The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5–9 June 2022, Oslo, Norway

REMAINING USEFUL LIFE PREDICTION WITH A DEEP SELF-SUPERVISED LEARNING APPROACH

A. AKRIM[1,2], C. GOGU[1], R. VINGERHOEDS[2], AND M. SALAÜN[1,2]

[1] Institut Clément Ader (UMR CNRS 5312) INSA/UPS/ISAE/Mines Albi,
Université de Toulouse, 3 rue Caroline Aigle, 31400 Toulouse, France
anass.akrim@gmail.com, christian.gogu@gmail.com

[2] ISAE-SUPAERO, Université de Toulouse, 10 Avenue Edouard Belin,
31400 Toulouse, France
rob.vingerhoeds@isae-supaero.fr, michel.salaun@isae-supaero.fr

Key words: Prognostics and Health Management (PHM), Remaining Useful Life, Deep Learning, Self-Supervised Learning

Abstract. With the increasing availability of data for Prognostics and Health Management (PHM), Deep Learning (DL) techniques are now the subject of considerable attention in Prognostics for Predictive Maintenance, achieving more accurate Remaining Useful Life (RUL) predictions. However, one of the major challenges for DL techniques resides in the difficulty of obtaining large amounts of labeled data on industrial systems. To overcome this lack of labeled data, an emerging learning technique is considered in this work : Self-Supervised Learning, a sub-category of unsupervised learning approaches. This paper aims to investigate whether pre-training DL models in a self-supervised way on unlabeled sensors data can be useful for downstream tasks in PHM (*i.e.* RUL estimation) with only limited amount of labelled data. A synthetic dataset composed of strain data is used. Results show that the self-supervised pre-trained models significantly outperform the non pre-trained models in downstream Remaining Useful Life (RUL) prediction task, showing promising results in prognostic tasks when only limited labeled data is available.

1 INTRODUCTION

Prognostics and Health Management (PHM) is a reseach domain that is interested in studying the failure mechanisms of real systems in order to better manage the use of information on equipment operating conditions [1]. Its implementation can improve the efficiency of maintenance support [2], optimize the maintenance plan [3], help industry to balance safety and economic profit [4]. Fatigue damage is one of the major modes of failure for many mechanical structures and notably aerospace structures. Therefore, fatigue monitoring and prediction of fatigue life in structures, *i.e.* Remaining Useful Life (RUL) estimation, represents one of the major challenges to be solved for paving the way towards predictive structural maintenance.

Among the approaches used for PHM, Data-Driven models have gained more and more attention in the PHM community, especially the latest Deep Learning (DL) techniques [5], redefining state-of-the-art performances in a wide range of areas in recent years [6]. However, their effectiveness depends on the quantity and quality of available labeled data. In Prognostics tasks, a label can constitute the RUL at each time step of measurements, which is generally difficult to acquire and often can be a time-consuming and expensive investment for experts since faults are rare and structures can be replaced before reaching failure. As a result, data scarcity is becoming one of the most important challenges in PHM [7], [8] and exploiting unlabelled data during training (e.g. raw sensors data from structures replaced before reaching failure) has become a major goal in Machine Learning (ML) in order to improve learning performance. The research question in this paper can be stated as follows : *is it possible to learn meaningful representations from unlabelled data and use it to enhance related supervised predictive tasks on a fatigue damage prognostics problem ?*

In several publications, the challenge of data scarcity was addressed, through learning techniques to extract knowledge from unlabelled data. For example, an emerging learning technique was proposed : Self-Supervised Learning (SSL) [9], a sub-category of unsupervised learning approaches. It consists in learning meaningful and general representations from unlabeled data, without requiring human-annotated labels, which are applicable to a wide range of related supervised tasks (*i.e.* downstream tasks) with only few labeled data (*i.e.* "Few-Shots Learning"). This approach has already shown tremendous performances in many fields such as in Natural Language Processing (e.g. Generative Pre-trained Transformer 3 [10]) or Image Processing [11]. However, there is only a limited amount of existing research that focuses on the contribution of Self-Supervised Learning to Prognostics [12], and particularly for fatigue damage prognostics problems. Despite demonstrating encouraging results, the domain of SSL is still largely unexplored in the prognostics field and is in contrast with the increasing amount of unlabelled data available, having the potential to enable predictive maintenance.

In order to address this limitation, this paper aims to investigate whether pre-training DL models in a self-supervised way on unlabeled sensors data can be useful for downstream tasks in PHM (*i.e.* RUL estimation) with only Few-Shots Learning. In this research, the issue of data scarcity in a fatigue damage prognostics problem is addressed. The interest is in estimating the RUL of aluminum alloy panels (typical of aerospace structures) subject to fatigue cracks from strain gauge data. A synthetic dataset is used, composed of a large unlabeled dataset (*i.e.* strain gauges data of structures before failure) for pre-training, and a smaller labeled dataset (*i.e.* strain gauges data of structures until failure) for fine-tuning.

The remainder of the paper is organized as follows. Section 2 describes the synthetic data set used in this paper, as well as the implementation of the Self Supervised Learning approach on the current problem. The deep learning-based models used to investigate the SSL approach are also presented and detailed in this section. Section 3 presents the experimental settings used in this study for pre-training and fine-tuning phases, and the results obtained with the deep learning-based approaches trained in a self-supervised manner are analyzed. Finally, Section 4 concludes the paper and identifies some research perspectives.

2 METHODOLOGY

In this section, a description of the dataset is provided, followed by a description of the problem considered in this paper and the way the Self-Supevised Learning approach is implemented on it.

2.1 Data Description

In the current research study, a synthetic dataset for a realistic fatigue damage prognostics problem is generated, based on a framework previously proposed by the authors and described in [13]. It consists of synthetic multivariate run-to-failure time series data for structures subject to fatigue (*e.g.* fuselage panels), containing the variations of the strains at $n_g = 3$ positions in the panel as a function of the number of cycles, where n_g is the number of the time series. This setup can be seen representative of real experiments under fatigue loading where the strain state is monitored at three strain gauge positions. The strain data, or measurement sequences, are obtained until the critical crack size is reached, considered as the time of failure (necessary to compute the RUL at each time step). In this synthetic dataset, an Aluminum alloy 7075-T6 plate was considered, which is typical of aeronautic structures. More details about the dataset are given in [13], and an illustration of a generated sequence (*i.e.* three placed gauges) for a single structure until failure is given in Fig. 1.

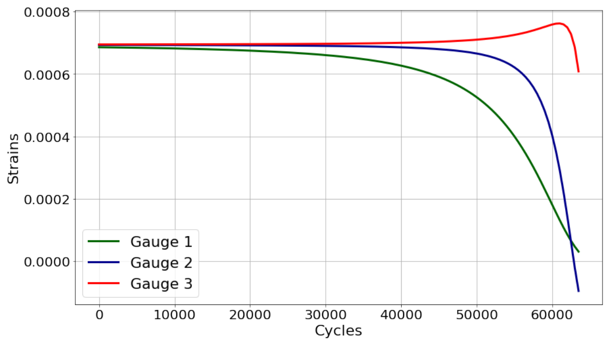
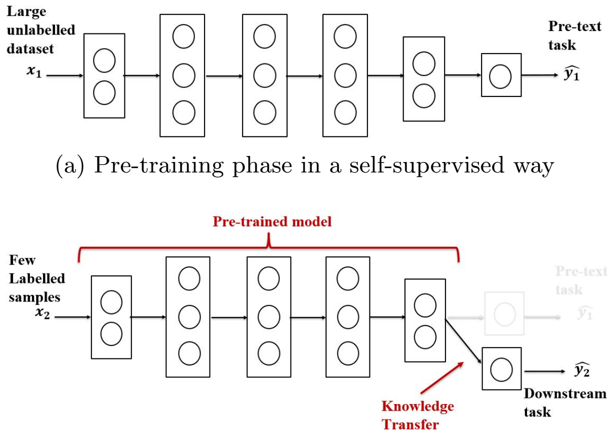


Figure 1: Strain values time series corresponding to a sensor sample generated.

In this paper, a training set and a testing set are generated. The testing set is used to evaluate the RUL estimation performance of the trained models (in fine-tuning) as a held-out data set that has not been used prior. The datasets used will be detailed in Section 3.

2.2 Self-Supervised Learning

Similarly to *self-taught learning* [14], Self-Supervised Learning (SSL) consists in learning meaningful and general representations from unlabeled data (*i.e.* pre-training phase) by solving a so-called *pretext task* without requiring human-annotated labels. These representations are applicable to a wide range of related supervised tasks (*i.e.* *downstream task*) with only few labeled data (*i.e.* "Few-Shots Learning"). The working of SSL can be illustrated in Figure 2.



(a) Pre-training phase in a self-supervised way

(b) Fine-tuning phase (supervised training on down-stream tasks)

Figure 2: A schematic view of the Self-Supervised Learning procedure.

The proposed self supervised learning approach is composed of a :

1. Pre-training phase : pre-train a Deep Learning model on unlabelled data (*i.e.* strain data of structures before failure) in a self supervised manner, called pre-text task. Also, in order to vary the pre-text tasks and inspired by [9], [15], two types of models are used : 1) Autoencoders (AE) and 2) Autoregressive models (AR).

2. Fine tuning phase : fine-tune the pre-trained model on a specific downstream Prognostics task (*i.e.* RUL estimation).

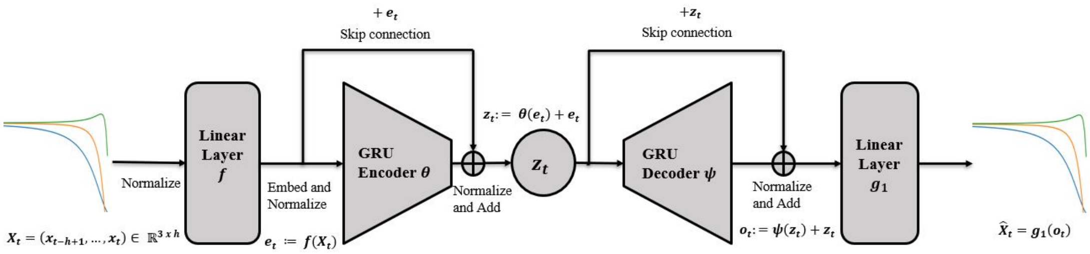
A schematic view of the investigated models used in the proposed SSL approach is given in Fig. 3 and Fig. 4. As in [13], a sliding window approach is used in training : at each time-step t, the input of the predictive models corresponds to the current and past measurements, such that $X_t := (x_{t-h+1}, \ldots, x_t) \in \mathbb{R}^{n_g \text{ x } h}$ where n_g is the number of the time series and $h = 30$ is the length of the sliding window (the value of parameter h was set after preliminary experiments).

In pre-training, the output of the Autoencoder (AE) is an estimation of the input signal such that $y_t = \hat{X}_t = (\hat{x}_{t-n_w+1}, ..., \hat{x}_t)$, while the output of the Autoregressive model (AR) is an estimation of the data of the next timestep such that $y_t = \hat{x}_{t+1}$. In both models, Gated Reccurent Unit (GRU [16]) networks are used as the basic deep prediction model, because of their sequential properties and good regressive performance in previous work [13]).

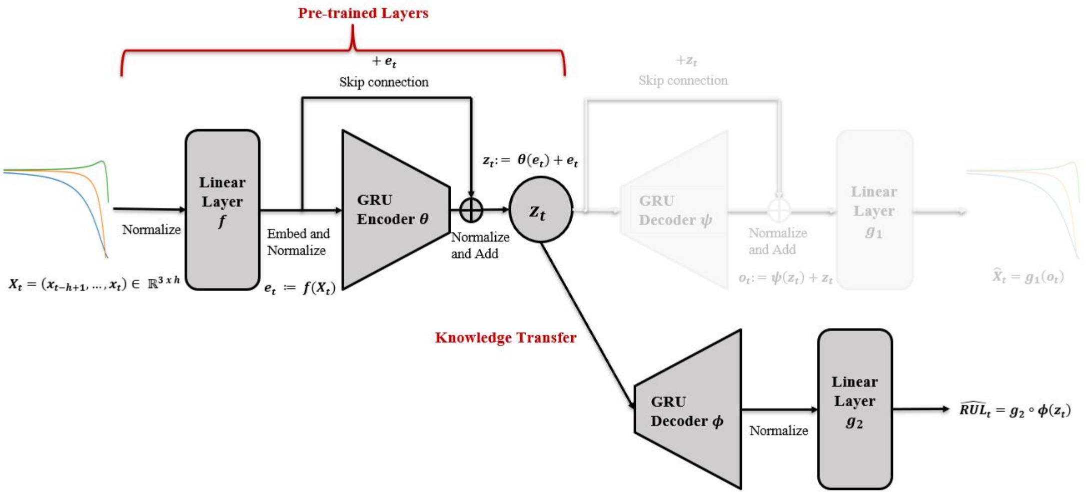
In the fine-tuning phase, an RUL estimation problem is considered, hence the output of the predictive models is a point-wise estimation of the RUL such that $y_t = R\hat{U}L_t$. The embedding z_t of the input data is extracted (see Fig. 3 and Fig. 4), the weights of the hidden pre-trained layers are frozen, then a simple GRU Decoder for fine-tuning is used, (composed of a stack of GRU layers), called *Knowledge Transfer*. Note that, in fine-tuning phase it is common to use only a linear layer for training, but the authors of the current paper found that adding a GRU model significantly improves the performance of the approach on this RUL estimation problem.

Finally, in order to investigate the added value of the SSL approach in prognostics, the pre-trained models are compared with their non pre-trained counterpart architecture, illustrated in Fig. 3-b) and Fig. 4-b).

Note that the pre-trained model with an autoregressive pre-text task is a decoder model and will be referred to as the "autoregressive model" in the following for simplicity.

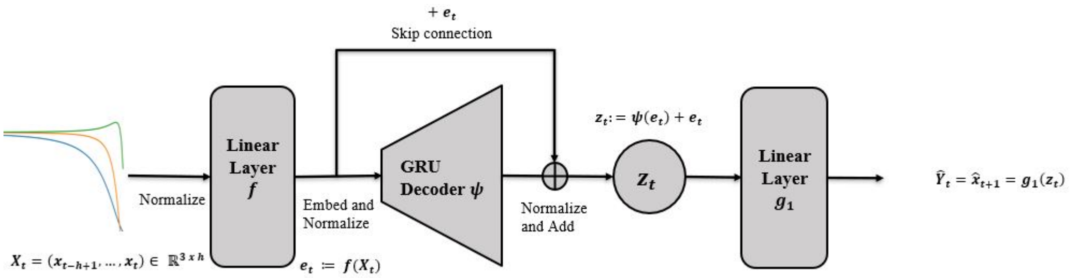


(a) Pre-training phase in a self-supervised way. The output of the model is an estimation of the input signal such that $y_t = \hat{X}_t = (\hat{x}_{t-n_w+1}, ..., \hat{x}_t)$.

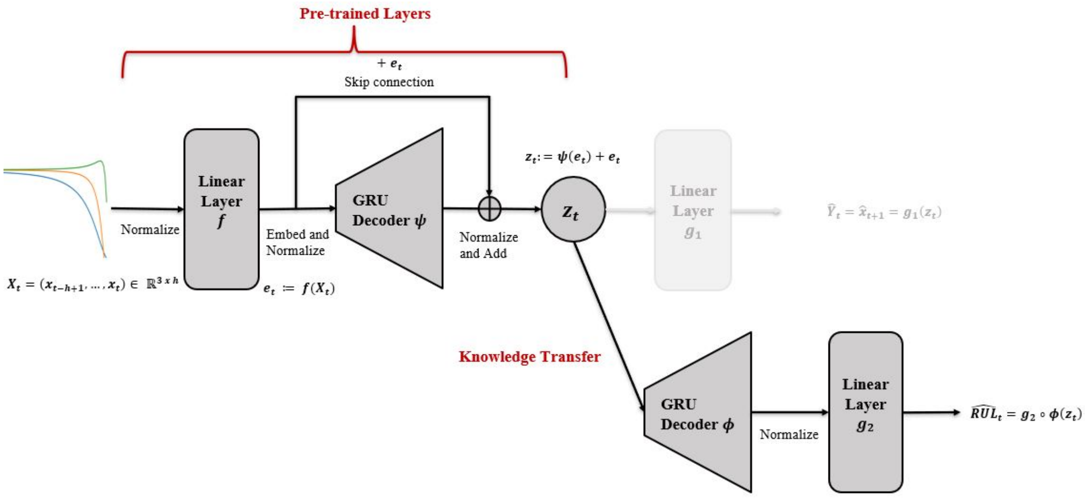


(b) Fine-tuning phase : a RUL estimation problem is considered, hence the output of the predictive model is a point-wise estimation of the RUL such that $y_t = R\hat{U}L_t$.

Figure 3: A schematic view of the Autoencoder model (AE) used in the Self-Supervised Learning procedure.



(a) Pre-training phase in a self-supervised way. The output of the model is an estimation of the data of the next timestep such that $y_t = \hat{x}_{t+1}$.



(b) Fine-tuning phase : a RUL estimation problem is considered, hence the output of the predictive model is a point-wise estimation of the RUL such that $y_t = R\hat{U}L_t$.

Figure 4: A schematic view of the Autoregressive model (AR) used in the Self-Supervised Learning procedure.

3 EXPERIMENTS AND RESULTS

3.1 Experimental settings in pre-training phase

In pre-training, the number of structures, denoted N_p, for which strain sequences were available was varied to investigate the effect of the amount of unlabelled data. The investigated models (autoencoder and autoregressive model) were therefore pre-trained on $N_p = 100, 1000, 5000$, and 10000 unlabelled structures.

As the structures subjected to fatigue can be replaced before reaching failure at any time, the proposed approach has been investigated on four degradation scenarios : for pre-training, available sequences of unlabelled data are incomplete at $d = 60\%$, 70%, 80%, and 90% of their total lifetime, where d is the ratio of the total lifetime of a sequence. To have an idea of the size of the strain data sequences available, these four degradation scenarios are illustrated in Fig. 5.

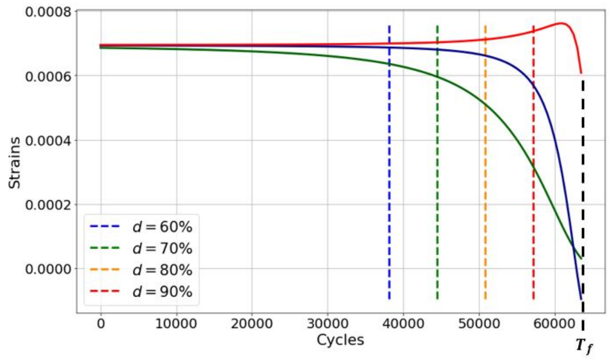


Figure 5: Four degradation scenarios depicted on the sequence of a structure. For each scenario, the available strain data correspond to the measurements from time 0 to time $t^* = d$ x T_f, where d is the ratio of the total lifetime of a sequence, and T_f is the time of failure.

In each training procedure, 95% of the dataset for training was used, while 5% of it is used for validation (*i.e.* monitoring and adjusting the training phase). The Adam optimizer [17] is used with default parameters and the learning rate is decreased incrementally. The learning rates of $10^{-2}, 10^{-3}, 10^{-4}$ were sequentially used for a predefined number of epochs, saving the model weights each time the validation loss decreases; the weights of the best model are loaded each time the learning rate is lowered. At the end of the procedure, the model is trained on the whole dataset (training and validation sets) with a lower learning rate of 10^{-5} until convergence. Note that the model hyperparameters were first optimized using a Grid Search algorithm :

- Autoencoder model illustrated in Fig. 3 : the GRU Encoder and GRU Decoder are each composed of 2 layers of GRU, $n = 64$ neurons, and a dropout of 0.1.

- Autoregressive model illustrated in Fig. 4 : the GRU Decoder is composed of 4 GRU layers, n = 64 neurons, and a dropout of 0.1.

3.2 Experimental settings in Fine-tuning phase

For fine-tuning, as illustrated in Fig. 3 and Fig. 4, the embedding z_t of the input data is extracted, the weights of the hidden layers are frozen, and a GRU Decoder is used for downstream task (composed of a single GRU layer, 32 neurons, 0.1 in dropout to regularize). The pre-trained models are then compared with their non pre-trained 'counterpart' (*i.e.* same architecture but the model weights are all reset) on few shots learning. The number of labelled structures N_{FT} are varied, such that : $N_{FT} = 5, 10, 20, 50$ and 100 labelled structures (*i.e.* strain data of structures reaching failure, from time 0 to time time of failure T_f).

The models were evaluated on a testing set of 100 structures, and since the RUL estimation problem is considered as a regression problem in this paper, the metric used to evaluate these models is the Mean Absolute Percentage Error (MAPE), such that :

$$MAPE = \frac{1}{S}\sum_{i=1}^{S}|\frac{RUL_i - R\hat{U}L_i}{RUL_i}| * 100 \tag{1}$$

where S is the number of samples with $R\hat{U}L_.$ being the prediction and $RUL_.$ the target value.

As a limited amount of labeled data leads to epistemic uncertainty, it is difficult to make a reliable comparison. Hence, a 5-fold cross validation is used by varying the split between the training and validation set, which gives an average MAPE error and its standard deviation to quantify the uncertainty when evaluated on the test set.

3.3 Results

The pre-trained models as well as their non pre-trained counterpart architecture are trained on labelled samples, then evaluated on 100 structures of the testing data set. First, the performance of pre-trained Autoencoder (AE) and non pre-trained counterpart in RUL estimation are compared when evaluated on the testing set, by varying the number of unlabeled samples in pre-training, illustrated in Table 1.

		$MAPE$ (%)	Mean ± St.	Dev.	
Labelled structures	5	10	20	50	100
Pre-trained model (d = 60%)					
Autoencoder N_p = 100	29.36 ± 3.81	21.16 ± 2.86	11.84 ± 4.12	1.95± 0.25	1.87± 0.36
Autoencoder N_p = 1.000	25.95 ± 0.74	14.78 ± 3.13	5.59 ± 1.47	1.42± 0.17	1.31± 0.06
Autoencoder N_p = 5.000	28.55 ± 0.99	**12.01 ± 2.31**	4.80 ± 0.92	**1.27 ± 0.11**	**1.08 ± 0.03**
Autoencoder N_p = 10.000	**24.60 ± 4.30**	12.70 ± 1.76	**3.48 ± 0.93**	1.41 ± 0.04	1.13 ± 0.12
Pre-trained model (d = 70%)					
Autoencoder N_p = 100	28.02 ± 1.96	20.99 ± 3.73	11.70 ± 3.78	1.61± 0.43	1.43± 0.25
Autoencoder N_p = 1.000	30.76 ± 1.83	18.20± 3.76	6.42± 1.69	1.65± 0.36	1.29± 0.16
Autoencoder N_p = 5.000	26.10 ± 0.88	18.73± 2.72	6.67± 1.30	1.58± 0.23	1.25± 0.08
Autoencoder N_p = 10.000	**25.38 ± 5.99**	**11.85 ± 2.64**	**3.77± 0.61**	**1.29± 0.06**	**1.08± 0.12**
Pre-trained model (d = 80%)					
Autoencoder N_p = 100	34.96 ± 9.59	17.72 ± 4.43	8.92 ± 3.33	1.33± 0.13	1.41± 0.20
Autoencoder N_p = 1.000	26.46 ± 2.51	15.78 ± 3.77	4.74 ± 0.79	1.31 ± 0.13	1.10± 0.05
Autoencoder N_p = 5.000	30.19 ± 6.56	17.18 ± 3.03	6.19 ± 2.07	1.60 ± 0.22	1.17± 0.18
Autoencoder N_p = 10.000	**24.77 ± 4.80**	**12.06 ± 3.82**	**4.27 ± 0.72**	**1.18 ± 0.13**	**1.01 ± 0.06**
Pre-trained model (d = 90%)					
Autoencoder N_p = 100	28.27 ± 2.47	21.56 ± 0.98	9.91 ± 3.29	1.57 ± 0.35	1.56 ± 0.16
Autoencoder N_p = 1.000	30.27 ± 1.15	18.53 ± 5.68	6.51 ± 1.63	1.74 ± 0.43	1.13 ± 0.15
Autoencoder N_p = 5.000	25.99 ± 1.73	17.06 ± 4.09	5.24 ± 1.60	1.43 ± 0.31	1.06 ± 0.07
Autoencoder N_p = 10.000	**22.97 ± 5.69**	**11.04 ± 3.61**	**3.39± 0.67**	**1.22± 0.12**	**0.88 ± 0.09**
Non pre-trained model					
Autoencoder architecture	27.72 ± 0.65	23.07 ± 5.94	8.12 ± 1.87	1.40 ± 0.33	**0.83 ± 0.18**

Table 1: $MAPE$ mean values (in %) plus or minus its standard deviation as a function of the number of labelled structures used and of the training scenario for the autoencoder case. Best performance is represented in bold for each case.

As can be seen in Table 1, pre-training the model is not always beneficial. For example, for a pre trained AE model on 100 structures, the efficiency on RUL estimation is not always better over the non pre-trained model (AE). The term *negative transfer* can be used when the transfer method decreases predictive performance [18]. However, it can also be observed that as the number of labelled samples increases, the pre-training becomes more efficient and allows to have better results than a non pre-trained model when few labelled structures are available, especially for the model pre-trained on 10000 structures. Overall, results in Table 1 show that for the Autoencoder model, the number of unlabeled samples in pre-training matters : the more the number of these samples increases, the more efficient the self supervised learning is for each of the 4 scenarios. The autoregressive model shows similar performances, illustrated in Table 2.

		$MAPE$ (%)	Mean ± St.	Dev.	
Labelled structures	5	10	20	50	100
Pre-trained model (d = 60%)					
Autoregressive N_p = 100	36.15 ± 13.48	20.18 ± 5.19	12.17 ± 3.31	2.29± 0.22	1.70± 0.19
Autoregressive N_p = 1.000	28.14 ± 3.20	16.65 ± 1.21	8.80 ± 3.13	1.48± 0.18	1.21± 0.13
Autoregressive N_p = 5.000	26.53 ± 1.57	13.58 ± 2.14	7.25 ± 3.09	1.22 ± 0.02	1.00± 0.01
Autoregressive N_p = 10.000	**22.48 ± 6.06**	**7.34 ± 0.95**	**2.63 ± 0.80**	**1.14± 0.04**	**1.03 ± 0.06**
Pre-trained model (d = 70%)					
Autoregressive N_p = 100	28.95 ± 1.74	16.98 ± 2.96	11.85 ± 2.73	2.01 ± 0.20	1.51 ± 0.34
Autoregressive N_p = 1.000	26.76 ± 2.49	16.34 ± 1.38	8.54 ± 1.89	1.53 ± 0.20	1.18 ± 0.16
Autoregressive N_p = 5.000	25.18 ± 2.70	10.96 ± 2.46	6.79 ± 2.51	1.33 ± 0.13	1.27 ± 0.21
Autoregressive N_p = 10.000	**24.43 ± 4.08**	**8.46 ± 1.53**	**2.42 ± 0.53**	**1.20 ± 0.06**	**1.14 ± 0.17**
Pre-trained model (d = 80%)					
Autoregressive N_p = 100	30.47 ± 3.47	20.04 ± 2.59	10.68 ± 4.25	2.34 ± 0.85	1.48 ± 0.07
Autoregressive N_p = 1.000	**26.10 ± 2.14**	13.66 ± 3.22	6.01 ± 1.45	1.44 ± 0.05	1.17 ± 0.03
Autoregressive N_p = 5.000	26.46 ± 2.46	12.60 ± 1.33	6.91 ± 1.28	1.38 ± 0.11	1.09 ± 0.05
Autoregressive N_p = 10.000	26.50 ± 2.58	**8.09 ± 3.28**	**2.39 ± 0.26**	**1.39 ± 0.20**	**1.07 ± 0.11**
Pre-trained model (d = 90%)					
Autoregressive N_p = 100	35.76 ± 7.41	17.64 ± 2.58	8.66 ± 2.26	1.60 ± 0.17	1.33 ± 0.19
Autoregressive N_p = 1.000	25.00 ± 3.89	17.31 ± 1.72	8.59 ± 2.79	1.45 ± 0.15	1.29 ± 0.20
Autoregressive N_p = 5.000	23.01 ± 3.23	12.97 ± 2.91	3.15 ± 0.45	1.21 ± 0.08	1.05 ± 0.03
Autoregressive N_p = 10.000	**22.69 ± 2.36**	**8.83 ± 1.61**	**3.39 ± 0.40**	**1.25 ± 0.07**	**0.99 ± 0.06**
Non pre-trained model					
Autoregressive architecture	28.09 ± 1.63	21.10 ± 1.92	7.52 ± 1.59	1.15 ± 0.09	**0.79 ± 0.09**

Table 2: $MAPE$ mean values (in %) plus or minus its standard deviation as a function of the number of labelled structures used and of the training scenario for the autoregressive case. Best performance is represented in bold for each case.

In the current paper, the Autoencoder model and the Autoregressive model are compared to SSL, in order to investigate the influence of pre-text task on pre-training. Results obtained do not allow to clearly distinguish between the two models when pre-trained on 5 structures, due to the limited number of labelled samples. Nevertheless, results show that both pre-trained models clearly outperform their non pre-trained counterpart in Few-Shots learning (more than 5 but less than 50 structures). The AR pre-trained model significantly outperforms the AE pre-trained when fine-tuned on 10 or 20 structures, and has almost three times less estimation error than the best non pre-trained model. These results make sense since the autoregressive task and the RUL estimation task have in common the task of predicting future outcome, and may need to capture the temporal dependencies of the input signal. However, it can also be seen that as the number of labeled samples increases, the difference between the pre-trained and non pre-trained models is reduced (e.g. trained on more than 50 structures).

4 CONCLUSION

In this paper, a Self-Supervised Learning approach for fatigue damage prognostics problem was presented and investigated. Four degradation scenarios were investigated and overall, results showed that self supervised learning is efficient in prognostics and can improve RUL estimation performances when a limited amount of labelled data is available. Also, results confirmed that the number of pre-training samples matters, as well as the choice of the pre training model or pre-text task.

In next steps, it would be interesting to explore other pre-text tasks (e.g. change prediction time horizon, etc.) or other models (e.g. variational autoencoders). Also another direction of research would be to investigate the robustness of pre-trained models to domain shift, *i.e.* evaluate the performance of the fine-tuned models on a related data set following a slightly different distribution.

References

[1] X. Shao-feng, E. Yun-fei, L. Xiao-ling, L. Yu-dong, and C. Yi-qiang, "Development and application of prognostics and health management technology," in *Proceedings of the 20th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, IEEE, 2013, pp. 3–7.

[2] D. Mao, C. Lv, J. Shi, Y. Zou, and Z. Guo, "Research of the military aircraft maintenance support mode based on the prognostics and health management," in *2010 Prognostics and System Health Management Conference*, IEEE, 2010, pp. 1–6.

[3] V. Atamuradov, K. Medjaher, P. Dersin, B. Lamoureux, and N. Zerhouni, "Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation," *International Journal of Prognostics and Health Management*, vol. 8, no. 060, pp. 1–31, 2017.

[4] Z. Wen and Y. Liu, "Applications of prognostics and health management in aviation industry," in *2011 Prognostics and System Health Managment Confernece*, IEEE, 2011, pp. 1–5.

[5] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, "Prognostics and health management: A review on data driven approaches," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, ISSN: 1476-4687. DOI: 10.1038/nature14539.

[7] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103 678, 2020.

[8] A. Theissler, J. Pérez-Velázquez, M. Kettelgerdes, and G. Elger, "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry," *Reliability engineering & system safety*, vol. 215, p. 107 864, 2021.

[9] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A Survey on Contrastive Self-Supervised Learning," en, *Technologies*, vol. 9, no. 1, p. 2, Mar. 2021, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/technologies9010002.

[10] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[11] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in neural information processing systems*, vol. 33, pp. 22 243–22 255, 2020.

[12] T. Krokotsch, M. Knaak, and C. G¨uhmann, "Improving Semi-Supervised Learning for Remaining Useful Lifetime Estimation Through Self-Supervision," en, *International Journal of Prognostics and Health Management*, vol. 13, no. 1, Jan. 2022, ISSN: 2153-2648, 2153-2648. DOI: 10.36001/ijphm.2022.v13i1.3096.

[13] A. Akrim, C. Gogu, T. Guillebot de Nerville, P. Strähle, B. Waffa Pagou, M. Salaün, and R. Vingerhoeds, "A framework for generating large data sets for fatigue damage prognostic problems," in *2022 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2022, pp. 25–33. DOI: 10.1109/ICPHM53196.2022.9815692.

[14] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759–766.

[15] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *International Conference on Machine Learning*, PMLR, 2020, pp. 1691–1703.

[16] R. Rana, "Gated recurrent unit (gru) for emotion classification from noisy speech," *arXiv preprint arXiv:1612.07778*, 2016.

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

[18] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI Global, 2010, pp. 242–264.