Remaining Useful Life Estimation in Aircraft Components with Federated Learning

Raúl Llasag Rosero, Catarina Silva, and Bernardete Ribeiro

University of Coimbra, CISUC - Department of Informatics Engineering, Coimbra, Portugal
{*rosero,catarina,bribeiro*}*@dei.uc.pt*

ABSTRACT

In this work, we compare distributed collaborative learning techniques for Prognostic and Health Management (PHM) systems, focusing on predictive aircraft maintenance. Aircraft industry components are usually evaluated using remaining useful life (RUL) estimations to describe the amount of time left before system health falls. Such estimates have been commonly achieved with traditional degradation estimation methods. These estimation methods have been widely applied in centralized processing architectures, limiting the scalability of PHM systems.

Concerns about data privacy and transfer of large amount of data have also been limiting the construction of decentralized processing architectures. Nevertheless, with the emergence of collaborative training methods of machine learning models, e.g. Federated Learning (FL), the previous referred concerns have being tackled by privacy-preserving communications while keeping data at the network edges. However, the effectiveness of federated learning algorithms using time-series data for prognostic and health management of aircraft systems has been minimally explored.

In this work, we use feed-forward neural networks on centralized and decentralized scenarios to compare the prediction error minimization of FL algorithms, such as, Federated Average (FedAvg) and Federated Proximal Term (FedProx). Our experiments take into account gradient descendent minimization and averaging weights of neural networks. Using FedAvg, we obtained similar prediction errors to the centralized scenario but presenting uncertain predictions along the aggregation iterations. On the other hand, using FedProx, the prediction error curve progressively decreases along the aggregation iterations if μ takes values ~ 0.01.

1. INTRODUCTION

The Prognostics field has been growing in parallel with Condition Based Maintenance (CBM) of system failures (Saxena et al., 2008). CBM identifies two types of failures for maintenance activities: critical and non-critical. Critical failures, which are usually solved replacing components after their run-to-failure, are avoided by preventive activities such as periodic or non-periodic maintenance tasks. Non-periodic maintenance approaches are usually based on Prognostic and Health Management (PHM) technologies (Canh, Kwok, Carl, Romano, & George, 2009).

PHM technologies combine diagnostics and prognostics of machinery component failures (Saxena & K., 2008). Diagnostics detects and isolates failures, while prognostics predicts future state or Remaining Useful Life (RUL) (Canh et al., 2009; Saxena & K., 2008). Using prognostics, companies have been reducing not only maintenance costs but also maintenance times.

The use of prognostics on aircraft maintenance is an important challenge for the aeronautics field because, at this moment, the replacement of parts of an aircraft is done either at fixed time intervals or after a failure (Azevedo, Ribeiro, & Cardoso, 2019). Airlines use this approach to ensure reliability and safety as detailed in regulatory laws, but usually fall short in optimizing the useful lifetime of their aircraft components.

Airlines have been identifying the enormous potential of prognostics on the degradation of components over time to assure the safety of airplanes and of its passengers. Additionally, airlines have also considered to interact using intensively collaborative processes supported by information technology, namely, Collaborative Networks (Simões & Soares, 2008). Collaborative networks have been implemented mainly by technological companies to better achieve common or compatible goals. Nevertheless, data security on these collaborative networks and the data sharing dependence have been concerns for companies which could directly affect lives, in the case of airlines.

Today, collaborative networks that avoid confidential data sharing and improve performance of a common task through artificial intelligence models aggregations are gaining interest and feasibility, namely through Federated Learning (FL) approaches (Yang, Liu, Chen, & Tong, 2019).

Federated Learning frameworks such are PySyft, PaddleFL, Tensorflow Federated (TFF), Federated AI Technology Enabler (FATE), Nvidia Clara, and others allow network constituents to collaboratively fit/learn a shared model while keeping data privacy in network nodes. These frameworks (Palau, Bakliwal, Dhada, Pearce, & Parlikad, 2018) avoid data sharing dependence of distributed collaborative prognostics. However, such frameworks usually only implement a limited number of federated learning techniques, restricting to collaborative prognostics systems and neural networks for using a basic federated algorithm which was named as Federated Averaging (FedAvg) (Sahu et al., 2019; Wang, Yurochkin, Sun, Papailiopoulos, & Khazaeni, 2019), being Turbo-FL-POC an example of them.

In this work, we use an aircraft multi-sensor time-series data to compare the performance achieved by different Federated Learning algorithms. Our experiments take into account gradient descendent minimization and averaging weights of feed-forward neural networks. So, in this federated introductory analysis, the RUL prediction accuracy obtained using FedAvg (McMahan & Ramage, 2017) and Federated Proximal Term (FedProx) algorithms are compared (Sahu et al., 2019).

Our experiments show that FL algorithms exhibit similar performance on RUL estimation when compared with data centralized models, where the main model is iterative centrally optimized. Also, our experiments show more learning oscillations in FedAvg than FedProx when its hyper-parameter μ was adequately chosen.

The rest of this paper is organized as follows. Section 2 includes the problem description, the dataset chosen, the feature space selection, and considerations on pre-processing. Section 3 details the techniques that can be applied on Federated Learning algorithms and the proposed approach for RUL estimation. Section 4 includes the centralized scenario setup, analyses the RUL estimation and the construction of the collaborative scenarios. Section 5 presents the proposed collaborative FL algorithms and the results achieved. Finally, Section 6 presents conclusions and possible future works.

2. PROBLEM DESCRIPTION

In this work, a Condition Based Maintenance problem is used, where the system condition is evaluated with muti-sensor data to detect failures for *a posteriori* prediction of Remaining Useful Life (RUL) of system components for specific flight trajectories.

In this particular problem, we use the term trajectory to refer a group of flight trajectories. Each trajectory is composed by a set of operational cycles which take similar states when flights restart, while the RUL refers to the difference between the current cycle and the cycle when the component becomes totally inoperable. For RUL prediction, artificial intelligence methods can be used, more specifically machine learning methods that learn from data to increase the problem-solving abilities of models when data is showed a few times.

Federated Learning algorithms facilitates the integration of different CBM systems if data-driven approaches and machine learning techniques are adopted (Hu, Sun, Chen, & Lu, 2019). However, the feature space of datasets needs to be previously explored before constructing collaborative models.

The absence of deeply explored datasets of aeronautic companies has been the bottleneck for the construction of collaborative models. Hence, we consider using Turbofan Engine Degradation Simulation dataset (Saxena & K., 2008), which have been analyzed by some researchers, especially on the PHM08 Challenge (Ramasso & Saxena, 2014).

2.1. Dataset Description

Turbofan datasets were provided by the Prognostics CoE at NASA Ames and are composed by the output of a engine degradation simulation carried out using Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) (Liu, Frederick, De Castro, & Litt, 2012). It consists of four datasets, namely: FD001, FD002, FD003 and FD004. All four have the same structure which corresponds to: trajectory id, cycle count, 21 sensors, and 3 operating settings. These 3 operating settings are Altitude, Mach Number and Throttle Angle Resolver variations on flight trajectories. According to (Leto, 2008), combinations of these operation settings refer to operating modes (or regimes).

Six operating regimes have been identified on FD002 and FD004 datasets, while the other only operate on one operating regime (Leto, 2008). According to (Azevedo et al., 2019), the main difference between datasets is the number of fault modes. FD004 presents two fault modes while FD002 presents only one fault mode. FD004 is potentially more complex, and was therefore we chosen for our experiments.

Turbofan datasets provide data in two splits, namely: training and testing. Also, turbofan provides a ground truth, which corresponds to the RUL evaluated on the last cycle of each trajectory of the testing split. FD004 contains 249 trajectories for training and 248 trajectories for testing. Thus, the ground truth for FD004 contains 248 RUL values.

2.2. Pre-processing data

Each trajectory on Turbofan datasets is composed of a set of operational cycles. However, these datasets only provide the

RUL associated to the last operational cycle of each trajectory. Hence, researchers have been applying some techniques to assume a RUL degradation curve. The curve degradation assumption is explained in more detail in Section 2.2.1.

A subgroup of operational cycles of each trajectory represents an aircraft flight. Each flight can be identified by the variation of operational cycles on operating regimes, six regimes specifically for the FD004 dataset.

According to (Leto, 2008; Olivares, Gonzalez, Tovar, & Gorrostieta, 2019), there is no mathematical dependency between sensor data, which could support machine learning models training. Thus, (Leto, 2008) has proposed a normalization based on clustering of these six operating regimes, normalization which is refereed in Section 2.2.2. Although a mathematical dependency on regimes can be found, not all the sensors exhibit this dependency. Hence, a feature space selection is proporsed in Section 2.3.

2.2.1. RUL assumption

In (Heimes, 2008) two degradation approaches are proposed, which are illustrated in Figure 1.

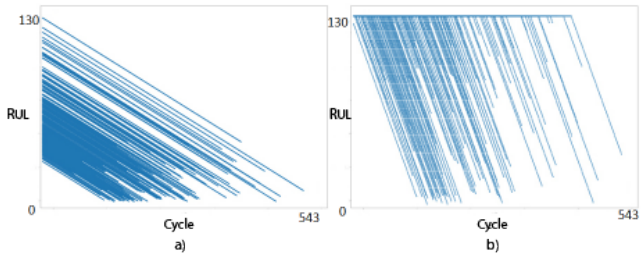

Figure 1. (a) Linear and (b) piece-wise degradation of FD004 test dataset

The first degradation approach that illustrates a family of linear degradation curves, corresponds to Eq. (1).

$$f(t) = t_{EoL} - t_i \tag{1}$$

Here, t_{EoL} corresponds to the RUL of the last operational cycle per trajectory and t_i corresponds to the current cycle. The second degradation approach that illustrates a family of piece-wise curves, was adopted by (Heimes, 2008; Olivares et al., 2019) to reproduce the third winner technique (Leto, 2008) of the PHM08 Challenge (Ramasso & Saxena, 2014). This piece-wise approach corresponds to Eq. (2).

$$f(t) = \begin{cases} Rc & if \quad 0 \leq t_i \leq t_{SoF} \\ t_{EoL} - t_i & if \quad t_{SoF} \leq t_{EoL} \end{cases} \tag{2}$$

Here, Rc is the initial constant value of RUL which varies from 120 to 130 cycles (Heimes, 2008); SoF is the time when engines start to failure is equal to $t_{EoL} - t_i$. After experimentation (Heimes, 2008; Olivares et al., 2019) limited Rc to take a value of 130, value which we adopted for our experiments.

2.2.2. Normalization

Before the normalization step, a clustering process was carried out using K-means algorithm. This algorithm was used by (Ramasso & Saxena, 2014; Olivares et al., 2019) among others to add a regime identifier r for each operational cycle x. After that, a data normalization per regime $N(.)$ described in Eq (3) was applied.

$$N(x^{(r,f)}) = \frac{x^{(r,f)} - \mu^{(r,f)}}{\sigma^{(r,f)}} \tag{3}$$

For the sensor f on regime r, $x^{(r,f)}$ represents the sensor data per regime, $\mu^{(r,f)}$ and $\sigma^{(r,f)}$ are the mean and standard deviation, respectively. After that process, (Olivares et al., 2019) considered the normalization of the three operating settings, the cycle number and the target (RUL), without considering the regime variation. Given the performance improvements, we also apply the method in (Olivares et al., 2019).

2.3. Feature Space selection

In (Sahu et al., 2019) and literature referring to the Turbofan Engine Degradation dataset, the sensors commonly selected are those whose data has an increasing or decreasing behavior. Those sensors are: 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21 (Olivares et al., 2019). Thus, we also use this feature space for the centralized model scenario construction.

3. MACHINE LEARNING TECHNIQUES

In the PHM08 Challenge, some techniques have been applied for RUL estimation (Ramasso & Saxena, 2014). The first three places of this challenge have used a Health Indicator (HI) as Principal Component Analysis (PCA) with kernel smoothing; Convolution Neural Networks (CNN) with a Kalman Filter (KF); and a Multilayer Perceptron (MLP) with a Kalman Filter (KF), respectively.

In this work, a Multilayer Perceptron (MLP) was chosen because this approach is implemented in the approaches on the challenge referred above. Additionally, MLP is the canonical example for Federated Learning, where weights of same layers but different models are averaged using the FedAvg algorithm.

3.1. Proposed approach

The MLP network used here is composed of three layers. The input layer takes data from 18 features, corresponding to 14 sensors, 3 operating settings and the operating cycle. The hidden layer contains 10 nodes with a sigmoid activation function. Finally, the output layer contains only one node, which is activated by a linear function. To train the MLP, an RMSE

loss function described in Eq (4) was used.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(RUL_i - R\hat{U}L_i)^2}, \tag{4}$$

where m is the number of samples, RUL is the ground truth for the sample i and $R\hat{U}L_i$ is the remaining useful life inferred with a lot of noise associated. Thus, in (Olivares et al., 2019), a Kalman Filter on the inference process was used to reduce this noise. We use this filter over other signal processing techniques, because through an iterative process and data measurements with uncertainty, this technique has generated the best estimates of the variable of interest (Heimes, 2001; Olivares et al., 2019).

3.2. Kalman Filter

The Kalman Filter (KF) described in Algorithm 1 was applied in the inference of n trajectories, basically consisting of two stages: prediction and update (Olivares et al., 2019). Those stages are composed from the second to the eighth step.

Algorithm 1 Kalman Filter for n trajectories

1: **for** $trajectory = 1, 2, \dots n$ **do**
2: **for** $k = 1, 2, \dots trajectory_{dim}$ **do**
3: $\hat{x}_k^- = \hat{x}_{k-1}$
4: $P_k^- = P_{k-1} + Q$
5: $K_k = \frac{P_k^-}{P_k^- + R}$
6: $\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-)$
7: $P_k = (1 - K_k) * P_k^-$
8: **end for**
9: $R\hat{U}L = \hat{x} * Rc$
10: **end for**

3.2.1. Prediction Stage

Here, $\hat{x}_k^-$ is the *a priori* estimate of the state vector $\hat{x}$ at time k, P_k^- is the *a priori* error estimate matrix, P is the *a posteori* error estimate matrix and Q is the degradation rate.

The initial conditions for the KF are $\hat{x}_0 = 1$ and the degradation rate $Q = 1/209$, corresponding to 209 flight cycles in average (Olivares et al., 2019). Here, we are not only assuming a normalized remaining useful life with an initial value of $\hat{x}_0 = 1$ but also assuming a null initial degradation error $P_0 = 0$.

3.2.2. Update Stage

The update of the state vector $\hat{x}$ and the *a posteori* error estimate matrix P_k depends basically on the value of the gain K. After an heuristic evaluation, (Olivares et al., 2019) adopted a $\sigma_z = 0.3$. So, the estimate of measurement variance $R = \sigma_z^2$ corresponds to $R = 0.09$.

Finally, on the 9th step, the $R\hat{U}L$ prediction is the result of the product of the initial constant Rc and the prediction normalized $\hat{x}$.

4. SCENARIOS

In order to evaluate the decentralized collaborative algorithms, we consider a variant selection of the centralized scenario. In Section 4.1 the linear and the piece-wise degradation for a selection of the centralized scenario are detailed, using all the data to train an unique node.

In Section 4.2, the training data was divided in nodes under some conditions, to simulate a Federated Learning approach using FedAvg and FedProx algorithms in a collaborative training scenario.

4.1. Centralized scenario

For the centralized scenario, the MLP network model was trained using the 85% and tested using the 15% of all data of the FD004 training split . To do that, the learning rate was set to $\eta = 0.001$ while the RMSE loss function described in Eq. 4 was used. Also, an early stopping callback was used, waiting for 10 epochs before stop if the validation didn't progress. This number of epochs is usually known as $patience$.

The training was processed on a virtual machine with 4GB of RAM and 4vCore, this process was done in 10 minutes approximately. However, the estimated values of the MLP illustrated in Figures 3 and 2 have required the KF to obtain a smoother or filtered curve (Olivares et al., 2019).

After obtaining smoothed inferred $R\hat{U}L$ samples, these m samples were compared with the ground truth to measure the performance of the algorithm. To do that, the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) metrics described in Eqs. (5,6) were used.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|RUL_i - R\hat{U}L_i| \tag{5}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(RUL_i - R\hat{U}L_i)^2 \tag{6}$$

The training and inferring processes were applied in two approaches for a variant centralized scenario selection. The first one refers to a linear degradation, while the second one refers to a piece-wise degradation. Figure 1 illustrates these two approaches on the FD004 test dataset, where EoL of trajectories takes different values.

Figures 3 and 2 illustrate the RUL predictions for the first and second trajectories using the degradation approaches de-

Degradation	MAE	MSE	RMSE
Linear	60.93	6983.06	83.56
Piece-wise	18.13	651.89	25.53

Table 1. Centralized scenario results

scribed in Table 1. Here, the green curve represents the ground truth, the red curve the RUL inferred by the MLP and the blue curve represents the smoothed RUL using the KF. Due to space limitations, the rest of the trajectories are not illustrated, but the evaluation metrics MAE, MSE, and RMSE were considered for all of them.

In Table 1 the results for the linear and the piece-wise degradation approaches are presented, where the piece-wise approach obtains considerable less difference compared to the ground truth. Therefore, we consider the use of a piece-wise degradation approach for the collaborative scenarios.

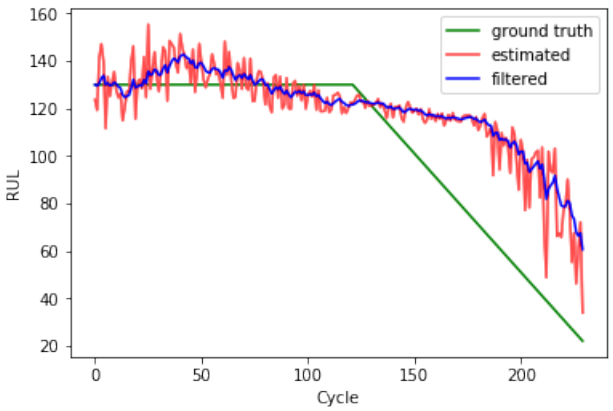

Figure 2. Piece-wise RUL degradation of 1st trajectory

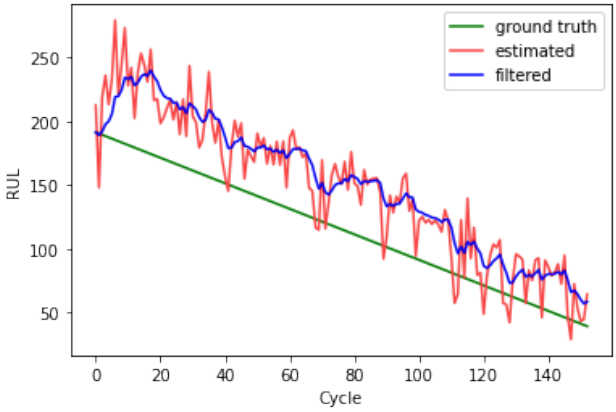

Figure 3. Linear RUL degradation of 2nd trajectory

4.2. Decentralized collaborative scenarios

For the construction of the decentralized collaborative scenarios, the data of the training split was separated in nodes. To do that, a clustering per trajectory identifier process was carried out with the use of the module operation, taking to trajectory identifier as the divided and the N number of nodes as the divisor. In Table 2, three data partitions were considered, where the main difference is the number of nodes and the number of trajectories per node.

Scenario	N Nodes	Trajectories
Centralized scenario	1	249
1st Decentralized	2	125+124
2nd Decentralized	4	63+62X3
3th Decentralized	8	32+31X7

Table 2. Collaborative scenarios

5. FEDERATED LEARNING

Decentralized collaborative prognosis has been considering the aggregation of machine learning models and features for improving prognostic accuracy. The feature aggregation proposed by (Hu et al., 2019) has considered a unique sensor type, reason for the wich, in this work, we aggregate models trained with multi-sensor data. The Federated Learning algorithms used in this work are described in the Section 5.1.

Before comparing federated algorithms, we compared the data divisions of the collaborative scenarios described in the Table 2, using the most basic FL algorithm referred in the Section 5.1.1.

The comparison referred in the Section 5.2 was useful to select a data division which provide less error predictions in less training steps, being useful to save time on future evaluations. Finally, the comparison results of federated learning algorithms over the scenario chosen on Section 5.2 are discussed in Section 5.3.

5.1. Algorithms

Basic collaborative algorithms such as Federated Learning Averaging (FedAvg) (McMahan & Ramage, 2017) and Federated Learning Proximal Term (FedProx) (Sahu et al., 2019) take into account the gradient descent minimization and averaging weights of feed-forward neural networks, i.e., our MLP approach is well suited. Therefore, we evaluate them on the collaborative scenarios described in the Table 2.

5.1.1. Federated Averaging

In Federated Averaging (FedAvg) algorithm (McMahan & Ramage, 2017), the local surrogate of the global objective function at node i is $F_i(.)$, and the local solver is Stochastic Gradient Descent (SGD), with the same learning rate η and number of local epochs E used on each node.

Algorithm 2 Federated Averaging

Input: Average Iterations T, Learning rate η, Epochs E, Initial weights w^0, Number of nodes N, Number of samples in node$i$ n_i
Output: Global weights w_i^{t+1}
for $t = 1, 2, \dots T-1$ **do**
  The server sends W^t to nodes
  for $i = 1, 2, \dots N$ **do**
    Using n_i, each device i updates W^t for E epochs of SGD on $F_i(w)$ with learning rate η to obtain w_i^{t+1}
    Ech device sends w_i^{t+1} back to server
  end for
  The server aggregates weights as $w^{t+1} = \frac{\sum_{t=1}^{N} w_i^{t+1}}{N}$
end for

At each average iteration t, each node i runs SGD locally for E number of epochs, and then the resulting model updates are averaged. These processes are iterative repeated at $T-1$ times, when the central model achieves performances expected.

5.1.2. Federated Proximal Term

Federated Proximal Term (FedProx) (Sahu et al., 2019) has the same steps of FedAvg, except at the 10th step. In FedProx, the local surrogate of the global objective function of FedAvg has several changes. Instead of minimizing a function $F(.)$, the surrogate objective to be minimized of FedProx is $h(w, w^t) = F_i(w) + \frac{\mu}{2}||w - w^t||^2$.

According to (Sahu et al., 2019), the proximal term given by μ was beneficial in two aspects. The first one addresses the issue of statistical heterogeneity by restricting the local updates to be closer to the central model without any need to manually set the number of local epochs.

Algorithm 3 Federated Proximal Term

Input: Average Iterations T, Learning rate η, Epochs E, Initial weights w^0, Number of nodes N, Number of samples in node$i$ n_i, Proximal Term μ
Output: Global weights w_i^{t+1}
for $t = 1, 2, \dots T-1$ **do**
  The server sends W^t to nodes
  for $i = 1, 2, \dots N$ **do**
    Using n_i, each device i finds w_i^{t+1} which is an t-inexact minimizer of $w_i^{t+1} \approx \arg min_w h(w, w^t) = F_i(w) + \frac{\mu}{2}||w - w^t||^2$
    Ech device sends w_i^{t+1} back to server
  end for
  The server aggregates weights as $w^{t+1} = \frac{\sum_{t=1}^{N} w_i^{t+1}}{N}$
end for

The second beneficial aspect refers to safely incorporating variable amounts of local work resulting from different computational resources on nodes. Due to different lengths of trajectories, different system resources of airplanes and high performance requirements, FedProx has been chosen for our experiments.

5.2. Decentralized collaborative scenario selection

The global models of collaborative scenarios described in the Table 2 were trained using the FedAvg algorithm to choose the best data partition for future evaluations. Global models illustrated in the Figure 4 and described on Table 3 was trained with a number of iterations $T = 12$, a learning rate $\eta = 0.001$ and a maximum number of $E = 180$. Here, we also used an early stopping callback with $patience = 10$ to prevent problems related with over fitting.

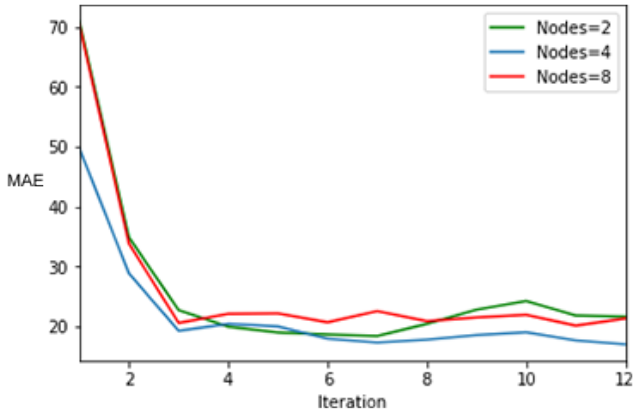

Figure 4. FedAvg using different N Nodes

The horizontal axis of the Figure 4 represents the number of iteration T, while the vertical axes represents error predictions of global models on MAE values. Error predictions of global models were evaluated using all trajectories of the testing dataset. In this figure, a data partition on 4 nodes is the best collaborative scenario, because it exhibits better MAE values than the other scenarios. Also, in the Table 3, the second collaborative scenario obtained less error predictions on less T number of iterations. Thus, this scenario was considered to evaluate the FedProx algorithm.

Using the FedAvg algorithm, all the decentralized scenarios described in Table 3 improved the performance of the centralized scenario at iteration T . However, there is no procedure to take the best prediction model at a T iteration which is not previously known.

5.3. Decentralized collaborative results

The experiments described in the Table 3 show that Turbofan datasets can be used on decentralized collaborative scenarios, but do not show FedAvg as the best decentralized collaborative algorithm. In Figure 5, the confidence principle has been severally affected because prediction curves oscillate around

Scenario	Nodes	Less Prediction Error			
		MAE	MSE	RMSE	T
Centralized	1	18.13	698.06	83.56	1
1st Decentralized	2	17.17	554.00	23.53	11
2nd Decentralized	4	16.41	437.14	20.90	3
3th Decentralized	8	18.70	572.43	23.92	3

Table 3. Collaborative Scenarios Results

an optimal prediction errors referred in the group of columns related to Errors on the Table 3. Thus, we consider the use of FedProx algorithm to analyze the convergence curve of the RUL prediction.

Principles of FedProx are based on the model convergence along time without any need to manually set the number of local epochs. Hence, we evaluated to FedAvg algorithm using a same number of E epochs to train nodes models described on the second decentralized scenario.

The prediction performance was seriously affected after setting the same number of E epochs for local training per node. The curves in Figure 5 correspond to the use of FedAvg with different number of epochs (20 and 60 Epochs) for local training. The blue curve uses different epochs to train each node model and an early stopping callback configuration with a patience of 10. The number of epochs for green and red curves were set to 20 and 60, respectively.

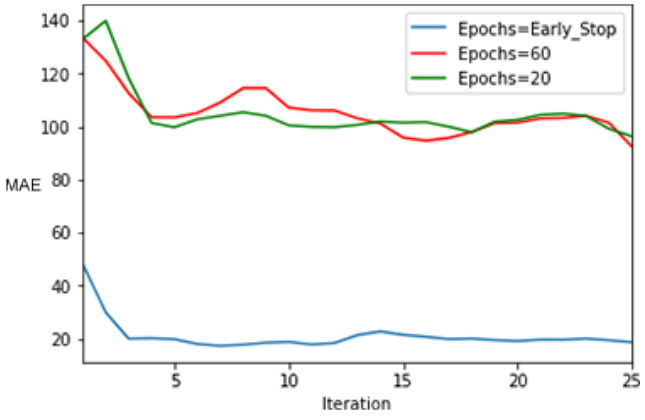

Figure 5. FedAvg using different E Epochs and $N = 4$

The number of epochs for green and red curves in Figure 5 were obtained after the training experience of the blue curve. In an initial training iteration $T = 0$ of blue curve, each node process data along approximately 60 epochs to train its model. In the other hand, in a training iteration $T > 3$, each node process data along approximately 20 epochs to train its model. We intuitively considered setting the number of epochs to $E = 20$ to minimize prediction errors of the global model throughout short prediction error minimization on each node.

5.3.1. Convergence of prediction error

In Figure 6, global models trained using different μ values on the FedProx algorithm were compared with global models trained using different number of epochs on the FedAvg algorithm. Here, The prediction error curves of FedAvg were previously illustrated in the Figure 5.

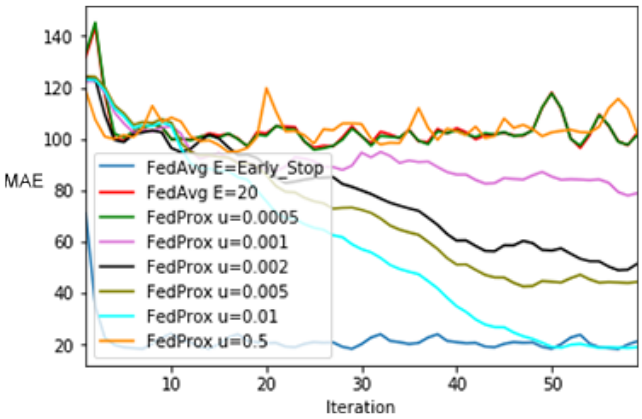

Figure 6. FedAvg vs FedProx

To obtain each prediction error curve illustrated in Figure 6, the global model of the second decentralized scenario was trained and evaluated in $T = 60$ iterations, process which took approximately 7 hours per curve. The rest of the training parameters and the computational resources were referred in the section 4.1, configurations, which were used to train the model of the centralized scenario.

The curve named "FedAvg early stop" shows the prediction errors which are possible to obtain, while the curve "FedAvg 20 Epochs" used the same E of FedProx curves. FedProx algorithm was evaluated using different μ values between zero and one. Nevertheless, Figure 6, contains more curves using $0 < \mu \leq 0.01$ because using $\mu > 0.01$ curves presented more oscillations, being the orange curve an example of them.

Over these experiments, we observed that using $\mu \sim 0.001$, prediction error curves takes similar values in comparison with the model trained on the centralized scenario. It can be observed after comparing the cyan curve with the blue curve. Here, FedProx curve not only achieved prediction errors of FedAvg curve but also presented less oscillations after 53 T iterations.

6. CONCLUSIONS AND FUTURE WORK

Experiments presented in this paper show that it is possible to achieve similar RUL prediction errors using Turbofan datasets in decentralized collaborative scenarios. FedAvg algorithm was able to achieve similar prediction errors in fewer aggregation iterations than FedProx but presenting uncertain

predictions along time. On the other hand, using FedProx, the prediction error curve progressively decreases, taking stable error predictions if μ is chosen adequately. However, founding an ideal μ value may take several tests and federated iterations.

In our experiments, FedProx required some federated iterations to obtain a stable prediction error. However, we consider that it is possible to take the same results using decentralized collaborative algorithms which consider the aggregation of deep learning models, e.g, Federated Matched Averaging (FedMA).

NOMENCLATURE OF PHM

EoL	End of Life
MAE	Mean Absolute Error
MSE	Mean Squared Error
RC	Initial constant value of RUL
$RMSE$	Root Mean Squared Error
RUL	Remaining Useful Life
$\hat{RUL}$	Remaining Useful Life predicted
SoF	Start of Failure

ACKNOWLEDGMENT

This Paper is part of a project that has received founding from the **European Union's Horizon 2020 research and innovation programme under grant agreement N°769288**

REFERENCES

Azevedo, D., Ribeiro, B., & Cardoso, A. (2019). Online simulation of methods to predict the remaining useful lifetime of aircraft components. In *2019 5th experiment international conference (exp.at'19).*

Canh, L., Kwok, T., Carl, S. B., Romano, P., & George, J. V. (2009). Fault diagnosis and failure prognosis for engineering systems: A global perspective. In *2009 ieee international conference on automation science and engineering.*

Heimes, F. O. (2001). A new approach to linear filtering and prediction problems. In *Journal of basic engineering* (p. 167-179).

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management.*

Hu, Y., Sun, X., Chen, Y., & Lu, Z. (2019). Model and feature aggregation based federated learning for multisensor time series trend following. *Advances in Computational Intelligence*(11506), 233-246.

Leto, P. (2008). Data driven prognostics using a kalman filter ensemble of neural network models. In *2008 international conference on prognostics and health management.*

Liu, Y., Frederick, D. K., De Castro, J. D., & Litt, J. S. (2012). *User's guide for the commercial modular aero-propulsion system simulation (c-mapss)* (Vol. 2; Tech. Rep. No. NASA/TM—2012-217432). NASA.

McMahan, B., & Ramage, D. (2017). *Federated learning: Collaborative machine learning without centralized training data* (Vol. http://ai.googleblog.com/2017/04/federated-learning-collaborative.html; Tech. Rep.). Google AI.

Olivares, A., Gonzalez, A., Tovar, S. T., & Gorrostieta, E. (2019). Remaining useful life prediction for turbofan based on a multilayer perceptron and kalman filter. In *2019 16th international conference on electrical engineering, computing science and automatic control - cce.*

Palau, A. S., Bakliwal, K., Dhada, M. H., Pearce, T., & Parlikad, A. K. (2018). Recurrent neural networks for real-time distributed collaborative prognostics. In *2018 ieee international conference on prognostics and health management (icphm)* (p. 1-8).

Ramasso, E., & Saxena, A. (2014). Benchmarking and analysis of prognostic methods for cmapss datasets. *Prognostics and Health Management Conference*, *2*(5), 1-15.

Sahu, A. K., Li, T., Sanjabi, M., Zaherr, M., Talwalkar, A., & Smith, V. (2019). On the convergence of federated optimization in heterogeneous networks. *CoRR*, *abs/1812.06127*. Retrieved from `http://arxiv.org/abs/1812.06127`

Saxena, A., J., C., Balaban, E., Goebe, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *2008 international conference on prognostics and health management.*

Saxena, A., & K., G. (2008). *Phm08 challenge data set* (Vol. http://ti.arc.nasa.gov/project/prognostic-data-repository; Tech. Rep.). NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA.

Simões, D., & Soares, A. L. (2008). Knowledge communities and interorganizational networks. In *Encyclopedia of networked and virtual organizations* (p. 6).

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y. (2019). Federated learning with matmatched averaging. *CoRR*, *abs/2002.06440*. Retrieved from `http://arxiv.org/abs/2002.06440`

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *CoRR*, *abs/1902.04885*. Retrieved from `http://arxiv.org/abs/1902.04885`

BIOGRAPHIES

Raúl Llasag Rosero received the B.Eng. degree in information science and technology at the University of Army Forces, Sangolqui, Equador, in 2017 and the MSc degree in the Polytechnic Institute of Leiria, Portugal, in 2019. He is currently a researcher in Prognostics and Health Management at the University of Coimbra, Portugal. His current research interests include computer vision, mixed reality, prognostic and health management and federated learning.

Catarina Silva has a PhD degree in Computer Engineering. She is Professor at the Department of Informatics Engineering at the University of Coimbra and Researcher at the Center of Informatics and Systems of the University of Coimbra (CISUC). Her main research interests lie in machine learning and applications. Author and co-author of 4 books, circa 20 journal articles and 50 conference papers. Scientific committee and paper reviewer of several conferences and journals. IEEE senior member of the Computational Intelligence Society. IEEE chair of the Portuguese Section.

Bernardete Ribeiro is Full Professor at the Department of Informatics Engineering at the University of Coimbra, where she teaches Programming, Pattern Recognition, Business Intelligence among other subjects. She is Director of the Center of Informatics and Systems of the University of Coimbra (CISUC). Her research interests are in the areas of Machine Learning, Pattern Recognition, and their applications to a broad range of fields. She has coordinated and participated in several national and international projects.