IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 August 2022, accepted 16 August 2022, date of publication 1 September 2022, date of current version 15 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203406

RESEARCH ARTICLE

A Deep Learning Model for Remaining Useful Life Prediction of Aircraft Turbofan Engine on C-MAPSS Dataset

OWAIS ASIF[1], SAJJAD ALI HAIDER[1], SYED RAMEEZ NAQVI[1], JOHN F. W. ZAKI[2], KYUNG-SUP KWAK[3], (Life Senior Member, IEEE), AND S. M. RIAZUL ISLAM[4], (Member, IEEE)

[1]Department of Electrical and Computer Engineering, COMSATS University Islamabad, Wah Campus, Wah Cantt 47040, Pakistan
[2]Department of Computer and Systems, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[3]Department of Information and Communication Engineering, Inha University, Incheon 22212, South Korea
[4]Department of Computer Science, University of Huddersfield, Huddersfield HD1 3DH, U.K.

Corresponding authors: Kyung-Sup Kwak (kskwak@inha.ac.kr) and S. M. Riazul Islam (s.mr.islam@hud.ac.uk)

This work was supported by the National Research Foundation of Korea-Grant funded by the Korean Government (Ministry of Science and ICT) under Grant NRF-2020R1A2B5B02002478.

ABSTRACT In the era of industry 4.0, safety, efficiency and reliability of industrial machinery is an elementary concern in trade sectors. The accurate remaining useful life (RUL) prediction of an equipment in due time allows us to effectively plan the maintenance operation and mitigate the downtime to raise the revenue of business. In the past decade, data driven based RUL prognostic methods had gained a lot of interest among the researchers. There exist various deep learning-based techniques which have been used for accurate RUL estimation. One of the widely used technique in this regard is the long short-term memory (LSTM) networks. To further improve the prediction accuracy of LSTM networks, this paper proposes a model in which effective pre-processing steps are combined with LSTM network. C-MAPSS turbofan engine degradation dataset released by NASA is used to validate the performance of the proposed model. One important factor in RUL predictions is to determine the starting point of the engine degradation. This work proposes an improved piecewise linear degradation model to determine the starting point of deterioration and assign the RUL target labels. The sensors data is pre-processed using the correlation analysis to choose only those sensors measurement which have a monotonous behavior with RUL, which is then filtered through a moving median filter. The updated RUL labels from the degradation model together with the pre-processed data are used to train a deep LSTM network. The deep neural network when combined with dimensionality reduction and piece-wise linear RUL function algorithms achieves improved performance on aircraft turbofan engine sensor dataset. We have tested our proposed model on all four sub-datasets in C-MAPSS and the results are then compared with the existing methods which utilizes the same dataset in their experimental work. It is concluded that our model yields improvement in RUL prediction and attains minimum root mean squared error and score function values.

INDEX TERMS Deep learning, long short-term memory networks, remaining useful life, turbofan engine.

I. INTRODUCTION

We are living in an era of industrial automation where our day-to-day activity depends heavily on a wide range of electrical and mechanical equipments varying from agriculture, process industry, power systems to the feild of transportation [1]. Every system requires maintenance operation at some point in time [2]. There are three types of maintenance techniques used in industries, reactive maintenance, preventive maintenance and predictive maintenance. In reactive maintenance repair operation is consider only when machine failure has occurred. In preventive maintenance the failure

The associate editor coordinating the review of this manuscript and approving it for publication was Fu-Kwun Wang.

can be prevented by performing the time-based maintenance operation, while the predictive maintenance (PdM) lets you estimate the time-to-failure of an equipment for scheduling the maintenance tasks [3]. PdM in contrast to the other two approaches needs to acquire various machine parameters for condition based monitoring (CBM) of industrial equipment. This technique focuses on forecasting the error by modeling the degradation trends between input sensors and time-to-failure duration of the machine. So, the benefits of this maintenance strategy is that we can eliminate unplanned downtime, reduced maintenance costs and maximize the machine lifetime for safety critical circumstances. One such example is aircraft engine which requires continuous monitoring of the engine performance. The fault diagnostics and prognostics of aircraft engine has gained great attention over the last few decades [4], [5], [6], [7]. One important component in aircraft engine maintenance is to accurately determine its remaining useful life (RUL) for reducing the maintenance costs while attaining the reliability [8], [9]. RUL prediction model is developed based upon the degradation trends among the various condition monitoring sensors. This model helps in development of maintenance strategy in a targeted manner to eliminate unplanned downtime and maximize machine lifetime for safety critical circumstances. Early anomaly detection and timely warning of a failure is vital for maximum utilization of the system. There are basically three types of prognostics techniques used for estimating RUL, physical model-based approaches [10], [11], data-driven approaches [12], [13] and hybrid approaches [14].

Model based approach initially required a comprehensive understanding of the physical architecture of the machine and then applying the laws of physics to obtain the mathematical model of the machine for RUL estimation [15]. Mathematical models often take some simplifying assumptions with uncertainty management for a complex industrial machinery, which can impose serious limitations on these techniques and hence degrade the RUL prediction accuracy [16].

Data-driven based prognosis approaches use various statistical and machine leaning (ML) algorithms to discover the trends or patterns in the underlying sensor data to estimate RUL of the system. These techniques are suitable for complex industrial machinery and further, it does not require a thorough understanding of a complete engine or the process. Hybrid method combines both the physics and data-driven based model techniques [17].

In the past decade, data-driven based prognostics methods have been exploited by many researchers. These models estimate the RUL by analyzing the degradation trend and target trajectory of sensor data. Deep learning methods like autoencoder, convolutional neural networks (CNN), long short-term memory (LSTM) networks and their varianta and combinations have achieved a massive success in the fields of computer vision, speech recognition, video segmentation and predictive maintenance [18]. The major drawback of deep learning algorithm is that it requires a large volume of data for offline training and in the field of prognostics, it is very challenging to gather run-time-to-failure sensor data especially for new machines. One way is that we can intentionally run a new system upto the failure mode but it is very prolonged, highly undesirable and expensive approach. Due to these limitations, researchers prefer some public datasets for the evaluation. In this work we have used commercial modular aeropropulsion system simulation (C-MAPSS) dataset which is basically a simulation of turbofan jet engine dataset provided by NASA prognostics center of excellence [19]. C-MAPSS dataset consist of four different multivariate time series units with different number of engines and each engine having different RUL. The dataset consists of twenty one sensors with three operating conditions with respect to time cycle for each engine.

In recent times, various work has been done on estimating the RUL of turbofan engine using deep learning methods such as CNN, LSTM along with their combinations and variants. LSTM networks have shown better results as compared to CNN based models [20], [21]. LSTM have shown excellent results because they are suitable for time-series data, they can learn the temporal features in multivariate system and minimize the root mean square error (RMSE) with respect to target predictions. In this paper, a LSTM based model has been proposed for RUL prediction of a turbofan engine. LSTM network can learn the association between target RUL values and sensor data but it alone cannot achieve state of the art performance due to various limitations like outliers, noise in the sensor values, un-normalized data and un-correlated sensor values. These shortcomings can reduce the performance of a LSTM network [22]. In this paper, we are focusing on implementing some preprocessing steps on the sensor data before it can be set as an input into the LSTM network. LSTM network when combined with effective pre-processing steps have the power to estimate the RUL with highly accuracy. These added steps involve correlation analysis, data filtering, normalization, and a modified piece linear degradation model for determining starting point of the degradation. It has been shown that the starting point of degradation which is also called the initial RUL has a great impact in determining accurate RUL predictions [23]. Our proposed modified piecewise linear degradation models help in efficiently calculating the starting point of degradation which in combination with the other pre-processing steps and LSTM network accurately predicts RUL for the given engines.

The main contributions of our work are enumerated as follows:

1) Novel piecewise linear degradation model for determining the starting point of engine degradation is proposed.
2) An LSTM network with effective pre-processing steps, i.e. correlation analysis with data normalization and moving median filter is proposed, which when augmented with the linear degradation model leads to an improved RUL prediction.
3) Hyperparameter for the proposed prediction model has been selected through iterative grid search based

approach [24] to further improve the accuracy of our framework.

The organization of remaining paper is given as follow. Section 2 gives the detailed literature review on the existing methods on turbofan engine RUL estimation. Section 3 explain the dataset taken for experimental testing. Section 4 discusses the proposed methodology with comprehensive block diagram that elucidate the entire work flow. Section 5 discusses the experimental results. Section 6 is the conclusion.

II. RELATED WORK

This section briefly reviews the existing literature on the turbofan engine RUL estimation. [25]. Traditional model-based techniques usually employ algorithms like Kalman filter (KF), extended Kalman filter (EKF) and particles filters to come up with mathematical formulation of machine based on multi sensor time series sequence data [26], [27], [28]. Classical degradation method such as Eyring model or Weibull distribution was implemented in [29]. Salahshoor *et al.* [30] used a unified framework of EKF based design for sensor data fusion algorithm to further enhanced the detection and diagnosis of degradation trends and system faults. Ordonez *et al.* [31] implemented the auto-regressive integrated moving average (ARIMA) model and support vector machine (SVR) methods collectively to estimate the RUL. The desired features can be created by analyzing prior learning about the degradation models as presented in [32]. In [33], it is suggested that failure thresholds or degradation state estimation is no longer required in learning-oriented approach. Khelif *et al.* [33] presented machine learning based support vector regression (SVR) model to project the direct association between multivariate sensor data or health index and the aircraft turbofan engine RUL.

Across all these techniques for turbofan engine RUL prediction, deep neural network-based methods have gained vast popularity. Zhang *et al.* [34] introduced a multi-objective evolutionary algorithm to expand and organized the deep belief network into multiple parallel networks simultaneously to accomplish the two convicting objectives i.e. diversity and accuracy. These networks attained a fine RUL prediction accuracy especially in case of complicated operations and in the presence of noise in input data [35], [36]. Saeidi *et al.* [37], proposed a naive Bayesian classification algorithm to measure the health index for turbofan engine. The pre-processing step takes the sensor data as input and apply moving average filter for removing the noise. It further categories the dataset into four different categories on the basis of time cycles i.e. time cycle values between 0 to 50 is labeled as urgent case which need immediate maintenance and further categorization is also done in a similar manner. Zheng *et al.* [38] proposed LSTM network combined with piece wise linear function for RUL for estimating the degradation trends. It achieves good results by applying piece wise linear function and data normalization. Wei *et al.* [39] proposed a Bi-LSTM network which can learn high level features in both direction and it can run training pass from forward to backward and backward to forward with back propagation algorithm. Wang *et al.* [40] proposed a hybrid network for turbofan engine in which trends and hidden pattern in long sequence sensor data is identified through LSTM network and short duration sequence was analyzed through time window method with gradient boosting regression (GBR). This method has two stages, offline stage to learn degradation pattern with LSTM network and TW-GBR used in online stage for extracting short sequence data. It also implemented standardization and sensor selection criteria. Babu *et al* [41], proposed deep CNN regression network for RUL estimation. The network consist of two dimensional convolutional layers for feature extraction followed by a fully connected regression layer for prediction. For 2-D convolution, first we have to convert our 1-D sensor data into 2-D with one dimension was taken as time and other taken as sensors amplitude. This model also extracts the spatial features very efficiently. Li *et al* [42], proposed deep convolutional neural network (CNN) for RUL estimation. The architecture of CNN is modelled in such a way that feature can be extracted from prepared 2-D sensor data by passing raw data into convolution layers, then flattened layer is added to convert extracted 2-D features into 1-D so that it can be given as an input to multilayer perceptron model with dropout layer for predicting RUL. ~~Jayasinghe *et al* [43], proposed temporal convolution in which combination of CNN-LSTM network was used for turbofan engine dataset. The layers of the model have stacked by first applying data augmentation to create similar type of data for avoiding overfitting followed by data normalization which was then followed by 1-D convolution for feature extraction, lastly fully connected layer act as the bridge between output of 1-D convolution layer and input of LSTM layer. LSTM layer was then followed by fully connected layer for output prognosis prediction.~~ Hong *et al.* [44], proposed a similar kind of network by stacking a 1-D convolution layer, residual layer, LSTM layer and a Bi-LSTM layer. Correlation analysis on sensor data for turbofan engine dataset was also performed. ~~Mo *et al.* [45], proposed multi-head neural network for RUL prediction of turbofan engine. This network is different from the series network in such a way that they have implemented the parallel branches of CNN layer in series with LSTM network. Furthermore, fisher method in combination with recursive least squares and single exponential smoothing was also employed to find the prediction error and given it as an additional input into CNN-LSTM head for optimum performance.~~ Zhao *et al.* [46], proposed an adjacnet neural network model for leanring the degradation pattern in a sensor data. The degaradation pattern mapping learns through morkov property i.e. estimating the next state of sequence with the assist of only present states.

Many researchers have used a piecewise linear degradation model in RUL prediction techniques. In this model, the starting point of the degradation is estimated often referred to as the initial RUL, many authors [47], [48], [49] have chosen its value on the basis of observations and no clear mechanism

of selecting it has been proposed. Lan *et al.* [23], proposed an LSTM algorithm for RUL prediction, it presented a piece wise linear degradation model. The dataset is divided into time windows and difference between geometric distance (Euclidian) of each window is used to determine the value of initial RUL or the starting point of degradation. They have validated their model on only FD001 sub-unit of the dataset and selection of window size and thresholds are also not explained. Inspired by the results of this model, we have proposed an improved version of the algorithm, which is applied to the complete dataset and results in overall increased in the accuracy of our prediction model.

The existing methods presented so far showed impressive performance on RUL tracking; our proposed method further improves the performance of these networks – summarized in the results and discussion section. This is mainly due to the fact that we have incorporated and implemented effective data preprocessing steps into our prediction pipeline. In the previous works, pre-processing stages have not been completely exploited in order to find the right combination for enhancing the learning capability of deep learning algorithms. So, we propose one such combination along with the training of LSTM with a hyperparameter selected through grid search algorithm in order to achieve the optimum performance.

A. PROBLEM STATEMENT

The predictive maintenance techniques is different from the reactive and preventive maintenance techniques in terms of safety, reliability, efficiency and unnecessary downtime for aircraft turbofan engine [50]. These methods ensure reliable solution managing the health of turbofan engine to reduce the downtime, which leads to significant loss in revenue. Therefore, failures in a turbofan engines can cause catastrophic accidents due to its sensitive nature and it needs to be estimated prior in time so that we can provide maintenance services in order prevent any fatal incident.

In current scenario, due to the implementation of cyber-physical system (CPS, that link the cyber world with a physical world, called smart manufacturing), the industrial sector such as health care, nuclear power plant etc. generate enormous volume, velocity, veracity and variety of data. Therefore, with the rise of AI and availability of hardware computing resources, data driven based artificial intelligence (AI) predictive maintenance models have a capability to process big amount of real-world machines data with ease and predict heath index of aircraft turbofan engine in time before failure to prevent unwanted breakdown [51].

III. DESCRIPTION OF C-MAPSS TURBOFAN ENGINE SIMULATION DATASET

C-MAPSS dataset released by NASA is developed in MATLAB environment as a tool for simulation of turbofan engines. C-MAPSS dataset was published in 2008 for 1st International conference on PHM [19]. This dataset was published some time ago but still it has been actively used in research for evaluating the prognostics model with a focus on accurate estimation of RUL. This model have fourteen input parameters related to five rotating components of engine to simulate different degree of fault and deterioration of the model. A total of twenty one variables out of fifty eight different sensor responses is considered from the model for predicting the RUL. The three operating parameters of C-MAPSS simulation model are given in table-1 and the details of *21* sensors are given in table-2. The legends of last column in table-2 *Trends* indicates the degradation pattern of sensor data with respect to time, where "$\sim$" represents irregular sensor behavior, *I* represent the parameter increasing with time and lastly, *D* is the variation of parameter that decreases with time.

TABLE 1. Operating parameters of C-MAPSS.

Parameter	Operating Range
Mach Numbers	0 to 0.90
Altitude	Sea level to 40,000 feet
Sea-level temperature	-60 to 103° F

The main components of turbofan engine include nozzle, low pressure turbine (LPT), high pressure turbine (HPT), fan, low pressure chamber (LPH) and high pressure chamber (HPC). There are total of fourteen editable input parameters such as fuel flow, HPC efficiency modifier, LPT efficiency modifier etc. that allows you to simulate various operating behaviors. C-MAPSS sensor trajectories are further divided into four different units namely FD001, FD002, FD003, and FD004 corresponding to different operating conditions and fault modes. This dataset contains 709 engines for the training and 707 engines for testing which are of same type but with distinct manufacturing variation and initial wear, unknown to the researcher. The description of four sub dataset units with train and test trajectories and other details are given in table-3. In the start, all the engines in each sub-dataset are operating normally as seen from sensor behavior and originate the fault sometime later in their life cycle. In training sequence, complete run-to-failure data is available with a specified RUL labels as faults grows in the system and in test time degradation values are given up to some time prior to engine failure. Moreover, with different initial health conditions, there are distinct number of time cycle even for the same engine in dataset. The objective of this dataset is to predict remaining useful life cycle of engines in each sub-unit. The actual RUL label are given in the test data, which is used to validate the prediction results.

It can be observed from table 3 that different sub-units of the main dataset have different running life cycle time e.g. in FD004 test data, the maximum life cycle time is 486 and minimum time 19.

IV. PROPOSED METHODOLOGY

This paper proposes LSTM based RUL prediction model for turbofan engines, which proves to be more robust than most of the existing models available in the literature. The increased

TABLE 2. Output parameters C-MAPSS turbofan engine datasets.

Sensor	Parameter	Description with units	Trend
1	T2	Total Temperature in fan inlet (oR)	~
2	T24	Total Temperature at LPC outlet (oR)	I
3	T30	Total Temperature at HPC outlet (oR)	I
4	T50	Total Temperature at LPT outlet (oR)	I
5	P2	Pressure at fan inlet (psia)	~
6	P15	Total pressure in bypass-duct (psia)	~
7	P30	Total pressure at HPC outlet (psia)	D
8	Nf	Physical fan speed (rpm)	I
9	Nc	Physical core speed (rpm)	I
10	Epr	Engine pressure ratio (—)	~
11	Ps30	Static pressure at HPC outlet (psia)	I
12	Phi	Ratio of fuel flow to Ps30 (psi)	D
13	NRf	Corrected fan speed (rpm)	I
14	Nrc	Corrected core speed (rpm)	D
15	BPR	Bypass ratio (—)	I
16	farB	Burner fuel air ratio (—)	~
17	htBleed	Bleed enthalpy (—)	I
18	NF-dmd	Demanded fan speed (rpm)	~
19	PCNR-dmd	Demanded corrected fan speed (rpm)	~
20	W31	HPT coolant bleed (lbm/s)	D
21	W32	LPT coolant bleed (lbm/s)	D

TABLE 3. Each sub-unit of C-MAPSS dataset.

Four-units of dataset	FD001	FD002	FD003	FD004
Engines in training set	100	260	100	249
Engines in testing set	100	259	100	248
Training trajectories	17731	48558	21120	56815
Testing trajectories	100	259	100	248
Max/min cycle for train	362/128	378/128	525/145	543/128
Max/min cycle for test	303/31	367/21	475/38	486/19
Operating Conditions	1	6	1	6
Fault Modes	1	1	2	2

robustness stems from the fact that we have integrated efficacious techniques from multiple alternatives and presented two types of improvements to be made in the LSTM based prediction model. The first type of improvement is the addition of more effective pre-processing steps, which in return puts a great influence on LSTM training. Secondly, enhancing the LSTM training procedure by a grid search approach for computing the effective hyper parameters. This framework manages to achieve higher prediction accuracy, measured in temrs of RMSE and score function values, than several existing works − see Section 5 for details. The framework of our proposed model is shown in Fig. 1. The specific steps of the prediction model are explained below.

A. CORRELATION ANALYSIS

First, we must prepare our health-to-failure data into an appropriate form for improving the accuracy of the LSTM network for effective training operation. C-MAPSS dataset consists of three operational settings and twenty one sensor signals of engines with machine life span length. These signals are then given as an input to correlation analysis method [31], [44], [52], [53] to discover the relevance of features with RUL. The algorithm excludes the sensor values which have a very little or zero correlation with RUL, this includes some parameter in engines that are basically controlled by a feedback controller and results into a near constant values or having an oscillatory behavior. These kinds of parameters do not play much part in RUL predictions. So, the selected feature signals are then given as an input into the data filtering stage. Statistical evaluation of turbofan engine degradation dataset gives us certain insight into the multivariate sensor data and furthermore reach towards the conclusion that whether a considered sensor is adequate for training the network or not. We can accomplish this abstraction by computing the value of correlation coefficient *'r'* which is a relationship between the sensors and RUL labels.

$$r = \frac{conv(x, y)}{s_x s_y} \tag{1}$$

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})(\sum_{i=1}^{n}(y_i - \bar{y}))}} \tag{2}$$

where conv (*x*,*y*) is covariance between input sensor data (*x*) and output RUL label (*y*), $\bar{x}$ and $\bar{y}$ are the mean of input sensor data and RUL label, *n* is the number of variables in a dataset and S_x & S_y are standard deviation of the two signals *x* and *y* respectively.

In [44] a correlation analysis is employed for dimensionality reduction to obtain the accurate results and to reduces the complexity of sensor data. This technique is primarily limited to FD001 sub-unit of C-MAPSS dataset, we have extended this approach to entire degradation dataset and comprehensively investigate the trends and irregular behaviors by analyzing the correlation matrix heat map [54] of C-MAPSS turbofan engine dataset. The correlation matrix heat map cells show the association of three operating settings and 21 sensors with output RUL labels as shown in Fig. 2. The correlation matrices are converted into percentage with dark green color representing higher correlation as opposed to light color which depict a low correlation value. The number of sensors selected from each sub-unit after the correlation process are given in table 4. After analyzing the correlation coefficient of sensor degradation dataset, we have concluded that 14 out 24 parameters in FD001 and 16 out of 24 parameters in FD003 reflect high strength of correlation with defined or monotonous behavior and we have omitted those variables that indicates less than 10% of correlation with the output RUL label. For the case of FD002 and FD004, correlation coefficient values in a heat map illustrated that these variables have little degree of association with output RUL labels and

TABLE 4. Sensors selection based on correlation matrix.

Correlation(r)	FD001	FD002	FD003	FD004
Correlation up to 25%	10	24	9	24
Correlation between 25 % to 50%	0	0	1	0
Correlation between 50 % to 100%	14	0	14	0
Sensor selected for training LSTM Model	14	24	16	24

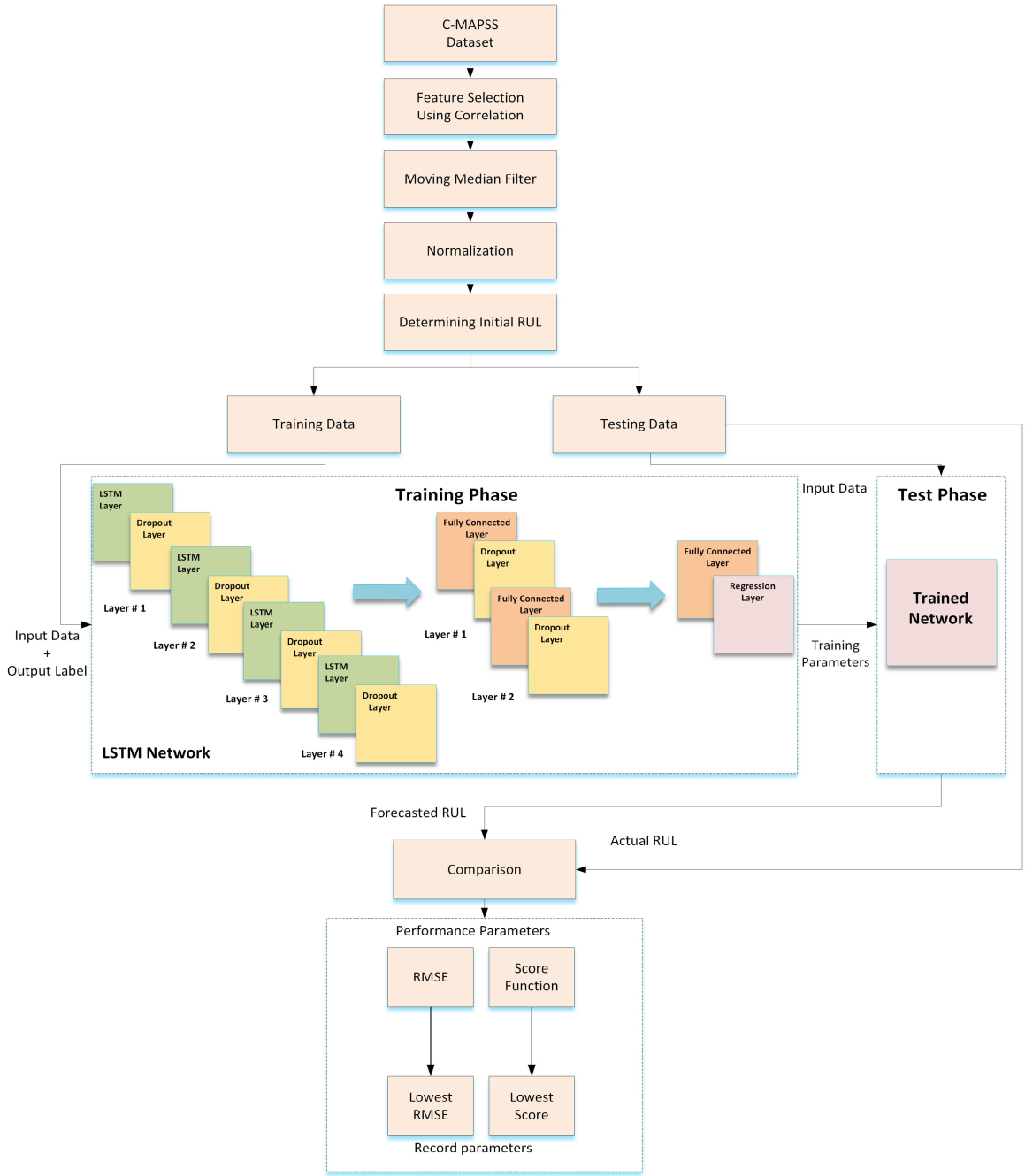

FIGURE 1. Proposed Framework for RUL predictions.

this correlation is more than 15% for best possible case. After analyzes the correlation matrix, un-correlated sensors are removed from the training process which are listed in table 5. So, only highly correlated sensor data are used for training the LSTM network which also increases the speed of learning for estimating RUL.

B. FILTERING

The correlated data is passsed through a moving median filter for removing the outliers and noises in the sensors data. The choice of filter is made on its ability in removing the outlier while preserving the high and low frequency contents in sensor data and avoids any loss of data values. The time window size of moving median filter is adaptive and vary with respect to sensor values. The moving median filter belong to a type of non-linear digital filter, which is used to remove random unwanted noise especially when there is a high spike and short-term outlier present in the data points but preserving the high frequency information contents [55]. Median filter is used to identify such sensor values in turbofan engine which

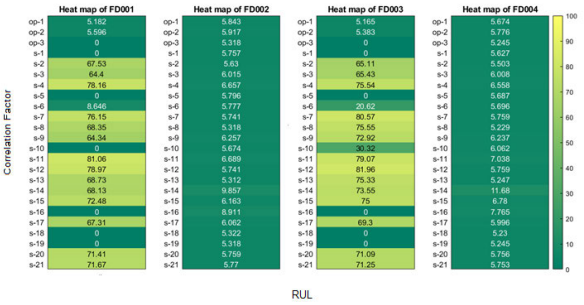

FIGURE 2. Correlation Matrix of C-MAPSS Dataset.

TABLE 5. Excluded sensor list.

C-MAPSS Sub-unit	Excluded Sensor List	Sensor Selection Criteria
FD001	op-1,op-2,op-3, s1, s5, s6, s10, s16, s18, s19	Correlation (r) > 10%
FD002	———	Correlation (r) > 5%
FD003	op-1,op-2,op-3,s1, s5, s16, s18, s19	Correlation (r) > 10%
FD004	———	Correlation (r) > 5%

are considerably different from the other data points and then eliminate them. The algorithm of moving median filter is accomplished by sliding the window of appropriate length over the sensor data entries by entries and then replacing the corresponding values by computing the median of neighboring points specified in the window. Mathematically it can be expressed as:

$$y_i^j[n] = median(x_i^j[n], \ldots, x_i^j[n+T]) \tag{3}$$

where n = total number of sensor data points in each engine, i = sensor variable, j = engines, $x_i^j[n]$ denotes input data for sensor i in engine j and $y_i^j[n]$ returns the medians values across each sensor variables from FD001 to FD004 with a same dimension as of input data.

The algorithm of filter is accomplished by sliding the window of length T over the neighboring elements in sensor data and computed the median for each considered window and median operation is then performed on these array vector to get the filtered output after applying ascending operation. So this processed data is then given to the next stages and hence put a significant impact on the output.

C. DATA NORMALIZATION

The range of sensor output after analyzing from the graphs is from tens to thousands and if we use these raw values for training the network then accuracy will drop significantly [56]. The filtered signals are normalized to have same degree of range for efficient training of the network. Z-Score normalization [57] is used in this paper which first compute the mean (μ) and standard deviation (σ) of each feature vector and then apply the following operation on each sensor output.

$$y_i^n = \frac{x_i^c - \mu^n}{\sigma^n} \tag{4}$$

where, y_i^n is the normalized value at i^{th} time cycle for sensor n, μ^n is the mean value of all output of sensor n, σ^n is the standard deviation of all n sensor output.

D. IMPROVED PIECEWISE LINEAR DEGRADATION MODEL

It is observed that RUL is linear decreasing function with respect to time as the efficiency of the system degrades. However, as the system starts their operation, there is no degradation present in the sensor readings. This pre-processing step is basically implemented on output labelled data that takes input from previous correlation analysis stage and employs a piece wise linear degradation function on sensor values for finding the initial RUL or the starting point of degradation. All the labels till this deterioration point are constant out to the initial RUL value while the remaining RUL lables are represented as a linear line from that degrdation point up to zero life cycle time.

In this paper, we have presented an improved version of the automatic piece wise linear function [23] for output RUL labeling. This approach is self-governing that is sensitive on variation of the degradation trends and will automatically calculate the early point of sensor deterioration. The computation of initial RUL starts by dividing entire sensor time cycle with non-overlapping pattern into equal sized window length of w and extract the sensor data from a given windows. We then calculate the centroid of each considered window by determining their mean values and geometric distance calculation is performed by subtracting the two subsequent windows to generate the trends in sensor data. As there are N number of time cycles for given variable and window length of w results into $(g=N/w)$ geometric points for a given dataset. These geometric distances are computed using Euclidean distance method which is then squared and the degradation pattern from g values is evidently detectable from the resulting plot as shown in Fig. 3. The centroid of window w_1 is first computed and subtracted from the other windows in a sequence to compute the variation in sensor values to reach on a point of deterioration based on the threshold value. The inflection point of the curve indicates the increase in sensor trends which is the initial RUL value.

The proposed algorithm is given as Algorithm-1 is implemented for each engine. The minimum value of initial RUL among all the engines in a sub-unit is taken as the initial RUL for that sub-unit. Threshold level is dependent on the rate of rise in raw sensor data, its visual perception and how early we need to predict the faults in the machines for maintenance purpose.

In this paper, we have set different ranges of threshold $(0.01 \text{ to } 0.2)$ and window size $(5,12)$ for calculating the initial RUL and validating the performance of our model. We have used different values of window sizes in order to compute the knee point in sensor data effectively. This choice stems

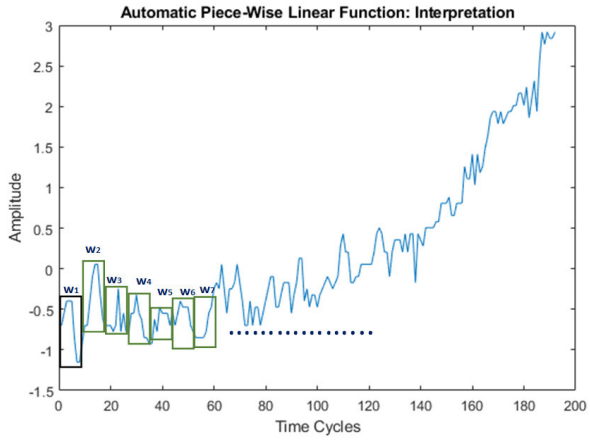

FIGURE 3. Degradation process represented in the form of windows of time cycle.

Algorithm 1 Improved Piece-Wise Linear Function for Initial RUL Calcultion for Each Engine

Inputs: Time cycle values (tc) and Sensors data after filtering and normalization
Parameters: w = window length, g = total number of windows, w_1,$w_2 \ldots w_g$ = subsequent windows, Th=threshold
Output: Initial RUL label ($irul$)
e = Extract sensors values from the given windows (w_1,$w_2 \ldots w_g$)
m = Calculate the centroid of each window by computing the mean for each window
for $i = (2\ to\ g-1)$ **do**
    s = Subtract the mean of two windows (w_1,w_i)
    sq = square (s)
    if $sq >= Th$ **then**
        $irul = tc - w * i$
    else
        $i = i + 1$
End

from the fact that by further increasing the window size, the algorithm bypasses the knee point, which is the starting point of increasing/decreasing trend in sensor variables. Fig. 3 confirms that the trend in sensor data changes within few cycles, and if we further increase the window size then we cannot achieve true inflection point in these variations. As a result, initial RUL computation will not reflect the ideal knee point in the sensors' behavior.

E. LSTM MODEL DEVELOPMENT

The data from the normalization stage with updated RUL labels from the degradation model is used to train a deep LSTM network. Our proposed LSTM model consists of LSTM layers, dropout layers, fully connected layers and the regression layer. The output node of fully connected layer is a regression layer that gives the estimated RUL of turbofan engine. Our proposed model consists of four layers connected

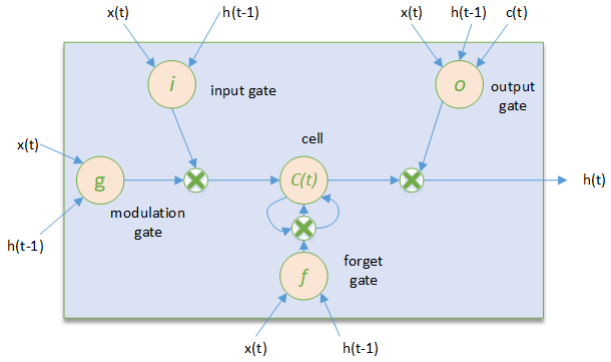

FIGURE 4. Structure of a LSTM cell.

in a sequential manner with different number of hidden units and a dropout layer is also added in between the LSTM layers for enhancing the generalization of network to avoid over fitting. It is then followed by two fully connected layer with dropout layers and the final layer is the regression layer as shown in Fig. 1. Fig.4 shows a basic LSTM cell that is essentially consist of three control gates: input gate, forget gate and output gate. The output of the cell is denoted by h_t, which is a short-term memory sate in a network and C_t is considered as a long-term cell state. The first gate in an LSTM cell is forgot gate f_t, which is used to unlearn selective information stored in previous LSTM cell. The forget gate equation is given below.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{5}$$

where $\sigma\ (\ )$, is called sigmoid activation function, which can control operation of forget gate. W_f is the weight matrix, h_{t-1} short term state from previous cell, x_t is the input of cell, and b_f is the bias vector of LSTM cell. The input gate controls the new information entering into the cell through following two equations:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{6}$$

$$\bar{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{7}$$

where $tanh = (e^x - e^{-x})/(e^x + e^{-x})$. This value is calculated by the same short term state vector h_{t-1} which is further used to update the new state of cell. W_i and b_i is the weight matrix and bias vector of an input gate respectively. The $\bar{C}_t$ computed from above equation is first filtered by it and then added to the long term state of the cell. W_c and b_c are the weight matrix and bias vector. After computing the value of forget gate (f_t), input gate (i_t) and ($\bar{C}_t$), long term state C_t of LSTM cell is updated after applying given below matrix operation

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \bar{C}_t \tag{8}$$

where, $\otimes$ is basically element wise matrix multiplication operation between a specified variable and C_{t-1} is the previous state of LSTM cell. Finally the output of LSTM cell is generated by the following two equation,

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t \otimes tanh(C_t) \tag{10}$$

The output state of LSTM cell h_t, is obtained by filtering the output gate o_t equation with a matrix of C_t. W_o and b_o are the weight matrix and bias vector of the output equation. The values of the weight matrix and biases are computed by training the LSTM network.

F. DROPOUT LAYER

The dropout layer is added to avoid the overfitting which inherently occur while training the deep neural network [58]. This regularization layer is added in between the fully connected layers and LSTM layers to increase the generalization of the whole algorithm to better track the predicted RUL with high accuracy. It drops some random portion of neuron according to the probability parameter defined in a network while remaining weights are trained by the backpropagation algorithm [59]. The expression for a dropout layer by choosing a linear activation function and considering the activity in unit i at layer h is given by the expression.

$$S_i^h(I) = \sum_{I<h} \sum_j w_{ij}^{hl} S_j^l \;\; with \; S_j^0 = I_j \tag{11}$$

where w is the weight and I is the input vector.

G. FULLY CONNECTED LAYER

The fully connected layers gets the data from the final combination of LSTM and dropiut layer, so that the features extracted from LSTM layer are used to generate the output [60]. Due to its fully connected nature between all the neuron present in the network, it has a large amount of weight parameters which needs to be computed by training the network. Fully connected layer along with a dropout layer is followed by single regression layer for predicting the RUL. The mathematical calculation of fully connected layer is expressed below.

$$H_0 = I \tag{12}$$

$$H_l = \phi(h_{(l-1)} W_{fc} + b_{fc}) \tag{13}$$

where I is the input vector, ϕ is activation function of a neuron which is primarily a ReLu fucntion, h_{l-1} is output from the previous layer, W_{fc} is weight matrix of a specified fully connected layer, b_{fc} is the bias vector and H_l is output at l^{th} layer.

H. INTERPRETATION OF OUR APPROACH

We implement deep neural network combined with per-processing steps for efficient RUL prediction of turbofan engine. Furthermore, we have separately discussed the above-mentioned stages of prediction pipeline breifly in order to delineate a comprehensive understanding of our work. Therefore, to summarize our approach, we have presented a pseudo code of our proposed pipeline in Algorithm 2.

I. HYPER PARAMETERS SELECTION

LSTM training process involves many different parameters. These parameters have a great impact on the accuracy of the model. These parameters include learning rate, batch size, number of layers, number of neurons in each layer, optimizer and the training epochs. There is no definite rule or process by which the best values of these parameters are selected, generally an iterative grid search method is used to get the best results. The model is run against different values of these parameters and combination of values with best result are recorded. Addition of more layers makes system more complex and timing consuming for training but could result in better predictions as compared to the single layer network which are more suited to the simple problems. The ranges/types of hyper parameters used for LSTM network training are given in table 6.

Algorithm 2 Proposed Methodology

Inputs: Training Data,$\{\delta_T = (x_1^1, y_1^1), (x_2^1, y_2^1) \ldots (x_N^E, y_N^E)\}$
Parameters: $N \leftarrow$ No. of sensor variables, $E \leftarrow$ No. of Engines, $\delta_N \leftarrow$ Filtered data, $\delta_C \leftarrow$ Correlated sensors, $M \leftarrow$ No of correlated sensors, $\delta_D \leftarrow$ Normalized data, $\delta_P \leftarrow$ Piece-wise linear RUL function, $L_N \leftarrow$ No. of layers in LSTM, $L_D \leftarrow$ Doprout layers, $R_L \leftarrow$ Regression layer, $L_F \leftarrow$ No. of neuron in FC layer, $W \leftarrow$ Window Length, $\alpha \leftarrow$ Learning rate, $G_D \leftarrow$ Gradient descent optimizer, $V_s \leftarrow$ Validation set, $S_{arc}^t \leftarrow$ Network architecture parameters
Output: Performance Parameter,$O_{pp} \leftarrow$[RMSE,Score]
for $i = (1\ to\ E)$ **do**
    for $j = (1\ to\ N)$ **do**
        δ_C = Correlation Analysis (x_j^i, y_j^i)
    end for^j
end for^i
for $k = (1\ to\ M)$ **do**
    δ_N = Median Filter(δ_C^k,W)
    δ_D = Data Normalization(δ_N^k, mean(δ_N), SD(δ_N))
    δ_P = Piece-wise linear function(δ_C^k)
end for^k
$S_{arc}^t \leftarrow (L_N, L_H, L_F, \alpha, G_D, L_D, V_s, R_L, \delta_D, \delta_P)$
for $l = (1\ to\ T)$ **do**
    F_D =Forward Pass (S_{arc}^t)
    E_t = Error(F_D)
    B_p = Back Propagation(E_t)
    $W_{new} = W_{old} + \alpha$*derivative($B_p$)
    $b_{new} = b_{old} + \alpha$*derivative($B_p$)
end for^l
$O_{pp}^i \leftarrow$ Calculate RMSE using equation 14.
$O_{pp}^{i+1} \leftarrow$Calculate score function using equation 15.

J. EVALUATION METRICS

After network development, training of LSTM network is carried out and selection of hyper parameters is made on the basis of root mean square error (RMSE) between actual RUL and predicted RUL for each engine of the turbofan dataset. Our proposed LSTM model is evaluated using two widely used evaluation methods: RMSE and scoring function.

TABLE 6. Ranges/types of hyper parameters in LSTM model training.

Hyper parameter	Range
Learning Rate	*0.01 to 0.05*
Mini-Batch Size	*10 to 30*
Max Epoch	*10 to 300*
No. of LSTM layers	*2 to 8*
No. of neurons in each LSTM layer	*30 to 100*
Dropout probability	*0.1 to 0.5*
Optimizer	*Adam, Stochastic Gradient Descent, RMSProp*

RMSE is the most used evaluation metric for predictions used by the researcher for past decades. It is a symmetrical scoring function which means it can assign equals weights or penalties to both early and late prediction. The equation of RMSE is given below.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N}} \tag{14}$$

where e_i, is the prediction error and N is the total number of samples. In PHM08 data challenge competition, a score function was employed to evaluate the performance of the prediction model [61]. It is an asymmetrical score function which means it can assign more weights to late prediction as opposed to early prediction. It is described mathematically as:

$$score = \sum_{j=1}^{N} s_j \tag{15}$$

$$s_j = \begin{cases} e^{\frac{-h_j}{13}} - 1, & \text{if } h_j < 0 \\ e^{\frac{-h_j}{10}} - 1, & \text{if } h_j \geq 0 \end{cases} \tag{16}$$

The performance of the model will degrade at greater extent in case of delayed prediction as compared to the early prediction. The prediction error between the actual and predicted RUL is close to zero then the value of score and RMSE is also smaller with advancing towards zero.

V. RESULTS AND DISCUSSION

In this section, a detail analysis on RUL prediction results by our proposed model is presented. There are various parameters that affects the model's accuracy. We can categorized them into two sets, first are the parameters related to the initial RUL value, these includes window size and threshold values. Second ones are LSTM hyper parameters. Combining these two sets of parameters gives a very large set of parameters affecting the model's accuracy. It is evident from the existing literature on deep learning techniques that there is no straight forward way of determining a single best set of values for these parameters. This requires very extensive and detailed simulations to analyze it further and an iterative grid search approach is followed. The simulation model is developed in MATLAB and hundreds of runs were carried out against the parameters ranges defined in the last section. For comparison,

TABLE 7. Selected LSTM models with hyper-parameters values.

LSTM Models	Hyper-parameters						
	Learning Rate	Mini-Batch	Max Epoch	No. of LSTM Layers	No. of Neurons in each LSTM Layer	Dropout Probability	Optimizer
Model 1	0.01	10	20	2	30	0.1	Adam
Model 2	0.02	15	150	4	60	0.1	Adam
Model 3	0.001	20	250	6	90	0.1	Adam
Model 4	0.001	25	300	8	100	0.1	Adam

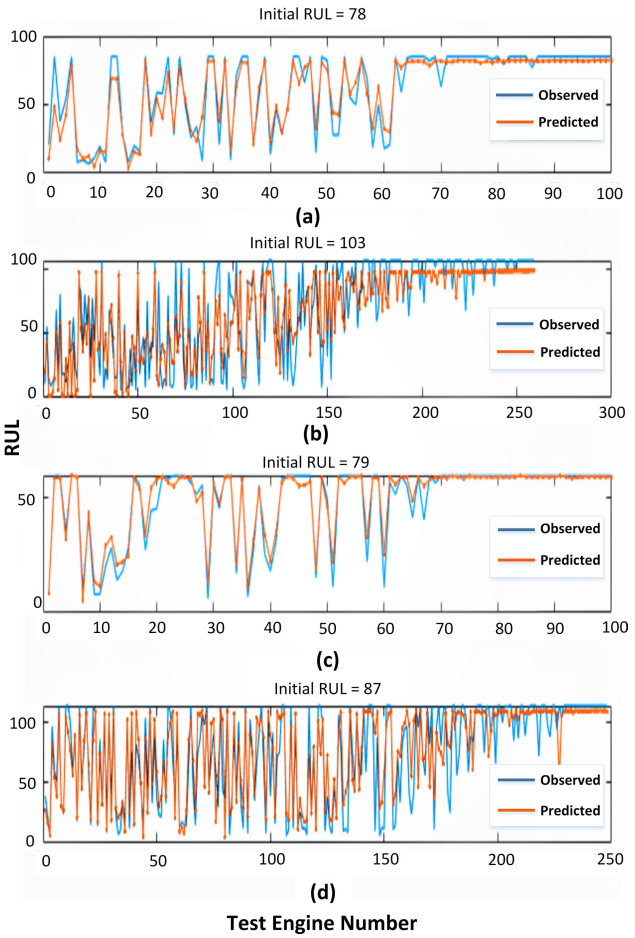

FIGURE 5. RUL Prediction on Test Set, (a) FD001 (b) FD002, (c) FD003, (d) FD004.

four LSTM models which gave us encouraging results were further tested and tuned for best performance. The models along with the hyper parameters are given in table-7, LSTM model structure remains same as described in the Fig.1. The output of the pre-processing stages which processed sensor data and RUL output labels are given as an input into the LSTM network. The network is trained using the different values of the hyper parameters given in the table-7 and the window size and threshold ranges defined in the last section. After following an iterative grid search approach, the best hyper parameters are selected for the proposed model on the

TABLE 8. Model accuracy in term of RMSE and Score function values for the four LSTM models.

Window Size	FD001				FD002				FD003				FD004			
	Threshold															
	0.1		0.2		0.1		0.2		0.1		0.2		0.1		0.2	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
LSTM Model 1																
5	18.04	451	15.63	290	22.74	2448	22.83	3006	17.54	716	15.93	414	27.61	10533	28.76	7632
8	15.85	342	14.1	225	22.70	2447	22.74	2347	15.6	391	12.87	222	27.86	7138	22.64	6804
10	16.07	324	13.86	254	22.04	2330	20.47	1431	15.96	550	12.68	204	24.76	7135	19.96	6168
12	15.7	335	12.1	206	21.60	2207	18.50	1319	15.77	440	11.06	158	20.76	3257	**17.63**	**2395**
LSTM Model 2																
5	17.96	879	17.09	472	30.07	20902	32.49	9876	19.33	522	12.45	292	30	27246	30.11	20228
8	17.75	606	9.83	144	30.01	20903	19.69	4719	12.06	258	10.60	234	28.67	26596	30.7	23291
10	15.75	362	8.60	112	21.45	5508	18.58	2460	12.31	264	9.98	142	26.35	23432	24.69	8752
12	15.71	361	**7.78**	**100**	22.32	2984	**17.64**	**1443**	11.94	230	11.60	189	27.86	29082	20.93	2530
LSTM Model 3																
5	18.01	746	13.99	325	29.2	7347	29.20	7349	16.16	1022	14.86	525	39.81	57200	39.53	45067
8	15.89	390	14.36	434	29.21	7346	28.11	6420	12.69	372	9.20	137	38.10	37646	34.48	16586
10	14.60	323	9.83	156	29.32	9195	26.81	4986	12.72	374	9.07	135	38.73	47543	32.65	13664
12	14.59	322	9.38	152	28.34	6428	26.76	5697	11.80	279	**8.30**	**104**	33.43	18125	29.32	16852
LSTM Model 4																
5	20.63	959	15.41	409	30.05	9270	29.98	9271	16.36	600	15.10	510	39.40	38185	39.21	37811
8	16.79	516	13.28	266	29.93	9188	28.52	7650	13.85	329	8.96	128	39.99	69027	36.76	36792
10	13.68	304	9.32	143	29.06	7685	26.99	5378	13.99	333	8.78	124	38.72	50620	33.27	15922
12	13.67	303	10.10	181	28.53	7648	26.63	5239	13.83	371	8.66	126	35.07	28303	30.26	9694

basis of lowest RMSE and score function values. The results of the these four LSTM models against different thresholds and window sizes values are given in table 8.

The results in table-8 show that there is no single set of best values parameters for the whole dataset. The best values of parameters for RMSE and score function are different for each sub-unit. For FD001 case the best results are with using LSTM Model-2 and window size and threshold values of 12 and 0.2 respectively. Similarly, for FD002 best results are with using LSTM Model-2 and window size and threshold values of 12 and 0.2 respectively. For FD003, best results are with LSTM Model-3 and window size and threshold values of 12 and 0.2 respectively and for FD004, best results are with using LSTM Model-1 and window size and threshold values of 12 and 0.2 respectively. It can be observed that although LSTM model hyper parameters are different for different sub-units, the best values of window size and threshold are same. The minimum initial RUL value among all engines in a sub dataset is selected as its initial RUL. The initial RUL values achieved from the training part of sub-units FD001 to FD004 are 78, 103, 79 and 87 respectively. These values are then used in the testing stage for each sub-unit. So, instead of using a single initial RUL value for the entire dataset which is used in most of the existing literature, we can achieve improve prediction accuracy by using separate initial RUL value and LSTM hyper parameters values for each sub-unit.

The prediction results using the best models are shown in Fig. 5, which shows highly accurate RUL predictions. The *x*-axis denotes the number of engines in a specific sub-unit and *y*-axis is the result of RUL prediction for each engine. The red and blue legends on the graph indicate the predicted and true value of RUL. Moreover, these results are further

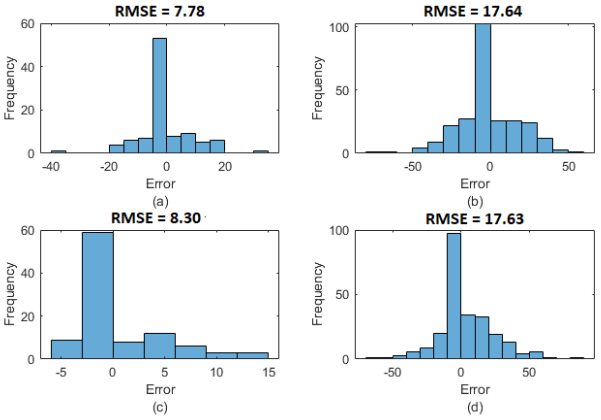

FIGURE 6. Histogram Distribution of Error, (a) FD001, (b) FD002, (c) FD003, (d) FD004.

observed by histogram distribution of prediction error in Fig. 6.

The histogram distribution of four sub-units indicates the variation of RMSE across the dataset. The *x*-axis indicates the error or difference between predicted observation and true RUL while the *y*-axis indicates the frequency of occurrence for the given error. It can be seen from the figure that large concentration for frequency of error lies in the range of *[-5 0]* in FD001 & FD003 while for FD002 and FD004, it is concentrated in *[-10 0]*. The data description in section III shows that FD002 and FD004 has 6 operating conditions with more than 200 tracking trajectories. In correlation analysis, we have demonstrated that these two sub-units possess irregular behavior so RUL prediction for these complex sequences is a challenge for the prediction model. FD001 has

TABLE 9. Comparison of RMSE and score with other methods.

Methods	Year	Pre-Processing Steps	FD001		FD002		FD003		FD004	
			RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
CNN [41]	2016	Data normalization, RUL target function	18.44	$1.29\text{x}10^3$	30.29	$1.36\text{x}10^4$	19.81	$1.60\text{x}10^3$	29.15	$7.89\text{x}10^3$
LSTM [38]	2017	Data normalization, RUL target function	16.14	$3.38\text{x}10^2$	24.49	$4.45\text{x}10^3$	16.18	$8.52\text{x}10^2$	28.17	$5.55\text{x}10^3$
BiLSTM [62]	2018	Feature selection, Data normalization, RUL target function	13.65	$2.95\text{x}10^2$	23.18	$4.13\text{x}10^3$	13.74	$3.17\text{x}10^2$	24.86	$5.43\text{x}10^3$
DAG [63]	2019	Feature selection, Data normalization, Piece-wise function	11.96	$2.29\text{x}10^2$	20.34	$2.73\text{x}10^3$	12.46	$5.35\text{x}10^2$	22.43	$3.37\text{x}10^3$
CNN+LSTM [64]	2019	Variance threshold, Data normalization, Health indicator	16.16	$3.03\text{x}10^2$	20.44	$3.44\text{x}10^3$	17.12	$1.42\text{x}10^3$	23.25	$4.63\text{x}10^3$
Multi-head CNN+LSTM [45]	2020	Feature selection, RUL target function	12.19	$2.59\text{x}10^2$	19.93	$4.35\text{x}10^3$	12.85	$3.43\text{x}10^2$	22.89	$4.34\text{x}10^3$
CNN+LSTM+ BiLSTM [44]	2020	Correlation analysis, Min-max scaling, RUL target function	10.41	—	—	—	—	—	—	—
AGCNN [65]	2020	Feature selection, Data normalization, RUL target function	12.42	$2.25\text{x}10^2$	19.43	$1.49\text{x}10^3$	13.39	$2.27\text{x}10^2$	21.50	$3.39\text{x}10^3$
LSTM+ FCLCNN [66]	2021	Feature selection, Data normalization, RUL target function	11.17	$2.04\text{x}10^2$	—	—	9.99	$2.34\text{x}10^2$	—	—
Hybrid model [67]	2021	Feature selection, Data normalization, Piece-wise RUL	15.68	—	22.26	—	16.89	—	22.32	—
BLS + TCN [68]	2022	Feature selection, Data normalization, Piece-wise RUL	12.08	$2.43\text{x}10^2$	16.87	$1.60\text{x}10^3$	11.43	$2.44\text{x}10^2$	18.12	$2.09\text{x}10^3$
Bi-LSTM based Attention method [69]	2022	RUL target function	13.78	$2.55\text{x}10^2$	**15.94**	$\mathbf{1.28\text{x}10^3}$	14.36	$4.38\text{x}10^2$	**16.96**	$\mathbf{1.65\text{x}10^3}$
Proposed (without automatic piece-wise linear RUL function)	2022	Correlation analysis, Median filter, Data normalization,	13.5	$2.38\text{x}10^2$	23.37	$2.6\text{x}10^3$	13.54	$4.11\text{x}10^2$	23.36	$3.97\text{x}10^3$
Proposed	2022	Correlation analysis, Median filter, Data normalization, Automatic piece-wise linear RUL function	**7.78**	$\mathbf{1.00\text{x}10^2}$	17.64	$1.44\text{x}10^3$	**8.03**	$\mathbf{1.04\text{x}10^2}$	17.63	$2.39\text{x}10^3$

lowest RMSE and score because of one fault mode. Finally, we have compared our proposed LSTM model with other method published in the last few years on C-MAPSS sensor degradation dataset. The validation of model is investigated by RMSE and score function values which is presented in table 9. Our proposed model achieves state of the art performance on FD001 and FD003 sub-dataset with minimum RMSE, score values and it shows second best result on FD002 and FD004 sub-dataset. The minimum values of FD001 and FD003 are lower than FD002 and FD004 because of the irregular behavior of sensor degradation data. The accuracy significantly depends on the improved automatic piece wise linear degradation function which decides how early you need to provide the maintenance operation by determining the starting point of the degradation.

The robustness of our proposed model is showed by predicting the full life cycle time of few randomly selected test engines in Figures 7-10. Fig. 7 shows the actual and predicted degradation results of RUL of four different types of engines from a total set of 100 engines for validating the performance of model for FD001. In a similar manner, Fig. 8-10 shows the actual and predicted results of full cycle for other three sub-units. From these prediction graphs, we analyze that the predicted and actual degradation of test engine in FD001 and

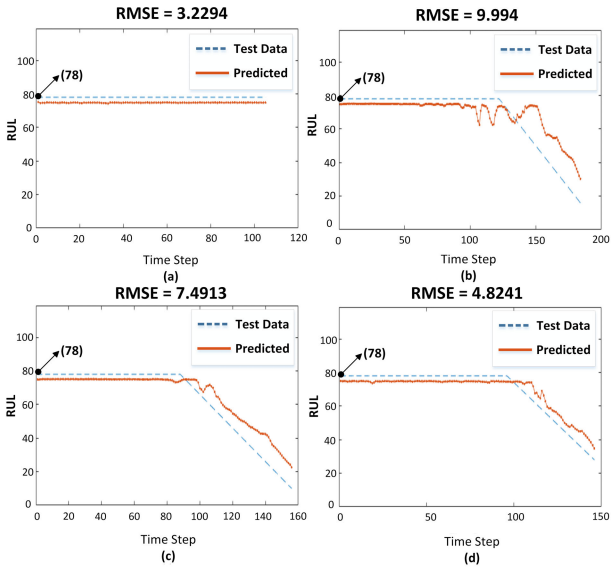

FIGURE 7. RUL Prediction of 4 engines in FD001 sub-dataset (a) engine # 6 (b) engine # 20 (c) engine # 42 (d) engine # 90.

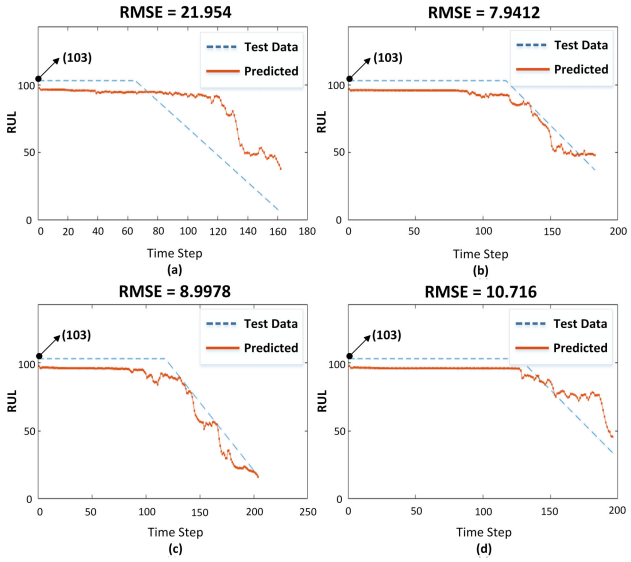

FIGURE 8. RUL Prediction of 4 engines in FD002 sub-dataset (a) engine # 1 (b) engine # 40 (c) engine # 84 (d) engine # 150.

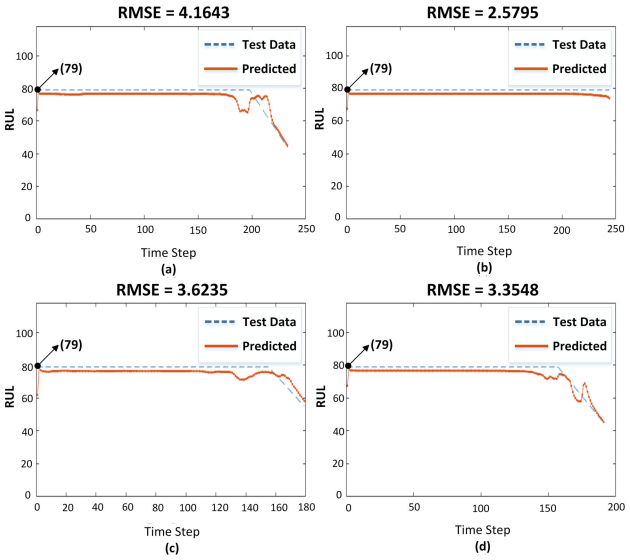

FIGURE 9. RUL Prediction of 4 engines in FD003 sub-dataset (a) engine # 1 (b) engine # 18 (c) engine # 48 (d) engine # 75.

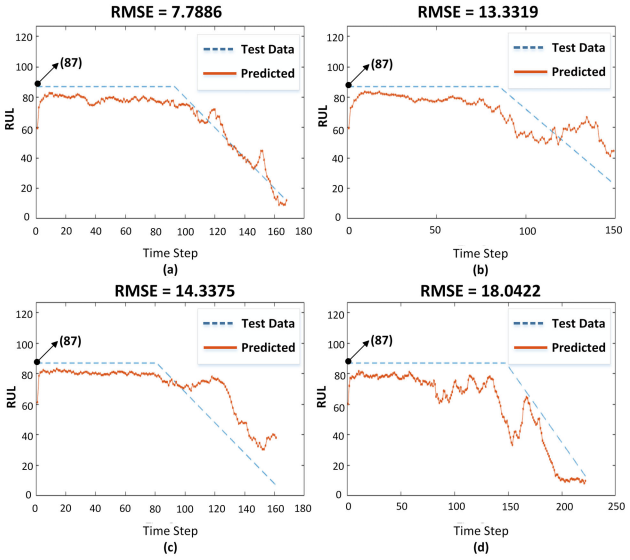

FIGURE 10. RUL Prediction of 4 engines in FD004 sub-dataset (a) engine # 15 (b) engine # 73 (c) engine # 155 (d) engine # 202.

FD003 sub-dataset are very efficient and accurate as shown in Fig. 7 and Fig. 9 but for FD002 and FD004, we have seen that there was some irregular and abrupt changes in the predicted results while tracking the actual target RUL of test engine due to the complex nature of theses sub-datasets, this is shown in Fig. 8 and Fig. 10. In case of prediction on actual RUL having a plot similar to a straight line as shown in Fig. 9(b), our model performs quite good on tracking the degradation trends obtained by applying piecewise linear function on a target labels.

To conclude, in what follows we reason why the proposed prediction framework is able to yield better results than the existing equivalent models. The proposed framework consists of multiple stages from correlation analysis to LSTM network. The significant stages in this framework that greatly enhance the overall accuracy of model are correlation function to filter out irrelevant sensor variables as given in Fig. 2, and the estimation of initial RUL value with piecewise linear function as given in algorithm 1. This function gives the starting point of degradation for sensor data, and hence, we have used those values for training

our deep LSTM network. Moreover, the hyper-parameters selection for the neural network plays a significant role on the effectiveness of model. Therefore, in our research work, the hyper-parameters have been picked through an iterative grid search based approach as shown in Table-6. Thus, these steps when combined in a single prediction pipeline achieved optimum results compared to the existing methods on all the four sub-units, and its tracking performance on the complete life cycle time prediction will be beneficial in preventing the real life failures of machines.

VI. CONCLUSION & FUTURE DIRECTION

This work presented a LSTM based model for predicting the RUL of turbofan engines. The proposed work shows that LSTM model when combined with effective pre-processing steps results in highly accurate RUL predictions. In addition to widely used pre-processing steps like feature selection, filtering and normalization, this work proposed an improved piecewise linear degradation model which estimated the initial RUL valuewhich is the starting point of the degradation. This value had a great impact on the RUL prediction accuracy. LSTM network of the proposed model which consisted of a combination of multiple LSTM layers, dropout layers, fully connected layers and a regression layer, was tested on multiple hyper parameter combinations to achieve the best result. The proposed model was tested on the C-MAPSS turbofan engine simulation dataset. the prediction accuracy depends upon various number of parameters which include parameters for initial RUL calculations and hyper parameters for LSTM model. It was observed that due to different dynamics of each sub dataset, it is better to use a separate prediction model for each sub-units instead of a single model for the whole dataset. RUL predictions of the proposed model were compared with recent benchmark publications and showed higher prediction accuracy. The threshold value and the window size are two important parameters in the degradation model which had great impact on the model accuracy. In future, this approach can be considered for online problems and other machines.

In future, we will further integrate novel deep learning algorithms into our prediction pipeline for processing the complex and real-world sensor data as opposed to traditional techniques. In the field of prognostics, normally we don't get large amount of real-world data for industrial machines specially in-case of faulty conditions. Therfore, we need to develop such novel methods that can be able to train on small dataset, but, it predict the output effectively on large scale practical data. Basically, we requires such types of models, which have strong generalization capability with respect to output. Morever, for further improvements, we have to evaluate the performance of models on other publicly available dataset.

REFERENCES

[1] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Rel. Eng. Syst. Saf.*, vol. 183, pp. 240–251, Mar. 2019.

[2] O. Merkt, "On the use of predictive models for improving the quality of industrial maintenance: An analytical literature review of maintenance strategies," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2019, pp. 693–704.

[3] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, p. 8211, Oct. 2020.

[4] B. Lu, D. Durocher, and P. Stemper, "Predictive maintenance techniques," *IEEE Ind. Appl. Mag.*, vol. 15, no. 6, pp. 52–60, Nov. 2009.

[5] J. J. M. Jimenez, S. Schwartz, R. Vingerhoeds, B. Grabot, and M. Salaün, "Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics," *J. Manuf. Syst.*, vol. 56, pp. 539–557, Jul. 2020.

[6] S. Behera, A. Choubey, C. S. Kanani, Y. S. Patel, R. Misra, and A. Sillitti, "Ensemble trees learning based improved predictive maintenance using IIoT for turbofan engines," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 842–850.

[7] K. T. P. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," *Rel. Eng. Syst. Saf.*, vol. 188, pp. 251–262, Aug. 2019.

[8] C. Zheng, W. Liu, B. Chen, D. Gao, Y. Cheng, Y. Yang, X. Zhang, S. Li, Z. Huang, and J. Peng, "A data-driven approach for remaining useful life prediction of aircraft engines," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 184–189.

[9] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar, "Prediction of remaining useful lifetime (RUL) of turbofan engine using machine learning," in *Proc. IEEE Int. Conf. Circuits Syst. (ICCS)*, Dec. 2017, pp. 306–311.

[10] N. Bolander, H. Qiu, N. Eklund, E. Hindle, and T. Rosenfeld, "Physics-based remaining useful life prediction for aircraft engine bearing prognosis," in *Proc. Annu. Conf. PHM Soc.*, vol. 1, 2009, pp. 1–10.

[11] C. Xiongzi, Y. Jinsong, T. Diyin, and W. Yingxun, "Remaining useful life prognostic estimation for aircraft subsystems or components: A review," in *Proc. IEEE 10th Int. Conf. Electron. Meas. Instrum.*, Aug. 2011, pp. 94–98.

[12] P. Adhikari, H. Rao, and M. Buderath, "Machine learning based data driven diagnostics & prognostics framework for aircraft predictive maintenance," in *Proc. 10th Int. Symp. NDT Aerosp.* Dresden, Germany, 2018, pp. 1–15.

[13] L. Liu, Q. Guo, D. Liu, and Y. Peng, "Data-driven remaining useful life prediction considering sensor anomaly detection and data recovery," *IEEE Access*, vol. 7, pp. 58336–58345, 2019.

[14] M. Aykol, C. B. Gopal, A. Anapolsky, P. K. Herring, B. van Vlijmen, M. D. Berliner, M. Z. Bazant, R. D. Braatz, W. C. Chueh, and B. D. Storey, "Perspective—Combining physics and machine learning to predict battery lifetime," *J. Electrochem. Soc.*, vol. 168, no. 3, Mar. 2021, Art. no. 030525.

[15] H. Oh, M. H. Azarian, M. Pecht, C. H. White, R. C. Sohaney, and E. Rhem, "Physics-of-failure approach for fan PHM in electronics applications," in *Proc. Prognostics Syst. Health Manage. Conf.*, Jan. 2010, pp. 1–6.

[16] C. S. Byington, M. Watson, D. Edwards, and P. Stoelting, "A model-based approach to prognostics and health management for flight control actuators," in *Proc. IEEE Aerosp. Conf.*, vol. 6, Mar. 2004, pp. 3551–3562.

[17] M. A. Chao, C. Kulkarni, K. Goebel, and O. Fink, "Fusing physics-based and deep learning models for prognostics," 2020, *arXiv:2003.00732*.

[18] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2213–2227, Sep. 2019.

[19] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–9.

[20] Y. Liao, L. Zhang, and C. Liu, "Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2018, pp. 1–8.

[21] S. Zhao, Y. Zhang, S. Wang, B. Zhou, and C. Cheng, "A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method," *Measurement*, vol. 146, pp. 279–288, Nov. 2019.

[22] S. Agarwal, V. Saxena, V. Singal, and S. Aggarwal, "LSTM based music generation with dataset preprocessing and reconstruction techniques," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 455–462.

[23] G. Lan, Q. Li, and N. Cheng, "Remaining useful life estimation of turbofan engine using LSTM neural networks," in *Proc. IEEE CSAA Guid., Navigat. Control Conf. (CGNCC)*, Aug. 2018, pp. 1–5.

[24] K. T. Chui, R. W. Liu, M. Zhao, and P. O. D. Pablos, "Predicting students' performance with school and family tutoring using generative adversarial network-based deep support vector machine," *IEEE Access*, vol. 8, pp. 86745–86752, 2020.
[25] A. P. Hermawan, D.-S. Kim, and J.-M. Lee, "Predictive maintenance of aircraft engine using deep learning technique," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 1296–1298.
[26] D. Simon and D. L. Simon, "Aircraft turbofan engine health estimation using constrained Kalman filtering," *J. Eng. Gas Turbines Power*, vol. 127, no. 2, pp. 323–328, Apr. 2005.
[27] T. Kobayashi and D. L. Simon, "Hybrid Kalman filter approach for aircraft engine in-flight diagnostics: Sensor fault detection case," in *Turbo Expo: Power for Land, Sea, Air*, vol. 42371. Barcelona, Spain: ASME, 2006, pp. 745–755.
[28] S. Chen, M. Wang, D. Huang, P. Wen, S. Wang, and S. Zhao, "Remaining useful life prediction for complex systems with multiple indicators based on particle filter and parameter correlation," *IEEE Access*, vol. 8, pp. 215145–215156, 2020.
[29] J. B. Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, and F. Fnaiech, "Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network," *Mech. Syst. Signal Process.*, vols. 56–57, pp. 150–172, May 2015.
[30] K. Salahshoor, M. Mosallaei, and M. Bayat, "Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended Kalman filter algorithm," *Measurement*, vol. 41, no. 10, pp. 1059–1076, 2008.
[31] C. Ordóñez, F. S. Lasheras, J. Roca-Pardiñas, and F. J. de Cos Juez, "A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines," *J. Comput. Appl. Math.*, vol. 346, pp. 184–191, Jan. 2019.
[32] N. Gebraeel, A. Elwany, and J. Pan, "Residual life predictions in the absence of prior degradation knowledge," *IEEE Trans. Rel.*, vol. 58, no. 1, pp. 106–117, Mar. 2009.
[33] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2276–2285, Mar. 2017.
[34] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017.
[35] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *J. Manuf. Syst.*, vol. 48, pp. 78–86, Jul. 2018.
[36] A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking LSTM for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, Jan. 2019.
[37] M. Saeidi, M. Soufian, A. Elkurdi, and S. Nefti-Meziani, "A jet engine prognostic and diagnostic system based on Bayesian classifier," in *Proc. 12th Int. Conf. Develop. eSyst. Eng. (DeSE)*, Oct. 2019, pp. 975–977.
[38] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2017, pp. 88–95.
[39] R. Wu and J. Ma, "An improved LSTM neural network with uncertainty to predict remaining useful life," in *Proc. CAA Symp. Fault Detection, Supervision Saf. Tech. Processes (SAFEPROCESS)*, Jul. 2019, pp. 274–279.
[40] S. Wang, X. Zhang, D. Gao, B. Chen, Y. Cheng, Y. Yang, W. Yu, Z. Huang, and J. Peng, "A remaining useful life prediction model based on hybrid long-short sequences for engines," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1757–1762.
[41] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2016, pp. 214–228.
[42] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.
[43] L. Jayasinghe, T. Samarasinghe, C. Yuenv, J. C. N. Low, and S. S. Ge, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 915–920.
[44] C. W. Hong, C. Lee, K. Lee, M.-S. Ko, D. E. Kim, and K. Hur, "Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction," *Sensors*, vol. 20, no. 22, p. 6626, Nov. 2020.
[45] H. Mo, F. Lucca, J. Malacarne, and G. Iacca, "Multi-head CNN-LSTM with prediction error analysis for remaining useful life prediction," in *Proc. 27th Conf. Open Innov. Assoc. (FRUCT)*, Sep. 2020, pp. 164–171.
[46] Z. Zhao, B. Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Reliab. Eng. Syst. Safe.*, vol. 164, pp. 74–83, Aug. 2017.
[47] A. Zhang, H. Wang, S. Li, Y. Cui, Z. Liu, G. Yang, and J. Hu, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Appl. Sci.*, vol. 8, no. 12, p. 2416, Nov. 2018.
[48] K. T. Chui, B. B. Gupta, and P. Vasant, "A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine," *Electronics*, vol. 10, no. 3, p. 285, Jan. 2021.
[49] Y. T. Wu, M. Yuan, S. Dong, L. Li, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.
[50] K. Liu, N. Z. Gebraeel, and J. Shi, "A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 652–664, Jul. 2013.
[51] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2017, pp. 87–90.
[52] C. W. Hong, K. Lee, M.-S. Ko, J.-K. Kim, K. Oh, and K. Hur, "Multivariate time series forecasting for remaining useful life of turbofan engine using deep-stacked neural network and correlation analysis," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 63–70.
[53] Y. Huang, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing* (Springer Topics in Signal Processing), vol. 2. Cham, Switzerland: Springer, 2009.
[54] M. Friendly, "Corrgrams: Exploratory displays for correlation matrices," *Amer. Statistician*, vol. 56, no. 4, pp. 316–324, Nov. 2002.
[55] M. Ohki, M. E. Zervakis, and A. N. Venetsanopoulos, "3-D digital filters," in *Multidimensional Systems: Signal Processing and Modeling Techniques* (Control and Dynamic Systems), vol. 69, C. Leondes, Ed. New York, NY, USA: Academic, 199, pp. 49–88.
[56] H. M. Elattar, H. K. Elminir, and A. M. Riad, "Towards online data-driven prognostics system," *Complex Intell. Syst.*, vol. 4, no. 4, pp. 271–282, Dec. 2018.
[57] C.-G. Huang, H.-Z. Huang, W. Peng, and T. Huang, "Improved trajectory similarity-based approach for turbofan engine prognostics," *J. Mech. Sci. Technol.*, vol. 33, no. 10, pp. 4877–4890, Oct. 2019.
[58] P. Baldi and P. Sadowski, "Understanding dropout," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 2814–2822.
[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[60] S. Lee, H. Yu, H. Yang, I. Song, J. Choi, J. Yang, G. Lim, K.-S. Kim, B. Choi, and J. Kwon, "A study on deep learning application of vibration data and visualization of defects for predictive maintenance of gravity acceleration equipment," *Appl. Sci.*, vol. 11, no. 4, p. 1564, 2021.
[61] E. Ramasso and A. Saxena, "Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2014, pp. 1–12.
[62] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Chongqing)*, Oct. 2018, pp. 1037–1042.
[63] J. Li, X. Li, and D. He, "A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction," *IEEE Access*, vol. 7, pp. 75464–75475, 2019.
[64] Z. Kong, Y. Cui, Z. Xia, and H. Lv, "Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics," *Appl. Sci.*, vol. 9, no. 19, p. 4156, Oct. 2019.
[65] H. Liu, Z. Liu, W. Jia, and X. Lin, "Remaining useful life prediction using a novel feature-attention-based end-to-end approach," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1197–1207, Feb. 2021.
[66] C. Peng, Y. Chen, Q. Chen, Z. Tang, L. Li, and W. Gui, "A remaining useful life prognosis of turbofan engine using temporal and spatial feature fusion," *Sensors*, vol. 21, no. 2, p. 418, Jan. 2021.
[67] U. Amin and K. D. Kumar, "Remaining useful life prediction of aircraft engines using hybrid model based on artificial intelligence techniques," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2021, pp. 1–10.

[68] K. Yu, D. Wang, and H. Li, "A prediction model for remaining useful life of turbofan engines by fusing broad learning system and temporal convolutional network," in *Proc. 8th Int. Conf. Inf., Cybern., Comput. Social Syst. (ICCSS)*, Dec. 2021, pp. 137–142.
[69] Y. Liu, X. Zhang, W. Guo, H. Bian, Y. He, and Z. Liu, "Prediction of remaining useful life of turbofan engine based on optimized model," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, 2021, pp. 1473–1477.

OWAIS ASIF was born in Wah Cantt, Pakistan, in 1994. He received the B.S. degree in electrical engineering from the Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan, in 2017, and the M.S. degree in electrical engineering from COMSATS University Islamabad, Wah Campus, in 2021. His research interests include prognostics and system health management, development of data-driven-based algorithms for contemporary research problems especially for multivariate time series as well as image data, and RUL estimation for improving the efficiency of critical industrial machines.

SAJJAD ALI HAIDER received the B.S. degree in from COMSATS University Islamabad (CUI), Islamabad, in 2005, the M.S. degree in embedded systems and control engineering from Leicester University, U.K., in 2007, and the Ph.D. degree from Chongqing University, China, in 2014. He is currently working as an Assistant Professor at the Department of Electrical and Computer Engineering, CUI, Wah Campus, Pakistan. His research interests include embedded systems, control systems, and machine learning.

SYED RAMEEZ NAQVI received the M.Sc. degree in electronic engineering from The University of Sheffield, U.K., in 2007, and the Ph.D. degree in computer engineering from the Vienna University of Technology, Austria, in 2013. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, COMSATS University Islamabad, Wah Campus. Since his Ph.D., he has published more than 35 research articles in international journals and symposiums of high reputation, and carried several funded research projects with various research and development organizations. His experience as an academic has led him to believe that there is a considerable gap between academia and industry in Pakistan, especially in the application of contemporary methods, such as data analytics, in real-life problems. To address this disparity, he has co-founded Convex Solutions Pvt. Ltd., and established a consortium of peers in academics, mainly responsible for developing curriculum for training on artificial intelligence and business analytics for young professionals and graduate students having diverse academic backgrounds. His research interests include asynchronous logic, computer architecture, hardware optimization, and artificial intelligence.

JOHN F. W. ZAKI received the master's degree from Cambridge University, the M.B.A. degree from Glasgow University, and the Ph.D. degree from Mansoura University. He worked in both technical and managerial positions in various multinational companies. He is experienced in establishing tech start-ups, project management, and business processes. He is currently the Vice Director of the Communication and Information Technology Centre and the Director of e-learning at Mansoura University. He is also an Assistant Professor at the Department of Computer and Systems, Mansoura University. His research interests include artificial intelligence, software engineering, control systems, and e-learning technology.

KYUNG-SUP KWAK (Life Senior Member, IEEE) received the Ph.D. degree from the University of California. He was with Hughes Network Systems and the IBM Network Analysis Center, USA. He was a Professor with Inha University, South Korea. He was also the Dean of the Graduate School of Information Technology and Telecommunications and the Director of the UWB Wireless Communications Research Center. In 2008, he was an Inha Fellow Professor (IFP). He is currently an Inha Hanlim Fellow Professor and a Professor with the School of Information and Communication Engineering, Inha University. His research interests include UWB radio systems, wireless body area networks and u-health networks, and nano and molecular communications. He received the Official Commendations for achievements of UWB radio technology research and development from the Korean President in 2009. In 2006, he was the President of the Korean Institute of Communication Sciences (KICS) and the Korea Institute of Intelligent Transport Systems (KITS) in 2009.

S. M. RIAZUL ISLAM (Member, IEEE) was with the University of Dhaka, Bangladesh, as an Assistant Professor, and a Lecturer with the Department of Electrical and Electronic Engineering, from 2005 to 2014. In 2014, he worked at the Department of Solution Laboratory, Samsung Research and Development Institute Bangladesh (SRBD), as a Chief Engineer. From 2014 to 2017, he was a Postdoctoral Fellow at the Wireless Communications Research Center, Inha University, South Korea. From 2017 to 2022, he worked as an Assistant Professor at the Department of Computer Science and Engineering, Sejong University, South Korea. He is currently a Senior Lecturer in computer science at the University of Huddersfield, U.K. His research interests include the Internet of Things (IoT), applied artificial intelligence (AI), and 5G/6G.