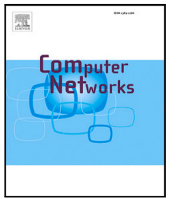
Computer Networks 253 (2024) 110681

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Impact of network topology on the performance of Decentralized Federated Learning

Luigi Palmieri *, Chiara Boldrini[1], Lorenzo Valerio[1], Andrea Passarella, Marco Conti

CNR-IIT, Via G. Moruzzi, 1, Pisa, Italy

ARTICLE INFO

Keywords:
Decentralized learning
Network topologies
Data distribution

ABSTRACT

Fully decentralized learning is gaining momentum for training AI models at the Internet's edge, addressing infrastructure challenges and privacy concerns. In a decentralized machine learning system, data is distributed across multiple nodes, with each node training a local model based on its respective dataset. The local models are then shared and combined to form a global model capable of making accurate predictions on new data. Our exploration focuses on how different types of network structures influence the spreading of knowledge – the process by which nodes incorporate insights gained from learning patterns in data available on other nodes across the network. Specifically, this study investigates the intricate interplay between network structure and learning performance using three network topologies and six data distribution methods. These methods consider different vertex properties, including degree centrality, betweenness centrality, and clustering coefficient, along with whether nodes exhibit high or low values of these metrics. Our findings underscore the significance of global centrality metrics (degree, betweenness) in correlating with learning performance, while local clustering proves less predictive. We highlight the challenges in transferring knowledge from peripheral to central nodes, attributed to a dilution effect during model aggregation. Additionally, we observe that central nodes exert a pull effect, facilitating the spread of knowledge. In examining degree distribution, hubs in Barabási–Albert networks positively impact learning for central nodes but exacerbate dilution when knowledge originates from peripheral nodes. Finally, we demonstrate the formidable challenge of knowledge circulation outside of segregated communities, and discuss the impact of class cross-correlations.

1. Introduction

The modern technology ecosystem is experiencing a dramatic shift characterized by the exponential growth of devices at the Edge of the network, together with the enormous expansion of data that they create. The most popular network paradigm for AI is currently a centralized one, that involves transferring data collected at the edge towards big data centers, where data are processed and large-scale AI models are trained. These trained models are then deployed to furnish AI-based services. Despite proving effective, this centralized approach raises several concerns related to data privacy and ownership. Even with the prospect of benefiting from centralized AI-based services, data generators at the network's edge are becoming less eager to disclose their private data to third parties. These worries are fueling a paradigm shift from centralized AI systems to decentralized ones.

To address these issues, solutions based on decentralized AI systems, such as Federated Learning (FL) [1], have been proposed. In FL, the idea is to avoid the transfer of raw data, keeping the data locally at the edge devices. Specifically, the devices collaboratively train an AI model without sharing any raw data with each other. They only share the parameters of the AI models that are trained locally. These parameters are then aggregated to produce a better and capable *global* model that is iteratively refined through successive rounds of collaboration.

The standard definition of the FL framework assumes a starred network topology, where a central server orchestrates the entire process, coordinating the operations of the participating devices. However, the presence of a centralized controller introduces some challenges. It could potentially serve as a single point of failure, a potential bottleneck when the number of involved devices reaches millions, and an impediment to spontaneous, direct collaborations among users. In response to these challenges, a rising trend is emerging in favor of supporting fully decentralized variations of FL. Decentralized Federated Learning (DFL) represents an alternative to centralized Federated Learning. In DFL the

* Corresponding author.
E-mail addresses: luigi.palmieri@iit.cnr.it (L. Palmieri), chiara.boldrini@iit.cnr.it (C. Boldrini), lorenzo.valerio@iit.cnr.it (L. Valerio), andrea.passarella@iit.cnr.it (A. Passarella), marco.conti@iit.cnr.it (M. Conti).

[1] C. Boldrini and L. Valerio contributed equally to this work.

https://doi.org/10.1016/j.comnet.2024.110681
Received 1 February 2024; Received in revised form 17 May 2024; Accepted 28 July 2024
Available online 5 August 2024
1389-1286/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

connectivity between devices is represented by a generic graph, and the devices involved in the learning process typically collaborate only with their neighbors. The lack of a central controller overcomes the single point of failure problem but introduces other aspects that need to be investigated. Specifically, we claim that, in this scenario, the information locality and the network topology strongly affect the dynamics of the learning process, i.e. how fast and effective the spreading of knowledge about the class labels is.

While previous work [2] assumes that the network topology can be controlled by the network operator and optimized to make the learning process more efficient or scalable, we argue that complete decentralization can only be achieved by letting user devices spontaneously organize themselves. This implies that the network topology, in these settings, cannot be controlled by the operator. For example, an edge between two nodes in the graph may represent a trust relationship or willingness to cooperate. If the edges are weighted, the strength of the weights expresses the intensity of trust or cooperation. With this approach, users are free to cooperate with whomever they want, and the operator has no control over the cooperation patterns. Although this scenario poses a challenge from a learning perspective, it also fully exploits the human-centric, impromptu potential of fully decentralized learning systems.

Given these considerations, the primary research question addressed in this paper concerns the impact of network topology on the learning process within a fully decentralized learning system. Specifically, we examine a scenario where a group of devices collaborates to train a unified AI model in a completely decentralized environment, connected to each other through a complex network topology. Following the DFL framework, each participating device receives a set of models from its graph-connected neighbors. At first, these models undergo an initial aggregation step with the local model, usually through a weighted average, resulting in a refreshed aggregate model. Subsequently, this aggregated model is updated via a number of local training epochs performed on local data. The newly updated models are then shared with neighboring devices, and this iterative process continues until a stopping condition is met.

In this paper, we examine three network topologies: Erdős-Rényi, Barabási-Albert, and Stochastic Block Model. Through simulations, we analyze how their underlying network structures impact the learning process, considering various non-IID data partitioning scenarios, three different datasets and two different neural network architectures. Specifically, we assume that a subset of nodes possesses more knowledge than others, represented by classes unavailable on other nodes. We identify this subset of nodes based on the highest or lowest values of properties such as degree centrality, betweenness centrality, and clustering coefficient. These properties capture the significance of nodes in the network and the connectivity within their local neighborhoods. The main take-home messages from our analysis are the following.

- The initial data distribution on high vs low degree nodes plays a key role in the final accuracy of a decentralized learning process.
- When high-degree nodes possess more knowledge, such knowledge spreads easily in the network.
- Vice versa, when low-degree nodes have more knowledge, knowledge spreads better when the network is less connected (at first counterintuitive, but connectivity dilutes knowledge in average-based decentralized learning).
- When users are grouped in segregated communities, it is very difficult for knowledge to circulate outside of the community.
- When considering different centrality measures, the system's performance exhibits a direct correlation with global centrality metrics. In contrast, centrality measures centered on local structure are not robust indicators of system performance.
- Unbalanced data split in decentralized learning scenarios may amplify cross-correlation effects among classes. In such cases, having more data samples may not necessarily be better than having fewer data samples for the cross-correlated classes. While these effects are not significantly dependent on the network topology, they can radically impact the performance of decentralized learning.

The remainder of this paper is organized as follows. We present a literature review in Section 2. Then, we provide the basics of Decentralized Federated Learning in Section 3. In Section 4, we discuss the network topologies that we have chosen for our analysis. The detailed settings of our experiments are discussed in Section 5. Then, we discuss our findings for Erdős-Rényi (Section 6.1), Barabási-Albert (Section 6.2), as well as a comparison between the two (Section 6.3). The case of segregated communities with SBM is presented in Section 6.4. The effect of class cross-correlation is discussed in Section 6.5. Finally, Section 7 concludes the paper.

2. Related work

DFL extends the typical settings of FL, e.g., data heterogeneity and non-convex optimization, by removing the existence of the central parameter server. This is a relatively new topic gaining attention from the community since it fuses the privacy-related advantages of FL with the potentialities of decentralized and uncoordinated optimization and learning. In [3], the authors define a DFL framework for a medical application where a number of hospitals collaborate to train a neural network model on local and private data. They make use of a decentralized Federated Learning setting where multiple medical centers can collaborate and benefit from each other without sharing data among them. In [4] the authors propose a Bayesian-like approach where the aggregation phase is done by minimizing the Kullback–Leibler divergence between the local model and the ones received from the peers. All these approaches are still considering that the nodes perform just one local update before sharing the parameters (or gradients) with the peers in the network. This aspect is relaxed in [2,5]. In [5], the authors propose a federated consensus algorithm extending FedAvg from [1] in decentralized settings, mainly considering industrial and IoT applications. The authors propose a consensus-based serverless Federated Learning approach to tackle the issues that stem from massive IoT networks. The proposed FL algorithms leverage the cooperation of devices that perform data operations inside the network by iterating local computations and mutual interactions via consensus-based methods. The proposed methodology is validated on an IIoT scenario typical of a 5G network infrastructure. A slightly different approach has been proposed by authors in [6]. They propose a variation in FL where the central server role is rotated among the participants, they refer to it as *flying master*, which is dynamically selected. They demonstrate a significant reduction of runtime using their proposed flying master FL framework compared to the original FL over real 5G networks. The work in [2] proposes a Federated Decentralized Average based on SGD in their research, where they incorporate a momentum term to counterbalance potential drift caused by multiple updates, along with a quantization scheme to reduce communication requirements. Finally, [7] addresses the heterogeneity and initialization problems of fully decentralized federated learning by proposing a novel learning strategy based on distance-weighted aggregation of models and knowledge distillation.

The approaches described so far assume that data remain within their hosting nodes, and models received from one-hop neighbors are aggregated with the local one. However, a complementary perspective was proposed a decade ago in a different domain, that of P2P networks [8]. P2P networks implement an overlay network where virtual links determine communication paths. The overlay topology is typically controlled by the algorithm used to establish it, while nodes can appear or disappear based on their usage patterns. In [8], an overlay network of this type is considered, and the authors propose a decentralized learning strategy based on the concept of multiple models taking

random walks over the network, and being trained using an online learning algorithm on each local dataset of the nodes they traverse. This framework leverages the gossip protocol for the model distribution, which traditionally involves nodes randomly communicating with their neighbors to exchange information. In their setup, local datasets contain one data point each, and the network topology assumes that 90% of the nodes are online at any time (the session duration follows a lognormal distribution). The local models are linear SVMs. There are significant differences in our settings: in our decentralized learning scheme, models are anchored on specific nodes and always trained on the same local data. Our topologies and data distributions are more complex and varied, and our local models are neural networks. Nevertheless, the work in [8] presents an interesting and complementary perspective to the approaches described at the beginning of the section and should be studied separately in future work.

In addition to the algorithmic-oriented contributions described above, there have been efforts to analytically characterize decentralized learning as a function of the graph topology. In [9], the authors introduce a comprehensive analytical framework that integrates various decentralized SGD methods. This paper develops a framework that supports local SGD updates, synchronous updates, and pairwise gossip updates within a network topology (the DFL algorithm applied in this article). It offers universal convergence rates for solving both convex and non-convex problems. This analysis operates under less stringent assumptions than earlier research about network properties assumptions. Nevertheless, a fundamental assumption is that the mixing matrix (i.e., the topology-dependent matrix that drives the model exchanges) is symmetric and doubly stochastic, a condition not inherently met by general complex network topologies. In our study, which is simulation-driven and not theoretical, we do not constrain the properties of the mixing matrix associated with the graph. In addition, while [9] focuses on the convergence properties, we are also interested in the transient regime of the learning process. In their exploratory study on the influence of network connectivity in decentralized SGD [10], the authors examine the generalization and stability of decentralized stochastic gradient descent (D-SGD). They provide a theoretical framework that correlates the network's topology with the generalizability of D-SGD, focusing on how the spectral gap affects learning outcomes. Unlike our paper, which considers more complex scenarios, the authors of [10] base their assumptions on an IID data distribution across nodes and restrict their analysis to simple network topologies (ring, grid, exponential, fully-connected).

In previous works, we have focused on the impact of different complex network topologies on the performance of DFL. Specifically, in [11], we started the exploration of such a relationship by observing how the data distribution, when carried out prioritizing low or high-degree nodes, affects the accuracy of the learning process for a set of standard network topologies such as Erdős-Rényi, Barabási-Albert, and Stochastic Block Model. In this paper, we proceed further to investigate the relationship highlighted in [11], considering additional datasets and centrality metrics for data assignment, and focusing on idempotent Erdős-Rényi and Barabási-Albert graphs (this enables direct comparison between the topologies, as explained in Section 5.1). In [12,13], we explore a problem that is orthogonal to the one addressed here, namely, the impact of network disruption on the evolution of the learning process. Specifically, we investigate a scenario where nodes (in a single Barabási-Albert graph) are "lost" during the learning process, potentially leading to the loss of valuable knowledge held by these nodes compared to the ones remaining active. Although related, these two problems are distinct and complementary.

3. Decentralized Federated Learning

Decentralized Federated Learning (DFL) is a decentralized machine learning solution that does not rely on a central coordinator. It consists of a variation of Federated Learning (FL) that operates in a fully decentralized setting where there is no central server. In this scenario, a number N of devices have to accomplish a distributed and collaborative learning task under the very same settings of federated learning [14]. Basically, in Decentralized Federated Learning, there is no central entity that coordinates the nodes: the orchestrator can either not exist at all as a central entity, or its role can be rotated among the participants. In a fully decentralized setting, the communication network is essentially peer-to-peer. The decentralized learning algorithms designed for such a system are typically composed of two main blocks: one step for the local training of the model using local data and one step devoted to the exchange and aggregation of the models' updates. These steps go on iteratively for n communication rounds. Here, we focus on the most popular decentralized learning approach, which we refer to as DecAvg. DecAvg is the decentralized equivalent of FedAvg [1], and it has been often used in the related federated and decentralized literature [2,5,15,16]. DecAvg combines reasonable effectiveness with simplicity, making it a suitable choice for our work. In fact, our aim is not to introduce a novel state-of-the-art decentralized learning solution but rather to assess a well-established method across various network topologies.

The communication network connecting the N devices can be represented as a graph. Therefore, we model the network connecting the nodes as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ the set of edges. We denote with ω_{ij} the weights on the edge between nodes i and j which would represent the trust between the two nodes. The self-trust ω_{ii} is a pseudo-parameter with which we capture the importance placed by node i on itself. We assume that only nodes sharing an edge are willing to collaborate with each other. Note that matrix $\Omega = \{\omega_{ij}\}$ is not generally symmetric. For this reason, certain theoretical results obtained in [9] may not hold in our case.

Each node $i \in \mathcal{V}$ is equipped with a local training dataset $\mathcal{D}_i$ (containing tuples of features and labels $(x, y) \in \mathcal{X} \times \mathcal{Y}$) and a local model h_i defined by weights $\mathbf{w}_i$, such that $h_i(\mathbf{x}; \mathbf{w}_i)$ yields the prediction of label y for input $\mathbf{x}$. Let us denote with $\mathcal{D} = \bigcup_i \mathcal{D}_i$ and with $\mathcal{P}$ the label distribution in $\mathcal{D}$. In general, $\mathcal{P}_i$ (i.e., the label distribution of the local dataset on node i) may be different from $\mathcal{P}$. This captures a realistic non-IID data distribution. At time 0, the model $h(\cdot; \mathbf{w}_i)$ is, as usual, trained on local data, by minimizing a target loss function ℓ – i.e., $\mathbf{w}_i = \text{argmin}_{\mathbf{w}} \frac{1}{|\mathcal{D}_i|} \sum_{k=1}^{|\mathcal{D}_i|} \ell(y_k, h_i(\mathbf{x}_k; \mathbf{w}_i))$, with $(y_k, \mathbf{x}_k) \in \mathcal{D}_i$.

We assume that nodes entertain a certain number of communication rounds, where they exchange and combine local models. At each communication round, a given node receives the local models from its neighbors in the communication graph, and averages it with its local model. Specifically, at each step t, the local model of the given node and the local models from the node's neighbors are averaged as follows:

$$\mathbf{w}_i(t) \leftarrow \frac{\sum_{j \in \mathcal{N}(i)} \omega_{ij} \alpha_{ij} \mathbf{w}_j(t-1)}{\sum_{j \in \mathcal{N}(i)} \omega_{ij}}, \tag{1}$$

where we have denoted with $\mathcal{N}(i)$ the neighborhood of node i including itself and α_{ij} is equal to $\frac{|\mathcal{P}_j|}{\sum_{j \in \mathcal{N}_i} |\mathcal{P}_j|}$ (and captures the relative weight of the local dataset of node j in the neighborhood of node i). Once the aggregation of models is performed, the local model is trained again on the local data.

From Eq. (1) we can see that the aggregation step takes into consideration only the 1-hop neighboring nodes of the *i-th* node (including the node itself). As a result, the topology determines the paths along which information, or "knowledge", travels and influences the speed and efficiency of information dissemination. The structure of the network, such as its degree distribution, clustering coefficient, and connectivity patterns, can impact the performance and effectiveness of decentralized federated algorithms. Thus, it is reasonable to expect the network topology to play a crucial role in DFL. This is the rationale for analyzing the impact of network topology in such settings.

4. Network topologies

In this paper, we analyze three different network topologies, i.e., Erdős-Rényi (ER), Barabási-Albert (BA), and Stochastic Block Model (SBM), to investigate how their properties impact on the diffusion of knowledge during a fully decentralized learning process. We consider non-directed graphs, assuming that node communication is always bidirectional.

The ER model is a model for generating random graphs with a homogeneous structure, where nodes are connected to each other with a fixed probability. ER is defined by two parameters: N, the number of nodes in the network, and p, the probability of an edge existing between any two nodes in the network (regardless of their degree). The ER model shows a phase transition when the fixed probability p approaches the critical value $p^* = \ln(N)/N$ [17]. Specifically, the value p^* is a sharp threshold for the connectedness of the network: for values of p above p^* the network goes from presenting isolated nodes to almost surely be made of a unique connected component such that all nodes are reachable within a finite number of hops.

The BA is an algorithm for generating random scale-free networks, i.e., networks with a power-law (or scale-free) degree distribution, using a preferential attachment mechanism [18]. In the BA model, nodes are connected preferentially based on their degree. Specifically, the probability of an edge forming between two nodes is proportional to the nodes' degree, which leads to the emergence of a scale-free degree distribution. Since the degree distribution follows a power law, few nodes have a very high degree while most nodes have a low degree. This can result in a structure with few well-connected hubs, which are known to facilitate information flow across the network. A BA network is defined by two parameters: N, the number of nodes in the network, and m, the number of edges added to the network for each new node (hence, the minimum degree of nodes).

The SBM is a probabilistic model for networks that exhibit a modular structure, i.e., the SBM generates a network with a clear community structure where nodes are grouped together based on their connectivity patterns [19]. Nodes belonging to the same group are more closely connected to each other than to nodes in another group. Formally, the SBM is defined by the following parameters: N, the number of nodes in the network; B, the number of communities (called blocks); $n_1, n_2, \dots, n_B$, the sizes of the blocks where n_i is the number of nodes in block i; p_{ij}, the probability of an edge existing between a node in block i and a node in block j (with p_{ii} the probability of links inside the block).

These three models capture important properties of complex networks. ER networks, which are random and well-mixed, provide insights into how information propagates in networks without a pronounced structure, with homogeneity in terms of degree and low clustering coefficient. While ER networks rarely mirror topologies observed in real large-scale socio-physical systems, it has been argued that they could approximate some ad-hoc wireless/sensor networks [20] or social random encounter networks (where people that do not know each other in advance start interacting). In addition, they provide a popular benchmark and optimal mathematical tractability [21]. BA graphs are characterized by a highly skewed degree distribution with few high-degree nodes and many low-degree nodes. The presence of nodes with a high degree can enhance rapid dissemination, contributing to efficient diffusion, but may overshadow the contribution of low-degree nodes. Since real networks such as the Internet, the World Wide Web, air transportation, and social networks often exhibit a power-law degree distribution [21], the BA model, with its preferential attachment mechanism, is considered an excellent generative graph model to replicate these realistic topological structures. Finally, SBM introduces the concept of community structure. In many real networks (e.g., collaboration networks, social networks), in fact, nodes also organize themselves into densely linked groups [22]. SBM graphs feature a well-defined community structure that allows us to investigate how knowledge spreads within and between communities. In practice, depending on the specific application, we expect a realistic communication graph behind decentralized learning tasks to blend, to different degrees, the preferential attachment element of BA with some community structure.

Table 1
Equivalent values of p for the ER networks idempotent to the corresponding BA networks.

ER-idem($m = 2$)	ER-idem($m = 5$)	ER-idem($m = 10$)
$p = 0.0396$	$p = 0.0960$	$p = 0.182$

5. Experimental settings

5.1. Network settings

In this paper, we consider unweighted graphs with 100 nodes, where edges are generated as follows.

Barabási-Albert. Three different cases regarding the parameter of preferential attachment are chosen: $m = 2, 5, 10$, leading to networks with increasingly higher node degrees, as illustrated in Fig. 1.

Erdős-Rényi. In order to draw direct comparisons against BA networks, for each BA graph we generate the corresponding idempotent ER network,[2] i.e., an ER network with the same number of nodes and edges with respect to its corresponding BA network. To this aim, for each BA network, we randomly reshuffled the links between the nodes keeping the density of the original BA networks. The reshuffling was done by employing the corresponding built-in function of the `graph-tool` python library [23]. We denote with ER-idem(m) the ER idempotent to the BA graph with parameter m. The resulting networks are illustrated in Fig. 2. In order to understand where the generated ER graphs are positioned with respect to the critical threshold (see discussion in Section 4), the equivalent p values of the ER-idem(m) are shown in Table 1. For the $m = 2$ case, the corresponding ER graph will have a p value close to the critical one for the connectedness of the network, which is $p^* = 0.046$ (obtained from $\frac{\ln 100}{100}$ [17]). For the other networks, the values are well above the critical value for connectedness. An increase in the values of p when the number of nodes stays fixed results in a higher density of edges, as shown in Table 2, which reports the main network indices for the considered graphs.

The clustering coefficient is higher for BA while the average shortest path length and the diameter tend to be higher for ER. Recall that a higher clustering coefficient implies that neighborhoods are better connected internally, while a higher average shortest path length implies that it takes more hops, on average, to reach a generic node in the graph (a similar consideration holds for the diameter).

Stochastic Block Model. Nodes are grouped into 4 communities (denoted as C_1, C_2, C_3, C_4) of equal size (25 nodes each). The probability of extrinsic connections ($p_{ij},\ j \neq i$) is set to 0.01, whereas the probability of intrinsic connections (p_{ii}) is set to 0.8 in one case and 0.5 in the second case. The resulting graphs are plotted in Fig. 3. The reason for such values of intrinsic connectivity is to explore the impact of a loosely or more tightly connected community structure. The level of connectivity obtained by the $p_{ii} = 0.5$ case represents a balanced or moderate degree of interconnectedness within the communities of the network. Whereas the intrinsic connectivity of 0.8 indicates a scenario where the connections within the blocks are denser (closer to 1).

The degree distributions of the three network models are illustrated in Fig. 4.

[2] Note that this implies that the ER graphs considered in this paper are different from those studied in [11].

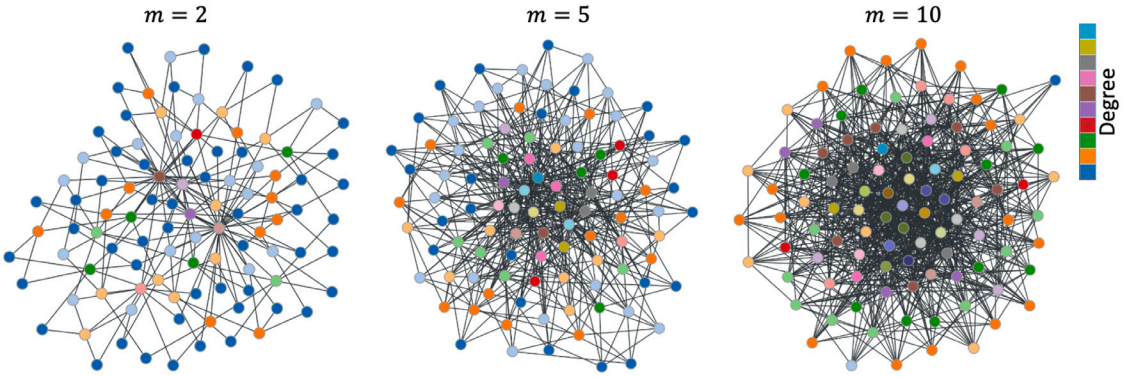


Fig. 1. The three BA networks considered in our experiments, with increasing value of the preferential attachment parameter m. Vertices are colored based on their degree. The color map is shown in the figure.

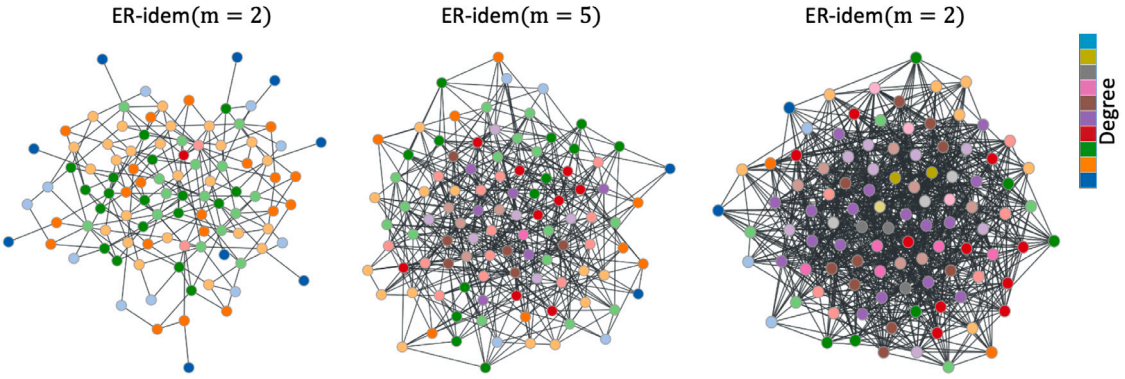


Fig. 2. The three different realizations of the Erdős-Rényi graph, idempotent to the corresponding BA graph with increasing value of the preferential attachment parameter m. Vertices are colored based on their degree. The color map is shown in the figure.

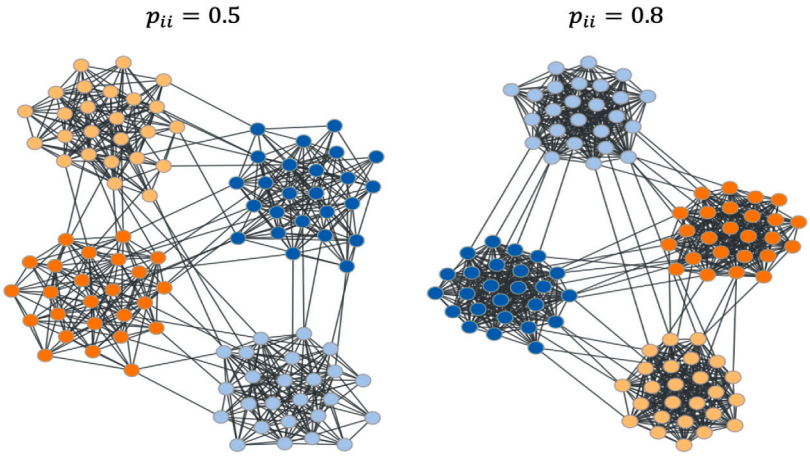


Fig. 3. The two SBM graphs. Coloring is based on the data distribution as described in the body of the paper.

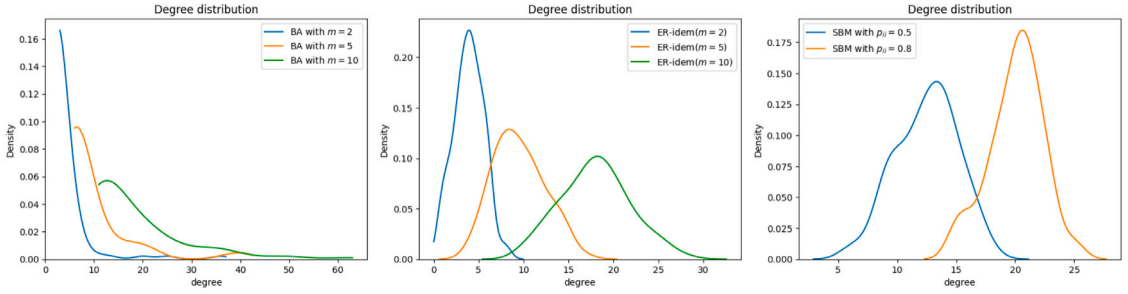


Fig. 4. Degree distributions for the analyzed networks. From left to right: Barabási-Albert, Erdős-Rényi and Stochastic Block model.

Table 2
Summary statistics for the considered networks.

	Barabási-Albert m			Erdős-Rényi idem(m)			SBM p_{ii}	
	2	5	10	2	5	10	0.5	0.8
Number of nodes	100	100	100	100	100	100	100	100
Number of edges	196	475	900	196	475	900	624	1003
Avg degree	3.92	9.50	18.00	3.92	9.50	18.00	12.48	20.06
Density	0.0396	0.0960	0.182	0.0396	0.0960	0.182	0.126	0.203
Clustering coef.	0.164	0.213	0.296	0.0394	0.0900	0.194	0.424	0.747
Avg shortest path l.	2.85	2.17	1.85	3.49	2.28	1.85	2.56	2.30
Diameter	5	3	3	7	4	3	5	4

5.2. Datasets and data distribution

For our experiments, we analyzed three different datasets: the widely used MNIST image dataset [24], the EMNIST dataset [25] and the Fashion-MNIST dataset [26]. This is done in order to verify that the results are generally valid and are not hindered by the particular choice of the dataset.

- MNIST: This dataset contains images of handwritten digits from 0 to 9 (10 classes). It has 60,000 samples for training and 10,000 for testing. Each image is grayscale and 28 × 28 pixels in size, and the digits are written by different people in various styles.
- Fashion-MNIST: This dataset contains images of ten types of clothing items (hence 10 classes). It has the same number of training/test images as MNIST, and the images are also grayscale and 28 × 28 pixels.
- EMNIST Letters: This dataset is a part of the larger EMNIST dataset, which contains handwritten digits and letters. EMNIST Letters focuses only on uppercase letters, and it has 26 classes (A to Z) with 20,800 training samples and 3,280 test samples.

In increasing order of classification difficulty, the datasets are ranked as MNIST, Fashion-MNIST, then EMNIST Letters, due to the increasing complexity of the corresponding images. The Fashion-MNIST images of clothing items tend to have more varied shapes and textures than MNIST digits. EMNIST Letters have many more classes (26 vs 10), with many similar shapes and more potential for confusion. We encounter one such confusing case in our scenarios with the Fashion-MNIST dataset, where the Shirt class distorts the classification of T-shirt-Top, Pullover, and Coat classes. For ease of presentation, the influence of these cross-correlations on decentralized learning is explored separately in Section 6.5. For the results in Sections 6.1–6.3, we simply remove the offending Shirt class to show that when cross-correlation is not significantly present, the trends are the same as with the other datasets.

The goal of the analysis is to characterize the effect of the network topology in the knowledge spreading process, i.e., the ability of nodes to learn data patterns they have *not* seen locally, but that other nodes in the network have seen. Therefore, we split the classes across nodes as follows. Note that, on the assigned classes, each node gets the same amount of images.

For ER and BA networks, we divide the classes into two groups: the first group (G1) is composed of the first half of the classes, e.g., for MNIST classes 0, 1, 2, 3, 4, and the second group (G2) of the other half of the classes, i.e 5, 6, 7, 8, 9. All nodes receive an equal share (selected randomly) of data from G1. Data from G2 are allocated only to a subset of nodes, selected as follows. For each type of network measure considered, we examine two cases where data in G2 are assigned to the 10% of nodes with the highest values and the 10% with the lowest values, respectively. The rationale is thus to allocate "full knowledge" (i.e., a complete subset of all classes) either to high-centrality or low-centrality nodes, and study the effect of the network topology in both cases. In the following, these configurations are referred to as "highest-focus" and "lowest-focus", respectively. The measures taken into account are:

- Degree centrality
- Betweenness centrality
- Clustering coefficient

For the degree centrality measure, we will refer to the highest-focus and lowest-focus cases as hub-focus and edge-focus to easily distinguish the degree results with the other centrality measures. Specifically, starting from the node(s) with the highest (lowest) centrality, we pick nodes until we reach 10% of the network. In case adding all nodes at a given centrality measure value results in more than 10% of the network, we randomly pick, among nodes with that centrality measure value, a subset that allows us to fill the 10% subset.

The *degree centrality* is calculated by taking the total number of edges connected to a node (i.e., its degree). Being directly connected with many other nodes, nodes with high degree centrality are generally expected to be more influential in the network, as they can reach many nodes at once leveraging their direct connections. When considering the degree centrality, low-centrality nodes are edge nodes (i.e., nodes at the periphery of the graph), hence we may use the terms lowest-focused and edge-focused interchangeably. Similarly for highest-focused and hub-focused.

The *betweenness centrality* measures the importance of a node in a network by quantifying its role as a connector or bridge between other nodes [27]. It identifies nodes that lie on a high number of shortest paths, which are the most efficient routes for information to travel between different parts of the network. These nodes play a critical role in facilitating communication and information flow, as they act as intermediaries between different communities or groups of nodes. Nodes with high betweenness centrality are often referred to as *bridges*. In contrast, nodes with low betweenness centrality lie on relatively few shortest paths and are less influential in connecting different parts of the network. They are often referred to as *isolates*.

The *clustering coefficient* of a node measures the proportion of its neighbors that are also connected to each other [21]. Nodes with higher clustering coefficient centrality have neighbors that are well-connected to each other, indicating the presence of cohesive groups or clusters in the network. The higher the clustering coefficient, the more homogeneous a group of nodes is in terms of connections, the more balanced information spread is expected to be within the cluster.

The rationale behind the choice of these centrality measures is their ability to capture distinct aspects of network dynamics and structure. Degree centrality highlights the importance of well-connected nodes, clustering coefficient emphasizes local connectivity patterns, and betweenness centrality identifies nodes critical for global information flow.

For SBM networks, instead, we divided the dataset classes into subsets based on the communities the nodes belong to, without overlap. Therefore, since we study SBM topologies with 4 communities, each community gets two classes: community 1 sees classes 0 and 1; community 2 sees classes 2 and 3; community 3 sees classes 4 and 5; community 4 sees classes 6 and 7. This data distribution is designed to challenge the knowledge-spreading process since maximum learning accuracy can only be achieved if information from all the external communities is brought into the local one. For the SBM networks, we

Table 3
Architecture and settings of local models.

Dataset	Model	Structure	Activation	Kernel size
MNIST	MLP	FC:512,256,128	ReLU	–
Fashion	CNN	Conv2d:32,64 FC:9216,128	ReLU	3 × 3
EMNIST	CNN	Conv2d:32,64 MaxPool(2) Dropout(.25) FC:9216,128 Dropout(.5) FC:128	ReLU	3 × 3

only analyzed the MNIST dataset since, as we will see, the system is unable to spread knowledge even in the simplest task of classifying the MNIST dataset.

5.3. Learning task and evaluation metric

For the learning task, we consider a simple classifier as a model to be trained, and we focus on two performance figures. We consider the accuracy over time at each node to assess the effectiveness and speed of knowledge diffusion across the network. When needed, we also analyze the confusion matrix. For SBM networks, we also investigated the average confusion matrix across nodes of the same community. Specifically, for each node, we compute the confusion matrix for the MNIST classes and then take the average across all nodes in the same community.

We implemented the DecAvg scheme within the custom SAISim simulator, available on Zenodo.[3] SAISim is developed in Python and leverages state-of-the-art libraries such as PyTorch and NetworkX for deep learning and complex networks, respectively. On top of that, SAISim implements the primitives for supporting fully decentralized learning. The local models of nodes are different based on the dataset under study. We described them below and their configuration (inherited from [7]) is provided in Table 3.

- MNIST dataset: Multilayer Perceptron (MLP) with three layers (sizes 512, 256, 128) and ReLu activation function. SGD is used for the optimization, with learning rate 0.01 and momentum 0.5. The number of local training epochs between each round of updates' exchange is set to 5.
- EMNIST: Convolutional Neural Network (CNN) with 9 layers and ReLu activation function. SGD is used for the optimization with learning rate 0.001 and momentum 0.9. The number of local training epochs between each round of updates' exchange is set to 5.
- Fashion-MNIST: Convolutional Neural Network (CNNs) with 8 layers and ReLu activation functions. SGD is used for the optimization with learning rate 0.001 and momentum 0.9. The number of local training epochs between each round of updates' exchange is set to 5.

6. Results

The rest of this section is organized as follows: in Sections 6.1 and 6.2, we present individually the results obtained for the ER and BA networks. Then in Section 6.3 we discuss how the two topopologies compare against each other. In Section 6.4 we present and discuss the results obtained for the SBM networks. Finally, in Section 6.5, we discuss the effect of cross-correlations between classes on decentralized learning done through DecAvg, using Fashion-MNIST as a study case. Due to space constraints and readability, the plots for EMNIST and Fashion-MNIST are provided in the body of the paper only for selected discussions to showcase that the same trend observed for MNIST is generally confirmed. However, the full set of plots is provided in Appendices A and B, for EMNIST and Fashion-MNIST respectively.

Please recall that the analysis carried out for Fashion-MNIST is twofold. In Section 6.5, the performance of DecAvg on the complete set of labels is presented, where we show that some classes suffer from cross-correlation and how this is detrimental to decentralized learning. Specifically, the cross-correlation is between the 7th class (Shirt) and the 1st (T-shirt/top), 3rd (Pullover), and 5th (Coat) class in the Fashion-MNIST dataset. To demonstrate that in the absence of significant cross-correlation, the trends are the same as for the other datasets, in Sections 6.1–6.3, we remove the problematic class and present the results without it.

6.1. Decentralized learning over Erdős-Rényi graphs

As explained in Section 5, for the ER model we consider three settings that are idempotent to the chosen Barabási-Albert graphs, discussed later on in Section 6.2. This allows us to draw direct comparisons between the main characteristics of a decentralized learning process on equivalent topologies (Section 6.3). Let us begin our analysis by considering the relationships between the centrality measures w.r.t. the considered network topology, as shown in the scatterplots of Figs. 5–6. The scatterplots in Fig. 5 show the relation between degree centrality and betweenness centrality for the three distinct configurations of the Erdős-Rényi (ER) graph. In these scatterplots, the highest-focus and lowest-focus nodes selected for the betweenness centrality metric are highlighted in red and blue, respectively. The scatterplots show a relationship of direct proportionality between betweenness centrality and degree. Nodes with higher degrees tend to exhibit higher betweenness centrality, indicating that well-connected nodes also play crucial roles as bridges or intermediaries in the network. Hence, we expect a similar behavior of the system when data is distributed either according to the betweenness centrality or their degree. Fig. 6 shows the scatterplots of the clustering coefficient vs the degree centrality: as it is common in such graphs, the higher the clustering coefficient, the lower the degree. The intuitive understanding behind nodes with high clustering coefficients having smaller degrees lies in the fact that high-degree nodes, connected to numerous other nodes, are less likely to have tightly knit connections among their neighbors, causing them to cluster in the lower degree range of the plot.

6.1.1. ER: Degree centrality used for G2 data assignment

We start our analysis with the case of data in the G2 subset being assigned based on the degree centrality. For clarity, we denote nodes exclusively assigned G1 data as *G1 nodes* and, conversely, refer to nodes assigned both G1 and G2 data as *G2 nodes*. In Fig. 7 we show the evolution of the accuracy per node over time (each curve corresponds to one node) on the MNIST dataset. The top-row plots refer to the edge-focused case, where the digits in class group G2 are assigned to edge nodes, while the bottom-row plots correspond to the hub-focused scenario, where the high-degree nodes get the G2 class data. As expected, the nodes that are assigned both G1 and G2 data (the group of curves with higher accuracy in the figure) show an excellent learning performance. It is more interesting to focus our attention on the other nodes, the ones assigned only data in G1 that can only learn the classes in G2 thanks to the knowledge (models) they receive from other nodes. In the bottom row of Fig. 7 we can observe the accuracy over time when G2 data are assigned to the nodes with the highest degree. Since a high degree implies better connectivity, we expected it to be easier for edge nodes to be reached by the "good" models of the central nodes, and this to be increasingly true as the average degree of the network increases (from left to right). This is confirmed in the figure, where we see that some edge nodes are struggling on ER-idem($m = 2$) but progressively catch up when m increases. *In other words, when the highest degree nodes*

[3] https://doi.org/10.5281/zenodo.5780042

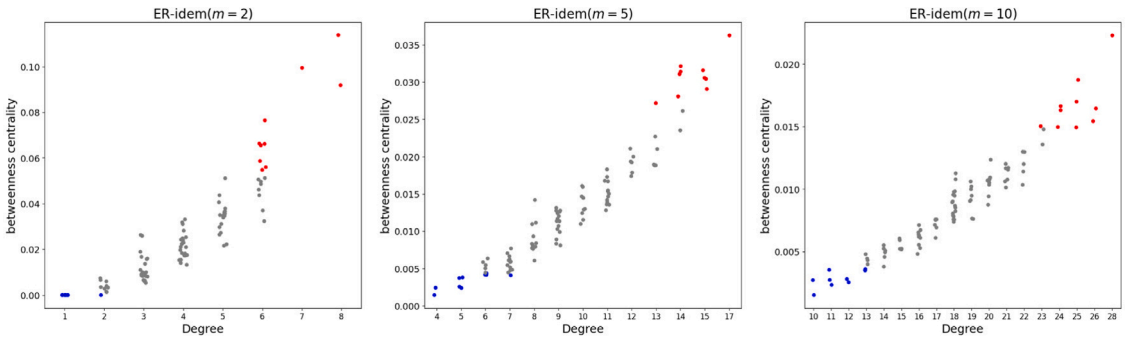


Fig. 5. Scatterplots of the nodes' betweenness centrality measures vs the degree in our ER graphs. Increasing value of connectedness going from left to right. Red dots are the nodes chosen for the highest-focused case, the blue dots are the nodes chosen for the lowest-focused case. The scatterplots show that the betweenness centrality is proportional to the degree. Gray dots represent nodes that are neither selected in the highest-focused case nor in the lowest-focused case, as they have intermediate values for both metrics.

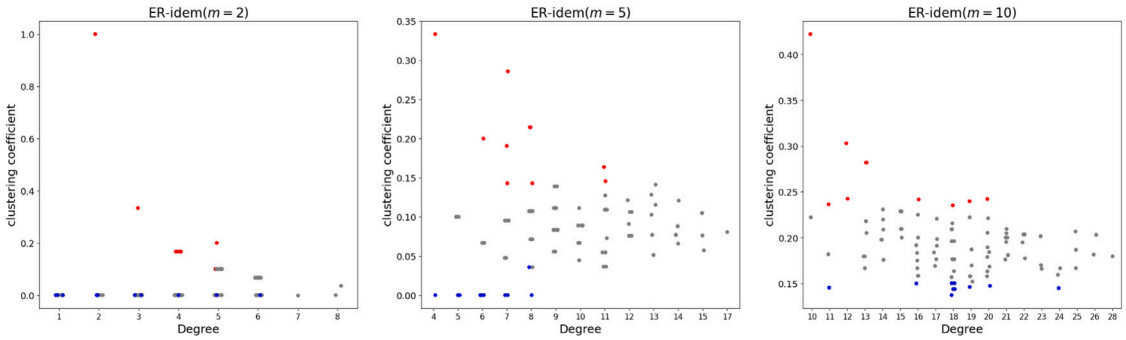


Fig. 6. Scatterplots of the nodes' clustering coefficient vs the degree in our ER graphs. Increasing values of connectedness going from left to right. Red dots are the nodes chosen for the highest-focused case, the blue dots are the nodes chosen for the lowest-focused case. There is no strong correlation between the two metrics. Gray dots represent nodes that are neither selected in the highest-focused case nor in the lowest-focused case, as they have intermediate values for both metrics.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

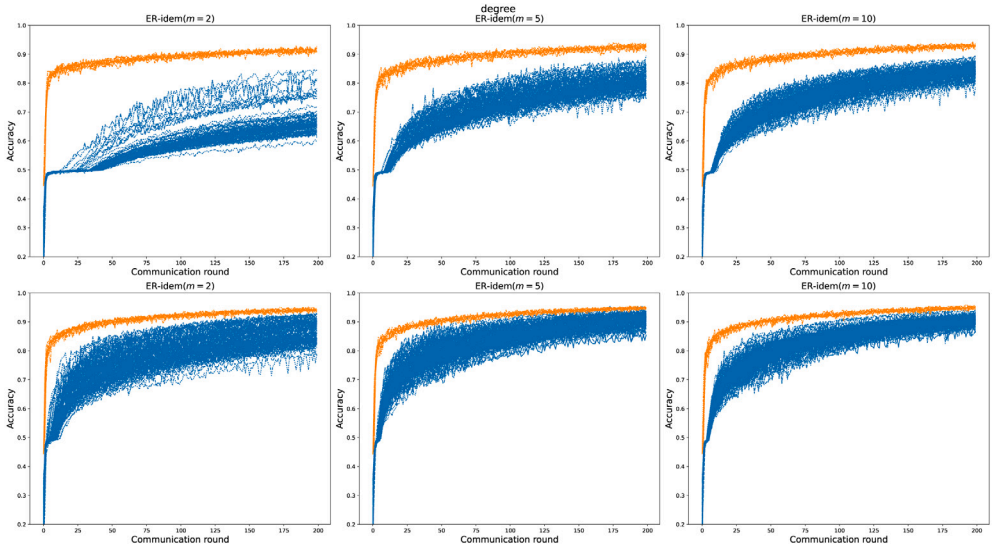


Fig. 7. MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

enjoy higher accuracy, they are able to drag all the other nodes closer to their performance efficiently.

Conversely, when G2 data are instead assigned to nodes with the smallest degree, we expect the decentralized learning process to be less effective. The intuition is that nodes with G2 data, in this case, are poorly connected. Hence, the knowledge they bring has a hard time percolating through the network. The top row of Fig. 7 confirms this. Especially when the average degree is smallest, corresponding to ER-idem($m = 2$), we see nodes that, after 200 communication rounds, have barely increased their accuracy above the baseline 0.5 (corresponding to the accuracy on the G1 class group that is available locally to all nodes). When comparing to the previous case, then, we conclude that *when G2 nodes are less central (according to the degree centrality) they have difficulties in dragging G1 nodes towards better accuracy by contributing*

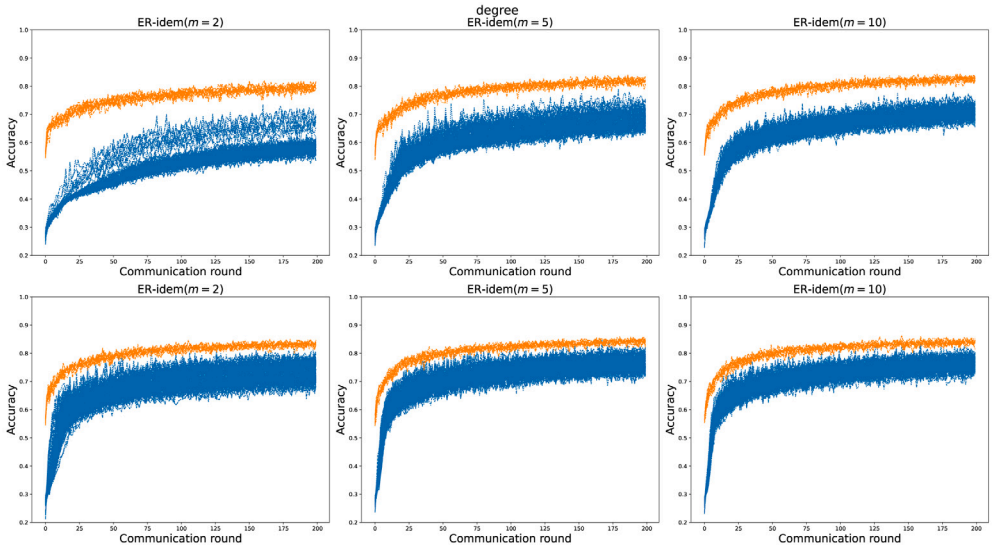


Fig. 8. EMNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

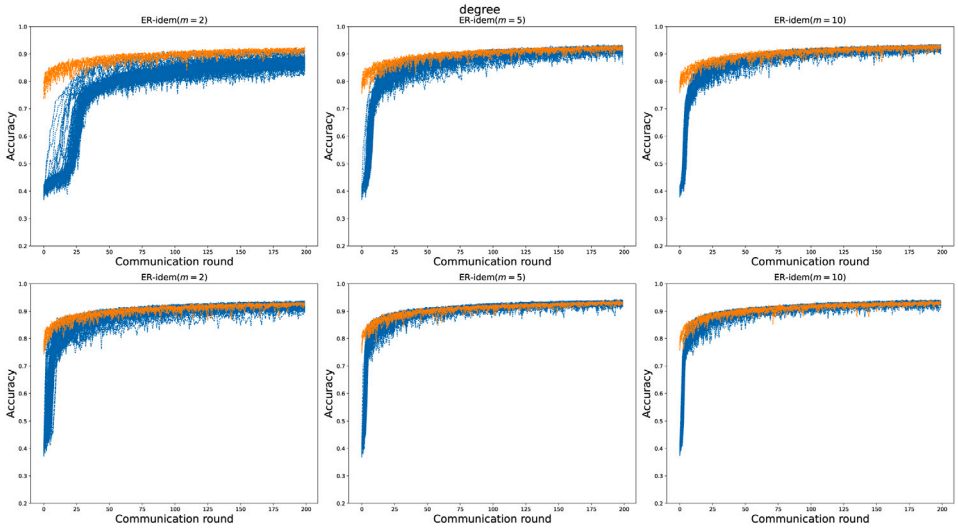


Fig. 9. Fashion-MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

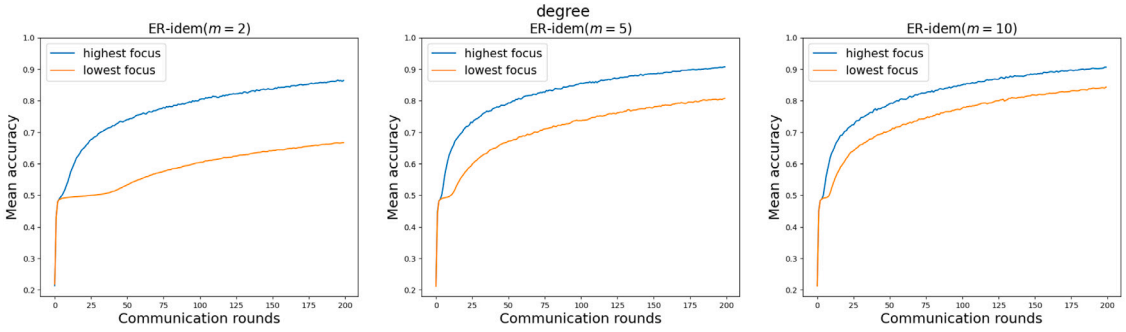


Fig. 10. MNIST – Mean accuracy over time in ER networks (average across G1 nodes only, G2 data assigned based on degree centrality). Highest-focused (blue) and lowest-focused (orange) cases.

their "good" models. This effect arises from the combination of the average-based aggregation strategy in DecAvg *along with the topological properties of the graphs.*

The same results for EMNIST and Fashion-MNIST are shown in Figs. 8 and 9, and substantially confirm the findings discussed above. The

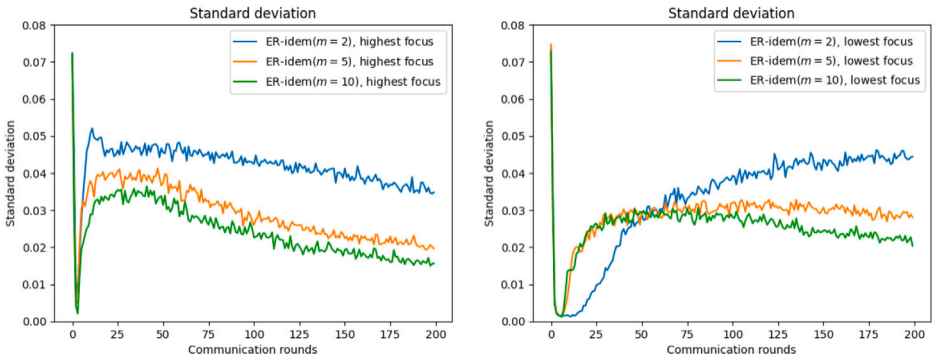


Fig. 11. MNIST – Standard deviation of accuracy over time in ER networks (std dev. across G1 nodes only, G2 data assigned based on degree centrality). Left to right: highest-focus and lowest-focus cases.

main difference with respect to the MNIST case is that it is evident that EMNIST-Letters is an intrinsically more difficult dataset to classify since even the G2 nodes are not able to reach 90%. Vice versa, Fashion-MNIST with one G2 classless (the Shirt class) makes classification easier for G1 nodes. Given that the trends are substantially confirmed, in the following we focus only on the MNIST dataset (but EMNIST and Fashion-MNIST plots can be found in Appendices A and B).

In Fig. 10, we aggregate the accuracy among all G1 nodes in the same experiment, and we show the evolution over time of the average of the accuracy. Coherently with the reasoning above, the average accuracy in the edge-focus case is much lower than in the hub-focus case. The gap decreases as we increase the overall connectivity of the network, i.e., as m increases. The poor average performance of the edge-focus case is due to high-degree nodes hindering the spreading of knowledge from edge nodes. This effect is due to the model averaging mechanism of decentralized learning, as we explain in more detail in Section 6.2. In Fig. 11, we present the standard deviation of the accuracy for all G1 nodes in the ER networks. As we can see, the hub-focused case (left) shows separated curves having a clear gap between ER-idem($m = 2$) and the others. This higher variability can be attributed to the effect of longer paths of ER-idem($m = 2$). Since the equivalent p value of ER-idem($m = 2$) is around the critical value for the connectedness, the network has longer paths, impeding the synchronization of local models across nodes. In this scenario, nodes are situated farther apart compared to the other more connected networks (values of p well above the critical value). In the edge-focus case, we observe two separate behaviors. When the connectivity is high enough, i.e., for ER-idem($m = 5$) and ER-idem($m = 10$), the trend of the standard deviations is more similar to the hub-focus case, with a decrease over time although at a slower pace. For the case ER-idem($m = 2$), the trend of the variability is increasing in the observed time window, proving that the connectivity is insufficient to let all nodes synchronize and collaborate constructively within a reasonable timeframe.

6.1.2. ER: Betweenness centrality used for G2 data assignment

In this section, we investigate the accuracy performance of decentralized learning when G2 data are assigned based on the betweenness centrality. We focus here on the MNIST datasets but the findings are the same for EMNIST and Fashion-MNIST (Appendices A and B). In Fig. 12 we show the evolution of the accuracy per node over time. Due to the high correlation between degree centrality and betweenness centrality in our ER graphs (Fig. 5), *a similar behavior as in the degree-based case emerges*. Consistently with the results obtained in the previous section, the highest-focused case has a better performance than the lowest-focused case.

In Fig. 13, we show the average accuracy of all the nodes with assigned G1 images for both the highest and lowest focus cases. As expected, the highest focus case shows higher accuracy levels than the lowest focus counterpart and an overall small difference in performance. Looking at Fig. 14, we obtain results similar to the degree-based analysis. Also, in this case, the connectedness of the graph influences the gap between the curves for the ER-idem($m = 2$) case and the other cases. As we increase the mean connectivity of the graph, not only does the gap decrease, but so does the standard deviation, since the augmented connectivity helps the other nodes to spread their information to the other nodes.

6.1.3. ER: Clustering coefficient used for G2 data assignment

As we remarked when discussing Fig. 6, the clustering coefficient has no significant relation with the degree centrality (and, as a result, with the betweenness centrality) in our ER graphs. This implies that the degree of nodes can be high or low regardless of their clustering coefficient. Given that degree centrality seems to be the main driver of knowledge spreading, we do not expect much variation in the accuracy between the highest and lowest focus cases. This is evident in Fig. 15. This implies that the *clustering coefficient does not reflect in any meaningful way the model aggregation and dissemination process. For this reason, whether G2 data are assigned to nodes with low or high clustering coefficients does not make a difference.* The similar performance between the highest and lowest focus cases is further confirmed by the mean accuracy curves of all G1 nodes showed in Fig. 16, where we can see that the curves have a small gap, with the highest focus case being above the lowest focus one with the exception of the last scenario. In the high-focus case an increase of connectivity does not help much since high-focus nodes are anyway poorly connected. In the low-focus case, this is the opposite, and the rightmost plot of Fig. 16 shows that, in this case, the effect of higher connectivity for the low-focus case is more important than allocating G2 on most central nodes.

In Fig. 17 we show the standard deviation of the accuracy for all the scenarios and all networks. As we can see, all the curves have the same trend and are closer together. Therefore, for better visualization, we plot the different network realizations separately. For the ER-idem($m = 2$) case, the highest-focus case is above the lowest-focus counterpart, meaning that the highest-focus shows a higher standard deviation. This is due to the locality of information dissemination given that G2 nodes are well-connected inside their neighborhood but might not be as homogeneously connected to the rest of the network.

The same conclusion holds for the EMNIST and Fashion-MNIST datasets, whose plots can be found in Appendices A and B.

6.2. Decentralized learning over Barabási-Albert graphs

In this section, we focus our analysis on the Barabási-Albert topology, with three different settings for the parameter related to the preferential attachment: $m = 2, 5, 10$. Preliminarily, we investigate, as we did for ER, the relationship between the different centrality measures used for G2 data assignment (Fig. 18). The relationship between degree and betweenness centrality is still one of direct proportionality, as expected, but with a more pronounced quadratic trend.

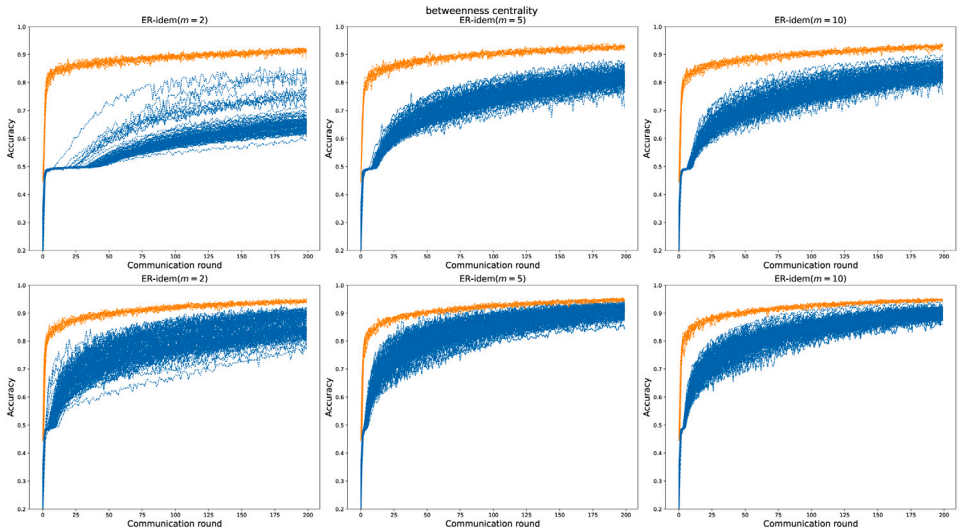


Fig. 12. MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on betweenness centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

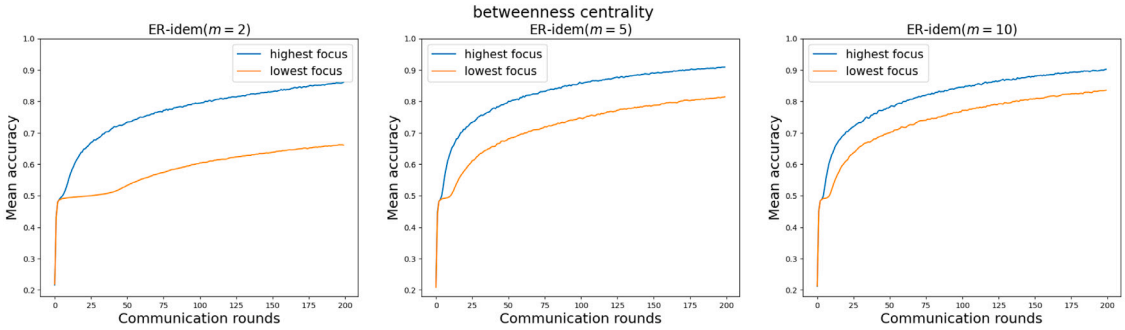


Fig. 13. MNIST – Mean accuracy over time in ER networks (average across G1 nodes only, G2 data assigned based on betweenness centrality). Highest-focused (blue) and lowest-focused (orange) cases.

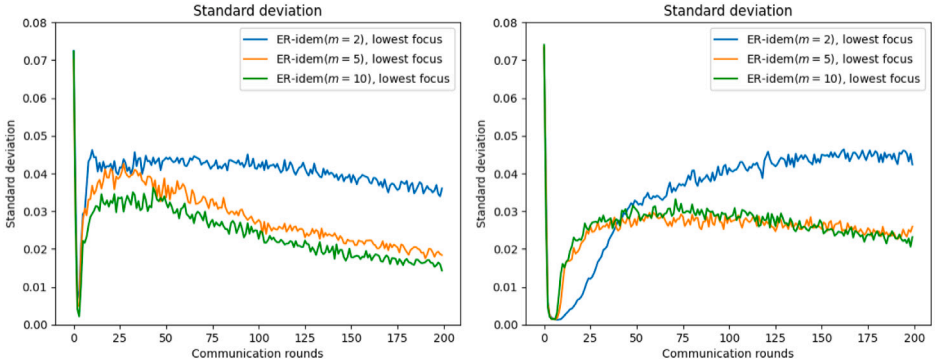


Fig. 14. MNIST – Standard deviation of accuracy over time in ER networks (std dev. across G1 nodes only, G2 data assigned based on betweenness centrality). Left to right: highest-focus and lowest-focus cases.

Note, also, that nodes reach much higher values for both centrality metrics with BA, which follows directly from the topological properties of BA networks. Moreover, due to the power law nature of the BA degree distribution, the nodes chosen for the highest focus case (those indicated by the red dots) span a wider range of degrees than in ER. When looking at the degree centrality vs clustering coefficient (Fig. 19), BA graphs exhibit nodes with both high and low clustering coefficients concentrated on the left-hand side of the scatterplot, suggesting a generally low degree for these nodes (with the exception of some nodes with a medium-range degree when $m = 5$).

6.2.1. BA: Degree centrality used for G2 data assignment

Focusing on MNIST, we show the accuracy over time for the case in which degree centrality is used to drive the G2 data assignment in Fig. 20. First, in the hub-focused case (bottom row), the performance for varying values of the minimum degree m is basically indistinguishable (this is confirmed by looking at the average and standard deviation of the accuracy in Fig. 24). This means that hubs (in this case, high-degree nodes are *real* hubs) spread knowledge extremely efficiently, irrespective of the connectivity of the rest of the nodes. The edge-focused case (top row of Fig. 20) is more challenging. As in the case of ER networks, edge nodes are unable to spread their

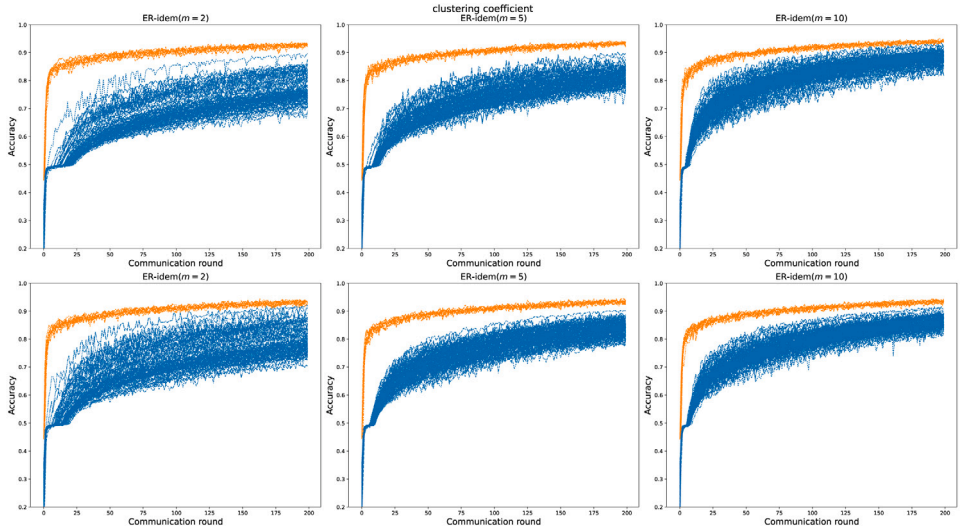


Fig. 15. MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on clustering coefficient). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

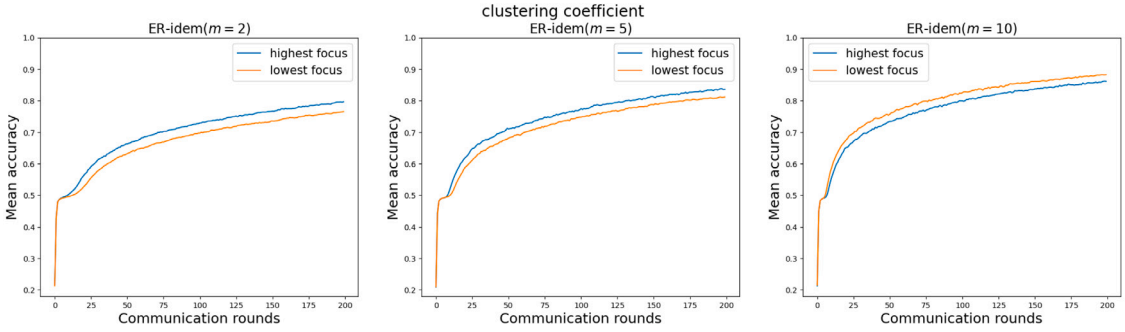


Fig. 16. MNIST – Mean accuracy over time in ER networks (average across G1 nodes only, G2 data assigned based on clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases.

Fig. 17. MNIST – Standard deviation of accuracy over time in ER networks (std dev. across G1 nodes only, G2 data assigned based on clustering coefficient). Left to right: increasing values of the connectedness.

knowledge efficiently, and the accuracy gap between edge nodes and non-edge nodes remains strong all throughout. In Fig. 24, we observe that larger values of m (i.e., stronger connectivity) help improve the average accuracy in the edge-focused case but not significantly, and the variability is reduced (as shown by the standard deviation curves on the right-hand side plot).

Given the distinctive features of high-degree nodes in the BA network, we ask whether they are at an advantage or disadvantage in an edge-focused scenario. Thus, in Fig. 23, we focus on the G1 nodes for the edge-focused scenario, highlighting in red the behavior of high degree nodes. For smaller m, the neighborhood of these hubs is smaller. Hence, the aggregation is more influenced by the capabilities of the specific nodes composing the neighborhood, resulting in higher variability. As m increases, the neighborhood of hubs becomes so large that it will include both "good" models (with G1 and G2 knowledge) and models knowing only about G1. The "good" models are then more likely to get lost in the average. Hence, the accuracy of previously well-performing hubs goes down as m increases. However, the overall accuracy for G1 nodes does not decrease. This is because G1 nodes that are not hubs increase their connectivity as m increases. This results in

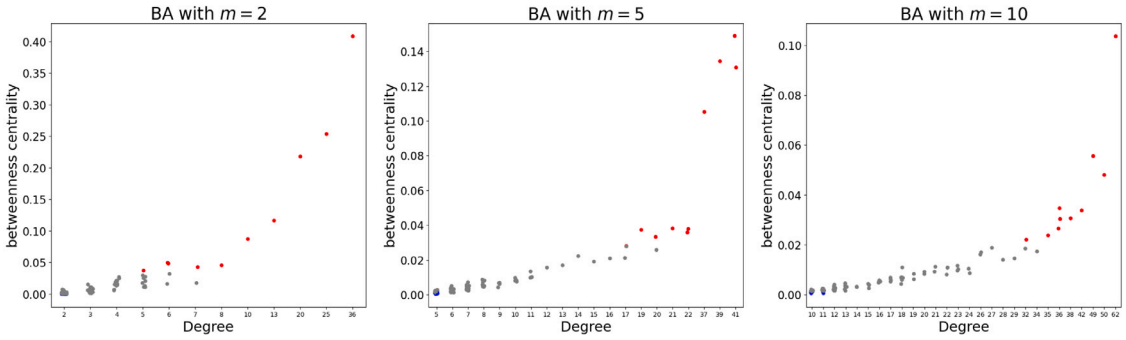


Fig. 18. Scatterplot of the BA graphs. Betweenness centrality vs degree. Gray dots represent nodes that are neither selected in the highest-focused case nor in the lowest-focused case, as they have intermediate values for both metrics. Red dots: highest-focus nodes; blue dots: lowest-focus nodes.

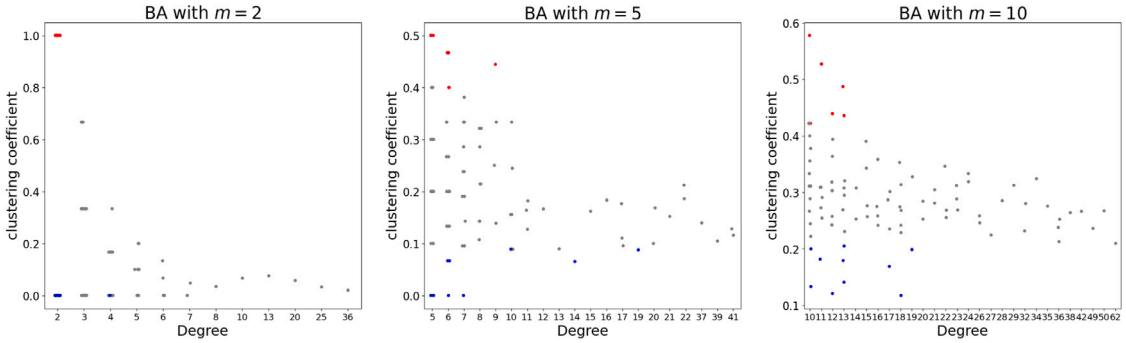


Fig. 19. Scatterplot of the BA graphs. Clustering coefficient vs degree. Gray dots represent nodes that are neither selected in the highest-focused case nor in the lowest-focused case, as they have intermediate values for both metrics. Red dots: highest-focus nodes; blue dots: lowest-focus nodes.

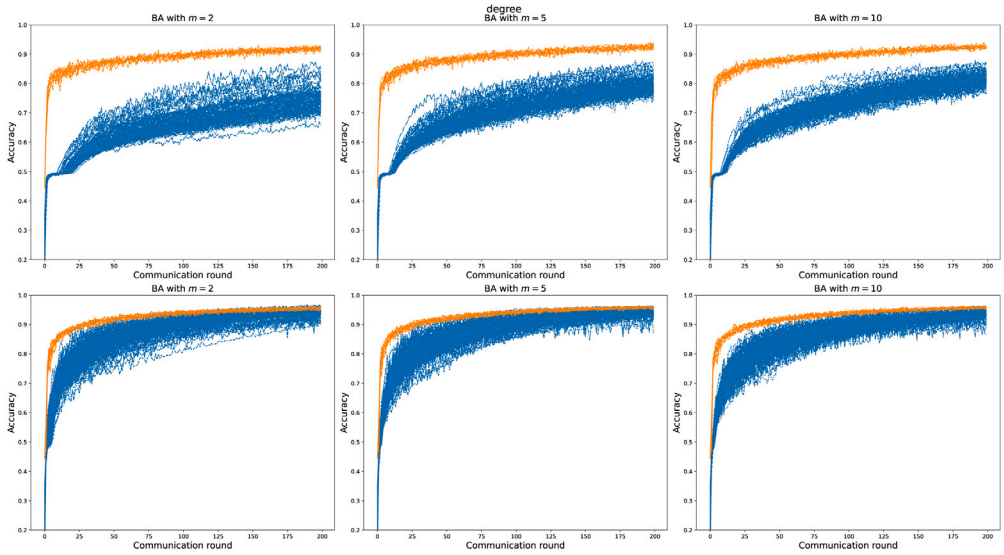


Fig. 20. MNIST – Accuracy in BA networks (G1 nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right, the parameter m increases; from top to bottom: edge-focused and hub-focused scenario. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

better average accuracy (some hubs get worse, but other G1 nodes get better) and reduced variability (Fig. 24).

When examining the results for EMNIST (see Fig. 21) and Fashion-MNIST (see Fig. 22), the same trends are observed, with the only difference being the absolute values of accuracy due to the varying classification difficulties of the datasets. To avoid overcrowding the body of the paper, further discussion on these two datasets will not be provided in the following sections, but their plots can be found in the appendices.

6.2.2. BA: Betweenness centrality used for G2 data assignment

Since the betweenness centrality is proportional to the degree (Fig. 18), we expect the G2 data assignment based on betweenness centrality to perform similarly to its degree-based counterpart. In Fig. 25, in the lowest focus case (top row), we see an increasingly narrow curve meaning that as we increase the connectedness, the knowledge is more easily spread and this results in less variation in the performances. The overall behavior is almost identical to the degree-based one, with almost no difference in the performances in the highest

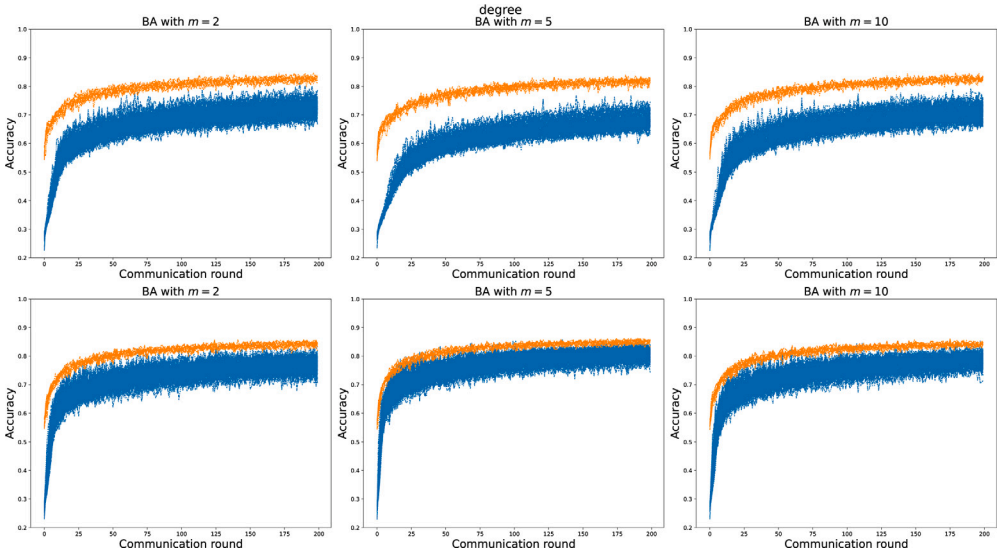


Fig. 21. EMNIST – Accuracy in BA networks (G1 nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right, the parameter m increases; from top to bottom: edge-focused and hub-focused scenario.

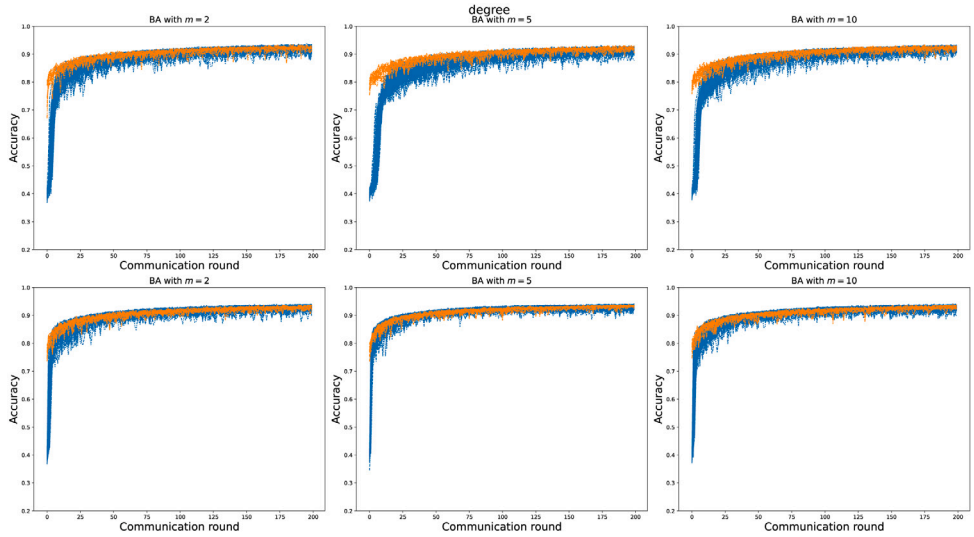


Fig. 22. Fashion-MNIST – Accuracy in BA networks (G1 nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right, the parameter m increases; from top to bottom: edge-focused and hub-focused scenario.

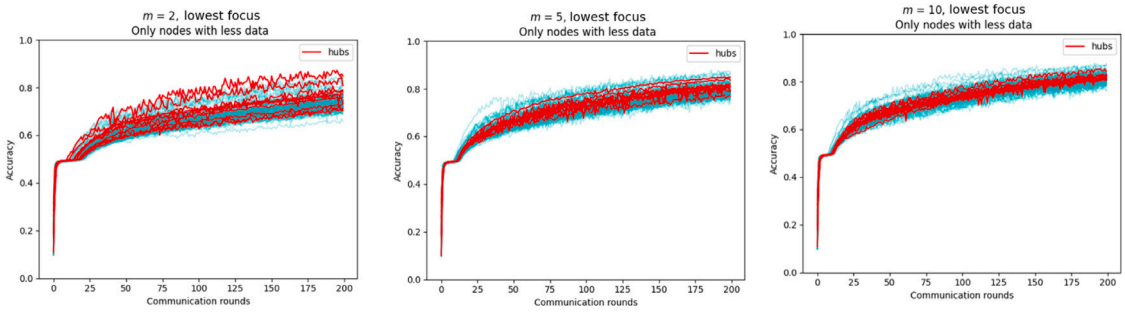


Fig. 23. MNIST – Accuracy in BA networks (nodes with only G1 images, edge-focused scenario, G2 data assigned based on degree centrality), high-degree nodes in red.

focus case. In Fig. 26, we show the mean accuracy and the standard deviation for each network and case. Again, the same behavior as in the degree-based case is observed.

6.2.3. BA: Clustering coefficient used for G2 data assignment

Looking at the individual curves of the accuracy over time in Fig. 27, we can see that the behavior is similar between the highest and lowest

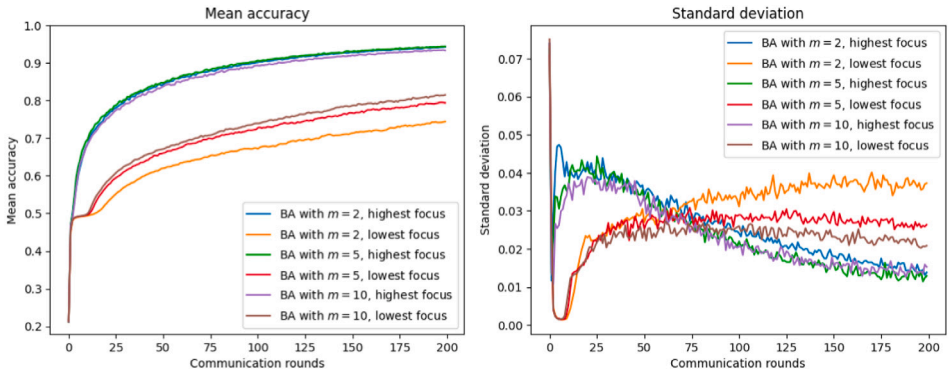


Fig. 24. MNIST – BA (G1 nodes, G2 data assigned based on degree centrality): accuracy mean and std.

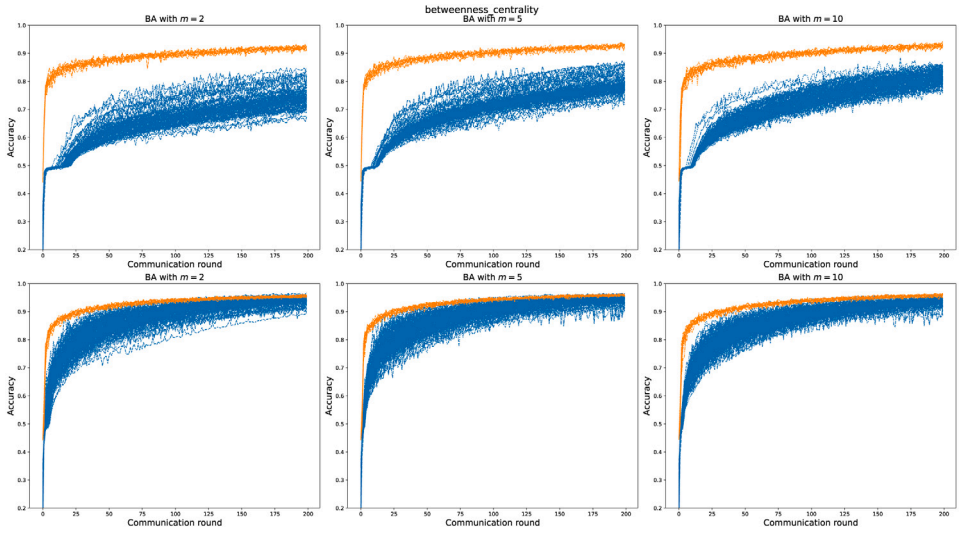


Fig. 25. MNIST – Accuracy in BA networks (all nodes, G2 data assigned based on betweenness centrality). G1 nodes in blue, G2 nodes in orange. From left to right, increasing values of m; from top to bottom: lowest-focused and highest-focused scenario.

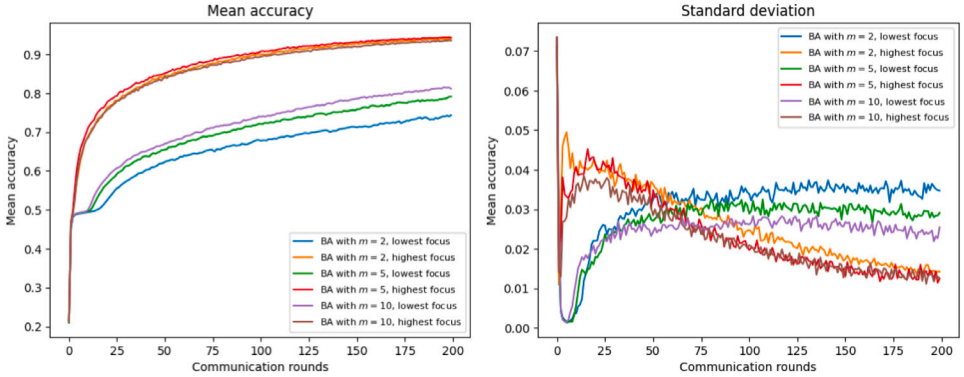


Fig. 26. MNIST – BA (G1 nodes), G2 data assigned based on betweenness centrality): accuracy mean and std for betweenness centrality.

focus cases, as a result of the negligible difference among the degrees of the selected nodes for G2 assignment (see Fig. 19). However, as expected from the scatterplot in Fig. 19, the lowest focus case shows a higher variability between the performances of the different nodes. As such, some nodes are able to reach higher values of accuracy (see case $m = 2$). The reasoning is the same as for the ER networks: the nodes with a high clustering coefficient tend to keep the information locally, whereas the ones with a low clustering coefficient spread it more broadly if they also have a high degree. This spreading results in higher variability. Figs. 28 and 29 simply confirm this trend. As for the ER, also in Fig. 29, the curves are closer together. Thus, we show the network realizations separately.

6.3. ER vs BA comparison

As explained in Section 5.1, we have chosen our ER and BA settings such that the corresponding graphs are idempotent. This allows us to compare directly and fairly the impact of the network topology on the learning process for the same value m. For clarity, we focus on the mean

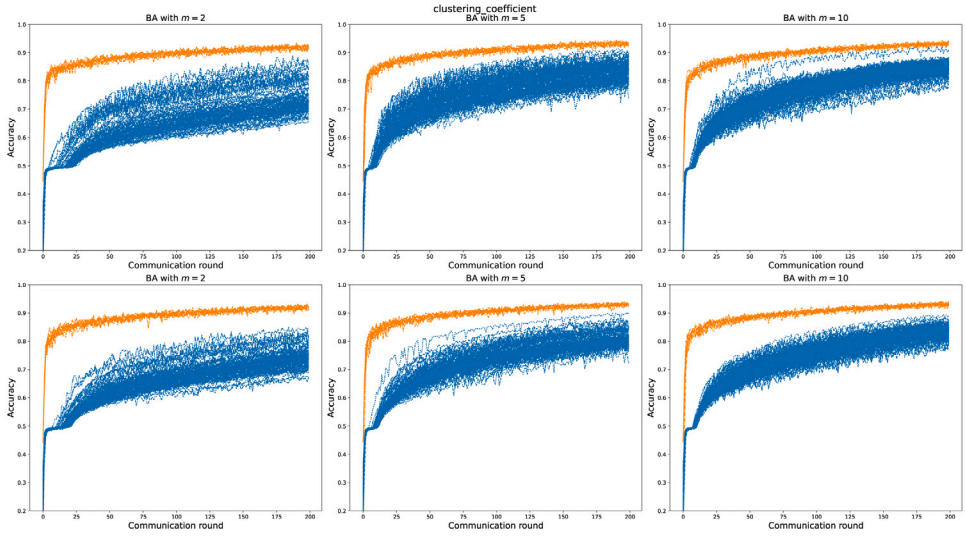


Fig. 27. MNIST – Accuracy in BA networks (all nodes, G2 data assigned based on clustering coefficient). G1 nodes in blue, G2 nodes in orange. From left to right, increasing values of m; from top to bottom: lowest-focused and highest-focused scenario.

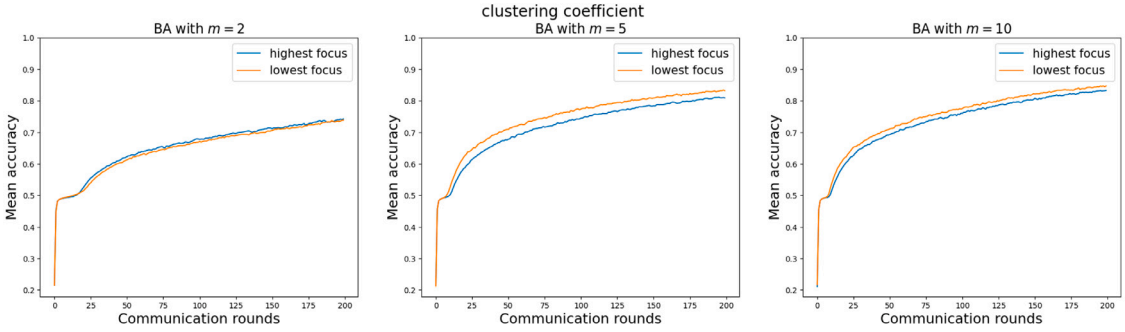


Fig. 28. MNIST – Accuracy in BA networks (average across all nodes, G2 data assigned based on clustering coefficient). From left to right, increasing values of m; from top to bottom: lowest-focused and highest-focused scenario.

Fig. 29. MNIST – BA (G1 nodes): standard deviation, G2 data assigned based on clustering coefficient.

accuracy in the MNIST case, so that the plot is not overclutted with curves and a direct comparison is more immediate.

6.3.1. Degree centrality

In the hub-focused case, as shown in Fig. 30, the Barabasi Albert performs better than the ER counterpart. We can see that the difference decreases as we increase the number of connections in the graphs (i.e. going from left to right). This is easily explained by looking at the characteristics of the topology of the two types of graphs. The Barabási-Albert graph, in fact, is a graph that exhibits a heavy tail in the degree distribution, meaning that it has few nodes with a high degree and many nodes with a low degree. This means that the high-degree nodes are well connected in the graph and can exert their influence much more efficiently than the Erdős-Rényi counterpart. On the other hand, the Erdős-Rényi graph is a random graph. Thus the nodes with a high degree are not necessarily well connected inside the graph. However, this difference gets smaller as we increase the mean number of connections: all nodes, in general, enjoy better connectivity, and the network can compensate for the lack of strong hubs.

The edge-focused case is more diverse. Here, BA performs better than the ER counterpart only for $m = 2$. In the other two cases, ER is the one with better performance, as reported in Fig. 31. This can

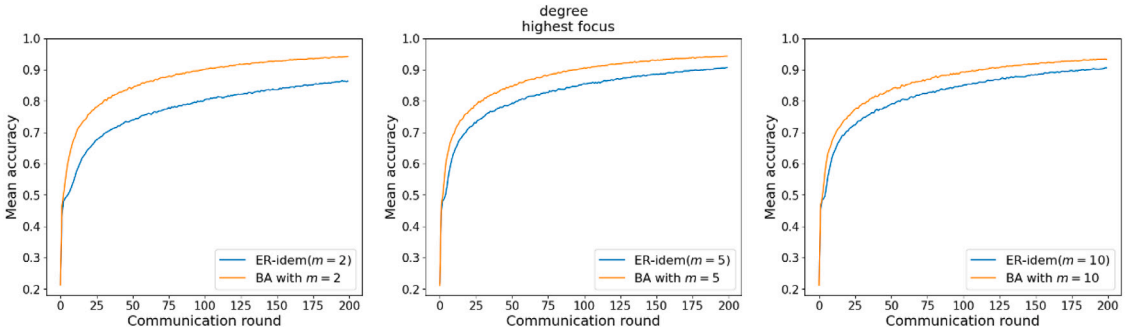


Fig. 30. MNIST – Mean accuracy of the Barabasi Albert (orange line) and the Erdős-Rényi counterpart (blue line) in the hub-focused case. Going from left to right: $m = 2, 5, 10$. G2 data are assigned based on the degree centrality.

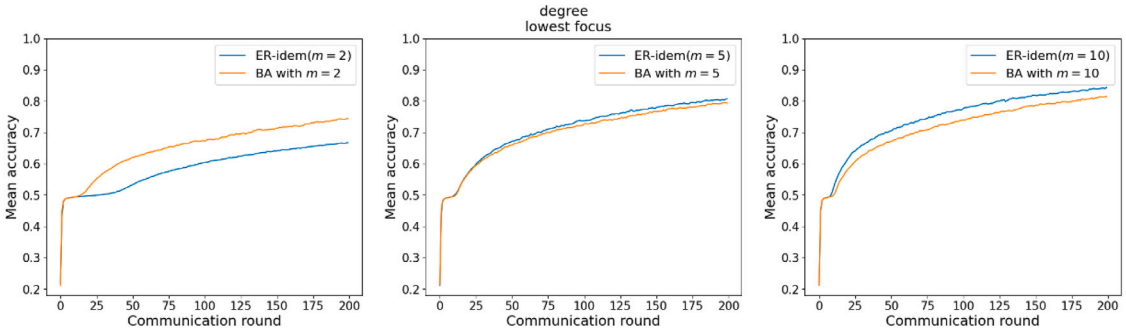


Fig. 31. MNIST – Mean accuracy of the Barabasi Albert (orange line) and the Erdős-Rényi counterpart (blue line) in the edge-focused case. Going from left to right: $m = 2, 5, 10$. G2 data are assigned based on the degree centrality.

Table 4
Table showing the algebraic connectivity values for the BA graphs and their ER counterpart for different values of m.

$m = 2$		$m = 5$		$m = 10$	
BA	ER	BA	ER	BA	ER
0.612	0.395	2.989	2.998	4.718	8.143

be confirmed by looking at the algebraic connectivity of the graphs (Table 4). Algebraic connectivity is a measure of the connectedness of a graph. A graph with high algebraic connectivity is well-connected and can quickly spread information, while a graph with low algebraic connectivity can only spread information slowly. The higher the algebraic connectivity, the higher the number of paths for information to travel through the graph. As shown in Table 4, the algebraic connectivity of the $m = 2$ case is lower for the Erdős-Rényi with respect to the Barabási-Albert.

6.3.2. Betweenness centrality

As discussed in the previous sections, the betweenness centrality shows a proportional relationship to the degree for both the ER and the BA networks. Therefore, we expect to have similar results for the betweenness centrality. Similarly to the degree case, in the lowest focus, the BA performs better than the ER counterpart only for $m = 2$, as shown in Fig. 32. This means that comparing these results with Fig. 31, G2 nodes are not able to sufficiently influence the hubs.

6.3.3. Clustering coefficient

Similarly as above, also the clustering coefficient shows for the ER-idem($m = 2$) a worse performance than the BA, see Fig. 33. Again, this is due to the algebraic connectivity that helps the knowledge spreading, as discussed for Fig. 32 and earlier.

6.4. Decentralized learning over Stochastic Block Model graphs

The SBM topology is different from the previous two as it features four clearly separated communities, with sporadic intercommunity links. For the intracommunity connectivity, we test two scenarios (lower and higher intracommunity connectivity, corresponding to $p_{ii} = 0.5$ and $p_{ii} = 0.8$). Recall that each community holds two non-overlapping MNIST classes (hence, classes 8 and 9 are discarded). Using only intracommunity information, nodes can, at most, achieve a 0.25 accuracy (perfect classification of the two classes in their training data, zero knowledge on the other six). In order to go beyond 0.25, knowledge must be circulated across communities. In these settings we are interested in analyzing (i) whether the sporadic edges between communities are sufficient for exchanging the knowledge built intracommunity and (ii) to what extent the external knowledge permeates through the communities. Fig. 34 shows that the latter is happening: with $p_{ii} = 0.5$ (less dense communities, blue lines) the average accuracy grows faster than with $p_{ii} = 0.8$. The figure also reveals the presence of stragglers, for whom catching up with the rest of the network takes some time. Interestingly, entire communities appear to be stragglers. Fig. 35 shows that communities are, as expected, very good at classifying classes they have in their local training data. Vice versa, it is very hard for external knowledge to enter the communities. Fig. 35 also shows the number of links pointing towards external communities, which are the conduit for knowledge diffusion. Community C_2 enjoys fewer external links, and indeed its learning process is very slow and mediocre (Fig. 34). However, Community C_1 has only slightly fewer

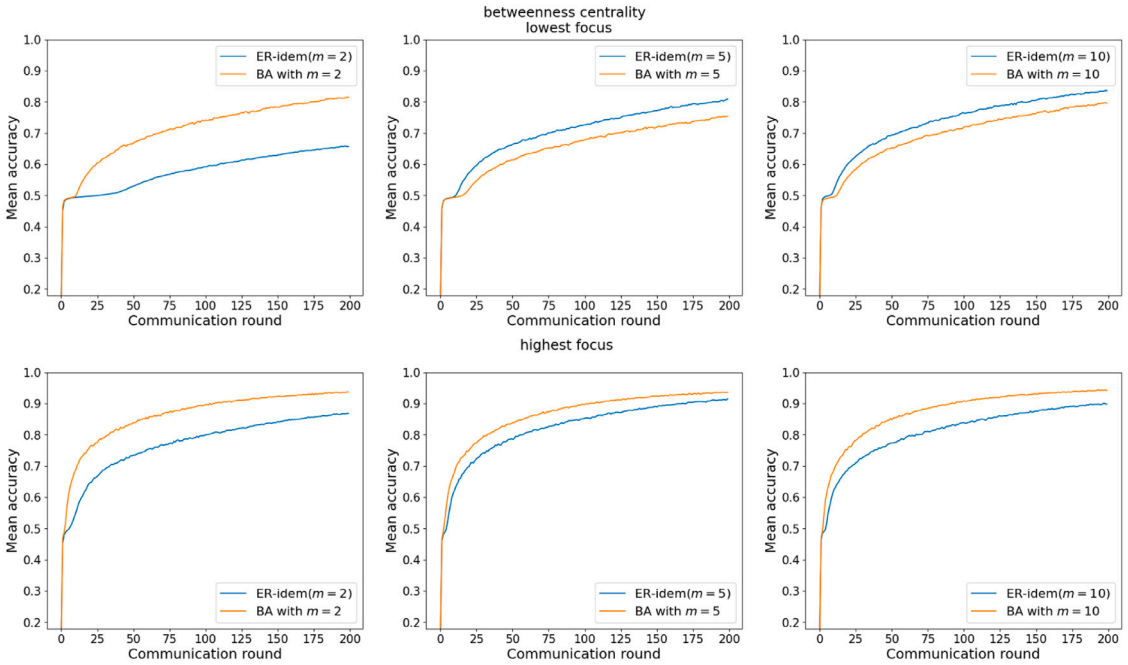


Fig. 32. MNIST – Comparison of mean accuracy ER vs BA. Top to bottom: lowest-focus and highest-focus cases. Left to right: increasing m. G2 data is assigned based on betweenness centrality.

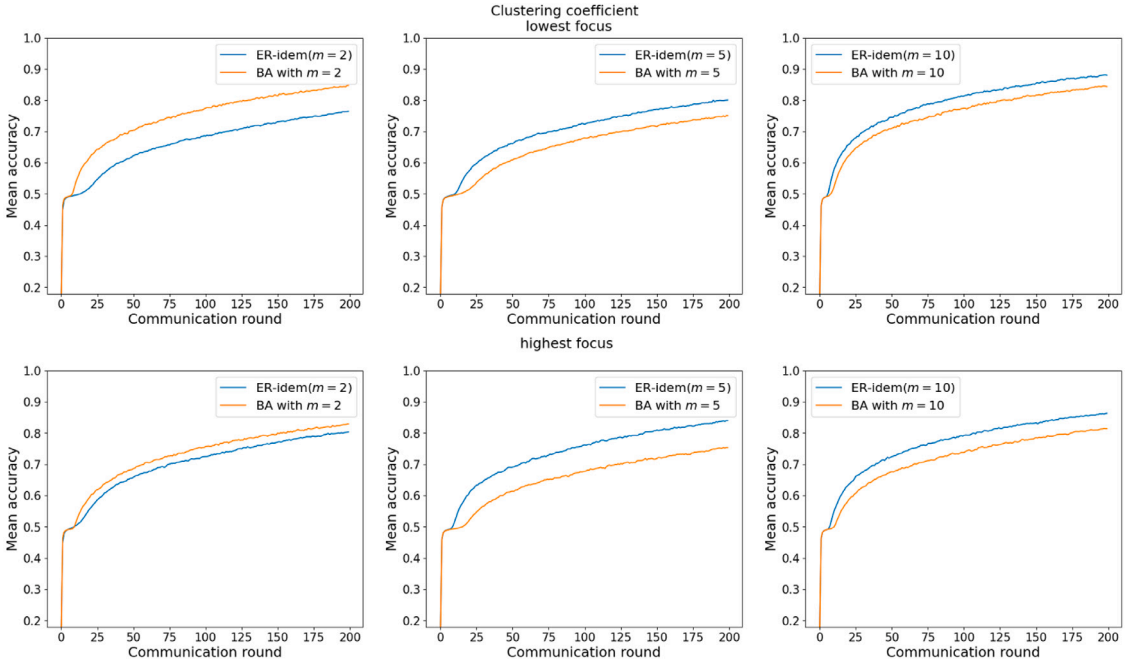


Fig. 33. MNIST – Comparison of mean accuracy ER vs BA. Left to right: increasing m. Top to bottom: lowest-focus and highest-focus. G2 data assigned based on clustering coefficient.

links than Community C_3, but its learning process is faster and more accurate. This observation suggests that the specific class distribution assigned to each community might play a crucial role in determining its learning performance.

To validate the above hypothesis, we swapped the datasets among the four communities as reported in Table 5. If the learning performance is highly dependent on the dataset, one would expect that when the dataset of C_1 is moved to C_2, the improved learning speed and accuracy would move with it. Conversely, if the characteristics of the community itself play a significant role, one might observe

Table 5
Data swap between communities.

Source → Destination
$C_1 \rightarrow C_2$
$C_2 \rightarrow C_3$
$C_3 \rightarrow C_4$
$C_4 \rightarrow C_1$

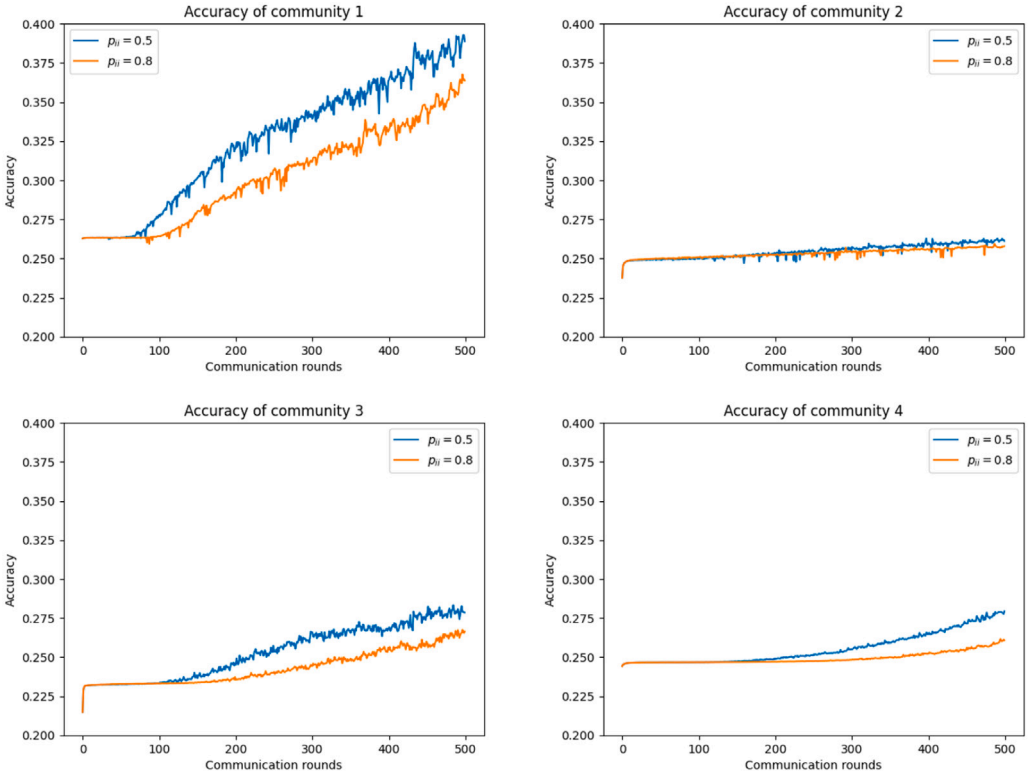


Fig. 34. MNIST – Mean accuracy over time in SBM communities.

different results. Our findings, shown in Fig. 36, confirmed this assumption, demonstrating that the specific class distribution significantly influences the learning behavior of each community.

Furthermore, we explored the impact of conducting multiple local training epochs on the dissemination of information within the community. The objective was to determine whether a high number of local training epochs in between global updates might negatively impact information diffusion when communities are tightly knitted. To examine this, we conducted a simulation under identical conditions as in the original experiment but with significantly fewer local training epochs per communication round: specifically, we reduced it to 5 local training epochs as opposed to the original 100. The results, presented in Fig. 37, reveal a pattern consistent with the original findings. This suggests that the challenge of information propagation within the community is not inherent to the local training phase but is primarily influenced by the structural properties of the network as well as by the correlation between data classes.

6.5. Decentralized learning with cross-correlations in the data

As anticipated in Section 5.2, in our decentralized configuration, we observed an important role of cross-correlations among classes on the performance of DecAvg. As previously noted, when class Shirt (belonging to group G2 based on our data split) is absent from both the training and test sets, nodes in group G2 achieve higher accuracy levels than G1 due to their greater data availability. This outcome is expected because G2 nodes possess more data overall. Interestingly, though, when class Shirt is included, G2 nodes consistently exhibit lower accuracy levels than G1 nodes. This pattern persists across all scenarios, irrespective of whether G2 nodes have the highest or lowest centrality values, indicating that the issue is related to data distribution rather than network topology. To investigate this further, we conducted simulations using the BA model, in the highest-focus case, using the clustering coefficient for selecting the top nodes. We compared three different data distributions: one without the class Shirt in G2, a balanced distribution where all images (both G1 and G2, and including class Shirt) are assigned 60 instances per class on each node, and an unbalanced distribution where G1 nodes are assigned ∼60 vs ∼600 images per class (including the Shirt class) on each node, respectively. This comparison aimed to assess how data distribution affects performance. In Fig. 38, we show that in the balanced data distribution (bottom row), the accuracy over time aligns with the expected behavior observed when class Shirt is absent (shown in Fig. 22), with G2 nodes performing consistently better the G1 ones.

The behavior described above can be further investigated by looking at the confusion matrices on each node towards the end of the decentralized training process. Specifically, in Figs. 39 and 40, we present the confusion matrices for two selected G1 and G2 nodes across the three cases: without class Shirt, balanced distribution (when class Shirt is present but not overrepresented with respect to the images in G1, which include the cross-correlated classes), and unbalanced distribution (in which class Shirt is overrepresented). In the unbalanced case (the right-most plot in the figures), the G2 node consistently misclassifies classes correlated with class Shirt (corresponding to rows 0, 2, and 4). Whereas the G1 node fails to learn class Shirt but maintains high accuracy for other classes, in the balanced case (middle column), both G1 and G2 nodes exhibit similar behavior, resulting in improved overall performance for G2 nodes (which do not misclassify as badly G1 images), though G2 nodes still misclassify class 6. The improved performance of G2 nodes is partially paid for by G1 nodes failing to classify the Shirt class. As a benchmark, note that the classification accuracy is very high on both G1 and G2 nodes when the offending class Shirt is dropped (left-most column in Figs. 39 and 40).

A reasonable explanation would be that this is due to the combined effects of local training and model aggregation function (DecAvg). During local training, each node updates its model using its local

Class	C_1 (-, 5,9,7)	C_2 (5,-,7,3)	C_3 (9,7,-,8)	C_4 (7,3,8,-)
0	**0.9961**	0.0002	0.0004	0.0043
1	**0.9992**	4e-05	0.0684	0.0079
2	0.0	**0.9868**	4e-05	0
3	0.0146	**0.9802**	0	0.0002
4	0.1824	0.0076	**0.9971**	0.0006
5	0.0011	4e-05	**0.9972**	0.0011
6	0.0039	0	0.0003	**0.9979**
7	0.272	0.0101	0.0225	**0.9966**

Fig. 35. Accuracy per MNIST class and community for SBM with $p_{ii} = 0.5$. For each community, we report in square brackets the number of edges pointing toward external community 1, 2, 3, 4, respectively.

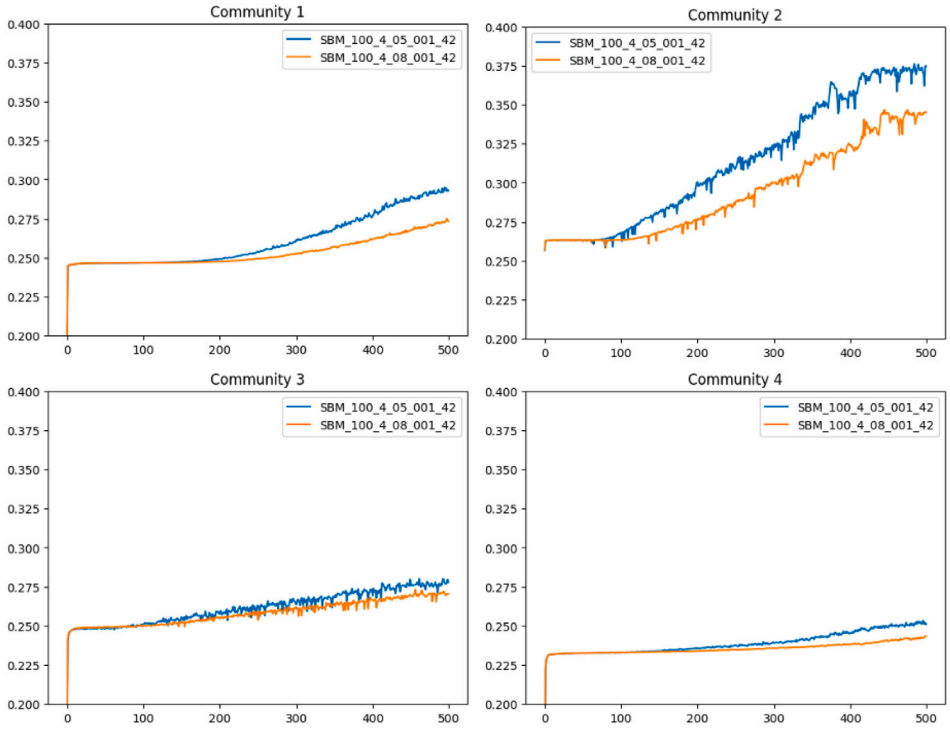


Fig. 36. MNIST – Mean accuracy over time in SBM communities, with swapped datasets.

dataset. In unbalanced data scenarios (G1 60 images/class vs G2 600 images/class), a high number of instances of a correlated class, not commonly found in other nodes, causes the locally updated model on G2 nodes to deviate significantly from those at G1 nodes. Conversely, in a balanced data distribution, each local model remains closer to the collective average model. In other words, on G2 nodes, when the offending class is overrepresented compared to others that are cross-correlated with it, G2 models classify all cross-correlated G1 classes as if they were the offending one. Vice versa, when the offending class is as represented as any other, its classification is skewed, but the others are properly identified. Consequently, in a balanced setting, local models struggle to classify instances of class Shirt accurately but perform better in other classes. Note that, in these cases, having more data samples may not be better than having fewer data samples for the cross-correlated classes. Looking at G1 nodes, they are slightly better off in the unbalanced case, as they are able to pick up the strong signal about class Shirt (which they do not possess locally) in G2 models, but the training on their local dataset (which is balanced) avoids the drift. To conclude, *unbalanced data split in decentralized learning scenarios may amplify cross-correlation effects among classes. In such cases, having more data samples may not necessarily be better than having fewer data samples for the cross-correlated classes. While these effects are not significantly topology-dependent, they can radically impact the performance of decentralized learning.*

7. Conclusion

Fully decentralized learning is becoming increasingly interesting from a research perspective, as it enables the training of AI models at the edge of the Internet. This approach helps alleviate issues related to data and bandwidth burdens on the infrastructure, latency, the necessity for continuous and reliable Internet connectivity, and also addresses principles of data locality and privacy. However, learning in a

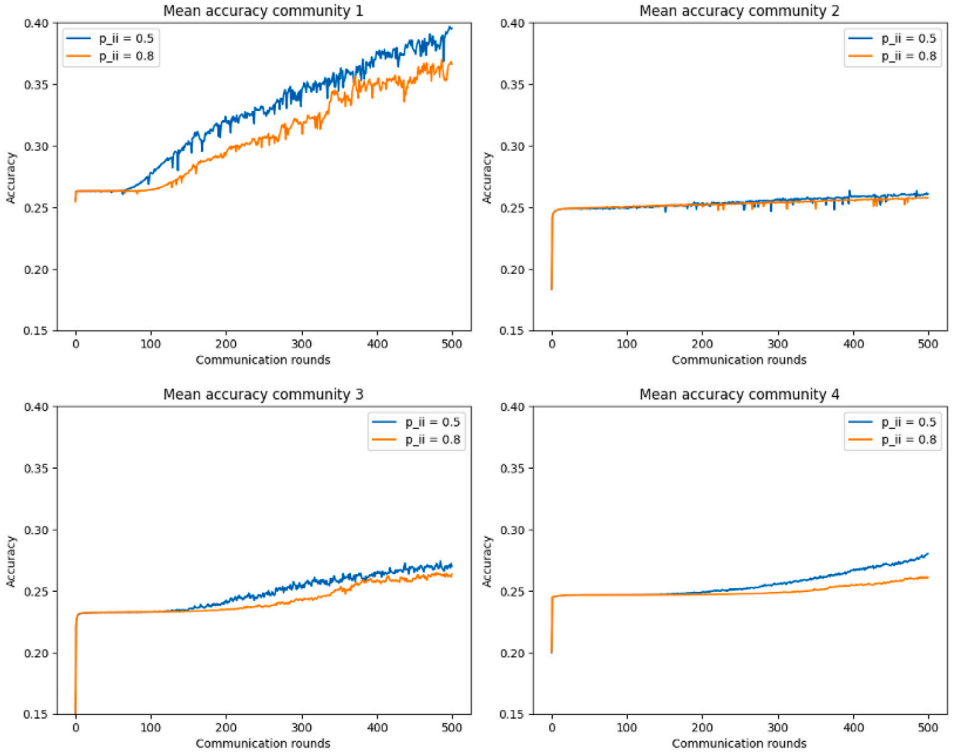


Fig. 37. MNIST – Mean accuracy in SBM communities, with 5 local training epochs instead of 100.

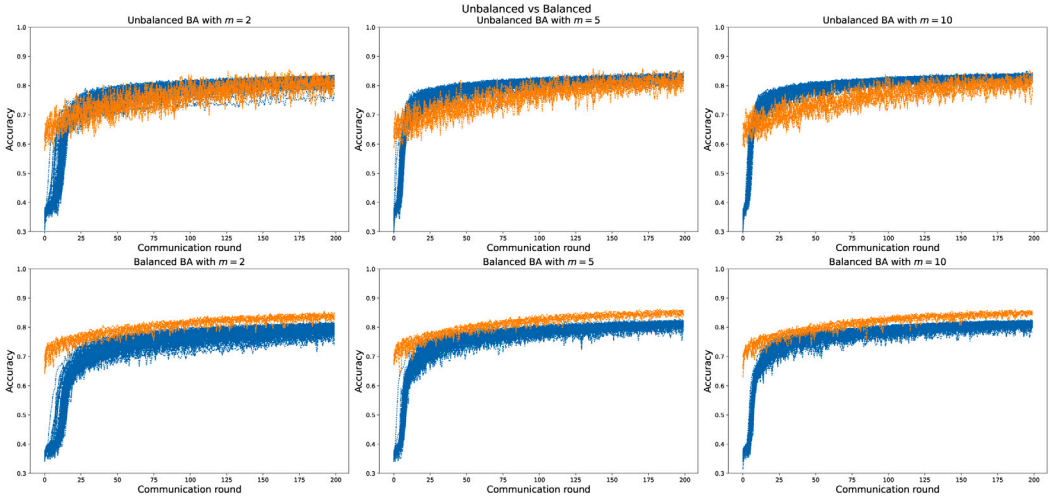


Fig. 38. Fashion-MNIST – Accuracy over time for BA network with G2 data assigned based on the clustering coefficient highest-focused. G1 nodes in blue, G2 nodes in orange. From top to bottom: unbalanced and balanced case.

fully decentralized manner poses significant challenges that researchers have yet to fully address. In this work, we focus on one of these challenges, specifically the interplay between the network structure, on which the learning process evolves, and the final performance of the learning itself. For this purpose, we selected three network topologies, each representing a dominant topological property, and six different methods of distributing training data among nodes. Below, we summarize the main findings of this work.

Centrality vs neighborhood density. Global centrality metrics directly correlate with the achieved performance of decentralized learning. No significant difference between the two global centrality metrics considered (degree and betweenness) is detected for the topologies we have considered. Vice versa, local clustering is a poor predictor of performance, meaning that belonging to tightly vs loosely knit neighborhoods does not make any direct difference.

Dilution effect. Exporting knowledge to central nodes is difficult because they perform model aggregation over a large neighborhood, where a single "good" model may get diluted into the neighborhood average.

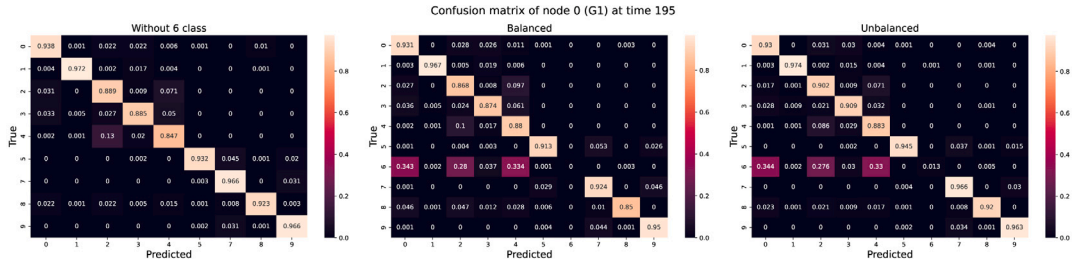


Fig. 39. Fashion-MNIST – Confusion matrix of a G1 node.

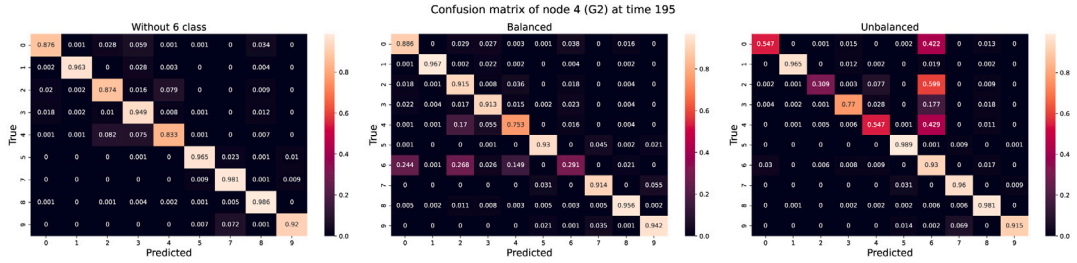


Fig. 40. Fashion-MNIST – Confusion matrix of a G2 node.

Centrality pull effect. Central nodes pull the others toward their knowledge (i.e., learned model). The centrality pull effect combined with the dilution effect explains why knowledge spreads easily if it originates on central nodes.

Degree distribution. The presence of hubs (originating from the Pareto distribution of the degree in BA networks) is a double-edged sword. Hubs can worsen the dilution effect when knowledge is on peripheral nodes but can boost the learning when knowledge is sourced by central nodes. Also, as a side effect of the dilution problem, the lower the average degree, the easier the knowledge spreading from non-central nodes.

Segregated communities. When users are grouped in segregated communities, it is very difficult for knowledge to circulate outside of the community.

Class cross-correlation Unbalanced data splits, common in decentralized settings, may amplify cross-correlation effects to the detriment of resulting accuracy. In such cases, having more data samples may not necessarily be better than having fewer data samples for the cross-correlated classes. The network topology does not seem to play a significant role in this situation.

The above summary illustrates the complex interplay between network topology, decentralized learning, and training data distribution. It conveys how a one-size-fits-all approach is not adequate for decentralized learning. This paper lays the groundwork for further studies, where more complex network topologies and network dynamics can be investigated. Moreover, it highlights the need for more nuanced model aggregation strategies, e.g., for mitigating the dilution effect and addressing the resulting unfair treatment of peripheral nodes.

CRediT authorship contribution statement

Luigi Palmieri: Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Chiara Boldrini:** Writing – review & editing, Visualization, Supervision, Methodology, Conceptualization. **Lorenzo Valerio:** Writing – review & editing, Visualization, Supervision, Methodology, Investigation, Conceptualization. **Andrea Passarella:** Writing – review & editing, Validation, Supervision, Investigation, Conceptualization. **Marco Conti:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by the H2020 HumaneAI Net (952026) and by the CHIST-ERA-19-XAI010 SAI projects. C. Boldrini's and M. Conti's work was partly funded by the PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR", A. Passarella's work was partly funded by the PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000001 - "RESTART", both funded by the European Commission under the NextGeneration EU programme.

Appendix A. EMNIST

We report here all the plots related to the analysis on the EMNIST-Letters dataset. For the convenience of the reader, we duplicate those plots that have also been included in the body of the paper (see Figs. A.41–A.54).

Appendix B. Fashion-MNIST

We report here all the plots related to the analysis on the Fashion-MNIST dataset. For the convenience of the reader, we duplicate those plots that have also been included in the body of the paper. Note that these results refer to a downsampled Fashion-MNIST dataset where the class Shirt was removed (see Figs. B.55, B.56, B.57, B.58, B.59, B.60, B.61, B.62, B.63, B.64, B.65, B.66, B.67 and B.68).

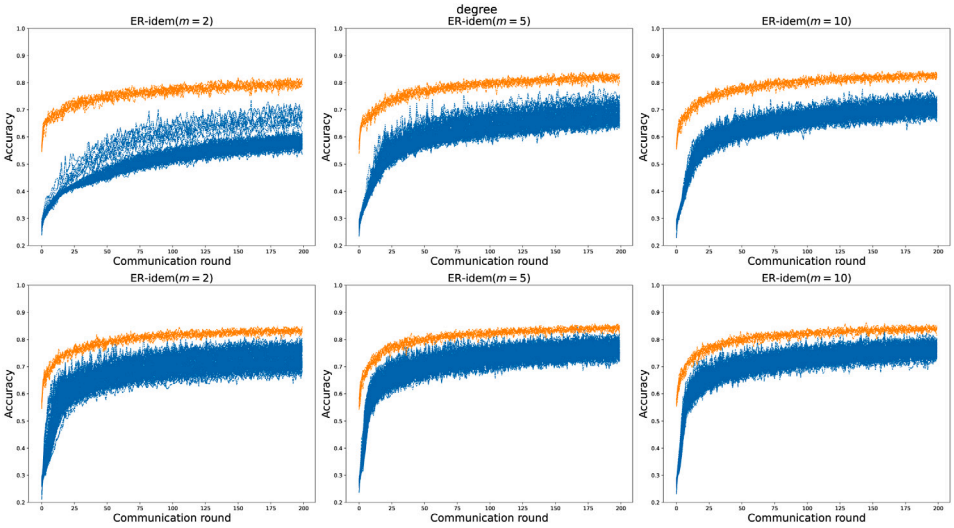


Fig. A.41. EMNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

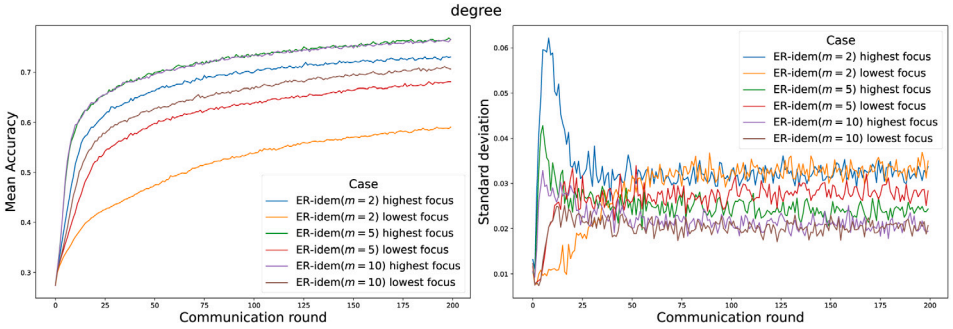


Fig. A.42. EMNIST – Mean accuracy over time and standard deviation in ER networks (average across G1 nodes only, G2 data assigned based on degree centrality).

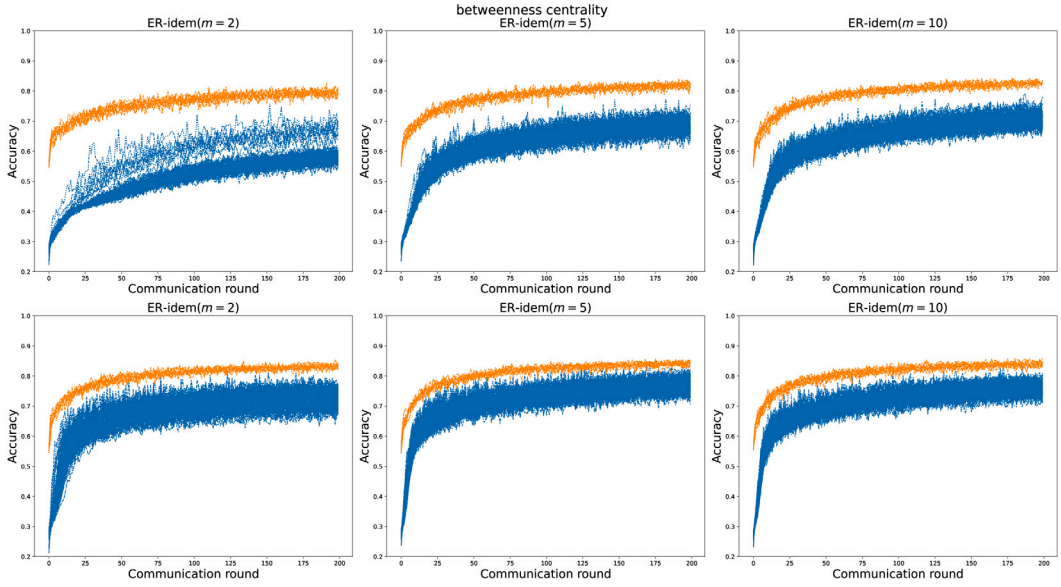


Fig. A.43. EMNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on betweenness centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

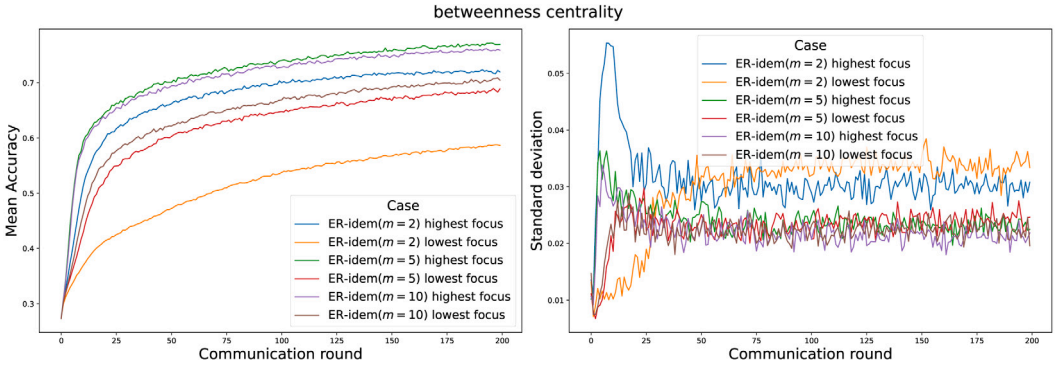


Fig. A.44. EMNIST – Mean accuracy over time in ER networks (average across G1 nodes only, G2 data assigned based on betweenness centrality). Highest-focused (blue) and lowest-focused (orange) cases.

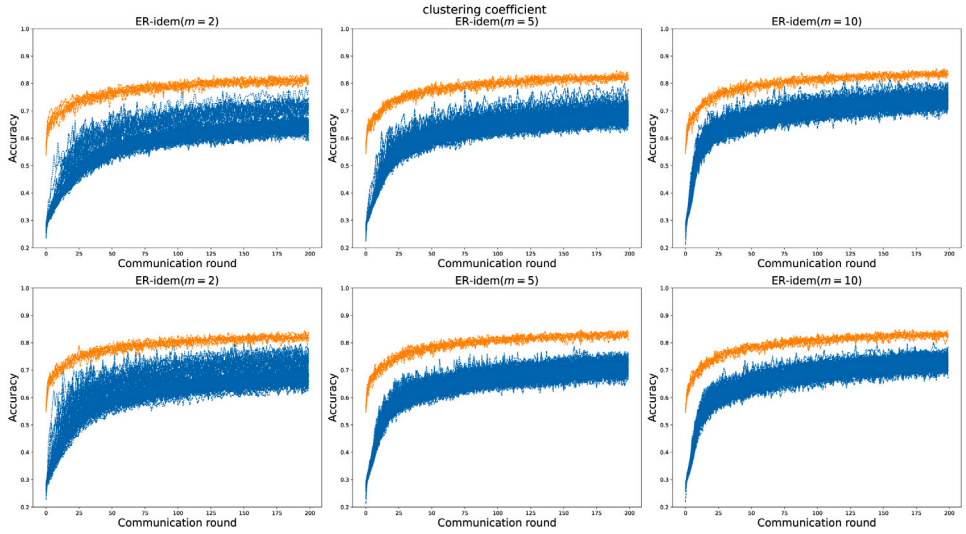


Fig. A.45. EMNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on clustering coefficient). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

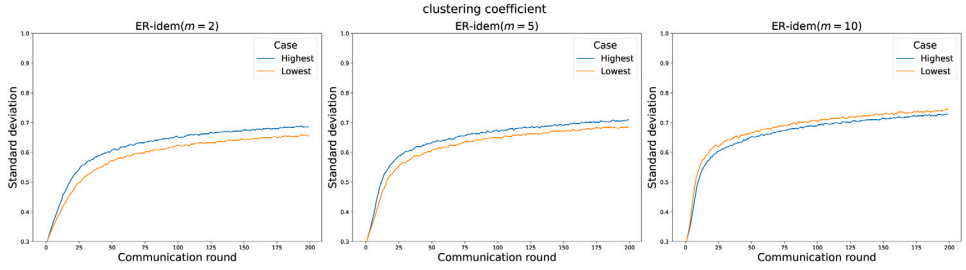


Fig. A.46. EMNIST – Mean accuracy over time in ER networks (average across G1 nodes only, G2 data assigned based on clustering coefficient).

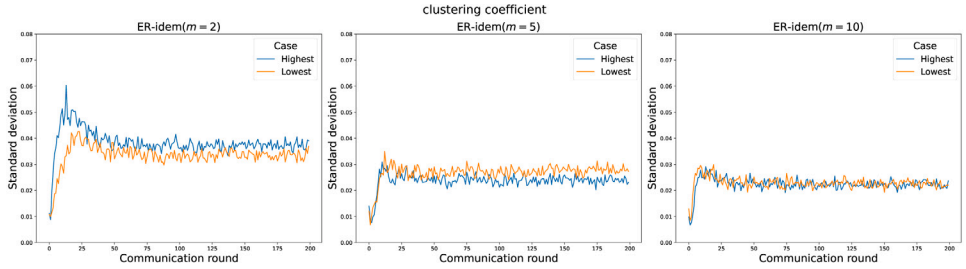


Fig. A.47. EMNIST – Standard deviation over time in ER networks (across G1 nodes only, G2 data assigned based on clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

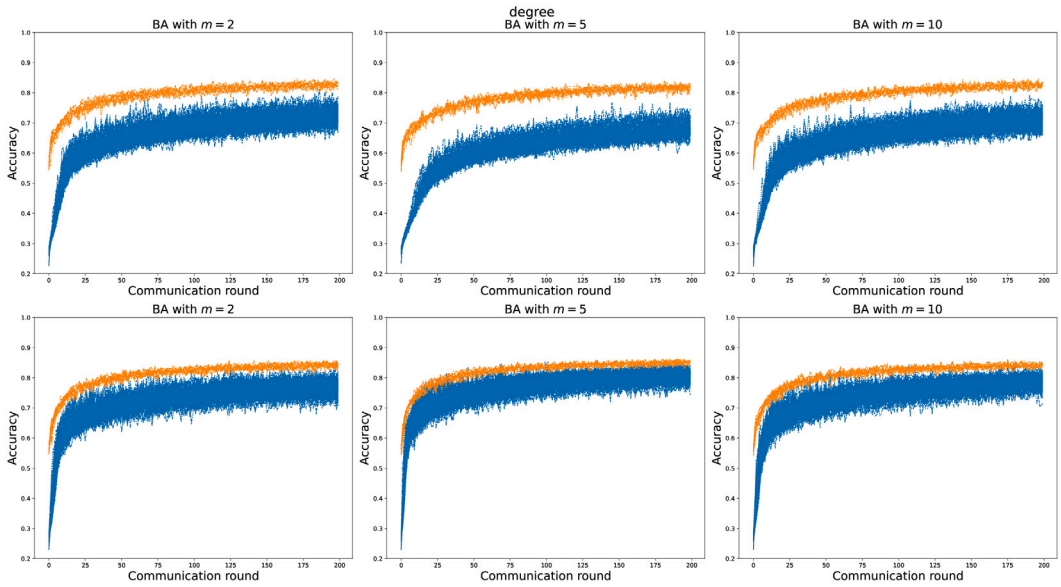


Fig. A.48. EMNIST – Accuracy over time in BA networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

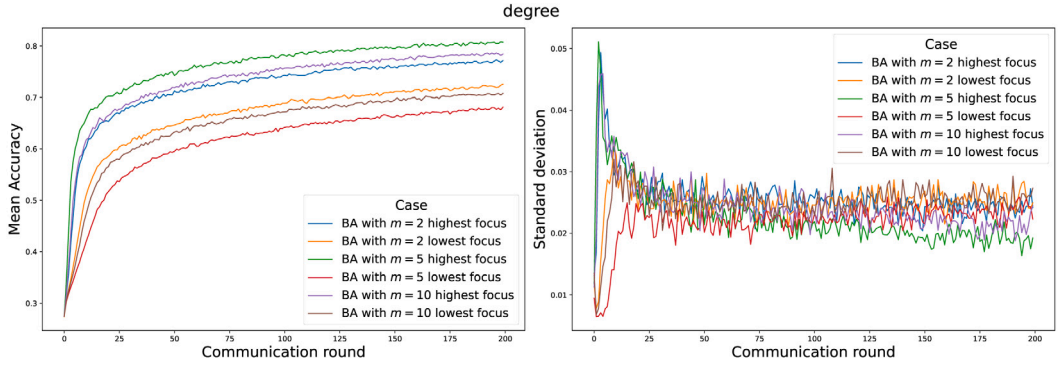


Fig. A.49. EMNIST – Mean accuracy over time and standard deviation in BA networks (average across G1 nodes only, G2 data assigned based on degree centrality).

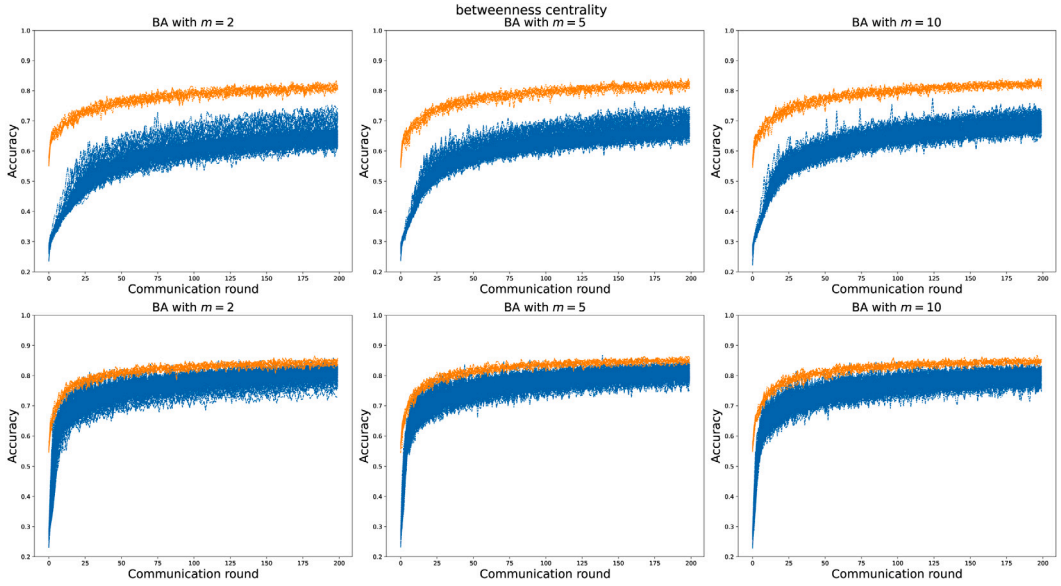


Fig. A.50. EMNIST – Accuracy over time in BA networks (all nodes, G2 data assigned based on betweenness centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

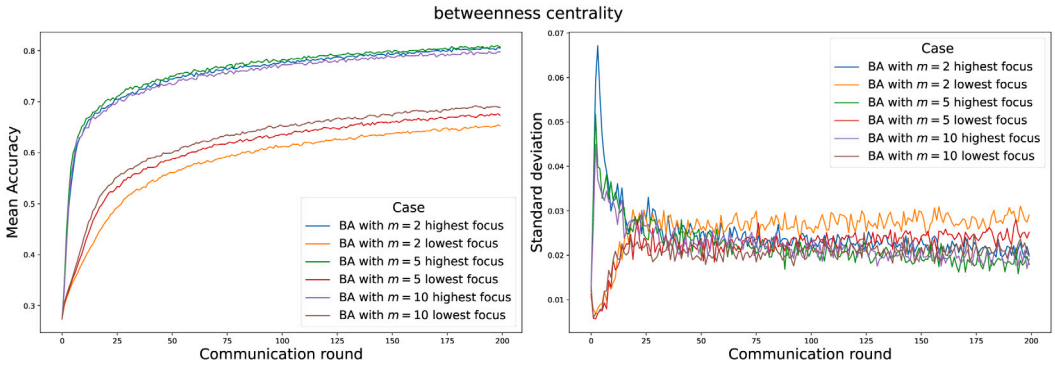


Fig. A.51. EMNIST – Mean accuracy over time and standard deviation in BA networks (average across G1 nodes only, G2 data assigned based on betweenness centrality). Highest-focused (blue) and lowest-focused (orange) cases.

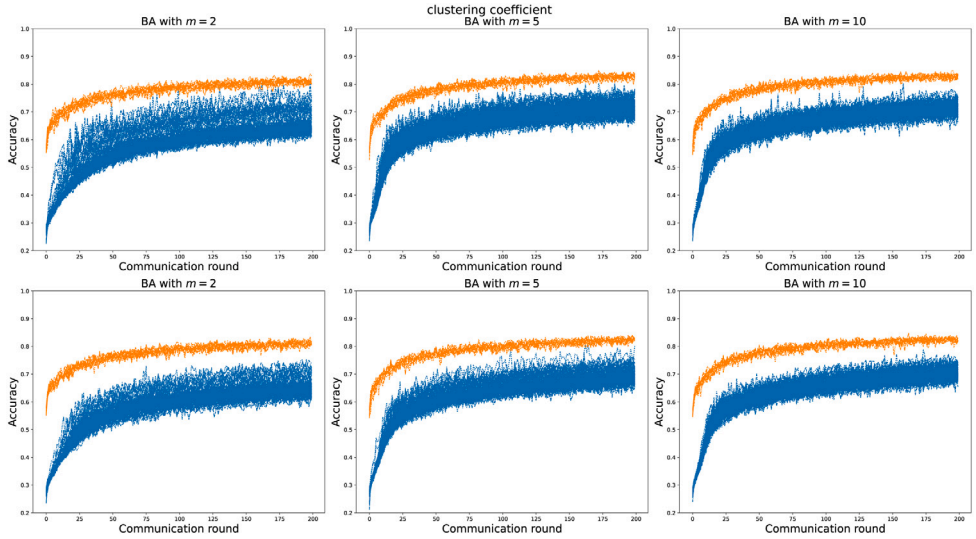


Fig. A.52. EMNIST – Accuracy over time in BA networks (all nodes, G2 data assigned based on clustering coefficient centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

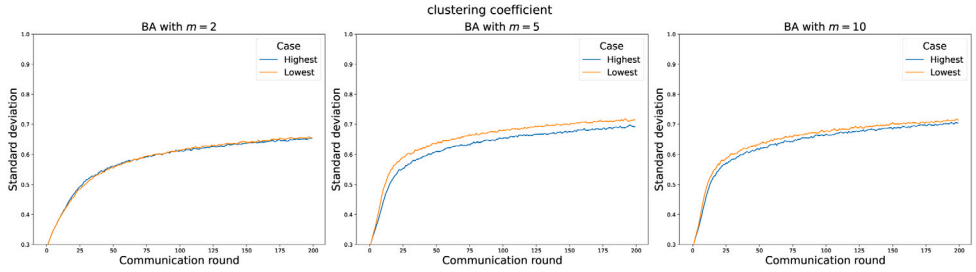


Fig. A.53. EMNIST – Mean accuracy over time in BA networks (average across G1 nodes only, G2 data assigned based on clustering coefficient).

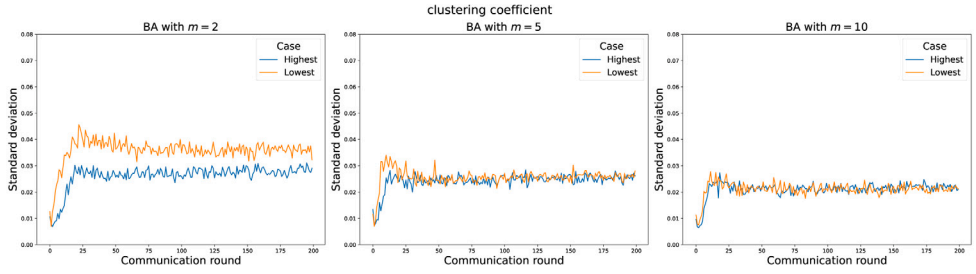


Fig. A.54. EMNIST – Standard deviation over time in BA networks (clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases.

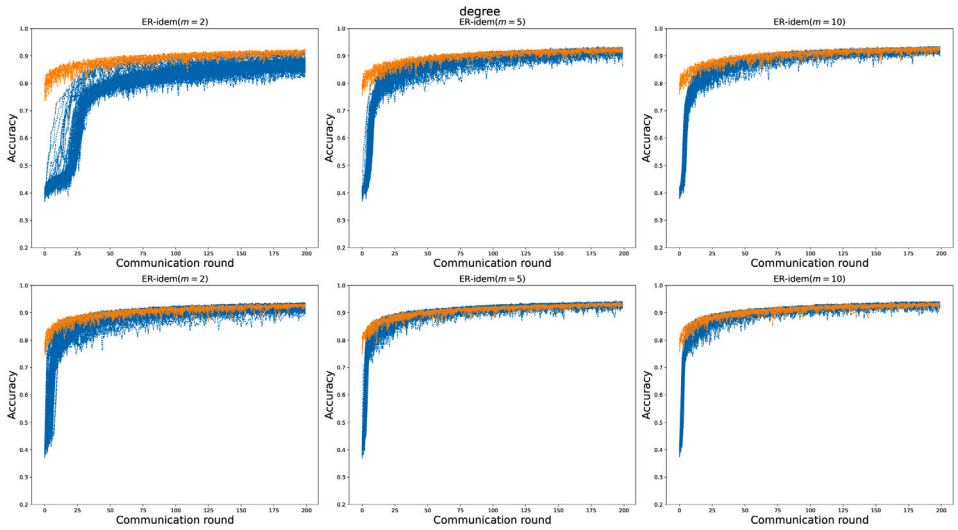


Fig. B.55. Fashion-MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

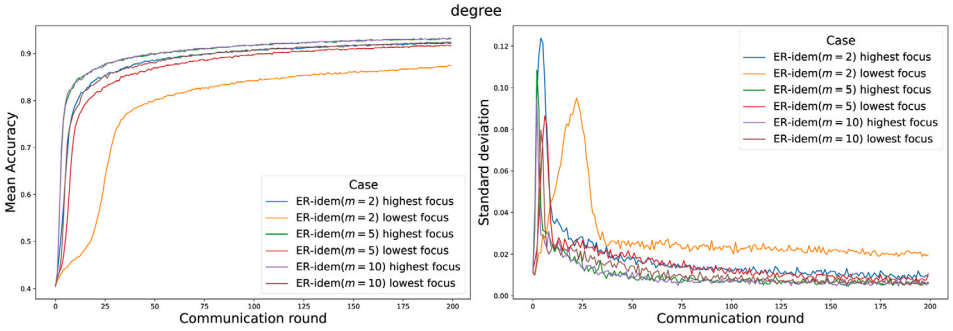


Fig. B.56. Fashion-MNIST – Mean accuracy and standard deviation over time in ER networks (average across G1 nodes only, G2 data assigned based on degree centrality).

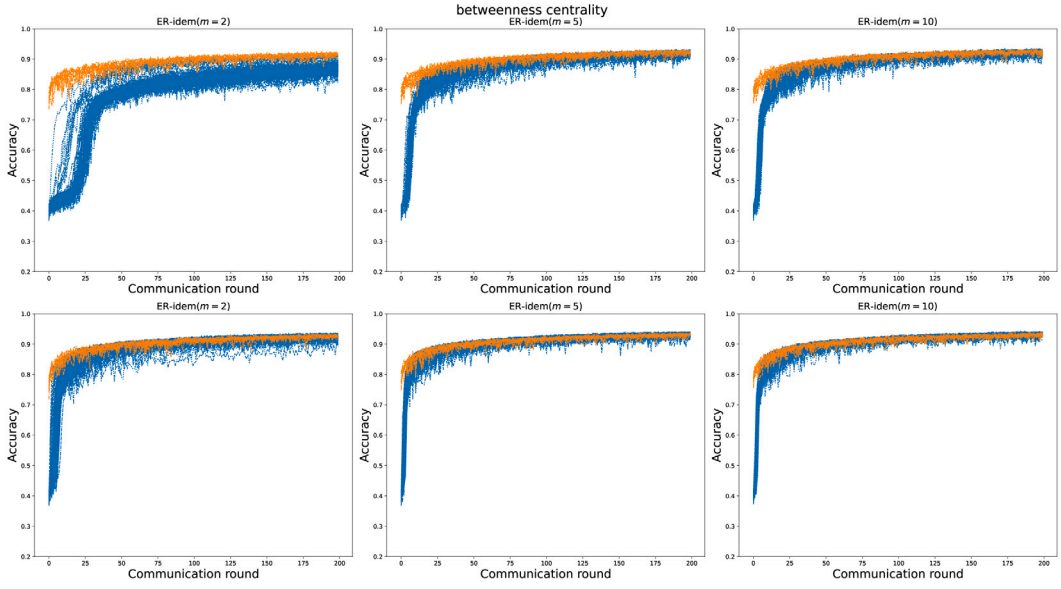


Fig. B.57. Fashion-MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on betweenness centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

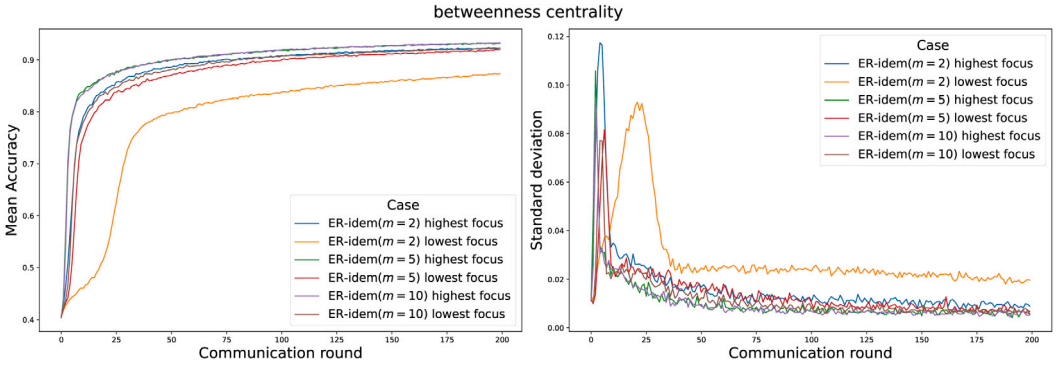


Fig. B.58. Fashion-MNIST – Mean accuracy and standard deviation over time in ER networks (average across G1 nodes only, G2 data assigned based on betweenness centrality).

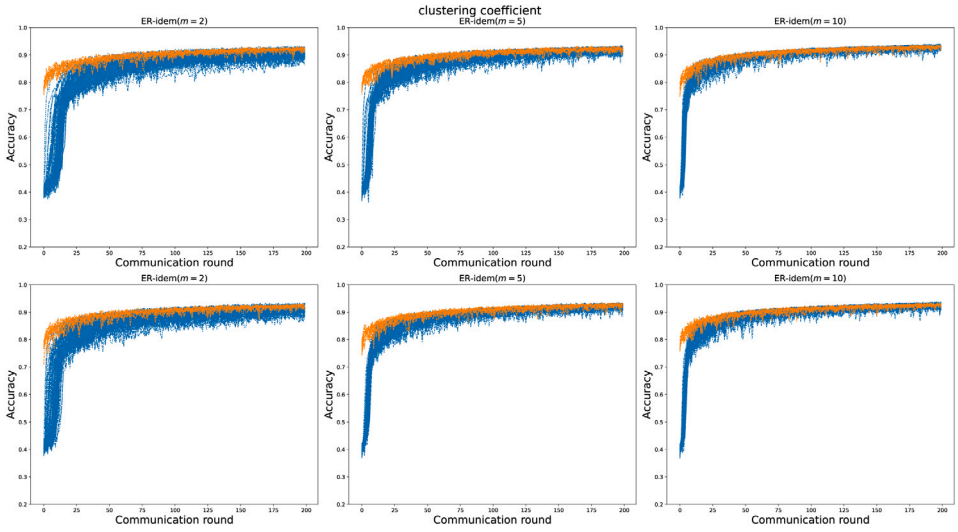


Fig. B.59. Fashion-MNIST – Accuracy over time in ER networks (all nodes, G2 data assigned based on clustering coefficient). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

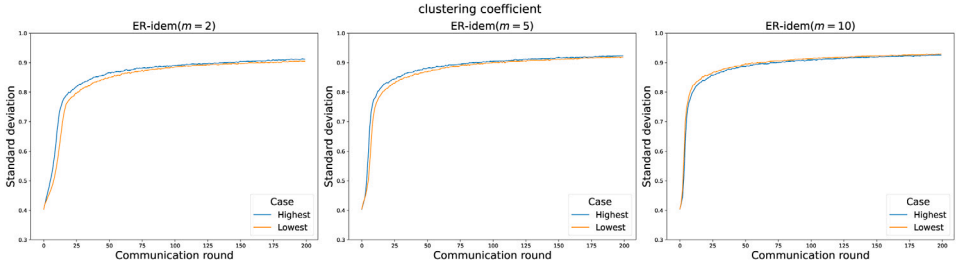


Fig. B.60. Fashion-MNIST – Mean accuracy over time in ER networks (average across G1 nodes only, G2 data assigned based on clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases.

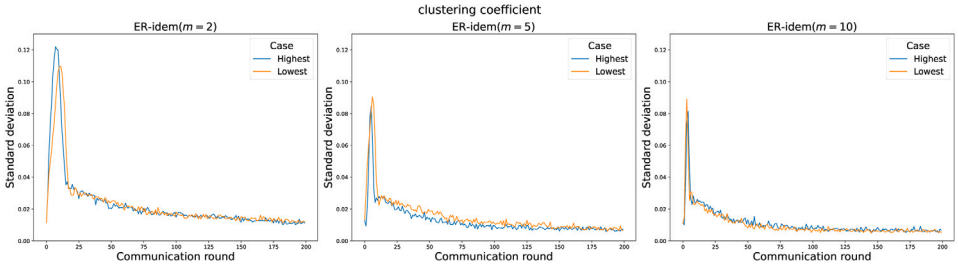


Fig. B.61. Fashion-MNIST – Standard deviation over time in ER networks (across G1 nodes only, G2 data assigned based on clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases.

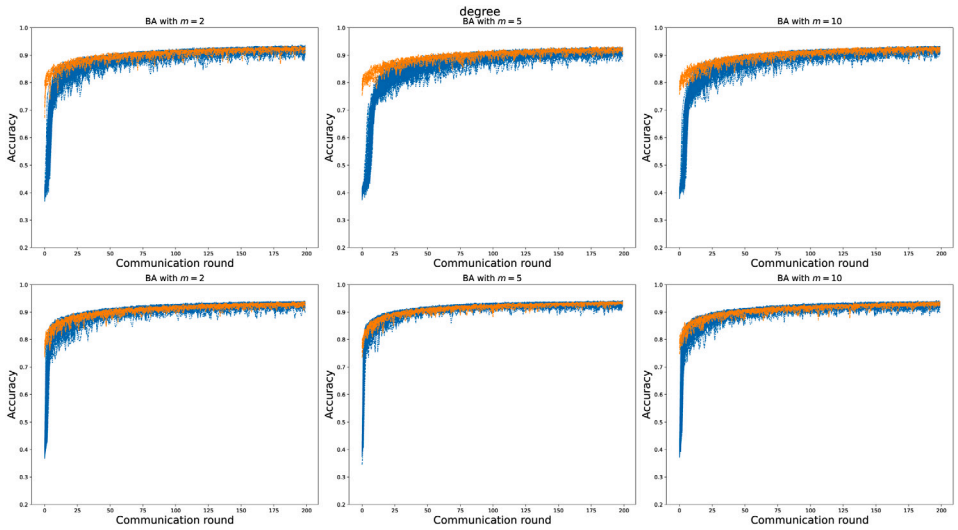


Fig. B.62. Fashion-MNIST – Accuracy over time in BA networks (all nodes, G2 data assigned based on degree centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

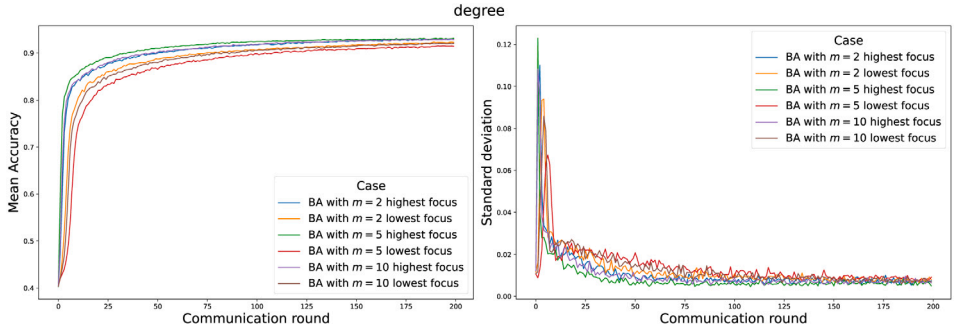


Fig. B.63. Fashion-MNIST – Mean accuracy and standard deviation over time in BA networks (average across G1 nodes only, G2 data assigned based on degree centrality).

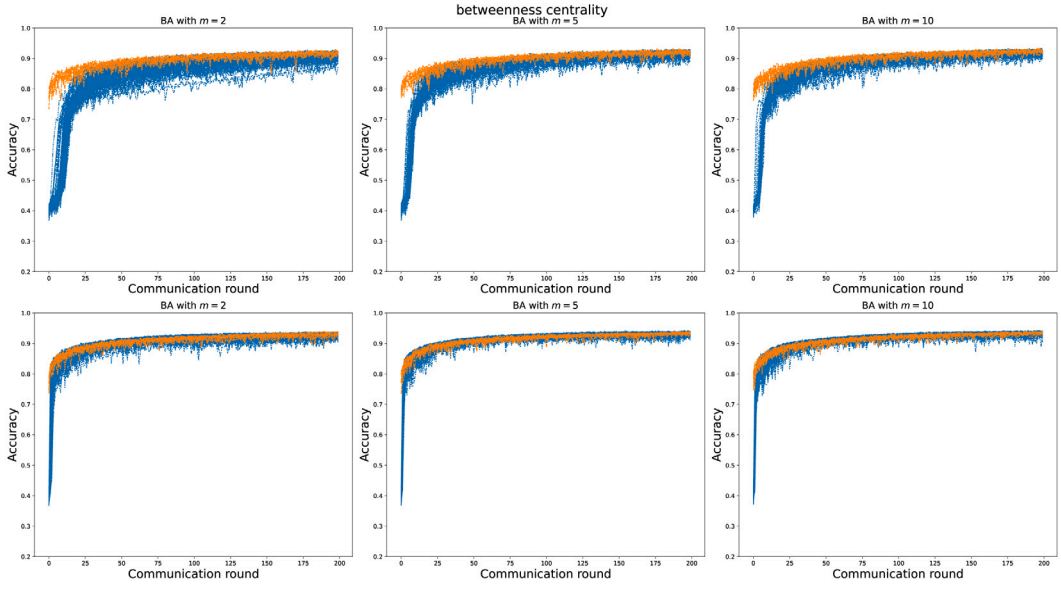


Fig. B.64. Fashion-MNIST – Accuracy over time in BA networks (all nodes, G2 data assigned based on betweenness centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

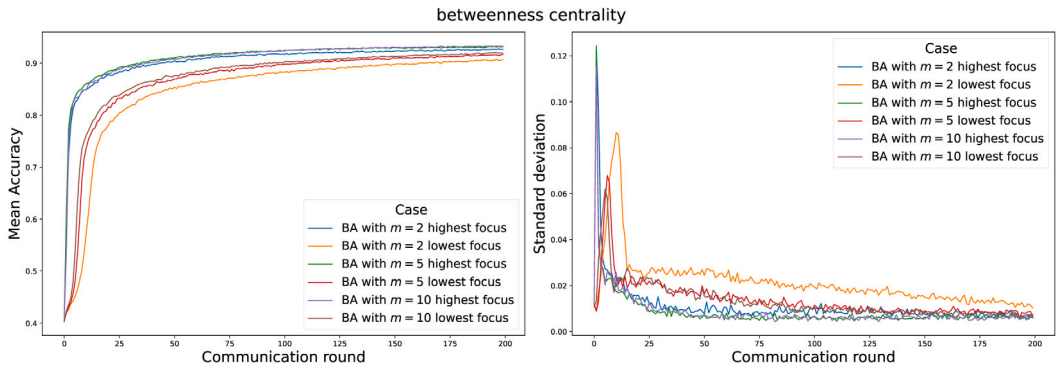


Fig. B.65. Fashion-MNIST – Mean accuracy and standard deviation over time in BA networks (average across G1 nodes only, G2 data assigned based on betweenness centrality).

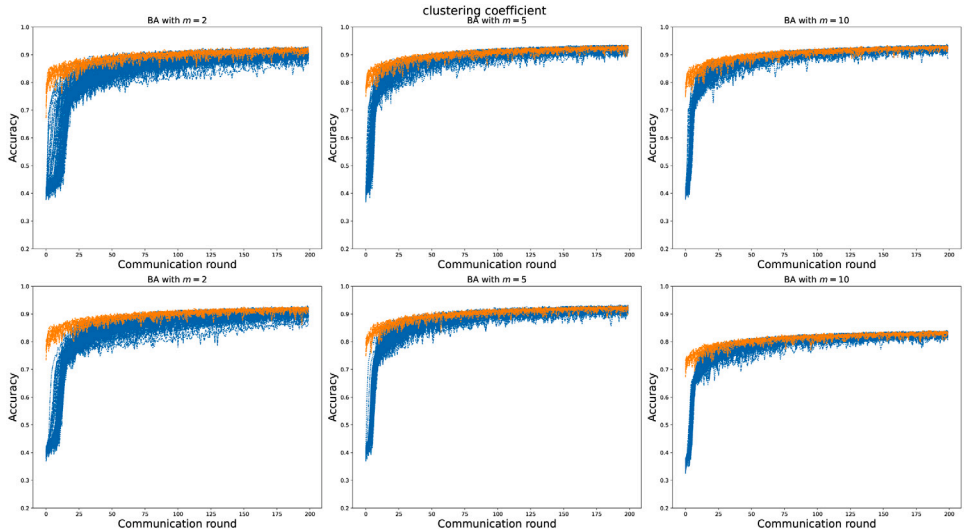


Fig. B.66. Fashion-MNIST – Accuracy over time in BA networks (all nodes, G2 data assigned based on clustering coefficient centrality). G1 nodes in blue, G2 nodes in orange. From left to right: increasing values of connectedness; from top to bottom: lowest-focused and highest-focused scenario.

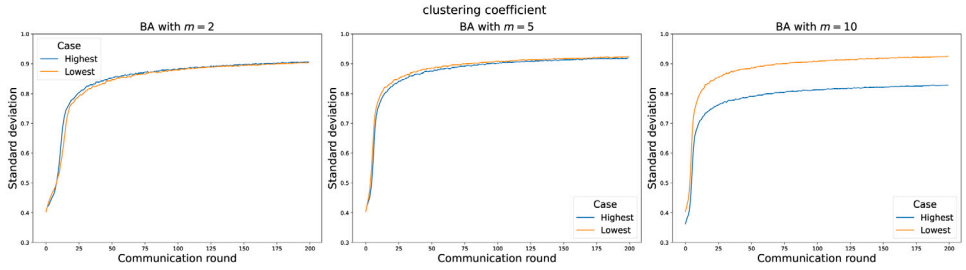


Fig. B.67. Fashion-MNIST – Mean accuracy over time in BA networks (average across G1 nodes only, G2 data assigned based on clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases.

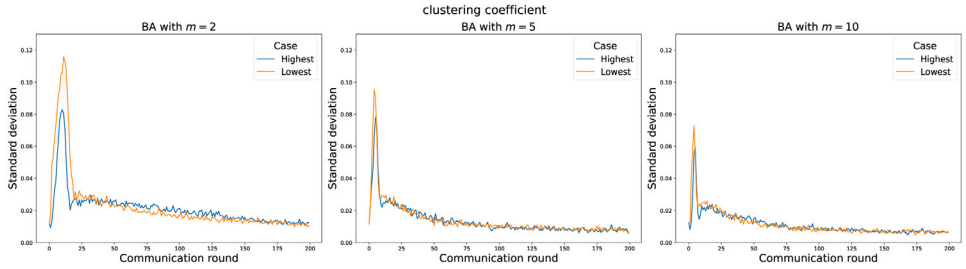


Fig. B.68. Fashion-MNIST – Standard deviation over time in BA networks (clustering coefficient). Highest-focused (blue) and lowest-focused (orange) cases.

References

[1] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-efficient learning of deep networks from decentralized data, in: AISTATS'17, 2017.

[2] T. Sun, D. Li, B. Wang, Decentralized federated averaging, IEEE Trans. Pattern Anal. Mach. Intell. 45 (04) (2023) 4289–4301, http://dx.doi.org/10.1109/TPAMI.2022.3196503.

[3] A.G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, C. Wachinger, BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated learning, 2019, pp. 1–9, arXiv, URL http://arxiv.org/abs/1905.06731.

[4] A. Lalitha, X. Wang, O. Kilinc, Y. Lu, T. Javidi, F. Koushanfar, Decentralized bayesian learning over graphs, 2019, arXiv.

[5] S. Savazzi, M. Nicoli, V. Rampa, Federated learning with cooperating devices: A consensus approach for massive IoT networks, IEEE Internet Things J. 7 (5) (2020) 4641–4654, http://dx.doi.org/10.1109/JIOT.2020.2964162, URL https://ieeexplore.ieee.org/document/8950073/.

[6] H.T. Nguyen, R. Morabito, K.T. Kim, M. Chiang, On-the-fly resource-Aware model aggregation for federated learning in heterogeneous edge, in: 2021 IEEE Global Communications Conference, GLOBECOM, IEEE, Madrid, Spain, 2021, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM46510.2021.9685893, URL https://ieeexplore.ieee.org/document/9685893/.

[7] L. Valerio, C. Boldrini, A. Passarella, J. Kertész, M. Karsai, G. Iñiguez, Coordination-free decentralised federated learning on complex networks: Overcoming heterogeneity, 2023, arXiv preprint arXiv:2312.04504.

[8] R. Ormándi, I. Hegedűs, M. Jelasity, Gossip learning with linear models on fully distributed data, Concurr. Comput.: Pract. Exper. 25 (4) (2013) 556–571.

[9] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, S. Stich, A unified theory of decentralized sgd with changing topology and local updates, in: International Conference on Machine Learning, PMLR, 2020, pp. 5381–5393.

[10] T. Zhu, F. He, L. Zhang, Z. Niu, M. Song, D. Tao, Topology-aware generalization of decentralized sgd, in: International Conference on Machine Learning, PMLR, 2022, pp. 27479–27503.

[11] L. Palmieri, L. Valerio, C. Boldrini, A. Passarella, The effect of network topologies on fully decentralized learning: a preliminary investigation, in: Proceedings of the 1st International Workshop on Networked AI Systems, 2023, pp. 1–6.

[12] L. Palmieri, C. Boldrini, L. Valerio, A. Passarella, M. Conti, Exploring the impact of disrupted peer-to-peer communications on fully decentralized learning in disaster scenarios, in: 2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), IEEE, 2023, pp. 1–6.

[13] L. Palmieri, C. Boldrini, L. Valerio, A. Passarella, M. Conti, J. Kertész, Robustness of decentralised learning to nodes and data disruption, 2024, arXiv preprint arXiv:2405.02377.

[14] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, J.S. Rellermeyer, A survey on distributed machine learning, Acm Comput. Surveys (Csur) 53 (2) (2020) 1–33.

[15] X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-iid data, in: International Conference on Learning Representations, 2019.

[16] B. Song, P. Khanduri, X. Zhang, J. Yi, M. Hong, Fedavg converges to zero training loss linearly for overparameterized multi-layer neural networks, in: International Conference on Machine Learning, PMLR, 2023, pp. 32304–32330.

[17] P. Erdős, A. Rényi, et al., On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci 5 (1) (1960) 17–60.

[18] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (1) (2002) 47.

[19] C. Lee, D.J. Wilkinson, A review of stochastic block models and extensions for graph clustering, Appl. Netw. Sci. 4 (1) (2019) 1–50.

[20] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS '02, Association for Computing Machinery, New York, NY, USA, 2002, pp. 41–47, http://dx.doi.org/10.1145/586110.586117.

[21] A.-L. Barabási, Network science, philosophical transactions of the royal society a: Mathematical, Phys. Eng. Sci. 371 (1987) (2013) 20120375.

[22] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl Acad. Sci. 99 (12) (2002) 7821–7826.

[23] T.P. Peixoto, The graph-tool python library, figshare, 2014, http://dx.doi.org/10.6084/m9.figshare.1164194, URL http://figshare.com/articles/graph_tool/1164194.

[24] The mnist database of handwritten digits, http://yann.lecun.com/exdb/mnist/.

[25] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, Emnist: Extending mnist to handwritten letters, in: 2017 International Joint Conference on Neural Networks, IJCNN, IEEE, 2017, pp. 2921–2926.

[26] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017, arXiv preprint arXiv:1708.07747.

[27] L.C. Freeman, A set of measures of centrality based on betweenness (1977) 35–41.

Luigi Palmieri is currently enrolled in the national Ph.D. program in Artificial Intelligence. He graduated at the University of Pisa, where he earned a master's degree in theoretical physics. One of his key areas of interest during his studies was Complex Systems and Nonlinear Dynamics, with an emphasis on the discipline of Neuroscience and the modeling of the brain as a Complex System in particular. His Ph.D. research focuses on the impact that topologies can have on how information spreads in different networks and Machine Learning tasks. Specifically, he will investigate complex topologies. Such topologies have several interesting properties and are the most similar to real life networks.

Chiara Boldrini is a Senior Researcher at IIT-CNR. Her research interests are in decentralized AI, computational social sciences, mobile and ubiquitous systems. She has published 50+ papers on these topics. She is the IIT-CNR PI for H2020 SoBigData++, NRRP FAIR, NRRP ICSC, and was involved in several EC projects since FP7. She is the Editor-in-Chief for Special Issue of Elsevier Computer Communications and she serves on the Editorial Board of Elsevier Pervasive and Mobile Computing. She serves as TPC chair of IEEE PerCom'24 and, over the years, has been on the organizing committee of several IEEE and ACM conferences/workshops, including IEEE PerCom and ACM MobiHoc.

Lorenzo Valerio is a researcher at IIT-CNR. His research activity focuses on the design of communication efficient distributed learning solutions, machine learning solutions for resource-constrained devices and machine learning based cellular traffic offloading solutions for the Future Internet. He has published in journals and conference proceedings more than 40+ papers. He has served as Workshop co-chair for IEEE AOC'15 and IEEE PerConAI'22, IEEE PerConAI'23. He has also been guest editor for the Elsevier Computer Communications and Elsevier Pervasive and Mobile Computing. He has been recipient for three Best Paper Awards and one Best Paper Nomination. He is currently in the editorial board of Elsevier Computer Communications.

Andrea Passarella (Ph.D. 2005) is a Research Director at the Institute for Informatics and Telematics (IIT) of CNR. Prior to joining IIT, he was with the Computer Laboratory of the University of Cambridge, UK. He has published 200+ papers on human-centric mobile networks, Online and Mobile social networks, opportunistic, ad hoc and sensor networks. He received four best paper awards, including at IFIP Networking 2011 and IEEE WoWMoM 2013. He was General Co-Chair of IEEE PerCom 2022 and WoWMoM 2019, and workshops co-chair for IEEE INFOCOM 2019. He was the PC co-chair of IEEE WoWMoM 2011, Workshops co-chair of several IEEE and ACM conferences. He is the founding Associate EiC of the Elsevier Journal Online Social Networks and Media (OSNEM). He is co-author of the book "Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs" (Elsevier, 2015). He is the chair of the IFIP WG 6.3 "Performance of Communication Systems".

Marco Conti is a CNR research director and, currently, he is the director of IIT-CNR institute. He has published more than 400 scientific articles related to design, modeling, and experimentation of computer and communication networks, pervasive systems, and online social networks. He is the founding EiC of Online Social Networks and Media, EiC for Special Issues of Pervasive and Mobile Computing, and, from 2009 to 2018, EiC of Computer Communications. He has received several awards, including the Best Paper Award at IFIP Networking 2011, IEEE ISCC 2012, and IEEE WoWMoM 2013. He was included in the "2017 Highly Cited Researchers" list compiled by Web of Science for the most cited articles in Computer Science. He served as the General/Program chair of several major conferences, including IFIP Networking 2002, IEEE WoWMoM 2005 and 2006, IEEE PerCom 2006 and 2010, ACM MobiHoc 2006, IEEE MASS 2007 and IEEE SmartComp 2021.