Prediction of Remaining Useful Life of Turbofan through a Similarity Approach: the CMAPSS Dataset Case Study

Francesco Maione[1], Paolo Lino[1], Ottar L. Osen[2], Erlend M. Coates[2], Giuseppe Giannino[3], and Guido Maione[1]

***Abstract*— The digitalization of industry is driving new challenges in predictive maintenance, particularly with machine learning and deep learning. A common strategy is to train predictive algorithms using as many data as possible from systems of the same type. This paper proposes an alternative approach to predict the remaining useful life (RUL) of turbofan engines using only a limited amount of data from applications that exhibit similar time behaviors to those of the monitored asset. The basic assumption is that engines designed for similar applications and operating under comparable conditions will exhibit analogous degradation. To learn the most relevant degradation patterns of a tested engine, the approach consists of: *a*) determining the two most similar training engines from the CMAPSS dataset by two different clustering mechanisms; *b*) training and testing an LSTM prediction using two different techniques. The first technique averages data from the two most similar engines to train and test the LSTM; the second technique uses one engine for training, another for testing. Preliminary results show that the second technique outperforms the first and yields higher accuracy than the conventional approach based on the entire dataset. Further validation tests are needed to confirm the results and better analyze the robustness of the proposed approach, but the preliminary analysis gives a clear idea of the benefits of the approach.**

I. INTRODUCTION

While condition-based maintenance (CBM) is well established [1], recent advancements in artificial intelligence (AI) enable significant progress. Most industrial systems use a time-based preventive maintenance approach. Hence, maintenance is scheduled in time intervals based on heuristics and historical data. The main drawback of this approach is that inspection or intervention is planned based on statistical observations, not on the actual wear condition of the monitored asset. CBM overcomes this limitation by continuously monitoring and planning maintenance operations based on the real state-of-health of the asset [2], [3]. Several studies have already used AI algorithms [2]–[7]. Unlike the model-based approach [8] and the knowledge-based approach [9], the Machine Learning (ML) and Deep Learning (DL) algorithms do not require prior knowledge of the degradation model of the monitored asset [4]. Hence, they are easily implemented for CBM of complex systems like marine and aeronautical engines.

Among the DL algorithms applied for CBM, the so-called Long Short-Term Memory (LSTM) has been widely applied. In [6], LSTM was combined with a cyclical-update training logic to predict future time-series patterns of sensor measurements. In [4] and [5], LSTM is utilized to predict the future condition of a marine diesel engine. In [5], LSTM is added to another DL algorithm, called Variational Autoencoder, to improve the learning process. In previous works, LSTM has been applied to a maritime propulsion system, while in [10] and [11] LSTM has been applied to predict the Remaining Useful Life (RUL) of a jet engine and to multi-stage manufacturing processes. In [12], [13], different ML and DL algorithms have been applied both for supervised and unsupervised monitoring of marine engines. Finally, a Random Forest algorithm has been applied for CBM of marine electrical plants [14].

Despite promising results, some drawbacks remain in applying the ML and DL. The main weakness is the need for a large amount of data to properly train the model [15]. Moreover, as explained by [14], the training of algorithms is impeded by the scarcity of data that can be labeled as *faulty behaviors*, thus creating unbalanced datasets. To overcome this problem, some authors have focused on time-series prediction techniques that acquire data directly from the field [6], [16]. Coraddu et al. have trained algorithms only on *healthy data* to compare supervised ML algorithms with unsupervised ones, the last being trained without labeled data [17]. Healthy data were considered also in [18], [19], where semi-supervised algorithms have been developed for an engine turbofan and an unsupervised algorithm was applied for fault detection of maritime components.

Although the described works have shown satisfactory results, their main limitation is that they considered the monitored asset in its individuality. Namely, they are closely related to the system and its application: the engine is designed and installed on a ship or an aircraft, then the data acquired from this engine are used to apply the CBM for *that application*. In contrast, we here propose a new *similarity approach* for fault diagnosis and prognosis. The approach is based on the following main hypothesis: the same type of engines that are used in similar applications (e.g. ships sailing on the same trajectory, aircrafts following the same air route, etc.) show similar wear and fault patterns.

[1]F. Miaone, P. Lino, G. Maione are with the Department of Electrical and Information Engineering, Polytechnic University of Bari, 70125, f.maione@phd.poliba.it; guido.maione/paolo.lino@poliba.it

[2]O. L. Osen and E. M. Coates are with the Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Alesund, 6009, Norway ottar.osen@ntnu.no; erlend.coates@ntnu.no

[3]G. Giannino is with Isotta Fraschini Motori S.p.A., Bari, 70132, Italy giuseppe.giannino@isottafraschini.it

This also occurs in automotive engines that have similar powertrain or run on the same fuel. For example, consider electric vehicles [20] or compressed natural gas engines [21], [22], which are highly susceptible to the same specific faults. Just as it is easy to detect physical outliers when tracking vessels via Automatic Identification Systems, we propose treating engine degradation over time as a 'trajectory' in a multidimensional sensor space. We compare an engine's degradation path to historical data from similar engines. The expected path is based on historical data from similar, but not necessarily equal, engines. Examples of anomaly detection in ships trajectory are reported in [23]–[26].

To verify the similarity approach, we use the well-known CMAPSS Dataset [28]. It is a jet engine dataset provided by NASA and widely used to test RUL prediction strategies [10], [18]. In detail, this work shows results related to the second dataset provided by CMAPSS. First, we extract the most relevant sensor measurements from the dataset using the Pearson Correlation Coefficient. Second, we identify the training engines most similar to the tested engine by using two distinct clustering and alignment strategies (the k-Means and Dynamic Time Warping). Third, we train an LSTM network exclusively on these highly similar profiles—either individually or averaged. Finally, we compare the prediction accuracy with that of conventional methods. It is worth noting that the novelty of the proposed approach lies in the way in which the prediction problem is faced. In fact, the common way to perform prediction is to collect as many data as possible and to train the predictive algorithm on them. In contrast, we propose to focus the algorithm's learning process only on the most similar and significant data. This idea is supported by the observations reported in [27], where the authors emphasized the need to train a predictive algorithm for each cluster of the trajectory of a ship to avoid the risk that a single, comprehensive model fails to capture the complex connections between the data. Moreover, we remark that this work aims to propose a new way to solve the challenge of CMAPSS-based RUL prediction, and thereafter to adapt this new method to other datasets and industrial applications.

The rest of this paper is divided as follows: Section II describes the background; Section III shows the proposed method; Section IV provides an overview of the simulation results; finally, Section V draws the conclusions.

II. BACKGROUND

A. CMAPSS Dataset

The CMAPSS consists of four datasets that show how damage develops over time in an aircraft gas turbine engine. The dataset has been obtained from a dynamical simulation model, which was run at different wear conditions so that the damage propagates until the failure of some component occurs. The time-series datasets represent dynamic trajectories of different engines, consist of 24 simulated operating settings and variables, affected by noise to make simulation realistic, and are further divided into training and test sets. In the training set related to an engine, each row shows the values of the operating settings and variables and the "operational cycle" associated to the RUL. In few words, a RUL is associated to each sensor measurement, s. On the contrary, the unknown RUL must be predicted in the test set. In the current study, we used the second data-set from CMAPSS as it is the largest, composed by 260 training trajectories and 259 test trajectories. Only one fault mode is simulated in this dataset. Further details are in [28].

B. Pearson Correlation Coefficient

We recall that, if x is an input variable and y is a dependent variable, then the Pearson Correlation Coefficient (PCC) evaluates the linear correlation between them. It belongs to the interval $[-1, +1]$, where the limit values indicate a perfect negative or positive linear correlation. Values close to zero mean no linear correlation between x and y. The PCC is evaluated by the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where x and y are the two variables and $\bar{x}$ and $\bar{y}$ are their means computed on n samples.

The PCC is commonly used to identify suitable parameters as predictors or indicators of a marine engine [3], [29].

C. Long Short-Term Memory

Evaluating the time-series dependencies is the crucial point of predictive maintenance. Among the several algorithms available in the literature, the LSTM is the most widely used for this activity. It was proposed for the first time by [30]. and is based on the following equations:

$$f(t) = \sigma(W_f \cdot [h(t-1), x(t)] + b_f), \tag{2}$$

$$i(t) = \sigma(W_i \cdot [h(t-1), x(t)] + b_i), \tag{3}$$

$$\tilde{c}(t) = \tanh(W_c \cdot [h(t-1), x(t)] + b_c), \tag{4}$$

$$c(t) = f(t)\, c(t-1) + i(t)\, \tilde{c}(t), \tag{5}$$

$$o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o), \tag{6}$$

$$h(t) = o(t)\ \tanh(c_t), \tag{7}$$

where t is time, $x(t)$ is the input, $h(t)$ is the hidden state, $f(t)$ is the forget state, $i(t)$ is the input state, $c(t)$ is the cell state, $o(t)$ is the output state, σ is the sigmoid activation function, W are weight matrices, b are biases. The forget cell eliminates the unnecessary information, the memory and update cell sigmoid functions update the information. Fig. 1 shows the basic LSTM architecture.

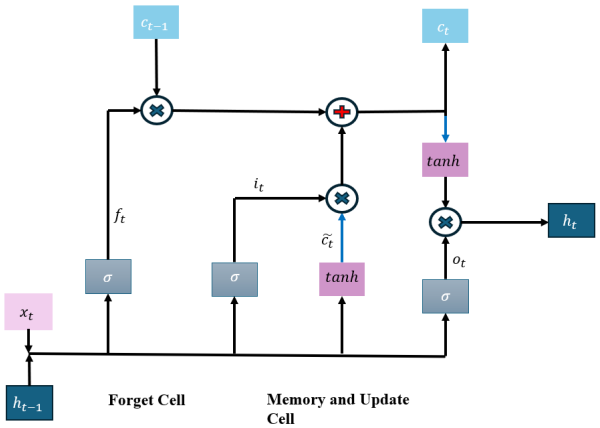

Fig. 1. Architecture of the LSTM algorithm.

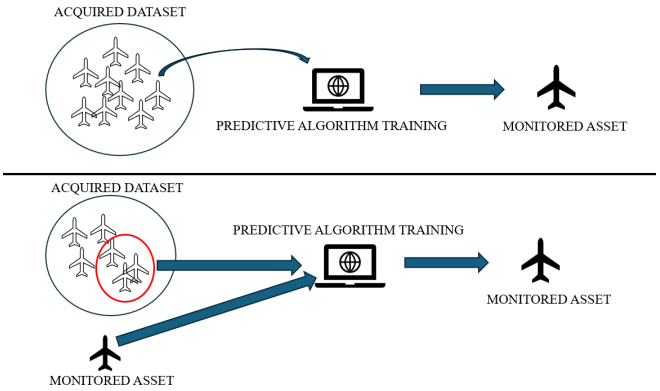

Fig. 2. Comparison between the common approach to train a predictive algorithm (above) and the proposed approach (below).

D. k-Means

The k-Means is an unsupervised algorithm widely used for clustering. As suggested by its name, it creates k mutually exclusive clusters, where k is inserted manually by the user based on the knowledge of the available data. We remark that, since it is an unsupervised algorithm, the data used for training are not labeled, while the algorithm puts the labels after training. The clusters are created by minimizing the distance between the data belonging to the same cluster, and by maximizing the distance between data belonging to different clusters. This approach guarantees that a single data sample belongs to only one cluster, so it is called "rigid".

Each cluster is determined by finding a centroid, namely the mean of all the data belonging to this cluster. The centroid is established through an iterative process during the training phase. Further details are in [31].

E. Dynamic Time Warping

The Dynamic Time Warping (DTW) algorithm is widely used to compare time-series. Unlike other methods, it does not compare the points of the time series based on their position, but rather compares the shape of the time series' patterns. In this way, it also allows us to compare dynamic trajectories with different time phases. Moreover, the strength of DTW lies in the possible application even when the trajectories have different lengths and in the case of noisy data. An overview of the DTW algorithm is in [23].

III. THE SIMILARITY APPROACH

The main idea of the proposed method is to detect faults by analyzing the similarity between different engine's behaviors. In fact, the main hypothesis is the following. Since the CMAPSS dataset is divided into smaller datasets with similar wear, healthy, and faulty behaviors, the hypothesis is that engines belonging to a dataset logged from a specific application will have similar time-series patterns before failing. Then, the method is divided into three sequential steps:

1. acquiring the data from the monitored engine;
2. finding the engines that, among the failed ones, show the most similar behavior;
3. training the predictive algorithm only on these engines.

This method differs significantly from the common approach to the prediction problem. In fact, predictive algorithms are usually trained by using as many data as possible; in contrast, we propose to use fewer data, but data that are more closely related to the degradation process of the monitored engine. In this sense, it is clear that only a subset of the original dataset is used to train the predictive algorithm. Fig. 2 explains the difference.

As noted in the introduction, the motivation of the proposed method lies in the fact that the usual algorithms can miss complex and crucial correlations between the data and the monitored asset because of training on large amounts of data. Moreover, our method uses only a subset of the original dataset and allows the algorithm to focus only on the most significant data [27]. Another important difference is that the methods widely applied in applications are usually implemented *offline*, while our method is an *online* approach. Hence, training is in real time since the model gets new data from the monitored engine as they are acquired.

It is clear that identifying the right cluster to which the test engine under consideration belongs to is the most crucial step to successfully implement the method. For this reason, we select a reference engine R, extract the most important measurements by applying PCC, and then choose the engines that are most similar by the following procedure, which was presented in the introduction:

1. The k-means algorithm is used to partition the engine dataset into distinct clusters.
2. The cluster to which the reference engine R belongs is determined.
3. The DTW is applied to the engines in this cluster and the l most similar ones to R are extracted.
4. The DTW is applied to the entire dataset to find other l engines that are most similar to R.
5. The two engines E1 and E2 in common to both methods of steps 3. and 4. are used to train the predictive algorithm.

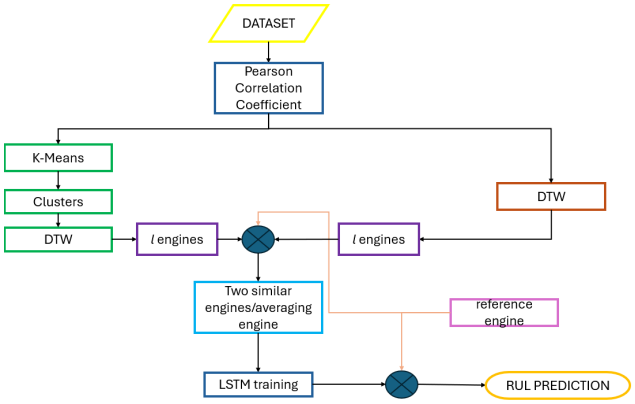

Fig. 3. Workflow of the proposed method for RUL prediction.

Once the two most similar engines E1 and E2 are identified, the next step is to choose how to train and test the LSTM predictive algorithm. In this work, we compare two different techniques:

- the first one is creating an average engine E3 from E1 and E2 and then training and testing the LSTM on E3.
- the second technique is training the LSTM on E1 and testing it on E2.

After the two techniques have been evaluated, the technique with the best training and test performance is chosen. The performance is evaluated by calculating the Root Mean Square Error (RMSE).

Finally, the best predictive algorithm is used to predict the RUL of the reference engine R. In detail, defined as $s_1, s_2, ..., s_n$ the successive sensor measurements obtained until the failure of the jet engine, the RUL is predicted through the *look back* and *look ahead* logic. Namely, the previous n sensor measurements are divided into m sliding windows of length $m < n$. The measurements $s_1, s_2, ..., s_m$ are used as inputs to the predictive algorithm, and the RUL associated to the s_{m+1} sensor measurement, say RUL_{m+1}, is the output. Then, the following m measurements are used to predict RUL_{2m+1}. And so on. An overview of the proposed RUL prediction method is given by the Fig. 3.

IV. SIMULATION RESULTS

The algorithms have been implemented in the Python programming language with the well-known TensorFlow (for the LSTM), scikit-learn (for the k-Means), and DTAIDistance (for the DTW) libraries.

As previously noted, among the four available datasets inside CMAPSS, we have chosen the FD002 dataset because it has the highest number of trajectories, namely 260. Consequently, the first engine of the FD002 test set was considered as reference engine R for the RUL prediction, while the FD002 training set will be used for the similarity approach.

Then, the PCC was applied to choose the best variables for the RUL prediction. The results are shown in Fig. 4. As the figure shows, the sensors number 2, 15, 21 have the lowest correlation ratio, therefore they have been removed.

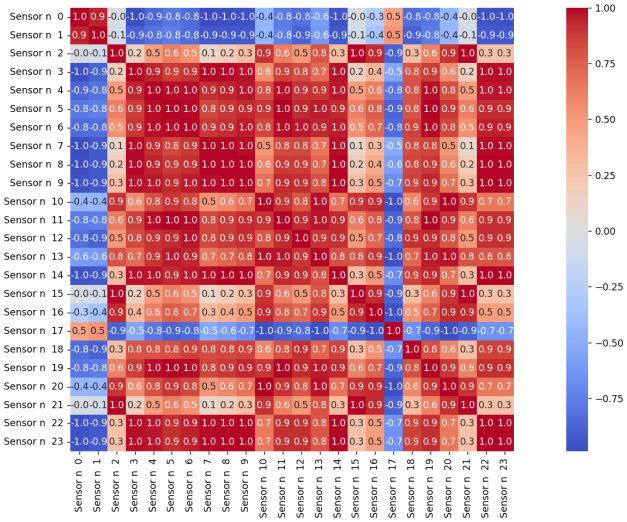

Fig. 4. PCC results for the FD002 CMAPSS dataset.

After choosing the most meaningful variables, the DTW was applied to the training dataset of the engines belonging to the FD002 dataset. The DTW parameters are the following: the distance is evaluated through the Euclidean function and Symmetric2 is the option for the Step Pattern. The results of DTW show that the engines most similar to the reference R are given by the engine numbers 13, 2, 155, 164, 42, 202 in ascending order of similarity.

Then, the next step is to apply the k-Means to determine the clusters. Before that, the number of clusters must be chosen. It has been set equal to 4 because we consider that a turbofan is subject to the following possible working conditions during an operating cycle, i.e. a trip: departure, flight, landing, and wear. We consider an independent cluster for the wear for the following reason: we assume that the wear is not always present in all the flights, but it slowly becomes evident during the flight time, since the departure and landing may be too short to correctly visualize the phenomenon. Note that the CMAPSS datasets assume that the initial wear of an engine is not a faulty condition and, due to some tolerance, is assumed as a normal working condition. Therefore, considering and deriving a separate cluster for wear allows us to model only the severe wear conditions, therefore to improve the separation from healthy conditions. After the k-Means has divided the training set into four clusters, it has been applied to identify the cluster to which the reference engine R belongs. Then, the DTW has been applied to all the engines of the clusters to find the most similar ones: the engines number 42, 202, 218, 101, 109, 28 were found.

The results show that the two engines in common to both methods are those numbered 202 and 42. Now, we apply the two mentioned techniques to train and test the LSTM:

- The first consists in creating an average engine by the values of the engines 202 and 42 and by training and testing the predictive algorithm on it.
- The second trains the predictive algorithm on one of the

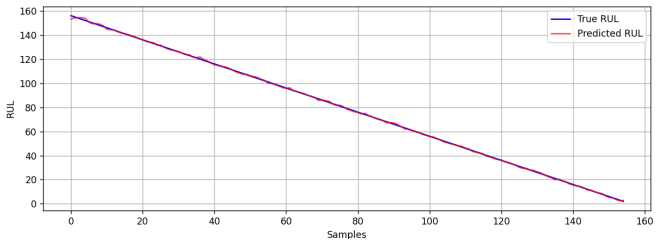

Fig. 5. LSTM predictions on the training set (Solution 2).

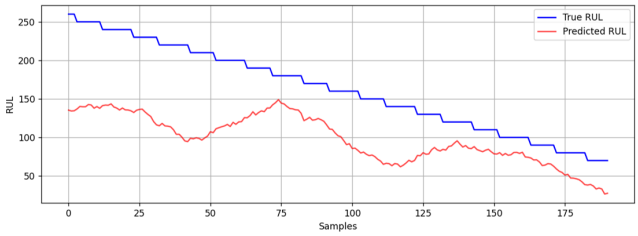

Fig. 6. LSTM predictions on the test set (Solution 2).

two extracted engines (42) and tests the algorithm on the second one (202).

The LSTM predictive algorithm is trained with a *look-back* set to 65 by trial-and-error and a *look-ahead* set to 1. The LSTM parameters obtained by trial and error during the training phase are the following: the learning rate has been set equal to 0.01; 200 and 32 have been the hidden neurons and the batch size, respectively; finally, the Stochastic Gradient Descent has been chosen as optimizer.

Then, named *Solution 1* the one obtained with the average engine and *Solution 2* the second one, Tab. I gives the RMSE for the two solutions. Solution 2 has a higher RMSE during the training phase, but has a better generalization capacity during the test phase. We may suppose that this improvement is due to the fact that the second LSTM has been trained on engine number 42 and tested on engine number 202. In this way, the LSTM has been trained on the entire engine 42 trajectory, starting from the healthy condition and ending at the faulty condition. This approach allows the LSTM to learn the entire life of the engine, thus generalizing the wear propagation. In contrast, in Solution 1, the LSTM has been trained and tested on the trajectory of a unique average engine. It means, that during the training phase, the LSTM learns the pattern of the initial 75% (training size) of the average trajectory, and tries to generalize it for the remaining 25% (test size).

TABLE I
RMSE RESULTS

RMSE in training: Solution 1	1.08
RMSE in training: Solution 2	2.13
RMSE in test: Solution 1	28.60
RMSE in test: Solution 2	17.62
RMSE in training: Traditional Approach	14.4
RMSE in test: Traditional Approach	46.1

More details regarding training and test of Solution 2 are shown in Fig. 5 and Fig. 6. The figures show that performance is lower when considering the test set, especially for the initial samples of the set. This suggests to us that the two engines considered in Solution 2 started from different levels of wear, while at the end they tend to have the same degradation, thus allowing an improved prediction.

Since the second solution provided better results, it will be chosen to predict the RUL of the reference engine taken from the FD002 test set. The predicted RUL for this engine is 27, while the true value is 18. The difference is just 9 cycles, which is a good result considering the complexity of the systems that are present in the CMAPSS dataset. The results of the last prediction highlighted that the developed similarity approach is suitable for training and testing predictive algorithms.

Finally, to show the effectiveness of the new method, the results obtained by training the LSTM only on the two most similar engines have been compared with the results obtained by the LSTM trained with a *fleet-level* approach, which is a traditional one. The LSTM is trained on all the engines belonging to the FD002 training set. The training has been conducted by dividing the set into sliding windows with a length of 65 sensor measurements and by dividing the overall set in a 75% amount of data for training and a 25% amount for test. The results, even if preliminary, are aligned with those shown in [18], and are shown in Tab. I.

The traditional approach performs better in training than in test, which means it is not able to generalize all the information learned from the entire dataset. This result is confirmed by applying the traditional approach to predict the RUL of the reference engine R by the FD002 test set; in fact, the predicted RUL is 5 while the true value is 18, with a difference of 13 cycles, which is an error higher than with the proposed approach based on only the two most similar engines. This preliminary result confirms that using fewer but more relevant data is helpful to increase the performance of the predictive algorithms. Too much information with the risk of missing important details is avoided. It is remarked that future tests will be applied on the overall CMPASS dataset to further validate and improve the proposed approach.

Note that the main problem with the proposed method is to have a sufficient number of *run-to-failure* acquisitions from real applications to be able to make the required prediction. However, the development of new technologies will overcome this weakness, so that in the future enough data could be collected and thereby correctly used to predict RUL in industrial applications akin to the engine turbofan.

V. CONCLUSIONS

The RUL prediction is one of the most challenging problems in condition-based maintenance. The complexity increases in case of compound systems such as jet engines, where several degradation modes occur and many sensor measurements must be considered. This complexity often

makes it difficult to train a single predictive algorithm for all the faulty behaviors. For this reason, this study aims to change the usual way of applying ML and DL algorithms. We propose to train predictive algorithms not on all acquired data but to use only data coming from similar engines with similar behaviors. In fact, the main assumption is that turbofan jet engines that are used in similar applications show similar healthy and faulty behaviors. To test this idea on data from jet engines, we have used the CMAPSS dataset. A reference engine has been extracted from this dataset, then two different methods have been introduced to find the most similar engines to the reference: the first method applies k-means for clustering and then DTW to find the most similar engines in the cluster of the reference engine; the second method applies only the DTW to the overall dataset. The two engines that are common to both methods have been selected. Finally, the LSTM predictive algorithm has been trained on both engines or on an average engine. The results have shown that an average engine alone gives worse performance than two similar engines. We finally remark that the proposed approach is a preliminary proposal for a new way to face the RUL prediction. In future work, the approach will extend the results to the entire CMAPSS dataset, with test-set validation and statistical analysis for robustness.

ACKNOWLEDGMENT

Activities described in this paper have been developed within the Fincantieri project "Wave 2 the Future" (W2F) funded by the European Union –NextGenerationEU. W2F project is part of IPCEI Hy2Tech.

REFERENCES

[1] A. H. C. Tsang, "Condition-based maintenance: tools and decision making," J. Qual. Maint. Eng., Vol. 1, No. 3, pp. 3—17, 1995.

[2] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifiers approach," IEEE Trans. Ind. Inform., vol. 11, no. 3, pp. 812–820, 2015.

[3] V. J. Jimeneza, N. Bouhmala, A. H. Gausdal, "Developing a predictive maintenance model for vessel machinery," Journal of Ocean Engineering and Science, vol. 5, issue 4, pp. 358-386, 2020.

[4] P. Han, A. L. Ellefsen, G. Li, V. Æsøy and H. Zhang, "Fault prognostics using LSTM networks: application to marine Diesel engine," IEEE Sensors Journal, vol. 21, no. 22, pp. 25986–25994, 2021.

[5] P. Han, A. L. Ellefsen, G. Li, F. T. Holmeset and H. Zhang, "Fault detection with LSTM-based variational autoencoder for maritime components," IEEE Sens. J., vol. 21, no. 19, pp. 21903–21912, 2021.

[6] F. Maione, P. Lino, G. Giannino and G. Maione, "A new deep learning strategy to predict time-series of marine Diesel engines," IEEE Transactions on Industry Applications, vol. 62, no. 2, pp. 2300–2310, March-April 2026.

[7] F. Maione, P. Lino, G. Maione, G. Giannino, "A machine learning framework for condition-based maintenance of marine diesel engines: A case study," Algorithms, 2024, 17(9):411.

[8] I. El-Thalji and E. Jantunen, "Fault analysis of the wear fault development in rolling bearings," Engineering Failure Analysis, vol. 57, pp. 470–482, 2015.

[9] N.P. Ventikos, P. Sotiralis, E. Annetis, "A combined risk-based and condition monitoring approach: developing a dynamic model for the case of marine engine lubrication," Transportation Safety and Environment, Vol. 4, Issue 3, Sep. 2022.

[10] D. Dong, X.-Y. Li and F.-Q. Sun, "Life prediction of jet engines based on LSTM-recurrent neural networks," 2017 Prognostics and System Health Management Conference (PHM-Harbin), Harbin, China, 2017, pp. 1–6, 2017.

[11] H. N. Hady, R. H. Hadi, O.H. Hassoon, A. M. Hasan, A. M., and A. J. Humaidi, "Predicting process quality in multi-stage manufacturing using ae-bila: an autoencoder-bilstm with attention mechanism," Engineering Research Express, vol. 7(1), 2025.

[12] F. Cipollini, L. Oneto, A. Coraddu, A. J. Murphy, D. Anguita, "Condition-based maintenance of naval propulsion systems with supervised data analysis," Ocean Eng., vol. 149, pp. 268–278, 2018.

[13] F. Cipollini, L. Oneto, A. Coraddu, A. J. Murphy, D. Anguita, "Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback," Reliability Engineering & System Safety, vol. 177, pp. 12–23, Sep. 2018.

[14] A. Bakdi, N. B. Kristensen, and M. Stakkeland, "Multiple instance learning with random forest for event logs analysis and predictive maintenance in ship electric propulsion system," IEEE Transactions on Industrial Informatics, vol. 18, no. 11, pp. 7718-7728, Nov. 2022.

[15] N. Kougiatsos and R. Vessa, "A distributed cyber-physical framework for sensor fault diagnosis of marine internal combustion engines," IEEE Trans. Control Syst. Technol., vol. 32, no. 5, pp. 1718-1729, Sept. 2024.

[16] W. B. Yousuf, S. M. U. Talha, A. A. Abro, S. Ahmad, S. M. Daniyal, N. Ahmad, A. A. Ateya, "Novel Prognostic Methods for System Degradation Using LSTM," IEEE Access, vol. 12, pp. 191955-191966, 2024

[17] A. Coraddu, L. Oneto, D. Ilardi, S. Stoumpos, and G. Theotokatos, "Marine dual fuel engines monitoring in the wild through weakly supervised data analytics," Engineering Applications of Artificial Intelligence, vol. 100, April 2021, 104179.

[18] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," Reliability Engineering & System Safety, Vol. 183, March 2019, pp. 240–251.

[19] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, and H. Zhang, "An unsupervised reconstruction-based fault detection algorithm for maritime components," IEEE Access, vol. 7, pp. 16101–16109, 2019.

[20] J. Hong, F. Liang, J. Yang, S. Du, "An exhaustive review of battery faults and diagnostic techniques for real-world electric vehicle safety," Journal of Energy Storage, vol. 99, Part A, 2024, 113234.

[21] P. Lino, G. Maione, "Design and simulation of fractional-order controllers of injection in CNG engines," IFAC Proceedings Volumes, vol. 46, Issue 21, 2013, pp. 582–587.

[22] P. Lino, G. Maione, "Fractional order control of the injection system in a CNG engine," 2013 European Control Conference (ECC), Zurich, Switzerland, 2013, pp. 3997–4002.

[23] C. Zhang, S. Liu, M. Guo, and Y. Liu, "A novel ship trajectory clustering analysis and anomaly detection method based on AIS data," Ocean Engineering, vol. 288, part 2, 15 Nov. 2023, 116082.

[24] H. Li, W. Li, S. Wang, H. Yang, J. Guan, and Y. Zhang, "STAD: Ship trajectory anomaly detection in ocean with dynamic pattern clustering," Ocean Engineering, vol. 313, part 3, 1 Dec. 2024, 119530.

[25] D. Xu, J. Yang, K. S. Qin, Y. Qi, and Z. Zhou, "Anomaly detection method for ship trajectory based on stay region mining," Ocean Engineering, vol. 331, 1 July 2025, 121364.

[26] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, "GeoTrackNet—A maritime anomaly detector using probabilistic neural network representation of AIS tracks and *a contrario* detection," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 6, pp. 5655-5667, June 2022.

[27] G. B. Karataş, P. Karagoz, and O. Ayran, "Trajectory pattern extraction and anomaly detection for maritime vessels," Internet of Things, vol. 16, December 2021, 100436.

[28] A. Saxena, K. Goebel, D. Simon and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 2008, pp. 1-9.

[29] H. Gharib and G. Kovács, "Development of a New Expert System for Diagnosing Marine Diesel Engines Based on Real-Time Diagnostic Parameters," Strojniški vestnik - Journal of Mechanical Engineering, Vol. 68, No. 10, pp. 642—653, 2022.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] McQueen, James B. "Some methods of classification and analysis of multivariate observations." Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.. 1967.