The International Journal of Materials and Engineering Technology (TIJMET) (2023) 6(2)
http://dergipark.gov.tr/tijmet
ISSN: 2667-4033

RESEARCH ARTICLE

A comparative study on data pre-processing techniques for remaining useful life prediction of turbofan engines

Meryem Erdoğan[1], Muharrem Mercimek[2]

[1]Yildiz Technical University, Department of Avionics Engineering, Istanbul, Türkiye
[2]Yildiz Technical University, Department of Control and Automation Engineering, Istanbul, Türkiye

Article Info

Article history:
Received: 14.11.2023
Revised: 24.12.2023
Accepted: 26.12.2023
Published Online: 31.12.2023

Keywords:
C-MAPSS
LSTM
Wavelet transforms
Savitzky-Golay
EMA
Min-Max
Z-score

Abstract

This study delves into the application of Long Short-Term Memory (LSTM) for predicting Remaining Useful Life (RUL) in Turbofan Engines using the Jet Engine Simulated Dataset (C-MAPSS), systematically examining the combined impact of diverse data pre-processing techniques on RUL prediction, with a particular focus on the application of filtering and normalization. The initial filtering of the dataset employs Savitzky-Golay (SG), wavelet transform, and exponential moving average (EMA) techniques to effectively mitigate noise. Subsequently, minimum-maximum and z-score normalization techniques are implemented. Each filtering method, paired with distinct normalization approaches, is meticulously evaluated, and the performance of LSTM models in RUL prediction is assessed for each combination. The quantitative analysis of experimental outcomes indicates that normalization and filtering contribute to the improvement of the training phase in LSTM models, ultimately enhancing the accuracy of RUL prediction. The study emphasizes that the selection of an optimal data pre-processing structure plays a crucial role in influencing the efficiency of network training, underscoring the potential for optimizing RUL prediction through the application of the LSTM model.

1. Introduction

Today, deep learning and data analytics techniques are widely used to monitor the health status of equipment and optimize predictive maintenance (PdM) in aviation industry. PdM is a maintenance method based on the condition data of the equipment. Based on historical equipment condition data, it predicts when equipment may be damaged in the future. PdM is used to monitor the past health data of equipment and make timely adjustments to the equipment. This is quite a different approach from the routine maintenance methods of the past. PdM saves unnecessary costs, allows for early repairs when equipment reaches the stage of breakdown and increases operational availability of the aircraft. It can prevent unforeseen equipment downtime caused by unexpected failures and improper operation. There are three primary approaches employed for predicting RUL of an equipment: data-driven, physics-based, and hybrid-based. While physics-based and hybrid-based approaches [1, 2] are widely utilized to enhance prediction accuracy, their complexity and demand for in-depth knowledge of aircraft systems render them less cost-effective and less preferable for adoption by airlines and aircraft manufacturers. In the aviation industry, particularly among airlines and aircraft manufacturing companies, a clear inclination exists towards cost-effective methodologies. Specifically, there is a preference for data-driven prognostic approaches over physics-based or hybrid-based alternatives. This preference is grounded in the inherent complexities and knowledge-intensive nature of aircraft systems. In aviation industry prioritizing operational efficiency and cost-effectiveness, the preference is for cost-efficient methodologies such as data-driven prognostic approaches. These strategies leverage data and advanced deep learning models, to deliver precise predictions regarding health of equipment and RUL. Aligned with the industry's commitment to real-time monitoring, predictive maintenance, and cost optimization, the data-driven approach not only tackles challenges associated with overfitting, limited data, and model complexity but also significantly elevates operational reliability.

Various data-driven deep learning models, such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Feed Forward Neural Network (FNN), Deep Belief Network (DBN), and Graphical Neural Network (GNN), have proven effective in predicting RUL [3-5]. To further optimize model performance and tailor them to specific tasks, various LSTM variants, such as BiLSTM, GRU, Peephole LSTM, and Vanilla LSTM, have also been developed [6]. Ensemble models, such as CNN-LSTM, are also used in RUL prediction, aiming to provide more comprehensive and reliable predictions by leveraging the strengths of different deep learning architectures [1, 6].

To monitor the health of equipment or systems, physical sensors measure various physical parameters, such as temperature, pressure, vibration, power, acoustic wave and speed. Virtual sensors, on the other hand, combine and calculate parameters to provide insights into the equipment or system's health. These parameters can be valuable indicators of equipment or system’s health. For instance, a sudden surge in engine temperature can alert us to a potential issue with the engine's cooling system. Time-series data, collected at regular intervals, captures this valuable information, enabling the monitoring of aircraft systems. Time-series data is especially beneficial because it allows us to identify trends and patterns that may not be evident from a single data point. For example, a

Corresponding Author: Meryem Erdoğan
E-mail: meryemilikosm@gmail.com

How to cite this article:

Erdoğan, M., Mercimek, M., A comparative study on data pre-processing techniques for remaining useful life prediction of turbofan engines, The International Journal of Materials and Engineering Technology (TIJMET), 2023, 6(2):50-58

gradual increase in engine vibration over time could signal wear and tear on the engine's components.

This study aims to enhance the accuracy of RUL estimation by employing sensor data processed through various normalization and filtering techniques as input for the LSTM model. The LSTM model was chosen based on the literature and its success in the field of RUL prediction. The study underscores the direct impact of selecting appropriate data pre-processing methods on the training efficiency of the LSTM model, emphasizing its potential to optimize RUL prediction. It also highlights the pivotal role played by machine learning and data analytics techniques in optimizing health monitoring and PdM for aviation industry. Furthermore, conducting additional investigations and comparisons with other deep learning models and ensemble methods can contribute to the progression of RUL prediction research.Enhancing the accuracy of RUL estimation hinges significantly on the pre-processing of sensor data. The normalization and filtering techniques employed in this stage play a pivotal role in refining the structure of the measured sensor data. These techniques are specifically designed to diminish noise and improve the overall quality of the data. In particular, normalization and filtering methods are extensively utilized to enhance the structure of the measured sensor data and minimize extraneous noise. The SG filtering technique, one among these methods, has found widespread application in various domains for noise reduction in time series data [7-9]. Another effective technique for reducing noise in time series data, such as sensor data or electrocardiogram (ECG) signals indicating heart rhythm, is the wavelet transform method [10-12]. In this method, sensor data is analysed at different scales and frequencies. Noise is usually concentrated in the high frequency components, while the actual sensor measurement data becomes more prominent in the low frequency components. By setting a threshold value, the noise components are detected and the components exceeding the threshold value are filtered out. This process reduces the noise in the sensor measurements while preserving the important components of the actual sensor data. As a result, cleaner and more meaningful data is obtained [13]. Finally, the EMA filtering technique is a method also used for time series data. This technique aims to obtain a smoother trend by reducing sudden fluctuations in the data. It is an effective filtering method to reduce the undesirable effects caused by sudden fluctuations, especially in noisy sensor data [14]. The EMA filtering technique has been successfully applied to sensor data in both the training and test datasets used in RUL prediction studies [15-18].

Normalization is used when sensor data are in different units or scales. Sensor measurement data used for RUL prediction often have different units or scales. For example, one sensor may represent temperature values while another sensor may represent vibration values. These different scales can make it difficult for the deep learning model to accurately learn patterns and relationships. Normalization transforms the data into a specific range or standard distribution, eliminating scale differences and enabling models to produce more consistent and comparable results [19]. In general, minimum-maximum normalization [20-22] and z-score normalization [23, 24] are widely used normalization techniques for RUL prediction. In this study, these widely used normalization and filtering techniques are applied separately and together on sensor data in test and training datasets and their effects on LSTM and RUL prediction performance are compared.

A review of the literature reveals that there are studies comparing only filtering techniques with each other [25, 26] and only normalization techniques with each other [27, 28]. However, this study, which compares normalization and filtering techniques in various combinations and investigates the most effective combined data pre-processing methods for use in LSTM. This study seeks to contribute to the aviation industry by offering practical insights into the prognostic approaches employed for aircraft systems.

2. Data set and methodology

2.1 C-MAPSS dataset

This study employs the C-MAPSS dataset, which comprises simulated data generated by NASA's model-based Turbofan engine degradation simulation program, C-MAPSS [29]. This dataset has subsets as training, test and validation data.

Figure 1 shows the maximum life cycles of 100 engines in the training dataset, along with the frequency of number of engines. Approximately three engines in the dataset were able to operate until the 350th cycle. Nearly 30 engines failed at the 200th cycle and stopped operating. Figure 2 presents the last operated cycles of 100 engines in the test dataset, showing the distribution of engine numbers. The engines are operated until specific cycles, and the NASA-provided test data does not include the exact end-of-life cycle for each engine. This information is provided in the validation dataset, which serves as the actual RUL values for the engines from the test set.

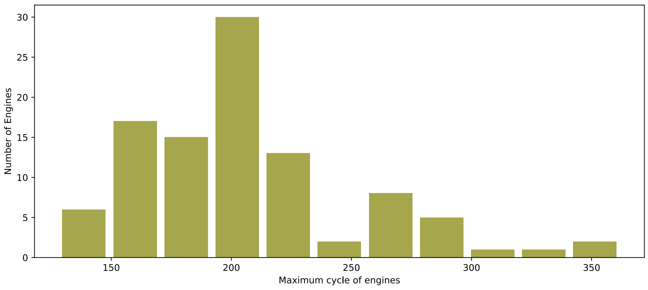

Figure 1 Frequency versus maximum life of engines in train dataset

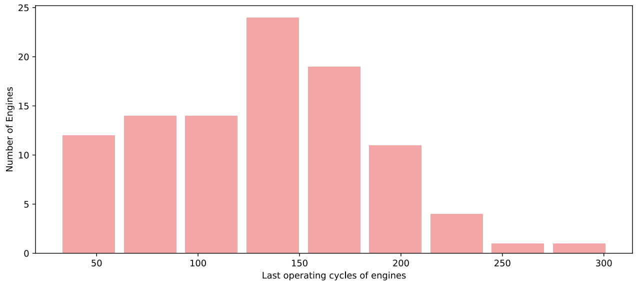

Figure 2 Last operating cycles of engines in test dataset

The size of the training data consists of 20631 rows and 26 columns, while the test data consists of 13096 rows and 26 columns. In both datasets, the column labels are defined as unit ID, cycles, operational setting {1-3} and sensor {1-21}. The unit IDs takes values between 1, 2, ..., 100. The cycle is a unit representing the operating time of each engine in steps. Each cycle contains the measurement values of the sensors used to monitor the health status of the engine. In the training data, there is an association between the cycles and the RUL, as the sensor data is provided until the last operated cycle of the engines' lifetime, the maximum cycle. RUL values are calculated in this dataset as follows:

$$RUL(t) = Max(t)_{unitID} - t \qquad (1)$$

For example, engine 1 or unit ID '1' stopped operating at cycle, *192*, in the training dataset. In this case, unit ID '1' has cycles, *t*, in the range 1, 2, 3, 4, ..., 192. Applying the formula used to calculate the RUL value of the engine at cycle, *t=1*, we obtain *RUL (1) =192-1=191.* A new column containing the calculated RUL values for each cycle has been added to the training dataset. In the training dataset, the RUL values were directly linked to the corresponding sensor values. The RUL values were then fed into the LSTM model as target input without undergoing any data pre-processing. This means that the RUL values were provided to the model in their original form, without any filtering or scaling. The LSTM model was trained on this data, along with the sensor data, to learn the relationship between sensor readings and RUL values. Since the maximum cycles of the engines in the test dataset were not known in advance, sensor data were provided up to a certain cycle and the RUL values at each cycle were attempted to be estimated and evaluated using the validation dataset which includes actual RUL values.

2.2 Filtering techniques

2.2.1 Savitzky-Golay filtering

SG filtering is utilized for smoothing and noise reduction in time-series data, with its parameters, polynomial degree (*l*) and window size (*n*), crucially influencing its performance. In the context of engine sensor data, such as sensor 7 over the initial 175 operating cycles of unit ID '1', a polynomial degree of *4* and a window size of *3* were chosen. The rationale behind these parameter choices is the need to balance noise reduction with responsiveness to changes in the sensor data. A lower polynomial degree and larger window size are preferred for smoother data, whereas data with rapid changes may require a higher degree and smaller window. Visual inspection of the filtered sensor data, as shown in Figure 3, is essential for fine-tuning these parameters to ensure optimal noise reduction without sacrificing important feature trends.

The SG filter estimates the values of a signal in a given range using an approximate polynomial function and then filters the signal using this function. The mathematical formula of the polynomials used in the SG filter is given below:

$$y_t = c_0 + c_1 x_t + c_2 {x_t}^2 + \cdots + c_n {x_t}^n \qquad (2)$$

In this study, the SG filtering technique is consistently applied to the entire sensor data, including both the training and test data sets, in various comparison scenarios. For each of the 21 distinct sensors, the measured values at cycle *'t'* are represented as $x_{t1}, x_{t2}, ..., x_{t20}, x_{t21}$. Correspondingly, the filtered values for these sensors are designated as $y_{t1}, y_{t2}, ..., y_{t20}, y_{t21}$. In this context, $y_{t,}$ represents the processed sensor value predicted by the polynomial to serve as input to the deep learning model, while x_t denotes the raw sensor value. The coefficients of the polynomial, $c_0, c_1, c_2, c_3, ..., c_n$, are calculated within the SG filter by minimizing the sum of the squared errors [30].

2.2.2 Wavelet transform

In this investigation, filtering is implemented using the Daubechies wavelet type, incorporating two different parameters for the noise reduction function: the threshold value and the decomposition level. By iteratively adjusting both the decomposition level and threshold value in the wavelet-based noise reduction process, we can navigate the trade-off between removing noise and preserving the data trends. Opting for a higher decomposition level and a lower threshold value can yield a more detailed denoised data, potentially retaining more of the original data but at the risk of keeping more noise. Conversely, choosing a lower decomposition level and a higher threshold value can effectively eliminate noise but may introduce more distortion to the data [13, 31]. This iterative exploration allows for fine-tuning the transform parameters based on the specific characteristics of the data and the desired balance between noise reduction and data fidelity.

Specifically, a threshold of *0.1* and a level of *9* are applied to process all sensors within both the test and training datasets across relevant scenarios. The choice of a threshold value of *0.1* implies a gentle noise reduction process, suitable for data without excessively high noise levels. This cautious approach in thresholding helps avoid distorting the essential structure of the sensor data. Simultaneously, the decision to use a decomposition level of *9* indicates a detailed analysis of the sensor data. The wavelet decomposition breaks down the data into various frequency bands, and this higher decomposition level is motivated by the belief that the data carries valuable information in its high-frequency components. The combined use of a moderate threshold and a high decomposition level seeks a balanced approach, ensuring effective noise removal while preserving the granularity of the original data, a crucial aspect for subsequent analysis.

Illustrated in Figure 4, which showcases the filtering result for a sensor in the training dataset, the figure depicts the measured values of sensor 7 during the initial 175 operating cycles of unit ID '1' alongside the corresponding filtered values over the given cycles.

2.2.3 Exponential moving average (EMA)

EMA filtering is a method for smoothing and reducing noise in time-series data. It works by assigning exponentially decaying weights to past data points, with more recent data points receiving higher weights and older data points receiving lower weights. This approach allows the filter to effectively capture recent changes in the data while also incorporating information from older data.

The EMA filter is typically implemented using the following formula:

$$y_t = \alpha * x_t + (1 - \alpha) * y_{t-1} \qquad (3)$$

where:

- y_t is the filtered value at cycle *t*
- x_t is the raw sensor value at cycle *t*
- y_{t-1} is the filtered value at the previous cycle *(t - 1)*
- *α* is the exponential weighting factor

The values which are belongs to each of the 21 different sensors at cycle *t* are represented as $x_{t1}, x_{t2}, ..., x_{t20}, x_{t21}$. Correspondingly, the filtered values for these sensors are designated as $y_{t1}, y_{t2}, ..., y_{t20}, y_{t21}$.

The exponential weighting factor *α* determines the relative importance of recent and historical data. A higher *α* value prioritizes more recent data, effectively capturing real-time changes in the data. Conversely, a lower *α* value emphasizes older data, enabling the preservation of longer-term trends [32].

In this study, the exponential weighting factor *α* is set to *0.7*. This chosen value strikes a delicate balance, effectively blending recent and historical data. By setting *α* to *0.7*, the filter is designed to prioritize more recent information while still incorporating valuable insights from older cycles. This approach ensures an effective smoothing of the sensor data while preserving the data trends.

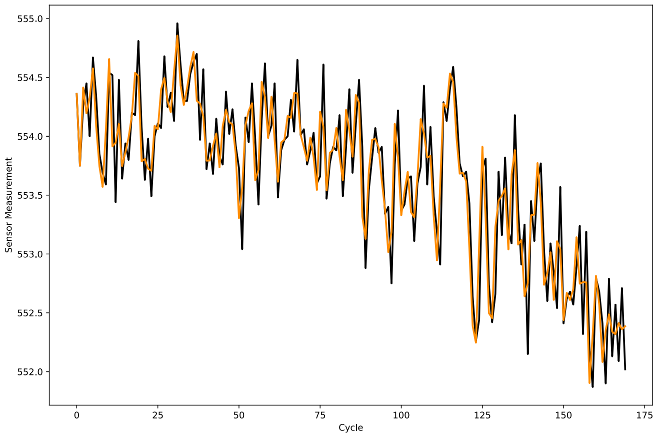

Figure 3 Noise reduction with Savitzky-Golay on Sensor 7

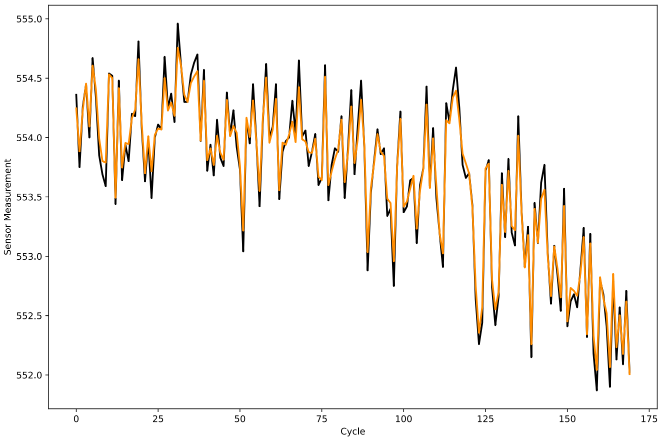

Figure 4 Noise reduction with wavelet transform on Sensor 7

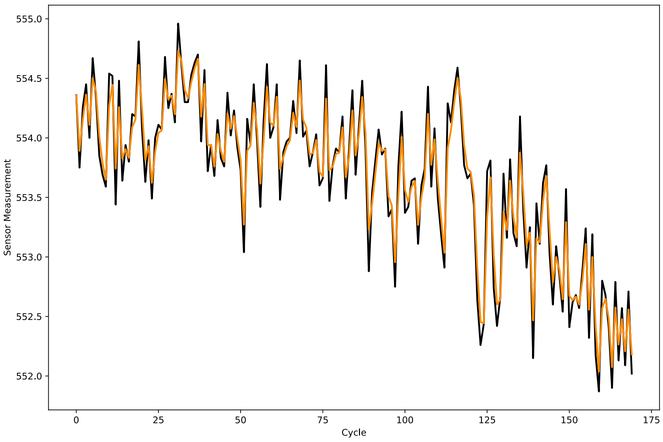

Figure 5 Noise reduction with EMA on Sensor 7

As an illustration, Figure 5 shows the filtering result for a sensor within the training data set. The figure shows the values of sensor 7 measured during the first 175 operating cycles of unit ID '1', alongside the filtered values over the cycles.

2.3 Normalization techniques

Minimum-maximum normalization is a method used to transform data into a specific range. This method compresses data values into a range of values between 0 and 1, independent of their original range. Its mathematical formula can be shown as follows [19]:

$$y_t = \frac{x_t - \min(x_t)}{max(x_t) - min(x_t)} \qquad (4)$$

Where x_t is the sensor data and y_t is the normalized sensor data at cycle, *t*. *min(x_t)* and *max(x_t)* represent the minimum and maximum sensor values at cycle, *t*.

The z-score normalization involves subtracting the mean of sensor data at time cycle, *t* and dividing this difference by the standard deviation. This method normalizes the sensor data for each cycles using the following formula [19]:

$$y_t = \frac{(x_t - \mu)}{SD} \qquad (5)$$

Where y_t is the normalized sensor data at cycle, *t and* x_t is the raw sensor data at cycle, *t*. *μ* is the mean of the sensor data over all time cycles and *SD* is the standard deviation of these values. For 21 different sensors, the measured values at cycle *t*, are denoted as $x_{t1}, x_{t2}, \ldots, x_{t20}, x_{t21}$. The normalized values of these sensors are also denoted as $y_{t1}, y_{t2}, \ldots, y_{t20}, y_{t21}$. In this study, all sensors in both the training and test datasets are normalized for the relevant scenarios using minimum-maximum with the range between {0, 1} and z-score normalization techniques.

2.4 Long-short-term memory networks (LSTM)

LSTM models are the most widely used choice for RUL prediction using time series data due to their ability to handle sequential data, capture long-term dependencies, resist noise, adapt to different RUL prediction scenarios and continuously improve through learning [35]. These characteristics make LSTM models powerful techniques for accurately predicting the RUL of equipment, enhancing its reliability, and optimizing PdM management.

When the LSTM is analysed structurally, the representation of the internal structure of the cell is given in Figure 6.

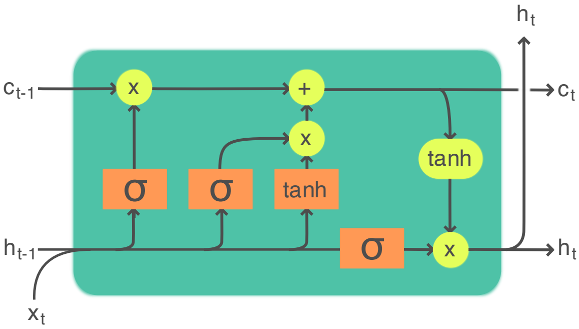

Figure 6 An illustration of the internal structure of the LSTM cell [33]

The LSTM network computes unit activations based on a given input across a time cycle. These activations are regulated by the gates and cell states within the network. The following equations define the activations of the LSTM units;

$$\begin{aligned} g_t &= tanh(W_g * [h_{t-1}, x_t] + b_g) \\ i_t &= (W_i * [h_{t-1}, x_t] + b_i) \\ f_t &= \sigma(W_f * [h_{t-1}, x_t] + b_f) \\ o_t &= \sigma(W_o * [h_{t-1}, x_t] + b_o) \\ c_t &= f_t * c_{t-1} + i_t * g_t \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{6}$$

Within the LSTM algorithm, the arrays *i, f, o* and *c*, denoting the input gate, forget gate, output gate and cell activation, are central components of the cell. These arrays, of the same size as the hidden array h, carry essential information. The W terms correspond to the weight matrices governing the gate sequences within the cell structure.

Conventionally, an activation function such as *tanh* is applied in the output layer of the LSTM configuration. These equations describe the gating activations within the LSTM units that influence the cell state updates. Consequently, the LSTM network establishes connections by processing inputs across time cycles and using memory mechanisms to capture sequential data patterns [34].

In Equation (6), x_t is the input to the LSTM model and in this study, the raw sensor data is referred as x_t, and the processed sensor data is referred as y_t .

This study aims to investigate the effect of different filtering and normalisation techniques on the performance of the LSTM model using the C-MAPSS dataset. In order to compare the data pre-processing techniques in different scenarios, the hyper-parameters of the LSTM model were set to be the same for all scenarios. These hyper-parameters are given in Table 1.

Table 1. LSTM model hyper-parameters

Hyper-parameter	Value
Number of LSTM layers	3
Number of dense layers	3
Learning speed	0.01 & 0.001
Activation function	tanh
Number of trainings	10
Batch size	64
Optimizer	Adam

The root mean square error (RMSE) performance metric and Pearson correlation coefficient (PCC) were used to evaluate the performance of the LSTM models. The mathematical expressions of RMSE and PCC are given in Equation (7) and Equation (8) respectively.

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

$$PCC = \frac{\sum_{i=1}^{N}\left(y_i - \mu_{y_i}\right)\left(\hat{y}_i - \mu_{\hat{y}_i}\right)}{\sqrt{\sum_{i=1}^{N}\left(y_i - \mu_{y_i}\right)^2}\sqrt{\sum_{i=1}^{N}\left(\hat{y}_i - \mu_{\hat{y}_i}\right)^2}} \tag{8}$$

In Equation (7) and Equation (8), y_i represents the actual RUL value for observation *i*, $\hat{y}_i$ represents the predicted RUL value and *N* represents the number of observations. The RMSE is commonly used to calculate the error between the actual and predicted values and to evaluate the predictive performance of the model. Lower RMSE values indicate that the model makes better predictions and fewer errors. Higher values indicate poor performance of the model.

The PCC value is between -1 and 1 and represents the similarity between the actual and predicted values. Its interpretation is as follows;

- If PCC is close to -1: There is a strong negative relationship. It indicates an inverse relationship between actual and predicted values. One variable increases while the other decreases.
- If the PCC is close to 1: There is a strong positive relationship. Indicates a direct relationship between actual and predicted values. When one variable increases, the other also increases.
- When PCC outputs NaN (invalid number): This indicates that there are not enough predicted values for the PCC to be calculated or that the predicted values have the same value.

3. Findings and discussion

In this study, different scenarios with different combinations of data pre-processing techniques were developed and applied to a same LSTM model. The aim of the experiments was to carefully investigate and evaluate the effectiveness of different data pre-processing combinations in improving the accuracy of RUL predictions using a LSTM model, ultimately identifying the optimal pre-processing strategies that lead to the most accurate RUL estimates. The labels for the scenarios are provided in Table 2, while the defined combinations of scenarios being compared and labels are listed in Table 3.

Table 2. Labels for scenarios

Script Label	Techniques
SN1	Raw data + LSTM
SN2	Raw data + SG filter + LSTM
SN3	Raw data + wavelet transform + LSTM
SN4	Raw data + EMA + LSTM
SN5	Min-max normalization + LSTM
SN6	SG filter + Min-max normalization + LSTM
SN7	Wavelet transform + Min-max normalization + LSTM
SN8	EMA + Min-max normalization + LSTM
SN9	Z-score normalization + LSTM
SN10	SG filter + Z-score normalization + LSTM
SN11	Wavelet transform + Z-score normalization + LSTM
SN12	Z-score normalization + EMA + LSTM

Table 3. Comparative scenarios and labels

Comparative Scenario Label	Scenarios
comparativeSN_1	SN2, SN3, SN4
comparativeSN_2	SN6, SN7, SN8
comparativeSN_3	SN10, SN11, SN12
comparativeSN_4	SN1, SN5, SN9
comparativeSN_5	SN2, SN6, SN10
comparativeSN_6	SN3, SN7, SN11
comparativeSN_7	SN4, SN8, SN12

While defining the comparative scenarios, we wanted to determine the best filtering technique (comparativeSN_1) applied to the raw sensor data. In addition, the best filtering technique when applied with the minimum-maximum normalization method (comparativeSN_2) and the best filtering technique when applied with the z-score normalization technique (comparativeSN_3) were investigated. In addition, the impact of the normalization technique on the performance of the LSTM model is investigated in the comparative scenarios, comparativeSN_4, comparativeSN_5, comparativeSN_6 and comparativeSN_7.

3.1 Impact of normalization techniques on the performance of the LSTM model

In order to evaluate the performance of normalization techniques on the LSTM model, different comparative scenarios were determined and the results were compared (Table 4, Table 5, Table 6 and Table 7). In the comparative scenarios, LSTM models are trained using raw sensor data, minimum-maximum and z-score normalized sensor data with different filtering techniques and model performances are compared with RMSE and PCC metrics.

According to the results obtained and the prediction-actual RUL value graphs, in scenarios where no normalization technique is applied (e.g., Figure 7) the model outputs have high RMSE values and there is no correlation between actual and predicted RUL values in the graphs.

When the LSTM model is trained using processed data through normalization techniques and raw data without any filtering methods, insights from Table 4 and Figure 10 reveal the following: the most efficient scenario emerges when normalized data is employed with the z-score normalization method.

When the best normalization technique was investigated in combination with the SG filtering technique, it was observed that the z-score normalization technique had a more positive effect on the performance of the LSTM model compared to minimum-maximum normalization (Figure 9 and Table 5).

When the best normalization technique was investigated in combination with the wavelet transform technique, it was observed that the minimum-maximum normalization technique had a more positive effect on the performance of the LSTM model compared to z-score normalization (Figure 11 and Table 6).

When the best normalization technique was investigated to be applied together with EMA, it was observed that the z-score normalization technique had a more positive effect on the performance of the LSTM model compared to minimum-maximum normalization (Figure 12 and Table 7).

According to the results of all comparative scenarios, the technique with the most positive effect on the performance of the LSTM model was observed as the z-score normalization method.

In a study, investigation of normalization techniques across various time series datasets was conducted to explore alternatives to the commonly favoured z-score normalization method [28]. Z-score normalization is typically the preferred [23, 24] choice over minimum-maximum normalization for most applications due to its greater robustness, versatility, and overall effectiveness. Z-score normalization's ability to handle outliers, preserve distribution shape, and ensure equal scaling across features makes it suitable for a wide range of analytical tasks. However, min-max normalization remains useful when there is a specific need to preserve the original data range, and its simplicity and faster application make it a viable option in certain scenarios. The choice between the two methods ultimately depends on the specific requirements and characteristics of the data in a given application.

Table 4. comparativeSN_4 result

Scenarios	RMSE	PCC
SN1	41.78	NaN
SN5	15.80	0.9314
SN9	14.81	0.9223
	The best scenario, SN9	The best scenario, SN9

Table 5. comparativeSN_5 result

Scenarios	RMSE	PCC
SN2	42.03	NaN
SN6	15.45	0.9201
SN10	13.57	0.9466
	The best scenario, SN10	The best scenario, SN10

Table 6. comparativeSN_6 result

Scenarios	RMSE	PCC
SN3	42.10	NaN
SN7	22.70	0.8503
SN11	24.18	0.8374
	The best scenario, SN7	The best scenario, SN7

Table 7. comparativeSN_7 result

Scenarios	RMSE	PCC
SN4	42.04	NaN
SN8	15.32	0.9301
SN12	14.28	0.9393
	The best scenario, SN12	The best scenario, SN12

Table 8. comperativeSN_1 result

Scenarios	RMSE	PCC
SN2	42.03	NaN
SN3	42.10	NaN
SN4	42.04	NaN
	The best scenario, SN2	

3.2 Effect of filtering techniques on the performance of the LSTM model

In order to evaluate the performance of the filtering techniques on the LSTM model, several comparative scenarios were defined and the results were compared (Table 8, Table 9 and Table 10). In the comparison scenarios with defined combinations, the raw sensor data and normalised sensor data were filtered using SG, wavelet transform and EMA techniques. The performance of the LSTM models trained on these filtered data is compared using the RMSE and PCC metrics.

The performance of the LSTM model with raw sensor data, as depicted in Table 8 and Figure 7, demonstrates poor results. Elevated RMSE values indicate significant predictive errors, while the stark dissimilarity between the actual and predicted graphs underscores the model's inability to effectively capture the underlying patterns within the sensor data.

Examining the scenarios using SG, wavelet transform, and EMA filtering on the minimum-maximum normalized sensor data, it became evident from Table 9 that SN8 stood out as the most effective approach. Specifically, applying EMA filtering with the minimum-maximum normalization technique produced the most favourable results, as evidenced in Figure 8. Furthermore, SN6 and SN7, employing different filtering techniques, demonstrated relatively low RMSE values and notably strong correlations in their respective plots.

Upon reviewing Table 10, it became evident that among the scenarios applying SG, wavelet transform, and EMA filtering techniques to sensor data, normalized using the z-score method, SN10 emerged as the most effective (as shown in Figure 9). Specifically, utilizing the SG filter on the sensor data normalized with the z-score method yielded the most optimal outcome. Additionally, it's worth noting that EMA filtering technique, as seen in SN12, showcased comparable result.

After assessing how filtering and normalization techniques impacted the LSTM model's performance, we can now contrast the scenarios that yielded the best results. Table 11 presents a comparison of these top-performing scenarios.

Upon analysis in Table 11, SN8, SN9, SN10, and SN12 displayed remarkably close RMSE and PCC values, indicating successful outcomes. Notably, the LSTM model trained using the SG technique in tandem with z-score normalization method emerged as the optimal scenario, showcasing the lowest RMSE and the highest PCC. Nevertheless, the results derived from implementing the wavelet transform and EMA filtering also showed a positive influence on the LSTM model's performance. This observation suggests that optimizing the LSTM model, fine-tuning filtering techniques, as well as employing wavelet transform or EMA techniques, can lead to optimal performance results, proving to be viable methods.

Table 9. comperativeSN_2 results

Scenarios	RMSE	PCC
SN6	15.45	0.9201
SN7	22.70	0.8503
SN8	15.32	0.9301
	The best scenario, SN8	The best scenario, SN8

Table 10. comparativeSN_3 results

Scenarios	RMSE	PCC
SN10	13.57	0.9466
SN11	24.18	0.8374
SN12	14.28	0.9393
	The best scenario, SN10	The best scenario, SN10

Table 11. Comparing the most effective scenarios

Scenarios	RMSE	PCC
SN7	22.70	0.8503
SN8	15.32	0.9301
SN9	14.81	0.9223
SN10	13.57	0.9466
SN12	14.28	0.9393
	The best scenario, SN10	The best scenario, SN10

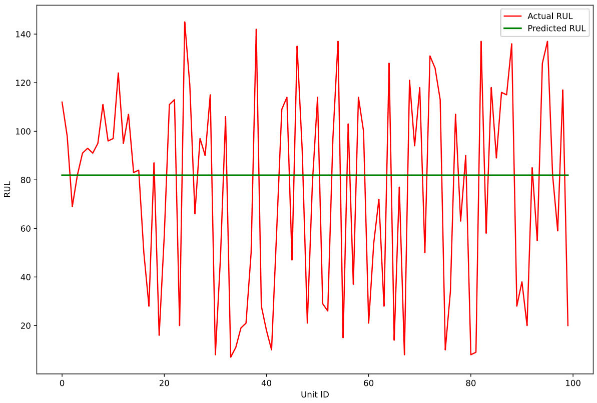

Figure 7 The most effective scenario, denoted as SN2, evaluated within the context of the comparativeSN_1

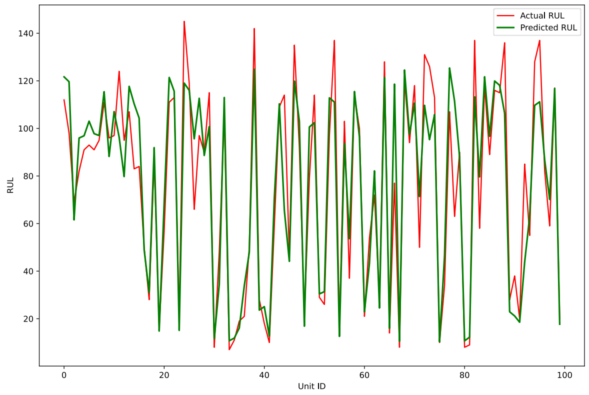

Figure 8 The most effective scenario, denoted as SN8, evaluated within the context of the comparativeSN_2

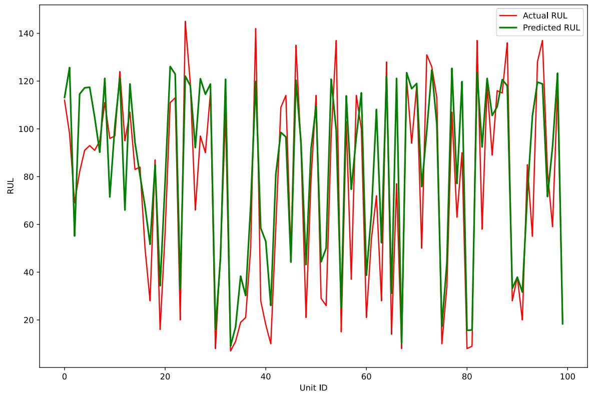

Figure 9 The most effective scenario, denoted as SN10, evaluated within the context of the comparativeSN_3 and comparativeSN_5

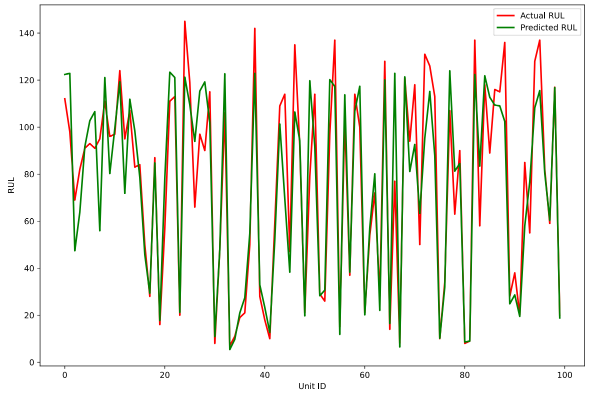

Figure 10 The most effective scenario, denoted as SN9, evaluated within the context of the comparativeSN_4

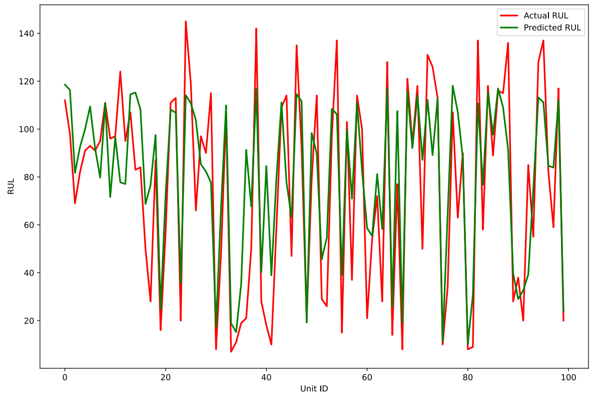

Figure 11 The most effective scenario, denoted as SN7, evaluated within the context of the comparativeSN_6

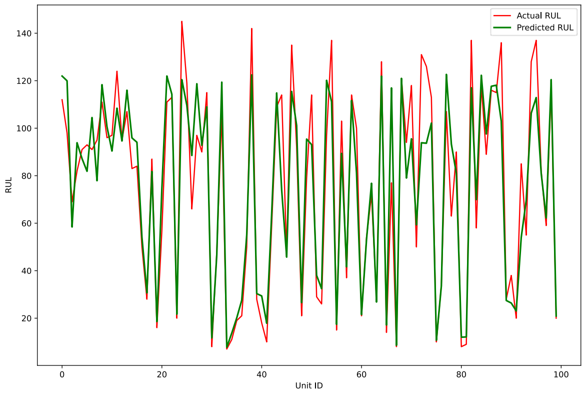

Figure 12 The most effective scenario, denoted as SN12, evaluated within the context of the comparativeSN_7

4. Results

The ability to accurately predict the RUL of equipment is crucial for PdM maintenance and asset management strategies. LSTM networks have emerged as powerful tools for RUL prediction, but their performance can be significantly impacted by the quality of the input data. This study delved into the impact of different data pre-processing techniques on the performance of LSTM model for RUL prediction, focusing on the C-MAPSS dataset, a benchmark for prognostics research.

Raw sensor data often exhibits inherent noise, non-stationarity, and various scales, which can hinder the LSTM model's ability to learn meaningful patterns and make accurate predictions. Data pre-processing techniques, such as normalization and filtering, play a critical role in preparing the data for effective model training and prediction. Normalization techniques like z-score normalization standardize the data within a specific range, ensuring that all features contribute equally to the model's learning process. Filtering techniques, on the other hand, aim to reduce noise and smooth out the data, allowing the LSTM model to focus on the underlying patterns rather than spurious fluctuations.

The study systematically evaluated the impact of various data pre-processing combinations on the LSTM model's performance. The results indicated that directly using raw data yielded suboptimal performance, with higher RMSE and lower PCC values. Conversely, employing normalization techniques consistently improved the model's performance, effectively scaling the data and enhancing its learning capabilities.

Among the normalization techniques, z-score normalization consistently demonstrated the best performance, reducing RMSE and enhancing PCC values. When combined with SG filtering, a technique specifically designed for time series data, the model achieved the lowest RMSE and highest PCC values, showcasing the synergistic effect of normalization and filtering. This dual approach effectively standardized the data and simultaneously reduced noise, leading to more accurate and consistent RUL predictions.

Beyond z-score normalization and SG filtering, the study also explored the effects of other filtering techniques, including wavelet transform and EMA. Both techniques demonstrated positive impacts on the LSTM model, further improving its ability to handle noisy data and produce reliable RUL predictions. Wavelet transform decomposed the data into different frequency bands, allowing the model to focus on the most relevant features, while EMA smoothed out short-term fluctuations and emphasized long-term trends.

The findings of this study underscore the significance on combination of data pre-processing in enhancing the performance of LSTM models for RUL prediction. By carefully selecting and combining appropriate normalization and filtering methods, researchers can significantly improve the accuracy and robustness of their models. Furthermore, applying these methods to other time series datasets and conducting comparative analyses can provide further insights into the optimal data pre-processing combinations for RUL prediction.

Author contributions

Second author contribution statement: Conceptualization, Methodology, Supervision, Validation, Review

References

1. Khaled, A., A Hybrid Deep Learning Based Approach for Remaining Useful Life Estimation, IEEE International Conference on Prognostics and Health Management (ICPHM), **2019**
2. Manuel, A.C., Fusing physics-based and deep learning models for prognostics, Reliability Engineering & System Safety, **2022**, 217, 107961
3. Zhu, K., Zhang, C., Data-driven RUL Prediction of High-speed Railway Traction System Based on Similarity of Degradation Feature, 9 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS), July 05-07, **2019**
4. Dong, D., Li, X.Y., Life Prediction of Jet Engines Based on LSTM-Recurrent Neural Networks, Prognostics and System Health Management Conference (PHM-Harbin), **2017**
5. Zhang, Y.Z., Xiong, R., A LSTM-RNN method for the lithuim-ion battery remaining useful life prediction, Prognostics and System Health Management Conference (PHM-Harbin), **2017**
6. Cui, J., Wang, Y., Prediction of Aeroengine Remaining Useful Life Based on SE-BiLSTM, 34th Chinese Control and Decision Conference (CCDC), **2022**
7. Xiaoxiong, W., Mingyang, P., and Chunxiao, X., Water Level Data Preprocessing Method Based on Savitzky-Golay Filter, International Conference on Modeling, Simulation and Big Data Analysis (MSBDA), **2019**
8. Lulu, W., Xiaoming, W., Hongbin, W., Data-driven SOH Estimation of Lithium-ion Batteries Based on Savitzky-Golay Filtering and SSA-SVR Model, IEEE 4th International Conference on Smart Power & Internet Energy Systems, **2022**

9. Lingyun, S., Ningyun, L., Xianfeng, M., Equipment Health State Assessment Based on MIC-XGBoost, The 13th Asian Control Conference (ASCC), **2022**
10. Honglin, L., and Wang, J., Based on Wavelet Threshold Denoising-LDA and Bilstm Aircraft Engine Life Prediction, Journal of Phys. Conf. Ser., **2022**
11. Rai, A., Upadhyay, S.H., The use of MD-CUMSUM and NARX neural network for anticipating the remaining useful life of bearings, Measurement 111, **2017**, 397–410
12. Harender, Dr. R. K. Sharma, EEG Signal Denoising based on Wavelet Transform, International Conference on Electronics, Communication and Aerospace Technology ICECA
13. Daubechies, I., Ten Lectures on Wavelets, SIAM **1992**
14. Kopuru, M.S.K., Rahimi, S., Recent Approaches in Prognostics: State of the Art, CSCE, **2019**
15. Nie, L., Xu, S., Remaining Useful Life Prediction of Aeroengines Based on Multi-Head Attention Mechanism, Machines, **2022,** 10(7):552
16. Yan, H., Zuo, H., Two-Stage Degradation Assessment and Prediction Method for Aircraft Engine Based on Data Fusion, Hindawi International Journal of Aerospace Engineering, **2021**:1-16
17. Ye, Z., Yu, J., Health condition monitoring of machines based on long short-term memory convolutional autoencoder, Applied Soft Computing, **2021**, 107:107379
18. Costa, N., Sánchez, L., Variational encoding approach for interpretable assessment of remaining useful life estimation, Reliability Engineering and System Safety 222, **2022**:108353
19. Han, J., & Kamber, M., Data Mining: Concepts and Techniques. Morgan Kaufmann, **2011**
20. Li, J., Jia, Y., Remaining Useful Life Prediction of Turbofan Engines Using CNN-LSTM-SAM Approach, IEEE Sensors Journal, **2023**, 23(9)
21. Olariu, E.M., Portase, R., Predictive Maintenance-Exploring strategies for Remaining Useful Life (RUL) prediction, IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP), **2022**
22. De Pater, I., Mitici, M., Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder, Engineering Applications of Artificial Intelligence, **2023**, 117:105582
23. Ruan, D., Wu, Y., Remaining Useful Life Prediction for Aero-Engine Based on LSTM and CNN, 33rd Chinese Control and Decision Conference (CCDC), **2021**
24. Zhu, Y., Liu, Z., Aircraft engine remaining life prediction method with deep learning, International Conference on Artificial Intelligence and Computer Information Technology (AICIT), **2022**
25. Kumari, S., Kumar N., and Rana, P.S., Comparative Performance Study of Different Filtering Techniques with LSTM for the Prediction of Power Consumption in Smart Grid, IETE Journal of Research, **2023**
26. Jongwoo, B., Filtering Correction Method and Performance Comparison for Time Series Data, Journal of information and communication convergence engineering **2022**, 20(2):125-130
27. Singh, N., Singh, P., Exploring the effect of normalization on medical data classification, International Conference on Artificial Intelligence and Machine Vision (AIMV), **2021**
28. Lima, F.T., A Large Comparison of Normalization Methods on Time Series, Big Data Research **2023,** 34:100407
29. Saxena, A., Goebel, K., Simon D., Eklund, N., Damage propagation modeling for aircraft engine run-to-failure simulation, International Journal of Prognostics and Health Management, **2008**, 1(1):9
30. Savitzky, A.G., M.J.E., Smoothing and differentiation of data by simplified least squares procedures, Analytical Chemistry, **1964**, 36(8):1627-1639
31. Hu, W., and Zhao, S., Remaining useful life prediction of lithium-ion batteries based on wavelet denoising and transformer neural network, Front. Energy Res., **2022**, 10:969168
32. de Miranda, A. R., de Andrade Barbosa, T.M., Conceiçao, A.G.S., Alcalá, S.G.S., Recurrent Neural Network Based on Statistical Recurrent Unit for Remaining Useful Life Estimation, 8th Brazilian Conference on Intelligent Systems (BRACIS), **2019**
33. Chevalier, G., LARNN: Linear Attention Recurrent Neural Network, **2018**
34. Hochreiter, S., Schmidhuber, J., Long Short-Term Memory, Neural Computation, **1997**, 9(8)
35. Berghout, T., Benbouzid, M., A systematic guide for predicting remaining useful life with machine learning, Electronics, **2022**, 11(7):1125