UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RODOLFO VITURINO NOGUEIRA DA SILVA

A DISTRIBUTED D2D CLUSTERING ALGORITHM TAILORED FOR HIERARCHICAL FEDERATED LEARNING IN A MULTICHANNEL ALOHA NETWORK

CURITIBA

2024

RODOLFO VITURINO NOGUEIRA DA SILVA

A DISTRIBUTED D2D CLUSTERING ALGORITHM TAILORED FOR HIERARCHICAL FEDERATED LEARNING IN A MULTICHANNEL ALOHA NETWORK

Um Algoritmo de Clusterização D2D Distribuído Adaptado à Aprendizagem Federada Hierárquica em uma Rede ALOHA Multicanal

Thesis shown as a requirement for obtaining the title of Master of Science from Universidade Tecnológica Federal do Paraná (UTFPR). Concentration Area: Telecommunications and Networks.

Advisor: Prof. Glauber Gomes de Oliveira Brante
Co-Advisor: Prof. Richard Demo Souza

CURITIBA

2024

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

RODOLFO VITURINO NOGUEIRA DA SILVA

A DISTRIBUTED D2D CLUSTERING ALGORITHM TAILORED FOR HIERARCHICAL FEDERATED LEARNING IN A MULTICHANNEL ALOHA NETWORK

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Telecomunicações E Redes.

Data de aprovação: 23 de Agosto de 2024

Dr. Glauber Gomes De Oliveira Brante, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Jamil De Araujo Farhat, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Jorge Augusto Meira, Doutorado - University Of Luxembourg

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 04/09/2024.

I dedicate this work to my family, for their
patience and encouragement.

ACKNOWLEDGMENTS

For the grace of life, I thank God.

For all the support and renounces made on my behalf, I thank my parents, Marco and Luana.

For the patience and comprehension during my sleepless nights dedicated to work, I thank my wife, Katlyn.

I extend my heartfelt appreciation to my advisor, Prof. Glauber Brante, for his exemplary guidance from start to finish, helping me even beyond his working hours.

I also thank my co-advisor, Prof. Richard Demo Souza, for his valuable corrections and suggestions.

Finally, I am grateful to all the professors on the examining board for accepting the invitation.

“A dream... It’s something you do for yourself, not for others.” (Miura, 1998)[a].

[a] “*Um sonho... É algo que você faz por si mesmo, não pelos outros.*” (Miura, 1998, tradução própria).

RESUMO

Em uma época caracterizada pela interconectividade contínua, a quantidade de dispositivos de internet das coisas registrou um crescimento substancial nos últimos anos, com projeções indicando uma maior expansão. Neste contexto, a aprendizagem federada desempenha um papel importante no futuro das comunicações sem fios, oferecendo inúmeras vantagens sobre as abordagens tradicionais de aprendizagem centralizada, incluindo preservação da privacidade dos dados, utilização reduzida de largura de banda, maior precisão e personalização. A aprendizagem federada está sendo muito utilizado em aplicações de internet das coisas, onde múltiplos dispositivos contribuirão com seu aprendizado sobre seus dados privados locais, a fim de construir um modelo global, que será posteriormente agregado e treinado pela estação base para que seja redistribuído de volta aos dispositivos. O objetivo deste trabalho é criar um algoritmo de agrupamento dispositivo-a-dispositivo (D2D), nomeado D2D de agrupamento de curta distância (D2D-SRC), capaz de otimizar uma comunicação ALOHA multicanal dentro de um sistema de aprendizagem federada hierárquica, sendo esta aprendizagem a principal motivação. A seleção de um protocolo sem fio e método de transmissão de dados apropriado é crucial para a aprendizagem federada, por isso foi adotado o protocolo multicanal ALOHA devido à sua natureza assíncrona e implementação simples em comparação com outros protocolos. Além disso, ALOHA é uma alternativa prática interessante à computação over-the-air (AirComp), que é uma técnica muito utilizada na literatura em sistemas de comunicação sem fio para computar funções de dados distribuídos diretamente pelo ar sem decodificar mensagens individuais. Os resultados numéricos mostram que a nova abordagem de agrupamento dispositivo-a-dispositivo proposta é capaz de atingir altas taxas de agrupamento, melhorando a eficiência geral do sistema ao reduzir drasticamente o erro alcançável, ao mesmo tempo que permite que uma estação base sirva mais dispositivos usando os mesmos recursos.

Palavras-chave: Agrupamento; Dispositivo-a-Dispositivo; Aprendizado Federado; ALOHA Multicanal.

ABSTRACT

In an age characterized by seamless interconnectivity, the quantity of Internet of Things (IoT) devices has experienced substantial growth in recent years, with projections indicating further expansion. In this context, Federated Learning (FL) plays an important role in the future of wireless communications, offering numerous advantages over traditional centralized learning approaches, including data privacy preservation, reduced bandwidth usage, improved accuracy, and customization. FL is being very used on IoT applications, where multiple devices will contribute with their learning over its local private data in order to build a global model, which will be further aggregated and trained by the Base Station (BS) so that it is redistributed back to the devices. The aim of this work is creating a Device-to-Device (D2D) clustering algorithm, named D2D Short Range Clustering (D2D-SRC), capable of optimizing a multichannel ALOHA communication within a hierarchical FL system, with this learning being the main motivation. Selecting an appropriate wireless protocol and data transmission method is crucial for FL, therefore it was adopted the multichannel ALOHA protocol due to its asynchronous nature and simple implementation compared to other protocols. Also, ALOHA is an interesting practical alternative to the over-the-air computation (AirComp), which is a technique much used in the literature in wireless communication systems to compute functions of distributed data directly through the air without decoding individual messages. The numerical results show that the novel D2D clustering approach proposed is capable of hitting high clustering rates, improving the general efficiency of the system by drastically reducing the achievable error, while allowing a BS to serve more devices using the same resources.

Keywords: Clustering; Device-to-Device; Federated Learning; Multichannel ALOHA.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND INITIALS

AirComp	Over-the-Air Computation
BS	Base Station
BLE	Bluetooth Low Energy
CH	Cluster Head
D2D	Device-to-Device
D2D-SRC	Device-to-Device Short Range Clustering
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FL	Federated Learning
HFL	Hierarchical Federated Learning
IoT	Internet of Things
MAC	Medium Access Layer
MAP	Multiple Access Protocol
ML	Machine Learning
OSI	Open Systems Interconnection
SGD	Stochastic Gradient Descent
UTFPR	Universidade Tecnológica Federal do Paraná

LIST OF SYMBOLS

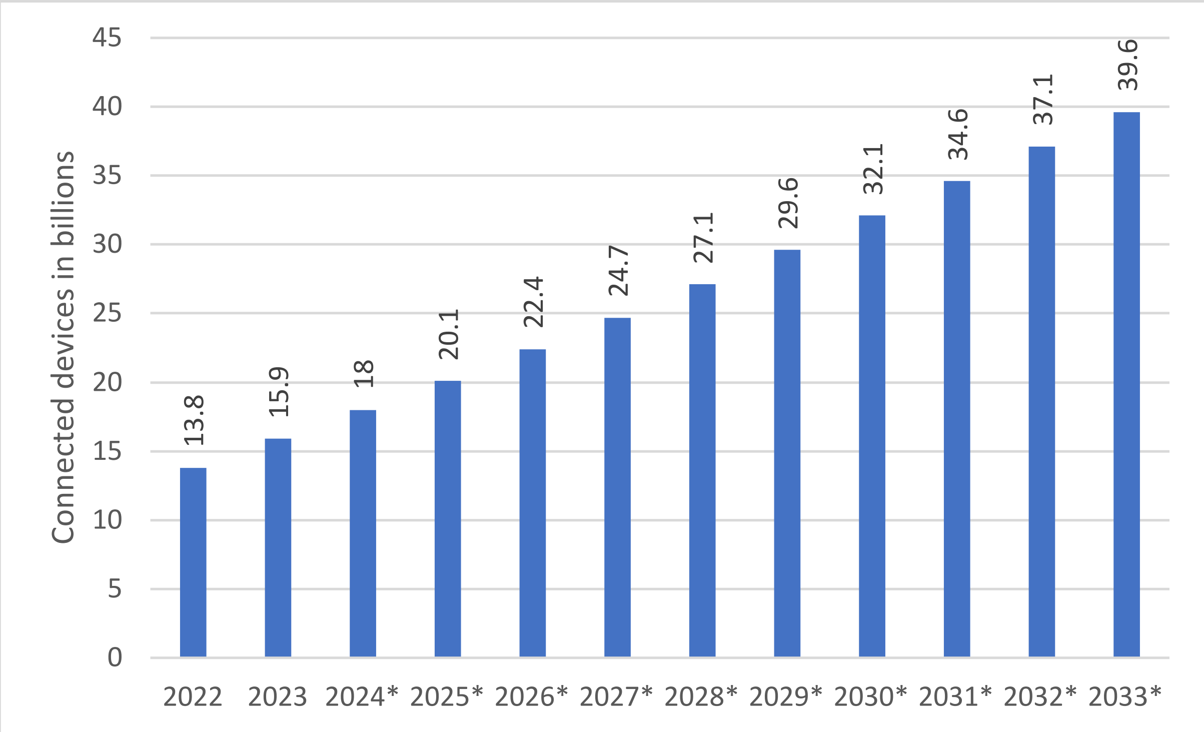
$\mathcal{M}$	Global ML model
f	Predictive function
D_{train}	Training dataset
$\mathcal{K}$	Set of IoT devices engaged in a FL task
K	Total number of devices
k	Device number
$\mathcal{D}_k$	Dataset of device k
A	Aggregator
Q	Local training function
F	Aggregator's fusion function
t	Number of iterations inside a FL round
$\mathcal{Q}$	Query for the current round of FL
q_t	Query for the current iteration
$u_{\text{k,t}}$	Model update
U_t	All expected model updates at the aggregator
α	Circular area that the device can communicate
R_{BS}	Radius of the BS
X_k	Cartesian position X of device k
R_k	Distance of the device k to the BS in meters
θ_k	Angle of the device k with respect to the BS
R_{D2D}	Communication range of the D2D protocol
$\mathcal{U}$	Uniform distribution
M	Number of available channels at the BS
p	Access probability of the devices
η	Average number of transmitted model updates without collisions
K_{ch}	Number of CHs
$\overline{C}$	Average cluster size
$\mathbf{x}_k$	Input data of device k
$\mathbf{y}_k$	Output data of device k
$\mathbf{w}$	Weight vector
L	Length of the weight vector
$f_k(\mathbf{w})$	Minimization of the loss function at device k
w_k	Local update of device k
h_k	Step size of device k
$w(t)$	Weight vector at iteration t
p_{comp}	Probability that a CH can compute its local update
p_i	Probability that cluster i transmit its local update
ψ	Feedback signal from the BS at the end of each FL iteration
w_i	Vector of weights of the aggregated model at the ith CH
μ	Step size
P	Sum of the probability to transmit the local update from all the CHs
$\hat{P}_t$	Estimate of P at iteration t
$S_{i,t}$	Activity variable of CH i at iteration t
ϵ	Radius parameter of the DBSCAN algorithm
C_{max}	Maximum number of devices that can belong to a single cluster
β	Balancing cluster parameter of the proposed clustering algorithm

SUMMARY

1 INTRODUCTION

In 2023 there were 16.7 billion active Internet of Things (IoT) devices according to (Analytics, 2023) and 15.9 billion active IoT according to (Statista, 2024), with a projection of 32.1 billion until 2030 (Statista, 2024), as illustrated in Figure 1. The rapid proliferation of connected devices in the Industry 4.0 era poses challenges regarding the scalability of the networks and efficient data transmission (Duan; Da Xu, 2021).

Figure 1 – Number of IoT connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033.



Source: adapted from (Statista, 2024)

In the last years, the focus of IoT has moved from acquiring data to data intelligence (Vasyutynskyy; Luntovskyy, 2022). In this sense, Federated Learning (FL) stands out, offering numerous advantages over traditional centralized learning approaches (Ludwig; Baracaldo, 2022). FL not only offers a certain level of privacy for the users (Eldar *et al.*, 2022), but also tends to transmit fewer data during this process, as the raw data remains local (Ludwig; Baracaldo, 2022). FL leverages the increased hardware and computational capabilities of modern IoT devices to train a model on the edge side. On the other hand, the server, or the information aggregator, orchestrates these local training procedures and, in addition, periodically combines the models into a single global one, which is then sent back to the IoT devices (Eldar *et al.*, 2022; Ludwig; Baracaldo, 2022).

Selecting an appropriate wireless protocol for FL is crucial (Hellström *et al.*, 2022), even more so when considering IoT networks, with their limitations in terms of bandwidth and energy. In this regard, one of the most used random access methods in IoT networks is the multichannel ALOHA protocol, in which devices randomly choose different frequency channels to transmit. A remarkable example of multichannel ALOHA tuned to FL is presented in (Choi; Pokhrel, 2019), which iteratively optimizes the access probabilities of users to transmit their local updates to the central aggregator, taking into account the number of available channels, achieving enhanced performance in terms of convergence time and utilization of wireless resources.

In addition, the work in (Silva *et al.*, 2022) introduces a novel method for allocating wireless resources in a multichannel ALOHA setup, specifically for FL systems where the devices train multiple models simultaneously. The method aims to improve the convergence time of FL systems compared to uniform and fully-shared channel allocations, and enhances the performance of FL systems with multiple models, ensuring fair and efficient use of wireless communication resources. Furthermore, the authors in (Da Silva; López; Souza, 2023) discuss an energy-aware approach for FL in edge devices that harvest energy. It focuses on integrating energy harvesting devices into an FL network with multichannel ALOHA, aiming to ensure low energy outage probability and successful execution of future tasks. The paper proposes a novel user sampling technique that allows devices to make informed decisions about their participation in FL iterations, considering their energy levels and the computation cost.

In a different direction, over-the-air computation (AirComp) with FL is attracting a lot of interest, as in (Yang *et al.*, 2020; Xing; Simeone; Bi, 2021; Zhu *et al.*, 2024), allowing multiple devices to access the wireless channels non-orthogonally, by means of the superposition property of multiple-access channels. However, although the spectral efficiency is increased with respect to ALOHA, a significant synchronization complexity is introduced in this case. ALOHA protocols, as those employed in (Choi; Pokhrel, 2019; Silva *et al.*, 2022; Da Silva; López; Souza, 2023), are perhaps more practical alternatives than AirComp. An important advantage of ALOHA-based methods over AirComp is the relaxed synchronization requirement. In IoT networks it may be very challenging, especially in terms of energy consumption, to maintain strict synchronization. However, ALOHA protocols are prone to collisions.

One way to reduce the impact of collisions in ALOHA is to apply clustering among the devices. Network clustering involves partitioning nodes into multiple clusters, each having a Cluster Head (CH) (Deshpande; Patil, 2013; Soni; Dey, 2014). The CH is responsible for forwarding to the Base Station (BS) the data from the cluster members (Deshpande; Patil, 2013; Soni; Dey, 2014), therefore leading to fewer transmissions to the BS and potentially reducing collisions. Moreover, a clustered network may present many other advantages, such as improving the energy efficiency (Deshpande; Patil, 2013; Soni; Dey, 2014; Ye *et al.*, 2005; Jubair *et al.*, 2021; Singh; Kumar; Mishra, 2015) and the scalability of sensor nodes (Ye *et al.*, 2005; Singh; Kumar; Mishra, 2015; Soni; Dey, 2014). Over the years, many authors have employed clustering strategies in Device-to-Device (D2D) enabled networks. For instance, the work in (Huang *et al.*, 2018) applied the K-means algorithm to cluster D2D devices based on the interference among them. Another work in (Lee *et al.*, 2012) combined a genetic algorithm with K-means for interference mitigation.

Besides, a FL system can be partitioned into layers, which is commonly known as Hierarchical FL (HFL) (Song *et al.*, 2023; Yuwen; Yu; Xiangjun, 2023; Khelf; Driouch; Ajib, 2023), introducing a tiered structure to the FL process, allowing for a more nuanced model aggregation. This can lead to improved performance in certain applications, especially those with complex infrastructure or geographical data dependencies (Rana *et al.*, 2022). Still, according to (Rana *et al.*, 2022), HFL extends the traditional FL process by enabling a more efficient model aggregation based on application needs or characteristics of the deployment environment, such as resource capabilities and/or network connectivity, and balances the processing across the cloud-edge continuum, which can be particularly beneficial for complex cyber-physical systems like the IoT. Moreover, note that D2D communication can be an enabler of HFL, by creating different tiers (devices to CHs, and CHs to BS) of communication and aggregation.

In this sense, this document proposes the use of D2D clustering to improve the performance of FL in an IoT network using multichannel ALOHA and a hierarchical structure, and is analyzed in terms of number of devices that a single BS can serve, and also in terms of the final achievable error. The novel proposed D2D clustering algorithm, named D2D Short Range Clustering (D2D-SRC), differs from the traditional clustering methods applied in the literature, such as the K-means or Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which is the algorithm that this work is based on. DBSCAN tries to cluster points based on their density and devices distance, and it works well when clusters are dense enough and separated by low-density regions, but fails for dense HFL systems, due to this type of system requiring some restrictions to be taken into account, such as the maximum number of devices in a single cluster, their inter-cluster distance and no need of specifying the number of clusters. As the results will show, the introduction of the clustering scheme is capable of lowering the final error achieved of the devices, while allowing a single BS to serve more gadgets, thus optimizing the resources' allocation. With the D2D-SRC algorithm, a simple polling strategy can behave better than an ALOHA strategy with fixed access probability that does not have the D2D-SRC.

1.1 Objectives

1.1.1 Main objective

Develop a D2D clustering algorithm for optimizing multichannel ALOHA communication within a hierarchical FL system, which enables a single BS to serve more devices while reducing the achievable error.

1.1.2 Specific objectives

- Promote a bibliographic review for the use of clustering techniques for optimizing the resource's allocation in wireless networks and FL;
- Develop a clustering algorithm that respects the constraints related to an HFL system that uses the multichannel ALOHA communication protocol;
- Formalize the FL in a multi-layer link, adapting local aggregations in the CH and in the BS using the ALOHA protocol;
- Analyze the impact of applying clustering solutions for a BS in terms of the number of devices served and achievable error;

1.2 Work structure

This study is organized in order to present the preliminary concepts in Chapter 2, which includes the study of clustering techniques for optimizing the resources' allocation in wireless networks, how FL works and why using it for wireless networks, and also ALOHA protocol. Next, Chapter 3 begins with the presentation of the system model, with the roles of the devices, CHs and BS, followed by the HFL strategy adopted and the details and presentation of the proposed D2D-SRC clustering Algorithm. Then, Chapter 4 presents the results obtained and their respective discussions and, finally, Chapter 5 shows considerations of the thesis and proposes future works to deepen this study.

2 PRELIMINARY CONCEPTS

This Chapter introduces a few preliminary concepts, important to the development of the scheme proposed in Chapter 3. Section 2.1 starts by giving an overview about Machine Learning (ML) and why it is a relevant technique, specially in the context of communications, as well as the categories of learning that exists in it. Going deeper on ML, Section 2.2 explores the decentralized process, and uses tree-based FL as a potential approach to distribute the steps of learning on edge devices, lowering the burden of the central server and securing privacy, while Section 2.3 explores more about clustering for optimizing resources, such as improving the number of devices that can participate in the learning process at each iteration, lowering the number of communications made with the main server, and compressing the information to be sent to the BS, which are essential features for scaling. Section 2.4 focus on explaining the needs for communication protocols and the motivation for choosing ALOHA.

2.1 Machine Learning and Communications

An agent is learning if it improves its performance after making observations about the world. Learning can range from the trivial, such as jotting down a shopping list, to the profound, as when Albert Einstein inferred a new theory of the universe. When the agent is a computer, we call it ML: a computer observes some data, builds a model based on the data, and uses the model as both a hypothesis about the world and a piece of software that can solve problems (Russell; Norvig, 2020). ML and wireless communications are two of the most rapidly advancing technologies of our time, and we are now witnessing the confluence of these two fields, as learning at the network edge has advantages in terms of latency and privacy, and it capitalizes on the fact that many learning tasks, such as those supporting automated driving, are locality specific (Eldar *et al.*, 2022).

Even though impressive advances in communication techniques over the last few decades were accomplished, there is still a large gap in approaching the theoretical limits in complex networks of heterogeneous devices. According to (Eldar *et al.*, 2022), current design approaches are built upon engineering heuristics, where it is tried to avoid or minimize interference through various access technologies. Many authors have proposed along the years ML-driven solutions for certain blocks of the communication chain, and these solutions were shown to outperform conventional designs in many scenarios, such as the works (Ye; Li; Juang, 2017; Mashhadi; Gündüz, 2021; Boloursaz Mashhadi; Gündüz, 2020), whose researches were based on channel estimation, or (Xu *et al.*, 2018) who presented research regarding channel equalization and (Wen; Shih; Jin, 2018; Mashhadi; Yang; Gündüz, 2020), whose researches were involved with channel state information compression and feedback.

In order to process the huge amount of data that is being generated each day, considering that, for example, an autonomous car can generate from 5 to 20 terabytes of data each day according to (Gündüz *et al.*, 2020), it is needed to change from the centralized solutions to the network edge, considering that a centralized cloud solution is not scalable and might violate the latency requirements of the underlying application, particularly in the inference stage. Not only the scale becomes a problem, but also having all the data centralized in a single server concerns security and privacy, as most of the information collected from edge devices, such as smart meters, health sensors, or entertainment devices collects personal sensitive information. (Gündüz *et al.*, 2020) states that, as a result, communication becomes key to an intelligent network edge, and potential solutions must allow edge devices not only to share their data but also computational resources in a seamless and efficient manner.

There are three categories of learning in ML, which can be classified as supervised, unsupervised and reinforcement, as can be seen in (Buduma, 2017). The goal of supervised learning is to teach the input-output relation of a function, and a well-known example inside wireless communications system is the use of regression for estimating the channel coefficients from noisy received versions of known pilot signals, as in (Mashhadi; Gündüz, 2021; Ye; Li; Juang, 2017).

The reinforcement learning class of problem aims to learn how to interact in a random unknown environment based on feedback received in the form of costs or rewards following each action. This kind of approach is very used on gaming environments and also communication problems, such as (Madhavan *et al.*, 2024), who used Q-learning for improving the resources' allocation. According to (Eldar *et al.*, 2022), RL methods can be used in wireless networking problems for which the environment is known to be stationary, yet it does not have a good model to characterize its statistical behavior, which might be the case, for example, when operating over unlicensed bands, where it is difficult or impossible to model the statistics of devices activation, spectrum activity, traffic arrivals, and channel variations.

Finally, in unsupervised learning problems, the training data are not labeled, and there is no explicit output value, while the goal is to learn functions that describe the input data. This kind of approach may help to get insights about the data and the problem that is being solved, but it depends on the human interpretation. Several problem solving approaches in communication systems involve clustering, dimension reduction, and density estimation. Specifically speaking of clustering, it works acting mainly as source compression or quantization (Eldar *et al.*, 2022).

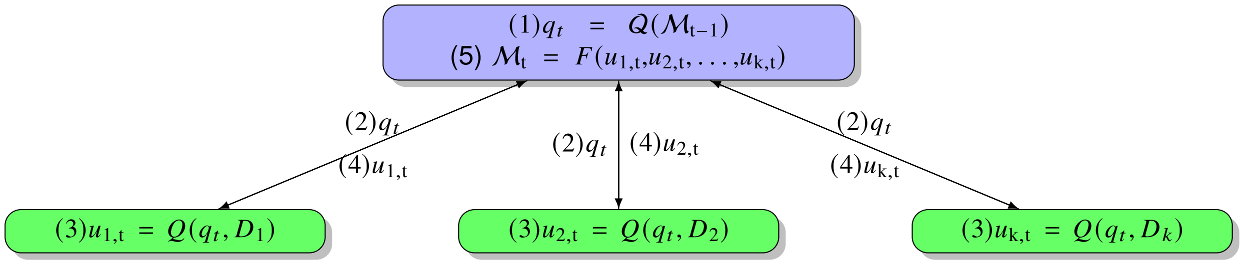
The learning process of a ML algorithm could be either centralized, or decentralized. As (Eldar *et al.*, 2022) describes, centralized learning is based on the star topology, which is a central or edge server that can act as an access point or a BS in the wireless context, orchestrating the devices participating in the collaborative training process. While this may be preferable in the presence of a BS that can communicate directly with all the devices in the coverage area, it may increase the burden on the cell-edge devices. The multiple access channel from the devices to a single central server may become a bottleneck and limit the number of devices that can participate in the learning process at each iteration, which is not what is desired in this work.

2.2 Federated Learning

The alternative to the centralized learning fashion is using a decentralized learning, where the devices directly communicate with each other in a D2D manner, and the main change happens in the consensus step, in which the devices combine their local models with those of their neighbors according to a certain rule that is driven by the network topology. The consensus step comes from the consensus problems, which is an optimization problem that can be deeper seen in (Boyd *et al.*, 2011).

Unlike the star topology used in the centralized learning, the models at different devices are no longer fully synchronized after each global averaging step, introducing additional noise in the framework (Eldar *et al.*, 2022). In this context, FL is an approach to ML in which the training data is not managed centrally, instead, the data is retained by data parties that participate in the FL process and is not shared with any other entity, and, for not bringing data together in a centralized repository, as it could be problematic

Figure 2 – FL process at an abstract level.



Source: adapted from (Ludwig; Baracaldo, 2022)

either for privacy, regulatory, or practical reasons, this makes FL an increasingly popular solution for ML tasks (Ludwig; Baracaldo, 2022).

The definition and the algorithm described below are adapted from (Ludwig; Baracaldo, 2022), and it is stated that like any ML task, FL trains a model $\mathcal{M}$ representing a predictive function f on training data D_{train}. In contrast to centralized ML, D_{train} is partitioned between K devices, where each device owns a private training dataset $\mathcal{D}_k$. A FL process involves an aggregator A and a set of devices $\mathcal{K}$. It is important to note that the dataset of the device is only accessible by itself, and A has no knowledge of any dataset. Figure 2 shows how the FL process is conducted at this abstract level.

In order to train a global machine learning model $\mathcal{M}$, the aggregator and the devices perform a FL algorithm that is executed in a distributed way at the aggregator and the devices. The main algorithmic components are each device's local training function Q, which performs the local training on dataset $\mathcal{D}_k$, and the aggregator's fusion function F, which combines the results of each device's Q into a new joint model. There can be a set of iterations of local training and fusion using the index t. The execution of the algorithm is coordinated by sending messages between devices and the aggregator. The overall process runs as follows:

1. The process begins at the aggregator and, in order to train the model, it is used a function Q that takes as input the model of the previous iteration of the training $\mathcal{M}_{\mathrm{t-1}}$ at iteration t, and generates a query q_t for the current iteration. When the process starts, $\mathcal{M}_0$ may be empty or just randomly seeded, and some FL algorithms may include additional inputs for Q, tailoring queries to each device, but for maintaining things simple and without loss of generality, it is used this simplified approach.
2. The query q_t is sent to the devices and requests information about their respective local model or aggregated information about each device's dataset.

Example queries include requests for gradients or model weights of a neural network, or counts for decision trees.

3. When receiving q_t, the local training process performs the local training function Q that takes as input query q_t and the local dataset $\mathcal{D}_k$ and outputs a model update $u_{\mathrm{k,t}}$. Usually, the query q_t contains information that the device can use to initialize the local training process. This includes, for example, model weights of the new, common model $\mathcal{M}_{\mathrm{t}}$ to initialize local training, or other information for different model types.
4. When Q is completed, $u_{\mathrm{k,t}}$ is sent back from the device k to the aggregator A, which collects all the $u_{\mathrm{k,t}}$ from all devices.
5. When all expected devices model updates $U_t = (u_{1,\mathrm{t}}, u_{2,\mathrm{t}}, \ldots, u_{\mathrm{k,t}})$ are received by the aggregator, they are processed by applying the fusion function F that takes as input U_t and returns $\mathcal{M}_{\mathrm{t}}$.

This process can be executed over multiple iterations and continues until a finish context is triggered, like the maximum number of training rounds, and it is possible to introduce different variants to this basic approach to FL, such as not all devices participating in every iteration t. Considering that the devices' selection can be random, based on devices' characteristics, or on the merits of prior contributions, in this study, it is considered that the devices' selection will happen according to (Choi; Pokhrel, 2019), focusing on the ALOHA communicating protocol.

Also, while the described approach with a single aggregator is the most common and practical for most scenarios, other alternative FL architectures have been proposed, as example, every device might have its own, associated aggregator A_k, querying the other devices. The set of devices might be partitioned between aggregators, and a hierarchical aggregation process may take place. This work uses a tree-based topology of FL, with this hierarchical aggregation process, as will be explained in Section 3.2.

2.3 Clustering

As discussed in Section 2.1, clustering is a type of unsupervised ML commonly used as source compression, and has a good synergy with FL, acting mainly as an add-on on top the FL scheme. Similar to what was described in Section 2.2 in a clustered FL, instead of aggregating the information of a single device, it is possible to aggregate the information of multiple devices inside clusters, optimizing the resources of the BS from the number of communications perspective. The properties that determine the organization of the clusters usually differ in the literature depending on what problem is being attempted to be addressed, for example, if one wishes to develop an architecture where it is desired for all devices to participate equally, but some of them are more prone to dropouts than others, it is possible to group them based on their dropout probabilities (Ludwig; Baracaldo, 2022).

There are a few challenges that could be addressed when developing these clustered FL systems, such as the clustering and selection criteria, profiling, dynamic properties, and privacy. In the concerns of clustering, the rules used to group devices are one of the most important design decision that needs to be made in such systems, since that it completely defines the priorities of the framework. After clustering, the second most important step is how to control the system, such that the clustered properties can be taken advantage of, and in order to clarify this problem, if one frequent selection of devices with biased datasets happens, this will lead to a biased model. The chosen rules and properties considered in this document of both the FL and the clustering scheme are presented on Chapter 3.

Over the literature, (Ghosh *et al.*, 2020) presented a clustering system based on the loss values of the devices' gradients, where the authors provided theoretical analysis on how losses can be used to mitigate the performance loss of a model due to data heterogeneity. The authors analyzed the convergence rate of this algorithm with squared loss, generic strongly convex, and smooth loss functions. Another work is (Sattler; Müller; Samek, 2020), which presented a novel Federated Multi-Task Learning framework, which exploited geometric properties of the loss surface of the FL training systems, and grouped the devices into clusters based on the cosine similarities of their weights, focusing more on the privacy aspects.

2.4 ALOHA protocol

Multiple Access Protocols (MAP) are nothing but channel allocation schemes that possess desirable performance characteristics. In (Gerla; Lazar; Kuehn, 1990), it is stated that in terms of known layering models, such as the Open Systems Interconnection (OSI) reference model, these protocols reside mostly within a special layer called Medium

Access Control (MAC) layer that is between the data link control layer and the physical layer. It is also stated that the need for MAPs arises not only in communication systems, but also in many other systems such as storage facilities, computer systems, or a server of any kind where a resource is shared, and consequently accessed by a number of independent users.

There are a magnitude of MAPs, such as Carrier Sense Multiple Access, Time Division Multiple Access, Frequency Division Multiple Access, Code Division Multiple Access or Space Division Multiple Access. In this thesis it is used the ALOHA protocol, a non-centralized MAP, considering that in this kind of protocol the nodes behave according to the same set of rules and, in particular, there is no single node coordinating the activities of the other, thus, facilitating the FL scheme. According to (Gerla; Lazar; Kuehn, 1990), over these many MAPs, the ALOHA family of protocols is probably the richest, and its popularity is due to being the first random access technique introduced, as well as for the simplicity of its straightforward implementation.

ALOHA family of protocols belongs to what is called random re-transmission protocol, in which the success of a transmission is not guaranteed, and the explanation for this is that when two or more devices are trying to transmit using the same channel during the same time-slot, a collision happens, impacting on the information to not being received correctly, and therefore making the device having to re-transmit its packet until it eventually transmits its full information. The scheduling of the transmission is one of the great concerns on this type of protocol, this way, numerous variations of ALOHA protocols have been implemented over the years. The pure ALOHA protocol is considered to be the baseline, which is a single-hop system, where a newly generated information is transmitted immediately in the expectation that no interference by others happens, and, in the case where there is a collision, it is scheduled a re-transmission to a random time in the future. This randomness exists in order to ensure that the same set of packets does not continue to collide indefinitely.

Meanwhile, the slotted ALOHA variation of the ALOHA protocol is simply that of pure ALOHA but with a slotted channel, where users are restricted to start transmission of packets only at the slot boundaries, thus making the vulnerable period reduced to a single slot. The results shown in (Roberts, 1975) derived slotted ALOHA protocol having a higher throughput compared with the pure ALOHA protocol.

One relevant aspect of choosing multichannel slotted ALOHA-based protocols for FL in IoT networks is that they present significant advantages over scheduling-based techniques. For example, when devices enter into deep sleep mode for saving battery, if the BS request for those dormant devices to send its updates, there is a resource waste, as no information is sent during that period. Or the fact that many times when the BS asks for some user content, the devices may still be calculating its local updates, also leading to a resource waste.

3 DEVELOPMENT

3.1 System Model

As the system model, it considered a BS serving a set $\mathcal{K}$ of IoT devices engaged in an FL task, with $K = |\mathcal{K}|$ denoting the total number of devices. It is assumed that the devices are grouped into clusters, so they do not transmit directly to the BS. Instead, the devices transmit their local updates to the CH via a short-range D2D protocol, which does not conflict with the transmissions from the CHs to the BS, and aggregates the cluster data locally to further transmit it to the BS using a multichannel ALOHA protocol. The BS, in turn, aggregates all local trained models into a global model, and broadcasts the model back to the devices. Therefore, this is a two-tier HFL approach, first aggregating cluster weights at the CH, which then transmits this local aggregation to the BS.

It is assumed that the IoT devices are distributed in a circular region with area $\alpha = \pi R_{\mathrm{BS}}^2$, where R_{BS} is the communication radius in meters within which the BS can interact with the IoT devices. The BS is positioned at the center of this area, such that its Cartesian position (X, Y) is $(0, 0)$, while the Cartesian position of the k-th device is

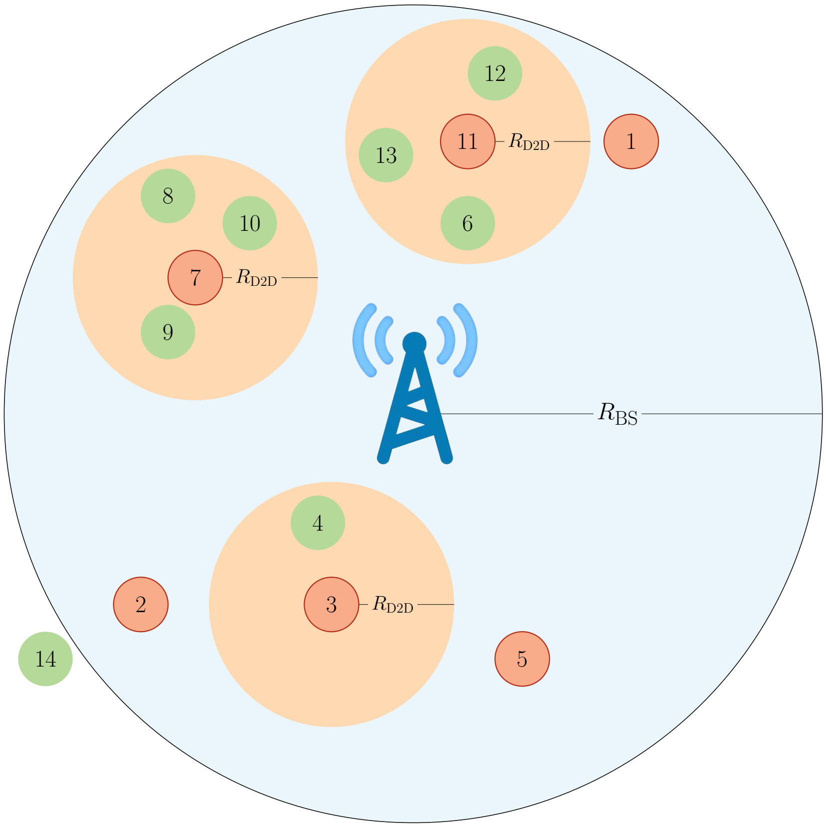
$$X_k = R_k \cos(\theta_k), \tag{1}$$

$$Y_k = R_k \sin(\theta_k), \tag{2}$$

where R_k is the distance from the k-th device to the BS. This distance follows a uniform distribution, $R_k \sim \mathcal{U}(1, R_{\mathrm{BS}})$, ensuring that the distance of the devices always falls within the communication range of the BS, while a minimum distance of 1 m with respect to the BS is assumed. In addition, $\theta_k \sim \mathcal{U}(0, 2\pi)$ is the angle of the devices with respect to the BS.

It is also assumed that the IoT devices can operate in D2D mode, employing a small range communication protocol such as Bluetooth Low Energy (BLE), as it enables the devices to last for years using a single coin battery with a connection time that happens in just a few milliseconds (Le, 2017). In this scenario, the IoT devices communicate with other nearby devices in order to create small clusters using a D2D clustering algorithm. The communication range of the D2D protocol is denoted as R_{D2D}, while each i-th cluster has C_i devices.

Figure 3 – System model, composed of a BS at the center of the cell of radius R_{BS}, serving a set of IoT devices organized into clusters of radius R_{D2D} operating in D2D mode.



Source: own authorship (2024)

Figure 3 illustrates a representative scenario. In this example, the BS can communicate with devices numbered from 1 to 13, but cannot communicate with device 14, as it is beyond its communication radius R_{BS}. Also, note that device 3 can communicate with device 4 using D2D technology, but it cannot communicate with any other device, as none of them are within its R_{D2D} communication radius. Out of many possible topologies for FL in a multi-level approach, in this work, the proposed scheme focuses on the tree-based topology (Wu *et al.*, 2024), which is also illustrated in Figure 3, where there are three clusters, and each cluster sends its information through the CHs, represented by devices 3, 7 and 11. Devices 1, 2, and 5 are not within any cluster. However, herein, they are considered as CHs of clusters with a single device. One last note in Figure 3 is that it would be feasible for device 14 to communicate with the BS through device 2, when employing the proposed D2D-SRC algorithm. However, throughout this thesis, it is not considered cases where devices are situated beyond the communication radius R_{BS}. This decision ensures fairness when comparing scenarios with and without the clustering algorithm using the same number of devices.

Moreover, it is employed a multichannel ALOHA MAC protocol, so that M different channels are available at the BS. By denoting p as the access probability of the devices – *i.e.*, the probability that a device sends its local update to the BS – it is known that the maximum average number of successful transmissions of local updates is achieved if $p = \frac{M}{K}$ (Choi, 2020). Then, the average number of transmitted model updates without collisions, in a non-clustered approach, is given by (Choi, 2020)

$$\eta_{\text{non-clustered}} = K\, p \left(1 - \frac{p}{M}\right)^{K-1} \approx M\, e^{-1}. \tag{3}$$

Clustering the devices reduces the competition to access the BS, which consequently reduces the collision probability and improves the accuracy of the FL model. Assuming that the number of CHs is $K_{\text{ch}} \leq K$, the maximum average number of transmissions of local updates is now achieved if $p = \frac{M}{K_{\text{ch}}}$. Then, given an average cluster size $\overline{C}$, the average number of transmitted model updates without collisions in a clustered approach becomes

$$\eta_{\text{clustered}} = \overline{C}\, K_{\text{ch}}\, p \left(1 - \frac{p}{M}\right)^{K_{\text{ch}}-1} \approx \overline{C}\, M\, e^{-1}, \tag{4}$$

which, since $\overline{C} \geq 1$, yields $\eta_{\text{clustered}} \geq \eta_{\text{non-clustered}}$.

3.2 Hierarchical Federated Learning

Essentially, HFL implements a distributed Stochastic Gradient Descent (SGD) approach consisting of three steps: local computation, aggregation (in two tiers), and device selection (Eldar *et al.*, 2022). Initially, each device updates itself with the latest global model available from the BS, and then locally minimizes its loss function with an SGD update. Each device k is assumed to have a data set $\mathcal{D}_k = \{\mathbf{x}_k, y_k\}$, with the same size $\forall k$, where $\mathbf{x}_k$ and y_k represent the input and the output of device k, respectively (Konecný *et al.*, 2016). In addition, there is a global weight vector $\mathbf{w}$, of length L, associated with the minimization of the loss function at each device, $f_k(\mathbf{w})$, given by (Choi; Pokhrel, 2019)

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{K} \sum_{k=1}^{K} f_k(\mathbf{w}), \tag{5}$$

while the loss function for linear regression is

$$f_k(\mathbf{w}) = \frac{1}{2}\left|\mathbf{x}_k^T\mathbf{w} - y_k\right|^2. \tag{6}$$

3.2.1 Local Updates

Local updates depend on the loss function (Choi; Pokhrel, 2019), and loss functions are data-dependent (Eldar *et al.*, 2022). This document considers that the minimization employs the gradient descent and that the local update of device k is

$$\mathbf{w}_k(t+1) = \mathbf{w}(t) - h_k\left(\mathbf{x}_k^T\mathbf{w}(t) - y_k\right)\mathbf{x}_k, \tag{7}$$

where $h_k > 0$ is a step size and $\mathbf{w}(t)$ is the weight vector at iteration t. All the devices, indiscriminately if they are a cluster member or CH, perform this procedure.

3.2.2 Aggregation: First Tier (at the CHs)

Because it is being used HFL, the updates of the devices, or the weight vectors, are first sent to the CHs of the cluster that the devices belong to, which is then responsible for aggregating the information into a single weight vector (this aggregation includes the CH own local update as well). This process is called the first tier, or layer, of the HFL. This aggregation step is written as

$$\mathbf{w}_i(t+1) = \sum_{k=1}^{C_i}\mathbf{w}_k(t+1), \tag{8}$$

where the index $i \in \{1,\ldots,K_{\text{ch}}\}$ represents the CH. It is also important to remark that the aggregated model vector $\mathbf{w}_i(t+1)$ has the same size as the weight vector of each device $\mathbf{w}_k(t+1)$. Therefore, the size of the aggregated model is not impacted by the number of devices C_i in the cluster, considering that all devices have fixed weight lengths.

3.2.3 Aggregation: Second Tier (at the BS)

The second layer is the transmission of the CH aggregated cluster updates to the BS, which is then responsible for making a second aggregation of the information sent by all CHs, finally creating the t-th version of the model. With K_{ch} CHs and K devices, the aggregation at the BS is given by

$$\mathbf{w}(t+1) = \frac{1}{K} \sum_{i=1}^{K_{\text{ch}}} \mathbf{w}_i(t+1). \tag{9}$$

Finally, $\mathbf{w}(t+1)$ is broadcast back to the devices, and this whole process is repeated until a determined number of iterations t is reached.

3.3 Selection of CHs

In the considered scenario, there are $M < K_{\text{ch}}$ channels to communicate with the BS, so that only a subset of M CHs can successfully upload their information simultaneously at each iteration t of the HFL round. In order to avoid collisions, there are many strategies in the literature to select or sample which devices participate during each iteration. Three models, as discussed in more depth in (Choi; Pokhrel, 2019), are considered in this work and detailed as follows.

3.3.1 Polling

In the polling strategy, the CHs are explicitly selected by the BS, where it is asked from the BS for batches of CHs to transmit their data, usually following an order. It is important to highlight that this approach often results in wasted resources, as the CHs selected may not carry much useful information or be inactive. Note that in this case CHs are scheduled, therefore it is not a random access-based method.

3.3.2 ALOHA with Fixed Access Probability

In this scheme, instead of the BS selecting CHs to transmit the data as in the Polling model, each CH independently chooses one of the multiple channels to upload its information. This selection follows the multichannel ALOHA protocol, with an access probability given by

$$p = \min\left\{\frac{M}{K_{\text{ch}}}, p_{\text{comp}}\right\}, \tag{10}$$

where p_{comp} is the probability that a CH can compute its local update.

3.3.3 ALOHA with Optimized Access Probability

As a means to increase the access efficiency of the multichannel ALOHA system, the norm of the local updates can be taken into account. In other words, the access probability p_i of each CH i can be adapted since the higher the norm, the larger the novelty of the information transmitted by the CH. Therefore, according to (Choi; Pokhrel, 2019), the access probability can be optimized by doing

$$p_i^* = \max\Big\{\min\big\{\,(e \ln ||\mathbf{w}_i|| - \psi)\,, 1\big\}, 0\Big\}, \tag{11}$$

where $\psi = e \ln \lambda$ is a feedback signal from the BS at the end of each iteration, while λ satisfies $\sum_i p_i^* = M$ and $\mathbf{w}_i$ is the vector of weights of the aggregated model at the i-th CH. In cases where the CH cannot send its local updates, p_i is considered 0. This approach takes into account both the novelty of the local updates as well as a measure of the channel utilization.

The feedback signal ψ is transmitted by the BS to the CHs at the end of each iteration t together with the novel global model. Such variable is adaptively determined by (Choi; Pokhrel, 2019)

$$\psi_{\mathrm{t+1}} = \psi_t + \mu(\hat{P}_t - M), \tag{12}$$

where μ is the step size and $\hat{P}_t$ is an estimate of $P = \sum_{i=1}^{K_{\mathrm{ch}}} p_i$ at iteration t given by

$$\hat{P}_t = \sum_{i=1}^{K_{\mathrm{ch}}} S_{i,t}, \tag{13}$$

with $S_{i,t} \in \{0, 1\}$ representing the activity variable of CH i at iteration t, so that $S_{i,t} = 1$ if the CH i transmits its local update at iteration t, otherwise $S_{i,t} = 0$.

3.4 Proposed D2D-SRC Algorithm

This section presents the proposed D2D-SRC algorithm intended to increase the number of participating devices in each FL round, therefore reducing the error norm at the BS. One important aspect that was considered during the creation of this algorithm is that the number of clusters does not need to be informed beforehand. In a practical application, it is desirable that the devices have the ability to form clusters by themselves.

IoT devices may enter or leave the network over time, creating dynamic and unpredictable environments, and an algorithm that can automatically determine the number of clusters can adapt to these changes without needing centralized intervention. Further ahead, it is made clear how the clusters are formed and how the CHs are defined in our algorithm. As a benchmark, it is used DBSCAN (Ester *et al.*, 1996), which is a classical approach for clustering, and which does not require the number of clusters beforehand.

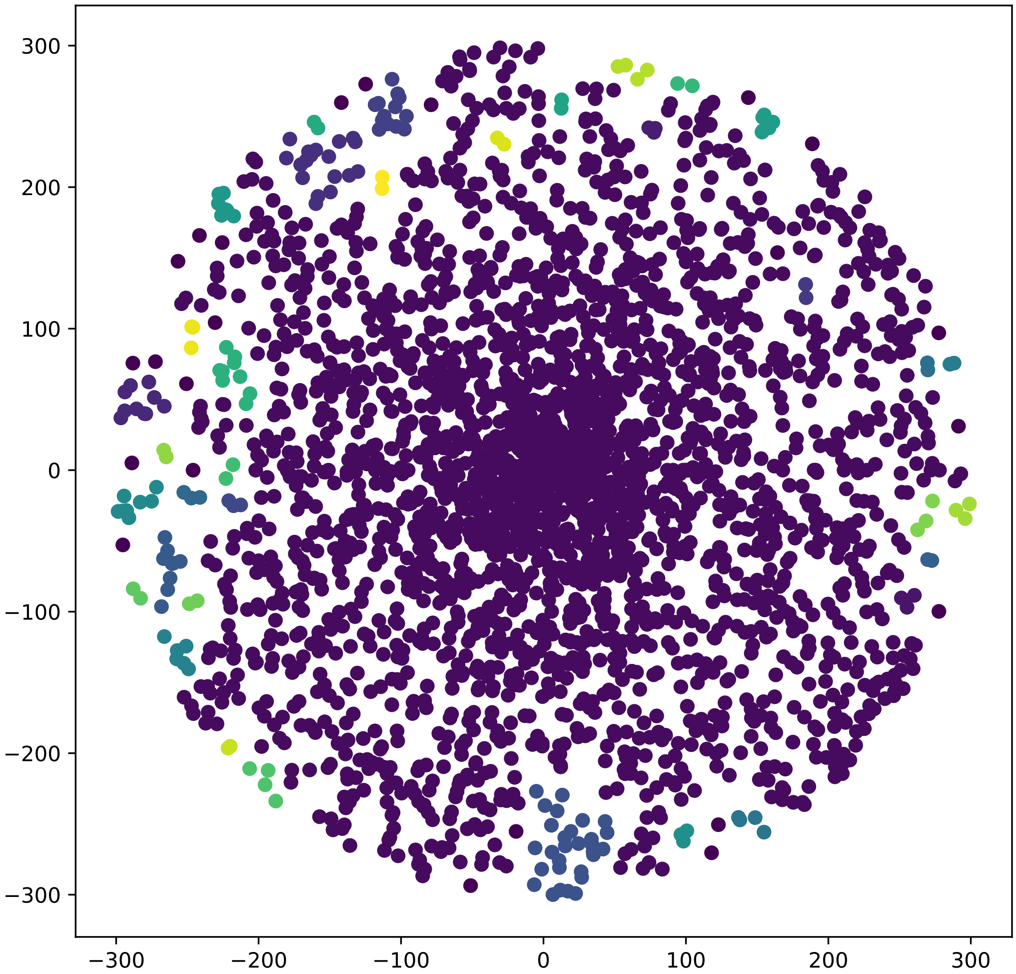
3.4.1 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

In a macroscopic view, DBSCAN can be described by the following steps:

1. **Parameters Initialization:** All devices, associated with a Cartesian position system, are considered non-clustered at this phase. The parameter ϵ acts as a radius around a reference device to evaluate if other devices can be considered its neighbors. Additionally, there is a parameter that dictates the minimum number of neighbors required within the ϵ radius for a device to be considered a core point.
2. **Neighborhood Counting:** This step consists of randomly selecting an unvisited device from the set and checking its neighborhood, then counting the number of neighbors within the range of the ϵ of that device.
3. **Classification:** During the neighborhood check in the previous stage, if the minimum number of neighbors is satisfied, then that device is labeled as a "core point". If not, it is labeled as noise. The noise labels can still change if they are absorbed by other core points, otherwise they remain as noise.
4. **Clusters Creation:** Core points mark all their neighbors and make them part of their cluster.
5. **Convergence:** Continuously iterate through the previous steps, going through the unvisited devices until all of them have been labeled.
6. **Cluster Chain Expansion:** By going through all core points, DBSCAN either creates a new separate cluster if there is no other nearby core point, or merges the clusters if they are within the distance of the ϵ parameter.

Notice that when the chain expansion is performed in step 6, it significantly affects the formation of the clusters when using the DBSCAN algorithm, so that devices can be clustered in a way that results in a few number of clusters, with a high variation in

Figure 4 – Example of the applied DBSCAN algorithm with $K = 3000$ randomly deployed devices, resulting in the formation of 36 clusters.



Source: own authorship (2024)

the number of devices participating in each of them. Figure 4 illustrates the application of the DBSCAN algorithm in a scenario with a single BS with coverage radius $R_{\text{BS}} = 300$ m, serving $K = 3000$ randomly deployed devices. It is also set $\epsilon = 15$ m, which would be equivalent to a BLE transmission range for D2D mode. It is possible to observe that devices are grouped into 36 clusters, which are illustrated in the figure by neighboring points with the same color, with a single cluster (in purple) grouping the vast majority of the devices.

However, such a configuration may not be practical for a D2D scenario, with a fixed communication range. DBSCAN can create clusters in a way some devices cannot communicate with a chosen CH, even when belonging to the same cluster, as they become too far away. Remembering that the goal in this work is to perform D2D clustering in a way that all members of a cluster are within one hop of the CH, then, R_{D2D} must dictate the maximum radius of the cluster. To address this issue, it is proposed a change from a minimum number of points to a maximum number of devices that can belong to the same cluster, augmented by a restriction of the R_{D2D} so that every cluster member can directly communicate with its CH.

Furthermore, another disadvantage of the DBSCAN algorithm is the lack of a specified method to choose CHs. Therefore, the D2D-SRC algorithm proposed in this work includes different constraints related to the D2D setup.

3.4.2 Proposed Algorithm

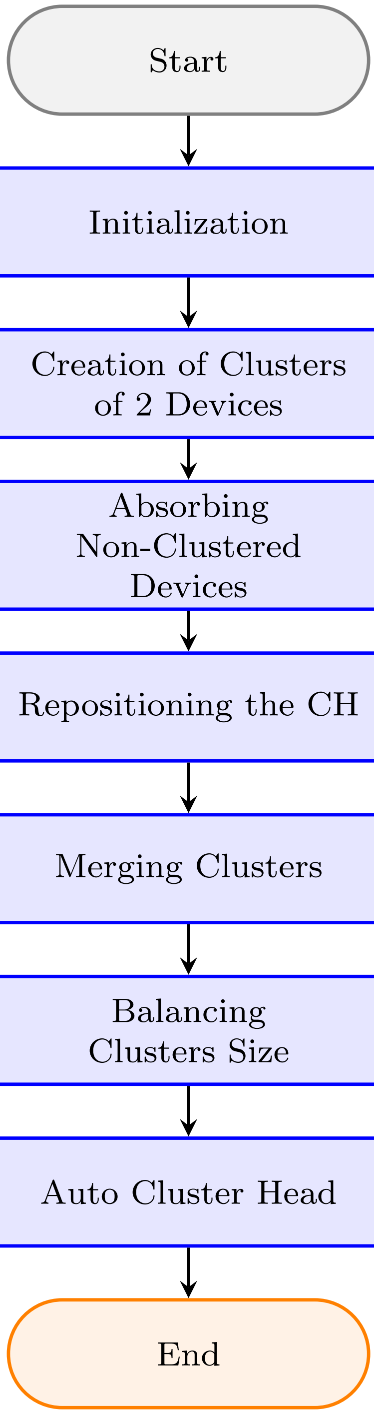
In the proposed D2D-SRC algorithm, there are two main variables: the maximum number of devices that can belong to a cluster $C_{\max}$, and a balancing parameter β, responsible for growing smaller clusters in size while reducing bigger clusters based on a threshold value. Essentially, clusters that are lower in size than β absorb devices from nearby clusters that are greater than β in size.

Figure 5 lists the seven steps of the proposed clustering algorithm with a flowchart, which are further detailed as follows [1].

1. **Parameters Initialization:** All devices around the BS, associated with a Cartesian position system, are non-clustered at this phase. The parameter R_{D2D} dictates the maximum range from any device to its CH. In order to start clustering, the BS broadcasts a signal to all devices, including the maximum duration of each step, which occurs in D2D mode.
2. **Creation of Clusters of Two Devices:** Devices transmit a *clustering request* aiming to find neighbors within their D2D range R_{D2D}. Reminding that there is no synchronization for this step, non-clustered devices transmit clustering requests randomly and wait for a reply. The first neighbor that replies to the clustering request forms a cluster. Without loss of generality, the device that

[1] All python code that was developed during this work may be accessed in: `https://github.com/rodolfoviturino/Clustering-For-Federated-Learning-With-ALOHA`.

Figure 5 – Flowchart illustrating all the steps necessary for the proposed D2D-SRC algorithm.

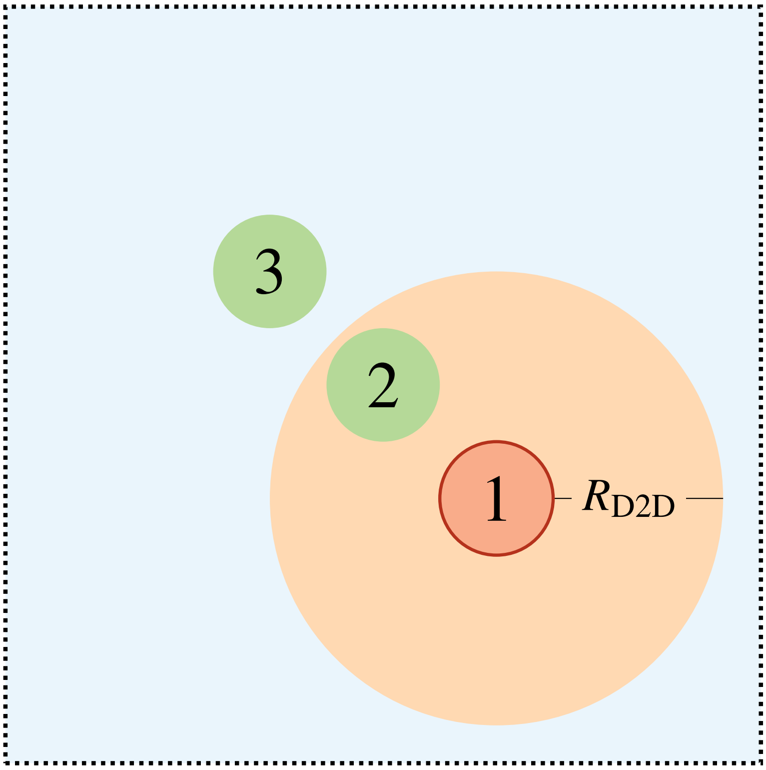


Source: own authorship (2024)

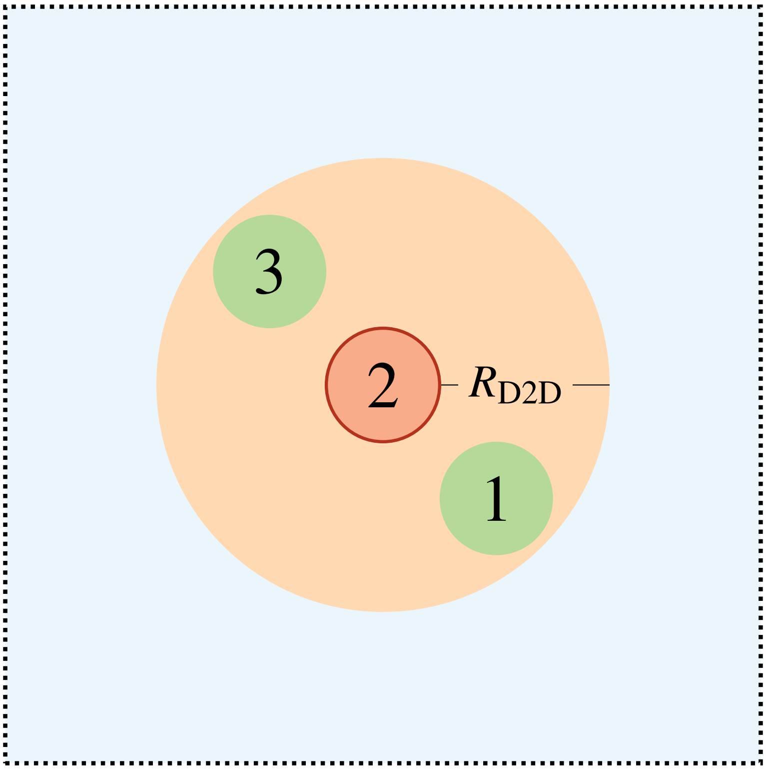
responded to a clustering request is defined as a cluster member, while the device that transmitted the request is elected as the CH. If a given device cannot find any non-clustered neighbor, it may still get clustered in the next step.

3. **Absorbing Non-Clustered Devices:** The previously created clusters attempt to grow by absorbing other non-clustered devices, respecting the D2D communication radius R_{D2D} and the maximum cluster size $C_{\max}$. In this step only the CHs transmit clustering requests in order to find non-clustered neighbors. Notice that this step creates clusters with different sizes. Nevertheless, step 6 will further attempt to reduce the size of big clusters, increasing the smaller ones.
4. **Repositioning CH:** For the clusters with size $C_i = 2$ after step 3 there is one additional attempt to absorb non-clustered devices, which is done by switching the status of CH with the other cluster member. This switch is performed only if this change allows the cluster to grow to size $C_i = 3$ devices. In order to understand the motivation behind this step, two different scenarios are shown in Figure 6. Figure 6(a) represents a first scenario with a cluster of size two, composed of devices 1 (CH) and 2 (cluster member). It is also possible to note that the CH cannot absorb the non-clustered device 3 because it is beyond the communication radius R_{D2D}. In the second scenario illustrated by Figure 6(b), the CH is switched from device 1 to device 2, thus allowing device number 3 to participate in the cluster and, consequently, growing the number of devices of the cluster from $C_i = 2$ to $C_i = 3$. Also, notice that this procedure of switching CHs could be extended to clusters with bigger sizes. However, the overhead of the clustering algorithm grows exponentially, so the search space to clusters of size $C_i = 2$ is restricted. In addition, the experimental results have shown that it is very rare to grow from $C_i = 2$ to $C_i > 3$, so this step only considers growing from $C_i = 2$ to $C_i = 3$ for limiting the algorithmic complexity.
5. **Merging Clusters:** Now, each CH checks for other CHs within the D2D communication radius. If so, the CHs confirm if both clusters can be fully merged, *i.e.*, if both CHs and all cluster members can be fully absorbed by one of the CHs, respecting R_{D2D} and $C_{\max}$. This step is a communication between CHs, so both CHs handshake and check if the merge is possible after having finished step 4. In addition, it is assumed that each CH exchanges the list (*i.e.,* the IDs) of its cluster members so that the R_{D2D} range with respect to the CHs can be checked.

Figure 6 – Switching the CH inside a cluster.



((a)) device 1 as CH, forming a cluster with $C_i = 2$ devices.



((b)) device 2 as CH, with the cluster size increasing to $C_i = 3$ devices.

Source: own authorship (2024)

6. **Balancing Clusters Size:** Each CH whose cluster is smaller than the threshold value β checks if there are other devices (from other clusters) that are in the R_{D2D} neighborhood, which could potentially be absorbed. If yes, the CH checks if the device belongs to a cluster bigger than β and, if affirmative, this device switches from the bigger cluster to the smaller one. This step assumes a two-way communication, so that a CH of a cluster with a small size sends an *increase request*, searching for neighbors from other clusters. Then, if a device receives the increase request, it consults its own CH for permission to switch membership and replies to the other CH request either affirmatively or negatively.
7. **Auto Cluster Head:** Finally, the devices that remain non-clustered, instead of being considered as noise as in the DBSCAN algorithm, are considered as clusters of size $C_i = 1$, where the device itself is the CH, unless any of the previous steps absorbs this device.

4 RESULTS AND DISCUSSIONS

This chapter evaluates the performance of the proposed D2D-SRC. As in (Choi; Pokhrel, 2019; Silva *et al.*, 2022), it is considered a linear regression problem where both the dataset and the ground-truth model are $L \times 1$ independent Gaussian random vectors with unity variance per element. The results consider the mean of 1000 FL rounds, where each round is composed of t iterations. The simulation parameters considered in this work are summarized in Table 1, and Section 4.1 shows the results step-by-step from each of the proposed stages of Section 3.4.2, followed by the results of its application to the FL system described in Section 3.2.

Table 1 – Simulation parameters.

Variable	Value
BS communication radius (R_{BS})	300 m
D2D communication radius (R_{D2D})	15 m
Scattered devices radius (R_k)	$\mathcal{U}(1, R_{\mathrm{BS}})$
Scattered devices angle (θ_k)	$\mathcal{U}(0, 2\pi)$
Number of devices (K)	$\{500, 1000, 2000, 3000\}$
Length of information (L)	(1×10)
Number of channels (M)	10
Probability to compute local updates (p_{comp})	0.1
Learning rate	0.01
Step size (μ)	0.1
Number of FL rounds	1000
Maximum cluster size ($C_{\max}$)	10
Balancing parameter (β)	4

Source: own authorship (2024)

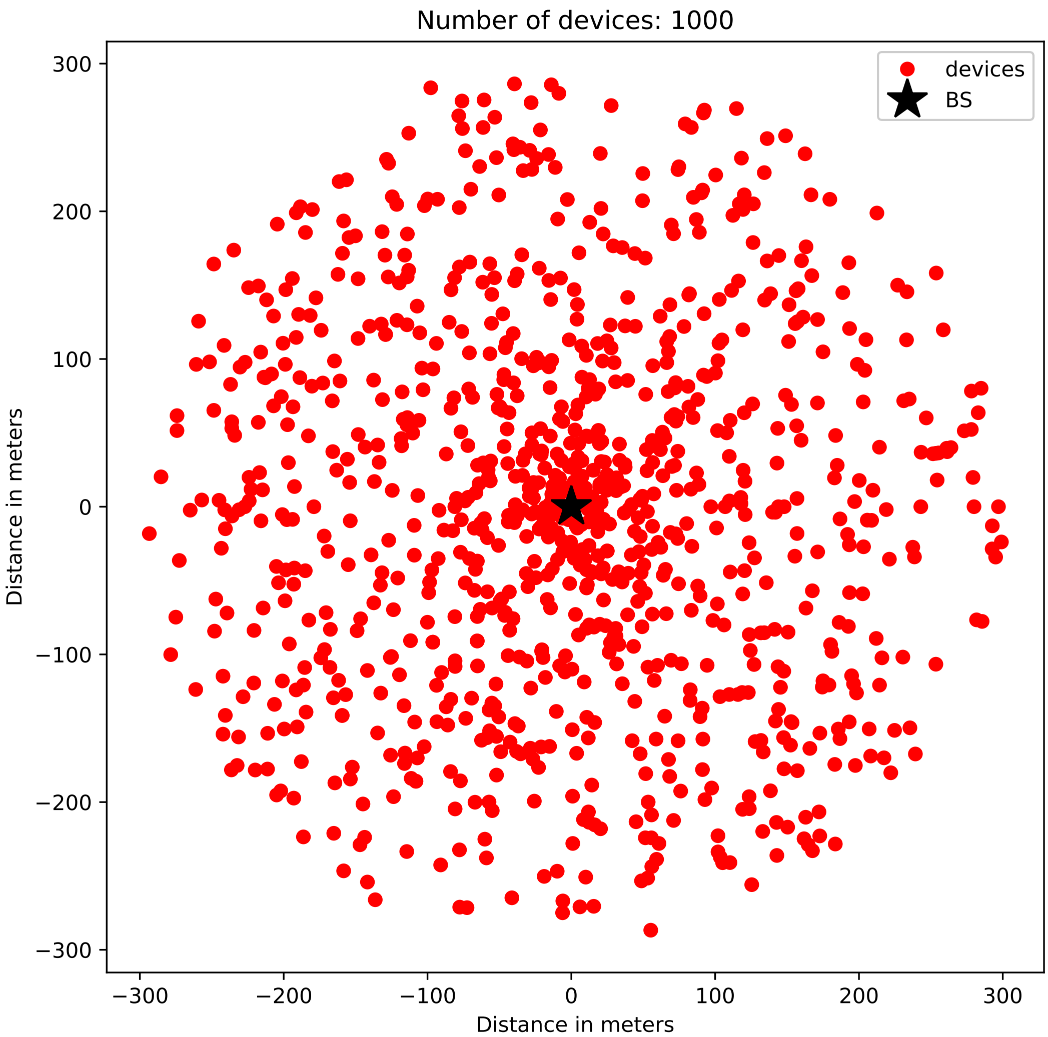
4.1 Clustering

For illustrating the impacts of each of the proposed steps in the novel D2D-SRC algorithm, it is considered $K = 1000$ devices and $R_{\mathrm{BS}} = 300$ m, with each cluster respecting the $R_{\mathrm{D2D}} = 15$ m distance from each device to its CH and the maximum cluster size of $C_{\max} = 10$.

In the following, it is described and illustrated the D2D-SRC algorithm results step-by-step:

1. **Parameters Initialization:** Figure 7 illustrates a network scenario at the beginning of the algorithm, where all devices are unclustered and circularly distributed around the BS.

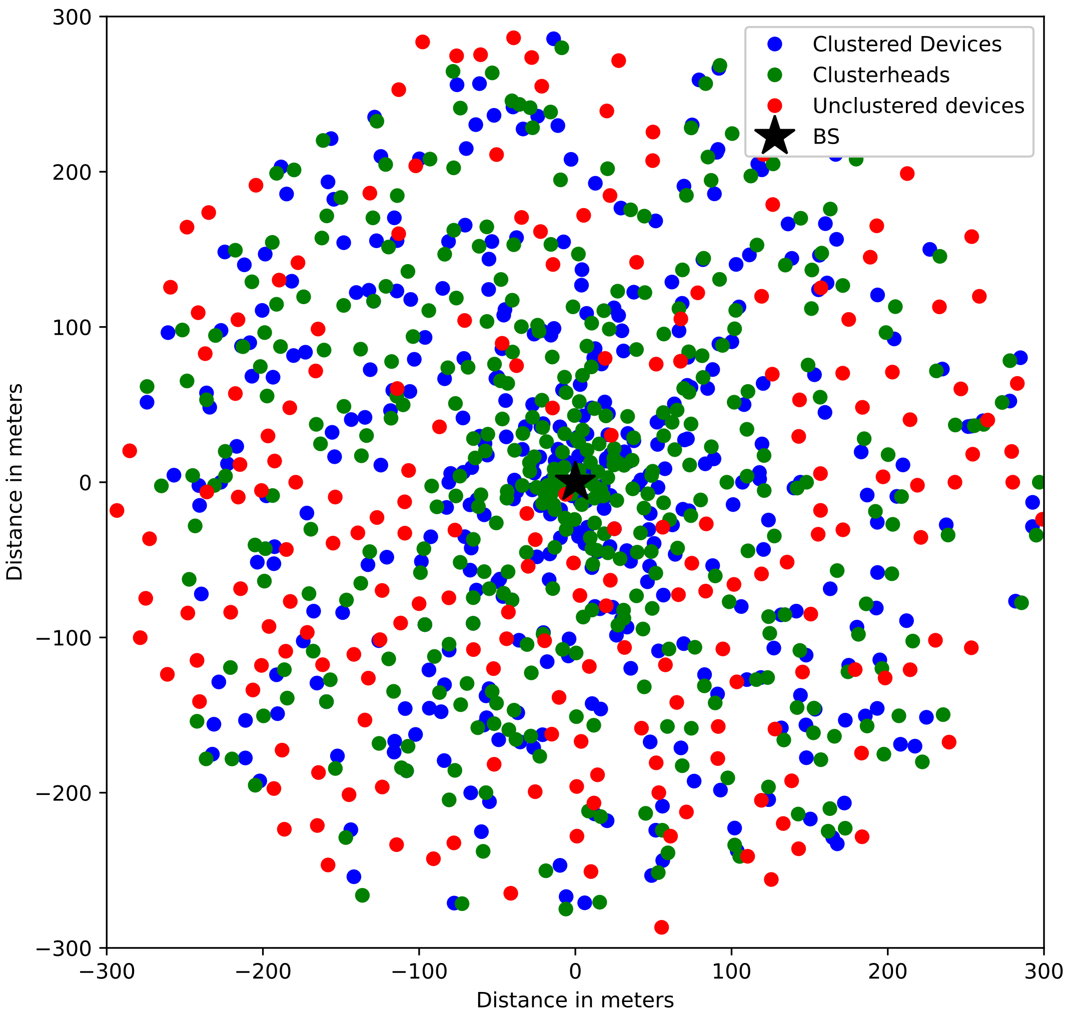
Figure 7 – Step 1: Parameters initialization.



Source: own authorship (2024)

2. **Creation of Clusters of Two Devices:** At this stage, only clusters of size 2 are formed, resulting in the creation of 400 clusters, as represented by Figure 8.

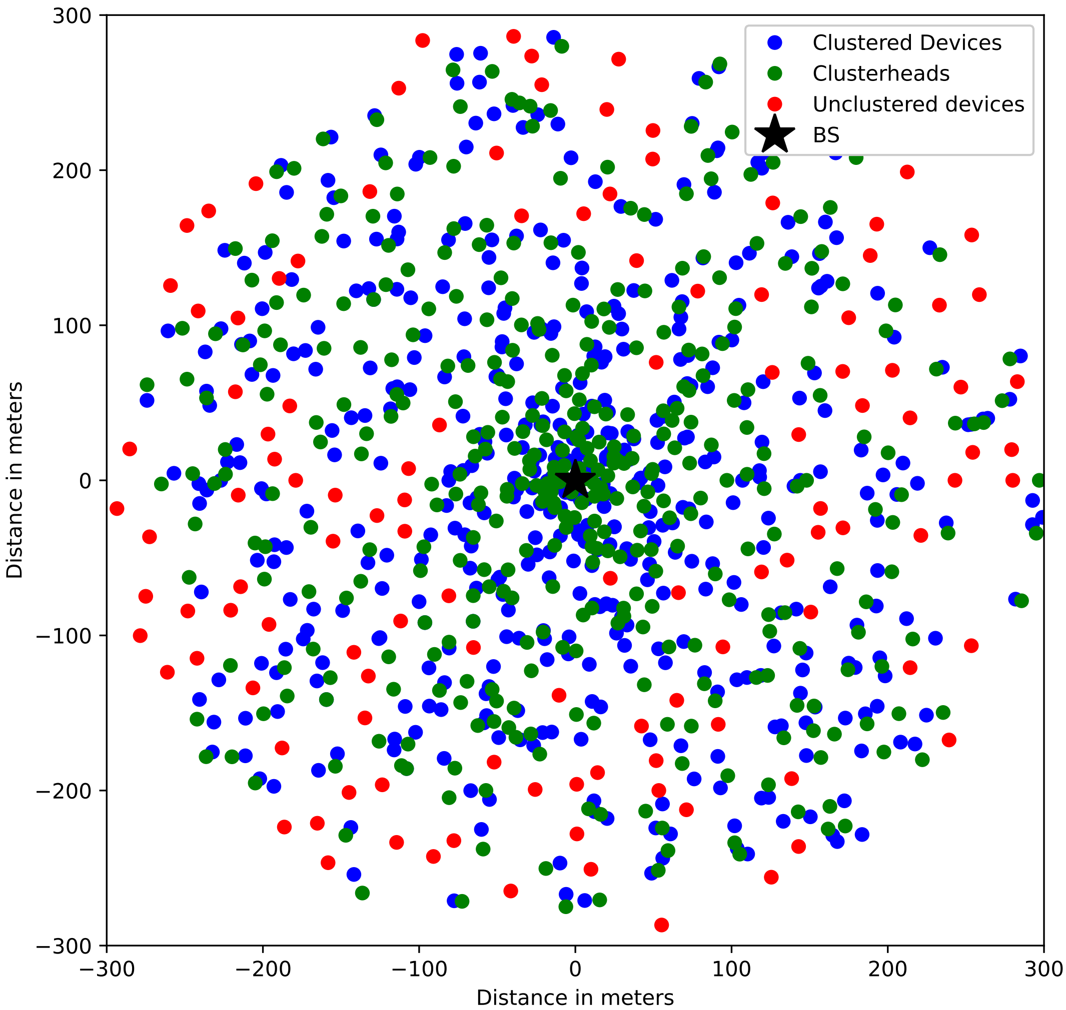
Figure 8 – Step 2: Creation of clusters of two devices.



Source: own authorship (2024)

3. **Absorbing Non-Clustered Devices:** Now, the number of clusters remains the same, and what changes is that some clusters grow in size. Thus, at this step, the proposed algorithm created 69 clusters of size 3, and 5 clusters of size 4, reducing the number of clusters with size 2, to 326. Figure 9 illustrates the network this step.

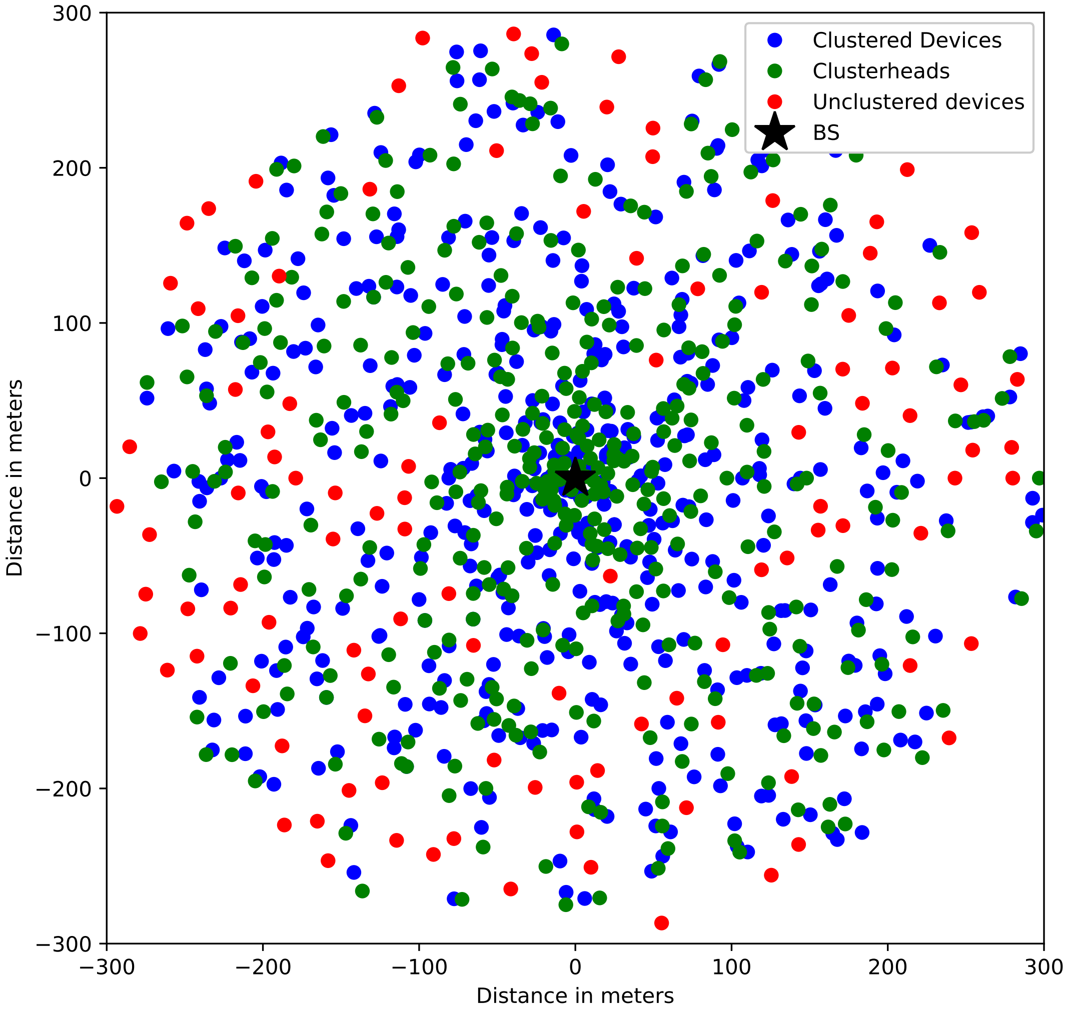
Figure 9 – Step 3: Absorbing non-clustered devices.



Source: own authorship (2024)

4. **Repositioning CHs:** Again, the number of clusters is not impacted by this step, and Figure 10 shows that by repositioning the CHs, only the clusters of size 2 and 3 are changed. With this step, the proposed algorithm created 318 clusters of size 2, and 77 clusters of size 3, while the 5 clusters of size 4 remained the same.

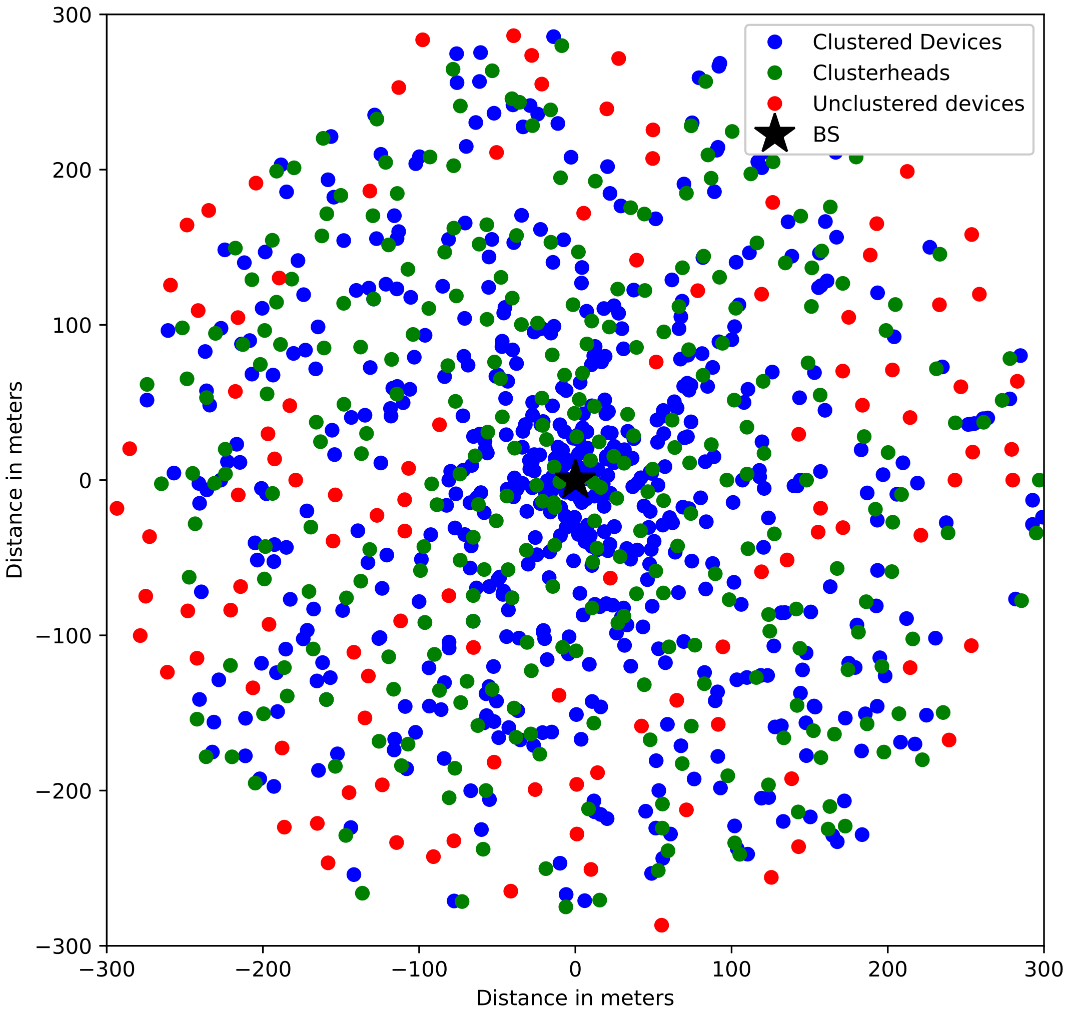
Figure 10 – Step 4: Repositioning CH.



Source: own authorship (2024)

5. **Merging Clusters:** This step is very impacted by $C_{\max}$, and Figure 11 illustrates this process. Now, the number of clusters suffers a downsize, but, in replace, some bigger clusters are created, resulting in 303 clusters. In a complementary way, Table 2 summarizes the number of clusters created, its behavior, and how many unclustered devices remained in each step. Thus, the impact of step 5 compared to step 4 can be observed in Table 2.

Figure 11 – Step 5: Merging clusters.



Source: own authorship (2024)

Table 2 – Summary of the number of clusters created step by step of the proposed algorithm considering $k = 1000$.

Quantity of clusters of size	Proposed algorithm step number					
	1	2	3	4	5	6
1	1000	200	121	113	113	113
2	0	400	326	318	181	163
3	0	0	69	77	57	59
4	0	0	5	5	31	57
5	0	0	0	0	13	10
6	0	0	0	0	5	4
7	0	0	0	0	6	4
8	0	0	0	0	3	2
9	0	0	0	0	1	2
10	0	0	0	0	6	2
Number of CHs (K_{ch})	0	400	400	400	303	303

Source: own authorship (2024)

6. **Balancing Clusters Size:** Once more, the number of CHs is not changed, but is also very impacted by $C_{\max}$. Big clusters suffer a downsize, while clusters around β see some growth. The figure that represents this step is virtually the same as Figure 11, considering that at this stage, there was only a repositioning of some cluster members to other clusters. The balanced clusters are described in Table 2 step 6.
7. **Auto Cluster Head:** In this last step, it is changed all the unclustered devices into clusters of size 1, thus becoming auto CHs. One important note here, is that for calculating the clusterization rate, it is used the previous step as the main source, considering that in this step it is not formed true clusters. This step only affects the number of clusters, resulting in 416 clusters, counting the previous 303 that already existed plus the 113 new clusters of size 1.

Table 3 complements the analysis by summarizing the number of clusters created with the proposed D2D-SRC algorithm, and the size of these clusters across simulations involving different numbers of devices. Notably, in the row indicating the clustering rate, it is possible to observe that the more devices are distributed around the BS (higher K), the better the metrics for the clustering algorithm, clustering up to 99.5% of the devices when $K = 3000$. This is because a larger density of devices makes them naturally closer to each other and, therefore, in the range of at least one device, thus lowering the number of non-clustered devices. As K increases, the quantity of larger clusters also grows, which is also impacted by this higher density of devices. Another important aspect to note here is that it is being considered a balancing parameter $\beta = 4$,

Table 3 – Summary of the number of clusters created by the proposed D2D-SRC algorithm, with $\beta = 4$.

Quantity of clusters of size	Total number of devices (K)			
	500	1000	2000	3000
1	132	113	57	15
2	75	163	160	142
3	39	59	121	107
4	14	57	147	272
5	6	10	20	32
6	1	4	23	40
7	0	4	11	24
8	0	2	13	25
9	1	2	7	16
10	0	2	19	38
Number of CHs (K_{ch})	268	416	578	711
Clustering Rate	73.6%	88.7%	97.2%	99.5%

Source: own authorship (2024)

so that it is possible to observe that this impacts the creation of more clusters of such size, while reducing the quantity of larger clusters. Finally, it is important to highlight that, the higher the number of clustered devices, fewer collisions are expected in the communication between the CHs and the BS, improving the HFL performance.

As a comparison, Table 4 shows the number of clusters created with the DBSCAN algorithm in the same scenario. One important aspect of DBSCAN is its tendency to create a single large cluster with most of the devices. This becomes clear when $K = 3000$ devices, where a single cluster contains 2794 devices, or 93.13% of the total, thus creating a very unbalanced clusterization. Such an aspect is very undesirable in practical applications, as it considerably increases the burden of the CH while reducing the reliability of the D2D communication. Another interesting aspect is that the proposed D2D-SRC algorithm and DBSCAN achieve very similar clustering rates, but the novel method creates much more balanced clusters.

Finally, Figure 12(a) shows the distribution of the clustering rate across 1000 FL runs with $K = 500$ devices. The curve presents a Gaussian shape, reflecting the uniform distribution of the devices around the BS. Figure 12(b) complements this analysis in a network with $K = 3000$ devices, by illustrating that not only does the clustering rate grow with more devices spread around the BS, but also the behavior becomes more uniform as the standard deviation decreases from 2.05 in Figure 12(a) to 0.15 in Figure 12(b).

Table 4 – Summary of the number of clusters created by the DBSCAN algorithm.

Quantity of clusters of size	Total number of devices (K)			
	500	1000	2000	3000
1	108	107	40	23
2	56	41	26	13
3	21	24	9	9
4	6	8	12	3
5	6	12	3	0
6	3	2	3	1
7	3	2	2	2
8	1	2	5	0
9	2	5	5	3
10	0	4	2	1
11	0	0	2	0
12	0	2	1	0
13	1	1	1	1
15	0	0	1	0
16	0	0	1	0
17	0	0	1	0
19	0	1	0	1
25	0	1	0	0
26	0	0	1	0
29	0	0	0	1
39	0	1	0	0
47	0	0	1	0
51	0	0	1	0
85	1	0	0	0
400	0	1	0	0
1462	0	0	1	0
2794	0	0	0	1
Number of CHs (K_{ch})	208	214	118	59
Clustering Rate	78.4%	89.3%	98%	99.23%

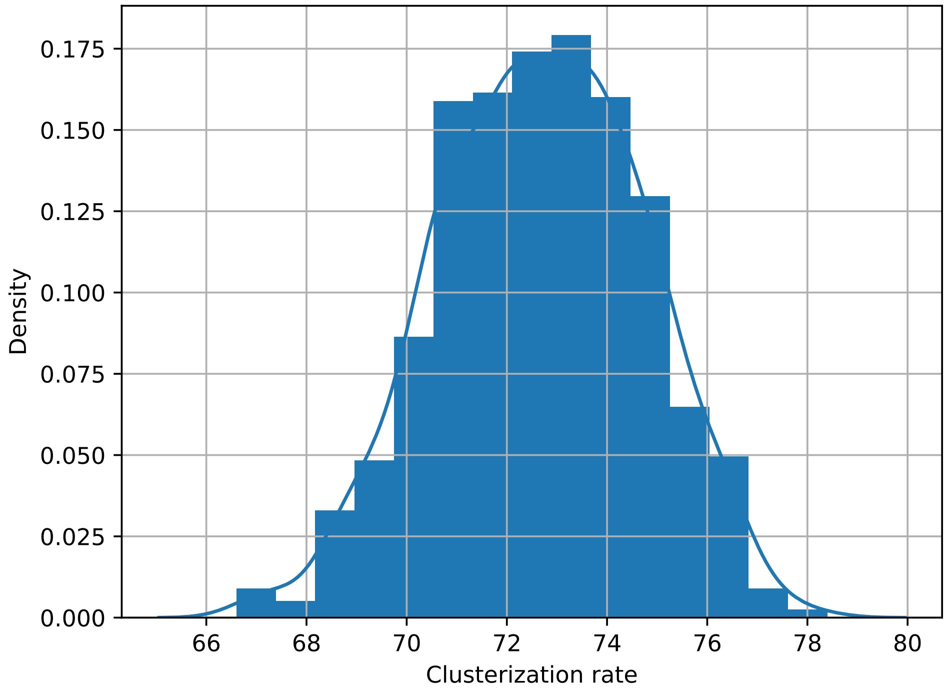
Source: own authorship (2024)

4.2 Proposed D2D-SRC Applied to HFL

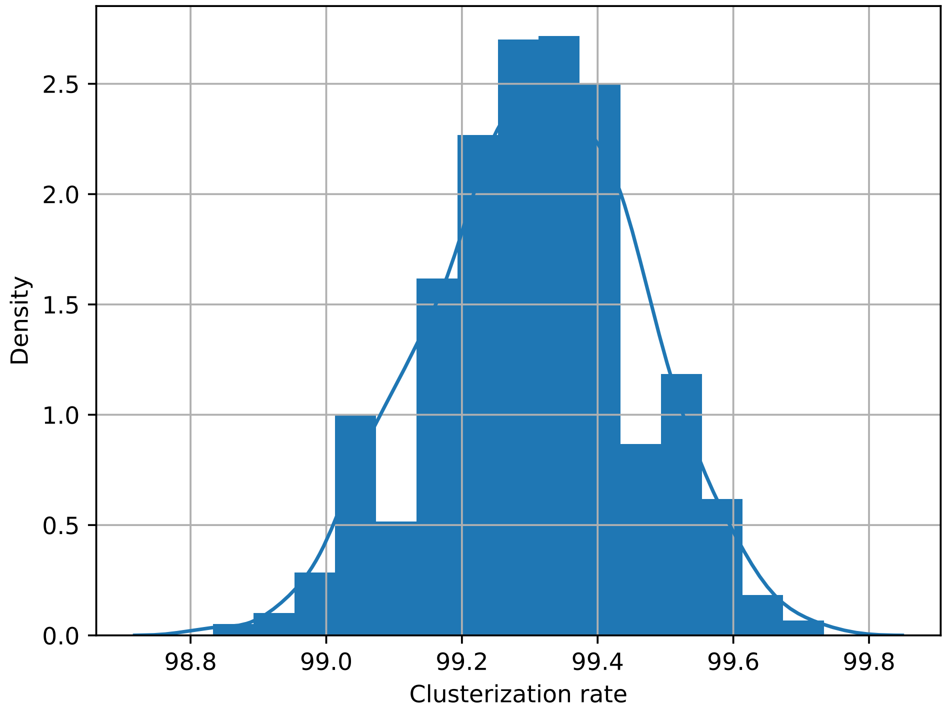
Here, the proposed D2D-SRC algorithm is applied to HFL. In addition, to select (or sample) the M CHs that upload their information at each iteration t it is employed the three strategies presented in Section 3.3, namely: polling, ALOHA with fixed access probability, and ALOHA with optimized access probability, following (Choi; Pokhrel, 2019).

It is considered that the number of devices in the network ranges from $K = 500$ to $K = 3000$. Figure 13 shows how many times the devices participated during the FL round after $t = 200$ iterations, for each of the three device selection models. One can note that the number of participating devices increases with the use of the proposed D2D-

Figure 12 – Distribution curves of the clustering rate for $K \in \{500, 3000\}$ devices randomly deployed around the BS.



((a)) $K = 500$**.**

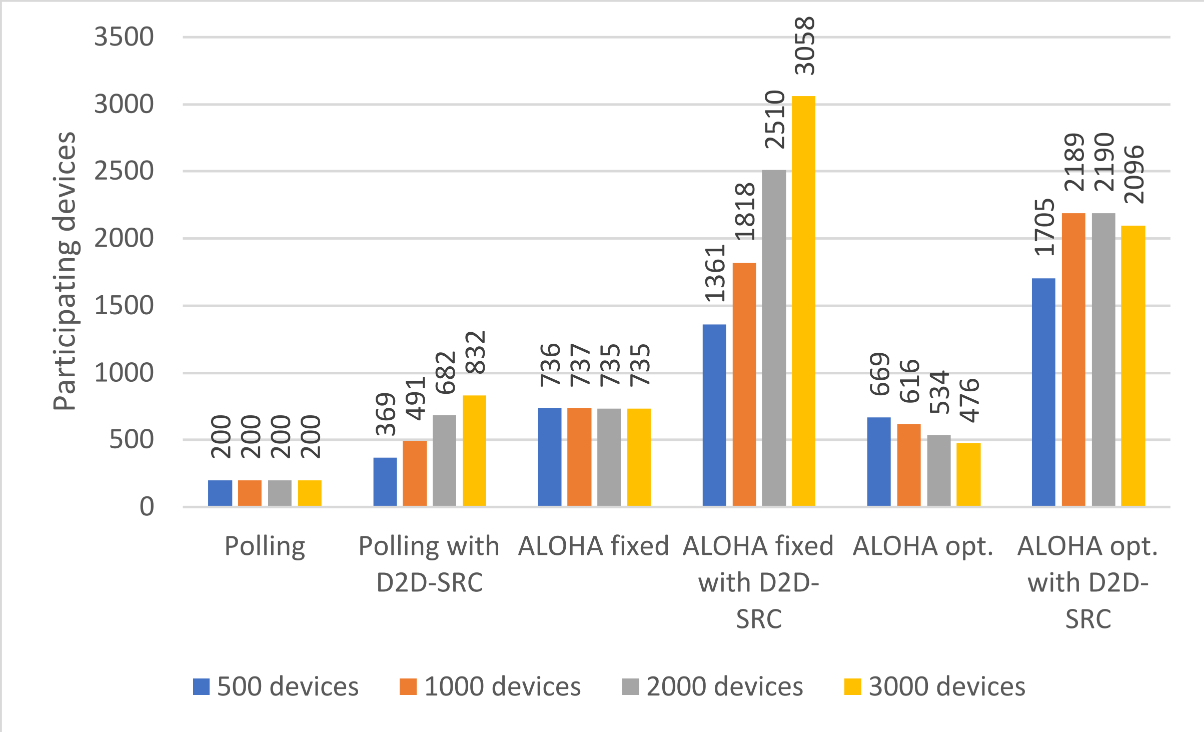


((b)) $K = 3000$**.**

Source: own authorship (2024)

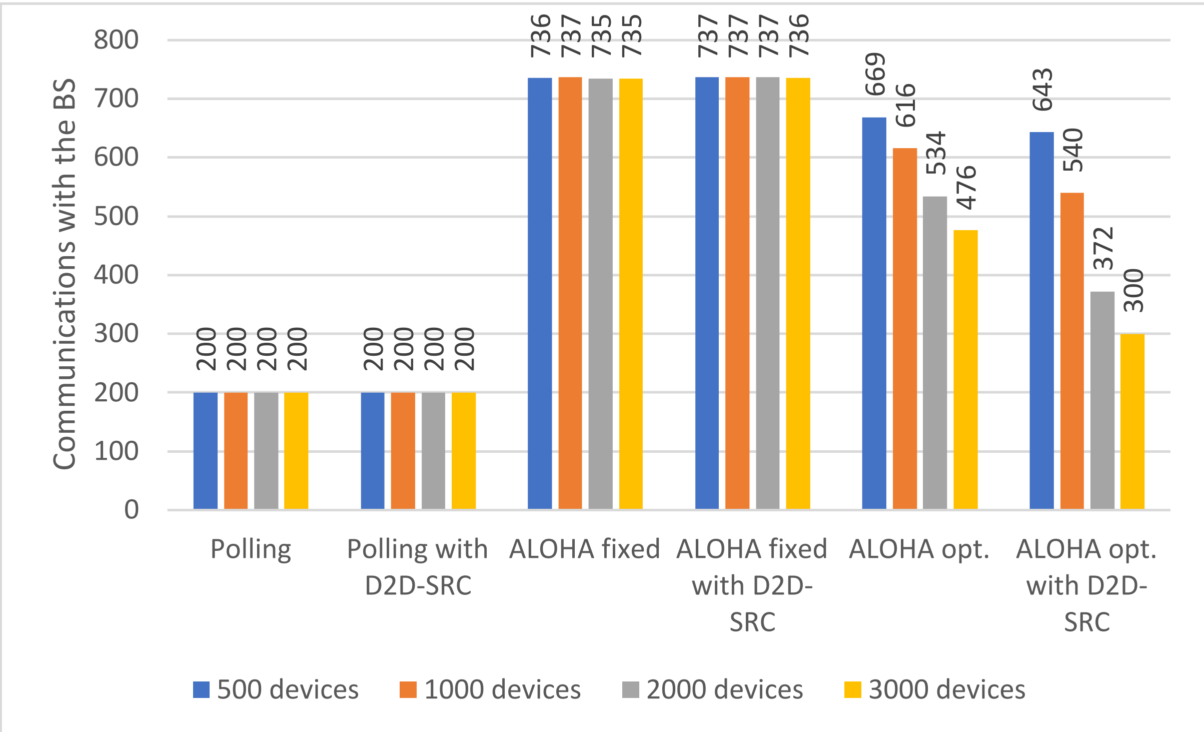
SRC algorithm, irrespective of the device selection method. It is important to note two considerations here. First, the values do not represent the number of communications

Figure 13 – Total number of successful participating devices after $t = 200$ iterations for each of the models.



Source: own authorship (2024)

Figure 14 – Total number of communications perceived from the BS perspective after $t = 200$ iterations for each of the models.



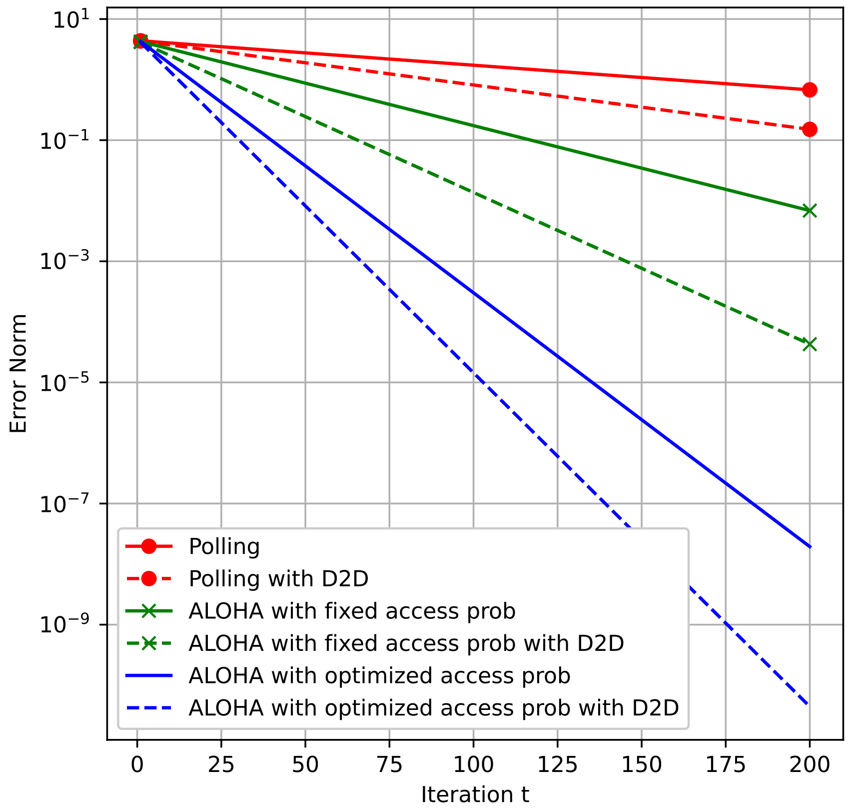
Source: own authorship (2024)

direct to the BS, as they only represent how many devices participated with their data through the clusters, since the CHs perform an aggregation of the data of their cluster members before uploading them to the BS. The second aspect is that devices have multiple opportunities (or iterations) to participate in a FL round, which justifies why we can have more participating devices in Figure 13 than the total number of devices around the BS.

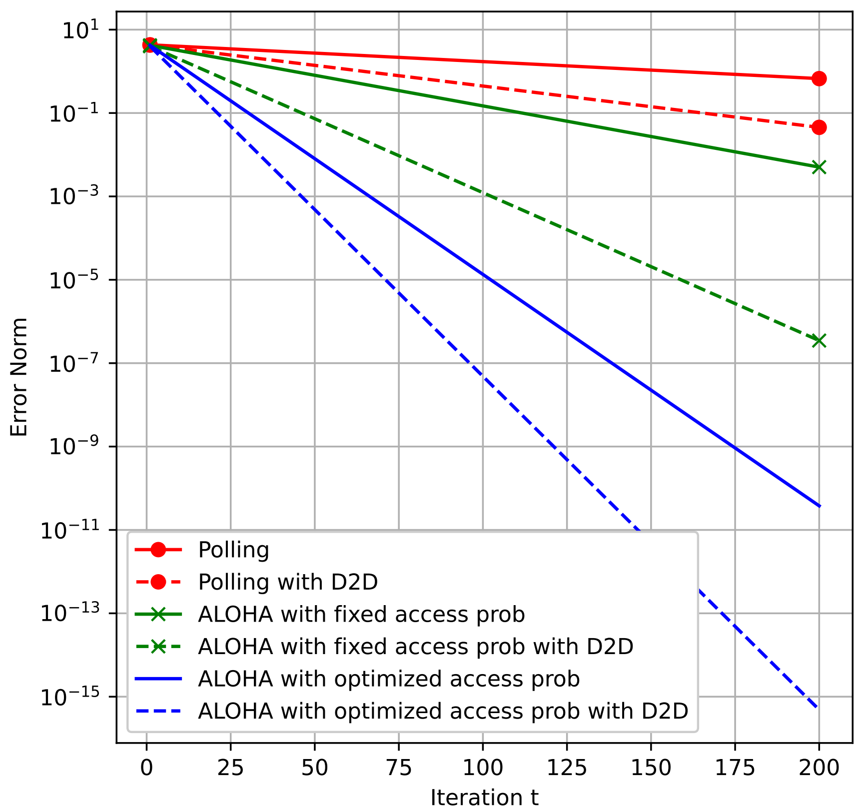
With respect to the number of devices that indeed communicated with the BS, the simulation results are summarized by Figure 14, and when cross-referencing Figure 13 and Figure 14 it is observed that the number of communications of the Polling and ALOHA with fixed access probability models remained constant, but the total number of devices that participated directly or indirectly (through the CH) increased, as the number of devices using the proposed D2D-SRC algorithm also increased. From the BS perspective, one can perceive an even more advantageous scenario in the ALOHA with optimized access probability model. As the number of devices increases, fewer communications are performed directly to the BS, but with these communications carrying more impactful information.

The gain in the quantity and quality of the information sent to the BS by using the D2D-SRC algorithm is illustrated in Section 4.2, where the error norm considers the difference between the true output weights and the weights predicted by the FL model. For instance, Figure 15(d) explicitly shows that the polling model with the D2D-SRC algorithm can perform better than ALOHA with fixed access probability without the D2D-SRC. Similarly, ALOHA with fixed access probability with the D2D-SRC algorithm can perform as well as ALOHA with optimized access probability without clustering. The trend suggests that with a further increase in the number of devices, the final error achieved could be even lower, indicating the potential for further improvement. In practical deployments, it is valid to note that clustering has more relevance than optimizing the access probability, considering that each time the topology changes, this same access probability would need to be updated, impacting the system's performance as the IoT environments are usually dynamic and unpredictable. Finally, it is also possible to observe that the error norm decreases when the number of devices K increases, in accordance with the devices' density and the growth in the cluster size, as previously noted in Section 4.1.

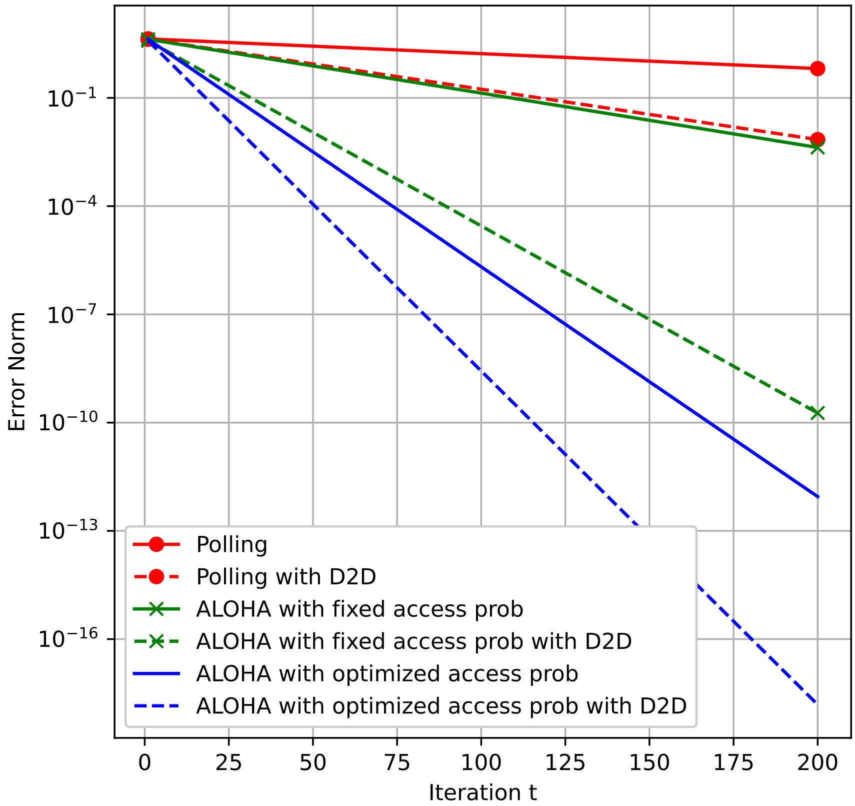
Figure 15 – Error norm of the HFL task, for the three CH selection models with and without applying the proposed D2D-SRC algorithm.



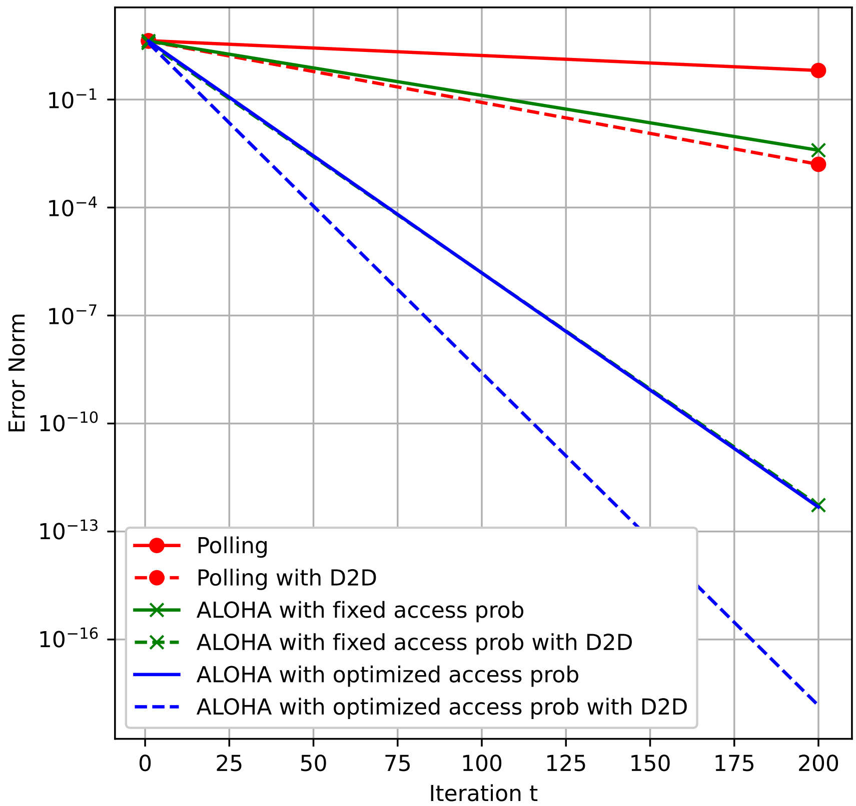
((a)) $K = 500$ **devices.**



((b)) $K = 1000$ **devices.**



((c)) $K = 2000$ **devices.**



((d)) $K = 3000$ **devices.**

Source: own authorship (2024)

5 CONCLUSIONS

This thesis has proposed a D2D clustering algorithm, named D2D-SRC, tailored for optimizing an HFL system utilizing multichannel ALOHA. The proposed method is different from classical clusterization algorithms such as the DBSCAN algorithm, as it diverges by focusing on the unique communication dynamics of IoT networks. By creating small sets of clusters, it achieved a positive impact on the HFL performance, while maintaining or reducing the number of communications with the BS, therefore optimizing the finite communication resources.

Chapter 2 reviews some of the preliminary concepts used in this thesis, with some important theoretical concepts focusing on ML and deepening the studies of ML to FL and clustering. It is also addressed the ALOHA protocol and the motivation for using it in combination with the FL scheme. Then, Chapter 3 details the system model, where the scenario consists of a BS serving a set of IoT devices engaged in an FL task, and the devices communicate with it through a multichannel ALOHA protocol. The implemented HFL is made of two layers, where on the first layer the devices update their local information based on the global model, and then the CHs aggregate these updates locally, so that in the second layer, the BS can get the compiled clusters information directly from the CHs, thus lowering the number of communications made between the devices and the BS. Three strategies of CHs selection were considered, named as Polling, ALOHA with Fixed Access Probability and ALOHA with Optimized Access Probability, where this last one considers the norm of the local updates of the CHs to adapt its access probability.

The results were presented in a way to analyze the effects of the proposed D2D-SRC algorithm within the described system model, and, in order to do so, the three CH selection models were compared with and without this clustering scheme. The metrics revealed that the proposed D2D-SRC algorithm increased the number of participating devices in each round of the FL, reducing the error norm at the BS and enhancing all three CHs selection strategies. Also, the proposed algorithm in comparison with DBSCAN, optimized the resources' allocation, as it was capable of achieving a higher clustering rate while considering some restrictions of a real wireless communications systems, such as the communication distance. Besides, the clustering approach significantly improved the performance of HFL over ALOHA and polling.

In future works, it is intended to take the battery level of the devices into account in order to optimize the energy efficiency of the network. Such an approach would impact the CH selection algorithms, yielding different optimizations. Furthermore, the proposed clusterization algorithm could also be adapted towards energy efficiency inside the clusters. In addition, the scheduling of the HFL approach can be optimized by considering if the device had already participated with its information before, as well as analyzing the freshness of the information provided by each device, which may be a relevant metric in order to improve the global FL model at the BS. Finally, it may also be implemented the soft collision state, which is making the device capable of retrying to send its local updates in cases where a collision happened before.

REFERENCES

ANALYTICS, IoT. **State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally**. 2023. Available from: https://iot-analytics.com/number-connected-iot-devices/. (accessed: 04.09.2024).

BOLOURSAZ MASHHADI, Mahdi; GÜNDÜZ, Deniz. Deep learning for massive MIMO channel state acquisition and feedback. **Journal of the Indian Institute of Science**, Springer, v. 100, n. 2, p. 369–382, 2020.

BOYD, Stephen *et al.* Distributed optimization and statistical learning via the alternating direction method of multipliers. **Foundations and Trends® in Machine learning**, Now Publishers, Inc., v. 3, n. 1, p. 1–122, 2011.

BUDUMA, Nikhil. Fundamentals of Deep Learning: Designing Next-generation Artificial Intelligence Algorithms/c Nikhil Buduma. **Beijing, Boston, Farnham, Sebastopol, Tokyo: OReilly**, 2017.

CHOI, Jinho. Multichannel ALOHA with exploration phase. *In*: IEEE Wireless Communications and Networking Conference (WCNC). [*S. l.: s. n.*], 2020. P. 1–6.

CHOI, Jinho; POKHREL, Shiva Raj. Federated learning with multichannel ALOHA. **IEEE Wireless Communications Letters**, IEEE, v. 9, n. 4, p. 499–502, 2019.

DA SILVA, Rafael Valente; LÓPEZ, Onel L Alcaraz; SOUZA, Richard Demo. Energy-Aware Federated Learning with Distributed User Sampling and Multichannel ALOHA. **IEEE Communications Letters**, IEEE, 2023.

DESHPANDE, Vaibhav V; PATIL, Arvind R Bhagat. Energy efficient clustering in wireless sensor network using cluster of cluster heads. *In*: IEEE International Conference on Wireless and Optical Communications Networks (WOCN). [*S. l.: s. n.*], 2013. P. 1–5.

DUAN, Lian; DA XU, Li. Data analytics in Industry 4.0: A survey. **Information Systems Frontiers**, Springer, p. 1–17, 2021.

ELDAR, Yonina C *et al.* **Machine learning and wireless communications**. [*S. l.*]: Cambridge University Press, 2022.

ESTER, Martin *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. *In*: 34. 2ND International Conference on Knowledge Discovery and Data Mining (KDD-96). [*S. l.: s. n.*], 1996. v. 96, p. 226–231.

GERLA, Mario; LAZAR, Aurel; KUEHN, Paul. **Telecommunication Networks and Computer Systems**. [*S. l.*]: Springer, 1990.

GHOSH, Avishek *et al.* An efficient framework for clustered federated learning. **Advances in Neural Information Processing Systems**, v. 33, p. 19586–19597, 2020.

GÜNDÜZ, Deniz *et al.* Communicate to learn at the edge. **IEEE Communications Magazine**, IEEE, v. 58, n. 12, p. 14–19, 2020.

HELLSTRÖM, Henrik *et al.* Wireless for Machine Learning: A Survey. **Foundations and Trends in Signal Processing**, v. 15, n. 4, p. 290–399, 2022. ISSN 1932-8346. DOI: `10.1561/2000000114`.

HUANG, Xu *et al.* A Full Duplex D2D Clustering Resource Allocation Scheme Based on a-Means Algorithm. **Hindawi Wireless Communications and Mobile Computing**, v. 2018, 2018.

JUBAIR, Ahmed Mahdi *et al.* Optimization of clustering in wireless sensor networks: techniques and protocols. **Applied Sciences**, MDPI, v. 11, n. 23, p. 11448, 2021.

KHELF, Roumaissa; DRIOUCH, Elmahdi; AJIB, Wessam. On the Optimization of UAV-Assisted Wireless Networks for Hierarchical Federated Learning. *In*: IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). [*S. l.: s. n.*], 2023. P. 1–6.

KONECNÝ, Jakub *et al.* Federated Optimization: Distributed Machine Learning for On-Device Intelligence. **ArXiv**, abs/1610.02527, 2016. Available from: `https://arxiv.org/abs/1610.02527v1`.

LE, Khanh Tuan. Bluetooth Low Energy and the automotive transformation. **Texas Instruments, September**, 2017.

LEE, Yang-Han *et al.* Using genetic algorithm with frequency hopping in device to device communication (D2DC) interference mitigation. *In*: IEEE International Symposium on Intelligent Signal Processing and Communications Systems. [*S. l.: s. n.*], 2012. P. 201–206.

LUDWIG, Heiko; BARACALDO, Nathalie. **Federated learning: A comprehensive overview of methods and applications**. [*S. l.*]: Springer, 2022.

MADHAVAN, Sowmya *et al.* Improving LoRa Low-Power Wide-Area Networks with Double Deep-Q Learning-Based Resource Allocation. *In*: IEEE. 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). [*S. l.: s. n.*], 2024. P. 1–5.

MASHHADI, Mahdi Boloursaz; GÜNDÜZ, Deniz. Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems. **IEEE Transactions on Wireless Communications**, IEEE, v. 20, n. 10, p. 6315–6328, 2021.

MASHHADI, Mahdi Boloursaz; YANG, Qianqian; GÜNDÜZ, Deniz. Distributed deep convolutional compression for massive MIMO CSI feedback. **IEEE Transactions on Wireless Communications**, IEEE, v. 20, n. 4, p. 2621–2633, 2020.

MIURA, Kentaro. [*S. l.*], 1998.

RANA, Omer *et al.* Hierarchical and decentralised federated learning. *In*: IEEE Cloud Continuum. [*S. l.: s. n.*], 2022. P. 1–9.

ROBERTS, Lawrence G. ALOHA packet system with and without slots and capture. **ACM SIGCOMM Computer Communication Review**, ACM New York, NY, USA, v. 5, n. 2, p. 28–42, 1975.

RUSSELL, Stuart J; NORVIG, Peter. **Artificial intelligence: a modern approach**. [*S. l.*]: Pearson, 2020.

SATTLER, Felix; MÜLLER, Klaus-Robert; SAMEK, Wojciech. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. **IEEE transactions on neural networks and learning systems**, IEEE, v. 32, n. 8, p. 3710–3722, 2020.

SILVA, Rafael Valente da *et al.* Multichannel ALOHA optimization for federated learning with multiple models. **IEEE Wireless Communications Letters**, IEEE, v. 11, n. 10, p. 2180–2184, 2022.

SINGH, Jitendra; KUMAR, Rakesh; MISHRA, Ajai Kumar. Clustering algorithms for wireless sensor networks: A review. *In*: IEEE. 2ND International Conference on Computing for Sustainable Global Development (INDIACom). [*S. l.: s. n.*], 2015. P. 637–642.

SONG, Lihua *et al.* Adaptive Edge-Level Personalization on Hierarchical Federated Learning. *In*: IEEE International Performance, Computing, and Communications Conference (IPCCC). [*S. l.: s. n.*], 2023. P. 397–402.

SONI, Suryakant; DEY, Biswanath. Dynamic selection of cluster head in cluster of cluster heads within the cluster in Heterogeneous Wireless Sensor Network. *In*: IEEE International Conference on Advanced Communications, Control and Computing Technologies. [*S. l.: s. n.*], 2014. P. 877–881.

STATISTA. **Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033**. 2024. Available from: `https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/`. (accessed: 07.15.2024).

VASYUTYNSKYY, Volodymyr; LUNTOVSKYY, Andriy. Data Management in IoT: From Sensor Data to Intelligent Applications. *In*: IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering. [*S. l.: s. n.*], 2022. P. 565–573.

WEN, Chao-Kai; SHIH, Wan-Ting; JIN, Shi. Deep learning for massive MIMO CSI feedback. **IEEE Wireless Communications Letters**, IEEE, v. 7, n. 5, p. 748–751, 2018.

WU, Jiajun *et al.* Topology-aware Federated Learning in Edge Computing: A Comprehensive Survey. **ACM Computing Surveys**, v. 56, n. 10, 2024. ISSN 0360-0300. DOI: 10.1145/3659205.

XING, Hong; SIMEONE, Osvaldo; BI, Suzhi. Federated learning over wireless device-to-device networks: Algorithms and convergence analysis. **IEEE Journal on Selected Areas in Communications**, IEEE, v. 39, n. 12, p. 3723–3741, 2021.

XU, Weihong *et al.* Joint neural network equalizer and decoder. *In*: IEEE. 2018 15th International Symposium on Wireless Communication Systems (ISWCS). [*S. l.: s. n.*], 2018. P. 1–5.

YANG, Kai *et al.* Federated learning via over-the-air computation. **IEEE Transactions on Wireless Communications**, IEEE, v. 19, n. 3, p. 2022–2035, 2020.

YE, Hao; LI, Geoffrey Ye; JUANG, Biing-Hwang. Power of deep learning for channel estimation and signal detection in OFDM systems. **IEEE Wireless Communications Letters**, IEEE, v. 7, n. 1, p. 114–117, 2017.

YE, Mao *et al.* EECS: an energy efficient clustering scheme in wireless sensor networks. *In*: 24TH IEEE International Performance, Computing, and Communications Conference (PCCC). [*S. l.: s. n.*], 2005. P. 535–540.

YUWEN, Wang; YU, Gao; XIANGJUN, Lin. Differential Privacy Hierarchical Federated Learning Method based on Privacy Budget Allocation. *In*: IEEE 9th International Conference on Computer and Communications (ICCC). [*S. l.: s. n.*], 2023. P. 2177–2181.

ZHU, Jingyang *et al.* Over-the-Air Federated Learning and Optimization. **IEEE Internet of Things Journal**, v. 11, n. 10, p. 16996–17020, 2024.