Remaining Useful Life Prediction of Equipment Based on XGBoost

Zhiyang Jia*
Department of Computer Science, China University of Petroleum-Beijing at Karamay, Karamay Xinjiang, China,
jiazhiyang@cupk.edu.cn

Zhibo Xiao
School of Science, Jiangnan University, Wuxi Jiangsu, China,
xiaozhibo2000@163.com

Yijin Shi
Department of Information, Lijiang Culture and Tourism College, Lijiang Yunnan, China,
shi_yi_jin@yeah.net

ABSTRACT

Remaining Useful Life (RUL) prediction is an essential task in the practice of predictive maintenance which aims at repairing equipment before it fails based on data received about it from sensors. Our simulation experiments use the Turbofan engine degradation dataset CMAPSS Data, which gained historical data to predict the remaining useful life and does not require participants to consider the underlying physical factors. RUL prediction is performed by machine learning methods including Decision Tree (DT), Random Forest (RF), Support Vector Regression (SVR), and XGBoost after data pre-processing and feature selection. XGboost is a kind of ensemble learning algorithm that can generate a series of weak learners by continuous training and then combine these weak learners to become a strong learner. Experimental results reveal that the performance of XGBoost based model is effective in such dataset comparing with the traditional machine learning models.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Machine learning approaches**; • **Classification and regression trees**;

KEYWORDS

Remaining useful life prediction, Machine learning, Predictive maintenance, Gradient boosting, XGboost

ACM Reference Format:
Zhiyang Jia*, Zhibo Xiao, and Yijin Shi. 2021. Remaining Useful Life Prediction of Equipment Based on XGBoost. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487075.3487134

1 INTRODUCTION

As one of the key innovations of Industry 4.0, predictive maintenance (Pdm) is a type of condition-based maintenance that monitors the condition of assets using sensor devices [1]. These sensor devices supply data in real-time, which is used to predict when the asset will be required maintenance and prevent equipment failure. Remaining Useful Life (RUL) [2] prediction is an essential topic in predictive maintenance. Data-driven and model-driven methods are traditionally adopted to predict RUL according to the data collected from sensors. With the help of RUL, engineers can schedule maintenance, optimize operational efficiency, and avoid unplanned downtime. Remaining useful life prediction plays a vital role in prognostics and health management (PHM) [3] for improving the reliability and reducing the cycle cost of numerous mechanical systems.

Some studies have predicted the RUL of equipment based on multiple operating stages. The ability to capture the wear and tear aging of equipment in different operating stages and calculate the RUL more in line with the actual operating conditions of the equipment can improve the accuracy of RUL prediction. Further, the possibility of overestimating the RUL can be effectively reduced. The continued use of the equipment after the RUL has been exceeded can be avoided, thus reducing the occurrence of catastrophic failures [4].

The RUL prediction model not only predicts but also provides confidence bounds for the prediction. The input terms to the model are state indicators, which are features extracted from sensor data or log data whose behavior changes with system degradation or changes in operating mode, and such changes are predictable.

Therefore, machine learning models are needed to perform the prediction of the RUL. Machine learning models can usually be performed in the following two ways: a classification approach, which predicts whether there is a possibility of failure in next n-steps, or a regression approach which predicts how much time is left before the next failure [5].

Although the classification method may have higher accuracy and efficiency, he regression models are mainly discussed in this paper.

Boosting [6] is an ensemble learning technique that uses a set of machine learning algorithms to convert weak learner to strong learners in order to increase the accuracy of the model. There are various types of ensemble algorithms, among which the Boosting algorithms obtain a series of weak learners by continuous training and then combine these base learners to become a strong learner. One of them is the extreme gradient boosting tree algorithm (XGBoost) [7, 8] proposed by Tianqi Chen et al. as a series of Boosting ensemble algorithm, which can combine numerous decision tree models and significantly improve the accuracy of model prediction results.

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487134

2 XGBOOST FOR TIME SERIES FORECASTING

XGBoost is an optimized distributed gradient boosting system designed to be highly efficient, flexible and portable [9]. XGBoost first generates CART trees, which are binary trees with two types of nodes: internal nodes and leaf nodes, where leaf nodes have no child nodes. If we assume that X and Y denote input and output variables, respectively, the training data set can be expressed as $D=\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$. If a constructed CART regression tree has K leaves, it means that CART divides the input space into K units: R_1, R_2, ... , R_K, and each R_K corresponds to a fixed output value c_k, traversing all the cut-off variables j and cut-off points s in turn. Find the optimal cut pair (j ,s) that minimizes the value of the following equation while slicing the sample space:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-C_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-C_2)^2\right] \quad (1)$$

Where c_1 and c_2 denote the output values after the dataset is divided into two parts. The mechanism of XGBoost algorithm is to continuously train new CART trees to fit the residuals calculated from the previous tree. And it needs to sum up the predicted values of all the decision trees to calculate the final output. and the result of the t training iteration is expressed as below:

$$\widehat{y}_i^{(t)}=\sum_{k=1}^{t}f_k(X_i)=\widehat{y}_i^{(t-1)}+f_t(X_i) \quad (2)$$

XGBoost is an additive model consisting of t base models (weak learners), and assuming that the tree model to be trained in at t iteration is $f_t(X_i)$, we have XGBoost controls over fitting by regularization, and its objective function is shown below:

$$\mathrm{Obj}^{(t)}=\sum_{i=1}^{n}l(y_i,\widehat{y}_i^{(t)})+\sum_{k=1}^{t}\Omega(f_k) \quad (3)$$

$$\Omega(f_k)=\lambda \mathrm{p_k}+\frac{1}{2}\lambda\|w_k\|^2 \quad (4)$$

The first term in the objective function $Obj^{(t)}$ is the loss function of the model, which can measure the difference between predicted values and true values, and it also reflects the ability of the model to fit the training data. The second term in the objective function $\Omega(f_k)$ is the complexity function of the model, where λ is the penalty factor of the leaf nodes, p_k is the total number of nodes in the tree, w_k is the weight value of the leaf nodes, and λ is the regularization factor of each leaf node weights.

The final objective function is derived by a second derivative Taylor expansion. In the loss function, g_i and h_i are the first derivative and second derivative gradient statistics. According to the following definition:

$$\Delta x=f_t(x_i) \quad (5)$$

$$f(x)=l(yi,\hat{y}_i^{(t-1)}) \quad (6)$$

$$g_i=\frac{\partial l(y_i,\hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \quad (7)$$

$$\mathrm{h}_i=\frac{\partial^2 l(y_i,\hat{y}_i^{(t-1)})}{\partial(\hat{y}_i^{(t-1)})^2} \quad (8)$$

The second order Taylor expansion is used to derive and simplify Equation 3) to conduce the final objective function.

$$Obj^{(t)}\approx\sum_{i=1}^{n}[l(y_i,\hat{y}_i^{(t-1)})+g_i f_t(x_i)+\frac{1}{2}h_i {f_t}^2(x_i)] +\Omega(f_t)+\text{constant} \quad (9)$$

Where $\mathrm{g_i}$ and $\mathrm{h_i}$ are the first derivatives and second derivatives of the loss function, respectively, the XGBoost algorithm will continuously train to produce new decision tree models and accumulate their predictions on the residuals, and finally, combine all the generated decision tree models.

This is the characteristic of XGboost, with this approximation; it is possible to define some loss functions (e.g., squared loss, logistic loss) by itself, as long as second derivability is guaranteed. Removing the constant term yields a simplified objective process for step t. The XGBoost algorithm will continuously train to generate new decision tree models and accumulate their predictions on the residuals; eventually combine all the generated weak learners.

If the loss function is defined as a squared loss function, the optimal objective function can be obtained by deriving the objective function

$$obj^*=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T \quad (10)$$

This formula is also known as the scoring function, which measures how fit the tree structure is, and a lower value indicate a better fit. Scoring function is the criteria used to choose a split point to build the CART tree. All the split points of the sample features are determined, and every single determined split point is scored, the criteria for good or bad split points are as follows

$$\mathrm{Gain}=\frac{1}{2}[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}]-\gamma \quad (11)$$

Gain denotes the difference between a single node obj^* and the tree obj^* of the two sub-nodes after splitting, traverse the splitting points of all features, search for the split point of the maximum Gain that is the best splitting point, and continue splitting nodes iteratively until the whole CART tree is generated.

3 RUL PREDICTION BASED ON XGBOOST

3.1 Experimental Data Set

The simulation experiments use the turbofan engine degradation dataset CMAPSS Data [10], which uses historical data to predict the remaining device life (RUL) and does not require participants to consider the underlying physical factors. The CMAPSS Data comes from the Turbofan engine, a modern gasoline turbine engine that is used by NASA's Space Exploration Agency. The data set includes a time series for each engine. All engines are of the same type, but the initial degree of wear and differences in the manufacturing process is different for each engine, which is unknown to the user. There are three optional configurations that can be used to vary the performance of each engine. Each engine has 21 sensors, which collect measurements related to the engine state when the engine is running. There is some sensor noise in the data collected. Gradually, each engine will have some deficiencies, which can be found in the sensor readings. The first column of the data set is numbered,

and each data set has one hundred engine damage processes. The second column is a time series of each process, where the training data set is the engine running until it is failed, while the testing data set has the last running time without failed and the remaining usage time is stored in the RUL set. The third to fifth columns are for three optional configurations. The sixth through twenty-six columns are twenty-one sensor variable data. When the engine works, every engine generates some faults, which can be detected in the sensor data.

3.2 Data Pre-processing

In engineering practice, the data obtained always have missing values, duplicate values, etc., and need to be pre-processed before training. There are three approaches to missing value handling: directly using features containing missing values; removing features containing missing values, which is valid when the attribute containing the missing values contains a large number of missing values and only a very small number of valid values; and missing value completion.

When exploring the sensor content, it will be found that the values vary greatly between different sensors, so this study does not take these data directly to the training. Suppose there are some sensors with large value expressions. In that case, it is likely that they will occupy a larger weight in the next model training, which will reduce the training accuracy of the model, so a scaling operation should be performed. In this study, the normalization process was chosen based on the basic situation. Therefore, the extracted feature data need to be normalized to facilitate the processing and operation of the model:

3.3 Feature Visualization

The feature visualization was drawn based on the sample set data. Line chart and distribution of variables in the dataset are shown in the Figure 1 and Figure 2

3.4 Feature Selection

When the data pre-processing is completed, meaningful features need to be chosen for the input of the machine learning algorithms for training. In general, two considerations are needed to feature selection: whether the feature is divergent or not: if a feature is not divergent, e.g., the variance is close to 0, which means that the samples basically do not differ in this feature, the feature is not useful for sample differentiation. Relevance of features to the target: This is more obvious, and features with high relevance to the target should be selected preferentially. Feature selection can be considered mainly from the data relevance perspective and the information gain perspective. The Pearson correlation coefficient is one of the simplest and helpful methods to understand the relationship between features and response variables, which measures the linear correlation between variables. Initially, feature selection was performed based on the Pearson correlation coefficient.

The dataset contains 26 variables: the first column is engine unit, the second column is the current cycle number, 3rd to 5th columns are the working conditions and 6th to 26th columns represent 21 raw sensor data [11]. However, some columns don't provide useful information for RUL estimation. Therefore, 14 of the 21 sensors

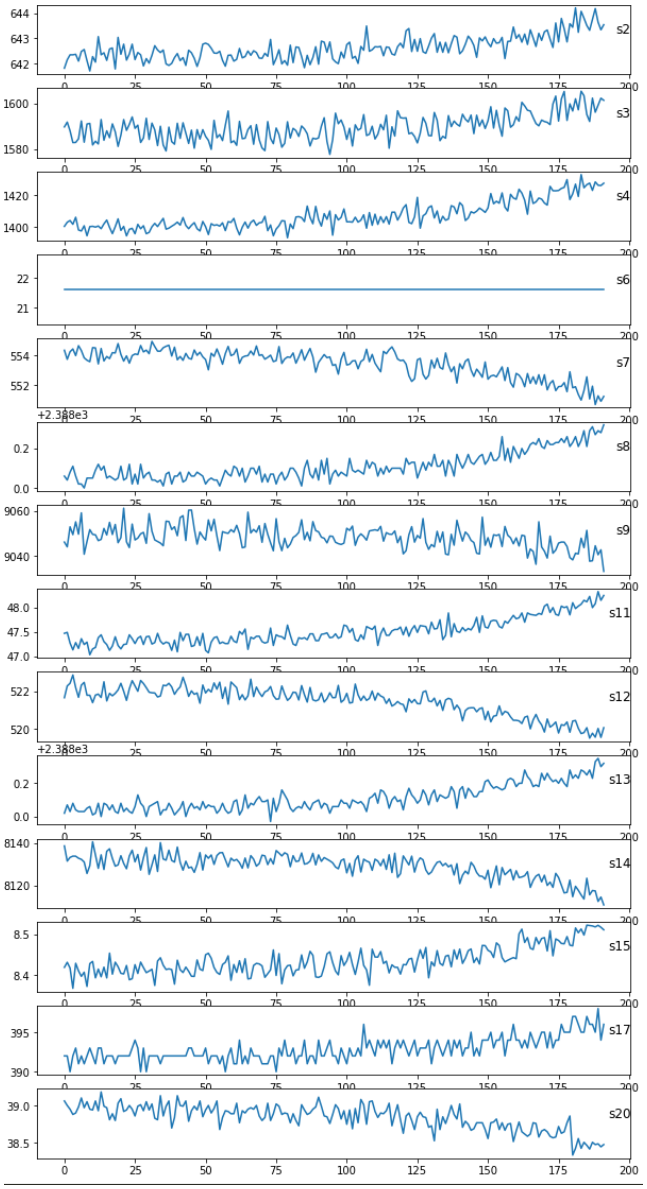

Figure 1: Line Chart of Variables IN The Dataset.

data are selected as the input feature to the model [12]. Pearson's Correlation method is used for finding the association between the continuous features and the class feature. The selected column numbers of variables are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21 based on the Pearson's Correlation method.

3.5 Traditional Machine Learning Models

For the task of RUL prediction of equipment, various supervised machine learning algorithms are available. In order to improve the generalization performance of machine learning models, an optimal algorithmic model should be chosen as the final choice

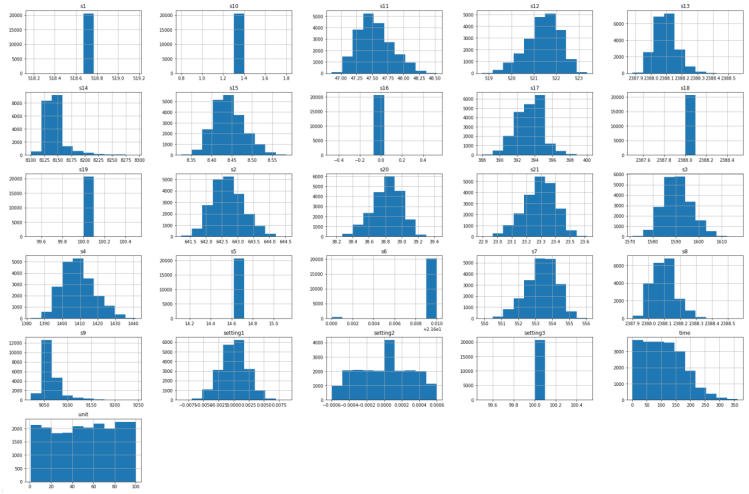

Figure 2: Distribution of Variables IN The Dataset.

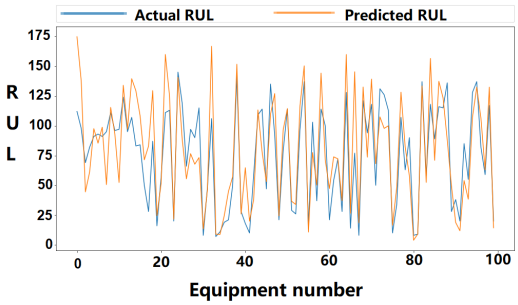

Figure 3: Result Visualization Performed by Decision Tree.

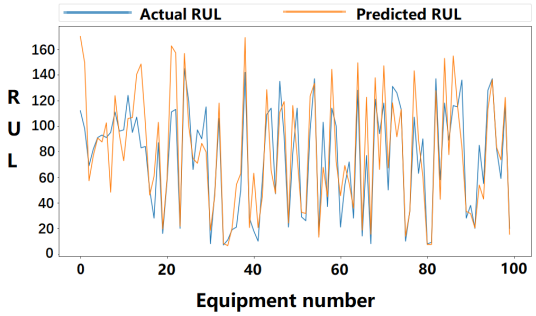

Figure 4: Result Visualization Performed by Random Forest.

by comparative experiment after data pre-processing and feature engineering.

Comparative experiments were performed on the same dataset based on several promising machine learning models, including Decision Tree (DT), Random Forest (RF), Support Vector Regression (SVM), and XGBoost.

The regression decision tree mainly refers to the CART (classification and regression tree) algorithm, where the internal node features take the values of "yes" and "no" and is a binary tree structure. The visualization comparison of the predicted RUL and the actual RUL is shown in the following Figure 3

A random forest is a forest that is constructed randomly, and this forest is composed of many mutually unrelated decision trees. A single prediction problem is solved by building a combination of n models. The visualization comparison of the predicted RUL and the actual RUL is shown in the Figure 4

The method of support vector classification can be generalized to solve regression problems, called support vector regression. The model generated by support vector classification relies only on a subset of the training data since the cost function for creating the model does not consider training points beyond the boundaries. Similarly, the model generated by support vector regression relies

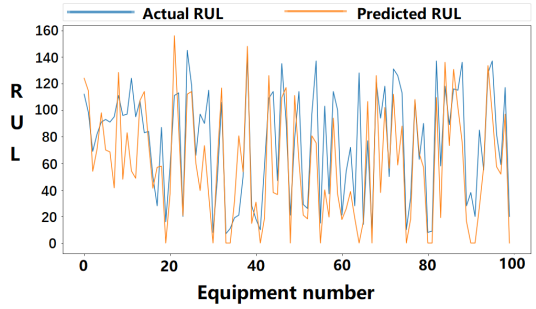

Figure 5: Result Visualization Performed by SVR.

only on a subset of the training data since the cost function for creating the model ignores any training data that is close to the model prediction. A visual comparison of the experimentally derived predicted RUL, as well as the actual RUL, is shown in the Figure 5

It can be observed that decision tree model is more effective on average, but has limited efficiency.

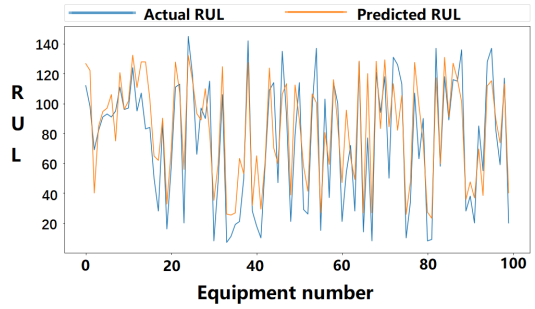

Figure 6: Visualization Comparison of Result Performed by XGBoost.

Table 1: Optimized Values of Hyperparameters Adopted

Hyperparameters	Value
n_estimators	20
learning_rate	0.10
max_depth	9
min_child_weight	5.0
scale_pos_weight	0.99
subsample	0.7
colsample_bytree	0.6
gamma	0.55

3.6 XGBoost Based Model

The XGBoost model has many hyperparameters, as listed bellows [13]:

1. n_estimators: The maximum number of iterations of weak learning machines, or the maximum number of weak learning machines.
2. learning_rate: step size shrinkage used to prevent over fitting. Range is [0,1].
3. max_depth: the maximum depth of the tree.
4. min_child_weight: Determine the minimum leaf node sample weights sum.
5. scale_pos_weight: When the sample is very unbalanced. Setting this parameter to a positive number can make the algorithm converge faster.
6. Subsample: percentage of samples used per tree.
7. colsample_bytree: percentage of features used per tree.
8. Gamma: controls whether a given node will split based on the expected reduction in loss after the split.

Optimized values of hyperparameters of XGBoost model adopted in the experiment are shown in Table 1. The visualization comparison of the predicted RUL and the actual RUL is shown in Figure 6. RUL prediction for the No. 1 unit of turbofan engine in the dataset based on XGBoost is shown in Figure 7

3.7 Evaluation

While predicting RUL, the goal is to reduce the error between the actual RUL and the predicted RUL. We will use Root Mean Squared Error since it penalizes large errors severely, which will force the algorithm to forecast RUL as close as possible [14].

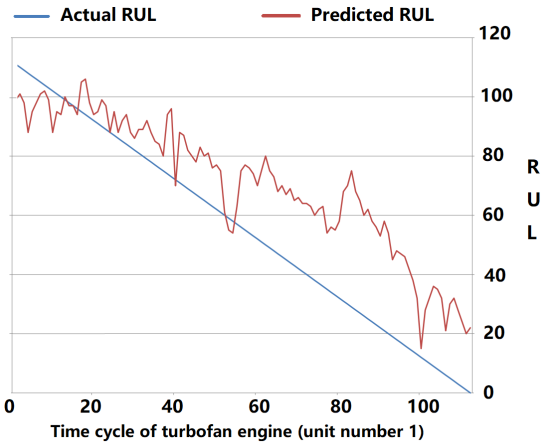

Figure 7: RUL Prediction for the No. 1 Unit of Turbofan Engine in the Dataset Based on XGBoost.

Table 2: RMSE and Time Consuming of the Experiment

Model	RMSE	Time Consuming (millisecond)
Decision tree	25.2	147
Random forest	23.8	8464
XGBoost	20.9	217
SVR	32.7	5173

According to the result of the experiment shown in Table 2, we can conclude that the model based on XGBoost has the best accuracy and the highest efficiency. It has a significantly better performance compared to the RF and SVR.

4 CONCLUSION

In this paper, Extreme Gradient Boosting (XGBoost) based equipment RUL prediction model is constructed. Experiments show that the model is better than traditional machine learning models, such as decision tree, random forest, and support vector machine models [15]. At the same time, the model has higher prediction accuracy than the traditional machine learning model, each error indicator is small, and the model training time consuming is shorter, which meets the timeliness of predictive maintenance requirements and has a certain application value. It should be noted that this paper does not consider the characteristics of time-series correlated data [16]. The next stage still needs to study the prediction for time-series correlated data and further conduct time-series multi-timescale analysis on the data, such as using Long Short-Term Memory (LSTM) or other Convolutional Neural Networks (CNN) models [17] to further enhance the model robustness and make it suitable for predictive maintenance of equipment in complex situations [18].

ACKNOWLEDGMENTS

This work is supported by Research Foundation of China University of Petroleum-Beijing at Karamay (No. XQZX20200006).

REFERENCES

[1] Sahba, R. , Radfar, R. , Ghatari, A. R. , & Ebrahimi, A. P. (2021). Development of industry 4.0 predictive maintenance architecture for broadcasting chain. Advanced Engineering Informatics, 49(December 2019), 101324.

[2] Lei, Y. , Li, N. , Gontarz, S. , Jing, L. , Radkowski, S. , & Dybala, J.. (2016). A model-based method for remaining useful life prediction of machinery. IEEE Transactions on Reliability, 65(3), 1314-1326..

[3] Li, Z. , Goebel, K. , & Wu, D. . (2019). Degradation modeling and remaining useful life prediction of aircraft engines using ensemble learning. Journal of Engineering for Gas Turbines and Power, 141(4), 041008.1-041008.10.

[4] Ge, C. , Zhu, Y. , & Di, Y. . (2017). Equipment remaining useful life prediction oriented symbiotic simulation driven by real-time degradation data. International Journal of Modeling, Simulation, and Scientific Computing, 1850009.

[5] V. Mathew, T. Toby, V. Singh, B. M. Rao and M. G. Kumar (2017). Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning," 2017 IEEE International Conference on Circuits and Systems (ICCS), pp. 306-311, doi: 10.1109/ICCS1.2017.8326010.

[6] RE Sphere. (2002). The boosting approach to machine learning an overview. Proc Msri Workshop on Nonlinear Estimation & Classification.

[7] Chen, T. , Tong, H. , & Benesty, M. . (2016). Xgboost: extreme gradient boosting.

[8] Dalzochio, J. , Kunst, R. , Pignaton, E. , Binotto, A. , & Barbosa, J. . (2020). Machine learning and reasoning for predictive maintenance in industry 4.0: current status and challenges. Computers in Industry, 123, 103298.

[9] Gao, C. , Sun, H. , Wang, T. , Tang, M. , Bohnen, N. I. , & Mltm, M. , *et al.* (2018). Model-based and model-free machine learning techniques for diagnostic prediction and classification of clinical outcomes in parkinson's disease. Scientific Reports, 8(1), 7129.

[10] A. Saxena and K. Goebel (2008). "Turbofan Engine Degradation Simulation Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

[11] Jiubin Zhao, Yuanxue Liu, Ming Hu (2018). Optimisation algorithm for decision trees and the prediction of horizon displacement of landslides monitoring. The 2nd 2018 Asian Conference on Artificial Intelligence Technology (ACAIT 2018). https://doi.org/10.1049/joe.2018.8305

[12] E. Ramasso and A. Saxena (2014). Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets," International Journal of Prognostics and Health Management, vol. 5, no. 2, pp. 1-15, 2014.

[13] Long, H. V., Son, L. H., Khari, M., Arora, K., Chopra, S., Kumar, R.& Baik, S. W. (2019). A new approach for construction of geodemographic segmentation model and prediction analysis. Computational intelligence and neuroscience, 2019.

[14] Kumar, P., Dayal, M., Khari, M., Fenza, G., & Gallo, M. (2021). NSL-BP: A Meta Classifier Model Based Prediction of Amazon Product Reviews. International Journal of Interactive Multimedia & Artificial Intelligence, 6(6).

[15] Pereira, F. , Mitchell, T. , & Botvinick, M. . (2009). Machine learning classifiers and fmri: a tutorial overview. Neuroimage, 45(1), S199-S209.

[16] Kang, Z. , Catal, C. , & Tekinerdogan, B. . (2021). Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks. Sensors, 21(3), 932.

[17] X. Li, Q. Ding, and J. Q. Sun (2018). Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks," Reliability Engineering & System Safety, vol. 172.

[18] K. Liu, N. Z. Gebraeel and J. Shi (2013). A Data-Level Fusion Model for Developing Composite Health Indices for Degradation Modeling and Prognostic Analysis," in IEEE Transactions on Automation Science and Engineering, vol. 10, no. 3, pp. 652-664, July 2013, doi: 10.1109/TASE.2013.2250282.