



HAL
open science

Boosting Regression Assistive Predictive Maintenance of the Aircraft Engine with Random-Sampling Based Class Balancing

Ibrahima Barry, Meriem Hafsi, Saeed Mian Qaisar

► **To cite this version:**

Ibrahima Barry, Meriem Hafsi, Saeed Mian Qaisar. Boosting Regression Assistive Predictive Maintenance of the Aircraft Engine with Random-Sampling Based Class Balancing. 13th International Conference on Information Systems and Advanced Technologies, Dec 2023, Hammamet, Tunisia. hal-04369687

HAL Id: hal-04369687

<https://hal.science/hal-04369687v1>

Submitted on 2 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Boosting Regression Assistive Predictive Maintenance of the Aircraft Engine with Random-Sampling Based Class Balancing

Ibrahima Barry¹, Meriem Hafsi^{1*}, Saeed Mian Qaisar^{1,2}

^{1*}CESI LINEACT, 15C Av. Albert Einstein, Villeurbanne, 69100, France.

^{2*}Electrical and Computer Engineering Department, Effat University, Jeddah, 22332, KSA.

*Corresponding author(s). E-mail(s): mhafsi@cesi.fr;
Contributing authors: ibarry@cesi.fr; smianqaisar@cesi.fr;

Abstract

This study presents the development of a data-driven predictive maintenance model in the context of industry 4.0. The solution is based on a novel hybridization of Remaining Useful Life (RUL) generation, Min-Max normalization, random-sampling based class balancing, and XGBoost regressor. The applicability is tested using the NASA's C-MAPSS dataset, which contains aircraft engine simulation data. The objective is to develop an effective and Artificial Intelligence (AI) assistive automated aircraft engine's RUL predictor. It can maximize the benefits of predictive maintenance. The rules based RUL generation provides a ground truth for evaluating the performance of intended regressors. The Min-Max normalization linearly transforms the intended dataset and scales the multi subject's data in a common range. The imbalance presentation among intended classes can lead towards a biasness in findings. This issue is intelligently resolved using the uniformly distributed random sub-sampling. Onward, the performance of robust machine learning and ensemble learning algorithms is compared for predicting the RUL of the considered aircraft engine by processing the balanced dataset. The results have shown that the XGBoost regressor, uses an ensemble of decision trees, outperforms other considered models. The root mean square error (RMSE) and mean absolute error (MAE) indicators will be used to evaluate the prediction performances. The devised method secures the RMSE value of 12.88%. It confirms a similar or better performance compared to the state-of-the-art counterparts.

Keywords: XGBoost Regressor a Random-Sampling; Class Balancing; Remaining Useful Life (RUL); C-MAPSS; Machine learning; Ensemble learning; Evaluation measures

1 Introduction

Over the past few decades, the manufacturing and industrial sector has witnessed continuous evolution and growth. We have just entered the era of Industry 4.0, and yet we are already beginning to explore and envision the prospects of Industry 5.0. In this context of constant evolution [1], companies must proactively adapt to changes to ensure their competitiveness, particularly in the maintenance of industrial equipment, which plays a crucial role. Indeed, for optimal levels of quality and profitability in production, it is imperative to maintain the equipment in an optimal operating condition [2]. Therefore, the central objective of industrial maintenance is to ensure the availability, reliability and efficiency of machines and components, minimizing unplanned downtime and optimizing production performance [3]. In this context, ensuring and maintaining the health status of equipments and their components is a significant challenge in complex industrial systems where fulfilling their function relies on ensuring their reliability and efficiency [4]. Prognostics and Health Management (PHM) is a discipline that focuses on understanding the degradation mechanisms of systems, aiming to estimate their health status and proactively anticipate potential failures [5]. Its ultimate objective is to optimize maintenance tasks, ensuring improved system performance and reliability. PHM encompasses different methods including monitoring, detection of anomalies, diagnostics of causes, prognosis of remaining useful life (RUL) and optimization of maintenance [6]. It thus makes it possible to monitor the state of

health of the system in real time and to provide relevant information for decision support on the actions to be implemented to maintain the system in optimal operational conditions. The integration of the Internet of Things (IoT) and hyper-connectivity of equipment in the industry has catapulted it into a new era [3], where the massive data generated represents a formidable treasure and offers a wide array of possibilities. At the same time, the integration of advanced technologies such as automation, exponential computing capabilities, data analysis and machine learning, industrial maintenance aims to prevent breakdowns and failures [4], anticipate repair needs and maximize the life of assets, enabling companies to remain competitive and promoting their productivity and efficiency [7]. Predictive Maintenance (PdM)[1, 6] is one of the most promising applications of PHM which aims to predict failures and errors by estimating the RUL of a system or equipment. Over the past few decades, the literature has put forth various methods and approaches – based on physics, data or experiences– [3] with diverse applications in manufacturing, telemetry [6], automotive, railway, aviation and aeronotic [2], centrifugal compressor industry, industrial rotating machines and semiconductor, batteries life estimation [8] and others domains [1]. Aircraft engine life estimation [2] is one of the applications that have aroused growing interest in recent years, notably accentuated by the accessibility of the C-MAPSS (Commercial Modular Aero-Propulsion System Simulation turbofan engine) dataset provided by NASA¹. for scientific communities working on the RUL prediction and development of new methods [9]. Indeed, aircraft engines are crucial assets with stringent reliability requirements and demanding airworthiness conditions. They face significant challenges as any unforeseen degradation can result in substantial operational disruptions, delays, costly downtime, and potentially even fatal accidents [10]. In the aerospace industry, maintenance is undergoing a swift transformation, moving from preventive strategy to condition-based maintenance (CBM) [5] and predictive maintenance (PdM). In traditional preventive maintenance, maintenance actions such as replacements, inspections, and data collection are scheduled based on fixed time intervals [3]. However, this method lacks the ability to assess each asset individually as their behavior may vary under different contexts or operating conditions [2]. PdM seeks to address this limitation by predicting the state and reliability of each asset, estimating its time to failure [10].

PdM aims to incorporate reliable RUL prognostic into maintenance planning effectively [9]. The literature is replete with data-driven works to estimate the RUL of aircraft engines. However, the machine learning-based approaches proposed lack rigor, and the results are not thoroughly analyzed, leading to a lack of continuation in the research. Additionally, while few works employ advanced conventional machine learning methods such as KNN (K Nearest Neighbors), RF (Random Forest), SVM (Support Vector Machine) [4, 6], many authors favor DL algorithms for its superior performance [9]. Nonetheless, the interpretation and explainability of results is often absent [3], likely due to the black box aspect of DL. Consequently, the prevailing observation is that the data-driven approach dominates the current research context [1]. In this article, we address the challenges of predictive maintenance for aircraft engines, focusing specifically on the RUL prognostic using ML techniques. Based on the issues highlighted in the literature, we propose a novel method using conventional ML methods that offer both high performance and improved interpretability of the training model.

1.1 Background

1.1.1 RUL Prediction

According to the previous section, PdM employs advanced data analysis techniques and artificial intelligence to predict failures before they occur by monitoring early warning signals of malfunction. it involves predicting the remaining useful life (RUL) of a component of equipment before it experiences a critical failure. This helps in effectively scheduling maintenance, reducing downtime, and preventing unexpected breakdowns [2].

¹C-MAPSS NASA Dataset: <https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xaut-bemq>

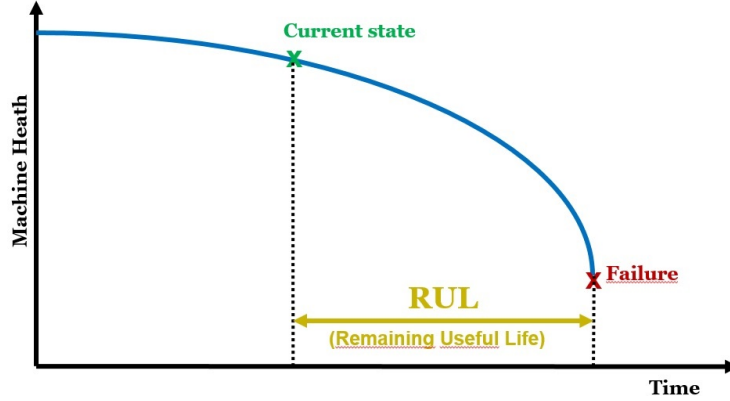


Fig. 1 System degradation over time and RUL estimation which represents the duration a system can continue operating before it needs to undergo repair or replacement due to failure.

The RUL concept (figure 1) is based on the recognition that equipment undergoes progressive wear and tear over time, which can lead to failures or malfunctions, by monitoring equipment operating parameters in real-time, such as vibrations, temperature, pressure, and electrical currents, and using sophisticated algorithms to analyze this data, collected from multiple sensors, it is possible to predict with some accuracy the point at which a piece of equipment is likely to fail [4].

1.1.2 PdM Approaches

Four approaches are mainly used in the literature for RUL prediction: physical model-based, knowledge-based model, data-driven model and hybrid model [1, 11]:

- **Physical-based approach:** describes the physical degradation process and fault propagation with mathematical models [3]. It is particularly useful for understanding complex physical phenomena and where data-driven models are not sufficient due to the lack of data, but it can be complex, computationally expensive, inflexible, and require data calibration, updates, and significant initial investment [10].
- **Data-driven approach:** aims to detect the degradation evolution and forecast the future health state of a system by leveraging historical data [?], and relies on techniques such as statistical analysis, ML, and DM to extract valuable insights. It is particularly adept at detecting patterns and uncovering hidden correlations, leading to enhanced accuracy in failure predictions. Nevertheless, the lack of data, system complexity, and varying operational conditions limit its widespread application. finally, the lack of explainability of ML-based models represent a major limitation [12].
- **Knowledge-based approach:** utilizes human expertise and equipment-specific knowledge to anticipate failures, by integrating and formalizing field observations, diagnostic rules, and technician experience [3]. It offer an explicative results and enhances data quality, models causal relationships, adapts to changing operational conditions and reduces dependence on large volumes of data. By incorporating domain expertise, it fills the gaps in data-driven approaches, leading to more accurate models and better failure anticipation[5]. However, it may face some limitations, such as the difficulty of obtaining precise knowledge for predictability pruposes from experience and limited access to expert sources.
- **Hybrid approach:** combines two or more of the previous PdM approach for increasing predictive accuracy [5, 12].

Table 1 Comparison of PdM Approaches

Approach	Advantages	Disadvantages
Physics-based	Without historical data, reliable	Complexity of systems
Data-driven	High accuracy	Requires a large amount of data, interpretation
Knowledge-based	Effective for simplified cases, explainability	Low accuracy, limited in complex systems
Hybrid	High Accuracy, explainability	Hard to buid, fusion and uncertainty management

in this present work, we are particularly interested in the data-driven approach using conventional ML applied to aircraft engine data. the next section discusses existing work in this area.

1.2 Related Works

Several works on RUL estimation of aircraft engines have been carried out since the release of the C-MAPSS dataset [2]. Many of these works have utilized DL algorithms [13–24], considering them the most appropriate

for this dataset: [Sateesh Babu et al.\[13\]](#) employed Convolutional Neural Networks (CNN) to RUL estimation and [de Pater et al.\[9\]](#) developed an alarm-based maintenance planning framework that integrates RUL prognostic based on CNN in the maintenance schedule. [Zheng et al.\[14\]](#) implemented Long Short-Term Memory (LSTM) networks in their research and [Wang et al.\[15\]](#) opted for Bidirectional LSTM (BiLSTM) in their analysis. [Liu et al.\[22\]](#) applied an Attention-based Bidirectional LSTM (Bi-LSTM) method. Some works have combined several approaches like CNN and LSTM [17] or with BiLSTM [25]. [Mo et al.\[18\]](#) developed a multi-head architecture that combines CNN and LSTM, [Kumar\[20\]](#) introduced a hybrid model for their analysis, [Asif et al.\[23\]](#) merged LSTM with Automatic Piecewise Linear Modeling, and [Peng et al.\[26\]](#) combined LSTM with a Fully Connected Layer CNN (FCLCNN) in their research. Other authors have proposed original methods such as Adaptive Gradient CNN (AGCNN) in [19], a Broad Learning System (BLS) coupled with a Temporal Convolutional Network (TCN) is presented in [21], [Li et al.\[16\]](#) integrated Directed Acyclic Graphs (DAG) in their methodology and [Liu et al.\[22\]](#) applied an Attention-based Bidirectional LSTM (Bi-LSTM) method. Finally, [Hu et al.\[24\]](#) applied a Deep Reinforcement Learning approach for RUL estimation (DRLRULE) in their investigation. However, in this particular study, we demonstrate that traditional ML algorithms can also be highly effective.

1.3 Contribution

In this paper, a novel hybridization of RUL generation, Min-Max normalization, random-sampling based class balancing, and XGBoost regressor is proposed for an automated aircraft engine’s RUL prediction.

- The rules based RUL generation provides a ground truth for evaluating the performance of intended regressors.
- The Min-Max normalization linearly transforms the intended dataset and scales the multi subject’s data in a common range.
- The imbalance presentation among intended classes can lead towards a biasness in findings. This issue is intelligently resolved using the uniformly distributed random sub-sampling.
- The performance of robust ML and ensemble learning algorithms is compared for predicting the RUL of the considered aircraft engine by processing the balanced dataset. The results have shown that the XGBoost regressor outperforms other considered models.
- The devised method secures the RMSE value of 12.88%. It confirms a similar or better performance compared to the state-of-the-art counterparts.

The rest of the paper is organized as follow: Section 2 describes the used materials and methods. The findings are presented and described in Section 3. Finally, the conclusion is made in Section 4.

2 Materials and Methods

2.1 Methodology

In order to answer the problem mentioned above, we have implemented our solution according to the methodology described by The functional diagram in Figure 2). The described architecture provides an overview of key system components and their interconnections, demonstrating how data flows through the system from one stage to another, and how the components interact with each other. All these steps will be detailed in the following sections.

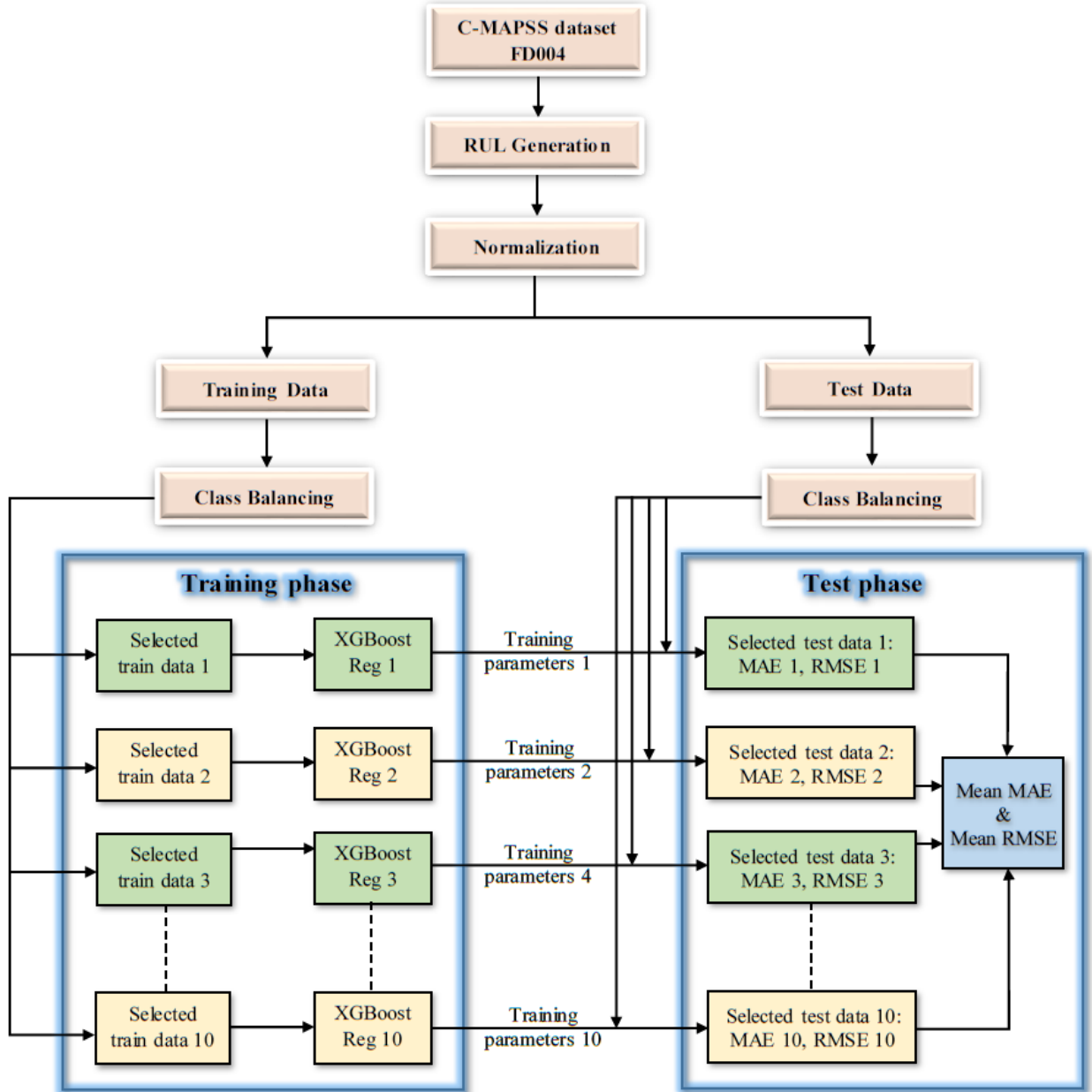


Fig. 2 Proposed framework for RUL prediction based on Random Sampling Based Class Balancing Method.

2.2 Dataset

C-MAPSS (Commercial Modular Aero-propulsion System Simulation) is provided by NASA (National Aeronautics and Space Administration) and was first made available in 2008 during the International Conference on PHM (Prognostics and Health Management). Since its release, it has been extensively researched to enhance the accuracy of RUL prediction, with the ultimate goal of planning component maintenance before failure occurs [27]. C-MAPSS dataset is widely recognized as the benchmark dataset for RUL prognostic. Moreover, its characteristics enable the findings to be applicable to various other research and industrial use cases. The dataset includes multivariate time series augmented with metadata on operational settings. Additionally, it encompasses multiple physical interactions among the recorded properties. The dataset also covers various fault modes and operating conditions, while considering manufacturing variations, distinct lifespans, and natural failures [2]. C-MAPSS is subdivided into four units: FD001, FD002, FD003, and FD004 with different multivariate time series and various numbers of engines. Each unit contains three files: training data, testing data and RUL data. Table 2 describes characteristics of each unit.

The dataset is composed of 21 sensors with the application of three different operating conditions in relation to the cycle time for each engine. A turbojet engine is mainly composed of: Low Pressure Turbine (**LPT**), High Pressure Turbine (**HPT**), Fan, Nozzle, Low Pressure Chamber (**LPC**), High Pressure Chamber (**HPC**), etc.(Fig. 3) shows how these different components and the way they are interconnected.

Table 2 Description of the properties and different batches of the C-MAPSS dataset.

	FD001 Unit	FD002 Unit	FD003 Unit	FD004 Unit
N° Engines available in train set	100	260	100	249
N° Engines available in test set	100	259	100	248
Operating conditions applied	1	6	1	6
Fault modes	1	1	2	2
Training set size	20,632	53,760	24,721	61,250
Testing set size	13,097	33,992	16,597	41,215

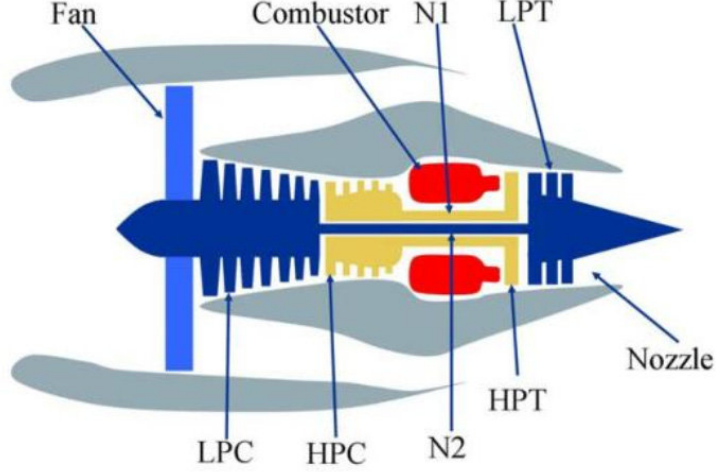


Fig. 3 Simplified C-MAPSS turbofan engine diagram [28]

According to the data description in Figure 2, units FD002 and FD004 exceptionally contain 6 operating conditions with more than 200 tracking trajectories. Asif et al. demonstrated in [23] that the two subunits possess irregular behavior, making the prediction of RUL for these complex sequences a challenging task. As a result, most works in the literature concentrate on the FD001 and FD003 datasets due to their simplicity and smaller data volume. Consequently, the FD004 dataset remains relatively underexplored in existing literature.

This paper presents preliminary findings for the subset FD004. In future a detailed study will be presented, in an upcoming journal paper, while considering all subsets of the C-MAPSS dataset

2.3 Remaining Useful Life (RUL) Generation

Each sub-dataset is composed of 26 columns in total, but the target RUL column is not included. The RUL is determined by calculating the difference between the maximum time (in cycle) for a given unit and the current time (in cycle). To achieve this, we proceeded as follows:

$$RUL_{\text{train}}(i) = \max(\text{Cycle})_{\text{per unit}}(i) - \text{Cycle}(i) \quad (1)$$

$$RUL_{\text{test}}(i) = \max(\text{Cycle})_{\text{per unit}}(i) + RUL \text{ given}(i) - \text{Cycle}(i) \quad (2)$$

where:

- $RUL_{\text{train}}(i)$ and $RUL_{\text{test}}(i)$ are the RUL for the train and test sets respectively,
- $\max(\text{Cycle})_{\text{per unit}}(i)$ is the maximum number of cycles for each unit i ,
- $\text{Cycle}(i)$ is the current cycle number,
- $RUL \text{ given}(i)$ is the given cycles for each unit i in the RUL file.

2.4 Min-Max Normalization

Min-Max Normalization, also known as feature scaling, is a simple method used in data preprocessing to scale numerical data. The goal of Min-Max Normalization is to transform features to fall within a specific range, $[0,1]$ in our case, thus ensuring that no particular feature dominates others during the learning process, which can be crucial for some MLg algorithms [23]. The transformation is performed according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3)$$

where:

- x is the current value that needs to be normalized.,
- x' is the normalized value,
- $\min(x)$ and $\max(x)$ correspond to the minimum and the maximum values in the dataset, respectively.

2.5 Random-Sampling Based Class Balancing

The performance of ML algorithms can be affected because of the ample imbalances in the intended classes [29]. This problem can be resolved to a certain extent using the subsampling, oversampling or reweighting approaches [30]. In literature, it is shown that the class balancing can raise the performance of contemporary classification algorithms in terms of accuracy and latency [31], [32].

In this study, we have devised a random sub-sampling approach for class balancing. It resolves the problem of data misbalancing while diminishing the impact of biasing. Moreover, it also diminishes the computational load and latency of the considered classifiers during the training and testing phases and allows to compute empirical results using affordable personal computers.

The approach is based on the uniformly distributed random indexes generation. Let $C = \{C_1, C_2, \dots, C_K\}$ is a set of intended classes, where K is presenting the considered count of classes. Let the count of instances for these classes is defined by the set $M = \{M_1, M_2, \dots, M_K\}$ and their corresponding indexes ranges are defined as: $R = \{R_1, R_2, \dots, R_K\}$. Where, elements of sets M and R belongs to the set of positive integers. Then using the inversion method, based on the idea that ‘‘Continuous Cumulative Distribution Functions’’ (CDFS) have a uniform distribution over the open interval $[0,1]$, the set of uniformly distributed random indexes $I_1 = \{n_1, n_2, \dots, n_N\}$ is generated for C_1 [33]. A similar process is followed for the generation of I_2, I_3, \dots, I_K which are sets of the random indexes respectively for the C_2, C_3, \dots, C_K .

The generation process of I_1 is achieved while respecting the corresponding index range, R_1 . In this scenario, If x is a uniform random number on $[0,1]$, then a random index i_m , from the standard normal distribution and the specified CDF, F , is generated as: $i_m = F^{-1}(x)$ [33]. Where m ranges from $[1, N]$. In next step, the value of the generated i_m is adapted as per R_1 , the process is given as: $n_m = \text{OS} + \text{round}(i_m \times \text{SF})$, where SF is the scaling factor and OS is the off-set. The values of SF and OS can be specific as a function of the intended range from the set R . Also, for an effective class balancing, the value of N should be less than or equal to the minimum element in the set M . In this study, the value of $N_{\text{Training}} = 50$ is chosen and the value of $N_{\text{Testing}} = 25$ is selected.

Algorithm 1: Random-Sampling Based Class Balancing Algorithm - RSBCB

Input : Set of intended classes $C = \{C_1, C_2, \dots, C_K\}$;
Count of instances for each class $M = \{M_1, M_2, \dots, M_K\}$;
Index ranges for each class $R = \{R_1, R_2, \dots, R_K\}$;
Scaling factor SF ;
Offset OS ;
Number of training samples N_{Training} ;
Number of testing samples N_{Testing} ;
Output: Randomly sampled dataset for class balancing

```
1 for each class  $C_i$  in  $C$  do
2   | Generate uniformly distributed random indexes  $I_i = \{n_1, n_2, \dots, n_N\}$  for  $C_i$  within the index
   | range  $R_i$ ;
3   for each index  $n_m$  in  $I_i$  do
4     | Generate a random number  $x$  as a uniform random number on  $[0, 1]$ ;
5     | Generate a random index  $i_m$  from the standard normal distribution and the specified CDF
   |  $F$  using the inverse method:  $i_m = F^{-1}(x)$ ;
6     | Adapt the value of  $i_m$  as per the index range  $R_i$ :  $n_m = OS + \text{round}(i_m \times SF)$ ;
7   end
8   | Select a random sample of  $N_{\text{Training}}$  instances from class  $C_i$  using the generated indexes  $I_i$ ;
9   | Store the selected instances in the training dataset;
10 end
11 for each class  $C_i$  in  $C$  do
12   | Select a random sample of  $N_{\text{Testing}}$  instances from each class  $C_i$  using the generated indexes  $I_i$ ;
13   | Store the selected instances in the testing dataset;
14 end
```

2.6 Boosting based Regression

Designed to have specific properties such as efficiency, flexibility and portability. The XGBoost (eXtreme Gradient Boosting) Regressor is an optimized distributed gradient boosting library. By implementing ML algorithms under the Gradient Boosting framework, it provides a parallel tree boosting (commonly known GBDT or GBM) allowing to solve many data science problems quickly and precisely [34]. The core XGBoost model is an ensemble of decision trees, where new trees are added to correct the errors made by existing trees. The trees are added sequentially until no further improvements can be made.

Given a set of n instances $(x_1, y_1), \dots, (x_n, y_n)$ where $x_i \in \mathbb{R}^m$ is a feature vector and $y_i \in \mathbb{R}$ is the corresponding label, XGBoost learns a function $f : \mathbb{R}^m \rightarrow \mathbb{R}$ that maps a feature vector to a label. The function f is a series of K decision trees.

$$f(x) = \sum_{k=1}^K f_k(x) \quad (4)$$

where f_k is a decision tree.

The objective of XGBoost is to minimize the following loss function:

$$L(y, f(x)) = \sum_{i=1}^n l(y_i, f(x_i)) + \sum_{k=1}^K \Omega(f_k) \quad (5)$$

where l is a differentiable convex loss function that measures the difference between the prediction $f(x_i)$ and the correct label y_i , and Ω is a regularization term that penalizes the model complexity.

XGBoost offers several advantages as a fast and accurate method, but it also includes several regularizations which help to reduce overfitting. Moreover, it supports missing values, allows for parallelizable processing and provides several flexible features for user customization. Finally, several applications are considered: regression solving, ranking, classification, user-defined prediction problems, etc.

Algorithm 2: XGBoost Regressor Algorithm

```
1 Function my_XGB_Regressor( $x, y$ ):  
   Input : Training data  $x$ , Testing data  $y$   
   Output: None  
2   mse_test_list  $\leftarrow$  [],  
3   rmse_test_list  $\leftarrow$  [],  
4   mae_test_list  $\leftarrow$  []  
5   for  $j$  in range(1, 11) do  
6     normalized_train_df  $\leftarrow$  normalised_df( $x$ )  
7     normalized_test_df  $\leftarrow$  normalised_df( $y$ )  
8     train_selected  $\leftarrow$  Class_Balancing_RSBCB(normalized_train_df,  $N_{Training}$ ,  $j$ )  
9     test_selected  $\leftarrow$  Class_Balancing_RSBCB(normalized_test_df,  $N_{Testing}$ ,  $j$ )  
10    X_train, Y_train, X_test, Y_test  $\leftarrow$  data_split(train_selected, test_selected)  
    // Create the XGBoost Regressor model  
11    model  $\leftarrow$  XGBRegressor( $n\_estimators=100$ ,  $max\_depth=3$ ,  $learning\_rate=0.01$ ,  
     $subsample=0.5$ ,  $colsample\_bytree=0.5$ )  
12    model.fit(X_train, Y_train)  
    // Test  
13    y_pred_test  $\leftarrow$  model.predict(X_test)  
14    mse_test  $\leftarrow$  mean_squared_error(Y_test, y_pred_test)  
15    rmse_test  $\leftarrow$  sqrt(mse_test)  
16    mae_test  $\leftarrow$  mean_absolute_error(Y_test, y_pred_test)  
17    mse_test_list.append(mse_test)  
18    rmse_test_list.append(rmse_test)  
19    mae_test_list.append(mae_test)  
    // Print results: mae_test, rmse_test  
20  end  
21  mean_mae  $\leftarrow$  mean(mae_test_list)  
22  mean_rmse  $\leftarrow$  mean(rmse_test_list)  
    // Print Mean MAE and Mean RMSE  
23  print("Mean MAE:", mean_mae)  
24  print("Mean RMSE:", mean_rmse)
```

Following the RUL generation and data normalization, the next stage is to proceed to ten iterations. During each iteration, a random sampling-based class balancing operation is performed. Subsequently, the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) are calculated to evaluate the performance of the model. These two error metrics are then stored and accumulated throughout the 10 iterations. At the end of the 10 iterations, the algorithm computes the average of the 10 MAE and RMSE values. These averages represent the global error rates of our model. In simpler terms, the algorithm performs 10 learning and evaluation experiments, and it utilizes the outcomes to deliver an overall performance evaluation of the model.

2.7 Evaluation Measures (RMSE and MAE):

In the domain of ML, a metric refers to a function employed to quantify the performance of a model or algorithm. In essence, metrics are utilized to measure how accurately a model's predictions align with the actual data [4]. There are several metrics divided into two categories based on the type of problem: classification problems and regression problems. In the context of regression, two of the most commonly used metrics are the MAE (Mean Absolute Error) and the RMSE (Root Mean Squared Error). Both MAE and RMSE express average model prediction error in units of the variable of interest; these two scores are said to be negatively-oriented, meaning that lower values are better.

1. **Mean Absolute Error (MAE)**: measures the average magnitude of the errors in a set of predictions, regardless of their direction. It can be considered as the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight as shown in Formula (6). MAE is less sensitive to outliers compared to RMSE [35].
2. **Root Mean Squared Error (RMSE)**: is a quadratic scoring rule that also measures the average magnitude of the error. It is based on the square root of the average of squared differences between prediction and actual observation. The RMSE gives a relatively high weight to large errors because the squared large difference of prediction and the actual observation leads to even larger contributions to the mean as shown in Formula (7). The effect is to pay more attention to large errors [35].

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

where:

- y_i is the actual RUL value of the aircraft engine,
- \hat{y}_i is the predicted RUL value by the model,
- n is the total number of observations.

3 Results & Discussion

In this work, two regression models namely, the KNN and XGBoost are considered for processing the subset FD004 – The KNN Regressor algorithm is well detailed in the article [36] – In our study, the work carried out with the KNN allowed us to find the most optimal parameters described as follows: best number of neighbors 46 and the distance metric as Manhattan distance in the 5th iteration out of a total of 10 tuning iterations with the class balancing. The performance of the considered regressors is evaluated for two different scenarios. In first case, the intended dataset is processed only after normalization. However, in the second case, the intended dataset is normalized and onward the class balancing is performed using the devised random-sampling based approach. The findings are outlined in Table 3.

In the first case the XGBoost regressor attains 19.22% RMSE and 14.48% MAE values respectively. KNN regressor performs better than the XGBoost regressor in this scenario by securing 15.71% RMSE and 11.58% MAE values respectively. In the second case the KNN regressor attains 13.10% RMSE and 09.55% MAE values respectively. XGBoost regressor outperforms by securing 12.88% RMSE value. However, the MAE value of the KNN regressor is 0.52% lesser than the MAE score achieved with the XGBoost regressor.

These results confirm the benefit of using random-sampling based class balancing on the normalized data before conveying it to the regression stage. For both considered regressors the accuracy is improved for the aforementioned second case. In the case of KNN regressor the respective error reduction in RMSE and MAE is 02.61% and 02.03%. In the case of XGBoost regressor the respective error reduction in RMSE and MAE is 06.34% and 04.41%.

Table 3 Evaluation Measures of Considered Regressors : KNN and XGBoost.

Method	FD004	
	RMSE	MAE
KNN Regressor + Normalization	15.71	11.58
KNN Reg + Normalization + Random-Sampling Based Class Balancing	13.10	9.55
XGBoost Reg + Normalization	19.22	14.48
XGBoost Reg + Normalization + Random-Sampling Based Class Balancing	12.88	10.07

In order to evaluate and compare the results of our study with literature works, we used the RMSE metric which is commonly used in the majority of studies conducted on the C-MAPSS dataset [13–24]. The results obtained with the tested methods are displayed in Table 3, they show that XGBoost Regressor model with normalization and Random-Sampling Based Class Balancing achieved the best result based on RMSE. Table 4 presents a comprehensive comparison of innovative and successful methods from the literature including our method. The table includes 13 different methods, their respective preprocessing steps, and a performance comparison based on their RMSE values.

Table 4 Performance comparison with state-of-the-art counterparts and preprocessing steps detailed.

N°	Methods	Author	Year	Normalization	Feature Selection	RUL Target Function	Piece-wise function	Variance threshold	Health indicator	Correlation analysis	Median move filter	STW Proc.	RUL Generation	RSBCB	RMSE
1	CNN	Sateesh Babu et al.[13]	2016	x		x									29.15
2	LSTM	Zheng et al.[14]	2017	x		x									28.17
3	BiLSTM	Wang et al.[15]	2018	x	x	x									24.86
4	DAG	Li et al.[16]	2019	x	x		x								22.43
5	CNN+LSTM	Kong et al.[17]	2019	x				x	x						23.25
6	Multi-head CNN+LSTM	Mo et al.[18]	2020		x	x									22.89
7	AGCNN	Liu et al.[19]	2020	x	x	x									21.15
8	Hybrid model	Kumar[20]	2021	x	x		x								22.32
9	BLS+TCN	Yu et al.[21]	2022												18.12
10	BiLSTM+Attention meth	Liu et al.[22]	2022			x									16.96
11	LSTM+Auto piecewise	Asif et al.[23]	2022	x		x	x			x	x				17.63
12	DRLRULE	Hu et al.[24]	2023	x	x							x			18.87
13	Proposed approach	Barry et al.	2023	x									x	x	12.88

Legend:

STW Proc.: Sliding Time Window Processing

RSBCB: Random Sampling Based Class Balancing

The methods include a range of techniques based on different types of neural networks (CNN, LSTM, BiLSTM), hybrid models, and other advanced ML techniques. Each method is accompanied by specific preprocessing steps such as normalization, feature selection, RUL target function, and others. The best RMSE from prior state-of-the-art methods is 16.96% using Bi-LSTM + Attention [22], while the worst RMSE is 29.15% from CNN [13]. Our proposed method achieves an RMSE of 12.88%, which is 4.08% lower than the best previous method and 16.27% lower than the worst previous method. This underlines the meaningful advancements in RUL prediction accuracy gained by the proposed method over current techniques. The results validate the efficacy of the proposed innovations in improving model performance.

4 Conclusion

In this paper, a novel data-driven predictive maintenance model is presented in the context of industry 4.0. The proposed method is based on a new fusion of XGBoost regressor, Min-Max normalization, Random-Sampling based class balancing, and Remaining Useful Life (RUL) generation. The performance of the intended regressors is assessed using the ground truth, computed by the rules-based RUL generation. The Min-Max normalization adjusts the data from several subjects in a common range by performing the linear transformation. An unbalanced presentation of the intended classes may result in biased findings. The uniformly distributed random sub-sampling is used to intelligently handle this problem. It not only diminishes the biasness in findings but also relaxes the requirements of computational resources. The performance of considered robust machine learning and ensemble learning algorithms is compared for predicting the RUL of the considered aircraft engine by processing the balanced dataset. The results have shown that the XGBoost regressor outperforms other considered models. The devised method secures the Root Mean Squared Error (RMSE) value of 12.88%. It is shown that the proposed approach secured a similar or better performance compared to the state-of-the-art counterparts. In this study, the performance of devised approach is evaluated only for the subset FD004. In future, a detailed analysis will be presented while considering all available subsets in the C-MAPSS dataset. Moreover, the incorporation of feature selection and deep learning approaches can enhance the performance of suggested solution. The feasibility of this concept will be investigated in future.

Abbreviations

AI	Artificial Intelligence
CBM	Condition-Based Maintenance
CNN	Convolutional Neural Networks
DL	Deep Learning
DM	Data Mining
IoT	Internet of Things
KNN	K-Nearest Neighbor
XGBoost	eXtreme Gradient Boosting
LSTM	Long Short-Term Memory
ML	Machine Learning
PdM	Predictive Maintenance
PHM	Prognostics and Health Management
RUL	Remaining Useful Life

References

- [1] Hafsi, M., NoraHamour, Ouchani, S.: Predictive maintenance for smart industrial systems: A roadmap. *Procedia Computer Science* **220**, 645–650 (2023) <https://doi.org/10.1016/j.procs.2023.03.082> . The 6th International Conference on Emerging Data and Industry 4.0 (EDI40)
- [2] Vollert, S., Theissler, A.: Challenges of machine learning-based rul prognosis: A review on nasa’s c-mapss data set. (2021). <https://doi.org/10.1109/ETFA45728.2021.9613682>
- [3] Nunes, P., Santos, J., Rocha, E.: Challenges in predictive maintenance – a review. *CIRP Journal of Manufacturing Science and Technology* **40**, 53–67 (2023) <https://doi.org/10.1016/j.cirpj.2022.11.004>
- [4] Berghout, T., Benbouzid, M.: A systematic guide for predicting remaining useful life with machine learning. *Electronics* **11**(7), 1125 (2022)
- [5] Montero Jimenez, J.J., Schwartz, S., Vingerhoeds, R.A., Grabot, B., Salaün, M.: Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems* **56**, 539–557 (2020) <https://doi.org/10.1016/j.jmsy.2020.07.008>
- [6] Cardoso, D., Ferreira, L.: Application of predictive maintenance concepts using artificial intelligence tools. *Applied Sciences* **11**(1) (2021)
- [7] Jagatheesaperumal, S.K., Rahouti, M., Ahmad, K., Al-Fuqaha, A.I., Guizani, M.: The duo of artificial intelligence and big data for industry 4.0: Review of applications, techniques, challenges, and future research directions. *CoRR* **abs/2104.02425** (2021) [2104.02425](https://arxiv.org/abs/2104.02425)
- [8] Meng, H., Li, Y.-F.: A review on prognostics and health management (phm) methods of lithium-ion batteries. *Renewable and Sustainable Energy Reviews* **116**, 109405 (2019)
- [9] Pater, I., Reijns, A., Mitici, M.: Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics. *Reliability Engineering & System Safety* **221**, 108341 (2022)
- [10] Maulana, F., Starr, A., Ompusunggu, A.P.: Explainable data-driven method combined with bayesian filtering for remaining useful lifetime prediction of aircraft engines using nasa cmapss datasets. *Machines* **11**(2) (2023) <https://doi.org/10.3390/machines11020163>
- [11] Cao, Q., Zanni-Merk, C., Samet, A., Reich, C., Beuvron, F., Beckmann, A., Giannetti, C.: Kspmi: A knowledge-based system for predictive maintenance in industry 4.0. *Robotics and Computer-Integrated Manufacturing* **74**, 102281 (2022) <https://doi.org/10.1016/j.rcim.2021.102281>
- [12] Hagemeyer, S., Zeiler, P., Huber, M.: On the integration of fundamental knowledge about degradation processes into data-driven diagnostics and prognostics using theory-guided data science. *PHM Society European Conference* **7**, 156–165 (2022) <https://doi.org/10.36001/phme.2022.v7i1.3352>
- [13] Sateesh Babu, G., Zhao, P., Li, X.-L.: Deep convolutional neural network based regression approach for

- estimation of remaining useful life. In: Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, April 16-19, 2016, Proceedings, Part I 21, pp. 214–228 (2016). Springer
- [14] Zheng, S., Ristovski, K., Farahat, A., Gupta, C.: Long short-term memory network for remaining useful life estimation. In: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 88–95 (2017). IEEE
- [15] Wang, J., Wen, G., Yang, S., Liu, Y.: Remaining useful life estimation in prognostics using deep bidirectional lstm neural network. In: 2018 Prognostics and System Health Management Conference (PHM-Chongqing), pp. 1037–1042 (2018). IEEE
- [16] Li, J., Li, X., He, D.: A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction. *IEEE Access* **7**, 75464–75475 (2019)
- [17] Kong, Z., Cui, Y., Xia, Z., Lv, H.: Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Applied Sciences* **9**(19), 4156 (2019)
- [18] Mo, H., Lucca, F., Malacarne, J., Iacca, G.: Multi-head cnn-lstm with prediction error analysis for remaining useful life prediction. In: 2020 27th Conference of Open Innovations Association (FRUCT), pp. 164–171 (2020). IEEE
- [19] Liu, H., Liu, Z., Jia, W., Lin, X.: Remaining useful life prediction using a novel feature-attention-based end-to-end approach. *IEEE Transactions on Industrial Informatics* **17**(2), 1197–1207 (2020)
- [20] Kumar, K.D.: Remaining useful life prediction of aircraft engines using hybrid model based on artificial intelligence techniques. In: 2021 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 1–10 (2021). IEEE
- [21] Yu, K., Wang, D., Li, H.: A prediction model for remaining useful life of turbofan engines by fusing broad learning system and temporal convolutional network. In: 2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), pp. 137–142 (2021). IEEE
- [22] Liu, Y., Zhang, X., Guo, W., Bian, H., He, Y., Liu, Z.: Prediction of remaining useful life of turbofan engine based on optimized model. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1473–1477 (2021). IEEE
- [23] Asif, O., Haider, S.A., Naqvi, S.R., Zaki, J.F., Kwak, K.-S., Islam, S.R.: A deep learning model for remaining useful life prediction of aircraft turbofan engine on c-mapss dataset. *IEEE Access* **10**, 95425–95440 (2022)
- [24] Hu, Q., Zhao, Y., Wang, Y., Peng, P., Ren, L.: Remaining useful life estimation in prognostics using deep reinforcement learning. *IEEE Access* **11**, 32919–32934 (2023) <https://doi.org/10.1109/ACCESS.2023.3263196>
- [25] Hong, C.W., Lee, C., Lee, K., Ko, M.-S., Kim, D.E., Hur, K.: Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction. *Sensors* **20**(22), 6626 (2020)
- [26] Peng, C., Chen, Y., Chen, Q., Tang, Z., Li, L., Gui, W.: A remaining useful life prognosis of turbofan engine using temporal and spatial feature fusion. *Sensors* **21**(2), 418 (2021)
- [27] Li, Q., Yang, Y.: Transformer model for remaining useful life prediction of aeroengine. In: *Journal of Physics: Conference Series*, vol. 2171, p. 012072 (2022). IOP Publishing
- [28] Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management, pp. 1–9 (2008). IEEE
- [29] Idrissi, B.Y., Arjovsky, M., Pezeshki, M., Lopez-Paz, D.: Simple data balancing achieves competitive worst-group-accuracy. In: *Conference on Causal Learning and Reasoning*, pp. 336–351 (2022). PMLR
- [30] Sun, J., Li, H., Fujita, H., Fu, B., Ai, W.: Class-imbalanced dynamic financial distress prediction based

on adaboost-svm ensemble combined with smote and time weighting. *Information Fusion* **54**, 128–144 (2020)

- [31] Fotouhi, S., Asadi, S., Kattan, M.W.: A comprehensive data level analysis for cancer diagnosis on imbalanced data. *Journal of biomedical informatics* **90**, 103089 (2019)
- [32] Duan, M., Liu, D., Chen, X., Liu, R., Tan, Y., Liang, L.: Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems* **32**(1), 59–71 (2020)
- [33] Cheung, R.C., Lee, D.-U., Luk, W., Villasenor, J.D.: Hardware generation of arbitrary random number distributions from uniform distributions via the inversion method. *IEEE transactions on very large scale integration (VLSI) systems* **15**(8), 952–962 (2007)
- [34] Patel, S.K., Surve, J., Katkar, V., Parmar, J., Al-Zahrani, F.A., Ahmed, K., Bui, F.M.: Encoding and tuning of thz metasurface-based refractive index sensor with behavior prediction using xgboost regressor. *IEEE Access* **10**, 24797–24814 (2022) <https://doi.org/10.1109/ACCESS.2022.3154386>
- [35] Hodson, T.O.: Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not. *Geoscientific Model Development* **15**(14), 5481–5487 (2022)
- [36] Azadkia, M.: Optimal choice of k for k -nearest neighbor regression. *arXiv: Statistics Theory* (2019)