* 104 (2021), p. 107235.

[31] 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)*. 2019, pp. 1–7. DOI: 10.1109/ICC.2019.8761315.

[32] S. Park, Y. Suh, and J. Lee. “FedPSO: Federated Learning Using Particle Swarm Optimization to Reduce Communication Costs”. In: *Sensors (Basel, Switzerland)* 21 (2021).

[33] Z. C. et al. “Dynamic Attention-based Communication-Efficient Federated Learning”. In: *ArXiv* abs/2108.05765 (2021).

[34] H. Larsson, H. Riaz, and S. Ickin. “Automated Collaborator Selection for Federated Learning with Multi-armed Bandit Agents”. In: *Proceedings of the 4th FlexNets Workshop on Flexible Networks Artificial Intelligence Supported Network Flexibility and Agility* (2021).

[35] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang. “Learning Private Neural Language Modeling with Attentive Aggregation”. In: *2019 International Joint Conference on Neural Networks (IJCNN)* (2018), pp. 1–8.

[36] M. Ribero and H. Vikalo. “Communication-Efficient Federated Learning via Optimal Client Sampling”. In: *ArXiv* abs/2007.15197 (2020).

[37] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. “Federated Learning: Strategies for Improving Communication Efficiency”. In: *ArXiv* abs/1610.05492 (2016).

[38] J. B. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *In Lucien M. Le Cam and Jerzy Neyman, editors, Proceedings of the Berkley symposium on mathematical statistics and probability* 1 (1967), pp. 281–297.

[39] P. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proc. IEEE* 86 (1998), pp. 2278–2324.

[41] H. Xiao, K. Rasul, and R. Vollgraf. “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: *ArXiv* abs/1708.07747 (2017).

[42] A. Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. In: 2009.

[43] S. e. a. Caldas. “LEAF: A Benchmark for Federated Settings”. In: (2018).

[44] Y. LeCun and C. Cortes. “The mnist database of handwritten digits”. In: 2005.

[45] J. Kang, Z. Xiong, D. T. Niyato, S. Xie, and J. Zhang. “Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory”. In: *IEEE Internet of Things Journal* 6 (2019), pp. 10700–10714.

[46] N. H. Tran, W. Bao, A. Y. Zomaya, M. N. H. Nguyen, and C. S. Hong. “Federated Learning over Wireless Networks: Optimization Model Design and Analysis”. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications* (2019), pp. 1387–1395.

[47] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik. “Federated Optimization: Distributed Machine Learning for On-Device Intelligence”. In: *ArXiv* abs/1610.02527 (2016).

[48] P. Zhou, P. Fang, and P. Hui. “Loss Tolerant Federated Learning”. In: *ArXiv* abs/2105.03591 (2021).

Paper V

Group-Personalized Federated Learning for Human Activity Recognition Through Cluster Eccentricity Analysis

Ahmed A. Al-Saedi, and Veselka Boeva
In: Engineering Applications of Neural Networks, Springer, León, Spain, 2023.

Abstract

Human Activity Recognition (HAR) plays a significant role in recent years due to its applications in various fields including health care and well-being. Traditional centralized methods reach very high recognition rates, but they incur privacy and scalability issues. Federated learning (FL) is a leading distributed machine learning (ML) paradigm, to train a global model collaboratively on distributed data in a privacy-preserving manner. However, for HAR scenarios, the existing action recognition system mainly focuses on a unified model, i.e. it does not provide users with personalized recognition of activities. Furthermore, the heterogeneity of data across user devices can lead to degraded performance of traditional FL models in the smart applications such as personalized health care. To this end, we propose a novel federated learning model that tries to cope with a statistically heterogeneous federated learning environment by introducing a group-personalized FL (GP-FL) solution. The proposed GP-FL algorithm builds several global ML models, each one trained iteratively on a dynamic group of clients with homogeneous class probability estimations. The performance of the proposed FL scheme is studied and evaluated on real-world HAR data. The evaluation results demonstrate that our approach has advantages in terms of model performance and convergence speed with respect to two baseline FL algorithms used for comparison.

1 Introduction

With the recent development of edge devices, such as mobile phones, wearables, IoT devices etc., a massive amount of data can be generated. Such data can be utilized by training-based intelligent applications for human activity recognition (HAR). Traditional solutions require sending these data to a central server and training there in a centralized way. However, this introduces huge communication overhead, consumes network resources, and brings privacy concerns [1]. To solve this problem, Google proposes a decentralized approach called Federated Learning (FL), where model parameters instead of data are transferred between the central server and edge nodes (called workers hereafter) [2]. A central server periodically sends the global model to a set of workers. These workers train the shared model without sharing their private data to generate updated local models, which are later submitted to the server [3]. Finally, the server aggregates the local models and generates a new global model. This process is repeated until a satisfactory global model is obtained. This approach is called naive FL, because the workers involved in training are usually randomly selected at each round, and the trained parameters are aggregated by averaging. This scheme works well for IID (independently identically distribution) data but has unsatisfactory performance for Non-IID data [2]. Practically, the assumption to consider that the local data of each edge device is always IID does not hold, often impacting overall model performance. However, compared with IID data, Non-IID datasets have significant variability in data class distribution and size [4]. Figure 1 illustrates the different ways to model FL. The traditional FL setting, presented in the middle plot, assumes a federation of distributed workers, each with its own private data. These workers join the FL global training to achieve a better model performance. As a result, the global model is generated from local models with different characteristics, derived from different types of data. Thus, these different characteristics captured by the local models' parameters will be later mitigated when global aggregation occurs [5]. Furthermore, the traditional FL paradigm faces fundamental challenges, such as heterogeneous data across workers and a lack of solution personalization. In contrast to the traditional FL paradigm, Personalized Federated Learning (PFL) addresses the mentioned two fundamental challenges. PFL takes a completely local approach. In this context, each worker represents a different ML task with a different data distribution, and a private model for each task will be trained and used to deal with the specific nature of the data. Therefore, the output is a unique personal model for each worker, but no peer learning [6]. This scenario is illustrated in the left plot of Figure 1. However, even though the tasks among workers are different, it is reasonable to assume that there is similarity across different tasks. On the other hand, in the traditional FL scenario (the middle plot of Figure 1), in case of imbalanced data, where each worker only has one specific class of data, the results of averaging of the model parameters for producing an aggregated global model can lead in a significant accuracy decrease, e.g., up to 11% for MNIST, 51%, and for CIFAR-10 datasets are

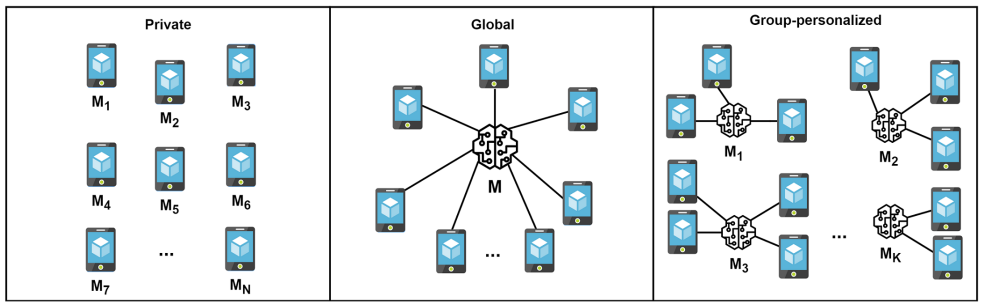


Figure 1: Comparison of three different federated learning scenarios: (i) the left plot presents a setup in which each model is trained on the worker's private data; (ii) the middle plot is a scenario where a global model is built from the locally trained workers' models; (iii) the right plot illustrates a setting accounting for similarity among the workers and building a global model from the local models of each group of similar workers.

reported in [4].

Our proposed approach tries to tackle the discussed challenges and find a trade-off between the two extreme cases described above. Namely, we propose grouping workers based on their empirical probabilities, reflecting their current data class distribution. In particular, the workers with similar empirical probability vectors are placed in the same cluster. Later when updating the global model, we average the parameters from the same group. In that way, only local updates uploaded by the workers within the same group will be aggregated. Then in the next round, the aggregated group global model is sent to the same group to train, as it is illustrated in the right plot of Figure 1. Evidently, our proposed Group-Personalized FL (GP-FL) algorithm is capable of training simultaneously several global models, one per each group of workers with similar activity patterns. At each training round, each worker's empirical probability vector is updated in order to reflect the information in its new data batch. In addition, cluster eccentricity analysis [7] is applied to the workers' current grouping. In that way, at the next round, some workers may change their cluster or even new singelton clusters may appear.

The GP-FL algorithm has been evaluated in a set of experiments, based on a well-defined evaluation setup in the HAR domain. The HAR problem is well suited to our FL scenarios, because various activities tend to have generic patterns while being highly idiosyncratic [8, 9]. The performance of our GP-FL algorithm is benchmarked to that of two other FL algorithms, Federated Averaging (FedAvg) [2] and Clustered Federated Learning (CFL) [10]. GP-FL demonstrates its superiority over both algorithms in the conducted experiments with respect to the achieved performance.

2 Related Work

Recently we have witnessed a lot of attention on personalization in FL. Our proposed work is related to PFL in HAR and distributed multi-task learning. We explore some existing approaches related to those topics. For example, in [11] a random forest based personalized FL model is proposed for recognizing many human activities. In this work, the authors use local sensitivity hashing for calculating the similarity between different users. Based on this similarity, a subset of the top-k most similar users is selected for training the federated forest model iteratively. In [12] a novel hybrid approach is suggested for HAR that combines semi-supervised and FL settings to build a global model for privacy awareness. Yu *e*t al. [13] have developed a method that relies on a semi-supervised gradient aggregation strategy for activity detection using sensor data for online HAR tasks. In [14], the authors have proposed the FedStack framework, which supports ensemble heterogeneous architectural client models for mobile health sensor datasets. FedStack has been applied to mobile health sensor data to recognize 12 different activities. Presotto *e*t al. [15] have proposed FedAR: a novel hybrid approach to unify federated learning with semi-supervised learning for activity recognition on mobile devices. It relies on active learning and label propagation to semi-automatically annotate the local streams of unlabeled sensor data. Tashakori *e*t al. introduce in [16] a novel personalized semi-supervised learning approach focusing on edge intelligence. SemiPFL creates a personalized autoencoder to enable learning from user data representation. In [17], the authors have presented FedCLAR: a novel federated clustering framework according to the similarity of the local updates for HAR. FedCLAR combines federated clustering with transfer learning methods to reduce the non-IID issue. In [18], a method called SS-FedCLAR that combines federated clustering and semi-supervised learning in FL settings is introduced to reduce the non-IID and the data lack issues simultaneously. The authors in [19], have presented a federated transfer learning method for wearable healthcare to address security and personalization challenges. Lu *e*t al. have proposed AdaFed: a weighted federated transfer learning framework to tackle domain shifts and to realize personalization for local clients in healthcare [20]. Ma *e*t al. [21] have focused on label concept drift. They have presented a variational Bayes framework for PFL based on hierarchical Bayesian inference.

Similarly to our work, Sattler *e*t al. [10] propose a hierarchical clustering FL scheme, forming client clusters, and those in the same cluster share the same model for training. This algorithm, called Clustering Federated Learning (CFL), is used as a baseline in the evaluation of our proposed GP-FL algorithm. Notice that most of the above mentioned works are aimed at the personalized training of deep learning models in a federated learning setup. Our work instead introduces a lightweight model based on logistic regression that is more suitable for modern resource-constrained wearable devices for HAR monitoring.

3 Preliminaries

In this section, we introduce the baseline FL algorithms used in the evaluation of our GP-FL algorithm. We also provide with a formal description of the FL setting, and motivate the methods and optimization procedures used in the proposed GP-FL algorithm.

3.1 Baselines

To assess the performance of our proposed method, we compare the GP-FL algorithm against two other FL methods, namely FedAvg [2] and CFL [10].

3.1.1 Federated Averaging (FedAvg):

FedAvg is the predominant algorithm for federated learning [2], following a server-client setup with two repeating phases (i) the clients train a shared global model locally on their data by making multiple local updates, and (ii) the server averages the locally updated models to obtain a new global model. In contrast to FedAvg method, GP-FL builds several global ML models, each one trained on a dynamic group of clients with homogeneous class probability estimations.

3.1.2 Clustered Federated Learning (CFL):

A clustering framework to deal with federated multi-task learning have proposed in [10]. The CFL groups clients into clusters of similar clients according to their local data distribution. Thus, the goal is to train a single global model for each cluster. Similarly to the CFL method, our proposed GP-FL algorithm trains a set of global models, one per each cluster of workers. In our work, the workers are however, clustered into groups according to their local data distribution with respect to the classes. In that way, two workers are grouped together if they have similar local class probability distributions, i.e. evidently different groups of workers have different learning tasks. In addition, the clustering is adapted at each training round by accounting the evolving nature of data distribution with respect to the classes.

3.2 Problem Setting

As we stated in Section 1, our aim is to show how the fundamental idea behind our GP-FL approach can be exploited to design a group-personalized solution of FL problem. To do so, let us briefly describe the GP-FL setting. Given a set of workers in contrast to the traditional supervised federated learning setting, the goal in our GP-FL framework is not finding a global model that performs well for all the workers, but training a set of global-personalized models, one per a group of workers. Therefore in our GP-FL solution, we initially segment workers into groups based on the similarity of their class probability estimations.

We consider a typical setting of FL with a model $\mathcal{M}$ that is learned iteratively by using a randomly selected subset, denoted by W_t ($W_t \subset W$), of the set W of all available workers. The workers in W_t participate at each round and compute the gradient of the loss over all the data held by them. Each worker $w \in W_t$ at round t has its own row of data D_t^w and a local model $\mathcal{M}_t^w$. At each round t, each worker trains its local model by iterating the local update multiple times of Stochastic Gradient Descent (SGD) before sending the next local model $\mathcal{M}_{t+1}^w$ to the server which holds the global model. The server, after collecting all the local models computed at round t, performs a synchronous update of the global model $\mathcal{M}_{t+1}$. The global model update can be computed using different criteria. In this paper, we assume it is calculated by means of federated averaging, that is the local model $\mathcal{M}_t^w$, $w \in W_t$ and global model $\mathcal{M}_t$ are updated by the following equations: [22]:

$$\mathcal{M}_{t+1}^w = \mathcal{M}_t^w - \eta g_t^w; \tag{1}$$

$$\mathcal{M}_{t+1} = \sum_{w \in W_t} \frac{n_w}{n} \mathcal{M}_{t+1}^w, \tag{2}$$

where $\mathcal{M}_{t+1}^w$ is the local update, g_t^w are the updated weights on its local data in the current model $\mathcal{M}_t^w$, $\mathcal{M}_{t+1}$ is the global model, η is a learning rate calculated by each worker, W_t is the set of workers which participate in the training, n is the total number of all data points and n_w is the number of local data points of the worker $w \in W_t$. In our GP-FL setup we compute a group global model for each cluster of workers ($C_{tj} \subset W_t$), then Eq. 2 is changed to

$$\mathcal{M}_{t+1}^{C_{tj}} = \sum_{w \in C_{tj}} \frac{n_w}{n} \mathcal{M}_{t+1}^w, \tag{3}$$

where $\mathcal{M}_{t+1}^{C_{tj}}$ is the group global model built for each cluster C_{tj} at round t. The server then distributes the group global model $\mathcal{M}_{t+1}^{C_{tj}}$ to its updated group of workers, i.e. for each $w_i \in C_{t+1j}$, where C_{t+1j} is the updated version of C_{tj}, to perform another iteration of local training and model update.

3.3 Data Smoothing

In our proposed GP-FL solution each worker is modeled by its class probability estimations, i.e. an empirical probability vector, where each value represents the relative frequency of the corresponding class among the all training examples. For example, if we have a set D of labelled examples, and the number of examples in D of class C_i is n_i ($i = 1, 2, \ldots, k$), then the empirical probability vector associated with D is given by $\hat{\mathbf{p}}(D) = (n_1/|D|, \ldots, n_k/|D|)$ Such empirical probability vector is initially calculated for each worker in our FL model and then update at each following round according to the current data batch at this worker.

In order to avoid issues with extreme values, such as 0 or 1, each empirical probability vector $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k)$ can be smoothed by applying Laplace correlation which is expressed as:

$$\hat{p}_i(D) = \frac{n_i + 1}{\mid D \mid + k}, \tag{4}$$

where D is set of labelled examples, n_i is the number of examples in class C_i, and k (the number of classes) is added to ensure that the posterior probabilities are never zero [23].

3.4 Markov Clustering Algorithm

Markov Cluster algorithm (MCL) is an unsupervised pattern recognition algorithm based on finding the optimal cluster of a connected graph, without any a priori knowledge of the cluster sizes. It is used to cluster sequence similarity or simple networks [24]. MCL can efficiently utilize 2000 compute nodes and cluster a network of about 70 million nodes with about 68 billion edges in approximately 2.4 h [25]. What really distinguishes MCL from other clustering techniques is that it does not require any input to form clusters, unlike k-means algorithm and other partitioning algorithms. This makes this algorithm crucial in network data, social network data, or even similarity detection.

3.5 Wasserstein distance

The Wasserstein distance, which is a metric used to measure the distance between probability distributions, is induced by the optimal transport problems [26]. Methods based on the advantages of Wasserstein distance have been used successfully in several research areas, including statistics, machine learning, natural language processing, and computer vision [27]. In such applications, the distance is measured by comparing one probability distribution with another, which arises from the theory of optimal transport problems [26]. In the implemented version of our proposed FL model, we use Wasserstein distance to measure the similarity between class probability estimations of each pair of workers. In this way, the individuals who have similar activity patterns will be grouped together by applying the MCL algorithm discussed above.

3.6 Eccentricity Analysis

New eccentricity-based anomaly detection analysis principles have been introduced in [7]. An algorithm, called AutoCloud, based on the introduced principles is proposed in [28]. Similarly to the idea in AutoCloud, we use eccentricity analysis in our proposed FL solution to maintain a dynamic grouping of the workers. In this

context, the eccentricity ξ^j of a worker w_i in relation to a cluster of workers C_j can be calculated as [28]:

$$\xi^j(w_i) = \frac{1}{n_j} + \frac{(\mu_i^j - \hat{p}_i)^T(\mu_i^j - \hat{p}_i)}{\sigma_i^j}, \tag{5}$$

where n_j is the size of C_j, $\hat{p}_i$ is empirical probability vector associated with the worker w_i, and μ_i^j and σ_i^j are the mean and variance, respectively, supposing $w_i \in C_j$.

Eq. 10 presents how eccentricity can be applied to determine whether a worker belongs to a given cluster. Furthermore, the Chebyshev inequality has been utilized to apply a threshold to check whether a worker still belongs to an existing cluster [29]. A particular worker w_i is considered to belong to a cluster C_j if the following condition is satisfied

$$\xi^j(w_i) \leq \upsilon_j \text{ and } \upsilon_j = (m^2 + 1)/2n_j, \tag{6}$$

where m ($m > 0$) is a user-defined parameter that directly affects the evaluation of clustering, and υ_j is the threshold associated with cluster C_j. Although it can be defined using multiple criteria, $m = 3$ is largely used as a standard value and leads to satisfactory results for different data sets and different configurations [30].

4 Proposed Approach

In this section, we formally present our proposed algorithm, namely group-personalized FL (GP-FL) to build a set of group global FL models. We propose to group the available workers according to their empirical class probability distributions. The workers with similar empirical probabilities are grouped together into the same cluster based on their similarity measured by Wasserstein distance. In addition, the built grouping is not static, but it is dynamically updated at each training round by applying cluster eccentricity analysis. This approach allows a global model at the cluster level to be built, overcoming the issue of personalization in traditional FL techniques.

The GP-FL algorithm foresees two distinctive phases: *initialization* and *iteration*. These phases are described in what follows. Let $W = \{w_1, w_2, \ldots, w_n\}$ be the set of all available workers, and W_t is a subset of W that contains the workers selected at round t.

Initialization Phase:

1. At time $t = 0$, the Server initializes the inputs for the GP-FL algorithm. These are model $\mathcal{M}_t$, set of workers W_t, and a number of iterations T.

2. The Server transmits the initial global model $\mathcal{M}_t$ to the set of workers W_t ($W_t \subset W$).

3. Each worker $w_i \in W_t$ receives the global model $\mathcal{M}_t$ and optimizes its parameters locally, i.e. the $\mathcal{M}_t^i$ initial update is produced alongside with a vector $\hat{p}_t(w_i)$ that represents the empirical probabilities of the classes distribution and sent back to the Server.

4. The Server performs the following operations:

   a) Laplace smoothing is applied to each vector $\hat{p}_t(w_i)$ of each worker $w_i \in W_t$.

   b) The smoothed vectors $\hat{p}_t(w_i)$, for $w_i \in W_t$, are used to create a distance matrix. This matrix is then passed as an input parameter to the predicted function of a Markov clustering. As a result, groups of workers with similar empirical probability vectors are produced, i.e. an initial clustering $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk}\}$ of the workers is created.

   c) For each cluster $C_{tj} \in C_t$, $(j = 1, 2, \ldots, k)$, a global group model $\mathcal{M}_t^j$, is built by averaging over the model parameters of the workers assigned to C_{tj}, i.e. a set of initial global group models is produced $\{\mathcal{M}_t^j \mid C_{tj} \in C_t\}$.

   d) For each cluster $C_{tj} \in C_t$ mean data vector μ_i^j and aggregated variance σ_i^j are calculated.

5. The Server aggregates the parameters $\{\mathcal{M}_t^i \mid w_i \in W_t\}$ uploaded by the selected workers W_t to update the global model $\mathcal{M}_t$ through the FedAvg algorithm (Eq. 2).

Iteration Phase:

1. The Server sends each group global model $\mathcal{M}_t^j$, $(j = 1, 2, \ldots, k)$ to its group of workers C_{tj}.

2. Each worker $w_i \in C_{tj}$ receives the group global model $\mathcal{M}_t^j$ and optimizes its parameters locally, i.e. $\mathcal{M}_{t+1}^i$ local update and the empirical probability vector $\hat{p}_{t+1}(w_i)$ are produced.

3. The Server updates the existing empirical probability vector $\hat{p}_{t+1}(w_i)$ by taking the average of it with the information provided by the previous data batch, i.e. $\hat{p}_t(w_i)$.

4. The Server applies Laplace smoothing to each vector $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_t \mid$.

5. The Server adapts the workers' grouping C_t to the current empirical probability vectors $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_t \mid$, by invoking eccentricity score $\xi^j(w_i)$ (see Eq. 9), $(j = 1, 2, \ldots, k)$ which assesses whether each worker

$w_i \in C_{tj}$ is still adequately tight with its current cluster, the one it was assigned at the previous round (t).

a) If $\xi^j(w_i)$ is below the threshold $\upsilon_j(t)$ (see Eq. 10) the worker does not change its cluster C_{tj}.

b) If $\xi^j(w_i) > \upsilon_j(t)$ then we calculate $\xi^l(w_i)$ for the other clusters, i.e. for each $C_{tl} \in C_t \setminus C_{tj}$, and will assign the worker w_i to the cluster for each $\xi^l(w_i) < \upsilon_l(t)$. In case this is true for more than one cluster we will assign the worker to the cluster for which the score is lowest.

c) If $\xi^l(w_i) > \upsilon_l(t)$ for each cluster $C_{tl} \in C_t \setminus C_{tj}$ then this worker w_i will give the start of a new singleton cluster, which means that this worker w_i cannot be assigned to any existing cluster in C_t. Note that $k_{(t+1)} \geq k_t$, where $k_{(t+1)} = \mid C_{t+1} \mid$, since new singleton clusters may appear due to the updating operation.

6. For each cluster $C_{t+1j} \in C_{t+1}$, mean data vector μ_i^j and aggregated variance σ_i^j are calculated, considering the current grouping of the workers and also using the current empirical probability vectors $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_{t+1} \mid$. These values of μ_i^j and σ_i^j will be needed at the next round to calculate the workers' eccentricity scores w.r.t. the current clusters.

7. The updated clustering C_{t+1} is produced, and the clusters in C_{t+1} may contain different workers from the clusters in C_t.

Steps 1–7 of the *iteration* phase are repeated until a certain number of training rounds T is reached.

5 Experimental Design

5.1 HAR Datasets

Despite the UCI dataset [31] has been widely used as a benchmark in the HAR domain, this dataset is not realistic, since it is acquired in-lab following strict scenarios [8]. In our study, the experiments are conducted on two realistic, and publicly available datasets: REALWORLD, a large, diverse device positioning dataset [32], and HHAR, a HAR dataset [33].

Table 1: A summary of datasets' properties

Dataset	Workers	No of data points	Activity	No of classes
REALWORLD	15	356,427	(ST,SD,W,U,D,J,L,R)	8
HHAR	51	85,567	(ST,SD,W,U,D,BK)	6

Each dataset has its own set of activities, as shown in Table 1 with only some overlapping. These datasets deal with various activities: Sit (ST), Stand (SD), Walk (W), Upstairs (U), Downstairs (D), Bike (BK), Jump (J), Lay (L), Run (R). The partitioning of the data has been performed as follows. For each dataset, 20% is left for testing at the central server, while the remaining 80% is used for training.

Note that the initial clustering of workers (individuals) has been produced by using the MCL algorithm, with parameters inflation 1.7 and threshold 2. The MCL algorithm is implemented in the MCL package in Python.

5.2 Evaluation strategy

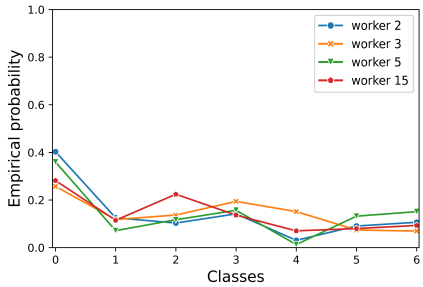
As previously explained, the aim of FL over classical learning is the ability to merge several worker models into a global one in order to improve model generalization without degrading specialization. To study and evaluate the performance of the proposed GP-FL algorithm, we have computed and compared three different evaluations calculated for each experiment for each worker's data:

- **Personal performance**: This is evaluated by computing the accuracy or F1 score achieved by the client's local model using its local data.
- **Global performance**: This is evaluated by calculating the accuracy or F1 score produced by the overall global model (aggregating all the clients' models) on each worker's local data.
- **Group performance**: This calculates the accuracy or F1 score achieved by each group global model on the local data of each worker from its group.

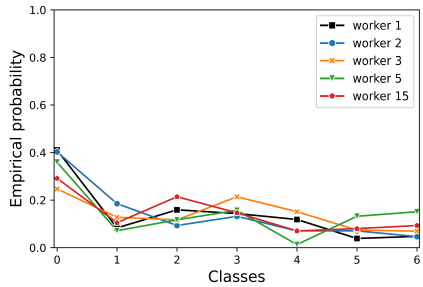
For each worker, the dataset is partitioned into a training set and a test set. The test set is used for the local evaluation of each worker's performance. These evaluation results are then aggregated to give the **personal performance** evaluation of each worker. The global model runs individually on each worker test set and then aggregates the obtained F1 scores, which is used to evaluate the **global performance**. Each group global model executes individually on the test set of each worker in the group associated with it to evaluate the **group performance**. In this way, three different models are evaluated on each worker test data, namely the worker locally trained model, the overall global model averaging over the parameters of all workers' local models and the group-personalized model based only on the local models of the workers having similar activity patterns, i.e. ones that are grouped together.

6 Experimental Results

We have initially compared the performance of workers' personal (local) models with that of both the traditionally built federated learning (global) model and global group



(a) "green" cluster workers' activity profiles at round 1



(b) "green" cluster workers' activity profiles at round 10

Figure 2: Comparison of the workers' activity profiles (empirical probability vectors) distributed in the "green" cluster in the first and tenth rounds, respectively.

models trained by our GP-FL algorithm. For each experiment, three evaluations have been performed. Namely, the three models (local, global, and group) associated with each worker are run on its test data at each round. The performance of each run is evaluated with respect to accuracy.

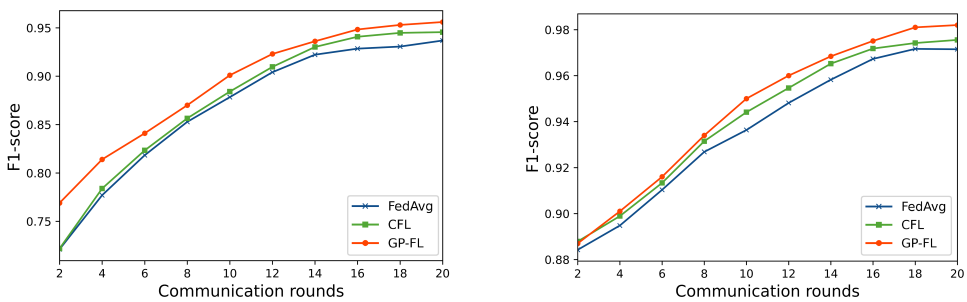
In order to illustrate the properties of the clustering scheme proposed in this study, we show in Table 2 the clustering updates in the first 10 global communication rounds of the GP-FL algorithm applied to the REALWORLD dataset. The performance in terms of accuracy of the three models (local (L), global (G), and group (Gr)) associated with each worker are compared in the table. As one can notice in the first round, the 15 workers have been clustered into 5 groups. Namely, ”pink” cluster has two workers (1 and 12), ”green” cluster has four workers (2, 3, 5, and 15), and the remaining three clusters (i.e. ”yellow”, ”orange” and ”cyan”) each one has three workers. It is interesting to notice that in round 10, worker 1 has moved to the ”green” cluster, due to its eccentricity score being higher than the threshold of the ”pink” cluster. Therefore, the eccentricity score of this worker has been calculated with respect to each one of the other clusters and as a result, it has been assigned to the ”green” cluster.

The workers’ empirical probability vectors distributed in the ”green” cluster in the first and tenth rounds, respectively are compared in Figure 2. As one can see the worker 1 activity pattern in the tenth round is very similar to the other individuals distributed in this cluster, i.e. this is the reason to be moved to the ”green” group. We can also observe that worker 8 has similar behavior of changing its cluster from ”cyan” to ”orange”. Notice that the clusters presented in Table 2 will continue to be optimized in the same fashion, discussed above, for the upcoming communication rounds. Overall, the group global models built by the GP-FL algorithm have produced accuracy scores that are higher or at least compatible with those generated by the global model as it can be noticed in the tenth round results in Table 2. The results generated on the experimental dataset of HHAR are similar.

We also compare the performance of our GP-FL algorithm with that of FedAvg

Table 2: Comparison of the accuracy scores produced by the local (L), global (G), and group global (Gr) models on each worker's data for the conducted training rounds (only the results from the first two and the last two rounds are depicted) using REALWORLD dataset.

		Workers														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	L	0.80	0.86	0.90	0.80	0.81	0.83	0.85	0.80	0.91	0.85	0.86	0.88	0.84	0.85	0.84
	G	0.78	0.82	0.87	0.77	0.78	0.80	0.82	0.78	0.88	0.81	0.82	0.84	0.81	0.80	0.81
	Gr	0.79	0.84	0.88	0.79	0.80	0.81	0.82	0.79	0.89	0.81	0.83	0.81	0.81	0.82	0.80
2	L	0.86	0.85	0.83	0.86	0.88	0.86	0.85	0.85	0.88	0.87	0.87	0.85	0.86	0.85	0.86
	G	0.84	0.82	0.80	0.80	0.84	0.83	0.80	0.81	0.83	0.83	0.84	0.81	0.81	0.82	0.83
	Gr	0.84	0.84	0.82	0.85	0.85	0.84	0.81	0.82	0.85	0.82	0.84	0.81	0.82	0.83	0.83
									.							
									.							
									.							
9	L	0.90	0.92	0.94	0.88	0.90	0.91	0.90	0.90	0.90	0.92	0.94	0.93	0.90	0.89	0.90
	G	0.89	0.88	0.90	0.84	0.86	0.87	0.88	0.85	0.84	0.89	0.90	0.90	0.85	0.86	0.85
	Gr	0.91	0.89	0.90	0.86	0.88	0.90	0.91	0.89	0.86	0.91	0.92	0.93	0.88	0.89	0.85
10	L	0.93	0.93	0.94	0.93	0.90	0.95	0.93	0.93	0.89	0.95	0.93	0.94	0.96	0.90	0.91
	G	0.90	0.88	0.90	0.89	0.86	0.90	0.87	0.85	0.82	0.91	0.89	0.89	0.92	0.86	0.90
	Gr	0.91	0.90	0.91	0.90	0.88	0.92	0.89	0.88	0.86	0.93	0.90	0.92	0.94	0.87	0.94



(a) REALWORLD, Non-IID (30%) **(b)** HHAR, Non-IID (30%)

Figure 3: Comparison of the achieved F1 scores versus the number of communication rounds for Non-IID data (30%) of the three FL algorithms: FedAvg, CFL, and GP-FL.

and CFL algorithms. The performance of the three compared algorithms has been evaluated by running 3-fold cross-validation on each experimental dataset of REALWORLD and HHAR for 20 communication rounds for Non-IID label skew data (30%). In Figure 3a, we compare the F1-scores of the three FL approaches in case of the Non-IID data distribution scenario of REALWORLD data. Within 10 communication rounds, CFL and FedAvg reach 88%, and 87% F1-scores, respectively, while our GP-FL algorithm achieves an F1-score of 90% with the same number of communication rounds. Similar to the REALWORLD dataset, we can see in the case

of HHAR Non-IID data (Figure 3b), the GP-FL algorithm has obtained F1-score of 96% in 12 communication rounds, while CFL and FedAvg have reached 95% and 94%, respectively.

7 Conclusion

In this paper, we have proposed a new approach for building a set of group personalized models in case of Non-IID data in federated learning framework. Initially, Markov clustering algorithm is applied to divide the workers into groups according to the similarity between their empirical probability vectors reflecting the distribution of their training examples among the classes. This allows for building a private global model for each cluster of workers. The built global models, each one trained on a group of clients with homogeneous class probability estimations, are adapted at each training round with respect to the new data batches. The performance of the proposed GP-FL algorithm has been studied and evaluated on public HAR data. The obtained results have shown that the global models trained by the GP-FL algorithm can achieve better performance compared with that of the trained overall global model. The algorithm performance is also compared with that of two other baseline FL algorithms, namely FedAvg and CFL. The GP-FL has outperformed both algorithms in the conducted experiments with respect to the achieved performance and convergence speed. Our future plans include the evaluation and further study of the properties and performance of the proposed GP-FL algorithm in other applied FL scenarios. In addition, we plan to research scenarios in which the recently arrived data batches have a higher importance on the trained group global models than the previous ones.

References

[1] Q. Zheng et al. “Research on hierarchical response recovery method of distribution network fault based on topology analysis”. In: *Int. J. Crit. Infrastructures* 17 (2021), pp. 216–236.

[2] H. B. M. et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *International Conference on Artificial Intelligence and Statistics*. 2016.

[3] Q. Xia et al. “A survey of federated learning for edge computing: Research problems and solutions”. In: *High-Confidence Computing* (2021).

[4] Y. Zhao et al. “Federated Learning with Non-IID Data”. In: *ArXiv* 1806.00582 (2018).

[5] C. Mei et al. “C2S: Class-aware client selection for effective aggregation in federated learning”. In: *High-Confidence Computing* 2.3 (2022). DOI: http://dx.doi.org/10.1016/j.hcc.2022.100068.

[6] A. Z. Tan et al. “Towards Personalized Federated Learning”. In: *IEEE transactions on neural networks and learning systems* PP (2021).

[7] P. Angelov. “Anomaly detection based on eccentricity analysis”. In: *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*. 2014, pp. 1–8. DOI: 10.1109/EALS.2014.7009497.

[8] E. Sannara et al. “Evaluation and comparison of federated learning algorithms for Human Activity Recognition on smartphones”. In: *Pervasive Mob. Comput.* 87 (2022), p. 101714.

[9] E. Sannara et al. “Evaluation of federated learning aggregation algorithms: application to human activity recognition”. In: *In ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiCompISWC ’20)* (2020).

[10] F. Sattler, K.-R. Müller, and W. Samek. “Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2019), pp. 3710–3722.

[11] S. Liu et al. “Federated personalized random forest for human activity recognition”. In: *Mathematical Biosciences and Engineering* 19.1 (2022), pp. 953–971.

[12] C. Bettini et al. “Personalized Semi-Supervised Federated Learning for Human Activity Recognition”. In: *ArXiv* 2104.08094 (2021).

[13] H. Yu et al. “FedHAR: Semi-Supervised Online Learning for Personalized Federated Human Activity Recognition”. In: *IEEE Transactions on Mobile Computing* (2021).

[14] T. B. Shaik et al. “FedStack: Personalized activity monitoring using stacked federated learning”. In: *Knowl. Based Syst.* 257 (2022), p. 109929.

[15] R. Presotto et al. “Semi-supervised and personalized federated activity recognition based on active learning and label propagation”. In: *Personal and Ubiquitous Computing* 26 (2022), pp. 1281–1298.

[16] A. Tashakori et al. “SemiPFL: Personalized Semi-Supervised Federated Learning Framework for Edge Intelligence”. In: *ArXiv* 2203.08176 (2022).

[17] R. Presotto et al. “FedCLAR: Federated Clustering for Personalized Sensor-Based Human Activity Recognition”. In: *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (2022), pp. 227–236.

[18] R. Presotto et al. “Federated Clustering and Semi-Supervised learning: A new partnership for personalized Human Activity Recognition”. In: *Pervasive and Mobile Computing* (2022).

[19] Y. Chen et al. “FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare”. In: *IEEE Intelligent Systems* 35 (2019), pp. 83–93.

[20] W. Lu et al. “Personalized Federated Learning with Adaptive Batchnorm for Healthcare”. In: *IEEE Transactions on Big Data* (2021).

[21] X. Ma et al. “Tackling Personalized Federated Learning with Label Concept Drift via Hierarchical Bayesian Modeling”. In: *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*. 2022. URL: https://openreview.net/forum?id=RBPvr4Ehojh.

[22] H. M. et al. “Federated Learning of Deep Networks using Model Averaging”. In: *arXiv preprint arXiv:1602.05629* (2016).

[23] P. A. Flach. “Machine Learning - The Art and Science of Algorithms that Make Sense of Data”. In: 2012.

[24] S. van Dongen. “Graph clustering by flow simulation”. In: 2000.

[25] A. Azad et al. “HipMCL: a high-performance parallel implementation of the Markov clustering algorithm for large-scale networks”. In: *Nucleic Acids Research* 46 (2018), e33–e33.

[26] S. Kolouri et al. “Optimal Mass Transport: Signal processing and machine-learning applications”. In: *IEEE Signal Processing Magazine* 34 (2017), pp. 43–59.

[27] V. N. L. Duy and I. Takeuchi. “Exact statistical inference for the Wasserstein distance by selective inference”. In: *Annals of the Inst. of Stat. Mathematics* 75 (2021), pp. 127–157.

[28] C. G. Bezerra et al. “An evolving approach to data streams clustering based on typicality and eccentricity data analytics”. In: *Information Sciences* 518 (2020), pp. 13–28. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.12.022. URL: https://www.sciencedirect.com/science/article/pii/S0020025519311363.

[29] J. G. Saw et al. “Chebyshev Inequality With Estimated Mean and Variance”. In: *The American Statistician* 38 (1984), pp. 130–132.

[30] I. Škrjanc et al. “Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey”. In: *Inf. Sci.* 490 (2019), pp. 344–368.

[31] D. Anguita et al. “A Public Domain Dataset for Human Activity Recognition using Smartphones”. In: *The European Symposium on Artificial Neural Networks*. 2013.

[32] T. Sztyler and H. Stuckenschmidt. “On-body localization of wearable devices: An investigation of position-aware activity recognition”. In: *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (2016), pp. 1–9.

[33] A. Stisen et al. “Smart Devices are Different: Assessing and MitigatingMobile Sensing Heterogeneities for Activity Recognition”. In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems* (2015).

Paper VI

Contribution Prediction in Federated Learning via Client Behavior Evaluation

Ahmed A. Al-Saedi, Veselka Boeva, and Emiliano Casalicchio
In: Submitted for journal publication.

Abstract

Federated learning (FL), a decentralized machine learning framework that allows edge devices (i.e., clients) to train a global model with preserving data/client privacy, has become increasingly popular recently. In FL, a shared global model is built by aggregating the updated parameters in a distributed manner. To incentivize data owners to participate in FL, it is essential for service providers to fairly evaluate the contribution of each data owner to the shared model during the learning process. To the best of our knowledge, most of the existing solutions are resource-demanding and are usually run as an additional evaluation procedure. The latter produces an expensive computational cost for large data owners. In this paper, we present simple and effective FL solutions that show how the clients' behavior can be evaluated during the training process with respect to reliability, and this is demonstrated for two existing FL models, CA-FL and GP-FL, respectively. In the former model, CA-FL, the frequency of each client to be selected as a cluster representative and in that way to be involved in the building of the shared model is assessed. This can eventually be considered as a measure of the respective client data reliability. In the latter model, GP-FL, we calculate how many times each client changes a cluster it belongs to during FL training, which can be interpreted as a measure of the client's unstable behavior, i.e., it can be considered as not very reliable. We validate our FL approaches on three LEAF datasets and benchmark their performance to two baseline contribution evaluation approaches. The experimental results demonstrate that by applying the two FL models we are able to get robust evaluations of clients' behavior during the training process. These evaluations can be used for further studying, comparing, under-

standing, and eventually predicting clients' contributions to the shared global model.

1 Introduction

Federated Learning (FL) [1], a novel decentralized machine learning paradigm, enables cooperative model training through multi-party collaboration by transferring model parameters instead of transferring local private data during the training process. From a high-level point of view, FL allows multiple data holders (clients hereafter) to train a shared global model with a central server through parameter interaction without disclosing their own data [2, 3]. Although FL takes advantage of data that is distributed across different sources, not all data sources have equal value and contribute equally to FL settings [4–6]. Hence preserving high-quality data is a prerequisite to training a high-performing FL model by selecting influential clients and removing unneeded ones. In most current FL applications, although participating clients can contribute differently to the final overall model in each communication round, they all obtain the same FL model. In this context, effective evaluation of the quality of each client's data to the federated global model becomes an essential challenge in FL applications [7, 8]. Therefore, a key question in FL is how to fairly and inexpensively value the behavior of clients, using different data sources, for completing an FL application [9]. To this end, several methods have been applied in the literature for contribution measurement. Deletion approach (based on the conventional Leave-one-out (LOO) analyses [10]), and FL-Cohort [11] have provided solutions to the contribution evaluation problem. The deletion approach [10] works by retraining the federated model without the contribution of a given client's data and measuring the importance of retrained model changes to decide this client's contribution to the overall federated model. Instead of removing a single client at a time, FL-Cohort [11] eliminates multiple, similar clients at each FL global training round. Note that these approaches evaluate the contribution of the clients in a separate procedure additional to the training process producing the shared global model. Thus, on one hand, these techniques perform independent evaluation of the clients' contribution, leading to increase of the computational resources and time consumption. On the other hand, FL-Cohort measures the global imbalance only once prior to training the FL model and does not take into account the data dynamic during the training process. Evidently, we need new effective FL frameworks, which can fairly alleviate the resources and time required to evaluate the behavior of the clients involved as part of the training process.

The deletion and FL-Cohort approaches only focus on iteratively removing a single client or group of clients from global training at each global round to measure the contribution of the parties involved in the federation. Therefore, applying these approaches consumes considerable time and computational resources as the number

of participants increases, making them inapplicable in any practical situation. In addition, those FL solutions assume that the data does not change significantly at each client (e.g., no concept drift, new data is from the same distribution, etc.). However, this might be an unrealistic scenario.

Provoked by the above challenges and the limitations of previous research efforts, we present an efficient contribution evaluation approach inspired by the ideas of the Cluster Analysis-based Federated Learning (CA-FL) [12] and Group-Personalized FL (GP-FL) [13]. We have found that the CA-FL mechanism can effectively handle the problem of clients' bahaviour assessment. Specifically, CA-FL can evaluate the frequency of each client to be a cluster representative during the training process. The calculated score can be considered as an indirect measure of the data quality (reliability) of a particular client. In general, the CA-FL selects the top-performing (the most reliable) clients at each training round for the global training of the shared model in FL scenario. Furthermore, GP-FL provides with another technique that can evaluate the reliability of each client. In detail, GP-FL evaluates how many times the client has changed its cluster during FL training process. In practice, if the client changes the cluster it belongs to very often, this means the quality of data of this client is varying, because it does not have a stable position in the clustering structure determined on the base of clients' class distributions. This can also be interpreted as an indirect evaluation of the client data quality. Our work's contribution in addressing the above-mentioned challenges is summarized as follows:

1. We propose two approaches, inspired by CA-FL and GP-FL federated learning algorithms, to be used to evaluate each client's behavior during the training process, i.e., without requiring any extra computational cost.

2. The CA-FL method is capable of recognizing the clients that demonstrate reliable behavior during the training. It selects at each training round from each cluster of similar clients only the clients with top performing local models to be aggregated into the global model. This additionally contributes to achieving high performance in terms of accuracy and computational overhead in comparison with the conventional FL approaches.

3. The GP-FL method is capable of identifying clients that demonstrate unreliable behavior during the training. It counts how many time each client changes its cluster, and in that way the clients that have highly unstable behavior can be drop out the federation, which will contribute to the increase of performance of the overall model.

4. Our approaches can provide with realistic evaluations of the clients' behavior during the training process. These allow to discriminate between the clients that can eventually contribute to the performance of the FL model and ones that are not capable to do this.

5. We evaluate our CA-FL and GP-FL methods in different experimental scenarios on three LEAF datasets ([14–16]), which specifically emulate federated environments to show their robustness and effectiveness. The obtained experimental results demonstrate the robustness of our approaches in monitoring and evaluating clients' behavior during the training process.

The remainder of this paper is organized as follows. The related works are discussed in Section 2. The motivation, preliminaries and the study details are described in Sections 3 and 4, respectively. Section 5 explains the experimental setup, while Section 6 describes the experimental results and provides a discussion. The paper is concluded in Section 7.

2 Related work

Our discussion in this section is focused on three topics related to our work: (1) federated learning, (2) cluster-based federated learning solutions and (3) contribution evaluation in FL.

2.1 Federated learning

Federated learning has received much attention as a distributed training of machine learning techniques due to its high performance and ability to mitigate many privacy concerns and costs. Subsequently, several excellent research achievements have been introduced to face different issues encountered in FL, including heterogeneity challenges [17–20], privacy and security threats [21–23], communication overhead [12, 24–26], and convergence analysis [27–30]. Caldas *e*t al. [31] proposed LEAF – a benchmark in the exploration of experiments on federated settings, which are more practical than the simulated datasets. However, the research status on current FL challenges is still in continuous improvement.

2.2 Cluster-based federated learning solutions

The formulation of clustered FL has significant attention in many recent works to design efficient client selection strategies and overcome challenges caused by non-IID data in a federated setting, where the clients are assumed to be in various groups. Sattler *e*t al. [32] proposed a Clustered Federated Learning (CFL), clustering framework that deals with federated multi-task learning settings. The CFL recursively separates clients into clusters of similar clients based on the geometric properties of their local data distribution. Authors in [33] propose a similar algorithm named FedGroup, for which a similarity among client gradients with Euclidean distance of cosine similarity is quantified. Likewise, GP-FL [13] trains a set of global models, one per cluster

of clients. It splits clients into groups based on local data distribution with respect to the classes. Ghosh *et* al. proposed in [34] an iterative federated clustering approach (IFCA). It alternates between identifying the group membership of each client and optimizing the group models. The work in [35] proposed federated SEM (FeSEM) by solving a joint optimization and assigning the center for each client. More recently, soft clustering FL like FedEM [36] and FedSoft [37] have been proposed which allow clients to follow a mixture of multiple distributions. In HyperCluster [38], the authors propose to assign each client to the group whose model produces the lowest loss on its local data. CA-FL [12] propose that an FL model can reduce communication overheads via clustering analysis of the client updates. It identifies the most representative clients to communicate with the server to reduce communication overheads.

2.3 Contribution evaluation

Contribution evaluation in FL has been widely studied in recent years to indicate performance improvement (e.g., client selection, incentive allocation, etc.) [39]. However, it is challenging to measure participants' contribution under FL settings since participants keep their original data— local and private. The majority of the existing works in the domain can be distributed into two main categories: a reduction in the number of sub-model evaluations required and an acceleration of a single round of evaluation. The Shapley Value [5] and Leave-one-out (LOO), namely the deletion approach [10], are applicable valuation methods that use local gradients as a tool for client contribution evaluation. [9] utilizes data quality, data volume, and data collection cost to determine each participant's contribution levels. Yu *et*. al. [40] develop an incentive approach that compensates participants for their contributions to the federation by measuring according to self-reported data quantity and quality. Zhao *et*. al. [41] measure the participant's contributions according to the similarity between local updates and the global model. They assume that local updates similar to the global model are more valuable. In [42], the authors use self-reported local data size and bandwidth for measuring utility with the Cobb-Douglas function. Lyu *et* al. [43] proposed an approach that compares a pairwise similarity to let participants perform a mutual evaluation on each others' generated samples. Literature [44] proposed a concept of the contribution index to quantify participants' contributions by considering local datasets and machine learning models. In [45], authors study the problem of data valuation using Shapley value to assess each participant's contribution. Shyn *et* al. [39] introduces an empirical approach called Federated Client Contribution Evaluation through Accuracy Approximation (FedCCEA) to evaluate client contribution using data size in a feasible time. FL-Cohort [11] removes a group of similar clients in each FL training and calculates the contribution of each cluster separately. However, the aforementioned studies require FL participants to incur significant computation

and communication costs, making such approaches difficult to apply in practice. In contrast, we evaluate each client's behavior without incurring any additional computation costs. We compare our work with the deletion approach and FL-Cohort, mentioned above.

3 Preliminaries

This section introduces preliminary and essential background concerning FL and clustering approaches. First, we briefly present the theoretical basis of FL (Section 3.1). Secondly, we introduce the clustering techniques used in our methods, i.e., k-medoids and Markov clustering. Third, the Silhouette Index cluster validation method and the Eccentricity Analysis are explained (see Section 3.2). Finally, we describe the Kendall's Tau Rank correlation used to compare the performance of the FL frameworks investigated.

3.1 Federated learning basis

In the FL setup, a group of distributed clients cooperatively train a global model $\mathcal{M}$ under the supervision of a central server, without releasing their private data. Suppose W is a set of all clients and there is a subset of clients W_t, (i.e., $W_t \subset W$) participating in the FL system. Each client $w_i \in W_t$ has a private training dataset

$$\mathcal{D}_i = \{(x_1^{w_i}, y_1^{w_i}), \ldots, (x_{n_i}^{w_i}, y_{n_i}^{w_i})\},$$

where $x_j^{w_i} \in \mathbb{R}^P$ is the P-dimensional vector representation of the jth data sample, $y_j^{w_i} \in \{1, 2, \ldots, k\}$ (a multi-classification learning task) is the corresponding label of $y_j^{w_i}$, and n_i is the size of dataset $\mathcal{D}_i$ (i.e., $\mid \mathcal{D}_i \mid = n_i$).

The FL framework is depicted in Figure 1. The main definition of FL given by the naive FedAvg algorithm [1] can be summarized as follows:

1. The central server randomly selects a set of several clients W_t (e.g., from a set W) to obtain the appropriate accuracy level of the learned global model $\mathcal{M}$ for several communication rounds.

2. The server sends a shared machine learning model $\mathcal{M}_t$ to the selected clients (initialized to $\mathcal{M}_0$ at the beginning).

3. Each client $w_i \in W_t$, for $t = 0, 1, \ldots$, updates the model based on the local data $\mathcal{D}_i$ according to Eq. 1

$$\mathcal{M}_{t+1}^i = \mathcal{M}_t^i - \eta g(\mathcal{M}_t^i), \tag{1}$$

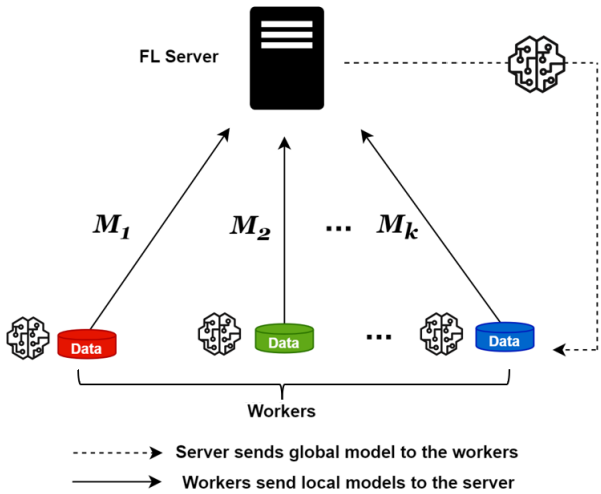


Figure 1: Framework of the federated learning.

where the superscript i indexes the client, η is the learning rate and $g(\mathcal{M}_t^i)$ refers to the stochastic gradient computed based on the local dataset $\mathcal{D}_i$ of the w_ith client, calculated as

$$g(\mathcal{M}_t^i) = \frac{1}{N_{w_i}} \sum_{(x_j^{w_i}, y_j^{w_i}) \in \mathcal{D}_i} \nabla \ell(x_j^{w_i}, y_j^{w_i}; \mathcal{M}_t^i) \cdot \tag{2}$$

4. Each client $w_i \in W_t$ sends the local model $\mathcal{M}_t^i$ to the central server.

5. The central server aggregates these model parameters to update the global model according to Eq. 3

$$\mathcal{M}_{t+1} = \mathcal{M}_t + \frac{\sum_{w_i \in W_t} p_i \mathcal{M}_{t+1}^i}{\sum_{w_i \in W_t} p_i}, \tag{3}$$

where p_i is the relative weight of client w_i.

6. The central server S sends the aggregated model $\mathcal{M}_{t+1}$ to each client $W_t \subset W$.

7. The above steps are iterated within a predetermined number of multiple global rounds T.

3.2 Clustering approaches

Clustering techniques allow to group of data points into k disjoint groups without sufficient prior information [46]. Many clustering methods compute the similarity between all pairs of data points. In specific, the clustering algorithms work by minimizing within-group variances while maximizing them between groups relying on a clustering criterion.

3.2.1 k-medoids

k-medoids method [47] is a variant of k-means that involves actual data points as the cluster centroids instead of derived ones. It is summarised as follows: given an integer k and an initial finite set containing n clients $W_t = \{w_1, ..., w_n\}$, $1 \leq k \leq n$.

The problem of the k-medoids algorithm is to find a set of objects (here clients) as medoids $O = \{o_1, \ldots, o_k\}, |W_t| = k$ as the centre of a cluster [48] from within the set W_t, such that W_t can be partitioned into k non-empty disjoint clusters $c_1, \ldots, c_k$ such that the overall sum of distances from every client w_i in the set to its nearest medoid o_j associated with its cluster is minimized.

$$\min_{O \subset W_t} \left\{ \sum_{i=1}^{n} \min_{j=1,\ldots,k} d(w_i, o_j), \mid O \mid = k \right\}. \tag{4}$$

In our CA-FL algorithm we use k-medoids for partitioning the available clients into groups of similar clients w.r.t. their local model updates. The Euclidean distance as a dissimilarity measure is adopted in CA-FL.

3.2.2 Markov Cluster

Markov Cluster (MCL) is an iterative process consisting of two steps called expansion and inflation, which are applied to a stochastic "Markov" matrix [49]. The expansion step is responsible for making the flow to connect different regions of the graph and the inflation step is responsible for both strengthening and weakening of current region. Suppose we have a matrix $M(n \times n)$ of the smoothed vectors $\hat{p}_t(w_i)$ which represents a distance matrix of each pair of clients $w_i \in W_t$ and generated by rescaling each column of the adjacent matrix $A(n \times n)$ from a graph. This matrix is then passed as an input parameter to the predicted function of a Markov clustering method as given in Eq. 5.

$$M(i,j) = \frac{Adj(i,j)}{\sum_{k=1}^{n} Adj(k,j)}. \tag{5}$$

where Adj is the adjacent matrix with self-loops. The expansion and the inflation steps are repeated until the matrix M is convergent [50]. In our GP-FL algorithm, a Markov cluster algorithm (MCL) is proposed to group a set of clients based on their empirical probability vectors. An initial clustering $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk}\}$ of the

clients is generated. The Wasserstein distance will be used to measure the distance between empirical probability distribution vectors.

3.3 Silhouette Index

*S*ilhouette score $s(i)$ is an unsupervised method for interpretation and validation of consistency within clusters of data points [51]. It is a combination of two ideas: cohesion (how close a data point is to other points in its own cluster) and separation (how far a data point is from points in other clusters) of the data point.

For data point $i \in C_j$ (data point i in the cluster C_j), we compute

$$a(i) = \frac{1}{\mid C_j \mid -1} \sum_{\substack{j \in C_j \\ i \neq j}} d(i,j), \tag{6}$$

Here $a(i)$ represents the average dissimilarity of data point i from all other data points in in the same cluster C_j.

Then we define the average dissimilarity of data point i to all data points in some cluster C_r (where $r \neq j$). For each data point $i \in C_j$, we define

$$b(i) = \min_{r \neq j} \frac{1}{\mid C_r \mid} \sum_{j \in C_r} d(i,j), \tag{7}$$

to be the average dissimilarity of data point i from all data points in any other cluster (i.e., in any cluster of which i is not a member). The silhouette value $s(i)$ of one data point i, is calculated as:

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i), \\ 0 & \text{if } a(i) = b(i), \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i), \end{cases}$$

which can be rewritten as

$$s(i) = (b_i - a_i)/\max\{a_i, b_i\}. \tag{8}$$

3.4 Eccentricity Analysis

AutoCloud [52] is an evolving clustering algorithm based on eccentricity data analytics principles [53] to autonomously create groups and merge them. Alike to AutoCloud model, eccentricity analysis was applied in our proposed GP-FL solution to update a dynamic clustering of clients. On this basis, the eccentricity ξ^j of a client w_i with respect to cluster C_j is [52]:

$$\xi^j(w_i) = \frac{1}{n_j} + \frac{(\mu_i^j - \hat{p}_i)^T(\mu_i^j - \hat{p}_i)}{n_j \sigma_i^j}, \tag{9}$$

where n_j is the size of C_j, $w_i \in C_j$, $\hat{p}_i$ is empirical probability vector held by the client w_i, and μ_i^j and σ_i^j are the mean and variance, respectively. In Eq. 10, the condition is verified to determine whether a particular client is considered to belong to a cluster C_j using Chebyshev inequality [54] as follows

$$\xi^j(w_i) \leq \upsilon_j \text{ and } \upsilon_j = (m^2 + 1)/2n_j, \tag{10}$$

where m is a pre-defined parameter that straightly impacts the clustering evaluation, and υ_j is the threshold associated with the corresponding cluster C_j. $m = 3$ is broadly set as a standard value, thus achieving appropriate results for various datasets [55].

3.5 Kendall's Tau Rank Correlation

Kendall's Tau Rank Correlation, also commonly known as Kendall correlation or Kendall's tau, is one of the standard correlation methods used to measure the consistency among ranked attributes. The Kendall's tau correlation coefficient can take values from -1.0 to 1.0. The positive values close to 1.0 represent the positive correlation between ranked variables, the negative values close to -1.0 indicate a negative relation, and the values close to zero imply no correlation [56]. If two rankings x and y are produced on the given clients, then Kendall's tau $\mathcal{T}$ can be calculated using Eq. 11 [57].

$$\mathcal{T} = \frac{A - D}{\sqrt{(A + D + T_x)(A + D + T_y)}}, \tag{11}$$

where, A is the number of pairs in agreement, D is the number of pairs in disagreement, T_x is the number of pairs tied w.r.t. x, and T_y is the number of pairs tied w.r.t. y.

We evaluate and compare the FL baseline approaches by computing Kendall's tau correlations between the ranking scores of the CA-FL, GP-FL, Deletion approach, and FL-Cohort.

4 Methodology

4.1 Problem Statement

The resource consumption issue is considered to be one of the fundamental burdens for FL paradigm due to limited network resources, and computing capabilities, which degrades learning performance in FL settings and results in a longer training time [58, 59]. Meanwhile, the evaluation of the client's influence on the federated model has attracted a lot of attention in recent years due to the need of providing an incentivized mechanism. However, most of the proposed solutions are procedures requiring to run

separately from the distributed training process that additionally consumes considerable recourse and time [10, 60].

Assume a set of clients, W, forms a federation for training a global model. Each client w_i, $w_i \in W$, trains at each training round t, $t = 0, 1, \ldots$, an ML model $\mathcal{M}_t^i$ locally on its data set and sends the model parameters to the central server, where a unified global model $\mathcal{M}_t$ is created by averaging the model parameters received from all the clients. We assume that all clients participate in all rounds of the federation. In this context, we are interested in evaluating the clients' behavior during the distributed model training process. The aim is to quantify each client behavior with respect to reliability, interpreted, e.g., in the context of demonstrating high performing local model for most of the training rounds. Those evaluations should provide an opportunity for comparing, understanding, and eventually predicting clients' contributions to the shared global model.

4.2 Deletion Approach

Naively, the deletion approach [10] aims to assign the contribution to a participant client by calculating the model performance change when the client is erased from the set of participants selected in FL learning. Thus, this approach can exclude the contribution of certain clients to the overall global model. Suppose we evaluate the effect of the client w_i on the model predictions, the influence measure [10] can be formulated as:

$$influence^{-i} = \frac{1}{n} \sum_{j=1}^{n} |\hat{y}_j - \hat{y}_j^{-i}|, \tag{12}$$

where n is the size of the dataset, $\hat{y}_j$ is the prediction on jth instance made by the model trained on all data, and $\hat{y}_j^{-i}$ is the prediction on jth instance made by a model trained with the ith client omitted.

Given the initial global model $\mathcal{M}_0$ and a target client u, whose private data is required to be erased from the FL model, deletion approach steps are summarized into Algorithm 1. In line 2, the server sends the global model $\mathcal{M}_t$ to all selected clients except target client $w_i \in W_t \setminus u$ from global training. The global model $\mathcal{M}_t$ is sent to a set of participants, and each client $w_i \in W_t \setminus w_u$ optimizes the model locally according to Eq. 1. Then, each client uploads the initial update to the central server (line 4). Later, the server averages the model parameters uploaded to update the global model $\mathcal{M}_t$ according to Eq. 3. In line 7, an unlearned model $\hat{\mathcal{M}_{t+1}}$ is returned for the client w_u during t round of training. Finally, in line 8, we calculate the influence of the client w_u on the global model according to Eq. 12.

Algorithm 1 Deletion Approach

Output: DELETION APPROACH procedure unlearned model $\hat{\mathcal{M}}_{w_i}$ and Influence ^{i}

procedure DELETIONAPPROACH($\mathcal{M}_0$, $W_t \subseteq W$, Target client with index u, T number of iterations)
  $\forall\ w_i \in W_t$ except client w_u, SEND(w_i, $\mathcal{M}_t$)
  for each client $w_i \in W_t \setminus w_u\ \ in\ \ parallel$ **do**
    $\mathcal{M}^i_{t+1} \leftarrow$ CLIENTUPDATE($i, \mathcal{M}_t$)
  end for
  $\hat{\mathcal{M}_{t+1}} = \sum_{w_i \in W_t \setminus u} \frac{n_i}{n} \mathcal{M}^i_{t+1}$ following Eq. 3 ▷ Unlearned model
  Influence $^{-u} = \frac{1}{n} \sum_{j=1}^{n} |\hat{\mathcal{M}}_j - \hat{\mathcal{M}_j^{-u}}|$ ▷ According to Eq. 12
  return Influence ^{i} for $\forall\ w_i \in W_t$
end procedure
function CLIENTUPDATE($(w_i, \mathcal{M}_t)$) ▷ Local update
  while True **do**
    RECEIVE(w_i, $\mathcal{M}_t$) ▷ Receive the global model
    LOCALTRAINING($w_i, \mathcal{M}_t$)
    $\mathcal{M}^i_{t+1} \leftarrow \mathcal{M}^i_t - \eta g^i_t$ ▷ Following Eq. 1
    SEND(i, $\mathcal{M}^i_{t+1}$) ▷ Send the local model to the server
  end while
end function

4.3 FL-Cohort

In contrast to the deletion approach, the FL-Cohort delete multiple, similar clients on each round to quantify the contribution of a client to the FL setting. Algorithm 2 presents the FL-Cohort process. In FL-Cohort, firstly, there are a bunch of clients $W_t \in W$. Each client $w_i \in W$ sends a vector $\vec{v_i} = [N_i^1, \ldots, N_i^Q]$, where Q is the number of classes and N_i^q indicates the number of samples for class q held by client w_i to measure the imbalance (line 2 in the Algorithm 2). On the server side, $\vec{V} = [\sum_i N_i^1, \ldots, N_i^Q]$ and the overall number of samples N are calculated using SA protocol. Then, the server announces the global measurement to all clients (lines $3 \sim 4$ in the Algorithm 2). Subsequently, $\vec{V}$ and N are distributed to each of the clients for calculating data imbalance within the cohort.

In FL-Cohort, data imbalance metrics, namely Label Imbalance (LI), Label Distribution Imbalance (LDI), and Quantity Imbalance (QI) are computed at each local party. In brief, these metrics are defined and computed as follows:

Label Imbalance (LI): LI of each client w_i is defined as follows:

$$LI_i = \frac{max_p\{N_i^p\}}{min_p\{N_i^p\}} \tag{13}$$

Therefore, LI_i is the ratio of the majority class size and minority class size for a local client i.

Label Distribution Imbalance (LDI): $\vec{V}$ and $\vec{v_i}$ are used to quantify imbalance in terms of label distribution. So, LDI is given as follows:

$$LDI_i = 1 - \frac{\vec{v_i} \cdot \vec{V}}{||\vec{v_i}|| \cdot ||\vec{V}||} \tag{14}$$

The cosine distance is utilized to measure of similarity between two vectors. As a result, LDI_i= 0 refers to perfect balance, whereas LDI_i= 1 confirms a very high imbalance.

Quantity Imbalance (QI): QI is defined as the ratio of the number of samples at client i and the mean number of samples overall i clients as:

$$QI_i = N_i / \frac{\sum_{l=1}^{I} N_l}{I} \tag{15}$$

Each client is represented by a vector $\vec{l_i} = [LI_i, LDI_i, QI_i]$. Then, the updated vector $\vec{l_i}$ of each client is sent back to the server, line 6 in Algorithm 2. Eventually, the server applies the k-means algorithm on the set of representations. As a result, k_t clusters of clients with similar imbalances are obtained (line 8 in Algorithm 2). In order to quantify the contribution of each client, FL-Cohort approach applies a leave-x-out (LXO) analysis, which iteratively deletes multiple and similar clients on each global training. Specifically, during LXO analysis, a model $\mathcal{M}_c^{LXO}$ leaves out the respective cluster c_j is trained, line 10. Lastly, the contribution of each cluster $c_j \in C_t$ is defined using a weighted F1 scores as follows [60]:

$$Cr_{c_j} = F1(\mathcal{M} - D^{Test}) - F1(\mathcal{M}_c^{LXO} - D^{Test}) \tag{16}$$

To assign a contribution cr_i for each client $w_i \in c_j$, simply the contribution of their respective cluster c_j is divided by the number of clients W_{c_j} belonging to c_j, line 12.

Steps (lines 2∼12 in the Algorithm 2) occur prior to the starting of FL training.

Algorithm 2 FL-Cohort framework

Output: The FL-Cohort procedure Contribution $Cont_{w_i}$ of each $w_i \in c_j$ $(j = 1, 2, \ldots, k)$

procedure FL-Cohort($\mathcal{M}_0$, $W_t \subseteq W$, Target cohort c_i, T)
  $\forall\, w_i \in W_t$, Send(w_i, $\vec{v_i}$)
  $\vec{V} \leftarrow [\sum_i N_i^1, \ldots, N_i^Q]$
  $\forall\, w_i \in W_t$, Send(w_i, $\vec{V}$, N)
  for each $w_i \in W_t$ **do**
    $\vec{l_{i+1}} \leftarrow$ clientUpdate(i) ▷ Defined in Alg. 1
  end for
  $C_t \leftarrow$ k-means(k_t, $\{\vec{l_i} \mid w_i \in W_t\}$, W_t)
  $\bar{C}_t \leftarrow \forall\, c_{tj} \in C_t, \{C_t \setminus c_{tj}\}$
  $\mathcal{M}_{c_j}^{LXO} = \sum_{w_i \in \bar{C}_t} \frac{n_i}{n} \mathcal{M}_{t+1}^{i}$
  $Cr_{c_j} \leftarrow (\mathcal{M}, D^{Test}, \mathcal{M}_c^{LXO})$
  $Cr_{w_i} \leftarrow (Cr_{c_j}, W_{c_j})$
  $t \leftarrow 0$ ▷ *FL training*
  while $t \leq T$ **do**
    $t \leftarrow t + 1$
    $\forall\, w_i \in W_t$, Send(w_i, $\vec{V}$, N)
    for each client $w_i \in W_t$ *in parallel* **do**
      $(\mathcal{M}_{t+1}^{i}, \vec{l_i}) \leftarrow$ clientUpdate($i, \mathcal{M}_t$)
    end for
  end while
end procedure
return cohort model $\mathcal{M}_{c_j}^{LXO}$

prior to FL training

4.4 Proposed Approach

In this section, we present a simple and effective FL framework for clients behavior evaluation based on two FL algorithms: CA-FL [12] and GP-FL [13]. Namely in this paper, we utilize the advantages of CA-FL and GP-FL approaches to predict FL clients' contributions to the built global model by evaluating their behavior. The main idea is to train a global model with the participation of a given set of clients and simultaneously evaluate the clients' behavior (e.g., reliable versus unreliable) as a part of the training process.

For example, the CA-FL [12] algorithm calculates during the training process for each client the frequency of being selected as a cluster representative from the clients distributed in the same cluster with it based on the similarity of their models' parameters. At each training round, a client with a top-performing model is selected

from each cluster. Based on this, each client will be assigned a score that can be interpreted as a measure of the client's reliability. Certainly, the higher score implies a higher influence (contribution) to the built global model. These scores can be used for ranking the clients with respect to their reliability, i.e., contribution to the global model. This ranking can be used in different ways, e.g., for selecting a given percentage of the top-ranked clients and training a new global model on these clients' data that ensures higher performance. In that way, we can minimize the impact of non-reliable clients on the shared global model. Based on the produced ranking, it is also possible to split the clients into three disjoint groups representing respectively top contributing clients, ones with comparatively good contributions, and finally ones with very modest, almost no contributions. Three separate FL models can be built on these three groups of clients. This will contribute to the fairness of the FL scheme by providing each client with a model that best reflects its contribution to the federation in general. In addition, the availability of three different FL (global) models will also facilitate the scenarios when a client wants to withdraw its participation from the federation: namely, only the affected model should be retrained in this case.

Opposite to the CA-FL, the GP-FL [13] method provides an opportunity for identifying clients exposing unstable behavior during the training process. The GP-FL initially splits the clients into a few clusters based on the similarity between their class distributions. This clustering is dynamically maintained during the training process by evaluating at each training round the clients' assignment among the clusters, and if necessary some are re-assigned to the new clusters. To evaluate the clients' behavior, we calculate how many times each client changes a cluster it belongs to during the distributed training process, which can be interpreted as a measure of the instability of the client's data, i.e., it can be translated to the client's unreliability.

Note that different criteria can be applied to split the clients into three (or even more) groups (highly reliable clients, clients with good level of reliability, and unreliable clients), with respect to the scores produced by CA-FL or GP-FL, e.g., splitting thresholds can be defined based on the analysis of the scores or binning can be applied. Binning is a technique used to discretize a continuous variable's values into bins (groups) [61]. In our experiments, we use this method to split clients into groups, both for CA-FL and GP-FL. In addition, the calculated scores can be normalized, e.g., by applying min-max normalization, to facilitate in this way the interpretability of the scores. The performance of the two algorithms, CA-FL and GP-FL, is studied under different experimental scenarios and they are additionally compared with two existing approaches (deletion [10] and FL-cohort [11]) that are specially devoted to the clients' contribution evaluation in FL context. The correlations between the rankings produced by our algorithms and those based on the deletion and FL-cohort approaches are studied by Kendall's tau, see Section 5.

The two FL methods, discussed above, are described in Section 4.4.1 and Section 4.4.2, respectively.

4.4.1 CA-FL

Algorithm 3 contains the pseudo-code of the proposed CA-FL phases [12]. In the Initialization phase (lines 2 – 6 in Algorithm 3) the central server sends the global model $\mathcal{M}_t$ to a subset of clients W_t ($W_t \subset W$) (line 3). Each client $w \in W_t$ receives the global model $\mathcal{M}_t$ and optimizes model parameters locally (line 4), sending back the initial update to the server according to Eq. 1. At line 5 the server aggregates the model parameters uploaded by the selected clients W_t to update the global model $\mathcal{M}_t$ through the FedAVG algorithm following Eq. 3. Finally, the local updates $\{m_w^t \mid w \in W_t\}$ are analyzed by using k-medoids approach. Thus, the clients are partitioned into k clusters of similar updates, i.e. $C^t = \{C_1^t, C_2^t, \ldots, C_k^t\}$.

In the Iteration phase (lines 7 – 14 in Algorithm 3), firstly, the server ranks the clients in each cluster $C_i^t, i = 1, 2, \ldots, k$ and selects the top-ranked client, i.e., the representatives (line 9). In line 10, the frequency for each client being of a cluster representative is calculated. The selected representatives form a new set of clients $W^{t+1} = \{w_1^{t+1}, w_2^{t+1}, \ldots, w_k^{t+1}\}$, where $k << \mid W^0 \mid$. Then, the server sends the global model M_t to each representative $w \in W^{t+1}$ for new global training. Each representative $w \in W^{t+1}$ receives the global model $\mathcal{M}_t$ and sent back m_w^{t+1} update according to Eq. 1 to the server after optimizing its parameters locally. The Server averages the received parameters $\{m_w^{t+1} \mid w \in W^{t+1}\}$ uploaded by the representatives to update the global model $\mathcal{M}_{t+1}$. Finally, the server applies the Silhouette score (SI) to assess whether each representative is still well tied to its current cluster (Eq. 8), i.e. the updated clustering C^{t+1} of the clients in W^0 is produced. Steps at lines 8 – 14 are iterated until the predetermined number of communication rounds T is completed.

4.4.2 GP-FL

The details of our GP-FL scheme [13] are shown in the Algorithm 4. In the Initialization phase (lines 2 – 9 in the Algorithm 4) the server initializes model $\mathcal{M}_t$ and the set of clients $W_t \subset W$. Then (line 3), the server sends the new global model $\mathcal{M}_t$ to each client $w_i \in W_t$. Each client receives the global model and optimizes its parameters locally, executing the CLIENTUPDATE function (line 4). In this function, the client produces the $\mathcal{M}_t^i$ initial update alongside a vector $\hat{p}_t(w_i)$ that represents the empirical probabilities of the classes distribution and sent back to the Server. To avoid issues with extreme values, such as 0 or 1, each empirical probability vector $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k)$ can be smoothed by applying Laplace correlation (line 5) which is expressed as:

$$\hat{p}_i(D) = \frac{n_i + 1}{\mid D \mid + k}, \tag{17}$$

where D is a set of labeled examples, n_i is the number of examples in class C_i, and k (the number of classes) is added to ensure that the posterior probabilities are never zero [62]. The smoothed vectors $\hat{p}_t(w_i)$, for $w_i \in W_t$, are used to create a distance

matrix (line 6) used by a Markov clustering algorithm executed by the server. To compute the distance matrix [63, 64], we used Wasserstein distance. That distance allows us to measure the similarity between class probability estimations of each pair of clients. In this way, the clients who exhibit similar activity patterns will be grouped together by applying Markov Cluster, as described in Section 3.2.2. The Markov Cluster algorithm produces groups of clients with similar empirical probability vectors, i.e. an initial clustering $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk}\}$ of the clients is created. For each cluster $C_{tj} \in C_t$, $(j = 1, 2, \ldots, k)$, mean data vector μ_i^j and aggregated variance σ_i^j are calculated (line 7). Moreover, a global group model $\mathcal{M}_t^j$, is built by averaging over the model parameters of the clients assigned to C_{tj} (line 8). The Server aggregates the parameters $\{\mathcal{M}_t^i \mid w_i \in W_t\}$ uploaded by the selected clients W_t to update the global model $\mathcal{M}_t$ through the FedAvg algorithm according to Eq. 3.

In the Iteration phase (lines 10 – 23) the server sends each group global model $\mathcal{M}_t^j$, $(j = 1, 2, \ldots, k)$ to its group of clients C_{tj} (line 13). Each client $w_i \in C_{tj}$, executing the CLIENTUPDATE function (line 14), receives the group global model $\mathcal{M}_t^j$ and optimizes its parameters locally, sending back to the server the local update $\mathcal{M}_{t+1}^i$ and the empirical probability vector $\hat{p}_{t+1}(w_i)$. Then the server, at line 14, updates the existing empirical probability vector $\hat{p}_{t+1}(w_i)$ by taking the average of it with the information provided by the previous data batch, i.e. $\hat{p}_t(w_i)$, and it applies Laplace smoothing to each vector $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_t \mid$ (line 15). At line 16, the Server adapts the clients' grouping C_t to the current empirical probability vectors $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_t \mid$, by invoking eccentricity score $\xi^j(w_i)$ following Eq. 9, $(j = 1, 2, \ldots, k)$ which assesses whether each client $w_i \in C_{tj}$ is still adequately tight with its current cluster, the one it was assigned at the previous round (t) (Cluster Optimization). The ECCENTRICITYSCORE pseudocode is in Algorithm 5 and is described in Section 4.4.3. For each cluster $C_{t+1j} \in C_{t+1}$, mean data vector μ_i^j and aggregated variance σ_i^j are calculated, considering the current grouping of the clients and also using the current empirical probability vectors $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_{t+1} \mid$. These values of μ_i^j and σ_i^j will be needed at the next round to calculate the clients' eccentricity scores w.r.t. the current clusters. The updated clustering C_{t+1} is produced, and the clusters in C_{t+1} may contain different clients from the clusters in C_t. These steps of the *iteration* phase are repeated until a certain number of training rounds T is reached.

4.4.3 Eccentricity score function

The function firstly computes the Eccentricity score $\xi^j(w_i)$, using Eq. 9, for each client in a cluster (line 4). If $\xi^j(w_i)$ is below the threshold $\upsilon_j(t)$ following Eq. 10, the client does not change its cluster C_{tj} (lines 5 and 6). Otherwise, if $\xi^j(w_i) > \upsilon_j(t)$ then we calculate $\xi(w_i)$ for the other clusters, i.e. for each $C_{tl} \in C_t \setminus C_{tj}$, and will assign the client w_i to the cluster for each $\xi(w_i) < \upsilon_l(t)$. In case this is true for more

than one cluster we will assign the client to the cluster for which the score is lowest. Finally (line 11), If $\xi(w_i) > \upsilon_l(t)$ for each cluster $C_{tl} \in C_t \setminus C_{tj}$, then this client w_i will give the start of a new singleton cluster, which means that this client w_i cannot be assigned to any existing cluster in C_t. Note that $k_{(t+1)} \geq k_t$, where $k_{(t+1)} =\mid C_{t+1} \mid$, since new singleton clusters may appear due to the updating operation.

5 Experimental setup

This section presents the experimental settings.

5.1 Dataset

We conduct experiments on three LEAF datasets [31] which are tailored to federated learning evaluation: Synthetic Dataset [14], Federated Extended MNIST (FEMNIST for short) [15] and CelebA [16].

- Synthetic: This dataset modifies the synthetic dataset presented in [14] to make it more challenging for current meta-learning methods. For this dataset, we use logistic regression as a basic ML model.

- FEMNIST [15]: This dataset is built by partitioning the data on the Extend MNIST based on the writer of the digit/ character. For this dataset, we use a convolutional neural network (CNN) containing two convolutional layers and three fully connected layers.

- CelebA [16]: This dataset is a large-scale face attributes' dataset with celebrity pictures. For this dataset, we have used the same CNN network as for the FEMNIST dataset.

Algorithm 3 CA-FL framework

Output: The CA-FL procedure the frequencies of representatives $freq_t$ for T iterations

procedure CA-FL($\mathcal{M}_0$, $W_t \subseteq W$, k_t, T)
Initialization Phase
  $t \leftarrow 0$
  $\forall\, w_i \in W_t$, the server exec SEND(w_i, $\mathcal{M}_t$)
  Each $w_i \in W_t$ exec CLIENTUPDATE($i, \mathcal{M}_t$) ▷ Defined in Alg. 1
  $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$ ▷ According to Eq. 3
  $C_t \leftarrow$ KMEDOIDS($k_t, \{\mathcal{M}^i_{t+1} \mid w_i \in W_t\}, W_t$)
Iteration Phase
  while $t \leq T$ **do**
    $t \leftarrow t + 1$
    $W_t \leftarrow$ SELECTTOPRANKED(p, C_t)
    $freq_t \leftarrow$ FREQUENCY(W_t) ▷ Calculate frequency for each client of being of a cluster representative
    $\forall\, w_i \in W_t$, the server exec SEND(w_i, $\mathcal{M}_t$)
    Each $w_i \in W_t$ exec CLIENTUPDATE($w_i, \mathcal{M}_t$) ▷ Defined in Alg. 1
    $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$
    $C_{t+1} \leftarrow$ SILHOUETTE(k_t, C_t, W_t)
  end while
end procedure
function SILHOUETTE((k_t, C_t, W_t)) ▷ Check whether each $w_i \in W_t$ still belongs to its cluster
  for $w_i \in W_t$ **do**
    for $j = 1, 2, \ldots, k$ **do**
      compute $s(w_i)$ ▷ According to Eq. 8
    end for
    if $s(w_i) < 0, \forall j \in \{1, 2, \ldots, k\}$ **then**
      $k_t \leftarrow k_t + 1$
      $C_{tk_t} \leftarrow w_i$
    else
      Assign w_i to the nearest cluster C_{tj}
    end if
  end for
  $\forall\, C_{tj}$ $(j = 1, 2, \ldots, k)$ recompute the cluster center
  return C_{t+1} ▷ The new set of clusters
end function

Algorithm 4 GP-FL framework

Output: The GP-FL procedure the number of recurrences of clients $recur_t$ that changes his/her cluster

procedure GP-FL($\mathcal{M}_0, W_t \subseteq W, T$)

Initialization Phase

$t \leftarrow 0$

$\forall\, w_i \in W_t$ the server exec SEND($w_i, \mathcal{M}_t$)

Each client $w_i \in W_t$ exec CLIENTUPDATE($i, \mathcal{M}_t$) ▷ Defined in Alg. 1

$\hat{p}_{t+1}(w_i) \leftarrow$ LAPLACESMOOTHINGCORRELATION($\hat{p}_{t+1}(w_i)$)

$C_t \leftarrow$ MARKOV($\{\hat{p}^i_{t+1} \mid w_i \in W_t\}, W_t$)

$\forall\, C_{tj}\ (1 \le j \le k)$ compute μ_i^j and σ_i^j

$\forall\, C_{tj}\ (1 \le j \le k)$ compute $\mathcal{M}_{t+1}^{C_{tj}} = \sum_{w \in C_{tj}} \frac{n_w}{n} \mathcal{M}_{t+1}^w$

$\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}_{t+1}^i$

Iteration Phase

while $t \le T$ **do**

$t \leftarrow t + 1$

$\forall\, w_i \in C_{tj}$ the server exec SEND($w_i, \mathcal{M}_t^j$)

Each client $w_i \in C_{tj}$ exec CLIENTUPDATE($w_i, \mathcal{M}_t^j$) ▷ Defined in Alg. 1

$\forall \hat{p}_{t+1}$, the server update the existing $\hat{p}_{t+1}(w_i)$

$\hat{p}_{t+1}(w_i) \leftarrow$ LAPLACESMOOTHINGCORRELATION($\hat{p}_{t+1}(w_i)$)

$C_{t+1} \leftarrow$ ECCENTRICITYSCORE(k_t, C_t, W_t)

$recur_t \leftarrow$ RECURRENCE(W_t) ▷ *C*alculate recurrence for each client that changes its cluster

$\forall\, C_{t+1j}\ (1 \le j \le k)$ compute μ_i^j and σ_i^j

$\mathcal{M}_{t+1}^{C_{tj}} = \sum_{w \in C_{tj}} \frac{n_w}{n} \mathcal{M}_{t+1}^w$ following Eq. 3

$\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}_{t+1}^i$

return (C_{t+1}, k_{t+1})

end while

end procedure

Table 1 summarizes the statistics of the three used datasets. Note that the datasets are partitioned among the clients in a non-IID manner, representing the most common scenario in FL settings. For all datasets, we use the default division in Pytorch. The entire test dataset is used as a common test dataset for model performance evaluation, and half of the test dataset is used as unlabeled data on the central server. To reproduce the experimental setup of federated learning, we need to partition the dataset to simulate individual clients. Each client may have fewer (or even no) data

Table 1: Statistics of the used LEAF datasets.

Dataset	Number of devices	Samples per devices	
		Mean	Stdev
Synthetic	1,000	107,55	213,22
FEMNIST	3,500	226.26	89.12
CelebA	9,343	21.44	7.63

samples in some classes.

Algorithm 5 EccentricityScore

Output: updated k_t and C_t

procedure ECCENTRICITYSCORE(k_t, C_t)
  for $j = 1, 2, \ldots, k_t$ **do**
    for $w_i \in C_{tj}$ **do**
      compute eccentricity score $\xi^j(w_i)$ ▷ According to Eq. 9
      if $\xi^j(w_i) < \upsilon_j(t)$ **then**
        $w_i \in C_{tj}$ ▷ According to Eq. 10
      else if $\xi^j(w_i) > \upsilon_j(t)$ **then**
        calculate $\xi^l(w_i)$ for each $C_{tl} \in C_t \setminus C_{tj}$
        Assign w_i to C_{tl} for which $\xi^l(w_i) < \upsilon_l(t)$ and $\xi^l(w_i)$ is lowest ▷ In case there is more than one cluster
      else if $\xi(w_i) > \upsilon_l(t)$ for each $C_{tl} \in C_t \setminus C_{tj}$ **then**
        w_i represents a new singleton cluster
      end if
    end for
  end for
end procedure
return (C_t, k_t)

The baseline algorithms used in this paper to assess the robustness of our proposed methods are explained in Section 4.2 and Section 4.3, respectively. Accuracy and F1 score are the classification metrics used to evaluate the performance of our methods. The reliability/ unreliability of the clients is evaluated by ranking them based on the scores calculated by our FL algorithms. This ranking is also used to correlate the performance of our approaches with the baselines applying Kendall's tau.

5.2 Experimental details

We implement all the code using PyTorch with various settings, including client local data training with SGD. Two ML models are used in our experiments: logistic regression (LR) and Convolutional neural network (CNN). We considered two different ML tasks on the three used federated benchmark datasets: logistic regression on a synthetic dataset and image classification for FEMNIST and CelebA datasets. The image classifier in all experiments is a standard CNN, which consists of two convolutional layers and one fully connected (FC) layer. Furthermore, for synthetic dataset, we use LR. Table 2 contains some necessary notations and values of different hyper-parameters used in this paper.

Table 2: Default notation, definitions, and corresponding values of different hyper-parameters.

Notation	Definition	Value
W	Total number of clients	100
W_t	Fraction of clients randomly selected	0.2
η	Learning rate	0.05
T	Number of communication rounds	100
$\mathcal{D}_i$	The local data in client w_i	---
n_i	The size of data in client w_i	---
n	Total size of data	---
$\mathcal{M}_t$	The global model at tth round	---
$\mathcal{M}_t^i$	The local model of client w_i at round t	---
g_t^i	The gradients computed	---
$s(.)$	Silhouette Index score	----

6 Evaluation and results

In this section, we elaborate and discuss the results of experiments. We compare the performance of our two methods, CA-FL and GP-FL, with that of the two baselines, Deletion approach and FL-Cohort, in different experimental settings in terms of their ability to evaluate the client's contributions to the shared global model. In our experiments, we use the non-IID setting, and in this setting, a federated learning scenario with 20 participating clients is simulated in the three used datasets described above.

6.1 Comparison of methods' performance

Each of the four studied approaches produces a ranking of the clients scoring their influence on the global model by applying its own evaluation mechanism. To compare the performance of the four approaches we have initially applied each approach to each of the three datasets used to calculate a single evaluation for each client. Notice that the meaning/interpretation implemented in the calculated clients' evaluations differs for different approaches. For example, the two baseline approaches

consider those as measures of clients' contribution (influence) to the global model. While the CA-FL (refer to Algorithm 3 (line 10)) and GP-FL (see Algorithm 4 (line 17)) interpret the calculated scores as evaluations of clients' behavior (reliable versus unreliable), i.e., their ability to have a positive/negative impact to the global model eventually.

Table **??** shows the calculated contributions of the clients by the Deletion method on the three different datasets. The scores of the top five contributing clients in each of the three datasets are marked red, while the lowest scores are printed blue. We compare those with the results produced by FL-Cohort approach and presented in Tables 3 and 4, respectively. Table 3 shows the contribution of each cluster in FL-Cohort on different datasets. It can be seen that cluster 1 has achieved the highest scores for all three datasets Synthetic, FEMNIST, and CelebA, 0.3504, 0.380, and 0.3677, respectively. The worst-performing cluster producing the lowest scores for all datasets is cluster 4.

Table 3: The contribution of each cluster in FL-Cohort approach on Synthetic, FEMNIST, and CelebA datasets.

Cluster No.	size	Synthetic	FEMNIST	CelebA
1	6	0.3504	0.380	0.3677
2	7	0.2861	0.3432	0.344
3	5	0.2694	0.1873	0.2096
4	2	0.0941	0.0895	0.0787

Table 4 lists the members of each cluster produced by the FL-Cohort with their contribution values. Each client in a cluster makes the same contributions as the other clients that belong to the same cluster. It is interesting to compare the overlap between cluster 1, i.e., the top-contributing clients according to FL-Cohort, and the top-performing clients identified by the Deletion approach in the three datasets. One can notice that all the top five clients identified by the Deletion approach in Synthetic dataset belong to cluster 1 while for the other two datasets some of the clients are distributed in cluster 2 (6 and 14), and even two clients (12 and 17) are assigned to cluster 3. In addition, it is interesting to mention that clients 1 and 4, which have the lowest contribution according to FL-Cohort in the three datasets, also received the lowest evaluations from the Deletion approach.

Table 4: The contribution of each client in FL-Cohort approach on the datasets of Synthetic, FEMNIST, and CelebA in non-IID setting.

Cluster	clients	Synthetic	FEMNIST	CelebA
1	3, 7, 8, 10, 13, 20	0.0584	0.0633	0.0612
2	2, 5, 6, 9, 11, 14, 15	0.0408	0.0490	0.0491
3	12, 16, 17, 18, 19	0.0538	0.0374	0.0419
4	1, 4	0.0470	0.0448	0.0393

We further analyze and compare the performance of our two approaches, CA-

FL and GP-FL, to that of the two discussed above baseline approaches. The CA-FL is first applied to the three datasets, and the calculated evaluations are presented in Table 5. The clients' evaluations are calculated by running 5-fold cross-validation on each experimental dataset for ten communication rounds for non-IID label skew data (30%). The clients' scores (frequency of being selected as representatives) produced by CA-FL on the three datasets are given in three different columns of Table 5. The top achieved scores for the three datasets are again marked red, while the lowest scores are blue. We benchmark first the top evaluated clients to the top contributing clients identified by the Deletion approach and FL-Cohort, respectively. In the case of the former approach, three clients (3, 10, and 20) are common with the top-scored clients identified by our CA-FL on the Synthetic dataset, and two for each of the two other datasets, clients 7 and 15 for FEMNIST and clients 10 and 14 for CelebA, respectively. In the case of FL-Cohort, the overlapping between the two lists of the top evaluated clients is even higher, namely three joint clients for the Synthetic and CelebA datasets and two for FEMNIST. Moreover, our CA-FL approach has assigned the lowest scores to the same clients that have received the lowest evaluations from the Deletion approach in the three datasets. Client 2 is only slightly higher evaluated by CA-FL in the Synthetic dataset.

Table 5: The clients' evaluations produced by CA-FL on Synthetic, FEMNIST and CelebA datasets in non-IID setting.

client	Synthetic	FEMNIST	CelebA
1	1.6	1	1.4
2	2.2	1.8	**4.2**
3	**5.2**	**3.4**	2.6
4	1.6	1.4	1.4
5	**5.6**	2.2	1.6
6	2.6	2	1.8
7	3.6	**4.8**	**4.8**
8	4.2	2.4	2.4
9	2.2	2.2	1.8
10	**5.2**	2.6	**4.2**
11	2.6	**4.4**	2.4
12	1.6	2.2	1.6
13	2.6	**4.8**	**4.4**
14	1.6	1	**4.4**
15	2.2	2.2	1.4
16	1.6	1.6	2.2
17	1.6	2	1.6
18	**5.4**	1.6	1.8
19	1.2	**3.8**	1
20	**5.6**	2.6	3

We study the performance of GP-FL by comparing its evaluation results first to that produced by CA-FL and then to the two baseline approaches. Notice that, opposite to the other three approaches, the GP-FL evaluates the clients with negative connotations. Namely, it evaluates each client's behavior by considering how many times the client has changed its cluster during the training process. The calculated score is interpreted as a measure of client's instability with respect to its data. Table 6

Table 6: The clients' evaluations produced by GP-FL on Synthetic, FEMNIST and CelebA datasets in non-IID setting.

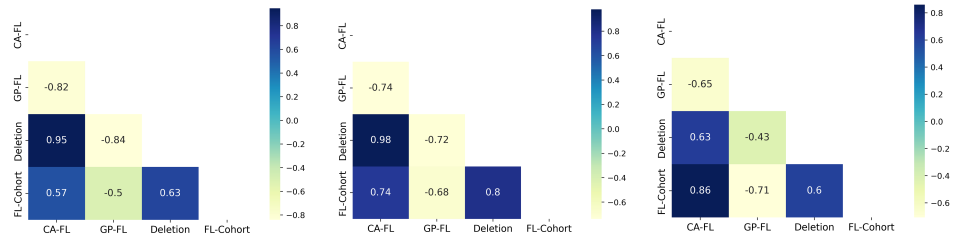
client	Synthetic	FEMNIST	CelebA
1	**4.4**	**4**	2.8
2	**3.2**	3.5	**3.2**
3	1.4	1.6	1.8
4	**3.6**	**4**	2
5	2.8	2.4	2.4
6	2.4	2.2	**4.2**
7	1	1	2
8	1.8	1.6	2.2
9	2.4	2	2.4
10	1.2	1.4	1.4
11	2.4	**3.8**	3
12	**3.6**	2.8	**3.6**
13	1.2	1.2	1.8
14	2.4	**3.6**	3
15	2.2	2	2.2
16	2	2.2	**3.8**
17	2	2.4	2.4
18	2.6	2.2	2.2
19	**4**	**3.8**	**4.2**
20	1	1.4	2

shows the clients' scores generated by GP-FL on the three experimental datasets evaluated by running 5-fold cross-validation for 10 communication rounds for non-IID label skew data (30%). The highest scores (worst performance) for the three datasets are marked blue, while the lowest scores (most stable behavior) are red. The lists of clients with red-marked scores by our two approaches, CA-FL and GP-FL, are first compared. In each dataset the two approaches have identified three joint clients. The number of overlapping clients between the blue lists of our two approaches is four for the Synthetic dataset and 3 and 2 for FEMNIST and CelebA, respectively. Very similar relations, as discussed above, are observed between the GP-FL and the Deletion approach.

6.2 Analysis of ranking correlations among the approaches

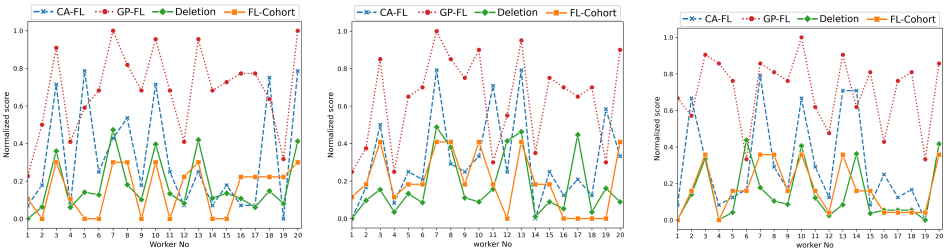
We can apply each of the four studied approaches to each of the three used experimental datasets to generate clients' rankings. These rankings can be used to analyze the correlations between any pair of approaches by applying Kendall's tau method introduced in Section 3.5. Figure 2 visualizes the heatmaps of Kendall's tau correlation ranking scores for the rankings produced by the four approaches, i.e., CA-FL, GP-FL, Deletion approach, and FL-Cohort, on the three different datasets. The darker the color, the stronger the correlation of the two methods, and vice versa. These correlations allow us to identify the similarity between the rankings generated by our two FL algorithms and the two baselines.

In case of the Synthetic dataset, one can notice that the correlation between CA-FL and the Deletion approach is very strong and positive, namely 0.95. This means



(a) Synthetic (b) FEMNIST (c) CelebA

Figure 2: Heatmaps of Kendall's tau ranking scores of CA-FL, GP-FL, Deletion approach, and FL-Cohort on Synthetic, FEMNIST and CelebA datasets.

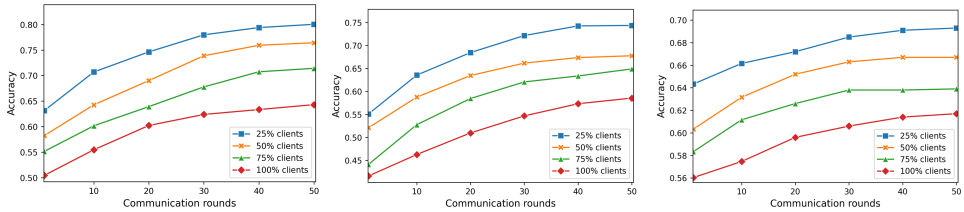


(a) Synthetic (b) FEMNIST (c) CelebA

Figure 3: Comparison of the ranking signatures of the four approaches produced on Synthetic, FEMNIST and CelebA datasets.

that our approach (CA-FL), which requires much less computational resources, can evaluate the clients' contribution similarly to the Deletion approach. The GP-FL produces scores that are interpreted as ones presenting the degree to which clients' behavior is unreliable. This is also the reason for lower correlation scores between the GP-FL and the two baseline approaches (-0.84 and -0.5, respectively), since the latter approaches evaluate how the global model performance will be affected if data of a client or a group of clients is not used in the training. We can generally observe that the FL-Cohort has lower rank correlations with our two approaches than the Deletion approach. Moreover, the correlation between our two approaches is demonstrated to be strong, namely -0.82.

In FEMNIST dataset, as one can see in Figure 2(b), similar to the Synthetic dataset, CA-FL has a stronger correlation to the Deletion approach than to the FL-Cohort. In contrast, in CelebA dataset, CA-FL has a higher correlation score with the FL-Cohort than the Deletion approach, see Figure 2(c). In general, in Synthetic and FEMNIST datasets, CA-FL correlations with the Deletion approach are above 0.9, while with FL-Cohort are between 0.5 and 0.7. To understand the correlations among CA-FL, GP-FL, Deletion approach, and FL-Cohort, we plot together and compare their ranking signatures produced on the three used datasets. These are presented in Figure 3. We have initially converted the ranking scores of our two approaches into the interval $[0, 1]$ using min-max normalization. In addition, the normalized scores



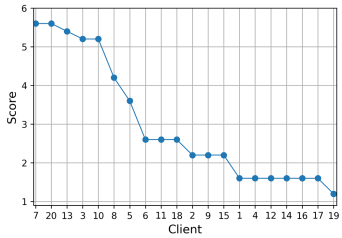
(a) Synthetic **(b)** FEMNIST **(c)** CelebA

Figure 4: Performance comparison for CA-FL when different percentages of clients are selected to participate in the training of the shared global model on Synthetic, FEMNIST and CelebA datasets.

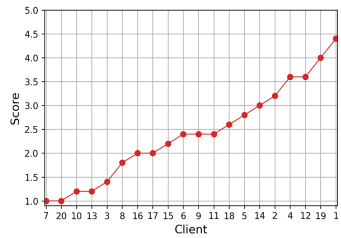
of GP-FL algorithm are further transformed by finding their complement to 1. Analyzing the plots in Figure 3, one can notice that the lower correlation scores between GP-FL and CA-FL and Deletion approach, respectively, are mostly due to the disagreement on the evaluation of a few clients, namely 15, 16, and 17 for all the three datasets. However, a bigger group of clients (2, 7, 13, 14, 16, 17, and 18) is responsible for the lower correlation score between CA-FL and the Deletion approach in the case of CelebA dataset. In addition, FL-Cohort has demonstrated a different evaluation pattern in comparison to the other three approaches on workers 16, 17, 18, and 19 in all three datasets. Interestingly, clients 16, 17, and 18 are repeated in the commented cases. The first two got low evaluations by our two approaches in all three datasets, while client 18 got a high score by CA-FL and a low one by GP-FL in Synthetic dataset. The same is noticed, e.g., for clients 11 and 19 in FEMNIST dataset. Evidently, the stable performance of some clients recognized by CA-FL is not supported by GP-FL for those clients, i.e., a client can be reliable w.r.t. its local model performance but also demonstrates unstable behavior w.r.t. the class distribution of its data.

6.3 CA-FL

In this section, we report the results of the additional experiments done to study further how the evaluations of the clients based on the CA-FL algorithm can be used to improve the global model performance. We first analyze how the performance of CA-FL will be affected when different percentages of the clients are selected to participate in federated learning. The clients are initially ranked in descending order based on their scores produced by applying the CA-FL algorithm. Then, we can select different percentages of the clients and train a global model only on those clients. Figure 4 depicts the performance comparison for CA-FL under different percentages of selected top-scored clients on Synthetic, FEMNIST and CelebA datasets. The performance of CA-FL algorithm has been evaluated by running 3-fold cross-validation on each experimental dataset for 50 communication rounds for Non-IID label skew data (30%). In general, for all three datasets we notice that the algorithm



(a) Three groups of clients can easily be recognized in CA-FL in descent order: high evaluated (most reliable behavior) clients (scores above 5), middle valued clients (scores above 2 and below 5) and finally, low evaluated (least reliable behavior) clients (scores below 2).

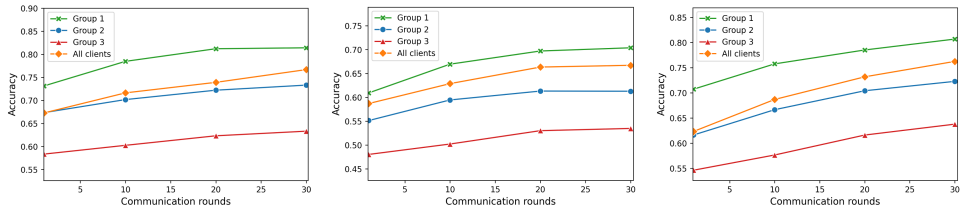


(b) Three groups of clients can be identified in GP-FL in ascent order: low scored (most reliable behavior) clients (scores below 2), middle valued clients (scores above 2 and below 3.5) and finally, high scored (least reliable behavior) clients (scores above 3.5).

Figure 5: A plot of the clients' scores calculated by CA-FL and GP-FL on the Synthetic dataset.

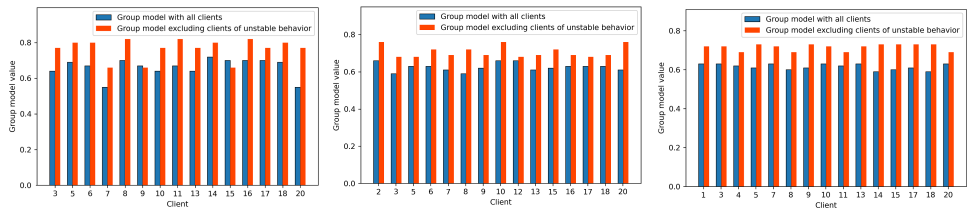
performance is declining proportionally to the increase of the percentage of the selected clients. It can be observed that CA-FL has performed better when the clients with the most reliable behavior (25%) are participating since it is faster to converge. In other words, as the number of clients with unreliable behavior increases, CA-FL performance degrades more slowly.

In addition, we conduct an experiment in which the clients are split in three groups applying binning on their CA-FL evaluations. Figure 5(a) visualizes the results of the clients' ranking on the Synthetic dataset. As one can notice the clients can be split in three groups as follows: the clients with most reliable behavior (scores above 5), denoted as **Group 1**, the clients demonstrating more modest behavior w.r.t. the reliability (scores above 2 and below 5), **Group 2**, and finally, the clients with least reliable behavior (scores below 2), **Group 3**. For each group of clients a separate federated learning model is trained. The performance of the three built models is benchmarked to that of the global model trained on the all clients' data. This experiment is conducted on each dataset by running a 3-fold cross-validation for 30 global rounds under Non-IID label skew data (30%). Note that the three groups have different size and as well as contain different clients in the three datasets. Figure 6 depicts the global model accuracy in different training rounds. The results demonstrate the robustness of CA-FL, since, in the three dataset scenarios, the global model built on clients' data with the most reliable behavior has achieved higher accuracy than the models built on the other two groups of clients. In addition, only this model outperforms the overall global model trained on all clients' data. Evidently, such splitting of clients in groups w.r.t. their behavior and building group global models will allow to ensure fairness in FL scenarios by providing each client a model properly reflecting its respective contribution to the federation.



(a) Synthetic (b) FEMNIST (c) CelebA

Figure 6: Performance comparison of the three CA-FL global models on Synthetic, FEMNIST and CelebA datasets. The global models are built on three groups of clients split w.r.t. their behavior evaluated during the training process. **Group 1** contains the clients with most reliable behavior, **Group 2** has the clients demonstrating modest behavior w.r.t. the reliability, and **Group 3** are the clients with least reliable behavior.

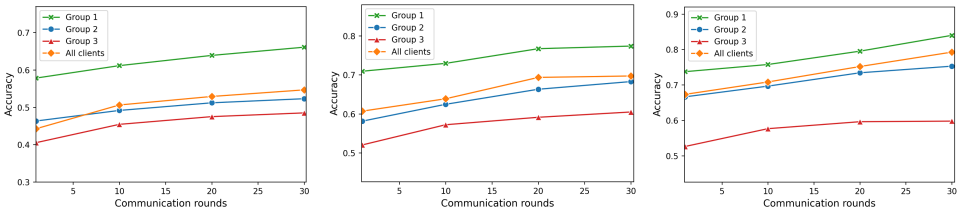


(a) Synthetic (b) FEMNIST (c) CelebA

Figure 7: Comparison of the performance of GP-FL models built respectively on all the clients data and when the five clients with most unreliable behavior are excluded on the Synthetic, FEMNIST and CelebA datasets.

6.4 GP-FL

In this section, we first analyze how the performance of GP-FL will be affected when the five clients with the most unreliable behavior are excluded from participating in federated learning. This scenario is studied on all three used datasets. The performance of the built GP-FL models is benchmarked to the models built on all the clients' data. See Figure 7. One can notice that the model trained on the set of clients that do not contain ones with non-stable behavior outperforms the overall global model using the data from all the clients in the three dataset scenarios. Similarly to the experiment conducted for CA-FL, the clients are split in three groups using binning on their GP-FL evaluations. Figure 5(b) plots the produced ranking signature on the Synthetic dataset. We have obtained a small group of clients with high scores respectively representing clients with unstable behavior (scores above 3.5), denoted as **Group 3**, and two almost equal size groups of clients demonstrating not so unstable behavior (scores above 2 and below 3.5), **Group 2**, and ones with stable behavior (scores below or equal to 2), **Group 1**, respectively. In addition, in Figure 8, we compare the performance of the three GP-FL global models in terms of convergence speed on Synthetic, FEMNIST, and CelebA datasets. It is worth noting that with different datasets, again Group 1 always achieves higher accuracy than other groups' models and also outperforms the overall global model trained on the data of all the clients. This may be because the clients in Group 1 have stable behavior, thereby, the results have a prominent gradation. Therefore, only the clients with stable behavior



(a) Synthetic (b) FEMNIST (c) CelebA

Figure 8: Performance comparison of the three GP-FL global models on Synthetic, FEMNIST and CelebA datasets. The global models are built on three groups of clients split w.r.t. their behavior evaluated during the training process. **Group 1** contains the clients with stable behavior, **Group 2** has the clients demonstrating not so stable behavior, and **Group 3** are the clients with unstable behavior.

can be selected for aggregating the global model.

7 Conclusion

In this paper, we have proposed new federated learning solutions for quantifying the contribution of each client to the overall global model according to its behavior, using our FL algorithms, CA-FL and GP-FL, respectively. We have studied the robustness of our approaches to evaluate the client's behavior during the training process on three LEAF datasets. The obtained experimental results have demonstrated the capability of our approaches in realistically assessing the client's contribution to the overall FL model. In summary, without incurring any considerable communication and computation costs, our methods have shown robustness in evaluating each client's behavior respectively contribution to the federated model and have the potential to be used for this purpose. The two FL models can also be extended to more complex cases with a robust unlearning procedure in future work. In addition, we will consider developing a hybrid approach that combines CA-FL and GP-FL, which could lead to an advanced FL model that selects the most reliable clients while ignoring the clients with unstable behavior to train a global model.

References

[1] H. B. M. et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *International Conference on Artificial Intelligence and Statistics*. 2016.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. "Federated Learning: Challenges, Methods, and Future Directions". In: *IEEE Signal Processing Magazine* 37 (2019), pp. 50–60.

[3] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li. “The Right to be Forgotten in Federated Learning: An Efficient Realization with Rapid Retraining”. In: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications* (2022), pp. 1749–1758.

[4] 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)* (2018), pp. 1–7.

[5] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. X. Song. “A Principled Approach to Data Valuation for Federated Learning”. In: *Federated Learning*. 2020.

[6] W. Zhang, T. Zhou, Q. Lu, X. Wang, C. Zhu, H. Sun, Z. Wang, S. K. Lo, and F.-y. Wang. “Dynamic-Fusion-Based Federated Learning for COVID-19 Detection”. In: *IEEE Internet of Things Journal* 8 (2020), pp. 15884–15891. URL: `https://api.semanticscholar.org/CorpusID:221836207`.

[7] Z. You, X.-z. Wu, K. Chen, X. Liu, and C. Wu. “Evaluate the Contribution of Multiple Participants in Federated Learning”. In: *International Conference on Database and Expert Systems Applications*. 2021.

[8] V. Siomos. “Contribution Evaluation in Federated Learning: Examining Current Approaches”. In: 2021.

[9] J. Zhang, C. Li, A. Robles-Kelly, and M. Kankanhalli. “Hierarchically Fair Federated Learning”. In: *ArXiv* abs/2004.10386 (2020).

[10] G. Wang, C. X. Dang, and Z. Zhou. “Measure Contribution of Participants in Federated Learning”. In: *2019 IEEE International Conference on Big Data (Big Data)* (2019), pp. 2597–2604.

[11] C. Düsing and P. Cimiano. “Towards predicting client benefit and contribution in federated learning from data imbalance”. In: *Proceedings of the 3rd International Workshop on Distributed Machine Learning* (2022).

[12] A. A. M. Al-Saedi, V. Boeva, and E. Casalicchio. “Reducing Communication Overhead of Federated Learning through Clustering Analysis”. In: *2021 IEEE Symposium on Computers and Communications (ISCC)* (2021), pp. 1–7.

[13] A. A. Al-Saedi and V. Boeva. “Group-Personalized Federated Learning for Human Activity Recognition Through Cluster Eccentricity Analysis”. In: *International Conference on Engineering Applications of Neural Networks*. 2023.

[14] T. Li, M. Sanjabi, and V. Smith. “Fair Resource Allocation in Federated Learning”. In: *ArXiv* abs/1905.10497 (2019).

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[16] Z. Liu, P. Luo, X. Wang, and X. Tang. “Deep Learning Face Attributes in the Wild”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3730–3738. DOI: 10.1109/ICCV.2015.425.

[17] F. S. et al. “Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31 (2019), pp. 3400–3413.

[18] C. Briggs, Z. Fan, and P. András. “Federated learning with hierarchical clustering of local updates to improve training on non-IID data”. In: *2020 International Joint Conference on Neural Networks (IJCNN)* (2020), pp. 1–9.

[19] Z. Li, Y. He, H. Yu, J. Kang, X. Li, Z. Xu, and D. T. Niyato. “Data Heterogeneity-Robust Federated Learning via Group Client Selection in Industrial IoT”. In: *IEEE Internet of Things Journal* 9 (2022), pp. 17844–17857.

[20] Y. Xie, Z. Wang, D. Gao, D. Chen, L. Yao, W. Kuang, Y. Li, B. Ding, and J. Zhou. “FederatedScope: A Flexible Federated Learning Platform for Heterogeneity”. In: *Proc. VLDB Endow.* 16 (2022), pp. 1059–1072.

[21] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi. “Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning”. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications* (2018), pp. 2512–2520.

[22] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. S. Doermann, and A. Innanje. “Ensemble Attention Distillation for Privacy-Preserving Federated Learning”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), pp. 15056–15066.

[23] J. Huang, C. Xu, Z. Ji, S. Xiao, T. Liu, N. Ma, and Q. Zhou. “AFLPC: An Asynchronous Federated Learning Privacy-Preserving Computing Model Applied to 5G-V2X”. In: *Security and Communication Networks* (2022).

[24] L. Wang, W. Wang, and B. Li. “CMFL: Mitigating Communication Overhead for Federated Learning”. In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[25] M. Asad, A. Moustafa, T. Ito, and M. Aslam. “Evaluating the Communication Efficiency in Federated Learning Algorithms”. In: *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (2020), pp. 552–557.

[26] N. Bouacida, J. Hou, H. Zang, and X. Liu. “Adaptive Federated Dropout: Improving Communication Efficiency and Generalization for Federated Learning”. In: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2020), pp. 1–6.

[27] B. Huang, X. Li, Z. Song, and X. Yang. “FL-NTK: A Neural Tangent Kernel-based Framework for Federated Learning Analysis”. In: *International Conference on Machine Learning*. 2021.

[28] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, and X. Wang. “Resource-Efficient and Convergence-Preserving Online Participant Selection in Federated Learning”. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)* (2020), pp. 606–616.

[29] Y. J. Cho, J. Wang, and G. Joshi. “Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies”. In: *ArXiv* abs/2010.01243 (2020).

[30] L. Liu, J. Zhang, S. Song, and K. B. Letaief. “Hierarchical Quantized Federated Learning: Convergence Analysis and System Design”. In: (2021).

[31] S. e. a. Caldas. “LEAF: A Benchmark for Federated Settings”. In: (2018).

[32] F. Sattler, K.-R. Müller, and W. Samek. “Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2019), pp. 3710–3722.

[33] M. Duan, D. Liu, X. Ji, R. Liu, L. Liang, X. Chen, and Y. Tan. “FedGroup: Ternary Cosine Similarity-based Clustered Federated Learning Framework toward High Accuracy in Heterogeneous Data”. In: *ArXiv* abs/2010.06870 (2020).

[34] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran. “An Efficient Framework for Clustered Federated Learning”. In: *IEEE Transactions on Information Theory* 68 (2020), pp. 8076–8091.

[35] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, and J. Jiang. “Multi-Center Federated Learning”. In: *ArXiv* abs/2108.08647 (2020).

[36] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal. “Federated Multi-Task Learning under a Mixture of Distributions”. In: *Neural Information Processing Systems*. 2021.

[37] Y. Ruan and C. Joe-Wong. “FedSoft: Soft Clustered Federated Learning with Proximal Local Updating”. In: *AAAI Conference on Artificial Intelligence*. 2021.

[38] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh. “Three Approaches for Personalization with Applications to Federated Learning”. In: *ArXiv* abs/2002.10619 (2020).

[39] S. K. Shyn, D. Kim, and K. Kim. “FedCCEA : A Practical Approach of Client Contribution Evaluation for Federated Learning”. In: *ArXiv* abs/2106.02310 (2021).

[40] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. T. Niyato, and Q. Yang. “A Sustainable Incentive Scheme for Federated Learning”. In: *IEEE Intelligent Systems* 35 (2020), pp. 58–69.

[41] B. Zhao, X. Liu, and W.-n. Chen. “When Crowdsensing Meets Federated Learning: Privacy-Preserving Mobile Crowdsensing System”. In: (2021).

[42] R. Zeng, S. Zhang, J. Wang, and X. Chu. “FMore: An Incentive Scheme of Multi-dimensional Auction for Federated Learning in MEC”. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)* (2020), pp. 278–288.

[43] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng. “Towards Fair and Privacy-Preserving Federated Deep Models”. In: *IEEE Transactions on Parallel and Distributed Systems* 31 (2019), pp. 2524–2541.

[44] T. Song, Y. Tong, and S. Wei. “Profit Allocation for Federated Learning”. In: *2019 IEEE International Conference on Big Data (Big Data)* (2019), pp. 2577–2586.

[45] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. X. Song, and C. J. Spanos. “Towards Efficient Data Valuation Based on the Shapley Value”. In: *International Conference on Artificial Intelligence and Statistics*. 2019.

[46] J. B. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *In Lucien M. Le Cam and Jerzy Neyman, editors, Proceedings of the Berkley symposium on mathematical statistics and probability* 1 (1967), pp. 281–297.

[47] X. Jin and J. Han. “K-Medoids Clustering”. In: *Encyclopedia of Machine Learning*. 2010.

[48] M. J. van der Laan, K. S. Pollard, and J. Bryan. “A new partitioning around medoids algorithm”. In: *Journal of Statistical Computation and Simulation* 73 (2003), pp. 575–584. URL: `https://api.semanticscholar.org/CorpusID:17437463`.

[49] S. van Dongen. “Graph clustering by flow simulation”. In: 2000.

[50] J. Vlasblom and S. J. Wodak. “Markov clustering versus affinity propagation for the partitioning of protein interaction graphs”. In: *BMC Bioinformatics* 10 (2009), pp. 99–99.

[51] P. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[52] C. G. Bezerra et al. “An evolving approach to data streams clustering based on typicality and eccentricity data analytics”. In: *Information Sciences* 518 (2020), pp. 13–28. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.12.022. URL: https://www.sciencedirect.com/science/article/pii/S0020025519311363.

[53] P. Angelov. “Anomaly detection based on eccentricity analysis”. In: *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*. 2014, pp. 1–8. DOI: 10.1109/EALS.2014.7009497.

[54] J. G. Saw et al. “Chebyshev Inequality With Estimated Mean and Variance”. In: *The American Statistician* 38 (1984), pp. 130–132.

[55] I. Škrjanc et al. “Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey”. In: *Inf. Sci.* 490 (2019), pp. 344–368.

[56] M. G. Kendall. “Rank Correlation Methods”. In: 1949.

[57] R. N. Forthofer and R. G. Lehnen. “Rank Correlation Methods”. In: *Public Program Analysis: A New Categorical Data Approach*. Boston, MA: Springer US, 1981, pp. 146–163. ISBN: 978-1-4684-6683-6. DOI: 10.1007/978-1-4684-6683-6_9.

[58] Y. Liu, S. Qin, Y. Sun, and G. Feng. “Resource Consumption for Supporting Federated Learning in Wireless Networks”. In: *IEEE Transactions on Wireless Communications* 21 (2022), pp. 9974–9989.

[59] G. Drainakis, P. Pantazopoulos, K. V. Katsaros, V. Sourlas, A. J. Amditis, and D. I. Kaklamani. “From centralized to Federated Learning: Exploring performance and end-to-end resource consumption”. In: *Comput. Networks* 225 (2023), p. 109657.

[60] J. Huang, R. Talbi, Z. Zhao, S. Bouchenak, L. Y. Chen, and S. Roos. “An Exploratory Analysis on Users’ Contributions in Federated Learning”. In: *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* (2020), pp. 20–29.

[61] G. Navas-Palencia. “Optimal binning: mathematical programming formulation”. In: *ArXiv* abs/2001.08025 (2020).

[62] P. A. Flach. “Machine Learning - The Art and Science of Algorithms that Make Sense of Data”. In: 2012.

[63] S. Kolouri et al. “Optimal Mass Transport: Signal processing and machine-learning applications”. In: *IEEE Signal Processing Magazine* 34 (2017), pp. 43–59.

[64] V. N. L. Duy and I. Takeuchi. “Exact statistical inference for the Wasserstein distance by selective inference”. In: *Annals of the Inst. of Stat. Mathematics* 75 (2021), pp. 127–157.