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 J1630, Campus Gräsvik

Supervisors
Prof. Veselka Boeva, Blekinge Institute of Technology
Assoc. Prof. Emiliano Casalicchio, Blekinge Institute of Technology

Faculty Opponent
Prof. György Dán, KTH Royal Institute of Technology
Grading Committee
Prof. Volker Markl, Technical University of Berlin, Germany
Prof. Paul Davidsson, Mälmo University
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. Finally, resource-aware techniques to evaluate a client's contribution by assessing its behavior during training are proposed.

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-?????

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

Dedication

This is a dedication.

“If we knew what it was we were doing, it would not be called research, would it?”

Albert Einstein

Acknowledgements

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.

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

Author's contribution to the papers

The author is the main driver and the first author of all the papers. For these studies, he was involved in all the phases of the research, that is idea generation, designing and conducting experimentation, analysis of results, writing the original draft, reviewing and editing the manuscript.

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.
MCL	Markov Clustering.
ML	Machine Learning.
NN	Neural Network.
Non-IID	Non-Independently and Identically Distributed.
PFL	Personalized Federated Learning.
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, developments in Artificial Intelligence (AI)-based approaches such as 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 vast amounts of training data [1]. The widely used approach involves gathering of data in a central location, processing it in a centralized manner, and then creating a unified model based on the processed data. However, in practical terms, sharing large amounts of IoT data in a central location is expensive 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 on these devices has gradually shifted from the central server to edge devices.

In light of this context, Google presented the concept of Federated Learning (FL) [5] as an emerging ML paradigm for decentralized data to address such issues, allowing 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, instead of sharing raw data, model updates are exchanged. 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] introduced the Federated Averaging (FedAvg) algorithm to implement this concept, originally intended for a cross-device scenario often observed in mobile phones. In this context, there exists a large pool of devices that possess limited computational capabilities. These devices are frequently characterized by their unreliability, both in terms of availability due to communication limitations and costs, as well as reliability. The ML model was effectively used to improve the predictive capabilities of Google Board (GBoard), which offers word suggestions to users while typing [7].

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 [8]. However, despite its potential, this approach still confronts considerable challenges in the domain. These include the elevated communication cost required for data 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 our papers; in the following section, we will discuss these challenges in more detail.

1.1 Federated Learning Challenges

FL continues to gain attention, researchers are actively working to address the associated challenges and improve system performance. These challenges, owing to the various nature of the federated setting from the traditional problems, introduce an interesting research paradigm to the research community. 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 explored further to improve the performance of the 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) [9], 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 ultra-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 [10, 11]. 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 [12, 13]. To reap the advantages of FL, these challenges must be addressed to allow for the wider adoption of FL systems [14]. These challenges have been discussed in **PAPERs I, II, IV, and VI**. Different FL models are suggested to address the issue 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 substantial. For instance, VGG-16, a NN used for image recognition, contains 138 million parameters [15] and requires 526 megabytes of storage when encoded with 32 bits. In addition, the number of devices involved in the FL systems could range from hundreds of thousands to millions, Figure 1.1 illustrates the run-time cost of FedAvg, with the blue and green 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. The run-time cost of each global averaging step in FedAvg 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 communication delay is influenced by the shared bandwidth among all client devices. Therefore, although numerous client devices can help accelerate model training, both the computation time and transmission time will increase significantly when W_t is large. Likewise, a decreased communication period T helps in the convergence of the model, but the cost of communication delay will increase compared to the computation time. Hence, the optimization of the FedAvg algorithm involves a complex trade-off between 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.

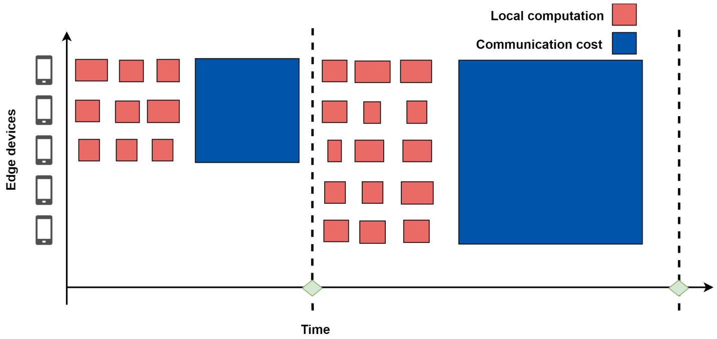


Figure 1.1: An example of the runtime expense of FedAvg algorithm, where the red and blue blocks indicate the time taken for local computation and transmission time, respectively.

In recent years, several techniques have been used 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 will be introduced later in Section 3.1.1. However, these methods often face heavy performance 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 [16].

Another method commonly used in communication-efficient FL is to minimize the number of model updates that are transferred to the server [12]. Since the exchanging of large model updates requires significant communication resources, it is vital to reduce the volume of data that has to be collected by the server [12, 13, 17]. Our proposed FL models also fall into this category, which has been explored in **PAPERS I** and **IV** focused on decreasing the number 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 class distributions. These distributions, known as Non-Independently and Identically Distributed (Non-IID), reflect real-world applications [18, 19]. Non-IID data represents one of the key challenges in FL [20, 21]. Non-IID data implies that the datasets may vary in size and distribution, which makes it hard to fit all local datasets with one global model. Moreover, the presence of Non-IID data may lead to client drift [22]. Such a phenomenon can considerably undermine the performance of FL [23, 24]. 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 towards the average of client optima. In the case of IID data, the average of client updates is close to the global optimum $\mathcal{M}$, as it is equidistant from both the local optima (e.g, two clients) $\mathcal{M}^1$ and $\mathcal{M}^2$. So, 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). Recent works [23–26] have shown that such data heterogeneity in FL approaches can significantly degrade the performance of the global model, which could eliminate the main motivation to participate in FL training. While tradtional FL approaches pursue a global optima of all client devices, the concept of Personalized Federated Learning (PFL) seeks to learn personalized models for each task or specific device [27,

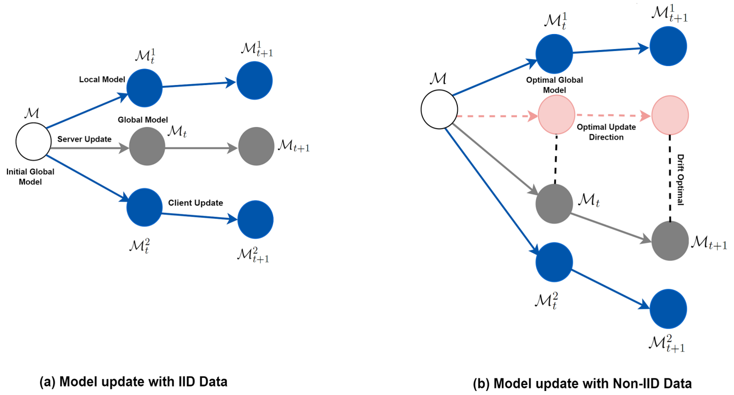


Figure 1.2: Visualization of client drift in FedAvg algorithm for two clients with two local epochs on different structured data. (a) IID data setting. (b) Non-IID data setting.

28] that fit the diverse local dataset. 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 epoch, without taking into account the available resources of each client. As a result, the performance of the federated model is negatively affected due to the slow convergence rate, decreased training efficiency, and the failure to fully utilize local updates from heterogeneous clients [29]. Meanwhile, the computing capacity and communication resources of the participant devices in FL vary, and may be reluctant to take part in FL training. More importantly, the performance of the federated model are significantly influenced by the client selection scheme involved [30, 31]. Therefore, it is essential to have an effective mechanism to select participants in the FL systems. This challenge has been discussed in **PAPER I**, **PAPER II**, and **PAPER IV**.

- **Personalization**: The vanilla FL approach creates a single global shared model by taking the average of all local models of client devices [32]. It assumes that all client devices have the same learning task. Furthermore, certain client devices may show poor performance, while others perform well, inevitably eliminating client personalization and decreasing the ability to represent client characteristics [33]. However, in real-world applications, different devices involved may encounter distinct ML problems 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 that is necessary 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 [34]. Therefore, while there are benefits to using PFL for personal learning, it did not fully take advantage of the potential for collaborative learning. This limitation is problematic for two main reasons: First, most of the client devices have a limited amount of data. Furthermore, despite differences among client devices or tasks, it is reasonable to assume that there is some similarity among devices. The challenge of 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 research indicates 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, the confidentiality, integrity, and availability of the data used [35, 36]. 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 [37]. 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 personalized and resource-aware FL models by using clustering analysis. These new solutions aim 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 design and evaluate FL algorithms that are based on clustering anaylsis with an emphasis on reducing resource consumption while maintaining an acceptable level of performance.*

2. *To develop groups of personalized FL models to address data heterogeneity.*

3. *To investigate and analyze the latest advances in intelligent applications in context awareness and gain insight into motivations, areas, challenges, and research gaps, focusing on Sensor Networks (SNs) and AI techniques.*

1.3 Research Questions

This thesis integrates three distinct directions: resource-conscious FL solutions, PFL solutions, and the exploration of edge-based AI for SNs. In this context, six different studies have been carried out as a component of this PhD thesis. Given the scope and objectives of this thesis. As a result, the answers to the following questions are sought:

RQ 1: *How we can develop FL models that reduce resource consumption without sacrificing the model performance?*
Motivation: In practical federated systems, the participating devices often have limitations in terms of 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 diverse fleet of client devices that differ in data quality, computational capacity, and battery lifetime levels. Several studies [38–40] have been conducted to minimize the resources of client devices in FL. Hence, it is prudent to develop effective methods in a FL system that considers the constraints of the device resource.
Papers: This research question is explored in PAPER I through the use of clustering analysis. In this method, the local updates made by representatives of a cluster are identified, and only these updates are uploaded to the central server. This helps to reduce the expenses associated with network communication in Human Activity Recognition (HAR) datasets. PAPER II tackles the question by incorporating clustering analysis into the Wireless Networks (WNs) environment as well. In this study, various factors such as energy consumption, bandwidth usage, and accuracy are taken into account when choosing a representative from a cluster to communicate with a central server. This study examined how these factors affect FL performance. In PAPER IV, we extended the work presented in PAPER I by improving the optimization of model aggregation and minimizing communication overhead. This was 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. However, FL still faces the challenge of data quality. Because federated learning fuses models trained by data from various client devices. The data quality from some clients could be low. Low-quality data has two negative impacts. First, they disrupt the training process and cause excessive computational costs [41, 42]. Second, they have the potential to adversely affect the models of other devices involved that possess high-quality

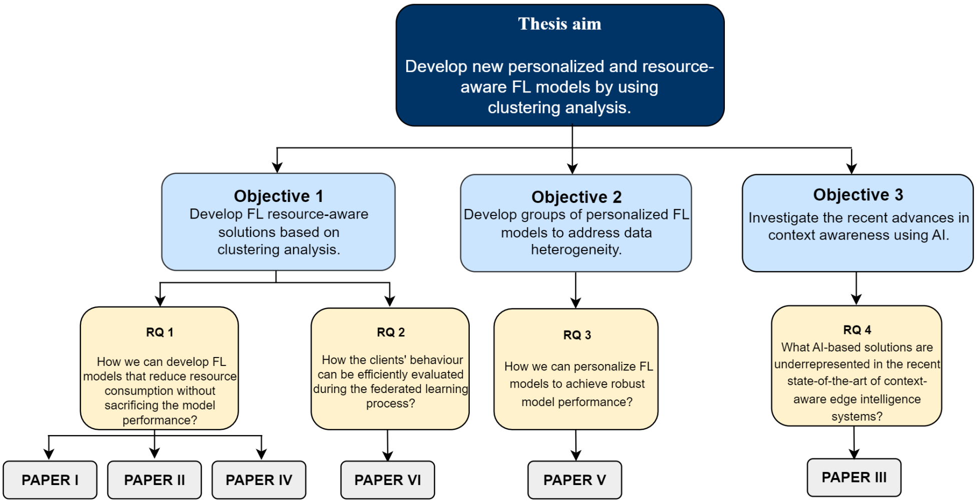
data [43, 44]. These concerns might cause certain federated learning participants to withdraw from federated learning training. To address the issue of variable data quality, federated learning requires a robust method for evaluating data quality.
Papers: The majority of current solutions require significant resources and are typically executed as an extra evaluation step. This results in a high computational burden for data owners with large datasets. PAPER VI investigates this research question by using the FL models proposed in PAPERS I and V. Primarily, in PAPER I, the evaluation of how often each client is selected as a representative group is carried out. This assessment can be interpreted as an indicator of the reliability of the client's data. Additionally, in PAPER V, the calculation of how often a client changes the clusters during FL training provides information on the client's instability, suggesting a lack of reliability. The FL models were used to assess the contribution of the client during FL training, without requiring additional resources or time.

RQ 3: *How we can personalize FL models to achieve robust model performance?*
Motivation: In terms of the present research advancements FL, data heterogeneity represents a significant challenge worthy of attention. In FedAvg (the first FL method), all participants jointly train a joint model [35, 45]. However, a single global model may struggle to fit all participants, thereby restricting its generalizability. Furthermore, given the wide range of application scenarios, each client device may need to build distinct local models that are customized to the specific characteristics and data, which cannot be satisfied in the existing FL setting. Although PFL 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 terms of tasks or data. Therefore, its existing limitation requires the development of a new personalized FL model.
Papers: The aim is achieved in PAPER V, which presents a clustering-based approach for group-personalized FL in the context of HAR applications. The FL model presented in this study aims to address the issue of heterogeneity of data and achieve a balance between the global model and local models.

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 moves 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 RQ aims to identify AI-based solutions that are not adequately represented in context-aware systems, particularly within SNs. In addition, it aims to determine any research gaps in current state-of-the-art AI-based solutions.

Papers: This goal is achieved in PAPER III, where a comprehensive investigation was carried out on the application of AI, ML, and DL methods in the latest developments in context awareness in WNs. In this study, an investigation of the existing literature was conducted to provide an overview of various domains, highlight the main obstacles within each field, outline the reasons behind the research, and identify any gaps in current studies.

The visualization of the included studies and their connection wih the aim, objectives, and research questions of the thesis is provided in Figure 1.3.


Thesis aim
Develop new personalized and resource-aware FL models by using clustering analysis.

Objective 1
Develop FL resource-aware solutions based on clustering analysis.

Objective 2
Develop groups of personalized FL models to address data heterogeneity.

Objective 3
Investigate the recent advances in context awareness using AI.

RQ 1
How we can develop FL models that reduce resource consumption without sacrificing the model performance?

RQ 2
How the clients' behaviour can be efficiently evaluated during the federated learning process?

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

RQ 4
What AI-based solutions are underrepresented in the recent state-of-the-art of context-aware edge intelligence systems?

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

1.4 Thesis Outline

The remaining sections of this thesis are organized as follows:

Chapter 2 - Background: Here, 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 - Methodology: The methodology employed is presented in this chapter. Additionally, we provide details of the datasets used and our implementation approach.

Chapter 5 - Results: This chapter discusses the results of the thesis. Each research question is addressed with the corresponding papers that are part of this thesis.

Chapter 6 - Summary of contributions: Here, we present a summary of the thesis and explore future research directions.

2 Background

2.1 Centralized Machine Learning

With the advancement in IoT devices, rapid connectivity, AI, and ML have led to the generation of large amounts of data, commonly known as big data [46]. ML techniques are employed to manage large sets of data. However, in traditional ML settings, most of these ML techniques are centralized techniques, meaning that data from multiple devices is aggregated into a central server. This central server is used to train a joint model that can then be shipped and applied to all devices that will be used for inference [47]. Figure 2.1 clearly shows that data must be collected from different edge devices to train a joint model. Despite its impressive success, several issues must be highlighted. In fact, the violation of data privacy is high when sensitive data is transferred to a central server. Furthermore, the upload of 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 [48].

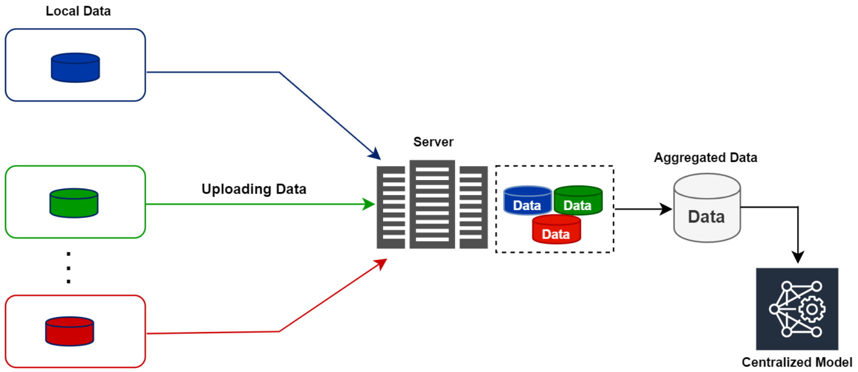


Figure 2.1: General framework for a centralized training approach.

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 scenario of distributed learning, the training of a model 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 [49].

The concept of FL was first proposed by Google in 2017 [5, 50] as a type of distributed machine learning approach that allows training a joint model by cooperating with edge devices without revealing training data under centralized server supervision. The main idea of FL is to cooperatively train ML models among numerous independent edge devices (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 edge devices on their local datasets. Updated models are then sent to the central server, which performs the aggregation of these trained local models to produce a unified model, in contrast to traditional centralized machine learning methods [51]. The updated model is then returned to the edge devices 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 the exchange of raw data between edge devices. In FL, multiple number of edge devices 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 just the updated model to the server, where they are aggregated to produce the global model. By utilizing the resources of edge devices, FL introduces a transition from expensive centralized ML training to a distributed approach [52].

2.2.1 Standard Federated Learning

The FL system consists of a central server and a group of client devices W_t, (i.e. $W_t \subset W$). Each client device 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 the most commonly used aggregation mechanism in the literature, namely the FedAvg algorithm [5]. In addition, it will serve as a reference for the conducted experimental analysis. Generally, FedAvg can be summarized as follows.

1. **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

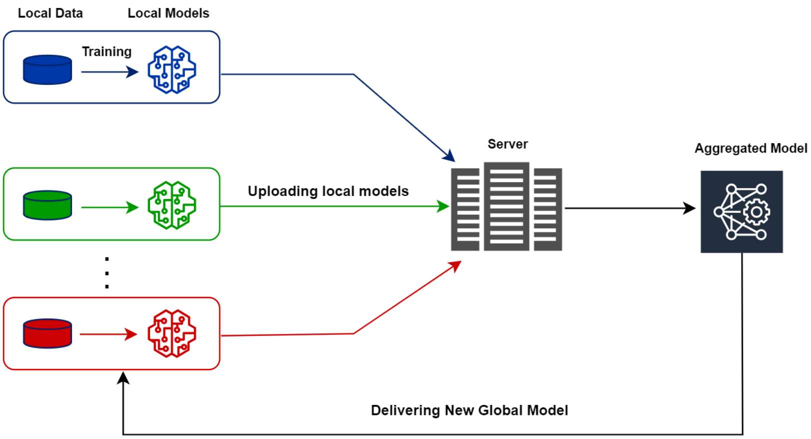


Figure 2.2: General working process of standard FL.

available data.

Step 2: The central server distributes the global model $\mathcal{M}_0$ to all participating devices $w_i \in W$.

2. **Local training**:

   Step 3: Each edge device $w_i \in W_t$ performs mini-batch Stochastic Gradient Descent (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:

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

3. **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 Eq. 2.3

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

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 pseudocodes for the FedAvg algorithm, which involves the participation of a set of client devices denoted as $W_t \subseteq W$, as stated in [53]. The structure of the model of both the global model $\mathcal{M}_t$ and all local models $\mathcal{M}^i_t$ 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 $\mathcal{M}^i_t$ 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 [54, 55].

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

```
 1: procedure FedAvg($\mathcal{M}_0$, $W_t \subseteq W, T$)
 2:     $t \leftarrow 0$
 3:     while $t \leq T$ do
 4:         $t \leftarrow t + 1$
 5:         $\forall\ w_i \in W_t$, the server exec Send($w_i$, $\mathcal{M}_t$)
 6:         Each $w_i \in W_t$ exec clientUpdate($w_i, \mathcal{M}_t$)
 7:         $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$        ▷ global update, (2.3)
 8:     end while                                                  ▷ Stopping criteria is met
 9: end procedure
10: function ClientUpdate(($w_i, \mathcal{M}_t$))
11:     $B$ = (split $\mathcal{D}_i$ into batches of size $\mathcal{B}$)
12:     Receive($w_i$, $\mathcal{M}_t$)
13:     for each local epoch i from 1 to $E$ do
14:         for each batch $b \in B$ do
15:             $\mathcal{M}^i_{t+1} \leftarrow \mathcal{M}^i_t - \eta g^i_t$        ▷ Local update, (2.1)
16:         end for
17:     end for
18:     Send($i$, $\mathcal{M}^i_{t+1}$)
19: end function
```

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 [22, 23]. Accordingly, a robust and efficient aggregation strategy is crucial to the success of FL.

2.2.2 Non-IID Data in Federated Learning

In most of ML and data science scenarios, it is typically assumed that the data are independently sampled from a similar joint distribution. This is known as assuming that the data are IID. To reliably 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

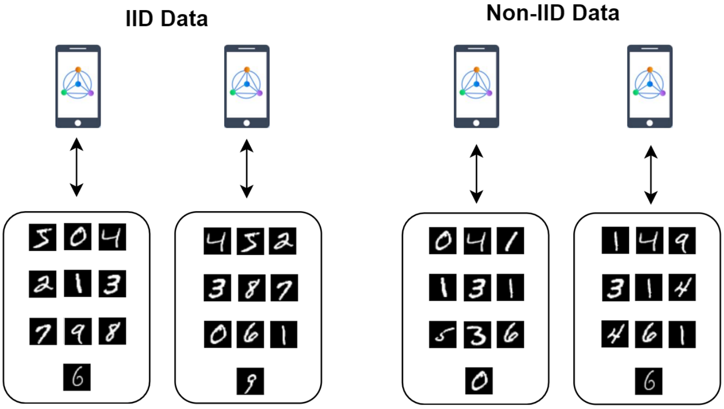


Figure 2.3: Illustration of IID vs. non-IID for MNIST dataset.

distributed). In other terms, the data is more uniform [56]. However, in a FL scenario, the devices involved in training are typically IoT devices that produce data that is unstructured and highly random. More specifically, client devices may have different label distributions, and some labels may be more available on some devices than others. As a result, Non-IID data is a reality in this setting [57, 58]. For instance, consider the classic scenario of digit recognition in the MNIST database of handwritten. In the global MNIST dataset, we would have 10% representation of each digit value from 0 to 9 [59]. In case the label distribution of the local datasets does not match the global value, the distribution is Non-IID, as we can see in Figure 2.3. This difference is significant as it greatly impacts the performance of the FL model, and research has demonstrated that Non-IID data distributions can reduce FL accuracy by as much as 55% [24]. 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 [24]. 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.

2.3 Clustering Analysis

Clustering 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 [60–62]. Cluster 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 [63]. Cluster 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 cluster 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 [64]. The clustering of client devices is an essential part of our proposed FL models, which involves grouping devices that share similar characteristics. This enables the participating devices to take advantage of collaboration with other devices that exhibit similar learning traits [65, 66]. This is advantageous in contrast to naive FL training, where irrelevant devices contribute to each other, which can potentially harm the performance of their respective datasets. 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. [67].
The focus of this thesis has been on k-medoids [68] and Markov Clustering (MCL) [69], due to their relevance to FL settings. k-medoids algorithm has a parameter to determine the number of clusters that defines how many clusters should be obtained. In contrast, MCL has a parameter called ”inflation” that indirectly affects the precision of the clustering. 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 point as the cluster center, k-medoids clustering chooses an actual point within the cluster to represent it. The k-medoids is the object within a cluster that is located at the center and has the lowest sum of distances to all other points [70]. In this algorithm, we begin by initially selecting k data points and iteratively moving towards the data 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 $P = \{p_1, p_2, \dots, p_n\}$, $i = 1, \dots, n$, we need to split into disjoint clusters k. The clustering of k -medoids selects k medoids $C = \{ob_1, ob_2, \dots, ob_k\}$ from the set P, to minimize the objective function known as the absolute error function (E) in 2.4:

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

Where E represents the sum of the absolute error. The variable p, which belongs to the set P, represents a data point that corresponds to an object in the cluster C_j from the set of clusters C. Additionally, ob_j denotes the representative object of the cluster C_j.

Therefore, medoids (also referred to as client devices in our thesis) are chosen from actual data to serve as cluster representatives. It is important to note that the Euclidean Distance (ED) will be adopted in k-medoids in this thesis.

- **MCL**: It is an efficient graph clustering algorithm. Unlike the k mean and k medoids, 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. Additionally, this algorithm is well suited for distributed computing [69]. 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 samples [71, 72], allowing the generation of valuable findings within large datasets. Moreover, these terms are often used in clustering techniques when data instances are split into clusters or to determine the similarity between data points within a given cluster [73]. Centroid-based algorithms represent a notable example of such methods. The selection of a distance metric has a significant impact on the effectiveness of the machine learning classifier. Therefore, how distances are calculated between objects is a critical factor in determining the performance of the classifier algorithm. Distance measures are formulated in Table 2.1.

Table 2.1: Distance measures.

Metric	Equation
Euclidean [74]	$D_{Euc}(p,q) = \sqrt{\sum_{i=0}^{n}(p_i - q_i)^2}$ 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 [75]	$Sim_{jac}(A,B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$ 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 [76]	$D_{was}(X,Y) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}$ 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) [77] is used to evaluate the clustering results. SI assess clustering validity and detect compact and well-separated clusters [78, 79].

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 $(j = 1, 2, \dots, k)$ of n_j objects 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 [80]. In [81], novel principles for anomaly detection analysis have been presented, focusing on eccentricity. Building upon these principles, a new algorithm named AutoCloud is proposed in [82]. Eccentricity refers to the degree to which a specific data instance differs from other instances and from its cluster. In this context, the calculation of the eccentricity ξ^j for the data sample i for a cluster of data C_j can be computed as [82]:

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

Where n_j represents the size of C_j, $\hat{p}_i$ denotes the empirical probability vector corresponding to the data sample 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 sample in a specific cluster.

In addition, the Chebyshev inequality has been used to apply a threshold to verify whether a data sample remains part of a current cluster [83]. A specific data sample i is considered to be a member of the group C_j if the following condition is met.

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

Where the parameter m $(m > 0)$ is defined by the user and directly affects the evaluation of the cluster, and v_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 and leads to satisfactory results for different datasets and different configurations [84]. We utilize eccentricity analysis, similar to the approach used in AutoCloud, to maintain the clustering solution of client devices.

3 Related Work

Our focus in this section is on previous work that tackles resources 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 PFL solutions.

3.1 Resource Aware Solutions

In the following, we will examine current research efforts that are closely related to resources, including communication expenses and energy efficiency of the smart device in FL.

3.1.1 Communication Efficiency

Current research solutions on communication efficiency in the field of a FL can be classified into the following categories.

- **Client selection**: During a FL training round, the communication bottleneck is exacerbated by the exchange of model updates between a large number of 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 that have the most favorable communication conditions in each round, it is possible to increase the average data rate and consequently reduce communication costs.

  – **Stochastic optimization**: Chen et al. [85] developed a method for selecting clients that achieves the most optimal probability sampling for clients during each round of communication. The study used SGD in unbiased estimation of the gradients of the loss function. In [86], the authors proposed a method called biased stochastic client Selection, which skews the probability distribution to select clients based on the loss value. The Power-of-Choice algorithm was introduced using the power of choices load-balance strategy [87]. The algorithm chooses a subset of clients and randomly selects an active set of clients from that subset. Clients with a higher global loss value have a greater chance of being selected. FLOB is a framework that utilizes biased stochastic optimization for client selection in FL [88]. FLOB identifies the best probability distribution that helps the server in choosing a subset of clients that achieves the minimum global loss. The work refers to this probability distribution as a lower-bound-near-optimal probability distribution. Another client selection strategy called AdaFL was developed with a bias [89]. It 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. Huang et al. [90] conducted a study in which they devised a stochastic approach that takes into account factors such as client communication or computational failures, as well as training biases. The multiple-play exponential-weight algorithm for the Exploration and Exploitation (Exp3) approach was used for client selections. The algorithm, called OCEAN (Online Client sElection and bAndwidth allocatioN), utilizes the existing resource status to efficiently choose the most optimal group of clients [90, 91].

- **Greedy selection**: The problem addressed by Balakrishnan et al. in [92] is tackled by FedAvg with Diverse Client Selection (DivFL), which is a greedy approach. Another study used a greedy algorithm for client selection, named FedMCCS in the context of IoT FL [93]. This model takes into account various criteria when selecting clients, including training time, memory size, CPU capacity, and energy consumption during training. In order to achieve early convergence without increasing communication overhead, Wang et al. [94] proposed a method called Communication-Mitigated Federated Learning (CMFL). This method selects a subset of clients whose parameter updates align the most with the overall trend of the global model. The relevance of an update is determined by comparing the signs of each parameter in the local update to those of the global update. Only clients whose updates meet the relevance threshold will communicate their updates for the current round.
- **Game theory**: Le et al. [95] 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 to participate in the round of communication, with the payoff being the client's selection. The authors in [96] developed a similar auction-based incentive strategy to efficiently choose clients in the FL context. In their work, they introduced a reputation model that considers the contributions of clients and their trustworthiness.

  - **Deep learning**: Deep Deterministic Policy Gradient (DDPG) was proposed, as a reliable method to enhance communication by effectively choosing clients for global aggregation in FL [97]. The authors in [98] introduced a method called FAVOR, which is based on deep Q-learning (DQL), to select the best sets of clients. Their approach specifically targets Non-IID data.

- **Compression**: Compression techniques play a crucial role in the efficient utilization of edge resources in FL. Specifically, these techniques aim to reduce the size of data and facilitate the exchange of models between edge devices and the central server, while maintaining the accuracy of the models [99]. Consequently, this reduces the time required to receive the data. In the context of FL, compression is particularly advantageous for handling DL models due to the typically large size of the communicated updates, allowing the execution of FL on edge devices with limited resources. Listed below are a variety of techniques that can be employed to compress models in FL:

  - **Quantization**: Quantization is a commonly used technique that involves decreasing the level of precision in both the data and models. For instance, quantization can be utilized to represent the models as integers of reduced precision, rather than transmitting high-precision floating point numbers [100]. By implementing this approach, it is possible to greatly decrease the amount of data and models that are transmitted, while still maintaining a high level of accuracy. The 1-bit quantization method proposed in [101] preserves only the sign of the gradient and addresses the impact on convergence speed by adding quantization error into the residual gradient. The QSGD algorithm was proposed in [102] and its convergence was proven in this work. It balances the trade-off between convergence and quantization levels and minimizes the communication expense by adapting the number of bits transmitted. The authors in [103] introduced the TernGrad algorithm, which involves quantizing floating-point numbers. They also provided a convergence proof, assuming that the gradient remains bounded. In [104], the authors suggested that gradients with larger magnitudes are more important. To address this, they introduced a nonlinear quantization algorithm called CosSGD, which utilizes the cosine function to allocate a finer quantization space for values with more significant gradients.

  - **Sparsifcation**: Gradient sparsifcation refers to selectively sending partial gradients and reducing communication costs by discarding some gradients with small contributions [105]. The gradients computed by the nodes in distributed SGD are frequently sparse, with the majority of gradient values being near 0. This exchange of gradients is not only unnecessary but also raises the cost of communication. The authors in [106] proposed to utilize a constant compression rate to choose the sent gradients. Sparsifying gradients using fixed proportions of positive and negative gradients was introduced in [107]. A sparse communication algorithm LAG, which adaptively computes a threshold in each communication round, targets to minimize part of the transmission of gradients [108].

  - **Pruning**

- **Updates dropping**:

- **Adaptive strategy**: Currently, several methods are employed to further reduce communication and resource utilization in the FL context. For example, communication strategies that can adapt, such as those induced by [94], have shown their effectiveness. A framework called 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 [109], the authors have indicated that if customers only partially participate in FL, there is the possibility of objective inconsistency, which could slow down convergence. In addition, they proposed a method that aimed at improving the convergence analysis by advocating the use of an optimal and unbiased sampling technique. The authors in [110] have introduced a method for adaptively aggregating partial models in FL using Reinforcement learning (RL). This strategy aims to optimize the selection of the client devices involved by determining the optimal number of devices.

3.1.2 Energy Efficiency

3.2 Personalized FL solutions

4 Methodology

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

4.1 Datasets and Baselines

First, we give details of the datasets and models applied to the different studies that demonstrate the performance of our proposed FL models on several real-world datasets, federated datasets, and models. Then, we introduce the baseline algorithms that are used to compare with our proposed FL models.

- **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 benchmark the proposed FL models on ten different datasets– MHealth [111], PAMAP2 [112], MNIST [113], FashionMNIST [114], CIFAR-10 [115], FEMNIST [116], CelebA [117], REALWORLD [118], HHAR [119] and Synthetic [120] – with respective learning models: (1) logistic regression; and (2) a CNN.

Table 4.1: Summary of the benchmarks used in this thesis.

Task	Model	Dataset	Classes	Papers
HAR	Logistic Regression	MHealth [111]	12	I, II
		PAMAP2 [112]	17	I, II
		REALWORLD [118]	8	V
		HHAR [119]	6	V
Image Classification	CNN	MNIST [113]	10	IV
		FashionMNIST [114]	10	IV
		CIFAR-10 [115]	10	IV
		FEMNIST [116]	62	IV, VI
		CelebA [117]	-	IV, VI
Cluster Identification	Logistic Regression	Synthetic [120]	-	VI

Table 4.1 provides details on the datasets used in our studies, including the model used and the available classes. In PAPERs I and II, we 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 used five datasets, namely the MNIST, Fashion MNIST, CIFAR-10, FEMNIST, and CelebA datasets. The MNIST, FashionMNIST, and CIFAR-10 are commonly used as benchmark datasets for image classification. Furthermore, we use LEAF datasets [120] that are more realistic than the simulated datasets. PAPER V introduces two practical datasets available online, REALWORLD and HHAR from the HAR domain. Finally, in PAPER VI, we used three LEAF datasets. FEMNIST, CelebA, and Synthetic Dataset are used to show the robustness of our proposed FL models.

- **Baseline methods**: In order to show that our FL proposed models can bring a better training performance and save communication costs, various methods are selected for comparison.
  - **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 2.2.1.
  - **FedProx** [121]: 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** [122]: 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)** [123]: 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 [124] calculates the contributions of each client in FL utilizing Shapley values.

This ensures that each party's contributions are correctly appreciated, and motivates high-quality ones to join as early as possible.

- **FL-Cohort**: The proposed algorithm [125] 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 [121]	MNIST, FashionMNIST, CIFAR-10	IV
CMFL [122]	MNIST, FashionMNIST, CIFAR-10	IV
CFL [123]	REALWORLD, HHAR	V
Deletion Approach [124]	Synthetic, FEMNIST, CelebA	VI
FL-Cohort [125]	Synthetic, FEMNIST, CelebA	VI

In PAPERs I, II, IV, V, naive FedAvg method was used as a benchmark to compare with our proposed FL models and with other FL methods in terms of communication overhead, accuracy and F-measure. FedProx was used in PAPER IV to compare the communication cost and accuracy of different FL methods. Similarly, in PAPER IV, CMFL, a method aimed at mitigating communication overhead in FL, was used to compare performance in terms of communication cost and accuracy with various FL methods. In PAPER V, CFL as a clustered FL algorithm is employed to compare the performance achieved to the proposed FL model. Finally, the deletion approach and the FL-Cohort, which are used to measure the client contribution, 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.

4.2 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 [126–128]. When a new algorithm is introduced, it is typical to demonstrate its enhancement over other 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 models against some other baseline methods in terms of communication cost, the model's accuracy, energy consumption, battery life of devices, etc. Table 4.3 lists the measures that were used primarily in the studies of our thesis. In FL, computational tasks are distributed among numerous less powerful 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. This communication involves rounds of communication between a server and its clients to exchange models among them. PAPERs I and IV proposed FL models which improve the performance of the FL scenario in terms of reducing communication overhead while achieving better F1 score/ accuracy values. The F-measure was used in PAPERs I, II, and V as an evaluation measure to compare the performance of our proposed FL methods with other FL methods, such as FedAvg and CFL methods. Energy consumption and battery lifetime measures are used in PAPER II as part of the multi-criteria evaluation to calculate the score of sensor nodes in the WNs settings. In PAPERs IV and VI, one of the measures used to compare the performance of different FL methods is the accuracy of FL used models. In PAPER VI, the frequency of each client selected or changed a cluster was used as a measure to calculate the contribution of each client, inspired by our studies, PAPERs I and V, respectively.

Table 4.3: Evaluation measures used across studies.

Evaluation measures	**Papers**
Communication overhead	I, IV
F-measure	I, II, V
Energy Consumption	II
Battery Lifetime	II
Accuracy	IV, VI
Frequency of client selection or modification of a cluster	VI

4.3 Research Methodology

Two main research methodologies were 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 [129, 130]. Our study focuses on reviewing current academic results related to context awareness using AI models, particularly in the context of SNs. In addition, in PAPERS I, II, IV, V, and VI, we use a research methodology based on implementation and experimentation [131]. In each of the studies presented in this thesis, new FL models are introduced and tested through experimentation. A range of experiments are carried out to verify the effectiveness of the algorithms using diverse datasets and 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]) using HAR datasets (MHealth [111] and PAMAP2 [112]), to reduce communication overhead between the central server and participants under the 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, a proposed *Energy-aware Multi-Criteria Federated Learning (EaMC-FL)* model was compared with FedAvg using 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 PAPAER 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 evaluated on publicly available datasets (MNIST [113], FashionMNIST [114] and CIFAR 10 [115]) and also on LEAF datasets (FEMNIST [116] and CelebA [117]) under IID and Non-IID data. The proposed algorithm is also compared with various state-of-the-art FL methods namely, FedAvg [5], FedProx [121], and CMFL [122]. 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 [118] and HHAR [119]). GP-FL has been compared to 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 in PAPERS I and V, respectively. Our proposed FL models have been compared to the Deletion Approach [124] and FL-Cohort [125] on three LEAF datasets. These LEAF datasets (Synthetic [120], FEMNIST [116] and CelebA [117]) are used to validate the approaches.

4.4 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.

- **Internal validity**: Internal validity refers to the impact of the experimental setup on the results [132, 133]. In this thesis, the threat of selection bias is presented, which can often be remedied by random sampling [134]. We split the experimental dataset into different sets. Specifically, in PAPERS I and II, we used 10 different test sets (cross-validation) of the dataset in the conducted experiments to avoid selection bias. 3-fold cross-validation on each training set was performed in PAPERS IV. In addition, in PAPER V, we performed 3-fold cross-validation on each experimental dataset. Finally, 5-fold cross-validation on each experimental data set was performed for several communication rounds in PAPER VI. Selection bias is not seen as a concern in PAPER III as it is a literature review.

- **External validity**: External validity refers to the extent to which the results of the experiment can be applied or generalized [132, 133] 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.

- **Construct Validity** Construct validity deals with issues surrounding the extent to which the outcomes align with the intended conceptual goals [135]. 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 discusses 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 run-time errors that are harder to detect than compile-time errors. In many instances, dedicated experiments are carried out to determine the appropriate parameters of the algorithm.

- **Conclusion validity**

5 Results

This chapter provides an overview of the results of the thesis. Three main directions have been recognized: resource-aware FL solutions, PFL solutions, and the study of edge-based AI for sensor networks. The achievements of the thesis are outlined and deliberated in these directions.

6 Summary of contributions

All research questions formulated in this thesis are stated and answered.

Bibliography

[1] K. M. Hazelwood, S. Bird, D. M. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, and X. Wang. “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. Qiu, T. Parcollet, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane. “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] Y. Zhang, D. Ramage, Z. Xu, Y. Zhang, S. Zhai, and P. Kairouz. “Private Federated Learning in Gboard”. In: *ArXiv* abs/2306.14793 (2023).

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

[9] 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.

[10] 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.

[11] 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`.

[12] 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).

[13] S. Huang, W. Shi, Z. Xu, I. W.-H. Tsang, and J. Lv. “Efficient federated multiview learning”. In: *Pattern Recognit.* 131 (2022), p. 108817.

[14] 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).

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

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

[17] 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.

[18] H. McMahan, E. Moore, D. Ramage, and B. A. y Arcas. “Federated Learning of Deep Networks using Model Averaging”. In: *arXiv preprint arXiv:1602.05629* (2016).

[19] 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).

[20] 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.

[21] 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.

[22] 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.

[23] 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).

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

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

[26] 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.

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

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

[29] H. B. M. et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *AISTATS*. 2017.

[30] 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.

[31] 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.

[32] 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.

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

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

[35] 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.

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

[37] 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.

[38] 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.

[39] 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).

[40] 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.

[41] J. Sun, A. Li, L. DiValentin, A. Hassanzadeh, Y. Chen, and H. H. Li. “FL-WBC: Enhancing Robustness against Model Poisoning Attacks in Federated Learning from a Client Perspective”. In: *Neural Information Processing Systems*. 2021.

[42] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. B. Calo. “Analyzing Federated Learning through an Adversarial Lens”. In: *International Conference on Machine Learning*. 2018.

[43] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. “How To Backdoor Federated Learning”. In: *ArXiv* abs/1807.00459 (2018).

[44] Y. Wen, J. Geiping, L. H. Fowl, M. Goldblum, and T. Goldstein. “Fishing for User Data in Large-Batch Federated Learning via Gradient Magnification”. In: *ArXiv* abs/2202.00580 (2022).

[45] H. B. M. et al. “Communication-efficient learning of deep networks from decentralized data”. In: *arXiv preprint arXiv:1602.05629* (2016).

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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).

[51] 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.

[52] 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).

[53] 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.

[54] 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.

[55] 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.

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

[57] 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.

[58] 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.

[59] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proc. IEEE* 86 (1998), pp. 2278–2324.

[60] E. E. Absalom, A. M. Ikotun, O. N. Oyelade, L. M. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu. “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.

[61] 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.

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

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

[64] 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.

[65] 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.

[66] 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.

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

[68] 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.

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

[70] 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.

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

[72] 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.

[73] 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).

[74] 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.

[75] 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).

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

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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`.

[82] 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`.

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

[84] 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.

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

[86] 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).

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

[88] 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.

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

[90] T. Huang, W. Lin, K. Li, and A. Y. Zomaya. “Stochastic Client Selection for Federated Learning With Volatile Clients”. In: *IEEE Internet of Things Journal* 9 (2020), pp. 20055–20070.

[91] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. “Gambling in a rigged casino: The adversarial multi-armed bandit problem”. In: *Proceedings of IEEE 36th Annual Foundations of Computer Science* (1995), pp. 322–331.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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).

[97] P. Zhang, C. Wang, C. Jiang, and Z. Han. “Deep Reinforcement Learning Assisted Federated Learning Algorithm for Data Management of IIoT”. In: *IEEE Transactions on Industrial Informatics* 17 (2021), pp. 8475–8484.

[98] H. Wang, Z. Kaplan, D. Niu, and B. Li. “Optimizing Federated Learning on Non-IID Data with Reinforcement Learning”. In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications* (2020), pp. 1698–1707.

[99] 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.

[100] 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.

[101] 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.

[102] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. “QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding”. In: *Neural Information Processing Systems*. 2016.

[103] 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.

[104] Y. He, M. Zenk, and M. Fritz. “CosSGD: Nonlinear Quantization for Communication-efficient Federated Learning”. In: *ArXiv* abs/2012.08241 (2020).

[105] 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).

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

[107] 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.

[108] 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.

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

[110] 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.

[111] 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.

[112] 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.

[113] 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`.

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

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

[116] 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.

[117] 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`.

[118] 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.

[119] 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).

[120] S. Caldas, P. Wu, T. Li, J. Konecný, H. B. McMahan, V. Smith, and A. Talwalkar. “LEAF: A Benchmark for Federated Settings”. In: *ArXiv* abs/1812.01097 (2018).

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

[122] 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.

[123] 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.

[124] 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.

[125] 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).

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

[127] 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).

[128] 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).

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

[130] 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.

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

[132] 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.

[133] 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.

[134] 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.

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