Machine Learning Enabled Estimation of Remaining Useful Life for Turbofan Engine using NASA CMAPSS Dataset

Arka Bera
Department of Mechanical Engineering
University of Maryland, College Park
College Park, USA
arkabera@umd.edu

***Abstract*—This study presents two machine learning models for evaluating the remaining useful life (RUL) of turbofan engines simulated with CMAPSS. Preprocessing is performed on multivariate time series data before training the models. For RUL prediction, we use a Random Forest Regressor and a 1D Convolutional Neural Network (CNN). In addition, a hybrid CNN-LSTM model is used on one of the datasets, offering a comprehensive method for RUL estimation in engine prognostics.**

I. Introduction

In the field of engineering and industrial maintenance, the ability to accurately predict the Remaining Useful Life (RUL) of machinery is of great importance. Predictive maintenance strategies, enabled by advanced data analytics and machine learning techniques, offer a proactive approach to equipment maintenance, helping to minimize downtime, reduce maintenance costs, and optimize asset utilization [1], [2]. Predictive maintenance is especially important in industries like aircraft, where advanced understanding of failure probability can save not just money but also lives. In this project, the focus is on developing machine learning models to predict the RUL of aircraft engines based on multivariate time-series data.

Engines, being complex mechanical systems, are subject to gradual degradation over time. This can be due to various factors such as wear and tear, operational conditions, and environmental factors [3]. Monitoring the health and performance of engines in real-time through sensor measurements and operational data provides valuable insights into their condition and potential failure modes [4]. By leveraging machine learning algorithms, it is possible to harness these data streams to build accurate and reliable models capable of forecasting the RUL of engines [5].

Throughout this project, different modeling approaches are explored, such as traditional machine learning algorithms like Random Forest Regressor [6] to more advanced deep learning architectures like Convolutional Neural Networks (CNN) [7] and hybrid CNN-LSTM models [8]. These models are trained and evaluated on datasets containing sensor measurements and operational settings of engines, with the objective of predicting the RUL before system failure.

The remaining paper is organized as: Section II gives a complete overview of the Preprocessing that was done for the algorithm. Section III provides details on the Experiments and Results obtained while Section IV concludes the paper.

II. Preprocessing

A. CMAPSS

CMAPSS stands for Commercial Modular Aero-Propulsion System Simulation [10]. It is an intricate system used to accurately simulate the performance of large commercial turbofan engines. Designed for use with MATLAB and Simulink, it has a number of customizable input configurations. Users can provide particular parameters for operational profiles, closed-loop controllers, and environmental conditions.

This simulation software is designed for turbofan engines with a thrust of up to 90,000 pounds. It includes an atmospheric model that can simulate a variety of situations, such as elevations (from sea level to 40,000 feet), Mach numbers (0 to 0.90), and sea-level temperatures (-60 to 103 degrees Fahrenheit). Moreover, CMAPSS includes a power management system, enabling users to operate the engine across a wide range of thrust levels under diverse flight conditions.

The built-in control system includes a fan-speed controller, as well as a set of regulators and limiters. These components ensure that the engine functions within its design parameters and delivers peak performance. For example, there are three high-limit regulators that keep the engine from exceeding its core speed, engine-pressure ratio, and High-Pressure Turbine (HPT) exit temperature limits. A limit regulator prevents the static pressure at the High-Pressure Compressor (HPC) outlet from falling too low, while an acceleration and deceleration limiter regulates core speed variations.

To ensure stability and avoid concerns like integrator windup, a comprehensive logic framework integrates these control system pieces, mimicking the operation of real-world engine controllers. Furthermore, the gains for the fan-speed controller and four limit regulators are precisely planned to provide optimal performance under all flying situations and power levels. The engine diagram in Fig. 1 represents the model's core components, whereas the flow chart in Fig. 2

depicts how the simulation's many subroutines are organized [4].

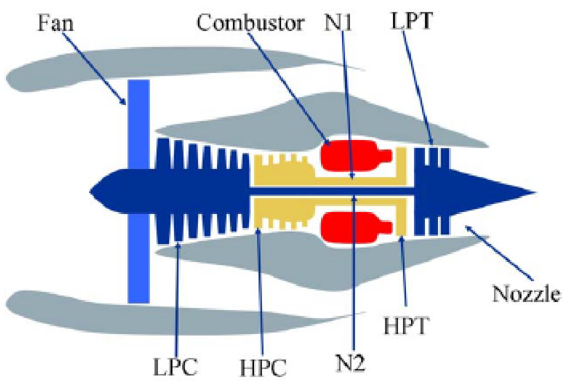

Fig. 1. Simplified diagram of engine simulated in C-MAPSS [4]

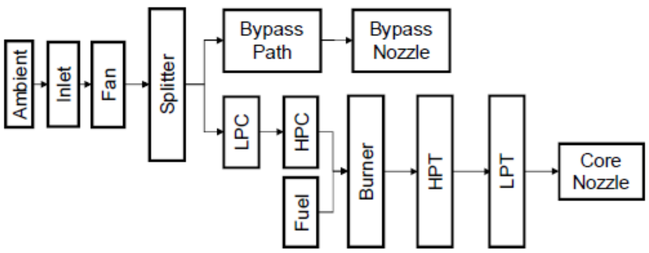

Fig. 2. Schematic showing various modules and their connections [4]

B. Simulated Data

The data sets are made up of several multivariate time series, each of which represents the operational behavior of a fleet of identical engine types. These engines show natural wear and manufacture variations, which are intrinsic and not considered fault events. Each engine's operational data is separated into training and testing subsets.

Within each time series, the engines begin operation under normal conditions before developing faults at some point. In the training set, these defects escalate until system failure, whereas in the test set, the time series terminates before system failure. It's worth noting that the number of cycles from the start for each engine in the training set represents the Remaining Useful Life (RUL) for that particular engine. The primary goal is to predict the remaining operational cycles before failure in the test set, which is the number of operational cycles remaining after the last observed cycle. A description of the provided dataset is shown in Table I.

C. Data preparation

In data preparation for this problem, the initial columns, such as engine ID and time, along with the three operational settings columns, are excluded from model training and evaluation. These columns provide metadata about the engines and their operating conditions but do not directly contribute to

TABLE I
DATA DESCRIPTION

Data Set	Training Data	Testing Data	Operating Conditions	Fault Modes
FD001	100	100	1	1
FD002	260	259	6	1
FD003	100	100	1	2
FD004	248	249	6	2

predicting engine failures. Therefore, they are omitted to focus solely on the sensor measurements that reflect the health and performance of the engines over time. In addition, some of the sensor measurements have constant values and they are also discarded. Therefore, 14 sensor measurements out of the 21 sensors are used as raw input features, whose indices are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21 [11].

To normalize the sensor measurements, the Min-Max Scaling technique is applied to rescale the data to the range [-1, 1]. This normalization method ensures uniformity and stability in model training. The Min-Max Scaling formula is as follows:

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

Here, X represents the original data point, X_{min} is the minimum value in the dataset, and X_{max} is the maximum value in the dataset.

The Remaining Useful Life (RUL) for each engine is determined based on the number of cycles in the training dataset. The RUL is obtained from the number of remaining operational cycles before an engine fails. For the training data, an engine can be considered to fail at the end of each operating cycle. The total number of cycles for each engine gives the RUL for the respective engine. The test data contains random number of engine cycles, and the task is to predict the number of cycles, or RUL, before the engine fails.

In addition to RUL calculation, piecewise linear degradation model with early RUL set at 130 [11], [12] is implemented, as shown in Fig. 3.

For the different machine learning models, windows of variable sizes are used depending on the specific dataset. These windows serve as temporal segments of the time series data, allowing the models to capture temporal patterns effectively. By sliding these windows across the time series data, the models can learn from historical patterns and make predictions based on recent observations within each window. This approach enables the models to capture the temporal dependencies and interactions among different variables, thereby improving their predictive performance for engine health prognosis.

III. EXPERIMENTS AND RESULTS

The pre-processed data is utilized to train two different models: a Random Forest Regressor, serving as the baseline model, and a Convolutional Neural Network (CNN) model. Different window sizes are employed for each dataset: 30 for FD001, 20 for FD002, 30 for FD003, and 15 for FD004.

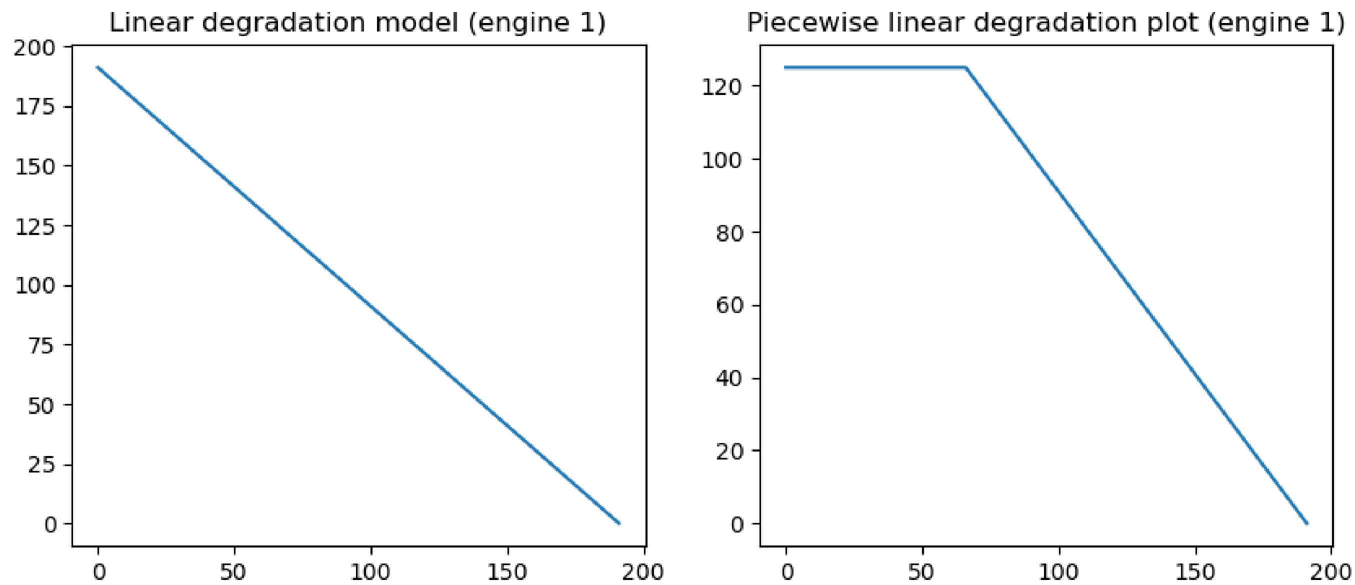

Fig. 3. Piecewise linear degradation model

In the prediction process, the Remaining Useful Life (RUL) for each engine is predicted from the last observed cycle.

Two performance metrics commonly reported in the literature for evaluating the models' predictive performance are Root Mean Squared Error (RMSE) and a scoring parameter [11]. RMSE measures the average magnitude of the errors between predicted and actual values, calculated as the square root of the average squared differences:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2}$$

where N is the number of data points, y_i is the actual value, and $\hat{y}_i$ is the predicted value.

The scoring parameter, denoted as s, is defined as:

$$s = \sum_{i=1}^{N} s_i \tag{3}$$

where s_i is calculated as follows:

$$s_i = \begin{cases} e^{-d_i/13} - 1, & \text{for } d_i < 0 \\ e^{d_i/10} - 1, & \text{for } d_i \geq 0 \end{cases}$$

Here, d_i represents the difference between the predicted and actual RUL for the i-th data point. The scoring parameter aggregates these penalties over all data points to assess the overall model performance. The scoring function penalizes late prediction more than early prediction, as late prediction could lead to severe consequences in aerospace applications.

A. Random Forest Regression

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training. Each tree in the forest is built independently using a random subset of the features and a random subset of the training data. The final prediction is then made by averaging the predictions of all the individual trees.

The Random Forest Regressor offers several advantages, including robustness to overfitting, flexibility in handling data, and the ability to capture non-linear relationships between input features and target variables. For the Random Forest model, the data is not scaled and the model is allowed to train on the original dataset.

Hyperparameter tuning is performed using GridSearchCV as listed in Table II, which exhaustively searches through a specified grid of hyperparameters to find the optimal combination that minimizes the negative root mean squared error (RMSE). The grid of hyperparameters includes the number of estimators (i.e., the number of trees in the forest) and the maximum number of features to consider when splitting a node. The negative RMSE is used as the scoring metric, indicating that lower values indicate better model performance.

TABLE II
HYPERPARAMETER TUNING FOR RF REGRESSOR

Hyperparameter	**Values considered**
n_estimators	100, 200, 300, 400, 500, 1000
max_features	sqrt, log2

The Random Forest Regression model forms the baseline with which the other models in this project are compared to. The RMSE and score values are reported in Table III. The true and predicted values of RUL, as obtained from the RF model, are shown in Fig. 4.

The Random Forest Model demonstrates strong performance with the FD001 and FD003 datasets, as evidenced by low RMSE values. These datasets feature fewer fault modes,

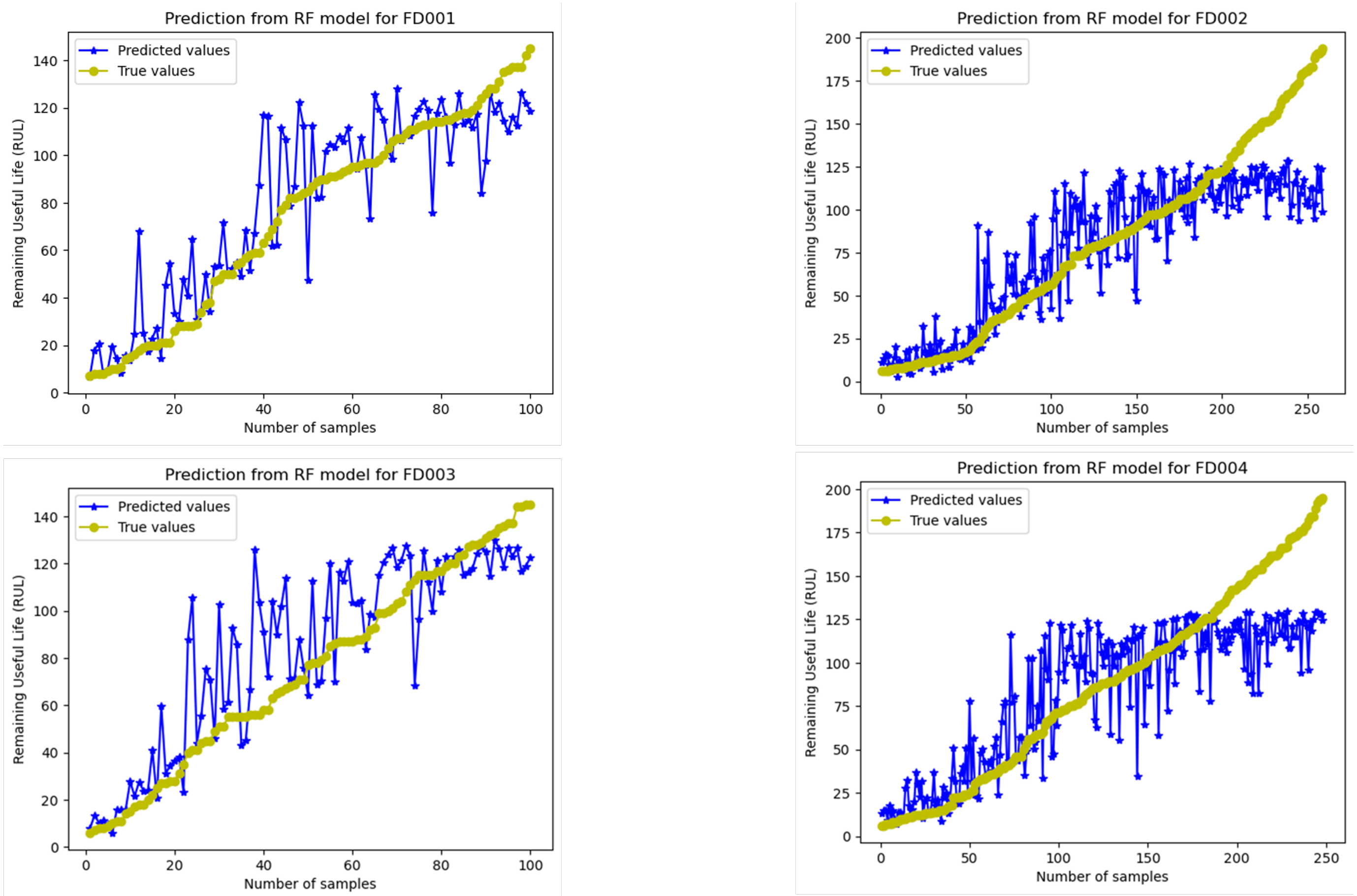

Fig. 4. RUL Values from Random Forest Regressor

TABLE III
PERFORMANCE METRICS FOR DIFFERENT DATASETS WITH RANDOM FOREST REGRESSOR

Dataset	RMSE	Score
FD001	18.15	958.09
FD002	28.59	10690.96
FD003	21.63	2833.31
FD004	30.09	8619.93

allowing the RF Model to effectively capture the variance attributable to the operating conditions. Conversely, the RMSE values are notably higher with FD002 and FD004. These datasets encompass a greater number of operating conditions and fault modes, posing challenges for the RF Model to adequately capture the variances in the data, thus resulting in poorer performance.

B. Convolutional Neural Network

The Convolutional Neural Network (CNN) implemented in this project consists of multiple layers as shown in Fig. 5. The model begins with a series of Convolutional layers followed by Max-pooling layers, which help in extracting important features from the input data. Specifically, the architecture includes three convolutional layers with increasing numbers of filters (32, 64, and 128, respectively), each followed by Max-pooling layers to downsample the feature maps.

The convolutional layers are configured with a kernel size of 3, padding set to 'same' to preserve the input dimensions, and ReLU activation functions to introduce non-linearity into the model.

After the final max-pooling layer, the feature maps are flattened into a one-dimensional vector using a Flatten layer. This flattened representation of the features is then passed through a Dense layer, followed by a ReLU activation function and a Dropout layer with a rate of 0.2. The Dropout layer helps prevent overfitting by randomly dropping out a fraction of the neurons during training.

Finally, the output layer consists of a single neuron, which predicts the remaining useful life (RUL) of the engines based on the last operating cycle from the test data. The model is compiled using the mean squared error (MSE) loss function and the Adam optimizer, with a learning rate scheduler implemented to decrease the learning rate after 50 epochs. Batch-size is set to 512 samples for training data.

The 1D-CNN model consistently demonstrates strong performance with the FD001 and FD003 datasets, both characterized by having only one operating condition and lower variance in the data. Conversely, the model's performance is less

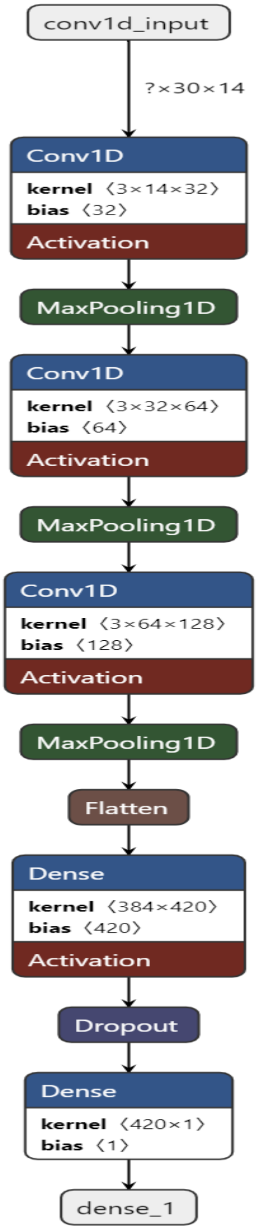

Fig. 5. 1D Convolutional Neural Network for RUL Prediction

satisfactory with FD002 and FD004, suggesting its inability to capture all the variance in the data. However, analysis of the results reveals that lower counts of Remaining Useful Life (RUL) are predicted with higher accuracy compared to higher values, as shown in Fig. 7. This observation is particularly significant, as it implies that the model can effectively identify engines likely to fail earlier, thereby facilitating timely maintenance interventions for engines with fewer operating cycles left.

C. Conv + LSTM Model for FD001 Dataset

The implemented model architecture shown in Fig. 8 combines Convolutional Neural Network (CNN) layers with Long Short-Term Memory (LSTM) layers, aiming to capture both spatial and temporal patterns in the multivariate time series data. The model begins with two Convolutional layers, each followed by Max-pooling layers to downsample the feature maps and reduce dimensionality. The Convolutional layers have different numbers of filters (18 and 36 respectively), with kernel sizes of 2 and ReLU activation functions.

TABLE IV
PERFORMANCE METRICS FOR DIFFERENT DATASETS WITH 1D-CNN MODEL

Dataset	RMSE	Score
FD001	17.24	507.05
FD002	29.68	33331.88
FD003	16.26	920.78
FD004	32.72	19835.46

After the final Max-pooling layer, the features are flattened into a one-dimensional vector and passed through a Dense layer, followed by a Dropout layer to prevent overfitting. Subsequently, two LSTM layers are introduced to capture temporal behaviour in the data. The first LSTM layer is configured to return sequences, allowing it to output sequences of hidden states for each time step, while the second LSTM layer processes these sequences and outputs a single vector representing the final state. Additional Dense layers with ReLU activation functions and Dropout layers are added to further capture complex patterns in the data. The output layer consists of a single neuron, predicting the Remaining Useful Life (RUL) of the engines.

The model is compiled using the mean squared error (MSE) loss function and the Adam optimizer. Batch-size of 32 samples is used, and the model is allowed to train for 10 epochs. The predictions from the model are shown in Fig. 9 (b).

TABLE V
PERFORMANCE METRICS WITH DIFFERENT MODELS FOR FD001 DATASET

Dataset	RMSE	Score
RF	18.15	958.09
1D Conv	17.25	507.05
Conv+LSTM	16.19	440.40

The hybrid Conv+LSTM model exhibits superior performance compared to both the Random Forest Regressor and the 1D-CNN model in terms of RMSE and score values as shown in Table III. The Conv+LSTM model demonstrates significantly faster convergence, requiring only 10 epochs to reach convergence, compared to the 150 epochs needed by the 1D-CNN model.

D. Evaluating Performance of Proposed Models

In this project, predictive models were developed for estimating the Remaining Useful Life (RUL) of engines based on multivariate time series data. The models explored include Random Forest Regressor, Convolutional Neural Network (CNN), and a hybrid CNN-LSTM architecture.

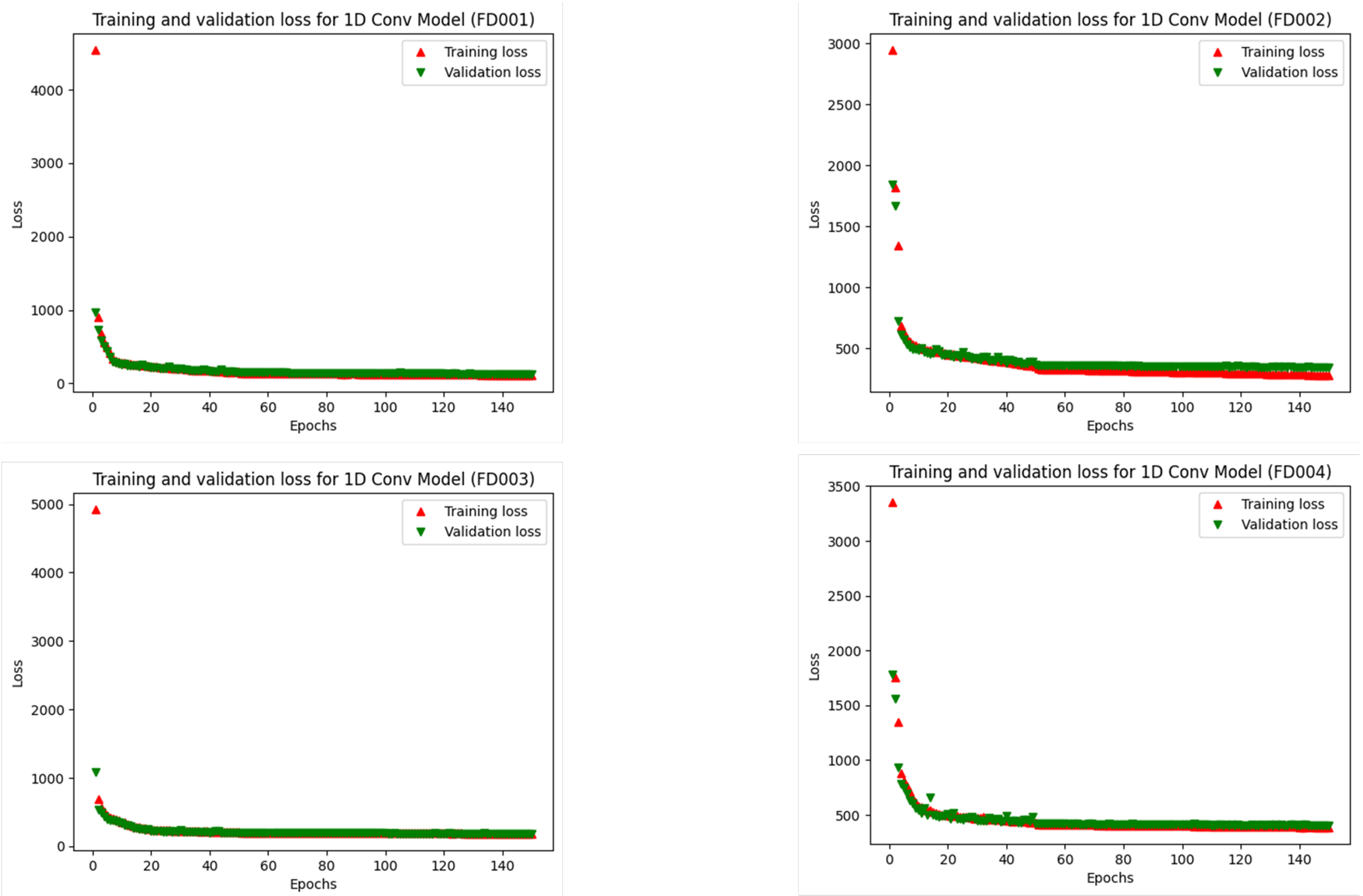

Fig. 6. Loss for 1D-CNN Model

Comparing the developed models with those available in the literature reveals several insights. The Random Forest Regressor, serving as a baseline model in this project, demonstrated robust performance in capturing complex relationships between input features and RUL. However, its performance might be limited in capturing temporal dependencies inherent in time series data.

The 1D-CNN model showed promising results by effectively extracting relevant features from the multivariate time series data. The convolutional layers in the CNN architecture enable the model to capture local patterns in the data, while the subsequent dense layers help in learning global patterns. This approach proved effective in capturing relevant features for RUL prediction. However, the performance of the 1D-CNN with FD002 and FD004 is slightly worse than the baseline established with the RF model, suggesting that there is some degree of overfitting with these datasets.

The hybrid Conv-LSTM model for the FD001 combines the strengths of both CNN and LSTM architectures. By incorporating LSTM layers, the model can capture temporal dependencies in the data, allowing for a more comprehensive understanding of the underlying patterns. This model performed better than the Random Forest model as well as the 1D-CNN model. The Conv-LSTM model outperforms the other models in terms of efficiency, as this model converged much faster than the two other models explored in this project.

A brief comparison with existing literature [11] is provided in Table VI. While the models explored in this project may not achieve the exact RMSE and score values reported in the literature, their performance remains comparable. Notably, the models achieve this level of performance while requiring only a fraction of the computational power needed for the models presented in the literature. This indicates the efficiency and practicality of the approach in this project, making it a viable option for real-world applications where computational resources may be limited.

TABLE VI
PERFORMANCE METRICS WITH DIFFERENT MODELS

Dataset	RF	1D-Conv	Conv+LSTM	Literature [11]
FD001	18.15	17.25	16.19	12.61
FD002	28.59	29.68	-	22.36
FD003	21.63	16.26	-	12.64
FD004	30.09	32.72	-	23.31

IV. CONCLUSION

This project focused on developing predictive models for estimating the remaining useful life (RUL) of engines based

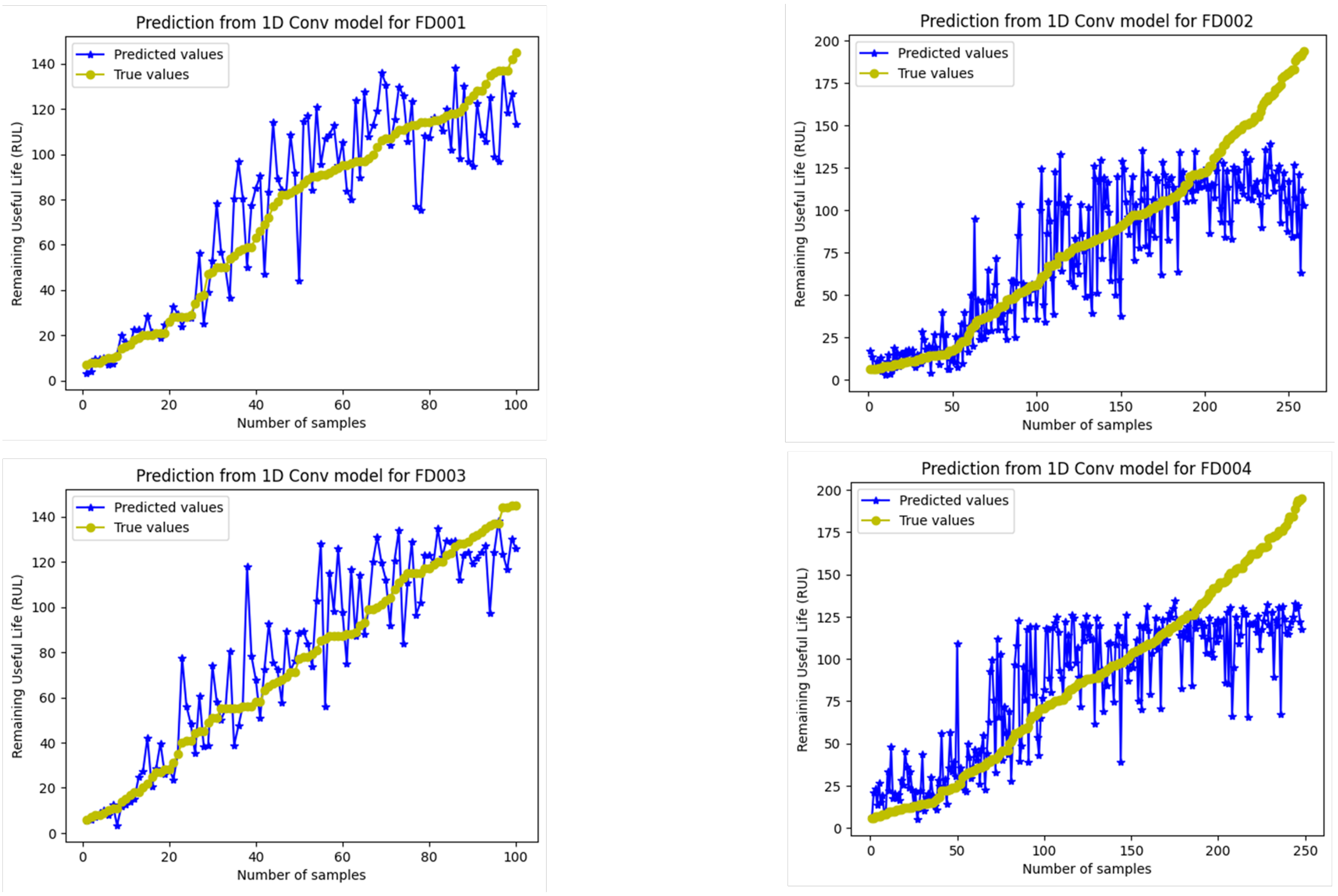

Fig. 7. RUL Values from 1D-CNN Model

on multivariate time series data. The models explored included Random Forest Regressor, 1D Convolutional Neural Network (1D-CNN), and a hybrid CNN-LSTM architecture. These models were trained and evaluated on datasets containing sensor measurements and operational settings of engines, with the goal of predicting the RUL before system failure.

The Random Forest Regressor was an effective baseline model, reflecting complicated correlations between input characteristics and RUL. However, its shortcomings in capturing temporal dependencies in time series data motivated further investigation into more advanced frameworks.

The CNN model showcased promising results by effectively extracting spatial patterns from the multivariate time series data. By leveraging convolutional layers, the model demonstrated an ability to capture local patterns, contributing to improved RUL prediction accuracy.

The hybrid CNN-LSTM model, implemented for the FD001 dataset, combined the strengths of both CNN and LSTM architectures. This hybrid approach allowed for the capture of both spatial and temporal features, resulting in accelerated convergence in the training process, as well as better predictions of the test data.

Overall, the developed models hold significant potential for real-world applications in engine health management, predictive maintenance, and prognostics. By accurately estimating the RUL of engines, especially those with lower RUL values, these models can empower maintenance teams to make informed decisions, optimize resource allocation, and mitigate the risk of costly downtime.

Moving forward, further research and development efforts could focus on enhancing the models' interpretability, scalability, and generalizability across different engine types and operational conditions. Additionally, incorporating advanced techniques such as attention mechanisms and ensemble learning could offer additional improvements in predictive performance. This project lays a solid foundation for leveraging machine learning in addressing critical challenges in engine health management and predictive maintenance.

References

[1] D. J. Mobley, "An introduction to predictive maintenance", Elsevier, 2002.

[2] Y. Liu, X. Zhang, and T. Peng, "A review of data-driven approaches for prognostics and health management in smart manufacturing", Journal of Manufacturing Systems, vol. 48, pp. 258-271, 2018.

[3] W. Zheng, "Data-driven remaining useful life estimation: A review", Mechanical Systems and Signal Processing, vol. 108, pp. 344-364, 2018.

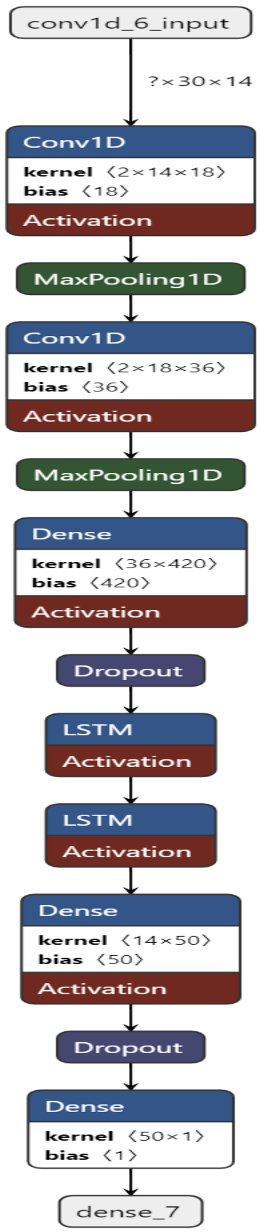

Fig. 8. Conv+LSTM Model for FD001

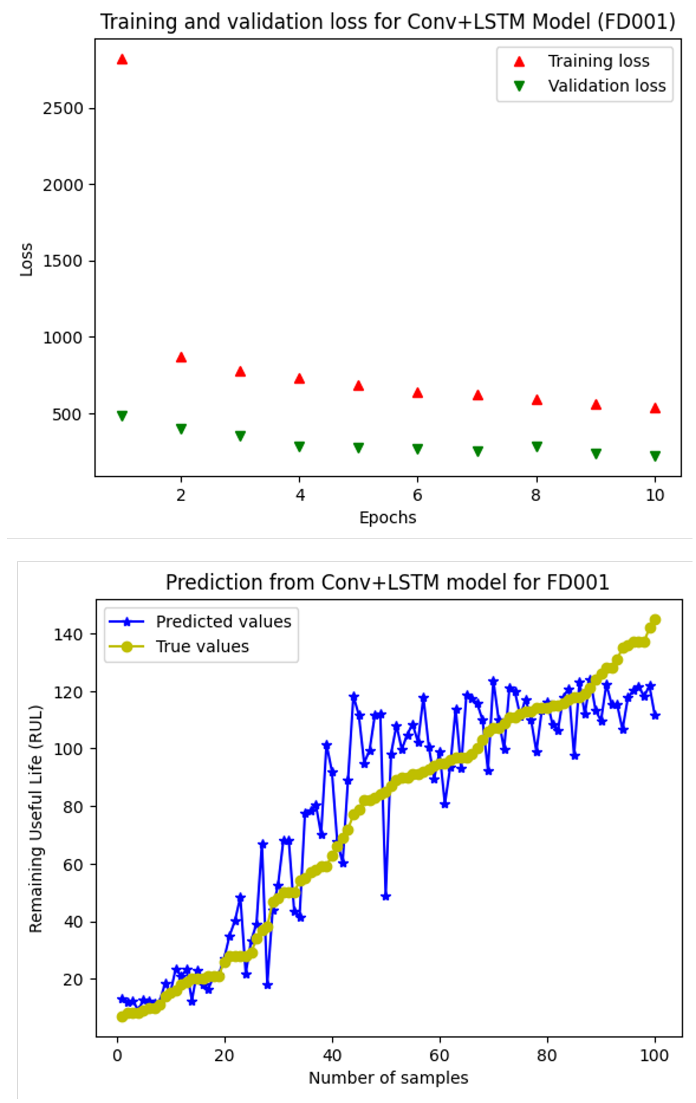

Fig. 9. (a) Loss for Conv+LSTM Model (b) RUL values from Conv+LSTM Model

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

[5] J. Wang, H. Tang, and L. Xie, ”Prognostics and health management: A review on data driven approaches”, Mathematical Problems in Engineering, vol. 2015, Article ID 793161, 2015.

[6] L. Breiman, ”Random forests”, Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ”Gradient-based learning applied to document recognition”, Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[8] J. Lipton, D. Kale, and R. Wetzel, ”Critical review of recurrent neural networks for sequence learning”, arXiv preprint arXiv:1506.00019, 2015.

[9] S. Saha, S. Poll, and A. Goebel, ”Prognostics methods for battery health monitoring using a Bayesian framework”, IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 5, pp. 841-850, 2017.

[10] D. Frederick, J. DeCastro, and J. Litt, ”User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS),” NASA/ARL, Technical Manual TM2007-215026, 2007.

[11] 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.

[12] F. O. Heimes, ”Recurrent neural networks for remaining useful life estimation,” 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 2008, pp. 1-6, doi: 10.1109/PHM.2008.4711422.