Journal of Artificial Intelligence and Data Science (JAIDA)
e-ISSN: 2791-8335
https://jaida.ikcu.edu.tr

Vol. 1, No. 2, pp. 165-175, 2021
Research Article

Siamese Inception Time Network for Remaining Useful Life Estimation

Ugur CEYLAN *, Yakup GENC

[1] Gebze Technical University, Computer Engineering Department, Turkey

Abstract
Predictive maintenance tries to reduce cost in engine maintenance in power plants, aircraft, and factories by predicting when maintenance is needed (or by estimating the remaining useful life). Recently, with significant advances in deep learning and the availability of high volumes of data extracted from manufacturing processes, data-driven methods for predicting remaining useful life (RUL) have received a lot of attention. A major problem with data-based prognosis is that it is costly and requires large amounts of training data that could be difficult to obtain in large numbers of failure cases (often impossible and expensive to obtain in the real world). This paper proposes a learning-based method for fault diagnosis requiring fewer failure data. To this end, we use a Siamese network architecture based on a specific deep Convolutional Neural Network (CNN) called InceptionTime. The Siamese part of the network allows repeated use of the existing data to establish a similarity metric for two separate time windows. The turbofan engines C-MAPSS dataset supplied by NASA is used to verify the proposed model. Experiments are conducted using various sizes of the turbofan engines data to compare the performance on small datasets between the proposed model and a base-line model. The results demonstrate that our model can be used in fault diagnosis and provide satisfying prediction results with fewer data with comparable performances against state-of-art methods for RUL prediction.
Keywords: C-MAPSS, deep learning, prognostics and health management, remaining useful life, siamese networks.

1. Introduction

Prognostics and health management (PHM) of mechanical material has attracted attention and predicting remaining useful life (RUL) which is the total amount of cycles/time a given machinery/component can still work before failure is in the core of PHM [1, 2]. By estimating the RUL of the machinery, it is likely to provide preventive mechanical actions and offer a maintenance plan in a targeted way [3]. There are two main methods to estimate mechanical equipment RUL: data-driven and model-based approaches. Basically, in the former, machinery is run until a fault occurs and the runs are repeated multiple times to collect data. The latter requires knowledge of the physics of the breakdown progression [4]. In recent years, the remaining useful life (RUL) studies for data-driven approaches have drawn considerable attention when physical modeling is difficult to obtain for complex systems. In this paper, RUL estimation is predicted using a data-driven method. Data-driven approaches offer a solution for fault detection after certain faults occur (diagnostics) and forecasts of the future operating situations and the time to failure (prognosis). The main challenge in data-driven approaches is that it is usually difficult to obtain a large number of samples for failure progression which could expensive and labor intensive [4]. This can appear in a few different scenarios: (1) industrial systems are not permitted to work until failure due to the outcomes, particularly for crucial systems and failures; (2) most of the mechanical breakdowns happen gradually and result in a degradation path such that failure degradation of a system might take months or even years. Various techniques have been used to address this challenge. The first is to quicken aging by operating the system in a lab with excessive loads, increased speed, or doing simulations of real components that are made by weak materials so that a failure proceeds faster than normal [5]. Another technique is organizing failure progress unnaturally using exponential degeneration to model natural failure progression [6]. Both techniques have their benefits and limitations with the ability to interpret the failure degradation. But, in real-world applications where the circumstance is very distinct from the lab conditions, these techniques are challenging to perform and to obtain sufficient predicting achievement.

Recently, it has been shown that machine learning approaches, especially deep learning methods, perform powerful outcomes in areas such as computer vision, image processing, and natural language processing [7]. Deep learning methods have been employed for fault diagnosis and applied to the RUL estimation problem, in particular using ensemble networks (different types of networks combination). Using a combination of a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) neural network proposes an encouraging method to estimate RUL with a convenient and precise method that can be extended and adapted to obtain more accurately. However, one of the biggest limitations is that they require a large number of labeled failure cases data. In many applications, it is sometimes not possible to collect a lot of data. We also need to retrain the model whenever we want to classify class or estimate the RUL of a new failure case data. Therefore, the proposed method

*Corresponding author e-mail address: uceylan@gtu.edu.tr

Received: 02/12/2021; Accepted: 27/12/2021

aims to solve those problems by using a Siamese network. The Siamese networks tries to model the discriminant nature of the data from two opposite samples. This can be leveraged to generalize the network's predictive ability not only to new data but to entirely new types from unknown distributions [8]. By allowing repeated use of the existing data to establish a similarity metric for two separate time windows, the network is able to generalize predictive ability on small datasets and address the problem of lacking training samples.

This paper proposes a Siamese network that is based on a deep Convolutional Neural Network (CNN), called Inception-Time to increase the precision of RUL prediction performance. The problem is posed as modeling a metric between two time-windows in the observed data, similar being on the same side of the failure start point, and different being on the opposite sides.

The rest of the paper is organized as Section 2 provides a brief literature review for RUL estimation methods, with focus on machine learning (ML) methods and deep learning (DL) approaches. Section 3 describes the preprocessing of input data to the Siamese InceptionTime network. Also, in this section details about the proposed Siamese InceptionTime Network are given. Experiments and results are explained in Section 4 with Section 5 concluding the paper.

2. Background

The state-of-art for predictive maintenance shows that the newest studies on the topic are data-driven approaches and model-driven studies are out of date. Data-driven studies can be classified into two main groups. The first group tries to predict the remaining useful life. Historical sensor data from the machinery/engine is employed as a time series and the problem is converted into a regression problem [9, 10, and 11]. In the second group study, the status of the machinery/engine is characterized as healthy, slightly damaged, or very damaged, and transformed into a classification problem [12, 13]. Many distinct methodologies have been stated in the literature to estimate RUL in this manner.

Early studies on predictive maintenance for RUL estimation are based on statistical methods and ML-based approaches. For instance, Ordóñez *et al.* [9] proposed a data-based approach on turbofan engine dataset using auto-regressive integrated moving average (ARIMA) forecasting prediction as input of the support vector machine (SVM) model. The best kernel function selection is performed using genetic algorithms strategies. The predictive capability of their proposed hybrid method is shown to be good compared to the ones obtained solving the problem once using a multivariate Vector Autoregressive Moving-Average (VARMA) models. Susto *et al.* [10] also proposed SVM for predictive maintenance. The model is based on the decision boundary produced by the SVM model. The data used is synthetic and has been generated using Monte Carlo simulation. The study does not propose a comparison between SVMs and other ML techniques. Khelif *et al.* [11] used SVM to model the direct relationship between sensor values or health index (HI) and predicted RUL directly from sensor values without estimating deterioration state or failure threshold.

Durbhaka *et al.* [14] use k-means to analyze the behavior of wind turbines using vibration signal analysis. In this study, k-NN and SVM algorithms are compared with the k-means algorithm to classify failure types in wind turbines. Recently as an ML-based approach, S. Behera *et al.* [12] proposed a predictive maintenance model that applies a data-driven method for identification of the condition engine and solution formulated as classification manner with one of the three possible values critical, warning, and normal depending upon the RUL value. Utilizing advanced feature engineering, they obtained more insight and used ensemble tree learning. Comprehensive experiments are conducted on a widely used C-MAPSS dataset. It has been seen that Gradient Boosted Trees (GBT) accuracy performs better than Random Forest (RF) accuracy. However, RF competitively performed with a much faster compute time in comparison to the GBT.

Deep neural network (DNN) based methods have gained dominance in the last decade. For instance, Xu *et al.* [15] the healthy status of the engine with the data obtained from 21 sensors belonging to 100 different engines was tried to be monitored. The degradation behaviors of the 21 sensory signals observed and seven of them were selected in this study. The remaining data were estimated by a mixed system consisting of Dempster-Shafer Regression (DSR), SVM, and Recurrent neural networks (RNN). Babu *et al.* [25] built the first attempt of a deep CNN for RUL estimation across two public data sets. The CNN structure is employed to extract the local data features through the deep learning network for better prognostics. Their approach outperformed SVM and MLP based methods. Zhang *et al.* [26] proposed a neural network-based approach that a multi-objective deep belief networks ensemble (MODBNE) method. Deep belief networks (DBNs) are used to handle two issues. The first issue is utilizing handcrafted features which can hardly adjust to different predictive applications. It is employed DBNs as a solution since DBNs offer a promising solution to extracting the most useful features relevant to the estimation task via learning powerful hierarchical attribute representations from data. The other issue is utilizing one single model which can hardly keep good generalization performance across diverse predictive methods. They handle this problem with a multi-objective evolutionary ensemble learning framework incorporated with the traditional DBN.

Li *et al.* [16] proposed a deep learning CNN architecture to estimate RUL. For better feature extraction with CNN, the sliding the time window approach is used for sample preparation. The turbofan engine raw sensor data measurements with scaled between [-1, 1] and then used as input into the proposed network deep CNN architecture. This helps the industrial applications that do not need any prior expertise in signal processing. The high-level abstract features can be extracted with deep CNN architecture and related RUL can be estimated from the basis of the representation learned. The proposed method using time window, data normalization, and deep CNN structure is expected to achieve higher prognostic accuracy compared to traditional machine learning methods. Hanga *et al.* [17] proposed an applied Long Short-Term Memory (LSTM) network, an architecture that specializes in exploring fundamental patterns embedded in time series to monitor system degeneration and eventually predict RUL. The purposes of this paper firstly explain raw sensor data into an interpretable health index to better characterize the health status, and secondly monitor system corruption of the past to predict the future health situation correctly. For those purposes they used the C-MAPSS engine dataset and results show that LSTM produces more robust and accurate in explaining degeneration patterns, enabled by discovering the variation pattern underlying time-series is examined to track the system degradation.

Hinchi *et al.* [18] proposed a deep neural network framework that is based on convolutional and LSTM recurrent units for RUL estimation. First, the neural network subtracts the local features directly from sensor data using the convolutional layer, followed by an LSTM layer to catch the degradation process is added, RUL estimated finally using LSTM output and estimated time value. Experiments used ball-bearing data with good performances. Once again, LSTM models requires representative large training data sets. Zhang *et al.* [19] proposed for RUL estimation, a transfer learning algorithm based on Bi-directional Long Short-Term Memory (BLSTM) recurrent neural networks that models can be trained on different but related datasets first and then fine-tuned by the target dataset. Experimental results show that transfer learning can improve prediction models in the dataset with a small number of samples with one exception when transferring learning leads to a worse result when changing from multi-type to single operating conditions. Transfer learning methods however rely on a large training data. Li *et al.* [20] proposed a directed acyclic graph (DAG) network combining LSTM and a CNN for RUL estimation. The proposed method rather than just using CNN for extracting features it combines CNN and LSTM. The resulting model shows improved performance. We will use this method as our baseline model.

Finally, Siamese networks have been used in the literature quite a lot. Most of these are in the image processing domain. Koch *et al.* [8] use a Siamese network and employed it for identifying characters that consist of comprehensive alphabet sets. By using the Siamese convolutional architecture, they can achieve powerful results on one-shot classification tasks that exceed those of other deep learning models with state-of-the-art performance. Generally, with the Siamese neural networks, they learn image representations through a supervised metrics-based strategy and then reuse the properties of that network without any retraining. Hsiao *et al.* [21] applied Siamese neural networks to the malware image classification task. They implemented a method for identifying distinct types of malware for malware propagation using malware image classification. They also address the issue of lack of training samples at the early stage of new malware appearance by implementing Siamese networks.

3. Methodology

This paper presents a Siamese network based on InceptionTime architecture for RUL estimation using time series data. It is used in the turbofan engine degradation dataset. We first provide the details of the turbofan engine degradation dataset. Then we present the proposed Siamese InceptionTime network, as well as its key components, i.e., InceptionTime and Siamese networks.

3.1. Data overview

The turbofan engine degradation dataset is a well-known data set based on a simulation by Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tool developed by NASA [22]. The dataset consists of four sub-datasets based on different operating conditions and fault modes. Each sub-dataset includes training data, test data, and actual RUL that corresponds to the last cycle of the test data. In this study, sub-dataset FD001 was used. A short description of the FD001 sub-datasets is given in Table 1. The training data comprises a certain number of engines data from a specific healthy state to failure, while the test data includes a certain number of engine data that end a bit before failure. Furthermore, the training and test data with different initial healthy states, respectively. Because of different initial healthy states, the operating cycles of distinct engines are different in the same database. For example, in sub-dataset FD001 of the database, the training dataset includes 100 engines, with a minimum operating cycle of 128 and a maximum operating cycle of 362 and the test dataset includes 100 engines, with a minimum operating cycle of 31 and a maximum operating cycle of 303, and collected under one operating condition with one fault mode.

Table 1. *Description FD001 sub-dataset of the C-MAPSS.*

Sub-dataset	FD001
Training set engines #	100
Test set engines #	100
Training max/min cycles	362/128
Test max/min cycles	303/31
Operating condition	1
Fault modes	1

3.2. Siamese InceptionTime network

InceptionTime is inspired by the Inception-v4 architecture and an ensemble of deep Convolutional Neural Network (CNN) models [23] for time series classification (TSC), each created by cascading multiple Inception modules, where each module has the same architecture [24]. The main building block of InceptionTime is the inception module. The core idea of an inception module is that utilize multiple filters simultaneously to an input time series. The module involves filters of different lengths enabling the network to automatically extract related features from both long and short time series.

The network consists of a sequence of Inception modules followed by a Global Average Pooling layer and a Dense layer with a softmax activation function. We make some changes after the Global Average Pooling layer in the framework of the proposed method to adapt our case. Furthermore, InceptionTime introduces a further element within its network's layers: residual connections at every third inception module. Thus decreasing the vanishing gradient problem by allowing a direct flow of the gradient.

The main idea of the Siamese network is that two input signals go through a shared network generating fixed-length feature vectors. Assuming that the appropriately trained neural network model, we can make the following hypotheses: If two input instances belong to the same instance classes, their feature vectors must be similar, if two input instances belong to different instance classes, their feature vectors will also be different. So absolute differences of the two feature vectors must be very different in both the above cases. The proposed network in shown in Figure 1.

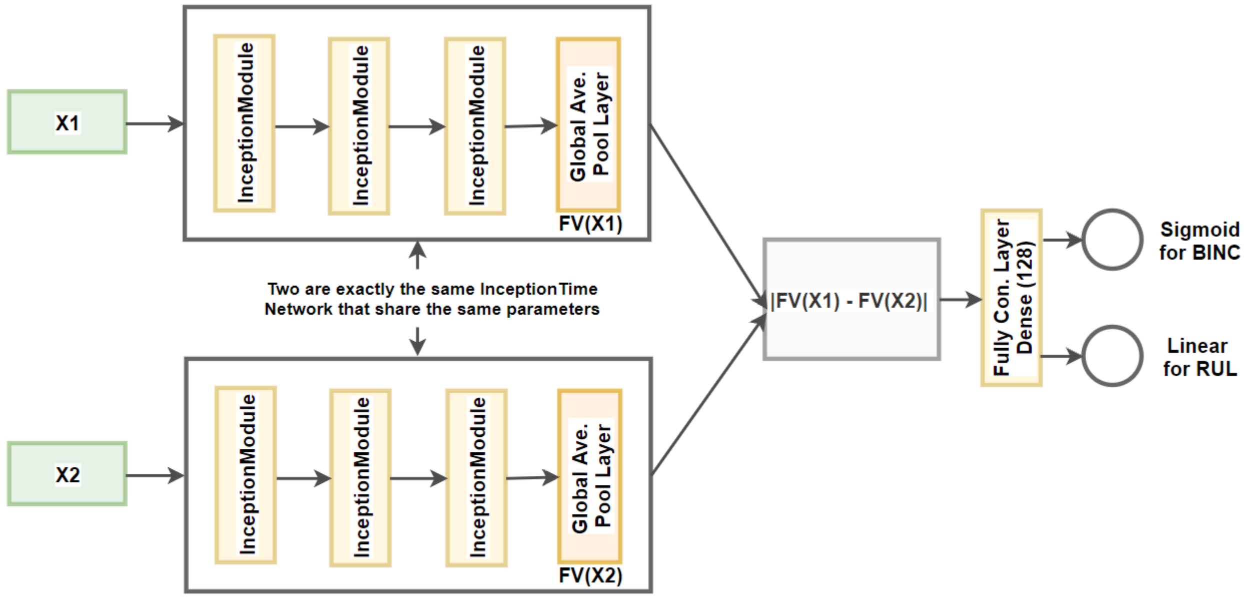

Figure 1. *The architecture of proposed Siamese InceptionTime Network.*

1) **Data preparation:** We first prepare pairs of healthy and failure samples as input for the Siamese network. The sub-dataset has N attributes, and the lifetime of the sensors are L_t cycles, i.e., machinery lifetime (see

Figure 2). The data is obtained by sliding the time window with window size w_l cycles, and the sliding step size is one cycle. For a given sequence we will have [L_t-w_l, L_t- w_l-1,. . .,0]. w_l is selected as 30 for the FD001 sub-dataset, because it gives the highest precision between results from other time window sizes as proposed [13] and with large time window size includes more data points within the time frame, so the degradation pattern may be simpler. To evaluate all engine's performance in sub-test data and make the comparison between the compared and the results of the proposed method, we cannot increase the time window size because of min cycle in the test engine is 31. In our experiment, we select 14 sensor outputs as in [20]. We further normalize these values within [0, 1].

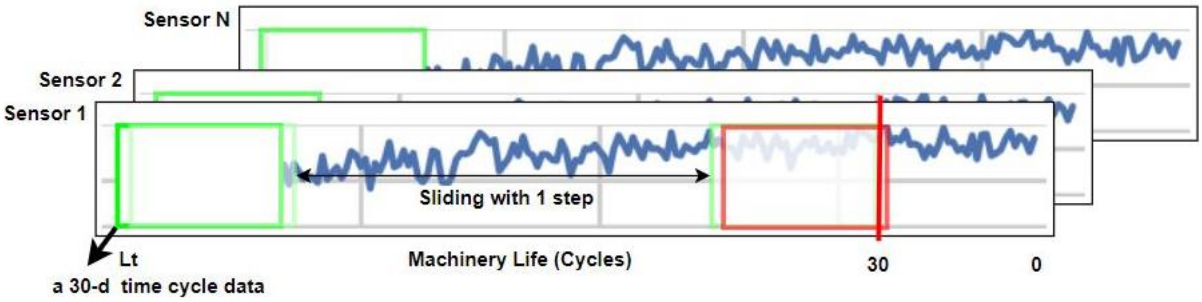

Figure 2. *Extracting health/failure samples*

Then, a breakpoint RUL value which is 30 specified (which is illustrated with a solid red vertical line in Figure 2) to discriminate between healthy and failure states for the FD001 sub-dataset, and according to this breakpoint value, healthy and failure samples were prepared. If the RUL value of input data is less than the breakpoint value, it was determined as a failure sample and other cases as an example of healthy samples. So, the input of the Siamese InceptionTime network is pairs of the combination of healthy and failure samples. For binary classification (BINC output, see Figure 1), data labeled as if a pair input data in the same status (in this study only pairing healthy sample vs healthy sample applied) then the label is 1 other case which is failure vs healthy samples then the label is 0. On the other hand, for output RUL (see Figure 1), for each pair of input data labeled as minimum RUL value of input pair samples.

2) **InceptionTime Siamese network:** We propose a Siamese network that is using InceptionTime Network for RUL estimation. The network includes two identical paths that share weights, and each is composed of three Inception modules followed by a Global Average Pooling layer. So, two input samples (x_1 and x_2) are passed through the Siamese InceptionTime Network to generate a fixed-length feature vector for each ($f_v(x_1)$ and $f_v(x_2)$), then, we compute the absolute distance between these feature vectors and passed to a fully connected layer (with a rectified linear unit activation) followed by another fully connected layer (with a sigmoid activation) determining the similarity of the two input. Depth of inception network set to 3 with 64 filters at each and three convolutions with [3, 5, 7] filters lengths. For training our network, we used Adam optimizer with learning rate 10^{-5} and batch size 64.

To evaluate the performance of the proposed Siamese Inceptiontime Network, root mean square error (RMSE) and Score metrics were used. RMSE is a frequently utilized metric for RUL prediction that gives the same penalties while the score has different penalties for early and late predictions [20]. If there is a prediction error *e* and the error between the predicted RUL and actual RUL is $e = RUL_{predict} - RUL_{actual}$ then RMSE and Score formulas are given in Eqs. (1) and (2) separately:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N} e_t^2} \tag{1}$$

$$Score = \sum_{t=1}^{N} s_t,\ s_t = \begin{cases} exp^{-e_t/13} - 1 & e_t < 0 \\ exp^{e_t/10} - 1 & e_t >= 0 \end{cases} \tag{2}$$

where e_t is the prediction error, *exp* denotes Euler's number and N is the test sample size.

4. Experiments and Results

We conducted experiments in the first dataset (FD001) of C-MAPSS to show the performance of the proposed method. We evaluate the performance of the proposed method and compare its results to a state-of-art best model [20]. First, we only used the data from 15 engines that were randomly selected for training. Selected engines ids are [52, 42, 37, 35, 100, 68, 41, 66, 40, 81, 53, 49, 61, 76, 18]. We obtained 30000 pairs of data from the selected engines with the balanced version of the dataset by weighting randomly selecting samples from each engine according to the total dataset size to train the proposed method.

The Siamese InceptionTime network was employed to estimate the RUL of the C-MAPSS engine dataset. For testing, each sample is to be predicted compared with the first healthy sample of a specific engine, and pairs are given to the network for prediction. The test root mean square error (RMSE) is 16.35 and the score is 460.77 (compare these with the best results in the literature [20] with 11.96 and 229.00 respectively). The predicted RUL of each test engine is shown in Figure 3. With the proposed method generally, the estimated value of the RUL can be observed close to the true value. We also compare the proposed model with some other model results in the literature.

The comparison result is quantified using RMSE of RUL prediction for the FD001 test dataset as listed in Table 2. It is noted that the developed proposed model outperforms SVR, MLP, DCNN, and RF. Also, the results of the proposed method are close to GB and MODBNE. DAG LSTM-CNN gives better than the proposed result. However, it should not be forgotten that the proposed method uses only 15 engines data for training.

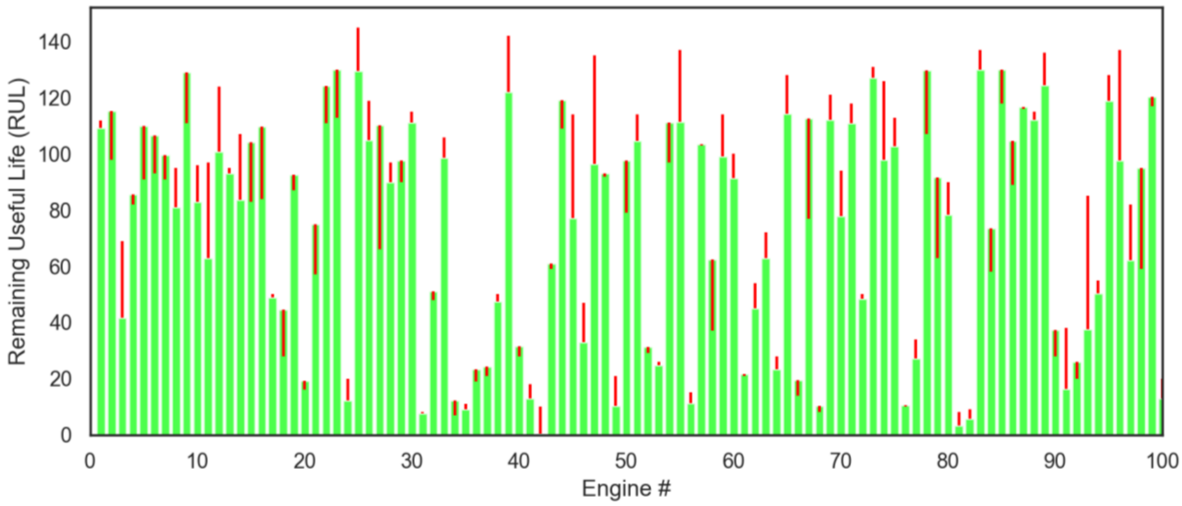

Figure 3. *Predicted RUL for each engine in the FD001 test sub-dataset by the Siamese InceptonTime network. The wider green bar shows the predicted RUL and the narrower red line exhibits the differences of RUL values between actual and predicted.*

Table 2. *Engine RUL prediction comparisons, in RMSE on FD001 test dataset.*

Methods	RMSE
SVR [25]	20.96
MLP [25]	37.56
DCNN [25]	18.44
RF [26]	17.91
GB [26]	15.67
MODBNE [26]	15.04
DAG LSTM-CNN [20]	11.96
Proposed method	**16.35**

Figure 4 demonstrates the predicted RUL and binary similarities generated by our algorithm on the two of the engines for the FD001 test sub-datasets. The green line shows the binary similarity (the first healthy sample is compared sequentially with other samples of the engine) and gives some signals before the failure point indicated in the dashed red line via decreasing score between data pairs. Also, the RUL prediction results are shown in blue and successfully estimate predictions as close to actual values and give the alarm signal before the failure point by lowering the predicted RUL value.

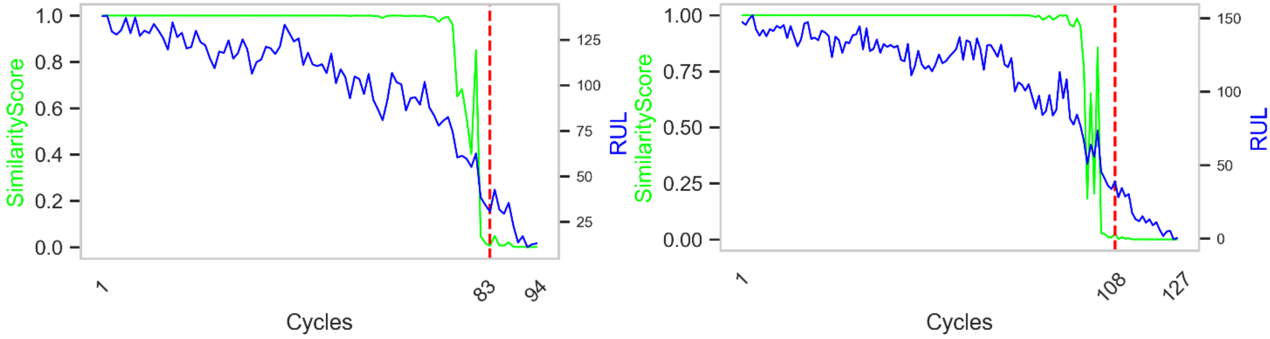

Figure 4. *Input data of first healthy sample vs others to the Siamese InceptionTime Network output results on engine #41 and #42 in FD001 test sub-dataset. Green shows binary similarity while blue shows the estimated RUL (best seen in color). The vertical dashed red line illustrates the degradation point of the engine.*

Classification results of the FD001 test dataset are illustrated in Figure 5 in a confusion matrix. The accuracy of our classifier is 95% while the F1 score is 0.96. This shows the discriminant nature of the windows before and during failure. The Siamese network captures this very accurately.

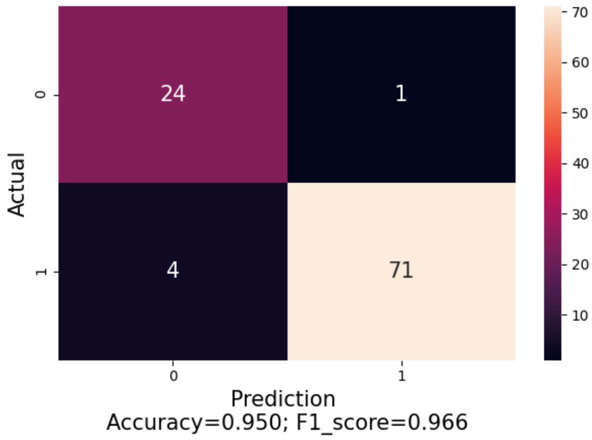

Figure 5. *Classification results input of pairs healthy/failure samples of test data.*

In order to evaluate compare the proposed method to a base-line system when a smaller amount of data is used, we have implemented the network described in [20]. For both proposed and compared models we obtained the best hyperparameters and run the model over one thousand epochs. We train these models with random hyperparameters and compared their results in RMSE and Score metrics.

We simulate the amount of data (smaller to larger) by using a fixed number of engines in the training (e,g., 5 engines randomly chosen from the training set). We change the number of engines from 5 to 24 indicating an increasing amount of data. We expect that 5 engines for training will be challenging and as the number increased, the models will perform better. For a fixed number of engines, models are trained 30 times. We call each of these as an *experiment*, and for each iteration, we select engines randomly and used the same engine's data for both models to compare performances on test data and recorded theirs results from with both hyperparameters and random parameters. For each experiment, we prepare 30000 data pairs from selected engines and train the proposed model for one thousand epochs with a simple early stopping strategy. The selected engines data for the baseline model is trained as described in [20]. The training procedure was repeated 30 times for both models.

RMSE and Score results that were obtained from hyperparameters on test data for both models are illustrated in Figure 6 while random parameters results are shown in Figure 7 with boxplots. Box plots result cut of values bigger than 50 for RMSE and 5000 for Score metrics for hyperparameter results while those values are 50 for RMSE and 15000 for Score metrics for random parameters. The proposed model gives a reasonable performance compared to the base-line model when optimal hyperparameters are used. For random parameters, the proposed model gives better results.

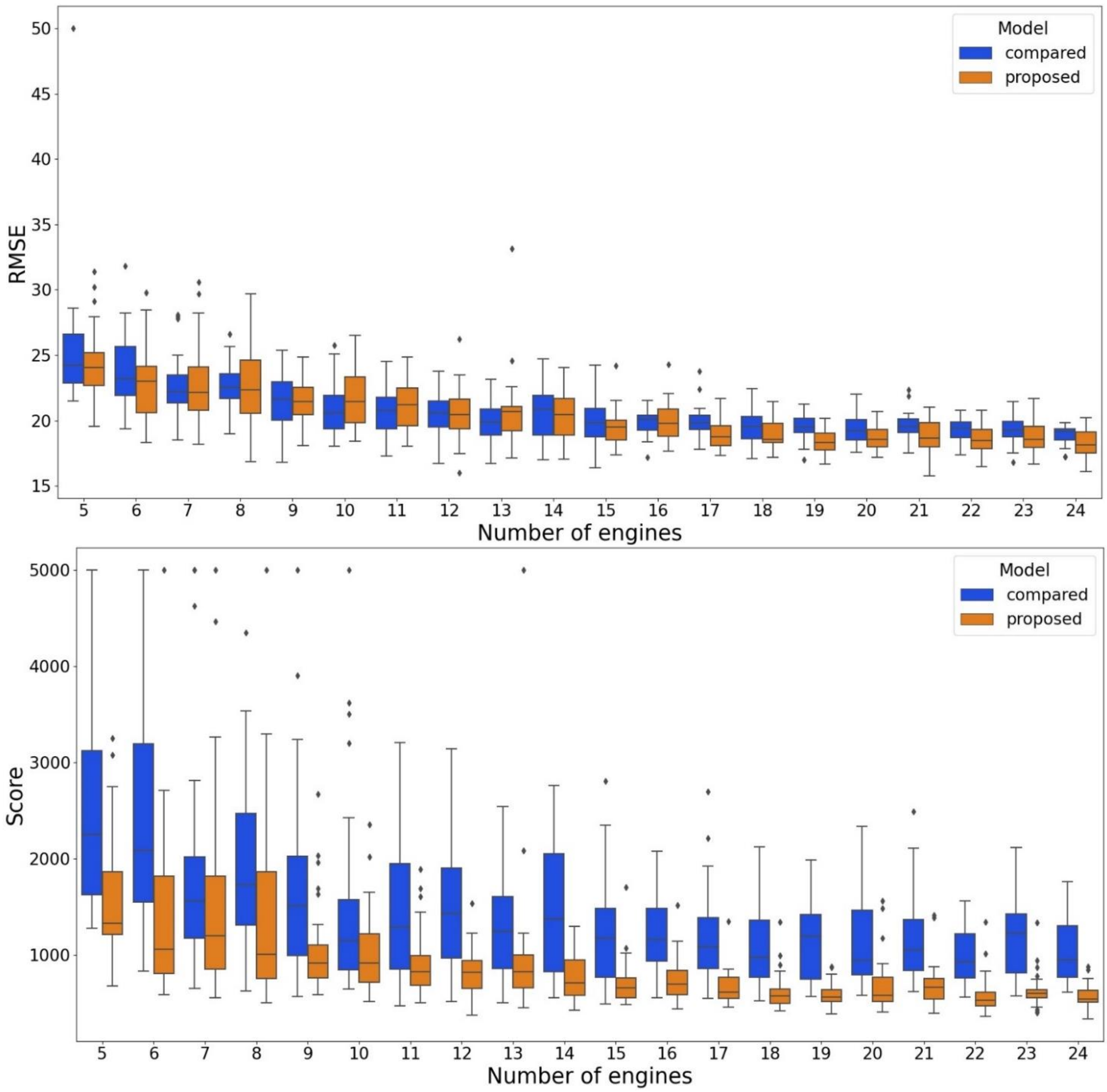

Figure 6. *Hyperparameters RMSE (top) and Score (bottom) results. Boxplot results were obtained from 30 experiments for changing number of engines.*

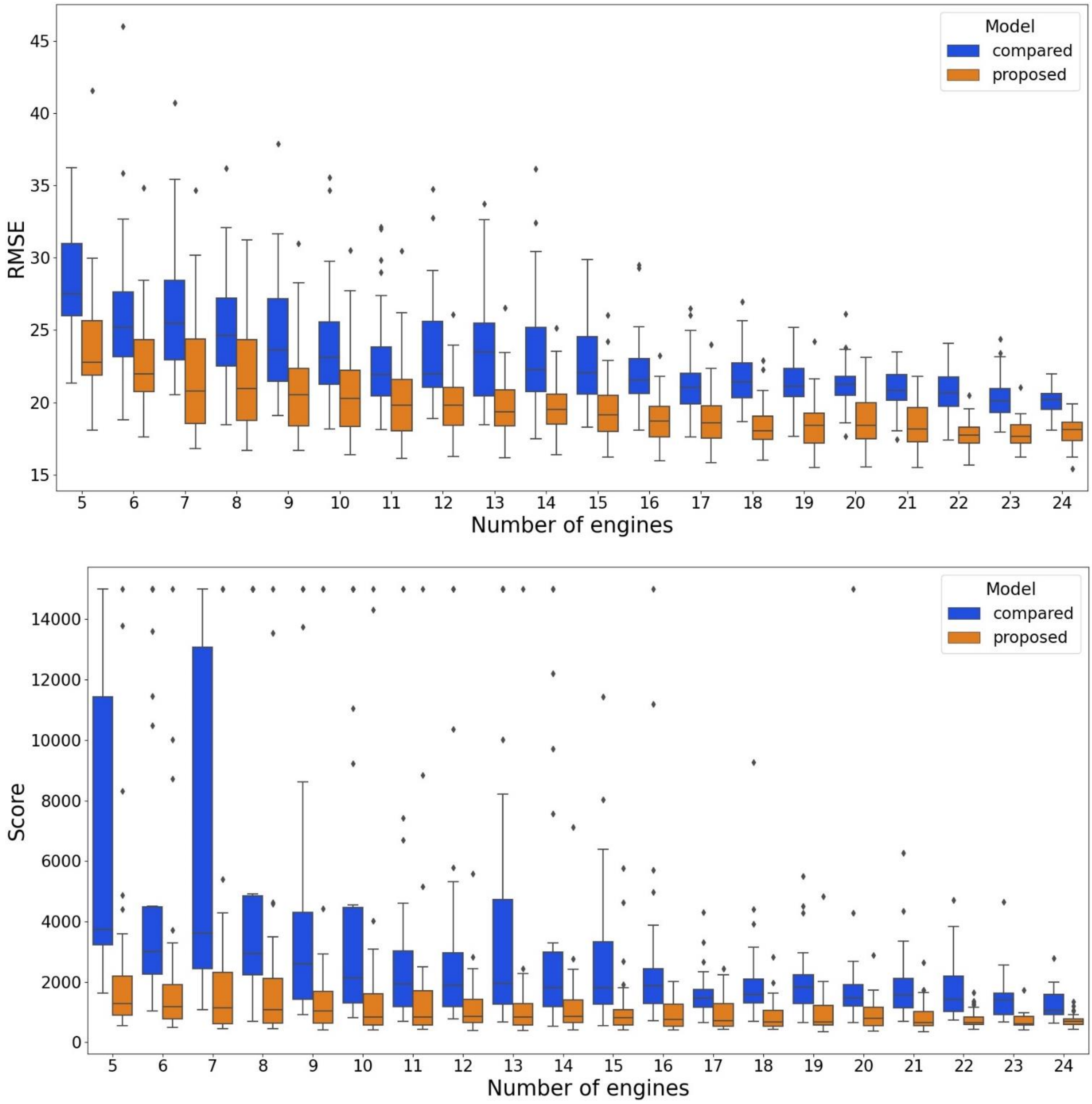

Figure 7. *Random parameters RMSE (top) and Score (bottom) results. Boxplot results were obtained from 30 experiments for changing number of engines.*

In order to see the effect of shortening the available sequence hence reducing the training data, we have trained the two models first with full data and then with first 15% of the data from the beginning is removed. The RMSE and Score metrics show slight improvements as seen in Figure 8. The proposed method performs slightly better when the sequence is shortened. This confirms our expectation that the model performs well with less data.

The improvement is slightly more with the base-line algorithm with shorter sequences. This might seem a bit counter intuitive. Yet, this result can still be explained with the fact that we are using a linear model for machine degradation. Ignoring the begging of the sequence removes any potential noise before the linear degradation starts.

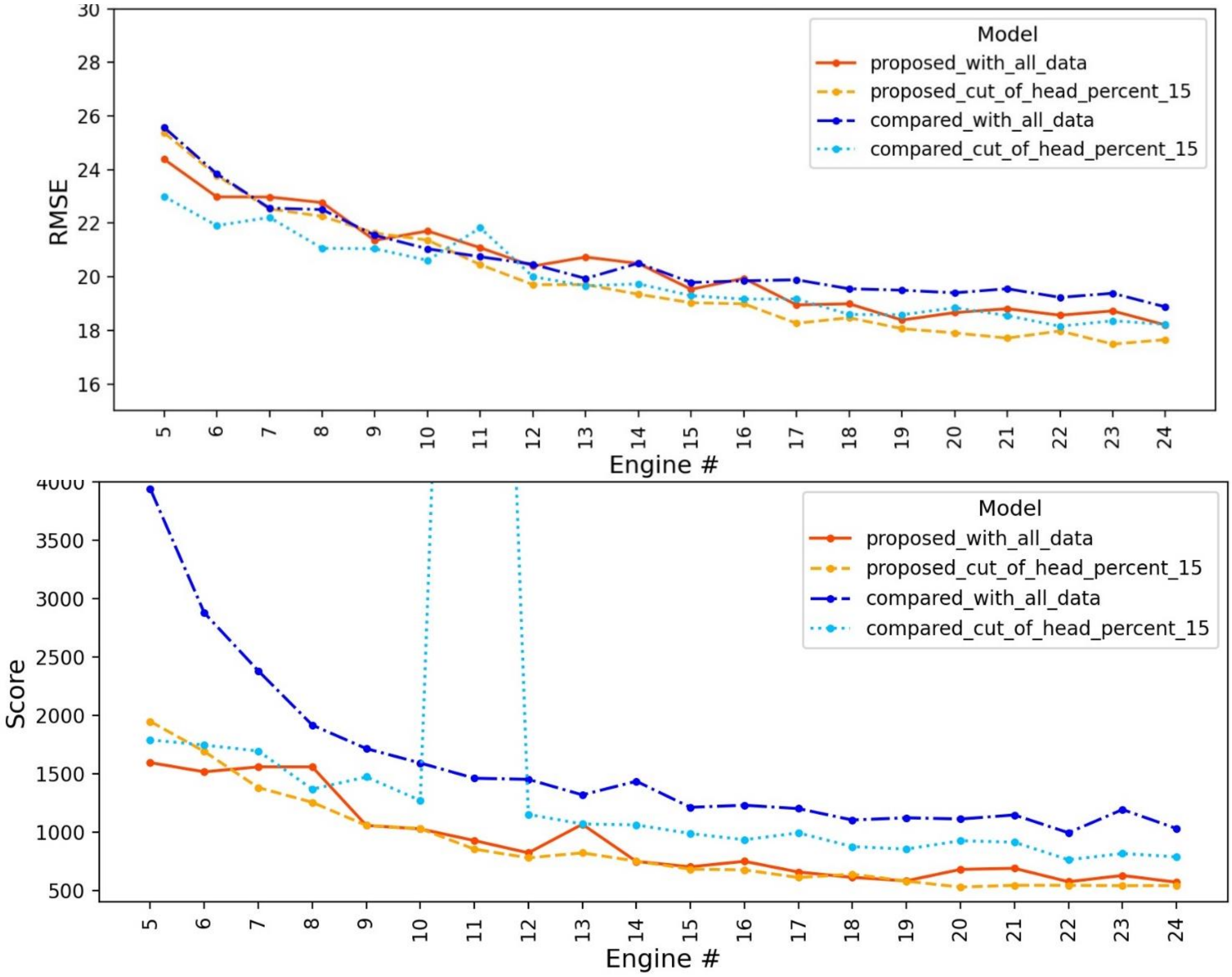

Figure 8. *Performance (top – RMSE, bottom – Score) change when the first 15% of each data is ignored during training. Mean of the metrics are given for changing number of engines used in training. Ignoring data in the beginning of a sequence improves results slightly.*

5. Conclusions

Accurate estimation of RUL has important advantages in many industrial applications where safety efficiency and reliability are among primary concerns. In the absence of a lot of failure data, we propose a Siamese network with temporal data processing capabilities for fault diagnosis classification and regression. Since typical deep learning models do not train well with small sets of training samples, the proposed method allows multiple uses of the data for learning a window-based similarity metric. InceptionTime network allows temporal processing. The experiments comparing the proposed method with a state-of-art baseline method show improved performance when less data is available. While these results are promising, further architecture and hyperparameters optimizations are still needed as the reported performance is slightly lower than the best state-of-art methods. Further experimental validations will be done on the other C-MAPSS sub-datasets.

Declaration of Interest

The authors declare that there is no conflict of interest.

Acknowledgements

An earlier version of this paper was presented at the ICADA 2021 Conference and was published in its Proceedings (Title of the conference paper: "Siamese Networks for Predicting Remaining Useful Life of Machinery")

References

[1] D. Wang, K. Tsui, and Q. Miao, “Prognostics and Health Management: A Review of Vibration Based Bearing and Gear Health Indicators”. IEEE Access, 2018, pp. 665–676.

[2] R. Zhao, “Deep Learning and its Applications to Machine Health Monitoring”. Mechanical Systems and Signal Processing 115, 2019, pp. 213–237.

[3] X. Si, “Remaining Useful Life Estimation - A Review on the Statistical Data Driven Approaches”. European Journal Operation Research, 213, 2011, pp. 1–14.

[4] Ö. Eker, F. Camci, and I. Jennions, “Major Challenges in Prognostics: Study on Benchmarking Prognostic Datasets”, Proc. European Conference of the Prognostics and Health Management (PHM) Society, 2012.

[5] N. Gebraeel, A. Elwany, and Jing Pan. “Residual Life Predictions in the Absence of Prior Degradation Knowledge”. IEEE Transactions on Reliability 58. 2009, pp. 106–117.

[6] O. Eker et al. “A Simple State-Based Prognostic Model for Railway Turnout Systems”. IEEE Transactions on Industrial Electronics 58, 2011, pp. 1718–1726.

[7] Y. Bengio, Y. LeCun, and G. E. Hinton, “Deep learning for AI”, Communications of the ACM. 64, 2021, pp. 58–65.

[8] G. Koch, R. Zemel, and R. Salakhutdinov. “Siamese Neural Networks for One-shot Image Recognition”. Proc. International Conference on Machine Learning, 2015.

[9] C. Gal´an et al. “A hybrid ARIMA-SVM Model for the Study of the Remaining Useful Life of Aircraft Engines”. Journal of Computational and Applied Mathematics, 346, 2019, pp. 184–191.

[10] G. A. Susto and A. Beghi. “Dealing with Timeseries Data in Predictive Maintenance Problems”. Proc. IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016, pp. 1–4.

[11] R. Khelif et al. “Direct Remaining Useful Life Estimation Based on Support Vector Regression”. IEEE Transactions on Industrial Electronics 64, 2017, pp. 2276–2285.

[12] S. Behera et al. “Ensemble Trees Learning based Improved Predictive Maintenance using IoT for Turbofan Engines”. Proc. ACM/SIGAPP Symposium on Applied Computing, 2019.

[13] C. Zhang, J. H. Sun, and K. Tan. “Deep Belief Networks Ensemble with Multi-objective Optimization for Failure Diagnosis”. Proc. IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 32–37.

[14] G. K. Durbhaka and B. Selvaraj. “Predictive Maintenance for Wind Turbine Diagnostics using Vibration Signal Analysis based on Collaborative Recommendation Approach”. Proc. International Conference on Advances in Computing, Communications, and Informatics (ICACCI), 2016, pp. 1839–1842.

[15] J. Xu, Y. Wang, and L. Xu. “PHM Oriented Integrated Fusion Prognostics for Aircraft Engines Based on Sensor Data”. IEEE Sensors Journal 14, 2014, pp. 1124–1132.

[16] X. Li, Q. Ding, and J. Sun. “Remaining Useful Life Estimation in Prognostics using Deep Convolution Neural Networks”. Reliability Engineering Systems, 172 (2018), pp. 1–11.

[17] J. Zhang et al. “Long Short-term Memory for Machine Remaining Life Prediction”. Journal of Manufacturing Systems 48 (2018), pp. 78–86.

[18] A. Z. Hinchi and M. Tkiouat. “Rolling Element Bearing Remaining Useful Life Estimation Based on a Convolutional Long-short-term Memory Network”. Procedia Computer Science 127 (2018), pp. 123–132.

[19] A. Zhang et al. “Transfer Learning with Deep Recurrent Neural Networks for Remaining Useful Life Estimation”. Applied Sciences 8 (2018).

[20] J. Li, X. Li, and D. He, “A Directed Acyclic Graph Network Combined with CNN and LSTM for Remaining Useful Life Prediction”, IEEE Access, June 2019, pp. 75464–75475.

[21] S. C. Hsiao, “Malware Image Classification Using One-Shot Learning with Siamese Networks”. Proc. Knowledge-Based and Intelligent Information & Engineering Systems (KES2019). 2019.

[22] A. Saxena, “Damage Propagation Modeling for Aircraft Engine Run-to-failure Simulation”. Proc. International Conference on Prognostics and Health Management, 2008, pp. 1–9.

[23] I. Fawaz, H. Lucas, B. Forestier, “InceptionTime: Finding AlexNet for Time Series Classification”. Data Mining and Knowledge Discovery, 34, pp. 1936–1962, 2020.

[24] C. Szegedy et al. “Going Deeper with Convolutions”. Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[25] G. S. Babu, P. Zhao, X-L. Li, “Deep Convolutional Neural Network-based Regression Approach for Estimation of Remaining Useful Life”. Proc. International Conference on Database Systems for Advanced Applications, 2016. pp. 214–28.

[26] C. Zhang, P. Lim, AK. Qin, KC. Tan, “Multi-objective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics”. IEEE Transactions on Neural Network and Learning System 2016, pp. 1–13.