A federated learning approach for rolling bearing fault diagnosis on data sources with imbalanced class distribution

F. De Fabritiis[1,2], K. Gryllias[1,2]

[1] Department of Mechanical Engineering, Faculty of Engineering Science, KU Leuven, Leuven, Belgium
[2] Flanders Make@KU Leuven, Leuven, Belgium

Abstract

Rotating machinery fault diagnosis is a field of intensive research, attracting the last years a particular interest for data-driven methodologies such as Machine Learning and Deep Learning. In order to build such models, the general assumption is that a sufficient number of healthy and fault samples, collected under various working conditions, are available for model training. This assumption is often not realistic in real industry. This limitation could be avoided by exploiting data sets collected at multiple industrial partners, but this is in practice not easily feasible since companies prefer not to share their data for privacy reasons. Federated Learning (FL) is an emerging machine learning approach proposed to train a global model without sharing data among users. In this context an FL methodology for fault classification based on Convolutional Neural Networks (CNN) is proposed in this paper. Local models are trained on local data sets, each one owned by a single client, i.e. an industrial participant, and are then aggregated at a server level. In the baseline Federated Learning approach, local models optimization fails to increase global accuracy with model aggregation in case there is significant statistical heterogeneity in the data distributions among clients. Thus the aim of this paper is the proposal of an enhanced strategy that accounts for adaptive local updates and the comparison of its performance with state-of-the-art techniques. Each participant computes the local stochastic gradients within an adaptive interval, set by the server at the aggregation step, when the models are loaded by the participants at the end of each communication round. The improved method is applied for bearing fault diagnosis and its effectiveness and accuracy are evaluated in the case of imbalanced class distribution in rolling bearing fault local data sets, i.e. considering a scenario where fault types are non-independent and identically distributed (non-i.i.d.) among clients. This case is addressed in literature to be one of the main challenges in FL and is of practical interest since skewed data sets are common in real-world factories.

1 Introduction

In recent years deep learning models have been intensively applied in fault machinery diagnosis problems due to the growing interest for machine learning in the field of pattern recognition [2]. The bearings fault diagnosis problem is particularly relevant for the industry due to the potential costs of unexpected production interruption in manufacturing environments and in the energy sector. Detecting an anomaly or identifying a fault in drive train bearings is still a challenging task due to the complex phenomenology of the degradation process and the several variables in an operational setting. Gathering great amount of data is essential in this application with data-driven approaches. Nevertheless, industries are not always willing to share the operational data of their machines and in some environments raw data communication has important limitations or implies additional costs. Federated learning is an emerging framework for training a deep neural network proposed by Google in 2016 [3] which promises to cross the mentioned barriers, accounting for data privacy related concerns and reducing communication costs when devices generate massive amount of data. For a fleet of cyber-physical systems such as industrial machines with edge devices, FL leads to the possibility of exploiting data sets from several industries to train a super model that outperforms state-of-the-art fault diagnosis neural networks trained on a limited portion of the global data set. The FL methodologies are divided in two main fields of application [4], the cross-device federated learning and the cross-silo federated learning. The cross-device federated learning is meant for a massive number of mobile or IoT devices, up to 10^{10}, where only a

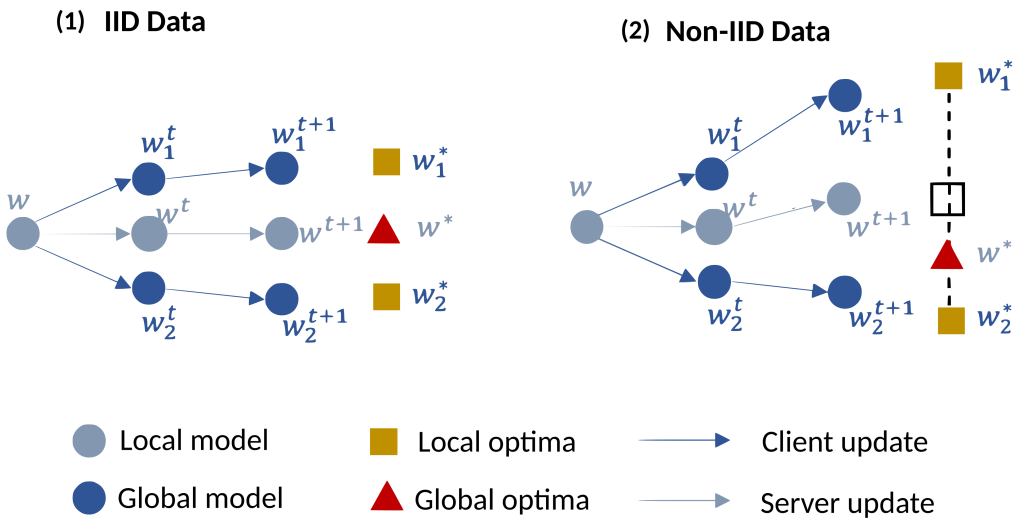


Figure 1: Clients drift behaviour in a two clients federated training [1]

fraction of clients are available over time. The cross-silo federated learning is suitable for data sets coming from different organizations (2-100 participants) almost always available. Important properties usually satisfied by participants in a cross-silo setting, in opposition to the cross-device one, are the following :
- Addressability: each client is identifiable and can be accessed by the system
- Statefullness: clients are likely participating all rounds of the computation, carrying state from round to round
- Reliability: the local training computation has a relatively low rate of failures
In this study a federated bearing fault diagnosis model is proposed for multiple industrial participants joining in a cross-silo setting. A further characterization of the federated learning approach relates to the data partitioning.

In case of multivariate data, clients can hold different sets of features or sensor signals: this case is referred as vertical federated learning. In many industrial applications where machines are monitored by the same set of variables, such as the vibration signals for the bearing fault diagnosis problem, the data partitioning is defined horizontal. Recently transfer learning techniques are joined in the federated learning framework [5]. One of the main challenges in training a federated learning model, especially in a cross-silo setting, refers to an heterogeneous data distribution. Among industrial participants different machine configurations and operative conditions implies non-i.i.d. data distribution causing client drifts, i.e. the local model updates toward local optimal solutions, resulting in performance degradation of the global model [6]. Figure 1 from [1] illustrates the implication of the client drift on the convergence to an optimal global solution. To deal with non-i.i.d. settings researchers have proposed alternative aggregation algorithms in recent years, such as FedDyn [7], FedProx [8] and Scaffold [9] to overcome the limitation of the baseline algorithm FedAvg [3]. In this study an adaptive variation of FedAvg is presented and compared with the Centralized model and the AFL [10], focusing on a new scheme for adjusting the aggregation interval to speed up convergence and to minimize communication costs.

2 Proposed methodology

2.1 Problem definition

The federated learning problem in rolling bearings faults diagnosis is tackled in this study with particular interest for the non-i.i.d. setting. The assumptions made in this framework are the following:
(1) Local data of the different participants are private, i.e. not shared during the training process. Nevertheless, the server receives information on the local data sets size before initializing the global model.
(2) The server initializes the local model with the initialized global model parameters $\mathbf{w}_g(n=1)$ and the local batch sizes.

(3) At the end of each round the server receives information regarding the results on the local validation set, such as the local validation loss and the local validation accuracy, from each participant.
(4) The server receives and aggregates the local models from all participants.

Each client holds a local training data set $D_k = \{(\mathbf{x}_j^k, y_j^k)\}_{j=1}^{|D_k|}$ and a local validation data set D_k^{val}, where k indicates the k-th participant. The j-th data sample $\mathbf{x}_j^k \in \mathbb{R}^L$ is a supervised input with j-th label y_j^k and length L. In this study the input $\mathbf{x}_j^k$ is a segment of a vibration signal associated with the faulty state of the machine component or with its normal/healthy state, assumed to be known during the experimental machine operation at a given working load.

2.2 Network architecture

The deep learning model chosen for the fault diagnosis task is a convolutional neural network that each client trains locally. Convolutional neural networks are well suited for machine fault diagnosis due to their property of learning patterns in timeseries such as monitored signals. Following [10], the network architecture is shown in Figure 2 and described in Table 1. The activation functions are Rectified Linear Units (ReLU). As will be described in section 2.3, the input is set to be bi-dimensional, as shown in the structure of the network layers. Following [10] the optimizer chosen for training of the model is the Stochastic Gradient Descent (SGD) with Momentum for faster convergence during local updating.

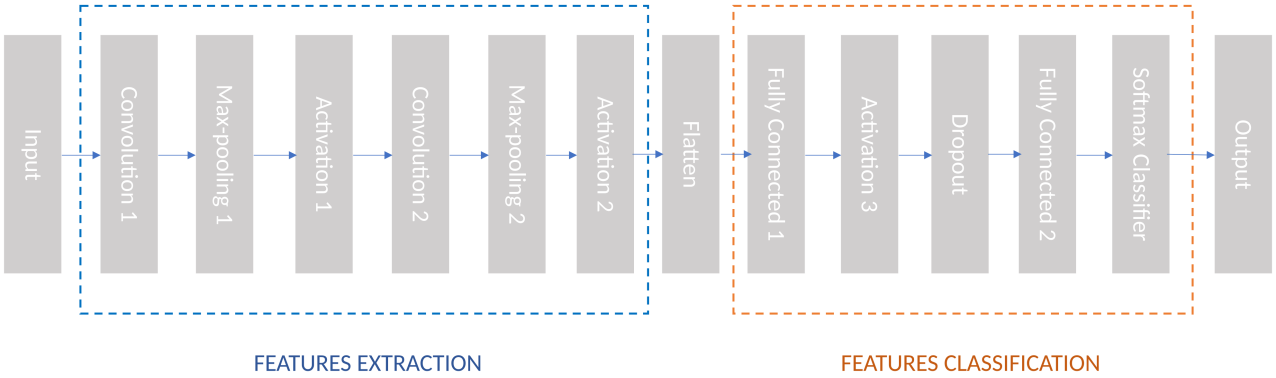


Figure 2: Sequential representation of the CNN layers

Layers	CNN structure training	CNN structure testing
Convolutional	2D $(5 \times 5, [1, 16])$	2D $(5 \times 5, [1, 16])$
Maxpooling	2D (2×2)	2D (2×2)
Convolutional	2D $(5 \times 5, [16, 32])$	2D $(5 \times 5, [16, 32])$
Maxpooling	2D (2×2)	2D (2×2)
Fully connected	$([960, 128])$	$([960, 128])$
Dropout	0.5	–
Fully connected	$([128, 10])$	$([128, 10])$

Table 1: Hyperparameters of the CNN layers

2.3 Input pre-processing

In order to realize a reliable fault diagnosis model, the monitored signals are pre-processed in order to ensure fast convergence of the network parameters to the desired optimum solution and to enhance the performance in the testing stage. Following [10] each vibration signal related to a state of the machine component at a given stable working condition is segmented in shorter data samples. The original data sample undergoes the transformations leading to the final input structure for the neural network. In detail, the pre-processing is composed by the following steps:

(1) The vibration signal is segmented in samples of length L data points. The segments are generated by a moving window of length L that can overlap the preceding segment of a random length with a minimum of 0 and a maximum of $L/2$. Part of the samples and their labels are associated to the participant holding them, defining the local training data set D_k and the local validation data set D_k^{val}. The remaining samples are hidden to all participants in the test data set D^{test} with their labels, being reserved for evaluation of the fault diagnosis model performance.
(2) The sample $\mathbf{x}$ is normalized with a standard pre-processing technique, subtracting from the sample the mean of their values and dividing by their standard deviation $\mathbf{x}' = \dfrac{\mathbf{x} - E(\mathbf{x})}{\sqrt{Var(\mathbf{x})}}$.
(3) The 1D sample $\mathbf{x}'$ is reshaped to provide the neural network with a 2D input $\mathbf{x}^{\diamond} \in \mathbb{R}^U \times \mathbb{R}^V$ where $U \times V = L$. For simplicity $\mathbf{x}^{\diamond}$ will be indicated as $\mathbf{x}$, which is the general input of the neural network.

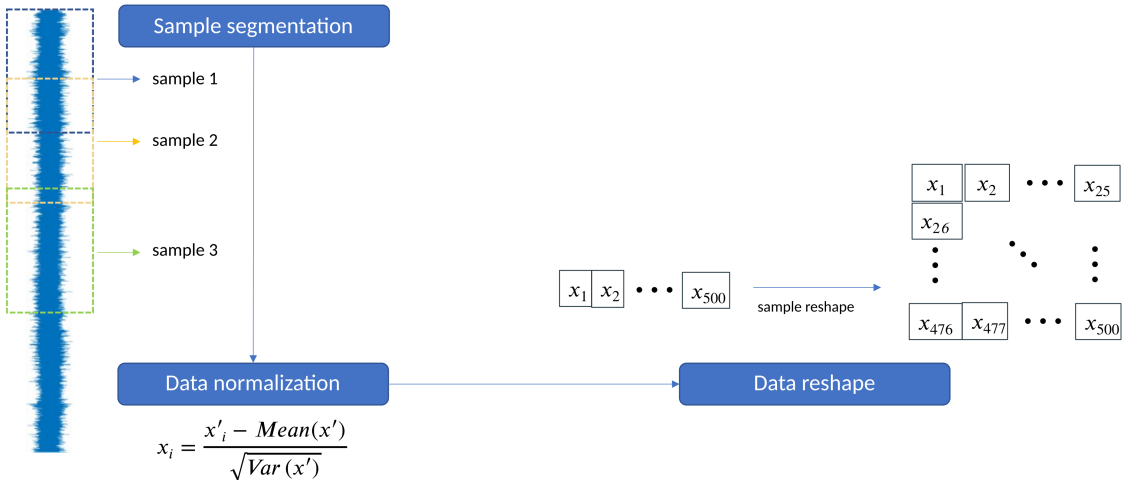


Figure 3: Sample segmentation and pre-processing

2.4 Federated learning approach

The aim of the federated learning approach is to provide a global fault diagnosis classifier $f_g(\mathbf{x}) = f(\mathbf{w}_g, \mathbf{x})$ resulting in a closer estimator of the perfect classifier than the local deep learning model $f_k(\mathbf{x}) = f(\mathbf{w}_k, \mathbf{x})$ trained on a local data set D_k. In order to achieve a global model, the participants, holding different private data sets, train deep learning models sharing the same architecture. In the proposed framework, the global model is obtained by aggregation of the weights of the local models $\mathbf{w}_k$, broadcasted by all participants $k = 1, \ldots, P$. The baseline algorithm for aggregation is the Federated Averaging (FedAvg) [3]. The pseudo code of FedAvg with momentum SGD is described in the Algorithm 1. In the adaptive framework each local batch size is set from the server depending on the local data set size that each client holds locally.

2.5 Proposed method

During the Federated Learning training each participant k broadcasts its local model parameters $\mathbf{w}_k(t = \tau)$ to the server at the end of the local training interval. Each participant trains its local model starting from the global model parameters $\mathbf{w}_k(t = 1) = \mathbf{w}_g(n)$ and performing a forward pass and backpropagation at each iteration $t = 1, \ldots, \tau$ on the local batch, with size

$$B_k = B_1 \frac{|D_k|}{|D_1|} \tag{1}$$

following [10], where $|D_k|$ is the size of the local training set of participant k. The local training followed by global aggregation goes on for a number of times N, equal to the total number of rounds. In this framework, the server also receives from each participant its local training set size $|D_k|$ before the initialization of the global

Algorithm 1 FederatedAveraging with momentum SGD

Server executes:
Initialize $\mathbf{w}_g(1)$
for all rounds n= $1,\ldots,N$ **do**
  for all clients $k = 1,\ldots,P$ in parallel **do**
    $\mathbf{w}_k(\tau) \leftarrow \text{ClientUpdate}(k, \mathbf{w}_g(n))$
  end for
  $\mathbf{w}_g(n+1) \leftarrow \sum_{k=1}^{P} \frac{|D_k|}{\sum |D_i|} \mathbf{w}_k(\tau)$
end for
ClientUpdate$(k, \mathbf{w})$: // *run on client k*
$\mathcal{B} \leftarrow$ (split D_k into batches of size B)
$\mathbf{w}(1) \leftarrow \mathbf{w}$
initialize $\mathbf{v}(1)$
for all local iterations $t = 1,\ldots,\tau$ **do**
  $\mathbf{v}(t+1) \leftarrow \gamma \mathbf{v}(t) + \nabla \mathcal{L}(f, \mathbf{w}(t), \mathbf{b}(t))$ where $\mathbf{b}(t) \in \mathcal{B}$ is the batch at iteration t
  $\mathbf{w}(t+1) \leftarrow \mathbf{w}(t) - \eta \mathbf{v}(t+1)$
end for
return $\mathbf{w}(\tau)$ to server

Algorithm 2 Adaptive FederatedAveraging

Server executes:
Initialize $\mathbf{w}_g(1)$
Initialize local batch size $B_k \leftarrow \text{LocalBatchSize}(B_1, \{|D_k|\})$ (equation 1)
Initialize $\tau \leftarrow \tau_{start}$
for all rounds n= $1,\ldots,N$ **do**
  for all clients $k = 1,\ldots,P$ in parallel **do**
    $\boldsymbol{a}_k^{val}(n) \leftarrow \text{LocalValidationAccuracy}(\mathbf{w}_g(n), D_k^{val})$ // *run on client k and return to server*
    $\mathbf{w}_k(\tau) \leftarrow \text{ClientUpdate}(k, \mathbf{w}_g(n), \tau, B_k)$
  end for
  $\mathbf{w}_g(n+1) \leftarrow \sum_{k=1}^{P} \frac{|D_k|}{\sum |D_i|} \mathbf{w}_k(\tau)$
  $\boldsymbol{a}_g^{val}(n) \leftarrow \text{GlobalValidationAccuracy}(\{|D_k|\}, \{\boldsymbol{a}_k^{val}(n)\})$ (equation 2)
  if $n > 1$ **then**
    $\boldsymbol{I}_a(n) \leftarrow \text{PerformanceIndex}(\boldsymbol{a}_k^{val}(n-1), \boldsymbol{a}_k^{val}(n))$ (equation 3)
  end if
  if $n = mW$ and $|\min(I_a^W)| > |\max(I_a^W)|$ and $\tau \neq 1$ ($I_a^W = \{I_a(n-W+2),\ldots,I_a(n)\}$, $m \in \mathbb{N}^+$) **then**
    $\tau \leftarrow \text{AggregationIntervalUpdate}(\tau_{start}, \boldsymbol{a}_g^{val}(n))$ (equation 6)
  end if
end for
ClientUpdate$(k, \mathbf{w}, \tau, B)$: // *run on client k*
$\mathcal{B} \leftarrow$ (split D_k into batches of size B)
$\mathbf{w}(1) \leftarrow \mathbf{w}$
initialize $\mathbf{v}(1)$
for all local iterations $t = 1,\ldots,\tau$ **do**
  $\mathbf{v}(t+1) \leftarrow \gamma \mathbf{v}(t) + \nabla \mathcal{L}(f, \mathbf{w}(t), \mathbf{b}(t))$ where $\mathbf{b}(t) \in \mathcal{B}$ is the batch at iteration t
  $\mathbf{w}(t+1) \leftarrow \mathbf{w}(t) - \eta \mathbf{v}(t+1)$
end for
return $\mathbf{w}(\tau)$ to server

model and the local batch sizes. Furthermore, the server receives with the local models the local accuracies $\boldsymbol{a}_k^{val}(n)$ of the global model on the local validation sets, computed before local training, at the beginning of each round $n \in \{1,\ldots,N\}$. The global accuracy is defined as:

$$\boldsymbol{a}_g^{val}(n) = \sum_{k=1}^{P} \frac{|D_k|}{\sum_i^P |D_i|} \boldsymbol{a}_k^{val}(n) \tag{2}$$

and is computed by the server at the aggregation step, where P indicates the number of participants. From the global accuracies of two consecutive rounds we can define an index to evaluate how the global models accuracies on the validation set are improving through aggregation rounds at the given aggregation interval:

$$I_a(n) = \frac{\boldsymbol{a}_g^{val}(n) - \boldsymbol{a}_g^{val}(n-1)}{1 - \max\left(\boldsymbol{a}_g^{val}(n), \boldsymbol{a}_g^{val}(n-1)\right)} \tag{3}$$

Due to fluctuations of the accuracies values, an observation window of width W is chosen to store W consecutives global accuracies values $\boldsymbol{a}_k^{val}(n-W+1),\ldots,\boldsymbol{a}_k^{val}(n)$ to compute at server level $\mathrm{W}-1$ values of I_a:

$$I_a^W = \left[I_a(n-W+2),\ldots,I_a(n)\right] \tag{4}$$

At server level every W rounds during the aggregation step the server checks if

$$\left|\min\left(I_a^W\right)\right| > \left|\max\left(I_a^W\right)\right| \tag{5}$$

or $\max\left(I_a^W\right) < 0$, which in practice is never verified. If the condition is satisfied, the aggregation interval is recomputed as:

$$\tau = \max\left(\left\lfloor \tau_{start}\left(1 - \boldsymbol{a}_g^{val}(n)\right)\right\rceil, 1\right) \tag{6}$$

where τ_{start} is the initial value of the aggregation interval and $\lfloor * \rceil$ is the "round to nearest integer" operator. The max operator is introduced for the case $\tau_{start}\left(1 - \boldsymbol{a}_g^{val}(n)\right) < 1$ which rarely verifies in practice. Once $\tau = 1$, the server deactivates the adaptive aggregation interval algorithm ensuring $\tau = 1$ till training ends.

3 Application of the methodology

3.1 Data set description

The methodology is applied to a data set for bearing fault diagnosis. The data set considered in this study is provided by the Case Western Reserve University [11] and is composed by vibration signals of an accelerometer conveniently placed to monitor the drive end (DE) bearing, presented in Figure 4. The accelerometer signals from this data set are segmented to generated samples from ten classes, one associated with the healthy state of the bearing and nine associated with three different type of faults and three different sizes for each fault type. The class labels are shown in Table 2. The data from the experimental tests at working load 0 HP are considered in this analysis.

From the original time signals samples are generated considering a window with $L = 500$ that includes one revolution of the shaft considering a rotational speed of 1797 rpm/min and a sampling frequency of 12 kHz. After normalization of the sample, this is reshaped in an input of size 20×25 for the first 2D convolutional layer of the neural network.

3.2 Non-i.i.d. experimental setting

In order to evaluate the effectiveness of the adaptive aggregation interval algorithm, the methodology is applied on three participants in a non-i.i.d. setting. The classes are distributed according to the following partition:

$D_1 = \{(\mathbf{x}_j^1, y_j^1)\}_{j=1}^{|D_1|}$ where $y_j^1 \in \{c_0, c_1, c_2, c_3, c_4\}$

$D_2 = \{(\mathbf{x}_j^2, y_j^2)\}_{j=1}^{|D_2|}$ where $y_j^2 \in \{c_5, c_6, c_7\}$

$D_3 = \{(\mathbf{x}_j^3, y_j^3)\}_{j=1}^{|D_3|}$ where $y_j^3 \in \{c_8, c_9\}$

In the same way the data sets D_k^{val} are defined for each participant. The number of samples generated for each class is described in the Table 3.

Label	Fault class	Severity (diameter, depth)	Working condition
c_0	-	-	0 HP
c_1	Inner race	0.18 mm, 0.28 mm	0 HP
c_2	Ball	0.18 mm, 0.28 mm	0 HP
c_3	Outer race	0.18 mm, 0.28 mm	0 HP
c_4	Inner race	0.36 mm, 0.28 mm	0 HP
c_5	Ball	0.36 mm, 0.28 mm	0 HP
c_6	Outer race	0.36 mm, 0.28 mm	0 HP
c_7	Inner race	0.53 mm, 0.28 mm	0 HP
c_8	Ball	0.53 mm, 0.28 mm	0 HP
c_9	Outer race	0.53 mm, 0.28 mm	0 HP

Table 2: Class labels and related fault description

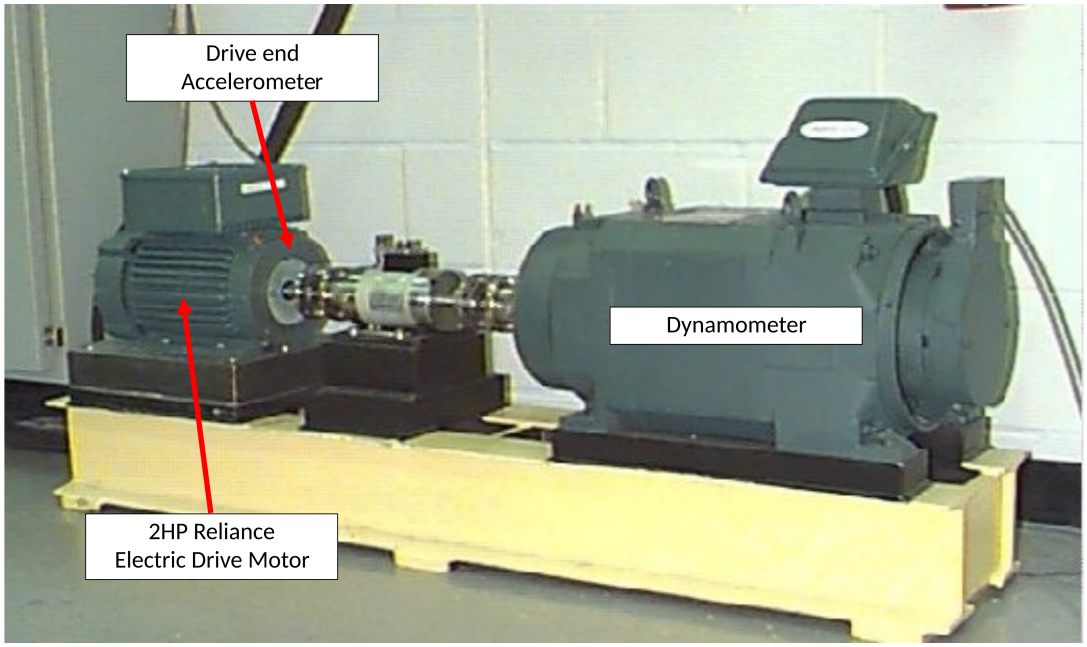


Figure 4: The experimental test rig configuration of the CWRU

3.3 Centralized model

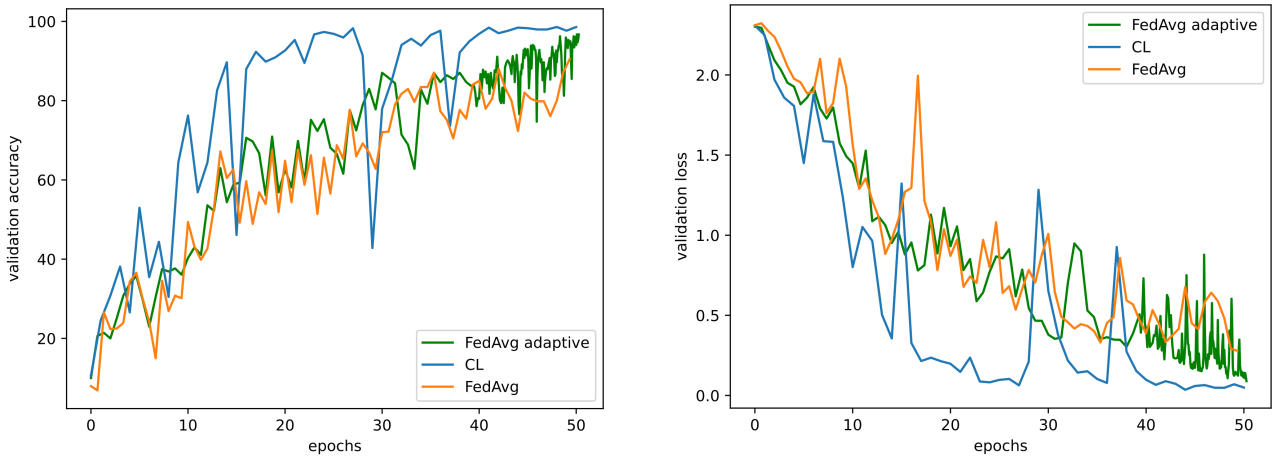
In order to evaluate the performance of our method the baseline centralized model is trained on the data set $D = D_1 \cup D_2 \cup D_3$ and tested on the data set D^{test}. The training is monitored on a validation set $D^{val} = D_1^{val} \cup D_2^{val} \cup D_3^{val}$. The batch size of the centralized model is set as $B = 128$. The learning rate and the momentum term are set respectively to $\eta = 0.05$ and $\gamma = 0.5$. For training and testing the cross-entropy function has been adopted for loss function computation.

3.4 Federated learning methods

The Adaptive Federated Averaging method is applied to the three participants considering a batch size $B_1 = 64$ for the first participant with the larger data set. The cross-entropy loss function has been adopted. The learning rate and the momentum term are set respectively to $\eta = 0.05$ and $\gamma = 0.5$ as for the centralized model. For the Adaptive Federated Averaging method described in Algorithm 2, in this application $W = 6$ and $\tau_{start} = 10$. The proposed method is compared with the Adaptive Federated Learning (AFL) from [10], which includes the personalized batch size and an adaptive aggregation interval to reduce communication costs. The AFL model is tested with the same architecture and the same hyperparameters of the proposed methodology, with adaptive aggregation interval parameters following [10], i.e. $W = 15$, $\delta = 5$, $decay = 0.95$. The proposed algorithm needs one additional parameter instead of the three mentioned, reducing complexity.

	D_k local training data set	D_k^{val} local validation data set	D^{test} global test set
Number of samples for class c_i	192	64	64
Number of classes for data set	5,3,2	5,3,2	10

Table 3: Samples and class distribution description



(a) Global validation accuracy

(b) Global validation loss

Figure 5: Training performance on the validation data set for FedAvg, CL and the proposed method

3.5 Results

The centralized, the Adaptive FedAvg and the AFL monitored global validation loss, the global validation accuracy and the aggregation interval are shown in Figures 5, 6, 7. Similarly to the validation accuracy, the global validation loss is computed at server level from the validation loss evaluated from each participant on the local validation set at round n:

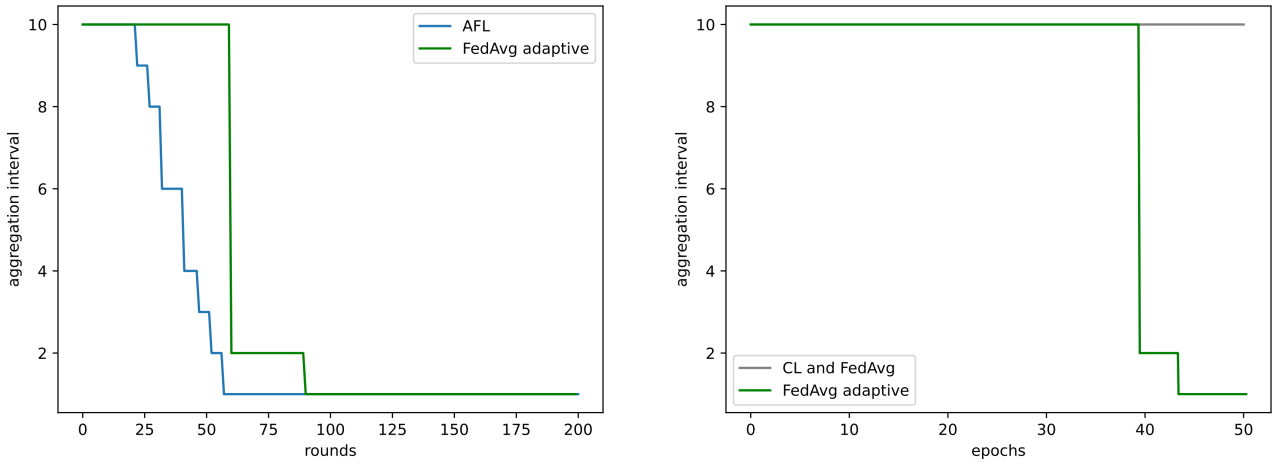
$$\mathcal{L}_g^{val}(n) = \sum_{k=1}^{P} \frac{|D_k|}{\sum_i^P |D_i|} \mathcal{L}(f, \mathbf{w}_g(n), D_k^{val}) \tag{7}$$

Where $\mathcal{L}(f, \mathbf{w}_g(n), D_k^{val}) \coloneqq \frac{1}{|D_k^{val}|} \sum_{(\mathbf{x},y)\in D_k^{val}} \mathcal{L}(f, \mathbf{w}_g(n), \mathbf{x}, y)$

The models have been implemented in Fed-BioMed v.4.3 [12]. For the adaptive methods, in order to select the final model to be tested, when $\tau = 1$ the server stores the global model parameters $\mathbf{w}_g(n)$ for each round n with the global validation loss. At the end of the training the global model with minimum global validation loss is retained for testing. The Table 4 shows the test results on four metrics: Precision, i.e. the average for all class of the fraction of instances correctly classified as belonging to a specific class out of all instances the model predicted to belong to that class, Recall, i.e. the average for all class of the fraction of instances in a class that the model correctly classified out of all instances in that class, Accuracy, i.e. the proportion of correctly classified cases from the total number of objects in the data set, and F1-score, i.e. the weighted average of Precision and Recall.

Results of the Federated Learning models on the test set						
Model	Precision	Recall	Accuracy	F1-Score	Round	Epochs
CL	0.979828	0.978125	0.978125	0.978168	-	50
FedAvg adaptive	0.971875	0.973255	0.971875	0.971860	193	~ 50
AFL	0.956630	0.9484375	0.9484375	0.9483477	196	~ 36
FedAvg($\tau = 10$)	0.907337	0.8890625	0.8890625	0.888795	75	50

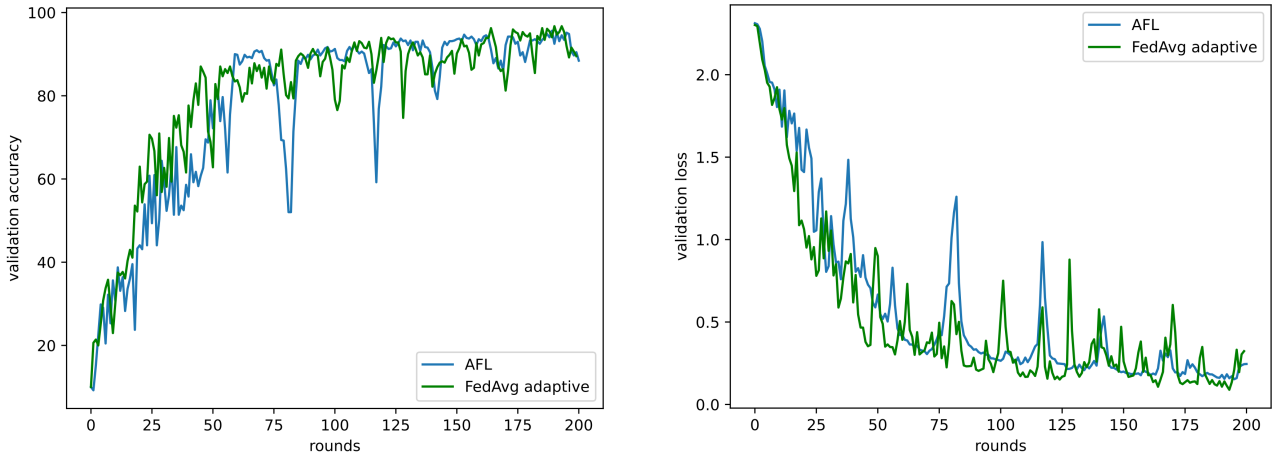
Table 4: Performance on the test data set



(a) aggregation interval along rounds

(b) aggregation interval along epochs

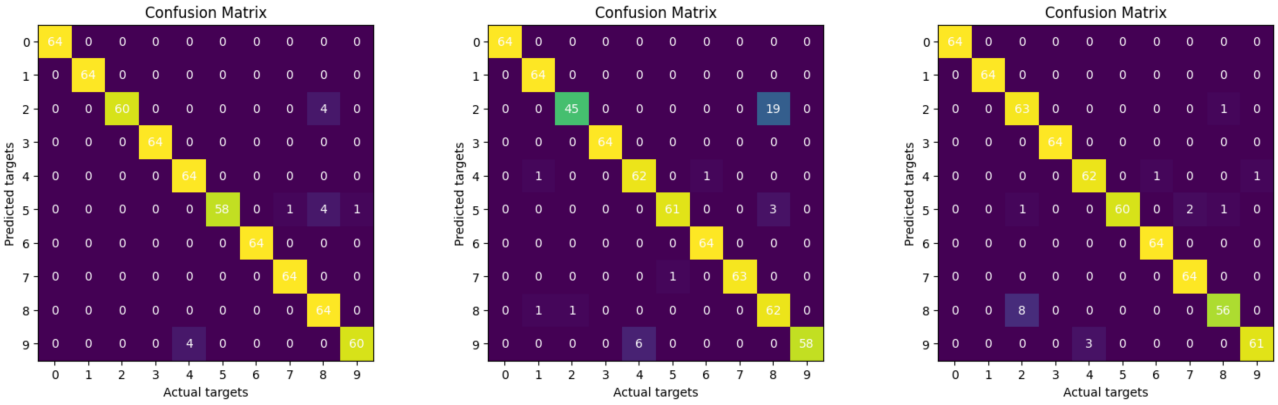
Figure 6: Adaptive aggregation interval during training for the compared models



(a) Global validation accuracy

(b) Global validation loss

Figure 7: Training performance on the validation data set for the AFL and the proposed method



(a) Centralized model

(b) AFL

(c) Adaptive FedAvg

Figure 8: Confusion matrix comparison between the centralized and the federated models

3.6 Conclusions

The Adaptive Federated Learning method has been compared to the AFL method to validate an algorithm that adaptively adjusts the aggregation interval for reducing communication costs and for enhancing convergence rate. The method has been tested on a public data set for bearing fault diagnosis, partitioned in a non-i.i.d fashion in a three clients setting. In future development of the methodology personalized federated learning (PFL) [13], which aims to provide a personalized model for each client, and the federated transfer learning (FTL) are worth to be investigated for the bearing fault diagnosis problem in a cross-silos scenario.

3.7 Acknowledgements

The authors gratefully acknowledge the European Commission for its support of the Marie Sklodowska Curie program through the H2020 ETN MOIRA project (GA 955681)

References

[1] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2022.

[2] Jinyang Jiao, Ming Zhao, Jing Lin, and Kaixuan Liang. A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing*, 417:36–63, 2020.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[4] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[5] Junbin Chen, Jipu Li, Ruyi Huang, Ke Yue, Zhuyun Chen, and Weihua Li. Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[6] Yong Shi, Yuanying Zhang, Yang Xiao, and Lingfeng Niu. Optimization strategies for client drift in federated learning: A review. *Procedia Computer Science*, 214:1168–1173, 2022. 9th International Conference on Information Technology and Quantitative Management.

[7] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[8] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020.

[9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal DaumÃ© III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020.

[10] Zehui Zhang, Xiaobin Xu, Wenfeng Gong, Yuwang Chen, and Haibo Gao. Efficient federated convolutional neural network with information fusion for rolling bearing fault diagnosis. *Control Engineering Practice*, 116:104913, 2021.

[11] Hai Qiu, Jay Lee, Jing Lin, and Gang Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289(4):1066–1090, 2006.

[12] Francesco Cremonesi, Marc Vesin, Sergen Cansiz, Yannick Bouillard, Irene Balelli, Lucia Innocenti, Santiago Silva, Samy-Safwan Ayed, Riccardo Taiello, Laetita Kameni, Richard Vidal, Fanny Orlhac, Christophe Nioche, Nathan Lapel, Bastien Houis, Romain Modzelewski, Olivier Humbert, Melek Ãnen, and Marco Lorenzi. Fed-biomed: Open, transparent and trusted federated learning for real-world healthcare applications. 2023.

[13] Lei Yang, Jiaming Huang, Wanyu Lin, and Jiannong Cao. Personalized federated learning on non-iid data via group-based meta-learning. *ACM Trans. Knowl. Discov. Data*, 17(4), mar 2023.