RUL Prediction using a fusion of Attention-based Convolutional Variational AutoEncoder and Ensemble Learning Classifier

Ikram Remadna, Labib Sadek Terrissa, Zeina Al Masry and Noureddine Zerhouni, Senior Member, IEEE

***Abstract*—Predicting the remaining useful life (RUL) is a critical step before the decision-making process and developing maintenance strategies. As a result, it is frequently impacted by uncertainty in a practical context and may cause issues. This paper proposes a new hybrid deep architecture that predicts when an in-service machine will fail to overcome the latter problem, allowing for an improved data analysis and dimensionality reduction capability providing better spatial distributions of features and increasing interpretability. A deep Convolutional Variational AutoEncoder with an Attention mechanism (ACVAE) has been developed and tested using the aero-engine C-MAPSS dataset. We defined two adapted threshold settings ($\alpha 1, \alpha 2$) by analysing the spatial distribution and minimizing the overlapping area between the degradation classes. To reduce the conflict zone, we used the soft voting classifier. The performance of our visual explainable deep learning model has reached a higher level of accuracy compared with previous existing models.**

***Index Terms*—Prognostics and Health Management, Deep Learning, Variational Autoencoder, Data visualisation, Attention Mechanism.**

I. Introduction

During the past years, ever-increasing quality requirements, decreasing costs, and improving production rates have exerted constant pressure on the companies. Therefore, the manufacturing industry is heading towards the smart Industry, or Industry 4.0 (I4.0) brought forward by new technologies, such as the Internet of Things, Big Data, Cloud Computing [1]. In this context, the volume of data is growing exponentially, representing a significant opportunity for the maturity of data science and analysis to deliver more effective and efficient maintenance services.

The remaining useful life (RUL) is a crucial characteristic for prognostics and establishing maintenance decisions. An efficient RUL estimation helps in reducing the cost of preventive and corrective maintenance. Uncertainty in RUL prediction remains a scientific problem that should be resolved since it affects the accuracy of Prognostics and Health Management (PHM) implementation in industries.

A novel approach for predicting RUL based on visual data analysis is proposed. Attention Convolutional Variational AutoEncoder (ACVAE) is used in order to automatically extract performance degradation features from multiple sensors. ACVAE effectively integrates convolution calculation with autoencoder to extract spatial information. Moreover, the attention layer is embedded between the encoder and decoder, which is used to dynamically increase the weights of the useful features in the encoding phase to make the network pay attention to these vital features for RUL classes estimation. The primary objective of applying the ACVAE is to provide a more structured and lower-dimensional representation of the data that shows the best distribution of the class over a 2D latent space and demonstrates how well the ACVAE generalizes. The encoder, part of the ACVAE, is leveraged for data projection in a 2D visualisation latent space. The input vectors are encoded and displayed into this 2D space, which helps the expert to visually analyze the spatial distribution of the training dataset. Three degradation classes are then defined according to two thresholds (α_1, α_2). The expert aims to determine the appropriate threshold setting by minimizing the overlapping region between the degradation classes by analyzing the spatial distribution. Following that, the RUL is predicted according to the latter degradation classes. The results will be validated using the C-MAPSS dataset of the aero-engine [2].

The paper is organised as follows: An overview of the related work as well as of the research gap is given in Section II. Section III provides the problem formulation, a theoretical background of the variational autoencoder (VAE) and the description of the proposed RUL estimation method. In Section IV, the analysis of the results is given. Finally, a conclusion is given in Section V.

II. Related work

This section analyzes recent research by focusing on four main challenges: architecture selection, dimensionality reduction and visual explanation techniques, attention mechanism, and model's hyperparameters optimization that can improve prognostics performance.

A. Data-driven methods for RUL estimation

The industrial dataset is possible to gather due to industrial IoT, which has promoted opportunities for industry and academia to leverage advanced data-driven techniques. Indeed, NASA's C-MAPSS turbofans time-to-failure data set have been extensively analyzed with RUL estimation as a primary focus. Many previous data-driven methods to machine status monitoring and RUL estimation, two main methods, including

I. Remadna and L.S. Terrissa are with LINFI Laboratory, Computer Science Department, University of Biskra, Biskra 07000, Algeria (e-mail: ikram.remadna@univ-biskra.dz, terrissa@univ-biskra.dz).

Z. Al Masry and N. Zerhouni are FEMTO-ST institute, Univ. Bourgogne Franche-Comté, CNRS, ENSMM, Besançon, France. (e-mail: zeina.al.masry@ens2m.fr, zerhouni@ens2m.fr)

conventional Machine Learning (ML) and Deep Learning (DL), have been applied and seen great success recently [3], [4]. Some studies [5], [6], [7], [8], [9] have used the promising Artificial Neural Network (ANN) approach to predict the RUL of various machines such as bearings, milling cutters, engines and drill pipe. In order to keep the optimum set of features, [10] applied Principal Component Analysis (PCA) as the first phase with ANN as the second phase to predict the RUL of the roller ball bearings. In [11], authors also have proposed a method to predict RUL based on neuro-fuzzy. However, this conventional ML cannot address sequential data; besides, the biggest limitation of the PCA approach lies in its linear projection.

Recently, burgeoning DL approaches have been widely applied in various research for prognostic and diagnostic tasks, known for their ability to process highly non-linear and varied data in their raw form without any human intervention. Within the deep learning architecture, the recurrent neural networks (RNNs) can mainly handle temporal data analysis that prompted researchers to applied it for the industrial PHM process. Some researchers [12], [13], [14] proposed an RNN-based methods toward the prognostic issue. However, RNNs had the vanishing gradient or exploding problem arising in long sequence input, which cannot keep the previous information, except only the latest one. To handle this issue, Long-Short Term Memory (LSTM) is the upgraded variant of RNN in which different gating mechanisms are proposed. [15] proposed an LSTM to determine the fault location and estimate the RUL of the aero engine. More recently, several works [16], [17], [18], [19], [20], [21] have suggested LSTM-based approaches for RUL estimation, showing the efficacy of performing LSTM over RNN. As an improvement, another variant of LSTM was used by [22] is Bi-directional LSTM (BLSTM) that can learn the bi-directional temporal dependencies from sensor data for Aircraft Engine RUL estimation. Thereby, it can capture long-range information in both future (forward) and past (backward) contexts of the input sequence simultaneously. Moreover, a new BLSTM model was presented by [23] for identifying the system degradation performance and subsequently predicting RUL. However, recurrent networks increase computational burdens.

Although Convolutional Neural Network (CNN) are one of the most dominant methods for image processing [24], [25], [26], [27], [28] CNNs have also been explored for RUL prediction by [29], [30] on the multi-channel time series. CNN architectures are designed to extract features through weight sharing filters and showed a noticeable improvement in prediction accuracy. Hybrid deep neural network models have also been reported in the literature [31], [32], [33] to leverage the power of different DL methods, which integrate CNN and LSTM models simultaneously to extract temporal and spatial features.

B. Dimensionality Reduction and Visual Explanation techniques

Massive and large-dimensional data often contain uninformative or redundant features, making data analysis difficult and increasing the processing time. Besides, reduced data visualisation is crucial to understand better how data is distributed, interpret and analyze classifier performance. Some classical data visualisation and dimensionality reduction methods such as PCA , Isometric Mapping (ISOMAP), and T-distributed Stochastic Neighbor Embedding (T-SNE) have been used and reported in academic research [34], [35], [36]. Additionally, several studies have shown the advantage of utilizing Auto-Encoder (AE) to decrease the data dimension and automatically extract the performance degradation features from multiple sensors, suitable with enhancing predictive accuracy and reducing the model's complexity. In [37], the authors proposed a new hybrid model integrating the advantages of AE and BLSTM to enhance the RUL's prediction accuracy. A similar study was suggested by [38] for RUL estimation based on a stacked Sparse Auto-Encoder (SAE) and Logistic Regression (LR).

Most of the generative applications deal with image processing as in [39] where a VAE was also trained to generate face images with much clearer and more natural noses, eyes, teeth, hair textures as well as reasonable backgrounds. Due to the ML algorithms black-box nature that imposes industry unwillingness to adopt it, more recently, in nonlinear processes monitoring, the deep method has been successfully applied to address both the curse of dimensionality and the scarcity of interpretability and transparency; by projecting the high-dimensional process data into a lower-dimensional space ([40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]). This latter is argued upon and supported by [53], that the ability to map a lower-dimensional space could increase a model's generalization capabilities. In [40], [51], [52], the authors propose a Variational Autoencoder (VAE) architecture as a 2D-Visualisation Tool of latent space to understand how data is distributed. This latter can help to get a better idea of how the model interprets the data.

Recently, Attention Mechanism (AM) has been widely applied in neural network architectures. It has been proved successful in natural language processing for machine translation tasks ([54], [55]), fault detection [56], RUL estimation task [57], [58], [59], and various computer vision tasks such as facial expression recognition [60], fruit classification [61]. Attention mechanism can make the neural network allocate more attention to useful features. [57] used a soft attention mechanism to provide visualisation of the learned attention weights at each RUL prediction step in order to gain its interpretability besides retaining the predictive power of LSTM networks. [56] employed a convolutional autoencoder with AM to enhance the local features of samples.

C. Model's Hyperparameters optimization

Deep learning models are full of hyperparameters in terms of architecture and training parameters (such as the number or type of layers and the learning rate). Their optimization by most of the reviewed papers are based on a trial-and-error approach [29], [15], [16], [17], [18], [22], [23], [30], [31], [32], [51], [33]. However, this approach can be time-consuming and error-prone due to a lack of understanding

of the impacts of parameters. To overcome these challenges, Automatic Hyperparameters Selection (AHPS) have been proposed, such as grid search [62], Random search [63], and Bayesian optimization [64], [65]. They typically carry out the research by discretizing the hyperparameter space. The grid search is the most applied strategy that tests all possible combinations (exhaustive searching) [37], [38], [57], [20]. Although this approach can theoretically obtain the optimal global parameters, it is extremely computationally expensive and suffers from the curse of dimensionality. The reason is that the number of combinations grows exponentially with the number of hyperparameters. Compared with grid search, random search eliminates the need for an exhaustive search of all possible combinations by picking them randomly. Bayesian optimization is an efficient hyperparameter tuning method and is widely used for the complex DNN methods ([66], [67], [68]). Its principle is to pick parameter combinations in a well-thought-out way, based on a probabilistic model. This probabilistic model uses previous evaluations to obtain the posterior predictive distribution by using the Bayesian formula. Therefore, we aim to apply Bayesian Optimization (BO) based on Gaussian Process (GP) to reduce the time spent on hyperparameter tuning, which disregards certain areas of the parameter space that are unlikely to yield the best results.

D. Research Gaps and Contributions

As summarized in Table VIII (See appendix), many contributions are proposed for the RUL prediction of turbofan engines using various DL architectures. Nevertheless, the application of DL techniques in the context of prognostics is still challenging. Inspired by these previous studies, the authors focus on six main points: architecture selection, Dimensionality Reduction, Visual Explanation techniques, Attention Mechanism, Model's Hyperparameters optimization, Ensemble learning method that can improve prognostics performance. Through this analysis, the main contribution of this work is a new methodology that combines the latter points and which can be summarized as follows:

- ACVAE integrates convolutional calculation with autoencoder to effectively extract spatial features.
- Attention mechanism is embedded to make the network pay attention to the useful features.
- ACVAE is more aimed at improving dimensionality reduction capability and achieving better spatial distribution and overall visualization.
- The approach aims to predict the probability that the machine will fail within different time windows (three degradation classes).
- The task of defining the three degradation classes takes into consideration the opinion of an expert.
- Ensemble learning is applied using the voting classifier to predict the class labels by averaging the class probabilities in order to reduce the conflict zone.
- AHPS is used to pick out the best configuration of hyperparameters to our hybrid architecture.

III. Methodology

A. Problem Formulation

There are N machines with the same type, such as the turbofan engine. Each engine consists of $maxT_i, i \in \mathbb{N}$ run to machine end of life (cycles) gathered by various sensors. The whole data can be formulated as

$$Dataset = \{(X^i, Y^i)\},\ i = 1, 2, 3, ..., N \tag{1}$$

where X^i refers to the engine-collected sensor measurements matrix and Y^i refers to the equipment operation cycles, as presented in Eq. (2) and Eq. (3), respectively.

$$X^i = [x_1, x_2, x_t, ..., x_{maxT_i}] \in R^{m \times maxT_i} \tag{2}$$

$$Y^i = [y_1, y_2, ..., y_{maxT_i}] \in R^{1 \times maxT_i} \tag{3}$$

where $maxT_i$ represents the total operation cycles of the i-th engine and $x_t = [x_t^1, x_t^2, ..., x_t^m] \in R^{m \times 1}$ represents an m-dimensional vector of sensor measurements at time t.
RUL can be estimated between the current time (y_t) after degradation detection and the failure time (T). The RUL can be summarized as follows [69]:

$$RUL_t^i = \{maxT_i - y_t^i\},\ t = 1, 2, ..., maxT_i. \tag{4}$$

To address the uncertainty in RUL, we propose to use $\varphi(.)$ for dimensionality reduction capabilities, which provides a better latent space distribution Z. Let X^i denote its input; sequential sensor measurement. Z^i is a latent representation generated by the encoder function $Z^i = f_\varphi(X^i)$ while $\hat{X}^i$ is an approximation or reconstruction of the real data X^i (see Eq. (5)).

$$Z^i, \hat{X}^i = \varphi(X^i). \tag{5}$$

The error between the X^i and reconstruction $\hat{X}^i$ is minimized as follows:

$$Minimize : \{\hat{X}^i, X^i\}. \tag{6}$$

Thus, the RUL values is divided into three RUL degradation classes Y_{class} according to two thresholds (α_1, α_2) that are defined based on the spatial distribution analyse with the expert.
To address the non-linearity function, ensemble ML method is proposed (θ) to RUL classes estimation. Let Z^i symbolise its input, and the observed Y_{class}^i symbolise its output. The predicted RUL classes is given by

$$\hat{Y}_{class}^i = \theta(Z^i, Y_{class}^i). \tag{7}$$

The prediction accuracy between the predicted RUL classes $\hat{Y}_{class}^i$ and the observed Y_{class} is maximized as follows:

$$Maximize : \{\hat{Y}_{class}^i, Y_{class}^i\}. \tag{8}$$

B. Variational autoencoders: theoretical background

An AE is an unsupervised Neural Network (NN) trained to reconstruct an input vector $X \in R^m$ where the dimension of X is denoted by $m \in N$ ([70], [71], [72]). As shown in Figure 10 (See appendix), the AE is comprised of two main parts: an encoder and a decoder, both of which are multilayered NNs parameterized with two weights vectors

ϕ and θ. The first one encodes the input data X via the encoder function $z = f_\phi(X)$ into a latent representation z, whereas the second one decodes this latent representation onto $\hat{X} = h_\theta(z)$ which is an approximation or reconstruction of the real data X. In AE, the input and output layers utilize the same number of units, whereas the latent space utilizes fewer. The AEs are typically employed for data compression (i.e., feature extraction/reduction), noise removal and pre-trained parameters for a complex network.

A VAE has the same functions as the AE in the sense that it is composed of an encoder and decoder (See Figure 10 in the appendix). VAE becomes a popular generative method by merging Bayesian inference and the efficiency of the NNs to obtain a nonlinear low-dimensional latent space ([48], [73], [41], [43]). The Bayesian inference is obtained by an additional layer applied for sampling the latent vector z with a prior specified distribution $p(z)$, usually assumed to be a standard Gaussian $N(0, I)$, where I is the identity matrix. Standard Gaussian is not the only distribution used for latent variables in VAEs, but the choice depends on the type of data we are modelling. Such as the multivariate Gaussian distribution is used in the case of real-valued data and the Bernoulli distribution is applied in the case of binary data [74]. Each element z_i of the latent layer Z is obtained as follow:

$$z_i = \mu_i + \sigma_i.\epsilon \tag{9}$$

where μ_i and σ_i are the i^{th} components of the mean μ and standard deviation σ vectors, ϵ is a random variable following a standard Normal distribution ($\epsilon \sim N(0, 1)$).

Unlike the AE which generates the latent vector z, the VAE generates vector of means μ_i and standard deviations σ_i. This allows to have more continuity in the latent space than the original AE. The VAE loss function given by the Equation 10 has two terms. The first term L_{rec} is the reconstruction loss function (Equation 11). Usually the negative expected log-likelihood (e.g., the cross-entropy function) is used ([39], [41], [45], [75], [73]), but the mean squared error can also be used [43]. The second term L_{KL} (Eq. (12)) corresponds to the Kullback-Liebler (KL) divergence loss term (similarity loss) that forces the generation of a latent vector with the specified Normal distribution ([74],[76]). The KL divergence is a theoretical measure of proximity between two densities q and p and it is noted by KL(q $\parallel$ p). The dissimilarities between these densities are asymmetric ($KL(q \parallel p) \neq KL(p \parallel q)$), non-negative and are minimized when $q(x) = p(x)\ \forall x$ [77]. Thus, the KL divergence term measures how close is the conditional distribution density q_ϕ(z | x) of the encoded latent vectors from the desired Normal distribution p(z). The value of KL is zero when two probability distributions are the same, which forces the encoder of VAE to learn the latent variables that follow a multivariate normal distribution over a k-dimensional latent space.

$$L = L_{rec} + L_{KL} \tag{10}$$

where

$$L_{rec} = -E_{q_\phi(z|x)}(log(p_\theta(x|z))), \tag{11}$$

$$L_{KL} = KL(q_\phi(z|x) \parallel p(z)) \tag{12}$$

with $p_\theta(x|z)$ is the conditional distribution density of the decoded latent vectors.

When the VAE is trained, each function (i.e., the encoder and the decoder) can be used separately, either to reduce the space dimension by encoding the input data, or to generate synthetic samples by decoding new variables from the latent space (Figure 10 is available in appendix).

C. Remaining useful life estimation based on CVAE with attention mechanism

The RUL classes estimation methodology is shown in Figure 1. VAE approach can generate new data (through continuity) as well as it has been demonstrated as a promising tool for dimensionality reduction in the context of machinery fault diagnosis [48], [78]. However, VAE still has to be fully explored for both fault diagnosis and prognosis. In this paper, we propose ACVAE which is a VAE based architecture for predicting RUL classes. It is composed of an encoder and a decoder, which are two symmetrical and reversed structures. Both the encoder and decoder have two convolutional layers. We utilized the same padding and a 6×1 kernel for convolutional layers in the encoder. The stride was 1×1 for the first convolutional layer and 2×2 for the second. The convolutional layer's output is expressed as follows:

$$C_i = f(\sum X \odot w_i + b_i) \tag{13}$$

where f is the activation function, $\odot$ is the convolution operation, w_i and b_i represents the weight parameter and bias of i^{th} convolutional kernel, respectively. The final convolution will generate an output $H^c = \{h_1, h_2, ..., h_{\frac{W_L}{2}}\}$ where $H^c \in R^{\frac{m \times W_L}{2}}$. Given input data of sequence length W_L with m number of features (sensor variables).
Moreover, the attention layer is also embedded in the encoding part, which is used to dynamically increase the weights of the useful features to make the network pay attention to these vital features [55], [79]. The attention computational is given as follows:
The attention weights:

$$\alpha_i = softmax(W_a \cdot h_i) \tag{14}$$

The context vector:

$$c_i = \sum_i \alpha_i \cdot h_i \tag{15}$$

The attention vector:

$$a_i = tanh(W_c[c_i; h_i]) \tag{16}$$

where h_i is the features extracted by the encoder.

The latent two-dimensional space is represented by two 2D-layers for the encoder: the mean and the standard deviation layers (i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$), and one 2D-sampling layer ($\mathbf{Z}$) for the decoder.

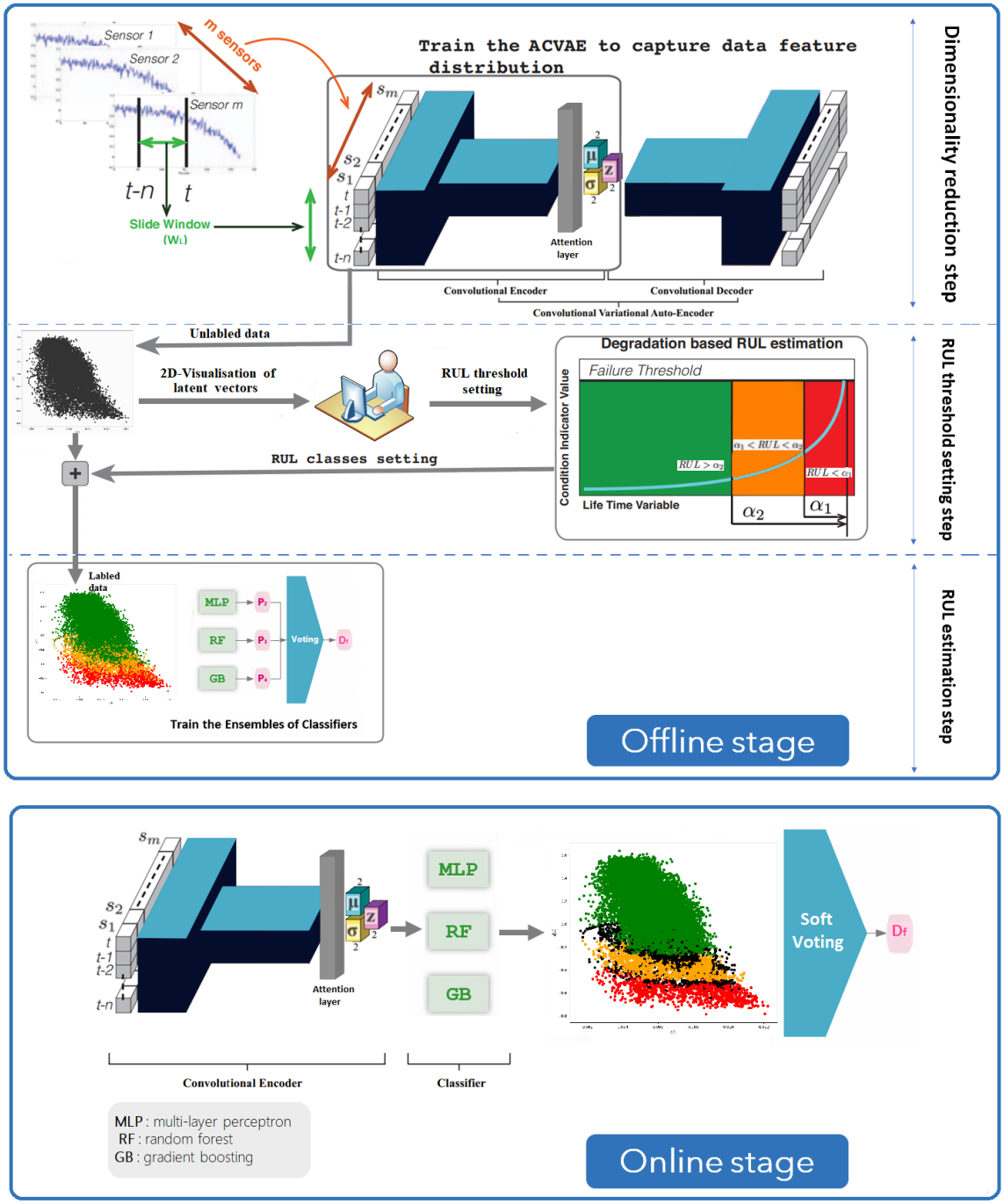

Fig. 1. A framework of RUL estimation using the soft voting classifier with MLP (Multi-layer perceptron), RF (Random forest), and GB (Gradient Boosting). D_f is the final class label.

The first step was to train the whole ACVAE architecture for the reconstruction of the input vector using the deconvolutional operation (decoder part), as shown in Eq. (17).

$$D_i = f(\sum Z \otimes \bar{w}_i + \bar{b}_i) \tag{17}$$

$\otimes$ is the deconvolution operation, $\bar{w}_i$ and $\bar{b}_i$ represents the weight parameter and bias of i^{th} deconvolutional kernel, respectively.

Training the ACVAE does not need the label information of the input data. The whole ACVAE is trained to attain a coupled optimization of both the quality of the reconstruction and the quality of disentanglement. The training loss of ACVAE is defined as the sum of the reconstruction loss and the similarity loss (See Eq. (10)). The used reconstruction loss is the Mean Squared Error (MSE), which measures how close the decoder output is to the original input, as expressed in Eq. (18).

$$L_{rec} = 1/N \sum_{i=1}^{N} (X_i - X_i')^2 \tag{18}$$

The similarity loss is the KL divergence between the latent space distribution and the standard Gaussian (zero mean and unit variance), which regularizes the distribution of the latent space (given in Eq. (19)).

$$L_{KL} = -1/2 \sum_{i=1}^{N} \sum_{j=1}^{k} (1 + log(\sigma_{ij}^2) - \sigma_{ij}^2 - \mu_{ij}^2) \tag{19}$$

where N is the number of samples, X_i denotes the real data while X_i' is its reconstruction, and k is the size of the latent vectors.

In this paper, we aim to predict the probability that the machine will fail within different time windows. To do this, the expert faced a challenging problem: how to select the most appropriate thresholds α_1 and α_2 that are used to label the data (three degradation classes)? Three degradation classes are then defined according to two thresholds (α_1, α_2) as follows:

- Degradation class 0 (Deg 0): $RUL > \alpha_2$,
- Degradation class 1 (Deg 1): $\alpha_1 < RUL \leq \alpha_2$,
- Degradation class 2 (Deg 2): $RUL \leq \alpha_1$.

When the training process of the ACVAE is successfully done, the encoder part is then used as a 2D-Visualisation tool by a human expert in order to analyse the spatial distribution of the data set that has been separated into three distinct degradation classes. The expert has made assumptions about picking out the values of thresholds α_1 and α_2, experimenting with different thresholds values ($\alpha_1 = \{10, 20\}, \alpha_2 = \{20, 30, 70, 90\}$). Each couple ($\alpha_1$, α_2) will generate a particular overlapping situation between the degradation classes, which is easily visualised and examined by the expert in the 2D-latent space. By analysing the spatial distribution of different couples (α_1, α_2), the expert tries to choose the appropriate threshold setting that minimizes the overlapping region between the degradation classes. Indeed, the appropriate thresholds gives thinner conflict areas (the boundaries) between classes, thus less instances (Z_i points) within boundaries which gives less conflicting decisions for the classification.

The second step is to train the ensemble of classifiers for the RUL classes estimation. The encoder parameters obtained by the previous step are frozen during the classifier training step. Algorithm 1 represents the active learning classifier that is used to construct a high-performance classifier by starting learning with a small training set. Via the incremental learning process, the misclassified points (uncertainty points) from $X_{validation}^{2D}$ are actively added into the training set X_{train}^{2D} based on a threshold of probability (As shown in [80], [81], [82], [83]).

Finally, in the operating stage, the convolutional encoder is used jointly with soft voting based ensemble classifiers to estimate the degradation RUL classes (as shown by Figure 1).

IV. Results analysis

A. C-MAPSS Dataset

We begin by describing the used dataset for our experimental study. In this research, the C-MAPSS dataset from the Nasa repository is used, an abbreviation of (Commercial Modular Aero-Propulsion System Simulation). It concerns the simulation of turbofan engine degradation datasets using a model-based simulation program. This dataset [84] represents the deterioration of the four fleets of aircraft gas turbine engines' life cycle. The four fleets are organized into four sub-datasets within 26 criteria and many data rows (to better understand the C-MAPSS dataset, see [2]). In this work, we only used the failure set FD001 to validate our hybrid method. As seen in Table I, the subset is separated into a training set and testing set, each one containing 100 trajectories. This FD001 consists of multiple multivariate time-series signals to simulate the deterioration of turbofan engines that experienced one operation condition and were characterized by failure deterioration owing to a failed High-Pressure Compressor (HPC).

TABLE I
CHARACTERISTICS OF C-MAPSS CHALLENGE DATASET.

FD001 Dataset	values
Engines of the training set	100
Engines of the test set	100
Fault conditions	1
Operating conditions	1
Maximum life span	362
Minimum life span	31

B. Evaluation metrics

To fairly evaluate the proposed model's performance on the test dataset, a range of various performance metrics is adopted (Eq. (20) - (25)). These metrics involve accuracy, precision, sensitivity, specificity, F-score, G-mean, Receiver Operation Characteristic (ROC), and Area Under Curve (AUC) [85]. The below formulae metrics could be assessed with $|TP|$ the number of the true positive, $|TN|$ the number of the true negative, $|FP|$ the number of the false positive (i.e. a false alarm), and $|FN|$ the number of the false negative (i.e. a missed alarm).

- **Accuracy :** refers to the ratio of the total correct predictions. It is formulated as:

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |FP| + |FN| + |TN|} \tag{20}$$

- **Precision :** expresses the ratio of correctly predicted positive instances. Formally, it can be expressed as:

$$Precision_{macro} = \frac{1}{n} \sum_{i=1}^{n} \frac{|TP_i|}{|TP_i| + |FP_i|} \tag{21}$$

- **Recall or Sensitivity :** measures how much a classifier can recognize positive instances, which are correctly

Algorithm 1 Train the classifier

Input: Sliding window training data $X_{train}, Y_{train} = \{(x^i, y^i)\},\ i = 1, 2, 3, ..., N$
Output: Trained classifier β

Train the ACVAE on X_{train}
X^{2D}_{train} = $f_\varphi(X_{train})$ ▷ Use the encoder to extract the 2D-latent space
X^{2D}_{train}, $X^{2D}_{validation}, Y_{train}$, $Y_{validation}$= split($X^{2D}_{train}, Y_{train}$)
Randomly Initialize β
for i=1 to Q **do** ▷ Number of query
    Train the classifier β on X^{2D}_{train}
    Validate the classifier with $X^{2D}_{validation}$
    Sort misclassified examples by error score $X^{2D}_{misclas}$
    X^{2D}_{train} = $X^{2D}_{train} \cup X^{2D}_{misclas}$ ▷ Add the misclassified examples to train
    $X^{2D}_{validation}$= $X^{2D}_{validation}$ - $X^{2D}_{misclas}$ ▷ Remove misclassified from validation set
end for
return The model β

identified by the classifier. It is computed using the following equation:

$$Sensitivity_{macro} = \frac{1}{n}\sum_{i=1}^{n}\frac{|TP_i|}{|TP_i| + |FN_i|} \tag{22}$$

- **Specificity :** calculates how much a classifier can recognize negative instances, which are correctly identified by the classifier. It is given by the equation:

$$Specificity_{macro} = \frac{1}{n}\sum_{i=1}^{n}\frac{|TN_i|}{|TN_i| + |FP_i|} \tag{23}$$

- **F1-score :** to maximize both precision and recall, the F1-score metric is the harmonic mean of the precision and recall. This combination reaches its highest possible value at 1,indicating perfect precision and recall, and its lowest possible value at 0, if either the precision or the recall is zero. It can be formulated as:

$$F1 - score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} \tag{24}$$

- **G-Mean** measures the trade-off between sensitivity (true positive rate) and specificity (true negative rate) by the following:

$$G - Mean = \sqrt{Sensitivity \times Specificity} \tag{25}$$

Only when both sensitivity and specificity are high can the G-mean attain its maximum, which indicates a better classifier.

- **ROC curve :** is a graphical plot showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate.
- **AUC :** is the fraction of total area that lies under the ROC curve. AUC provides a single value for assessing the performance of the classifier and an examination of the classifier's stability and consistency.

C. Data pre-processing

Before applying the proposed model, it is extremely necessary to prepare heterogeneous data adequately. The specific steps are described as follows:

1) Feature selection:
In the FD001 dataset, there are three operating indicators, and 21 distinct aircraft engine sensors plotted on a histogram to observe the variations throughout the whole lifecycle of engines. The sensors 1, 5, 6, 10, 16, 18, and 19 exhibit constant values throughout the engine, which cannot provide relevant degradation information to accomplish the task and only will increase the training time of neural networks. Therefore, 14 out of 21 sensors were selected, whose indices are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21. Besides, the three operational settings are removed because these datasets are exposed to a single operating condition.

2) Data Normalization:
The sensor measurements are with a varied range of values. Therefore, there are several normalization methods for the differentiation issue that guarantee the same range scale of all sensor measurements [86]. This paper uses the Min-Max normalization given in Eq. (26) to map the selected features within the range of [0,1].

$$x'_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}. \tag{26}$$

3) Sliding Window:
The multivariate time-series sensor signals can provide more degradation information, which leads to an accurate prediction. Therefore, the Sliding Window (SW) method is used to segment the data samples into a sliding time window along the engine life span (as shown in Figure 2).

The extracted data by SW each time is a 2D matrix $W_L \times W_f$, W_L as a length of sliding time window, and W_f as a number of the selected prognosis feature. Moreover, the SW is moved with only one data point. Thus, the number of sliding time windows generated from data is $\sum_{i=1}^{n} maxT_i - W_L$, where $maxT_i$ is the engine lifespan and n number of engines.

4) Data rebalancing: :
Class Imbalance problem is encountered when one of the classes is underrepresented over others. It is challenging to train classifiers on imbalanced data, as they become biased towards a set of classes. A widely implemented approach for handling imbalance is resampling, either using various undersampling (removing some of the majority class data points)

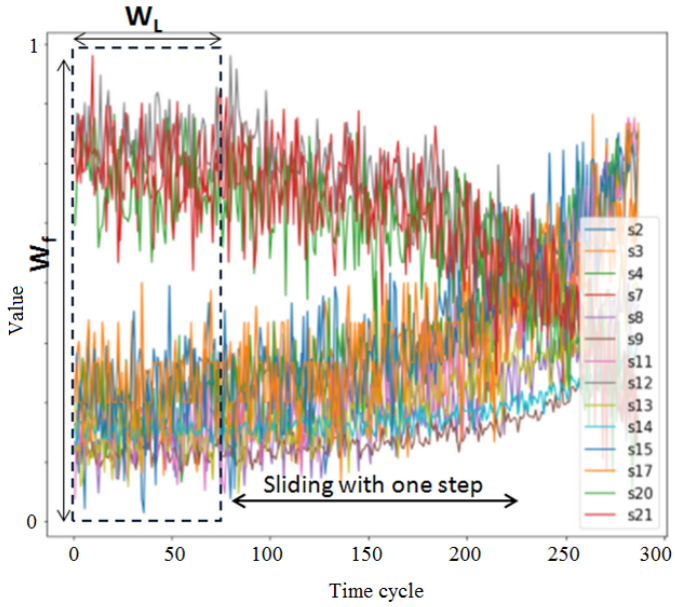

Fig. 2. One training sample using sliding window method.

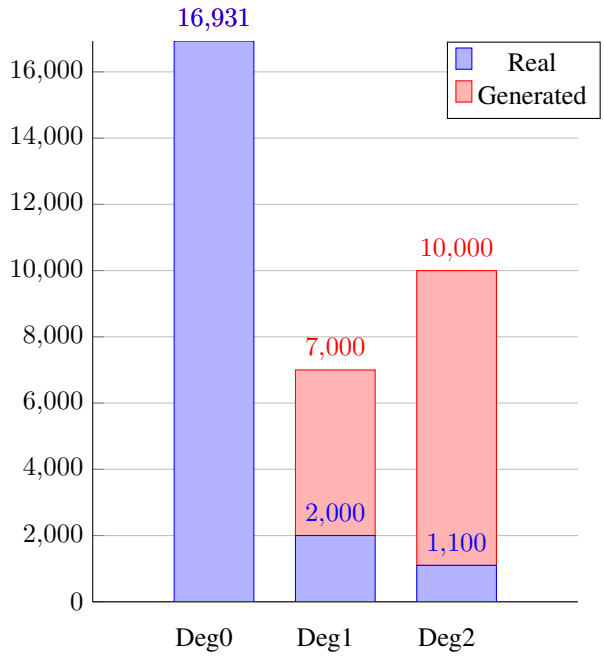

Fig. 3. Data rebalancing using a SMOTE-based KNN on training set FD001.

or over-sampling (adding more of minority class data points) algorithms. The under-sampling can cause wastage of important information. On the other hand, the Synthetic Minority Oversampling TEchnique (SMOTE) is one of the dominant oversampling methods in literature [87], [88]. Consequently, in this work, the SMOTE-based K-Nearest Neighbors (KNN) method is implemented to handle the class imbalance problem.

Figure 3 depicts the class distribution bar charts, where the blue bar refers to real data showing that the data used was highly imbalanced. The red bar indicates the generated synthetic instances in minority classes that follow the original distribution utilizing the SMOTE-KNN algorithm. Instead of excessively increasing the number of synthetic instances in the minority classes (equal to majority class), we experimented with different oversampling thresholds and picked the best ones (see Figure 3).

D. Results

1) Data Pre-processing Parameters Analysis:

Several sliding windows sizes W_L were tested and evaluated according to different degradation thresholds α_1 and α_2, and they are the most sensitive pre-processing parameters. Table II reflects the performance of our proposed approach obtained for each test through evaluation measures, with different W_L and different thresholds (α_1, α_2). In our experiments, we considered values of W_L in $\{6, 16, 26\}$ taking into account the minimum engine life span available in the test dataset. The values of thresholds (α_1, α_2) are in {(10,20), (10,30), (10,70), (10,90), (20,30), (20,70), (20,90)}.

Results in Table II show that the proposed method with a sliding window equal to six timesteps ($W_L = 6$) yields the best performing scores overall. We should point out that if the thresholds are α_1=10 and α_2=20, the classifier maximizes the recall but with low precision, in the $W_L = 16$ compared with $W_L = 6$. The model with $W_L = 16$ gives us a 58%, 77%, 97%, 61%, and 97% for precision, recall, specificity, F1-Score, and accuracy accordingly. On the other hand, the model has achieved 71%, 73%, 96%, 63%, and 98% for precision, recall, specificity, F1-Score, and accuracy, respectively with $W_L = 6$. The high recall and low precision indicate that most of the faults are correctly recognized, but there are a lot of false alarms (as in $W_L = 16$). Contrarily, a low recall and high precision appear that some faults are missed, but those real faults are flagged, and there are no false alarms (as in $W_L = 6$). In addition, The classifier with a high specificity shows fewer false alarms (as in $W_L = 16$).

The critical goal is to get the best classifier that maximizes both precision and recall (Best F1-score) with a modest specificity. It is also observed that the result obtained by the sliding window W_L=6 showed up a higher accuracy with these thresholds (α_1 = 10, α_2 = 20) and (α_1 = 10, α_2 = 30). On the other hand, based on the F1-score, the better estimation is obtained by the sliding window W_L=6 with α_1 = 10 and α_2 = 30. Note that the sliding window W_L=26 obtains the worst result with the thresholds α_1 = 10 and α_2 = 30.

The tests confusion matrices are depicted in Figure 4 where the ordinate shows the reference label while the abscissa represents the predicted one. We mention that classifier fails to predict the last class (Deg 2) when the thresholds of α_1 and α_2 are too near. Even with the SW size variation, the predicted degradation 2 is almost nonexistent for $\alpha_1 = 10$ and $\alpha_2 = 20$. Furthermore, the performance of the RUL prediction is improved with the increase of the gap between the two degradation thresholds. The good results are obtained by the slide window W_L = 6 with α_1 = 10, α_2 = 30 and W_L = 26 with α_1 = 10, α_2 = 70. For W_L = 26 with α_1 = 10, α_2 = 70, the proportion of the true positive classification

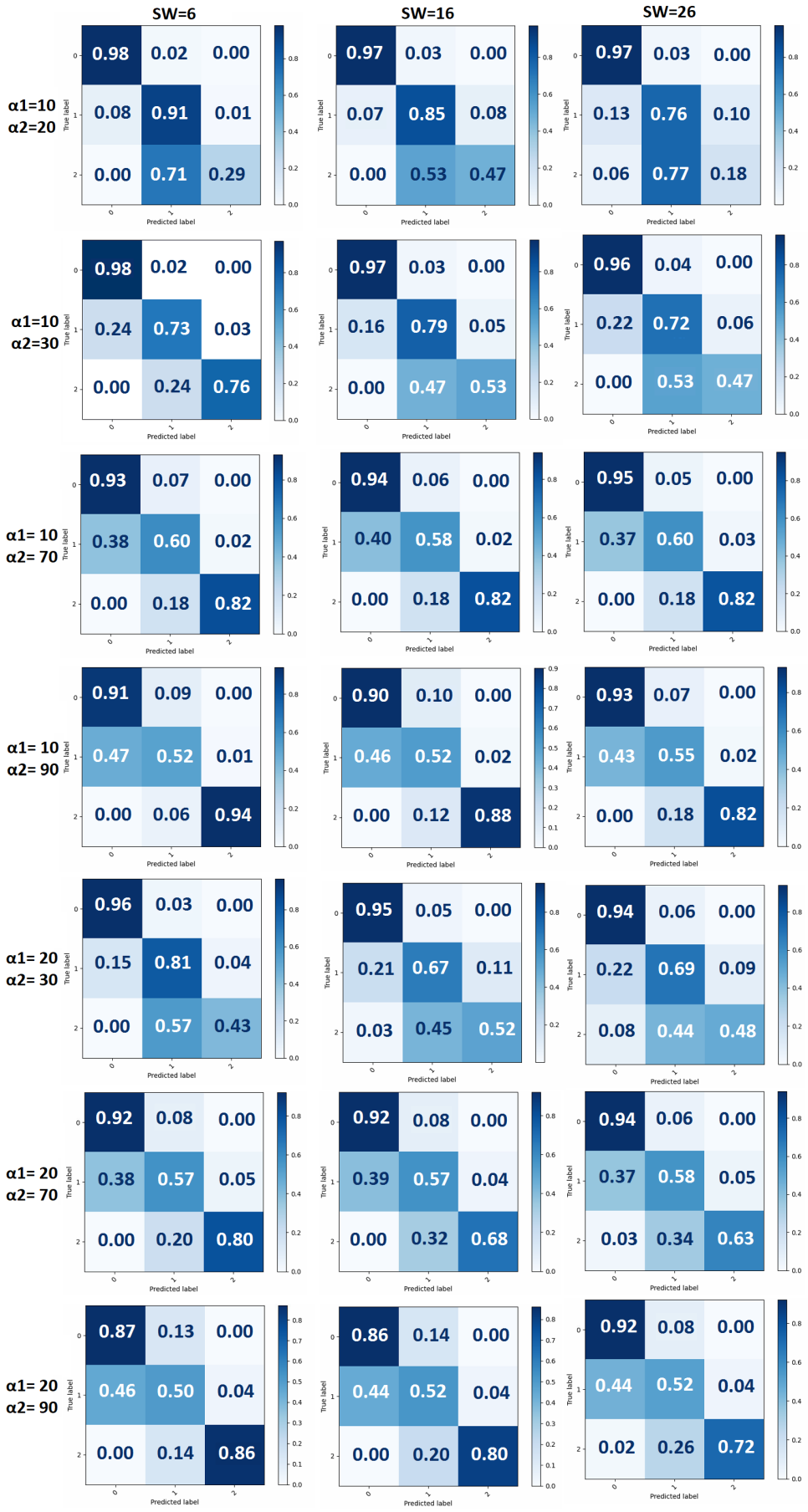

Fig. 4. Confusion matrices of test set FD001 vs. (W_L) and (α_1,α_2).

TABLE II
RESULTS OF OUR APPROACH ON TEST SET FD001 VS. (W_L) AND (α_1,α_2).

		Precision	Sensitivity	Specificity	F1-Score	AUC	Accuracy	G-mean
W_L=6	$\alpha_1 = 10$, $\alpha_2 = 20$	**0.71**	0.73	0.96	0.63	**0.99**	**0.98**	0.84
	$\alpha_1 = 10$, $\alpha_2 = 30$	**0.71**	**0.83**	0.91	**0.76**	**0.99**	**0.98**	**0.87**
	$\alpha_1 = 10$, $\alpha_2 = 70$	0.62	0.79	0.84	0.67	0.93	0.89	0.82
	$\alpha_1 = 10$, $\alpha_2 = 90$	0.61	0.79	0.81	0.66	0.88	0.83	0.80
	$\alpha_1 = 20$, $\alpha_2 = 30$	0.69	0.73	0.95	0.64	0.98	0.96	0.84
	$\alpha_1 = 20$, $\alpha_2 = 70$	0.67	0.77	0.84	0.71	0.92	0.87	0.81
	$\alpha_1 = 20$, $\alpha_2 = 90$	0.64	0.75	0.81	0.68	0.88	0.80	0.78
W_L=16	$\alpha_1 = 10$, $\alpha_2 = 20$	0.58	0.77	**0.97**	0.61	**0.99**	0.97	0.86
	$\alpha_1 = 10$, $\alpha_2 = 30$	0.60	0.76	0.93	0.66	0.98	0.96	0.84
	$\alpha_1 = 10$, $\alpha_2 = 70$	0.61	0.78	0.85	0.65	0.91	0.89	0.81
	$\alpha_1 = 10$,$\alpha_2 = 90$	0.57	0.77	0.8	0.61	0.87	0.82	0.79
	$\alpha_1 = 20$, $\alpha_2 = 30$	0.62	0.72	0.93	0.62	0.97	0.94	0.82
	$\alpha_1 = 20$, $\alpha_2 = 70$	0.67	0.72	0.85	0.69	0.91	0.87	0.78
	$\alpha_1 = 20$, $\alpha_2 = 90$	0.62	0.73	0.80	0.66	0.86	0.77	0.77
W_L=26	$\alpha_1 = 10$, $\alpha_2 = 20$	0.47	0.64	0.94	0.50	0.98	0.97	0.78
	$\alpha_1 = 10$,$\alpha_2 = 30$	0.55	0.72	0.91	0.61	0.97	0.96	0.81
	$\alpha_1 = 10$, $\alpha_2 = 70$	0.61	0.79	0.85	0.64	0.92	0.89	0.82
	$\alpha_1 = 10$, $\alpha_2 = 90$	0.59	0.76	0.82	0.61	0.89	0.83	0.8
	$\alpha_1 = 20$,$\alpha_2 = 30$	0.60	0.70	0.91	0.60	0.96	0.93	0.8
	$\alpha_1 = 20$, $\alpha_2 = 70$	0.68	0.71	0.85	0.70	0.90	0.88	0.78
	$\alpha_1 = 20$, $\alpha_2 = 90$	0.66	0.72	0.82	0.68	0.87	0.82	0.77

for the degradation classes 0, 1 and 2 are respectively 95%, 60% and 82%. On other hand, for W_L = 6 with $\alpha_1 = 10$, $\alpha_2 = 30$, the proportion of the true positive classification for the degradation classes 0, 1 and 2 are respectively 98%, 73% and 76%.

Based on the results, one remarks that our approach is entirely adapted for RUL estimation. To be more precise, all the false positive predictions belong to the less critical degradation class. The following performances can be observed in the confusion matrix generated by the sliding window W_L= 6 and (α_1 = 10, α_2 = 30):

- **Deg 0:** 98% are correctly classified as degradation 0, and 2% are incorrectly classified as degradation 1.
- **Deg 1:** 73% are correctly classified as degradation 1, and 24%, 3% are incorrectly classified as degradation 0, degradation 2, respectively.
- **Deg 2:** 76% are correctly classified as degradation 2, and 24% are incorrectly classified as degradation 1.

From a maintenance viewpoint, our approach could alert the critical degradation level, which signifies that the predicted RUL is lower than the actual one ($RUL_{predicted} \leq RUL_{True}$).

Table III gives the detail of the used ACVE architecture. The optimal ACVE parameters have been selected using AHPS based on Bayesian Optimization with a Gaussian Process model [89], which is handled by the Keras-tuner library [90], as described in Algorithm 2. The acquisition function used is Upper Confidence Bound (UCB), for more details can be found [91]. The range of the hyperparameter detailed in Table IV, with the selected by BO. These most sensitive hyperparameters are chosen due to their highest impact on the performance.

2) Visualisation of latent vectors and identification the conflict zone:

TABLE III
THE PROPOSED HYBRID DEEP CONVOLUTIONAL VARIATIONAL AUTO-ENCODER ARCHITECTURES WITH ATTENTION MECHANISM.

Layer	Type	Neurons	Kernels
	Encoder		
0	Input vector	W_L × 14 ×1	-
1	Convolution	W_L × 14 × 64	6 × 1
2	Convolution	3 × 7 × 128	6 × 1
3	Convolution	3 × 7 × 1	6 × 1
4	Reshape	3 × 7	-
5	Attention vector	128	-
6	Mean layer	2	-
7	Standard deviation layer	2	-
	Decoder		
0	Sampling layer	2	-
1	Deconvolution	3 × 7 × 128	6 × 1
2	Deconvolution	W_L × 14 × 64	6 × 1
3	Output vector	W_L × 14 × 1	-
	Ensemble learning		
0	Sampling layer	2	-
1	Gradient Boosting	Estimators=500	-
2	Random Forest	Estimators=32	-
3	Multi-layer Perceptron	10	-
4	Output	3	-

To quantitatively assess the proposed method's performance, we have compared the visualisation performance of the AC-VAE method with three state-of-the-art dimension reduction methods, including PCA, ISOMAP, and T-SNE. Figure 5 displays the 2D-space distribution of six different dimensionality reduction methods. As a recall, going from the green to the red colour point means a decrease in the health state of the engine machine (colours indicate the three degradation classes). In looking at Figure 5, it can be deduced that ACVAE seems to be able to cluster more effectively the dataset according to RUL degradation reasoning. It can also be argued that the ACVAE resulted in a more compacted spatial distribution compared

Algorithm 2 Hyperparameter Tuning using Bayesian Optimization with Gaussian Processes

Required: D : *Hyperparameters combination*, **u :** *Acquisition function*
Initial Settings: Randomly Initialize D
for $n = 1$ *to* T **do** ▷ T represents the maximum trial run.
Find x_t by minimizing u over GP :
$x_t = argmin_x\ u(x|D_{1:t-1})$
Evaluate the objective function y_t = f(x_t).
Augment the observation set $D = D \cup (x_t, y_t)$, update the posterior of function f.
end for
return Choosing the best hyperparameters combination

TABLE IV
DESCRIPTION OF THE HYPERPARAMETERS, THEIR RANGE AND THE SELECTED VALUES.

Name	Range	Selected
N° of convolutional layer	`Min=1, Max=3, Step=1`	2
N° of filters per layer	`Min=8, Max=256, Step=8`	(64,128)
Filter size	`Min=1, Max=16, Step=1`	6×1
Learning rate	`Min=1e-4, Max=0.5, Sampling=LOG`	0.001
Batch size	`Min=32, Max=1024, Step=32`	128
Activation function	`relu, tanh, sigmoid,softplus,softsign,selu,elu`	elu
Optimizer	`Adam,Adadelta,Adamax,SGD,RMSprop,Adagrad,Nadam,Ftrl`	RMSprop

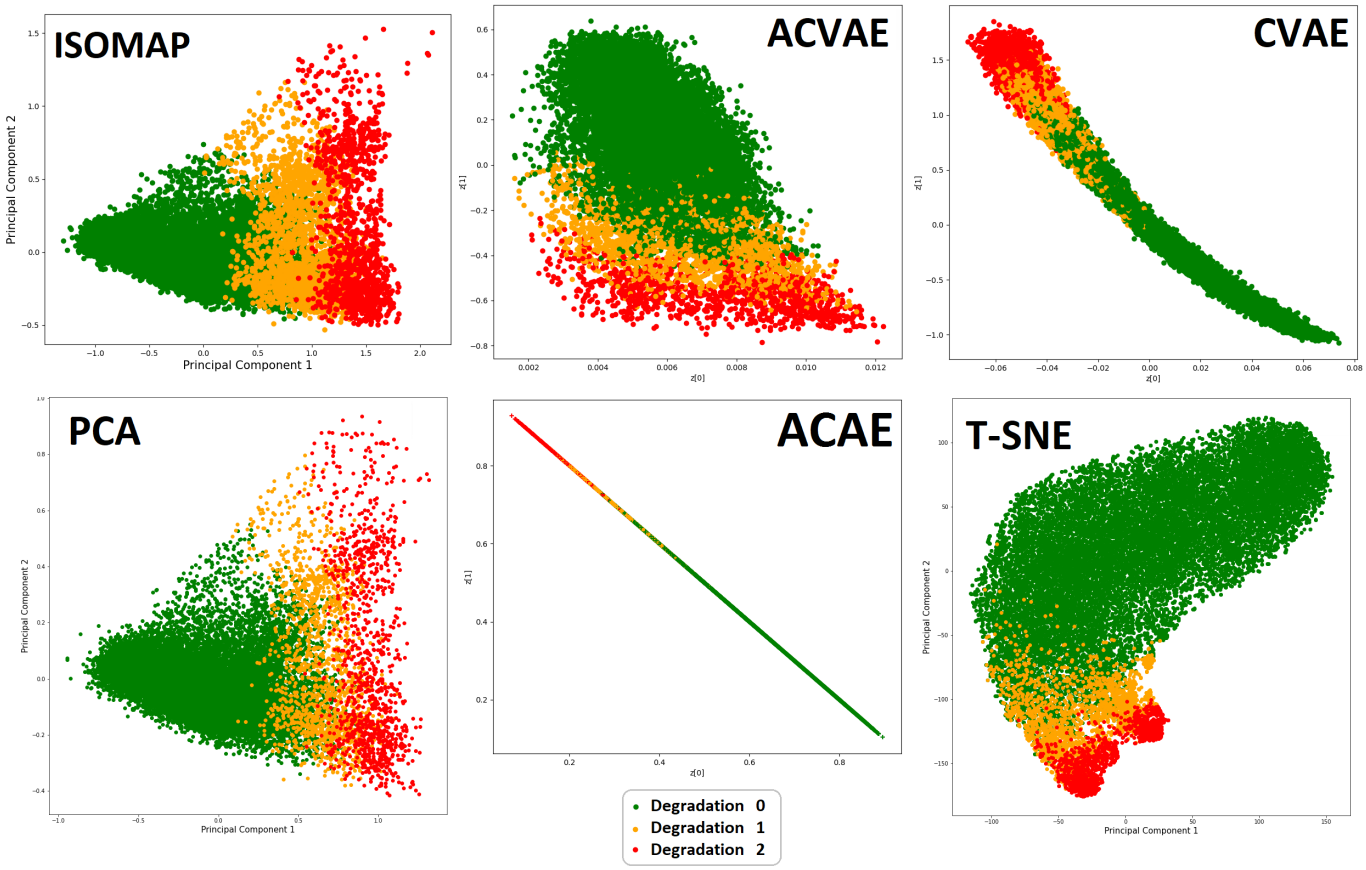

Fig. 5. The 2D-visualisation using five dimensionality reduction methods: ISOMAP, ACVAE, CVAE, PCA, ACAE and T-SNE.

to the covered areas by other methods. The horizontal and vertical axis extends from approximately 0.002 to 0.012, and -0.8 to 0.6, respectively.

Indeed, the best average accuracy was gained by ACVAE and ACAE around 98%, and the worst result obtained by the T-SNE method is approximately 78% of accuracy, as

TABLE V
COMPARISON OF THE PROPOSED ACVAE WITH OTHER METHODS: ISOMAP, PCA, TSNE, ACAE AND CVAE BASED ON TEST SET FD001.

		Precision	Sensitivity	Specificity	F1-Score	AUC	Accuracy	G-mean
CVAE With AM	Deg 0	0.99	0.98	0.77	0.99	0.98		
	Deg 1	0.55	0.73	0.984	0.62	0.98		
	Deg 2	0.59	0.76	1.00	0.67	1.00		
	Mean	**0.71**	**0.83**	**0.917**	**0.76**	**0.99**	**0.98**	**0.87**
CVAE Without AM	Deg 0	0.99	0.98	0.743	0.99	0.982		
	Deg 1	0.49	0.66	0.981	0.56	0.978		
	Deg 2	0.42	0.88	0.998	0.57	0.998		
	Mean	**0.63**	**0.84**	**0.907**	**0.70**	**0.986**	**0.97**	**0.87**
CAE With AM	Deg 0	0.99	0.99	0.617	0.99	0.978		
	Deg 1	0.60	0.55	0.99	0.57	0.975		
	Deg 2	0.40	0.71	0.998	0.51	0.999		
	Mean	**0.66**	**0.75**	**0.868**	**0.69**	**0.98**	**0.98**	**0.81**
ISOMAP	Deg 0	1.00	0.94	0.95	0.97	0.985		
	Deg 1	0.21	0.61	0.942	0.31	0.932		
	Deg 2	0.12	1.00	0.99	0.21	0.998		
	Mean	**0.44**	**0.85**	**0.960**	**0.50**	**0.971**	**0.93**	**0.90**
PCA	Deg 0	1.00	0.95	0.958	0.97	0.99		
	Deg 1	0.12	0.44	0.95	0.19	0.92		
	Deg 2	0.11	1.00	0.99	0.19	0.998		
	Mean	**0.41**	**0.80**	**0.966**	**0.45**	**0.969**	**0.94**	**0.87**
T-SNE	Deg 0	1.00	0.79	0.92	0.88	0.88		
	Deg 1	0.00	0.02	0.92	0.01	0.42		
	Deg 2	0.01	1.00	0.85	0.02	0.93		
	Mean	**0.34**	**0.60**	**0.896**	**0.30**	**0.743**	**0.78**	**0.73**

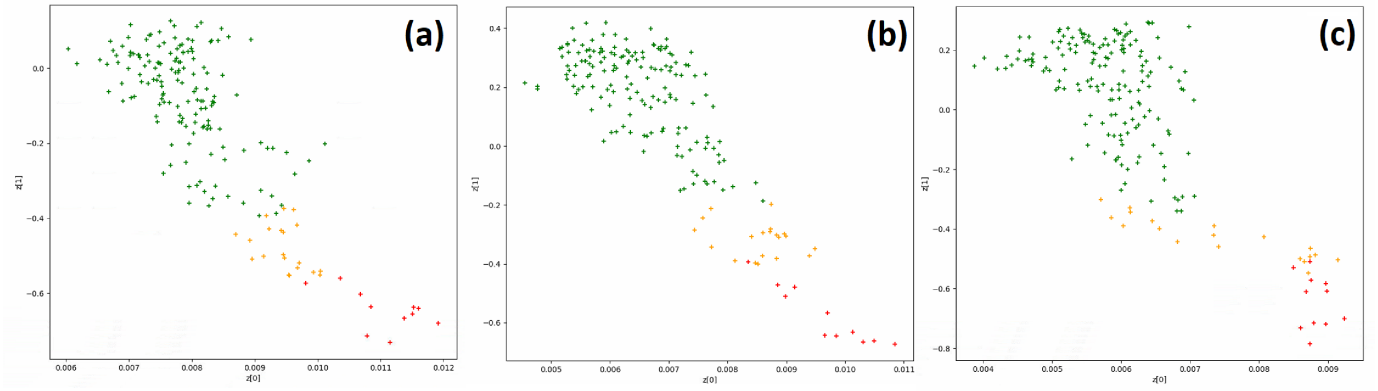

Fig. 6. The 2D-latent representation of three training engine units in FD001 using our proposed hybrid architecture: (a) engine 1, (b) engine 6, and (c) engine 100.

shown in Table V. The quantitative results showed that the precision of degradation classes 1 and 2 are almost non-existent with ISOMAP, PCA, and T-SNE. Besides, the T-SNE method shows the low precision of 0%, sensitivity of 2%, F1-Score of 1%, and AUC of 42% for degradation class 2, making degradation class 2 unpredictable. The performance of CVAE is slightly improved when the attention mechanism is added across all measures except the sensitivity, with 8%, 1%, 6%, 0.4%, 1%, for Precision, Specificity, F1-Score, AUC, and Accuracy, respectively. The CVAE with AM gave us the highest performance compared to the other dimensionality reduction methods.

Besides, we can observe that our proposed VAE-based architectures outperforms an attention convolutional autoencoder (AE-based architecture), as these results prove its effectiveness in extracting useful performance degradation features. It is also interesting to see the difference between the distribution and the clustering of degradation classes in both VAE-based architecture (ACVAE) and AE-based architecture (ACAE). Our approach seems to be able to map degradation features into a less disentangled latent space (as shown in Figure 5). These results give a hypothesis that the standard autoencoder would suffice in the case of dissimilar data classification. Therefore, the power of the VAE architecture comes with similar-looking data (degradation features) that usually overlap in some areas where AE fails to disentangle.

According to these results, we can visually perceive them by analyzing the clusters obtained by the worst and best method, concentrated in different areas. T-SNE distribution seems to cover more area with the horizontal and vertical axis extends from approximately -100 to 150, and -150 to 100, respectively.

The 2D-latent space representation of three training engine units is presented to visually appreciate the effect of our proposed ACVAE method on the spatial distribution (See Fig-

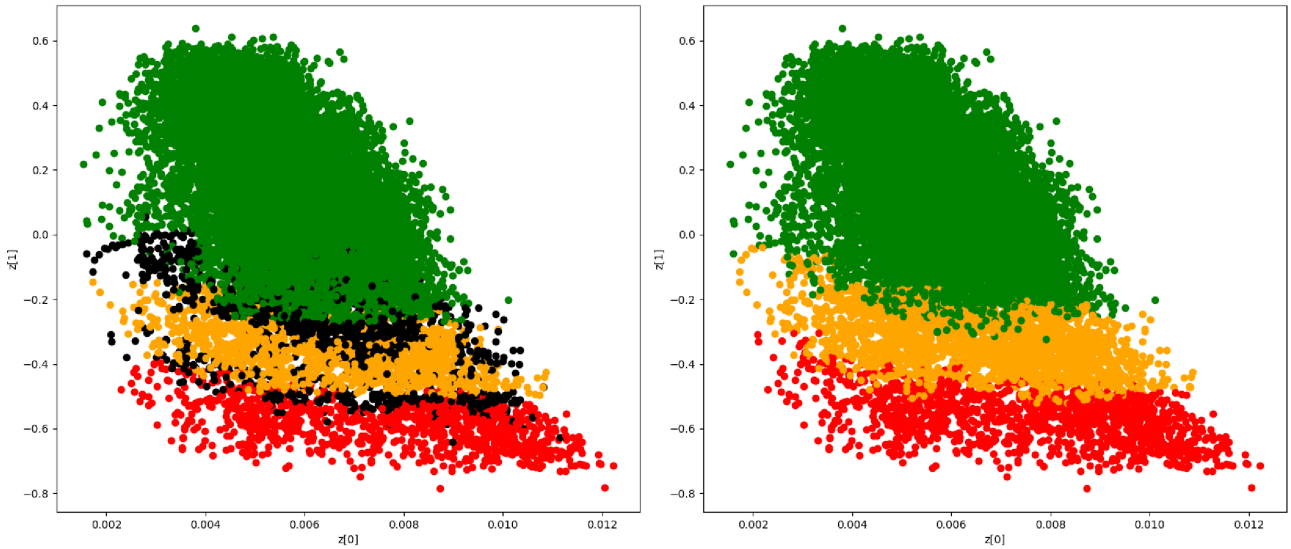

Fig. 7. Identification and Reduction of the conflict zone: (a) 2D-visualisation for the conflict zone where black points indicate the samples with uncertain classification (b) 2D-visualisation of the soft voting classifier results.

Algorithm 3 Identification and reduction of the Conflict Zone

Input: Sliding window training data $X_{train} = \{(x^i, y^i)\},\ i = 1, 2, 3, ..., N$
Output: Conflict Zone

Step1: Identification of the conflict zone
Train C classifiers β_j on X^{2D}_{train} as follow :
for j=1 to T **do**
  Train the classifier β_j on X^{2D}_{train} ▷ Algorithm 1
  Save the parameters of β_j
end for

for Each input sample I of the X^{2D}_{train} **do**
  for Each two classifier β_j and β_k with j $\neq$ k **do**
    if $\psi_j(I) \neq \psi_k(I)$ **then** ▷ ψ_j is the output class obtained by the classifier β_j
      The I is considered to be part of the conflict zone
    end if
  end for
end for

Step2: Reduction of the conflict zone
for Each input sample I of the X^{2D}_{train} **do** ▷ *Soft Voting in Ensemble Learning*
  $P_f = Average^{classes}_{i=1} \sum_{j=1}^{C} P_{ij}$ ▷ P_{ij} *is probability of each target variable*
  $D_f = argmax\ P_f$
end for

ure 6). We can approximately see that the spatial distribution seems to be the RUL engine degradation, where the RUL's engine is linearly decreasing with time until the degradation engine reaches failure.

Figure 7 (a) shows the conflict zone obtained by the learning process described by the Algorithm 3. This conflict area is represented by black points when the classifiers gave opposite responses for the same input data I. These are considered samples with uncertain classification in the boundary between classes. As indicated in Algorithm 3, several classifiers β_j were trained with the 2D-latent space X^{2D}_{train}. If two classifiers for each input sample I of the X^{2D}_{train} give two opposite responses β_j and β_k with j $\neq$ k, sample I is regarded to be part of the conflict zone (uncertain samples). This conflict area is reduced by applying a soft voting classifier that combines the decisions of different classifiers by averaging the class probabilities (Figure 7 (b)). The soft voting classifier selects the class with the highest average probability.

TABLE VI
COMPARISON OF THE SOFT VOTING CLASSIFIER WITH OTHER POWERFUL ML MODELS, WITH AND WITHOUT SMOTE.

		Precision	Sensitivity	Specificity	F1-Score	AUC	Accuracy	G-mean
With SMOTE								
LR	Deg 0	0.99	0.98	0.77	0.99	0.98		
	Deg 1	0.51	0.69	0.98	0.59	0.98		
	Deg 2	0.37	0.76	1.00	0.50	1.00		
	Mean	0.63	0.81	0.916	0.69	0.986	**0.98**	0.86
GD	Deg 0	0.99	0.99	0.75	0.99	0.978		
	Deg 1	0.54	0.67	0.985	0.60	0.977		
	Deg 2	0.42	0.82	1.00	0.56	0.999		
	Mean	0.65	**0.83**	0.91	0.72	0.985	**0.98**	0.867
MLP	Deg 0	0.99	0.98	0.76	0.99	0.98		
	Deg 1	0.51	0.69	0.98	0.59	0.98		
	Deg 2	0.44	0.82	1.00	0.57	1.00		
	Mean	0.65	**0.83**	0.91	0.72	0.987	**0.98**	**0.87**
RF	Deg 0	0.99	0.99	0.707	0.99	0.98		
	Deg 1	0.53	0.63	0.985	0.58	0.97		
	Deg 2	0.41	0.88	0.998	0.56	1.00		
	Mean	0.64	**0.83**	0.896	0.71	0.98	**0.98**	0.86
KNN	Deg 0	0.99	0.98	0.74	0.99	0.96		
	Deg 1	0.51	0.67	0.98	0.58	0.956		
	Deg 2	0.44	0.82	1.00	0.57	0.97		
	Mean	0.65	**0.83**	0.91	0.71	0.96	**0.98**	0.866
Voting Classifier	Deg 0	0.99	0.98	0.77	0.99	0.98		
	Deg 1	0.55	0.73	0.984	0.62	0.98		
	Deg 2	0.59	0.76	1.00	0.67	1.00		
	Mean	0.71	**0.83**	**0.917**	**0.76**	**0.99**	**0.98**	**0.87**
Without SMOTE								
LR	Deg 0	0.99	1.00	0.57	0.99	0.9836		
	Deg 1	0.71	0.53	0.99	0.61	0.981		
	Deg 2	0.64	0.41	1.00	0.50	0.998		
	Mean	0.78	0.65	0.85	0.70	0.987	**0.98**	0.74
GB	Deg 0	0.99	1.00	0.52	0.99	0.9776		
	Deg 1	0.70	0.50	1.00	0.58	0.975		
	Deg 2	1.00	0.24	1.00	0.38	0.999		
	Mean	**0.90**	0.58	0.839	0.65	0.98	**0.98**	0.695
MLP	Deg 0	0.99	0.99	0.57	0.99	0.983		
	Deg 1	0.69	0.54	0.99	0.61	0.981		
	Deg 2	0.80	0.47	1.00	0.59	0.999		
	Mean	0.83	0.67	0.85	0.73	0.987	**0.98**	0.755
RF	Deg 0	0.99	0.99	0.54	0.99	0.959		
	Deg 1	0.62	0.51	0.99	0.56	0.9558		
	Deg 2	0.75	0.35	1.00	0.48	0.997		
	Mean	0.79	0.62	0.84	0.68	0.971	**0.98**	0.72
KNN	Deg 0	0.99	0.99	0.542	0.99	0.949		
	Deg 1	0.65	0.51	0.992	0.57	0.945		
	Deg 2	0.83	0.29	0.999	0.43	0.998		
	Mean	0.82	0.60	0.84	0.67	0.964	**0.98**	0.7124
Voting Classifier	Deg 0	0.99	0.99	0.57	0.99	0.9828		
	Deg 1	0.70	0.55	0.99	0.61	0.981		
	Deg 2	1.00	0.41	1.0	0.58	0.999		
	Mean	**0.90**	0.65	0.85	0.73	0.987	**0.98**	0.745

3) Performance analysis:

In this section, a comparison of the soft voting classifier with other powerful existing models has been conducted using different evaluation metrics. Table VI depicts the performance of the different ML classifiers on FD001 subset, with and without oversampling method. It can concluded from Table VI, all classifiers with and without oversampling method show a high accuracy, that is 98%. Therefore, the performance of classifiers has to be investigated on other measures, such as Precision, Sensitivity, Specificity, F1-Score and G-mean. The results in Table VI indicate a trade-off between precision and sensitivity (Recall). Our model cannot have both high Sensitivity and high precision simultaneously, either in adding synthetic samples or not (although we do aim for high precision and high recall value). When applying the oversampling SMOTE method (adding synthetic samples in both Deg 1 and 2), the performance of classifiers is slightly improved with 18%, 6.7%, 3%, and 12.5% for sensitivity, specificity, F1-Score, and G-mean, respectively, except for the Precision (SMOTE has a negative effect in both degradation class 1 and 2 Precision rate). There is a cost associated with getting lower sensitivity or precision. Ideally, in this case, we aim to

completely avoid any machine failure situations, reduce the probability of occurrence of unscheduled downtime (due to the high cost), and eliminate the causes of serious accidents (safety). Therefore, we choose to maximize sensitivity rather than precision, where the classifier can catch many faults (predicting failure) but end up with many false alarms. In other words, we are able to be endured if a non-failure is flagged as a failure, but a failure should not be labelled as a non-failure. Additionally, we compared the obtained results using F1-score that conveys both precision and sensitivity into one coherent metric, where it can be concluded that our proposed method with SMOTE showed up a higher F1-score rate compared to without SMOTE. We can clearly observe from Table VI that the method of combining classifiers (soft voting classifier) has achieved maximum precision and F1-Score value of 71% and 76%, respectively, compared to other prevailing ML algorithms.

As depicted in Figure 8, the true positive classification obtained without oversampling SMOTE gave us the lowest accuracy, which means that the soft voting classifier fails to predict the least critical degradation classes (Deg 1 and 2).

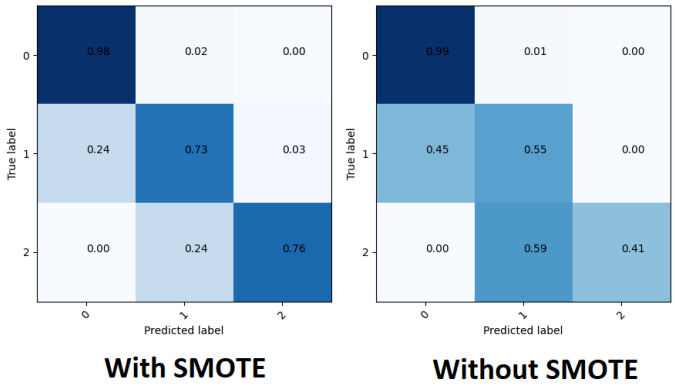

Fig. 8. The confusion matrices were obtained on the test set FD001 for the soft voting classifier, with and without SMOTE.

Figure 9 represents the ROC curve of our proposed approach ACVAE with soft voting classifier over sliding windows W_L=6 and α_1 = 10, α_2 = 30. This curve plots the true-positive rate (sensitivity) against the false-positive rate (specificity), which assesses the AUC of the degradation classes.

We also compared the obtained results from the proposed approach with two published works that have used the same subset for the same objective (See Table VII). In [51] and [19], authors have used confusion matrix as the only performance evaluation. Table VII represents the proportion of the true positive classification obtained on the test sets FD001, as well as the mean sensitivity metric. Note that Deg 0 refers to low degradation, Deg 1 indicates medium degradation, and Deg 2 points out to high degradation. The latter triggers various maintenance interventions. As it can be seen from the results, our proposed method delivers better results with different alpha values than the existing methods, as it showed up a higher sensitivity rate (as shown in Table VII). In [19], the authors used the LSTM approach for temporal features extraction and predicted the probability that the equipment

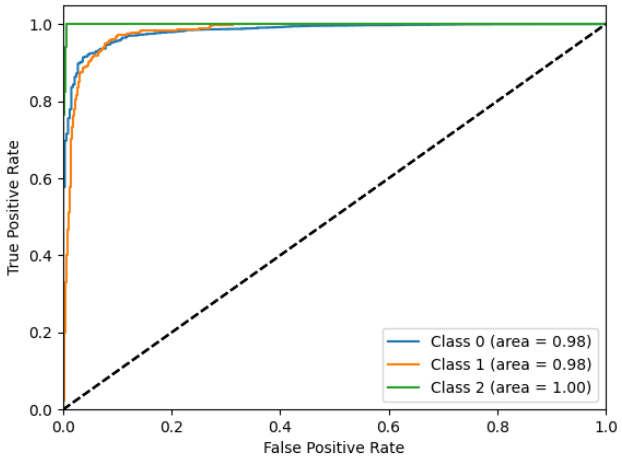

Fig. 9. ROC cure of our proposed approach over sliding windows W_L=6 and α_1 = 10, α_2 = 30.

will fail within a prespecified time window. It can be seen that the sensitivity of Deg 1 (true positive classification of Deg 1) is improved by increasing the gap between the two thresholds (α_1, α_2). Contrarily, the sensitivity of Deg 0 is decreasing. We mention that this approach fails to predict Deg 1 when the thresholds α_1 and α_2 are too near. As shown in Table IX (given in the appendix), when the machine belongs to the state Deg 0, the mean confusion probability that the machine belongs to Deg 2 is not negligible. The worst case is 3.06% where $\alpha_2 = 20$. From a maintenance viewpoint, all the false positive predictions belong to high critical degradation (such as the Deg 1 belongs to Deg 2). This estimation leads to early maintenance and significant lost costs. Contrary to their approach, all the false positive predictions of our approach belong to the less critical degradation class (this may be in some situations have a positive effect in others negative effect also). Considering a more compacted time interval leads to more precise maintenance decisions. Thus, assuming that the expert or the maintenance managers are interested in the machine's probability of deteriorating in three different time ranges where α_1 and α_2 are too near (it is also assumed by [19] for the decision making). To be more precise, we assume that the expert is interested in $\alpha_2 \leq 30$. We should point out that the predicted degradation class 1 is almost nonexistent for $\alpha_2 \leq 30$ in [19], therefore, in order to prevent this lot of false alarms (early maintenance), our proposed approach maximizes the probability of predicted degradation class 1. It can also be argued that with $\alpha_2 = 30$ both critical degradation classes (Deg 1 and Deg 2) were correctly predicted using our approach compared to both related works. The proportion of the true positive classification for the degradation classes 0, 1 and 2 are respectively 98%, 73% and 76%. Furthermore, the proposed method allows a significantly reducing of the maintenance cost rates, where the continuity in the latent space leads to the machine belonging to the state Deg 0 (low degradation) does not belong to Deg 2 (high degradation). On other hand, in our approach, it can be seen that the sensitivity of Deg 2 is improved as well as the sensitivity of Deg 1 decreases with increasing α_2. Contrarily, in [51] the sensitivity of Deg 1 is

TABLE VII
COMPARISON WITH PREVIOUS WORK.

		Probability confusion matrix			
		$\alpha_1 = 10, \alpha_2 = 20$	$\alpha_1 = 10, \alpha_2 = 30$	$\alpha_1 = 10, \alpha_2 = 70$	$\alpha_1 = 10, \alpha_2 = 90$
[19]	Deg 0	94.25	88.74	**66.67**	46.19
	Deg 1	0.39	14.45	**67.46**	77.33
	Deg 2	99.98	99.98	**99.97**	99.99
	Mean Sensitivity	64.87	67.72	**78.03**	74.5
[51]	Deg 0	100	99	98	**91**
	Deg 1	7	46	46	**60**
	Deg 2	94	75	75	**69**
	Mean Sensitivity	67	73.33	73	**73.33**
Our approach	Deg 0	97	**98**	95	91
	Deg 1	85	**73**	60	52
	Deg 2	47	**76**	82	94
	Mean Sensitivity	76.33	**83**	79	79

improved as well as the sensitivity of Deg 2 decreases with increasing α_2.

V. CONCLUSION

In this paper, a new RUL prediction approach based on ACVAE jointly with a soft voting classifier has been presented as conducive to the predictive maintenance of aero-engines. Thus, the efficiency of the proposed method has been highlighted in the numerical analysis using the C-MAPSS dataset. This approach starts with an automatic extraction of performance degradation features from multiple sensors using the ACVAE method. In this model, the power of the attention layer is to dynamically increase the weights of the useful features in the ACVAE encoding phase to make the network pay attention to these vital features for RUL classes estimation. As articulated in the results section, it was also demonstrated that the ACVAE could cluster the dataset more effectively according to RUL degradation reasoning. In this context, 2D-latent space of ACVAE behaves better than the existing dimension reduction methods (PCA, ISOMAP, and T-SNE), which resulted in a more compacted and better spatial distribution compared with the covered areas by other methods. The conflict zones, which are located near the boundaries between classes, are identified when the classifiers give opposite responses for the same input data. Therefore, to reduce this conflict zone, the soft voting classifier is used. It selects the highest probability class by combining the decisions of different classifiers using the probability classes average.

In this work, it was clearly observed that the obtained results show significant improvements of the RUL prediction compared with previous similar works. However, many future works could focus on : i) optimizing conflict zones by identifying outliers and omitting them, ii) giving a formal approach to define the suitable thresholds $(\alpha_1, \alpha_2, \ldots, \alpha_i)$, iii) combining a maintenance strategy with our RUL prediction approach could be useful in the decision making process, and IV) improving generation performance of synthetic samples using our ACVAE method combined with generative adversarial network (GAN) [92].

ACKNOWLEDGEMENT

We acknowledge DGRSDT (General Directorate of Scientific Research and Technological Development (Algeria), for supporting the present work.

REFERENCES

[1] M. L. H. Souza, C. A. da Costa, G. de Oliveira Ramos, and R. da Rosa Righi, "A survey on decision-making based on system reliability in the context of industry 4.0," *Journal of Manufacturing Systems*, vol. 56, pp. 133–156, 2020.

[2] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 international conference on prognostics and health management*. IEEE, 2008, pp. 1–9.

[3] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications—a survey of recent literature," *IEEE transactions on reliability*, 2019.

[4] I. Remadna, S. L. Terrissa, R. Zemouri, and S. Ayad, "An overview on the deep learning based prognostic," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*. IEEE, 2018, pp. 196–200.

[5] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," *Mechanical systems and signal processing*, vol. 21, no. 1, pp. 193–207, 2007.

[6] S. Saon, T. Hiyama *et al.*, "Predicting remaining useful life of rotating machinery based artificial neural network," *Computers & Mathematics with Applications*, vol. 60, no. 4, pp. 1078–1087, 2010.

[7] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 227–237, 2012.

[8] A. K. Jain and B. K. Lad, "Predicting remaining useful life of high speed milling cutters based on artificial neural network," in *2015 International Conference on Robotics, Automation, Control and Embedded Systems (RACE)*. IEEE, 2015, pp. 1–5.

[9] A. Lahmadi, L. Terrissa, and N. Zerhouni, "A data-driven method for estimating the remaining useful life of a composite drill pipe," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*. IEEE, 2018, pp. 192–195.

[10] P. Kundu and B. K. Lad, "Pca-ann based approach for remaining useful life prediction for roller ball bearings," in *International Conference on Business Analytics and Intelligence*, vol. 17, no. 2015, 2015.

[11] E. Ramasso and R. Gouriveau, "Remaining useful life estimation by classification of predictions based on a neuro-fuzzy system and theory of belief functions," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 555–566, 2014.

[12] A. Malhi and R. Gao, "Recurrent neural networks for long-term prediction in machine condition monitoring," in *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No. 04CH37510)*, vol. 3. IEEE, 2004, pp. 2048–2053.

[13] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 international conference on prognostics and health management*. IEEE, 2008, pp. 1–6.

[14] A. Malhi, R. Yan, and R. X. Gao, "Prognosis of defect propagation based on recurrent neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703–711, 2011.

[15] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network," in *2016 IEEE international conference on aircraft utility systems (AUS)*. IEEE, 2016, pp. 135–140.

[16] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE international conference on prognostics and health management (ICPHM)*. IEEE, 2017, pp. 88–95.

[17] O. Aydin and S. Guldamlasioglu, "Using lstm networks to predict engine condition on large scale data processing framework," in *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*. IEEE, 2017, pp. 281–285.

[18] C.-S. Hsu and J.-R. Jiang, "Remaining useful life estimation using long short-term memory deep learning," in *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE, 2018, pp. 58–61.

[19] K. T. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," *Reliability Engineering & System Safety*, vol. 188, pp. 251–262, 2019.

[20] J. Wu, K. Hu, Y. Cheng, H. Zhu, X. Shao, and Y. Wang, "Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network," *ISA transactions*, vol. 97, pp. 241–250, 2020.

[21] Y. Sun, J. Kang, L. Sun, P. Jin, and X. Bai, "Condition-based maintenance for the offshore wind turbine based on long short-term memory network," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, p. 1748006X20965434, 2020.

[22] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining useful life estimation in prognostics using deep bidirectional lstm neural network," in *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. IEEE, 2018, pp. 1037–1042.

[23] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *Journal of manufacturing systems*, vol. 48, pp. 78–86, 2018.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[25] F. F. dos Santos, P. F. Pimenta, C. Lunardi, L. Draghetti, L. Carro, D. Kaeli, and P. Rech, "Analyzing and increasing the reliability of convolutional neural networks on gpus," *IEEE Transactions on Reliability*, vol. 68, no. 2, pp. 663–677, 2018.

[26] Y. Lin, H. Zhao, X. Ma, Y. Tu, and M. Wang, "Adversarial attacks in modulation recognition with convolutional neural networks," *IEEE Transactions on Reliability*, vol. 70, no. 1, pp. 389–401, 2020.

[27] A. Belaala, Y. Bourezane, L. S. Terrissa, Z. Al Masry, and N. Zerhouni, "Skin cancer and deep learning for dermoscopic images classification: A pilot study." 2020.

[28] I. Remadna, S. L. Terrissa, R. Zemouri, S. Ayad, and N. Zerhouni, "Leveraging the power of the combination of cnn and bi-directional lstm networks for aircraft engine rul estimation," in *2020 Prognostics and Health Management Conference (PHM-Besançon)*. IEEE, 2020, pp. 116–121.

[29] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*. Springer, 2016, pp. 214–228.

[30] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[31] J. Li, X. Li, and D. He, "A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction," *IEEE Access*, vol. 7, pp. 75 464–75 475, 2019.

[32] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for remaining useful life estimation," *Computers in Industry*, vol. 108, pp. 186–196, 2019.

[33] I. Remadna, L. S. Terrissa, S. Ayad, and N. Zerhouni, "Rul estimation enhancement using hybrid deep learning methods," *International Journal of Prognostics and Health Management*, vol. 12, no. 1, 2021.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[35] J. Pareek and J. Jacob, "Data compression and visualization using pca and t-sne," in *Advances in Information Communication Technology and Computing*. Springer, 2021, pp. 327–337.

[36] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne)," *Computer Science Review*, vol. 40, p. 100378, 2021.

[37] Y. Song, G. Shi, L. Chen, X. Huang, and T. Xia, "Remaining useful life prediction of turbofan engine using hybrid model based on autoencoder and bidirectional long short-term memory," *Journal of Shanghai Jiaotong University (Science)*, vol. 23, no. 1, pp. 85–94, 2018.

[38] J. Ma, H. Su, W.-l. Zhao, and B. Liu, "Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning," *Complexity*, vol. 2018, 2018.

[39] X. Hou, K. Sun, L. Shen, and G. Qiu, "Improving variational autoencoder with deep feature consistent and generative adversarial training," *Neurocomputing*, vol. 341, pp. 183–194, 2019.

[40] R. Zemouri, M. Levesque, N. Amyot, C. Hudon, O. Kokoko, and S. A. Tahan, "Deep convolutional variational autoencoder as a 2d-visualization tool for partial discharge source classification in hydrogenerators," *IEEE Access*, vol. 8, pp. 5438–5454, 2019.

[41] S. Lee, M. Kwak, K.-L. Tsui, and S. B. Kim, "Process monitoring using variational autoencoder for high-dimensional nonlinear processes," *Engineering Applications of Artificial Intelligence*, vol. 83, pp. 13–27, 2019.

[42] Z. Zhang, T. Jiang, S. Li, and Y. Yang, "Automated feature learning for nonlinear process monitoring–an approach using stacked denoising autoencoder and k-nearest neighbor rule," *Journal of Process Control*, vol. 64, pp. 49–61, 2018.

[43] Z. Zhang, T. Jiang, C. Zhan, and Y. Yang, "Gaussian feature learning based on variational autoencoder for improving nonlinear process monitoring," *Journal of Process Control*, vol. 75, pp. 136–155, 2019.

[44] F. Cheng, Q. P. He, and J. Zhao, "A novel process monitoring approach based on variational recurrent autoencoder," *Computers & Chemical Engineering*, vol. 129, p. 106515, 2019.

[45] K. Wang, M. G. Forbes, B. Gopaluni, J. Chen, and Z. Song, "Systematic development of a new variational autoencoder model based on uncertain data for monitoring nonlinear processes," *IEEE Access*, vol. 7, pp. 22 554–22 565, 2019.

[46] Y.-r. Wang, Q. Jin, G.-d. Sun, and C.-f. Sun, "Planetary gearbox fault feature learning using conditional variational neural networks under noise environment," *Knowledge-Based Systems*, vol. 163, pp. 438–449, 2019.

[47] M. Mastroleo, R. Ugolotti, L. Mussi, E. Vicari, F. Sassi, F. Sciocchetti, B. Beasant, and C. McIlroy, "Automatic analysis of faulty low voltage network asset using deep neural networks," *The Journal of Engineering*, vol. 2018, no. 15, pp. 851–855, 2018.

[48] G. San Martin, E. López Droguett, V. Meruane, and M. das Chagas Moura, "Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis," *Structural Health Monitoring*, vol. 18, no. 4, pp. 1092–1128, 2019.

[49] W. Mao, Y. Liu, L. Ding, and Y. Li, "Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study," *IEEE Access*, vol. 7, pp. 9515–9530, 2019.

[50] G. S. Chadha, A. Rabbani, and A. Schwung, "Comparison of semi-supervised deep neural networks for anomaly detection in industrial processes," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 214–219.

[51] R. Zemouri, Z. Al Masry, I. Remadna, S. L. Terrissa, and N. Zerhouni, "Hybrid architecture of deep convolutional variational auto-encoder for remaining useful life prediction," 2020.

[52] R. Zemouri, M. Lévesque, N. Amyot, C. Hudon, and O. Kokoko, "Deep variational autoencoder: An efficient tool for phm frameworks," in *2020 Prognostics and Health Management Conference (PHM-Besançon)*. IEEE, 2020, pp. 235–240.

[53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[54] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[55] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[56] X. Liu, J. Yu, and L. Ye, "Residual attention convolutional autoencoder for feature learning and fault detection in nonlinear industrial processes," *Neural Computing and Applications*, pp. 1–17, 2021.

[57] P. R. d. O. da Costa, A. Akcay, Y. Zhang, and U. Kaymak, "Attention and long short-term memory network for remaining useful lifetime predictions of turbofan engine degradation," *International Journal of Prognostics and Health Management*, vol. 10, p. 034, 2019.

[58] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning

approach," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2020.
[59] W. M. Tan and T. H. Teo, "Remaining useful life prediction using temporal convolution with attention," *AI*, vol. 2, no. 1, pp. 48–70, 2021.
[60] J. Li, K. Jin, D. Zhou, N. Kubota, and Z. Ju, "Attention mechanism-based cnn for facial expression recognition," *Neurocomputing*, vol. 411, pp. 340–350, 2020.
[61] G. Xue, S. Liu, and Y. Ma, "A hybrid deep learning-based fruit classification using attention model and convolution autoencoder," *Complex & Intelligent Systems*, pp. 1–11, 2020.
[62] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *25th annual conference on neural information processing systems (NIPS 2011)*, vol. 24. Neural Information Processing Systems Foundation, 2011.
[63] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.
[64] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *arXiv preprint arXiv:1206.2944*, 2012.
[65] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
[66] T. T. Joy, S. Rana, S. Gupta, and S. Venkatesh, "Hyperparameter tuning for big data using bayesian optimisation," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2574–2579.
[67] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.
[68] H. Bertrand, "Hyper-parameter optimization in deep learning and transfer learning: applications to medical imaging," Ph.D. dissertation, Université Paris-Saclay, 2019.
[69] K. Javed, R. Gouriveau, and N. Zerhouni, "State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels," *Mechanical Systems and Signal Processing*, vol. 94, pp. 214–236, 2017.
[70] S. Yu and J. C. Principe, "Understanding autoencoders with information theoretic concepts," *Neural Networks*, vol. 117, pp. 104–123, 2019.
[71] Y. Bengio, "How auto-encoders could provide credit assignment in deep networks via target propagation," *arXiv preprint arXiv:1407.7906*, 2014.
[72] Y. J. Fan, "Autoencoder node saliency: Selecting relevant latent representations," *Pattern Recognition*, vol. 88, pp. 643–653, 2019.
[73] S. W. Canchumuni, A. A. Emerick, and M. A. C. Pacheco, "Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother," *Computers & Geosciences*, vol. 128, pp. 87–102, 2019.
[74] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[75] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," *IEEE Access*, vol. 6, pp. 33 353–33 361, 2018.
[76] D. P. Kingma, "Variational inference & deep learning: A new synthesis," 2017.
[77] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
[78] A. Proteau, R. Zemouri, A. Tahan, and M. Thomas, "Dimension reduction and 2d-visualization for early change of state detection in a machining process with a variational autoencoder approach," *The International Journal of Advanced Manufacturing Technology*, vol. 111, no. 11, pp. 3597–3611, 2020.
[79] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.
[80] M. S. Lee, J.-K. Rhee, B.-H. Kim, and B.-T. Zhang, "Aesnb: active example selection with naïve bayes classifier for learning from imbalanced biomedical data," in *2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering*. IEEE, 2009, pp. 15–21.
[81] S. Oh, M. S. Lee, and B.-T. Zhang, "Ensemble learning with active example selection for imbalanced biomedical data classification," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 8, no. 2, pp. 316–325, 2010.
[82] B. Settles, "Active learning literature survey," 2009.
[83] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2019.
[84] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, pp. 1551–3203, 2008.
[85] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020.
[86] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on nuclear science*, vol. 44, no. 3, pp. 1464–1468, 1997.
[87] D. Elreedy and A. F. Atiya, "A novel distribution analysis for smote oversampling method in handling class imbalance," in *International Conference on Computational Science*. Springer, 2019, pp. 236–248.
[88] A. Belaala, L. S. Terrissa, N. Zerhouni, and C. Devalland, "Computer-aided diagnosis for spitzoid lesions classification using artificial intelligence techniques," *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, vol. 16, no. 1, pp. 16–37, 2021.
[89] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using bayesian optimization," *Evolving Systems*, vol. 12, pp. 217–223, 2021.
[90] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Keras tuner," *Retrieved May*, vol. 21, p. 2020, 2019.
[91] M. Zhang, H. Li, J. Lyu, S. H. Ling, and S. Su, "Multi-level cnn for lung nodule classification with gaussian process assisted hyperparameter optimization," *arXiv preprint arXiv:1901.00276*, 2019.
[92] R. Zemouri, "Semi-supervised adversarial variational autoencoder," *Machine Learning and Knowledge Extraction*, vol. 2, no. 3, pp. 361–378, 2020.

APPENDIX

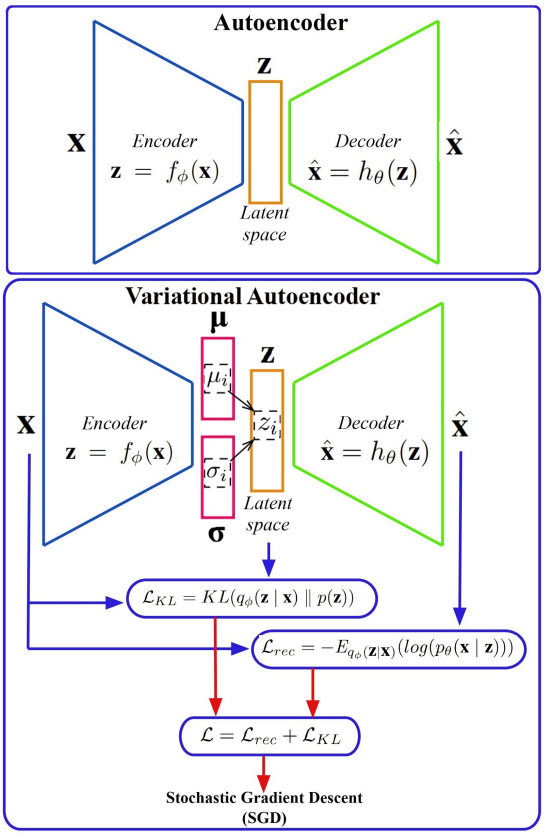

Fig. 10. The architecture of an autoencoder and a variational autoencoder with VAE loss function.

BIOGRAPHIES

TABLE VIII
PREVIOUS WORK COMPARISON.

Works	Data analysis Task	Architecture	Attention Mechanism	Dimensionality Reduction	Visual Explanation	AHPS	Ensemble learning
[29]	Regression	CNN	✗	✓	✗	Trial-and-error	✗
[15]	Regression	LSTM	✗	✗	✗	Trial-and-error	✗
[16]	Regression	Deep LSTM	✗	✗	✗	Trial-and-error	✗
[17]	Classification	LSTM	✗	✗	✗	Trial-and-error	✗
[18]	Regression	LSTM	✗	✗	✗	Trial-and-error	✗
[22]	Regression	BLSTM	✗	✗	✗	Trial-and-error	✗
[23]	Regression	BLSTM	✗	✗	✗	Trial-and-error	✗
[30]	Regression	CNN + FFNN	✗	✗	✗	Trial-and-error	✗
[37]	Regression	AE + BLSTM + FFNN	✗	✓	✗	Grid search	✗
[38]	Regression	SAE + LR	✗	✓	✗	Grid search	✗
[31]	Regression	LSTM + CNN	✗	✓	✗	Trial-and-error	✗
[32]	Regression	LSTM + CNN + FFNN	✗	✓	✗	Trial-and-error	✗
[57]	Regression	LSTM + FFNN	✓	✗	✓	Grid search	✗
[19]	Classification	LSTM	✗	✗	✗	Trial-and-error	✗
[20]	Regression	DLSTM	✗	✗	✗	Grid search	✗
[51]	Classification	CVAE + MLP	✗	✓	✓	Trial-and-error	✗
Our proposed approach	Classification	CVAE + Stacking classifiers	✓	✓	✓	Bayesian optimization	✓

(✓): verified (✗): not verified

TABLE IX
COMPARISON WITH PREVIOUS WORK, CONFUSION MATRIX.

		Probability confusion matrix											
		$\alpha_1 = 10, \alpha_2 = 20$			$\alpha_1 = 10, \alpha_2 = 30$			$\alpha_1 = 10, \alpha_2 = 70$			$\alpha_1 = 10, \alpha_2 = 90$		
		Deg 0	Deg 1	Deg 2	Deg 0	Deg 1	Deg 2	Deg 0	Deg 1	Deg 2	Deg 0	Deg 1	Deg 2
[19]	Deg 0	**94.25**	2.7	3.06	**88.74**	9.81	1.44	**66.67**	33.18	0.16	**46.19**	53.78	0.03
	Deg 1	0	**0.39**	99.61	0.04	**14.45**	85.51	3.8	**67.46**	28.74	3.94	**77.33**	18.73
	Deg 2	0	0.02	**99.98**	0	0.02	**99.98**	0	0.03	**99.97**	0	0.01	**99.99**
[51]	Deg 0	**100**	0	0	**99**	1	0	**98**	1	1	**91**	9	0
	Deg 1	45	**7**	48	45	**46**	9	53	**46**	1	38	**60**	2
	Deg 2	0	6	**94**	0	25	**75**	0	25	**75**	0	31	**69**
Our approach	Deg 0	**97**	3	0	**98**	2	0	**95**	5	0	**91**	9	0
	Deg 1	7	**85**	8	24	**73**	3	37	**60**	3	47	**52**	1
	Deg 2	0	53	**47**	0	24	**76**	0	18	**82**	0	6	**94**

Ikram Remadna received the B.S. degree in experimental science in 2011, the M.S. degree in mathematics and computer science (Artificial Intelligence) from the University of Mohamed Khider Biskra, Biskra, Algeria, in 2016. She is currently pursuing PhD degree in computer science with LINFI Laboratory at Biskra University. Her current research interests include maintenance, Prognostics and Health Management and Deep Learning.

Labib Sadek Terrissa is Professor in computer science at Biskra University, Algeria. He conducts his research activities within LINFI Laboratory where he is the "CoViBio" team head. He is also a senior consultant in digitalization project management. After receiving an engineering degree in electronics, he received his Ph.D. in computer engineering in 2006 from Le Havre University, France. His current research interests include Cloud computing, Machine learning, Smart maintenance, and Prognostic and Health Management.

Zeina Al Masry received her PhD. in Applied Mathematics from the University of Pau and Pays de l'Adour in France in 2016. She joined École Nationale Supérieure de Mécanique et des Microtechniques (ENSMM) in Besançon, France as an associate professor in 2017. She is doing her research activities at FEMTO-ST institute in the Prognostics and Health Management (PHM) research group. Her research works concern data management, applied statistics and stochastic processes for PHM purposes.

Noureddine Zerhouni holds a doctorate in Automatic-Productivity from the National Polytechnic Institute of Grenoble (INPG), France, in 1991. He was a lecturer at the National School of Engineers (ENI, UTBM) in Belfort. Since 1999, he is Professor of Universities at the National School of Mechanics and Microtechnics(ENSMM) in Besançon. He is doing his research in the Automatic department of the FEMTO-ST Institute in Besançon. His areas of research are related to the monitoring and maintenance of production systems. He is also an expert in adult education in the areas of process improvement and project management.