Fair swarm learning: Improving incentives for collaboration by a fair reward mechanism

Mohammad Tajabadi, Dominik Heider

Knowledge-Based Systems 304 (2024) 112451

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Fair swarm learning: Improving incentives for collaboration by a fair reward mechanism

Mohammad Tajabadi *, Dominik Heider

Department of Computer Science, Faculty of Mathematics and Natural Sciences, Heinrich Heine University Düsseldorf, Graf-Adolf-Str. 63, Düsseldorf, 40215, North Rhine-Westphalia, Germany

ARTICLE INFO

Code link: https://github.com/mohamadtaj/FairSL

Keywords:
Federated learning
Swarm learning
Fairness
Collaborative fairness
Decentralized machine learning

ABSTRACT

Swarm learning is an emerging technique for collaborative machine learning in which several participants train machine learning models without sharing private data. In a standard swarm network, all the nodes in the network receive identical final models regardless of their individual contributions. This mechanism may be deemed unfair from an economic perspective, discouraging organizations with more resources from participating in any collaboration. Here, we present a framework for swarm learning in which nodes receive personalized models based on their contributions. The results of this study demonstrate the efficacy of this approach by showing that all participants experience performance enhancements compared to their local models. However, participants with higher contributions receive better models than those with lower contributions. This fair mechanism results in the highest possible accuracy for the most contributive participant, comparable to the standard swarm learning model. Such incentive structure can motivate resource-rich organizations to engage in collaboration, leading to the development of machine learning models that incorporate data from more resources, which is ultimately beneficial for every party.

1. Introduction

Federated learning (FL) is a machine learning technique in which several devices or organizations collaboratively train a model without the need to share their data with one another [1]. The utilization of FL can be attributed to three main factors: First, with the advancement of technology, devices such as Internet-of-Things (IoT) devices or cell phones generate vast amounts of data, and it is not feasible to collect all the data in a centralized location to perform a machine learning task [2]. Second, due to data protection regulations such as the European General Data Protection Regulation (GDPR) or other security reasons, sharing data with other parties is constrained [3], which is especially important in applications with sensitive data, such as financial or medical applications. Lastly, data scarcity in certain applications makes it difficult to build machine learning models with good predictive performance [4]. Therefore, FL can be used to allow multiple organizations to build models trained on larger amounts of data, leading to better performance, however, without sharing any raw data.

Generally, in FL, devices start training on their local data and then share their model parameters with a central server. These parameters are then aggregated at the server and then sent back to devices to update their models. Depending on the machine learning algorithm used, this process may involve one or more communication rounds with the server [1,5]. However, there are some concerns with this client–server mechanism that should be considered. First, the reliance on a central server implies that all the clients must trust a single party to govern the learning process. Since the central server has access to all the clients' updates, a malicious server may attempt to access sensitive information from one or more clients. Secondly, this structure concentrates power in a single party and also decreases fault tolerance [6].

To address these shortcomings, an alternative method has been proposed in which nodes communicate with one another in a peer-to-peer manner rather than through a central server (Fig. 1). This framework, known as Swarm Learning (SL) [7], prevents the monopoly of power and increases flexibility in the whole learning process.

In a conventional FL or SL network, nodes train on their local data and then share their model parameters with each other to ultimately build a single global model that outperforms any other model locally trained by individual nodes. Consequently, all the nodes will enjoy the same global model and potentially experience the same predictive performance. However, this approach does not consider the contribution of each node in the learning process. Nodes vary in terms of their data size, data quality, processing capabilities, and other factors that may influence the learning process. If all the nodes in the network eventually

* Corresponding author.
E-mail address: mohammad.tajabadi@hhu.de (M. Tajabadi).

https://doi.org/10.1016/j.knosys.2024.112451
Received 19 January 2024; Received in revised form 13 August 2024; Accepted 31 August 2024
Available online 2 September 2024

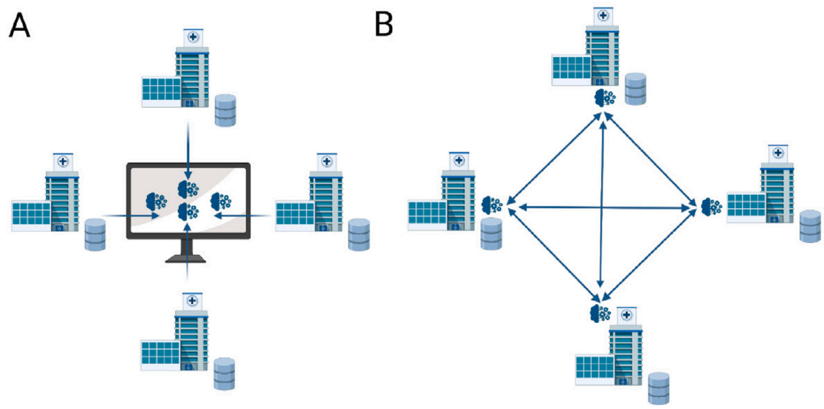


Fig. 1. Federated learning (A) and swarm learning (B) frameworks [8].

receive the same model, it will not be fair to the nodes that contribute more [8].

From an economic perspective, it is reasonable to argue that participants with greater contributions should receive relatively better models. If all the nodes eventually receive the same model, nodes with large amounts of data are likely not to participate in such a collaborative learning process. Since data collection could be costly, it is fair that nodes with larger investments receive better models [8,9].

In this paper, we aim to address the concept of collaborative fairness by designing an SL system in which nodes receive personalized models with regard to their contributions. We argue that this approach could serve as a motivator, encouraging organizations to contribute more resources to the learning process. The main contributions of this paper are as follows: We propose a novel swarm learning algorithm that ensures fair treatment of all participants by optimizing the balance between performance and resource allocation. We test and validate the algorithm on multiple datasets, demonstrating its effectiveness in real-world scenarios. By ensuring fairness, our approach incentivizes organizations to participate in the collaborative learning process.

The rest of this paper is organized as follows: We review the related work in Section 2. In Section 3, we present our proposed methods. Section 4 presents our simulations and the results. In Section 5, we discuss the implications of our findings, shortcomings, and potential future research. Finally, Section 6 concludes the paper.

2. Related work

There are three main notions of fairness discussed in FL. The first notion concerns discrimination towards certain groups or individuals [10]. In the context of group fairness, the objective is to build systems that do not discriminate against specific groups (e.g., based on sex, religion, or race) [11–14]. Discrimination in ML algorithms often arises when the training data is imbalanced, such as having a majority of male samples and only a few female samples. For instance, Yang and Jiang [14] propose a semi-centralized adversarial training framework to generate under-represented samples to de-bias the training data. Similarly, individual fairness states that individuals should be treated similarly regardless of their group membership [15]. The second notion of fairness, known as client-level or device-level fairness, centers around decreasing the variance of the predictive performance across nodes [16,17]. Given that some devices can have more impact on the learning process, research shows that the performance of the final model might favor those nodes [16,18]. To mitigate this issue, researchers have developed methods to reduce this variance, ensuring a consistent level of predictive performance across all the nodes [19–23]. For example, Hosseini et al. [16] proposed Prop-FFL for histopathology images, a scheme based on an optimization objective function to provide uniform performance across hospitals. DRFL is another method proposed by Zhao and Joshi [21], which dynamically adjusts the weight assigned to each participant to decrease the performance variations.

While most of the current research on fairness in FL focuses on the two previous views, there exists another notion of fairness known as collaborative fairness that views the problem from an economic perspective [8]. The economic notion of fairness deals with reward systems in which nodes receive rewards corresponding to their contributions. Most existing research concerning incentive mechanisms for collaborative fairness relies on motivating clients through monetary rewards [24–26]. For instance, Zhang et al. [24] introduced RRAFL, a horizontal FL incentive mechanism leveraging reputation and reverse auction theory. It aims to encourage active participation among parties and allows the requester (server or model owner) to select reliable, high-quality data contributors (clients or participants).

However, there is limited research that views this issue from a performance point of view. Lyu et al. [9] proposed CFFL, a reputation-based mechanism, which evaluates the participants' contributions based on the quality of their model updates in the learning process and iteratively updates their respective reputations. However, it relies on a central server to manage the reputations. In another approach, Lyu et al. [27] proposed FPPDL, a decentralized Fair and Privacy-Preserving Deep Learning framework to address fairness in federated settings. in this framework, parties rate other parties through a credibility initialization phase. Then, they determine the amount of meaningful parameters to share based on the credibility list. However, the credibility ratings are derived from each party's ability to predict artificially generated samples. This approach may not accurately reflect the true quality or relevance of the underlying data from each party, potentially leading to biased parameter-sharing decisions.

3. Proposed methods

In this section, we present our proposed framework for a fair SL system. In the context of this paper, we define fairness as follows:

Definition 1. Fairness refers to ensuring that all participating entities in a collaboration receive benefits proportional to their investments. Therefore, we identify an FL or SL system as fair if participants receive models that are commensurate with their contributions to the learning process.

Based on this definition, we can formulate fairness as follows: Suppose we have n nodes that want to collaboratively train machine learning models in a swarm network. In order to introduce fairness to the system, we assume that the nodes that provide more data to the process should be rewarded with better models due to their higher contributions [9,28]. Let C_i be the contribution of node i (i.e., number of data samples), and B_i be the benefits received by node i (i.e., model accuracy). Although the exact ratio B_i/C_i does not need to be equal for all participants, a fair system should ensure that B_i increases with C_i. We introduce the function f as:

$$f(C_i) = B_i \tag{1}$$

This function maps contributions to expected benefits. For fairness, the function should be monotonically increasing, i.e., if $C_i \leq C_j$, then $B_i \leq B_j$.

For training a machine learning model, choosing the right algorithm depends on several factors, one of which is data availability. For instance, Neural Networks, and in particular deep learning (DL), have demonstrated remarkable performance when there are large amounts of data available. On the other hand, other machine learning techniques such as Random Forests (RFs) work reasonably well when there are fewer data samples available. Accordingly, we have designed two mechanisms, one for DL and one for RFs, to build the SL system. In the following sections, we will explore the specifications of each of these mechanisms.

Algorithm 1 Fair swarm network for deep learning

Initialize peer models $w_0^1, w_0^2, \ldots, w_0^n$ for nodes 1 to n.
for cycle $c = 1$ to n **do**
  for training round $t = 1$ to T **do**
    for node $i = c$ to n **do**
      Train locally on section c ($w_t^i \leftarrow w_{t-1}^i - \eta \nabla F(w_{t-1}^i, D_c^i)$).
      Send w_t^i to the coordinator.
    end for
    Merge the parameters at the coordinator ($w_{merged} \leftarrow \frac{1}{n-i} \sum_{i=c}^{n} w_c$).
    Send the updated parameters to all the active nodes c to n
    for node $i = c$ to n **do**
      Update model parameters at node i ($w_t^i \leftarrow w_{merged}$)
    end for
  end for
  Node c exits with the final model at cycle c
end for

3.1. Deep learning

In a swarm network of n nodes with different amounts of data, we consider n cycles of training, each cycle resulting in an output model for a specific node. Starting from the nodes with fewer data samples, each cycle personalizes a model for each node which then quits the process, while other nodes continue building their models. Therefore, at cycle i, $n+1-i$ nodes are participating. Each cycle consists of multiple iterations in which nodes train on their local data, share their model parameters with other active nodes in the network, and then update their models using parameters received from other nodes.

It is important to note that before starting the training, nodes need to report the amount of their available data to others. There are challenges with this self-reporting scheme, such as the possibility of having malicious or untruthful nodes in the system that report false data. However, addressing issues related to dishonest nodes and fraudulent approaches falls beyond the scope of this paper.

Suppose that each node i has S_i training samples, and we have:

$$S_1 < S_1 < \cdots < S_n \tag{2}$$

Each node divides its dataset into k sections ($1 \leq k \leq n$) such that the size of section m, L_m, equals to:

$$L_m = \begin{cases} S_m & \text{if } m = 1 \\ S_m - \sum_{j=1}^{m-1} S_j & \text{if } 1 < m \leq k \end{cases} \tag{3}$$

By way of illustration, suppose we have three nodes labeled 1, 2, and 3, with the first node having the lowest number of samples and the third node having the highest number of samples. They will divide their datasets into 1, 2, and 3 sections, respectively. Node 1 will have 1 section comprising its whole dataset. Node 2, on the other hand, will split its data into 2 sections. The first section matches the size of Node 1's dataset, while the second section comprises the remaining data. Finally, node 3 divides its dataset into 3 sections, with the first section matching the size of the first section of the other nodes, the second section matching the second section of node 2, and the third section comprising the remaining data.

The fair collaborative deep learning process is shown in Algorithm 1. At cycle c, nodes c to n start training on their respective cth data section, D_c, for a fixed number of iterations (see Fig. 2). After each iteration, nodes send their model parameters to a designated coordinator node. The coordinator then merges the parameters by taking the average and then sends the updated parameters, w_{merged}, to all the active nodes in the network. Upon receipt of the new parameters, nodes update their models and start training for the next iteration. The coordinator also randomly chooses one of the active nodes as the next coordinator and informs the other nodes about the next acting coordinator. When cycle c ends (after a certain number of iterations), all the active nodes will have the same model. At this point, node c will leave the network with this model, while the remaining nodes start the next cycle, training on the next section of their respective datasets. At the final cycle, there remains only the last node training on its remaining data (final section), while all the other nodes have already left the network with their own personalized models.

Through this process of multiple cycles, nodes leverage each other's data but in a fair way. The contribution of all the nodes will be equal at each cycle, that is, each node will receive model parameters from other nodes but only from a subset of their dataset which is equal to the node's own data size.

3.2. Random forest

While neural networks are extensively used in tasks such as image classification, speech recognition, and machine translation, they are not always the optimal choice for a learning task, especially when there is a limited number of samples. In the absence of large amounts of data, other models, such as RFs, demonstrate significant robustness.

To collaboratively train a DL model, nodes need to communicate with each other multiple times. In the simplest form, nodes share their model parameters after each iteration. However, there are other feasible strategies as well, such as sharing the parameters after training multiple batches within the same iteration or after multiple iterations. The former, when employed, might yield a model with better generalization across the nodes in the network but at the expense of more communication rounds. The latter, on the other hand, is more efficient in terms of communications but may result in a model with inferior performance.

Unlike DL, however, RFs are not trained in multiple iterations. Therefore, to merge several RFs, only one round of communication is needed when the nodes share their parameters with each other after the training is complete.

In this section, we present our implementation of the fair RF swarm framework.

We assume the same scenario where n nodes are connected to each other in a swarm network, with the objective of training RF models with collaboration. First, nodes start training RFs on their local data. After the training is finished, nodes need to share their model parameters with each other. In an unfair scheme, nodes essentially share all their parameters with other nodes in the network, ultimately resulting in the same global model for all the nodes. To introduce fairness to the system, we have designed a mechanism in which nodes share only a portion of their parameters with other nodes based on each node's contribution. Once again, we consider each node's contribution as the number of data points they provide for the learning process. Since having more

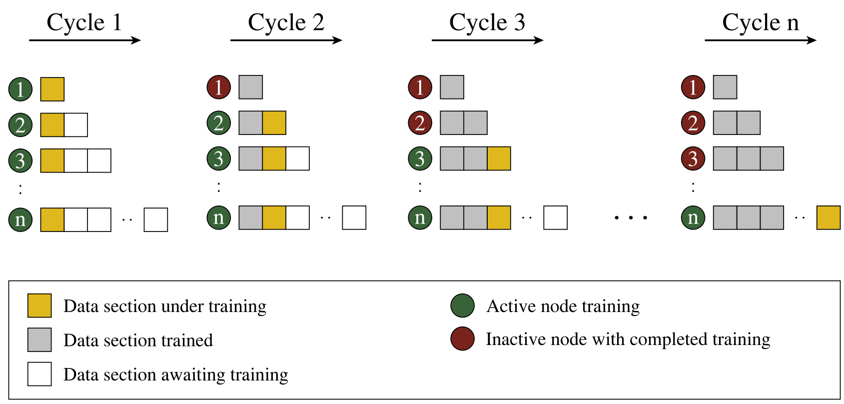


Fig. 2. The process of training on data sections through multiple cycles: each node i has i sections. At cycle c, nodes c to n are active, while previous nodes have completed training and exited the network (marked as inactive). The sizes of the sections under training at a given cycle are equal across nodes, but the section sizes at each node are not necessarily equal.

data generally results in better models, nodes with more data can be considered to contribute more to the system. The process for the fair SL system with random forest is shown in Algorithm 2.

Algorithm 2 Fair swarm network for random forest

```
Initialize RF classifiers with t_1, t_2, ..., t_n estimators for nodes 1 to n.
Train an RF classifier locally at each node.
for node i = 1 to n do
    for peer j = 1 to n − 1 in node i's connections do
        if node i has fewer samples (s_i < s_j) then
            t ← t_i
        else
            t ← t_i × (s_i / s_j)^2
        end if
        Randomly select t estimators from the trained classifier.
        Send the t estimators to the peer node j.
    end for
end for
for node i = 1 to n do
    Update the classifier by incorporating the received estimators into it.
end for
```

The process starts with each node defining a personal RF model. The number of trees (estimators) for each RF corresponds to the number of data points at the node. Consequently, in our scheme, a node with a larger dataset starts with a larger RF than a node with a smaller one. After defining the models, nodes fit the models on their local data. Upon the completion of training at a given node, it sends a portion of its estimators or trees to every other node in the network. The amount of estimators to share with others depends on their respective data sizes.

Suppose that node i has s_i samples and t_i trees and node j has s_j samples and t_j trees where:

$$\frac{s_i}{s_j} = \frac{t_i}{t_j} \qquad s_i > s_j. \tag{4}$$

The amount of estimators that node j needs to share with node i is equal to t_j. The amount of estimators that node i needs to send to j is equal to:

$$N_{i\to j} = t_j \times \frac{s_j}{s_i} = t_i \times (\frac{s_j}{s_i})^2. \tag{5}$$

The final personalized model for each node consists of its own estimators plus other estimators received from all the other nodes in the network. In this way, each node builds a model that has training information from all the other nodes but in a fair manner.

4. Simulations

In this section, we present our simulations for the two proposed methods for DL and RFs and then discuss the results for each part.

4.1. Deep learning

To evaluate the performance of the proposed system for DL, we simulated an SL system in which multiple nodes collaboratively train DL models based on their contributions. We used four standard datasets namely CIFAR-10 [29], CIFAR-100 [29], and Fashion-MNIST [30] for image classification and the Reuters Newsletter dataset [31] for text classification. To examine the differences in contributions, we explored two setups. In the main setup, we distributed the dataset across three nodes with respective ratios of 0.1, 0.3, and 0.6. In another setup, we used four nodes, dividing the data in proportions of 0.10, 0.12, 0.38, and 0.40, to investigate the impact of near-equal contributions from pairs of nodes (the first two and the last two having almost identical contributions). We used CIFAR-100 and Fashion-MNIST to examine the second setup. We then trained models on the training set at each node and evaluated their performance using a common external test set.

For the CIFAR-10 and Fashion-MNIST datasets, a 3-layer VGG architecture (VGG8) [32] was used in conjunction with regularization techniques [33]. For the Reuters dataset, we trained a neural network consisting of an embedding layer and two dense layers. Lastly, for CIFAR-100 dataset, we used EfficientNetV2S [34], a deep convolutional neural network pretrained on the ImageNet dataset [35]. Moreover, we used the Adam optimizer [36] and Cross-Entropy loss function for the experiments. Table 1 summarizes the setup for the experiments. Note that for the Fashion-MNIST dataset, we randomly selected 5% of the entire dataset for training to investigate cases with limited data points at each site.

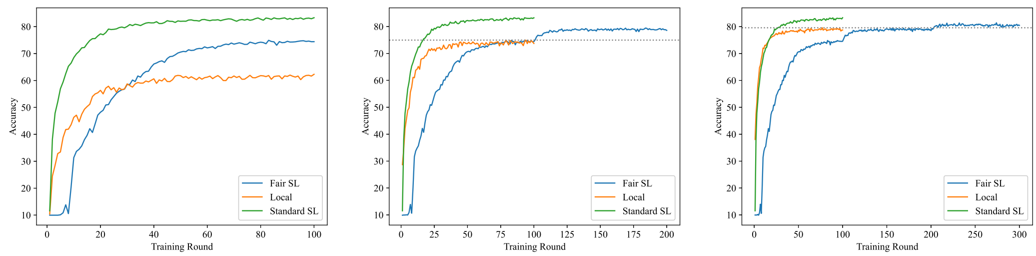
Our simulations consist of three distinct scenarios. In the first scenario, nodes train their models based on our fair SL system. In the second scenario, nodes train their models locally without any collaboration. Finally, in the third scenario, nodes participate in a standard SL network to train a single global model regardless of their contributions.

In the training process of the fair SL model, at the end of each cycle, one of the nodes leaves the network. The other nodes remain with an already partially trained model, and they need to train it on new data. To ensure the preservation of the existing models, we employed the principle of transfer learning [37]. This involved freezing certain layers within the models and reducing the learning rate so that the models converge better by leveraging prior knowledge and learning from new data.

As we discussed earlier, in a fair collaborative system, we expect a gain in performance as contributions increase. Figs. 3 to 6 show the

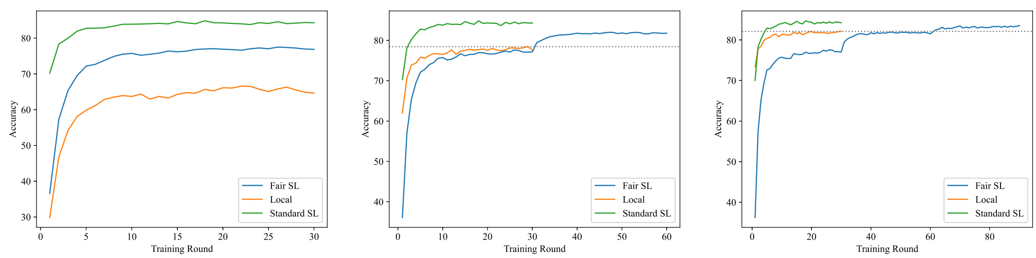
Table 1
Experimental setup for the deep learning frameworks.

Dataset	Epochs per cycle	Learning rate	Batch size	Train samples	Test samples	Classes
CIFAR-10	100	0.001	128	50,000	10,000	10
CIFAR-100	30	0.005	128	50,000	10,000	100
Fashion-MNIST	100	0.001	32	3000	10,000	10
Reuters	200	0.003	128	8982	2246	46



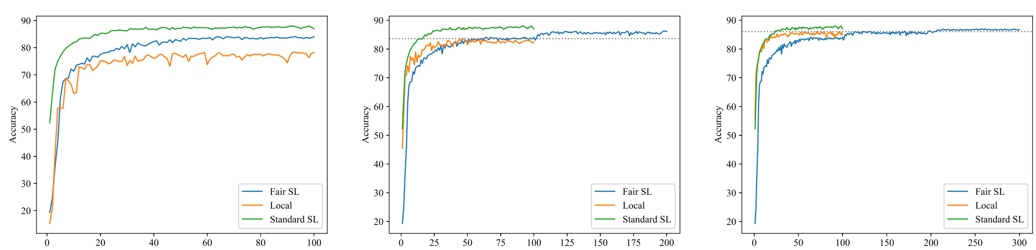
(a) Node 1 (b) Node 2 (c) Node 3

Fig. 3. CIFAR-10 test accuracy for three nodes.



(a) Node 1 (b) Node 2 (c) Node 3

Fig. 4. CIFAR-100 test accuracy for three nodes.



(a) Node 1 (b) Node 2 (c) Node 3

Fig. 5. Fashion-MNIST test accuracy for three nodes.

results of the simulations for the four datasets for three nodes. As illustrated in the results, for each node, the fair SL model outperforms the locally trained model. Furthermore, nodes with higher contributions experience better performance than those with lower contributions when employing the fair SL system.

Figs. 7 and 8 present the results from the second setup with four nodes. The results indicate that the performance of the models at Nodes 1 and 2 is equal, as is the performance of the models at Nodes 3 and 4. However, Nodes 3 and 4 exhibit better performance due to their higher contributions, which aligns with the expectations of our fair SL system.

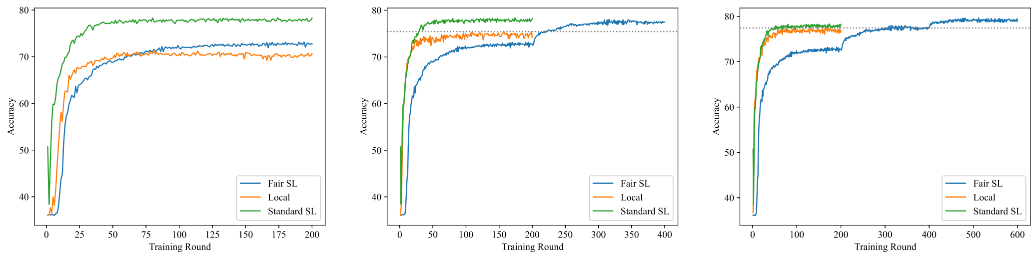
Table 2 shows the differences in error rates for neural networks in the first setup where we have three nodes in the system with data split among them with the ratios 0.1, 0.3, and 0.6. Similarly, Table 3 illustrates the differences in error rates in the second setup with four nodes with data ratios of 0.10, 0.12, 0.38, and 0.40. As it is shown, the error rate decreases when contributions increase.

Table 2
Decrease in error rate in neural networks for the first setup with three nodes with data ratios 0.1, 0.3, and 0.6.

Dataset	Error-rate decrease	
	Node 1 to Node 2	Node 2 to Node 3
CIFAR-10	16.24%	8.91%
CIFAR-100	21.16%	9.65%
Fashion-MNIST	12.82%	4.60%
Reuters Newsletter	17.24%	6.96%

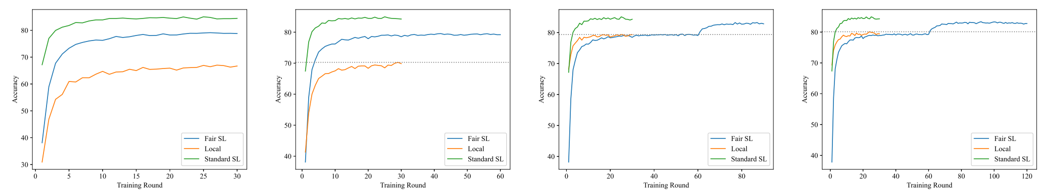
4.2. Random forest

To evaluate the proposed method for RFs, we built an SL network with three nodes with different amounts of data split among them. We conducted simulations using six different datasets. We used the



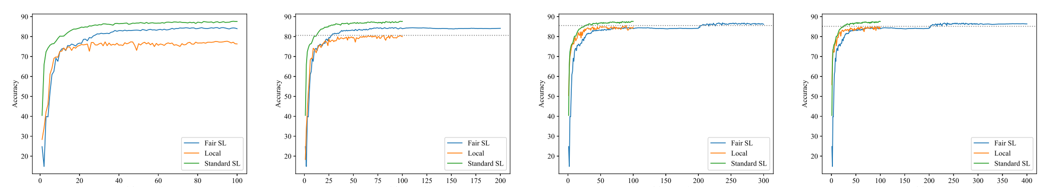
(a) Node 1 (b) Node 2 (c) Node 3

Fig. 6. Reuters Newsletter test accuracy for three nodes.



(a) Node 1 (b) Node 2 (c) Node 3 (d) Node 4

Fig. 7. CIFAR-100 test accuracy for four nodes.



(a) Node 1 (b) Node 2 (c) Node 3 (d) Node 4

Fig. 8. Fashion-MNIST test accuracy for four nodes.

Table 3
Decrease in error rate in neural networks for the second setups with four nodes with data ratios 0.10, 0.12, 0.38, and 0.40.

Dataset	Error-rate decrease		
	Node 1 to Node 2	Node 2 to Node 3	Node 3 to Node 4
CIFAR-100	1.81%	17.49%	−0.35%
Fashion-MNIST	1.18%	12.36%	2.01%

Table 4
Datasets specifications and initial estimators for nodes.

Dataset	Samples	Classes	Nodes estimators		
			Node 1	Node 2	Node 3
NATICUSdroid	29,332	2	50	150	300
Breast Cancer	116	2	10	30	60
Heart Failure	299	2	50	150	300
Maternal Health	1014	3	50	150	300
Auction Verification	2043	2	100	300	600
Students' Dropout	4424	3	30	90	180

NATICUSdroid (Android Permissions) Dataset [38] for malware detection, Breast Cancer Coimbra dataset [39] for breast cancer detection, Heart Failure dataset [40] for survival prediction, Maternal Health Risk dataset [41] for maternal mortality prediction, Auction Verification [42] for verification of auctioning frequency spectra, and Students' Dropout dataset [43] for predicting students' dropout and academic success.

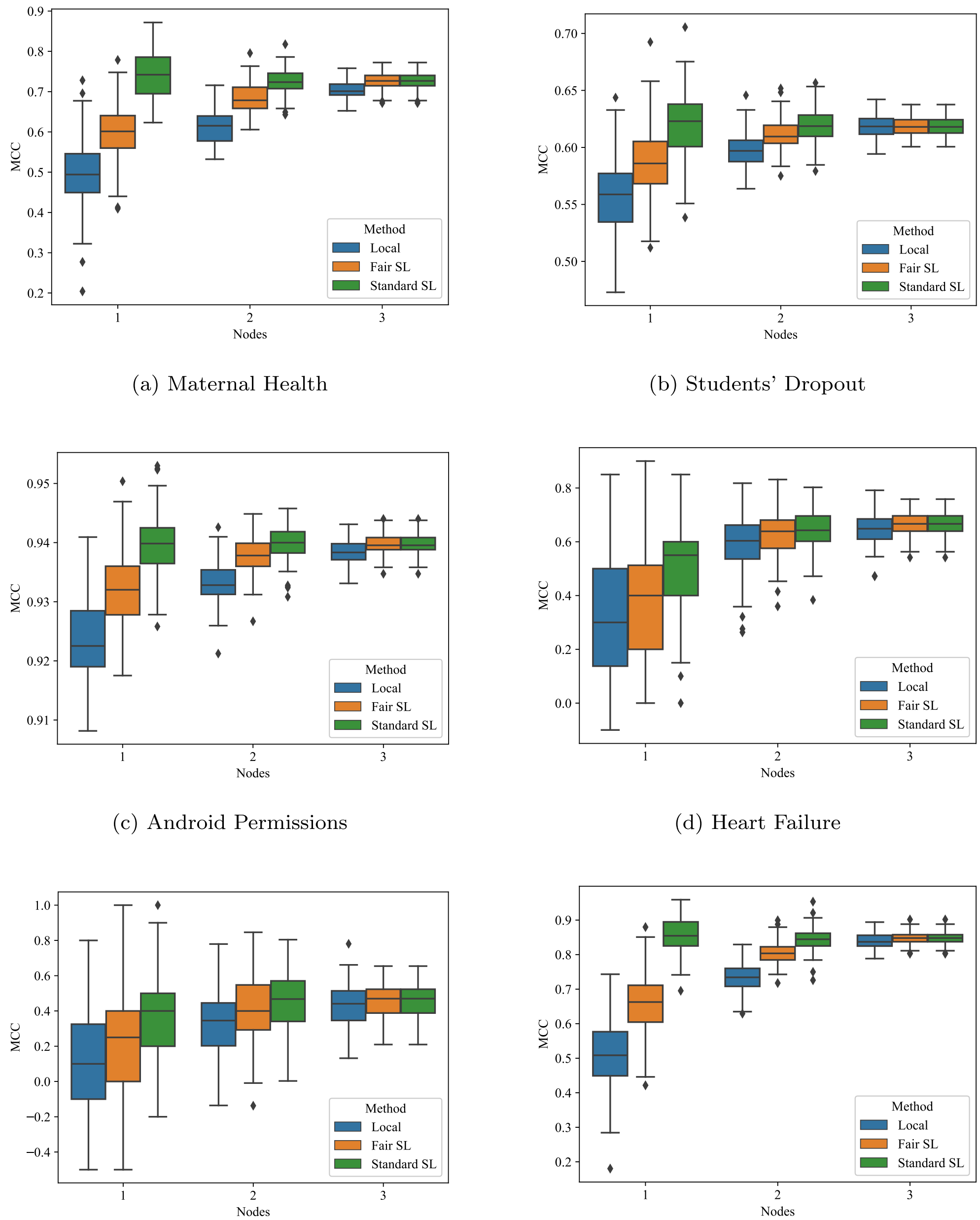
Table 4 shows the specifications of the datasets and the initial estimators defined for each node.

To introduce variations in nodes' contributions, we divided each dataset among the nodes with ratios of 0.1, 0.3, and 0.6. For training the models, we performed 5-fold and 10-fold cross-validation and repeated the experiment 100 times for each dataset with different data distributions among the nodes but with the same ratio to account for the variations in data quality at each node. For comparison, we performed the same experiments for local training in which there is no collaboration among the nodes and also for the typical unfair SL in which nodes send all their parameters to all the other nodes, resulting in the same global model.

Given the datasets in our study do not necessarily have balanced classes, we used the Matthews Correlation Coefficient (MCC) as the performance metric [44]. Fig. 9 illustrates the MCC scores for the six datasets. As is evident, nodes acquire superior models when they collaborate with each other compared to when they train individually. Moreover, in the fair SL system, nodes that contribute more get better final models than those with lower contributions. For the unfair SL system, on the other hand, all the nodes get the same model with the same predictive performance.

Similar to our experiments with neural networks, we implemented another setup with four nodes for the Maternal Health and Auction Verification datasets to explore the effect of equal contributions. The data was distributed among the four nodes in the ratios of 0.10, 0.12, 0.38, and 0.40. The results (Fig. 10) demonstrate that in the fair SL framework, the first two nodes receive identical models with equal performance, while the second two nodes receive the same models but with better performance due to their higher contributions.

Tables 5 and 6 show the gain in the mean MCC scores for random forests for the two setups with three and four nodes, respectively. We



(a) Maternal Health (b) Students' Dropout

(c) Android Permissions (d) Heart Failure

(e) Breast Cancer (f) Auction Verification

Fig. 9. Distribution of random forest MCC scores For fair SL, standard SL, and local models across three nodes with data ratios of 0.1, 0.3, and 0.6.

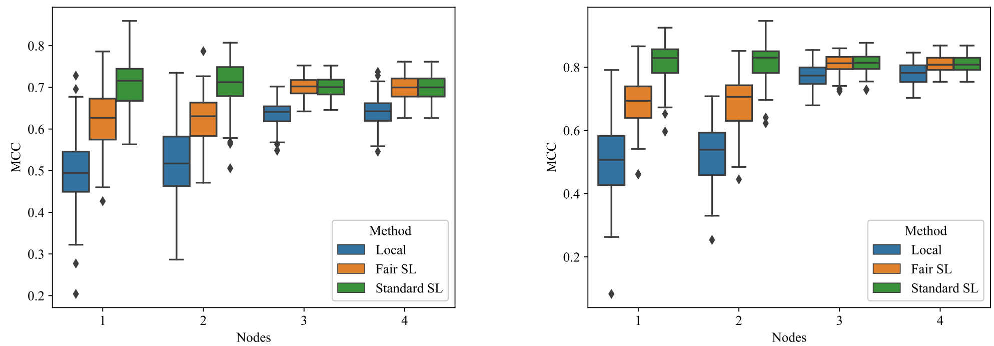
used the t-test to assess the statistical differences in the mean scores in the experiments. As anticipated, there is an increase in the MCC scores for higher contributions, aligning with expectations in a fair collaborative system. Furthermore, Table 6 reveals that there is no statistical difference between the mean scores of Nodes 1 and 2 and between Nodes 3 and 4 due to their near-equal contributions.

5. Discussion

In the previous section, we demonstrated how participants with more data can benefit more from a collaborative learning process. We interpret this as an incentive mechanism to motivate more participation in collaboration. Generally, a lack of incentive can arise in two ways. The first type is related to client-level fairness, emphasizing the need for minimal performance disparity in the ultimate model among all participants. This argument stems from the observation that in many FL systems, clients with limited resources are not favored by the system, and the resulting model has a lower performance for them. Consequently, these clients lack sufficient incentive to participate in collaboration. The second type, however, views the problem from an

Table 5
Increase in the mean MCC scores in random forests for the first setup with three nodes with data ratios 0.1, 0.3, and 0.6.

Dataset	MCC increase			
	Node 1 to Node 2		Node 2 to Node 3	
	Gain	p-value	Gain	p-value
NATICUSdroid	0.65%	4.35e−16	0.19%	3.35e−07
Auction Verification	23.00%	1.49e−37	5.30%	7.76e−24
Breast Cancer	95.80%	9.04e−09	11.10%	0.01
Heart Failure	63.23%	1.73e−21	5.78%	0.0001
Maternal Health	14.50%	8.72e−22	6.00%	3.14e−19
Students' Dropout	4.05%	9.52e−12	1.16%	9.64e−06



(a) Maternal Health (b) Auction Verification

Fig. 10. Distribution of random forest MCC scores For fair SL, standard SL, and local models across four nodes with data ratios of 0.10, 0.12, 0.38, and 0.40.

Table 6
Increase in the mean MCC scores in random forests for the second setup with four nodes with data ratios 0.10, 0.12, 0.38, and 0.40.

Dataset	MCC increase					
	Node 1 to Node 2		Node 2 to Node 3		Node 3 to Node 4	
	Gain	p-value	Gain	p-value	Gain	p-value
Auction Verification	−0.16%	0.92	17.5%	2.4e−32	−0.09%	0.85
Maternal Health	1.00%	0.48	11.90%	7.18e−25	−0.11%	0.82

economic perspective, suggesting that if all the participants receive the same reward regardless of their contributions, those with higher contributions may lose motivation to take part in the collaboration process. The rationale behind this viewpoint is that the gains derived from collaboration are inversely proportional to their contribution. Various mechanisms can be employed to reward participants with more contributions such as receiving monetary rewards or superior models. A reward for contribution can serve as an incentive to motivate participants to engage in the collaboration. In this paper, we viewed the incentive mechanism from an economic perspective. Moreover, for the reward mechanism, we approached the problem from a model-performance point of view. The reason is that models with good performance can bring potentially long-term benefits for companies that may outweigh a mere monetary reward. Nevertheless, it is worth noting that the problem could be viewed and addressed in either way based on the application and objectives.

Another noteworthy point is that the contributions of participants can be viewed from different perspectives. For example, clients vary in terms of the resources they provide, the quality and size of their data, and other aspects. In this study, our emphasis was laid on the reward system rather than contribution assessment. Therefore, for simplicity, we considered a participant's contribution as the number of data points it provides for the learning process. Since more data generally leads to better-performing models, we used the data size to compare the contributions of individuals.

Additionally, it is worth mentioning that FL and SL frameworks have been introduced to enable different parties to collaboratively train models without sharing sensitive data. However, these methods still need to incorporate additional security and privacy measures, such as homomorphic encryption or differential privacy, to prevent any disclosure of private data. In this paper, we assumed that the participants were honest and there were no malicious activities within the network. Addressing privacy issues and the presence of malicious or dishonest parties can be the focus of future research.

Furthermore, in this paper, we focus on cross-silo applications rather than cross-device applications, as cross-silo settings are more common in SL systems. In cross-silo settings, the participating entities are typically well-established organizations with stable and robust internet connections, unlike the more variable and often weaker connections that might be encountered in devices such as IoT devices [45]. Given the nature of these collaborations, with a smaller number of participants that typically maintain reliable connectivity, connection issues are not expected to significantly impact our system and are therefore excluded from our consideration.

Lastly, it is crucial to note that this fair approach is solely seen from an economic point of view, and it is advised to be used only in contexts aligned with economic considerations. For instance, from an ethical point of view, for medical applications, it might be best to focus on providing the best possible global model for everyone rather than concentrating on individual benefits.

6. Conclusion

In this paper, we addressed fairness in swarm learning from an economic perspective. We introduced two frameworks for DL and RFs in which multiple nodes could collaboratively train AI models in a fair manner without sharing private data. In contrast with federated learning, nodes communicate with each other directly in a peer-to-peer manner. In the DL framework, nodes share their model updates with each other through multiple iterations to build their final models. The final models are personalized and are based on each node's contribution to the learning process to ensure fairness. In the RF framework, nodes only send a portion of their parameters to other nodes once the training is complete. The amount of parameters to share is set with regard to each node's contribution. Our findings show that this framework yields superior model performance compared to local training across all nodes. Moreover, the models are fair in a way that nodes with more contribution receive better final models than other nodes.

CRediT authorship contribution statement

Mohammad Tajabadi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Dominik Heider:** Writing – review & editing, Validation, Supervision, Methodology, 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

The source code for this research can be found on GitHub at "https://github.com/mohamadtaj/FairSL". The repository contains all the codes needed to run the experiments and reproduce the results.

References

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

[2] J. Mills, J. Hu, G. Min, Communication-efficient federated learning for wireless edge intelligence in IoT, IEEE Internet Things J. 7 (7) (2019) 5986–5994.

[3] R. Torkzadehmahani, R. Nasirigerdeh, D.B. Blumenthal, T. Kacprowski, M. List, J. Matschinske, J. Spaeth, N.K. Wenke, J. Baumbach, Privacy-preserving artificial intelligence techniques in biomedicine, Methods Inf. Med. 61 (2022) e12–e27.

[4] M. Wedyan, A. Crippa, A. Al-Jumaily, A novel virtual sample generation method to overcome the small sample size problem in computer aided medical diagnosing, Algorithms 12 (8) (2019) 160.

[5] A.-C. Hauschild, M. Lemanczyk, J. Matschinske, T. Frisch, O. Zolotareva, A. Holzinger, J. Baumbach, D. Heider, Federated random forests can improve local performance of predictive models for various healthcare applications, Bioinformatics 38 (8) (2022) 2278–2286.

[6] D. Monschein, J.A.P. Pérez, T. Piotrowski, Z. Nochta, O.P. Waldhorst, C. Zirpins, Towards a peer-to-peer federated machine learning environment for continuous authentication, in: 2021 IEEE Symposium on Computers and Communications, ISCC, IEEE, 2021, pp. 1–6.

[7] S. Warnat-Herresthal, H. Schultze, K.L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N.A. Aziz, et al., Swarm learning for decentralized and confidential clinical machine learning, Nature 594 (7862) (2021) 265–270.

[8] M. Tajabadi, L. Grabenhenrich, A. Ribeiro, M. Leyer, D. Heider, Sharing data with shared benefits: Artificial intelligence perspective, J. Med. Internet Res. 25 (2023) e47540.

[9] L. Lyu, X. Xu, Q. Wang, H. Yu, Collaborative fairness in federated learning, Fed. Learn.: Priv. Incentive (2020) 189–204.

[10] P.K. Lohia, K.N. Ramamurthy, M. Bhide, D. Saha, K.R. Varshney, R. Puri, Bias mitigation post-processing for individual and group fairness, in: Icassp 2019-2019 Ieee International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 2847–2851.

[11] Y.H. Ezzeldin, S. Yan, C. He, E. Ferrara, A.S. Avestimehr, Fairfed: Enabling group fairness in federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 7494–7502.

[12] T. Salazar, M. Fernandes, H. Araújo, P.H. Abreu, FAIR-FATE: Fair federated learning with momentum, in: International Conference on Computational Science, Springer, 2023, pp. 524–538.

[13] R. Poulain, M.F. Bin Tarek, R. Beheshti, Improving fairness in AI models on electronic health records: The case for federated learning methods, in: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023, pp. 1599–1608.

[14] Y. Yang, B. Jiang, Towards group fairness via semi-centralized adversarial training in federated learning, in: 2022 23rd IEEE International Conference on Mobile Data Management, MDM, IEEE, 2022, pp. 482–487.

[15] J. Li, T. Zhu, W. Ren, K.-K. Raymond, Improve individual fairness in federated learning via adversarial training, Comput. Secur. (2023) 103336.

[16] S.M. Hosseini, M. Sikaroudi, M. Babaie, H. Tizhoosh, Proportionally fair hospital collaborations in federated learning of histopathology images, IEEE Trans. Med. Imaging (2023).

[17] A. Papadaki, N. Martinez, M. Bertran, G. Sapiro, M. Rodrigues, Minimax demographic group fairness in federated learning, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 142–159.

[18] T. Li, M. Sanjabi, A. Beirami, V. Smith, Fair resource allocation in federated learning, in: International Conference on Learning Representations, 2019.

[19] A. Sultana, M.M. Haque, L. Chen, F. Xu, X. Yuan, Eiffel: Efficient and fair scheduling in adaptive federated learning, IEEE Trans. Parallel Distrib. Syst. 33 (12) (2022) 4282–4294.

[20] H.J. Rad, M. Abdizadeh, A. Szabó, Federated learning with taskonomy for non-iid data, 2021, CoRR, vol. abs/2103.15947.

[21] Z. Zhao, G. Joshi, A dynamic reweighting strategy for fair federated learning, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 8772–8776.

[22] A. Linardos, K. Kushibar, K. Lekadir, Center dropout: A simple method for speed and fairness in federated learning, in: International MICCAI Brainlesion Workshop, Springer, 2021, pp. 481–493.

[23] T. Li, S. Hu, A. Beirami, V. Smith, Ditto: Fair and robust federated learning through personalization, in: International Conference on Machine Learning, PMLR, 2021, pp. 6357–6368.

[24] J. Zhang, Y. Wu, R. Pan, Incentive mechanism for horizontal federated learning based on reputation and reverse auction, in: Proceedings of the Web Conference 2021, 2021, pp. 947–956.

[25] L. Gao, L. Li, Y. Chen, W. Zheng, C. Xu, M. Xu, Fifl: A fair incentive mechanism for federated learning, in: Proceedings of the 50th International Conference on Parallel Processing, 2021, pp. 1–10.

[26] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, Y. Zhang, Fair: Quality-aware federated learning with precise user incentive and model aggregation, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.

[27] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, K.S. Ng, Towards fair and privacy-preserving federated deep models, IEEE Trans. Parallel Distrib. Syst. 31 (11) (2020) 2524–2541.

[28] P. Kokol, M. Kokol, S. Zagoranski, Machine learning on small size samples: A synthetic knowledge synthesis, Sci. Prog. 105 (1) (2022) 00368504211029777.

[29] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, 2009.

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

[31] D.D. Lewis, Y. Yang, T. Russell-Rose, F. Li, Rcv1: A new benchmark collection for text categorization research, J. Mach. Learn. Res. 5 (Apr) (2004) 361–397.

[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[34] M. Tan, Q. Le, Efficientnetv2: Smaller models and faster training, in: International Conference on Machine Learning, PMLR, 2021, pp. 10096–10106.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.

[36] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[37] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, IGI global, 2010, pp. 242–264.

[38] A. Mathur, NaticUSdroid (android permissions) dataset, 2022, http://dx.doi.org/10.24432/C5FS64.

[39] M. Patrcio, J. Pereira, J. Crisstomo, P. Matafome, R. Seia, F. Caramelo, Breast cancer coimbra, 2018, http://dx.doi.org/10.24432/C52P59.

[40] Heart failure clinical records, 2020, http://dx.doi.org/10.24432/C5Z89R.

[41] M. Ahmed, Maternal health risk, 2023, http://dx.doi.org/10.24432/C5DP5D.

[42] E. Ordoni, J. Bach, A.-K. Fleck, Analyzing and predicting verification of data-aware process models–A case study with spectrum auctions, IEEE Access 10 (2022) 31699–31713.

[43] V. Realinho, M. Vieira Martins, J. Machado, L. Baptista, Predict students' dropout and academic success, 2021, http://dx.doi.org/10.24432/C5MC89.

[44] D. Chicco, N. Tötsch, G. Jurman, The matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, BioData Min. 14 (2021) 1–22.

[45] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, Found. Trends. Mach. Learn. 14 (1–2) (2021) 1–210.