J Inno Sci Eng,2023, 7(1):88-105
http://doi.org/10.38088/jise.1206920

Research Article

Remaining Useful Life Estimation via Cascaded Self-Attention and ResNet Models

Adem Avci [1*], Nurettin Acir [2]

[1] *Bursa Technical University, Department of Electrical and Electronics Engineering, Bursa, Turkey*
[2] *National Defence University, Turkish Air Force Academy, Department of Electronics Engineering, Istanbul, Turkey*

Cite this paper as:
Avci, A. and Acir, N. (2023). *Remaining Useful Life Estimation via Cascaded Self-Attention and ResNet Models* 7(1):88-105.

*Corresponding author: Adem AVCI
E-mail: adem.avci@btu.edu.tr

Received Date: 18/11/2022
Accepted Date: 23/02/2023

Abstract

Prognostics and Health Management occupy an important place in modern industrial maintenance to increase the reliability of systems. Determining the Remaining Useful Life of the system or its parts is vital accurately to maintaining critical parts of the system and successful prognostics and health management. This study proposes a data-based Remaining Useful Life prediction method with a network consisting of a cascade-connected Self-Attention and Residual Network layer. The network is fed by multiple sensor signals to monitor the aero-engines. The proposed model contains four main parts: The Gaussian Noise Layer, the Self-Attention Layer, the Residual Network Layer, and the layer to estimate Remaining Useful Life. The model is created to be more robust and susceptible to noise using the Gaussian Noise Layer. The Self-Attention Layer focuses on crucial points through time. The Residual Network Layer uses feature extraction and makes the model more profound help of the skip connection. Finally, the Remaining Useful Life estimation is made using highly correlated features obtained from the fully connected layer and the output layer. In addition, a new loss function has been offered, similar to the evaluation metrics in the literature. With the proposed model and loss function, 11.017 and 12.629 in root mean square error, 157.19 and 218.6 in score function are obtained in the FD001 and FD003, respectively. The superior performance of these results on the C-MAPSS dataset is demonstrated by comparing the other state-of-the-art methods in the literature.

Keywords: Remaining useful life, Self-attention, Prognostics and health management, Deep learning, Residual Layer

1. Introduction

As the complexity of systems increases in modern industry, maintenance and prognostic become more crucial. With the developments in sensor technologies, computing systems, and learning algorithms, Prognostic and Health Management (PHM) put to use in various areas such as aero-engines [1], electric motors [2], battery systems [3], and nuclear plants [4]. PHM aims to monitor systems and their parts with the help of sensor data and estimate the degradation of this systems. Also, PHM diagnoses abnormal activity, detects possible failures, estimates machines' state of health, and predicts the Remaining Useful Life (RUL) of the system or its equipment. Hence, in industry, maintenance schedules can be planned effectively, unnecessary parts replacement during maintenance can be avoided, and maintenance costs are reduced. PHM can prevent catastrophic failures by estimating the current and future states of systems and, with this, improve the reliability and performance of systems. RUL can show the remaining life with time cycles or hours according to the area studied. RUL is defined in the literature as the time from the current time to time occurred failure [5].

The RUL estimation has generally been categorized in recent PHM studies with three different approaches [6]. These are model-based approach, data-driven approach, and hybrid approach. Life estimation is being studied by creating mathematical models with prior knowledge of the system examined in model-based PHM approaches [7]. The studies in the literature identify degradation in systems and developed the model-based approach such as Paris's law [8] and exponential models [9]. However, as the system's complexity increases, the performance of these mathematical models in life estimation problems begins to decline. In addition, deviations may occur in the models created as the operating conditions of systems change.

On the other hand, data-driven approaches include the detection of errors with the help of sensors on the systems and estimating RUL. Big data collected from sensors provides the development of degradation models. Machine learning and deep learning structures, which have developed in recent decades, can show more successfully the status of a machine's health together with big data. Hybrid models aim to achieve better results by eliminating the deficiencies of model-based and data-driven approaches. However, unlike the data-based approach, particular expertise is required to develop and integrate a physical system into the degradation model. The following subsection provides a brief overview of research studies using the data-driven approach to estimate RUL.

1.1. Literature Review

In the literature, better predictions are made using machine learning and deep learning algorithms in data-driven approaches. Each data point is committed independently in machine learning approaches, and estimations are performed. In this context, studies were carried out with Support Vector Regression (SVR) [10], Relevance Vector Regression (RVR) [10], Random Forest (RF) [11], and similar algorithms. In addition, the data points were evaluated independently in the Multilayer Perceptron (MLP) algorithm, and RUL estimation was made [10]. With the performance of deep learning algorithms in the last decades, it has been frequently used in RUL estimation. Among these algorithms, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and their derivatives produced successful results in RUL estimation. Babu et al. proposed a two-dimensional deep CNN to estimate aero-engines RUL [10]. Firstly, sensor signals were normalized, and the time windowing method was employed to train deep CNN (DCNN). Also, their models included two convolutional and average pooling layers,

then used a Fully Connected (FC) layer. Automatic feature extraction has been provided with the convolutional layer, and RUL estimation has been made. In another study, Li et al. offered a novel DCNN [12]. They set the C-MAPSS dataset as two-dimensional inputs with a time windowing pre-processing technique. They employed five convolutional layers in their model, each of which is made up of a 1-D filter. The dropout technique was used to avoid overfitting problems in their model.

Since the RUL dataset is a time sequence, it has a suitable structure for dynamic networks. Zheng et al. suggested a dynamic neural network consisting of Long Short-Term Memory (LSTM) layers [13]. Their model included two LSTM layers, two FC layers, and a neuron as the output layer. Then, they tested their networks with a different number of LSTM cells in LSTM layers and a different number of neurons in FC layers. LSTM layers were able to extract complex features in the time domain and thus obtained better Root Mean Square Error (RMSE) values compared to RNN. Wu et al. wielded a vanilla LSTM network in their work [14]. Also, new features were extracted from operating conditions values by the dynamic difference technique. These features were used with chosen sensor signals that change with time. In their work, Yu et al. accomplished RUL prediction in two stages [15]. First, the dataset is processed using selecting sensor signals, the normalization process, and the time windowing technique, which has different window lengths. Then, the machine's Health Index (HI) is estimated with a Bi-directional LSTM(BLSTM)-based autoencoder model and linear regression. Finally, RUL values are mapped with estimated HI. In another study, Wang et al. propounded a BLSTM-based network [16]. Their model included two BLSTM layers, 2 FC layers, and the output layer, which has a neuron. Palazuelos et al. proposed a novel capsule neural network that Hinton et al. offered to overcome the shortcomings of CNN [17]. The capsule structure was first used for the C-MAPSS dataset in this study.

Some studies use the feature extraction attributes of LSTM and CNN networks together. Al-Dulaimi et al. wielded LSTM and CNN structures as parallel branches in their study [18]. Thus, capturing the CNN structure's spatial and the LSTM structure's temporal features are considered together. Al-Dulaimi et al. added gaussian noise layers to parallel network structure in their study [19]. The LSTM path in the previous work was changed with the BLSTM path that added noise layers. Thus, they obtained better results in RUL estimation. J. Li et al. have adopted the parallel network structure in their study [20]. The parallel branches consisted of Convolution layers and LSTM layers. Summing the outputs of these paths is connected to the LSTM layer, FC layer, and output layer, respectively. Song et al. proposed a novel neural network in series connection [21]. Their model had two BLSTM layers and two FC layers stacked after the autoencoder layer. The feature extraction was made by autoencoder, and long-range dependencies of temporal features were utilized with BLSTM. The dropout technique was used as regularization. Ragab et al. offered autoencoder-based LSTM [22]. In the decoder part of their model was placed attention mechanism. A path to reconstruct the input data and another to estimate RUL was offered. In model training, summing reconstruction loss and RUL loss were used. In their model, Liu et al. wielded a feature extraction layer consisting of a channel attention mechanism and a transformer with temporal attention [23]. Tan et al. proposed a novel network consisting of an attention layer [24]. They used the attention layer after four Convolution layers and connected the FC layer and output layer. Two different lengths of the time window are used in their study. In [25], on the other hand, LSTM and self-attention structure were used together. With the self-attention structure, it was ensured to focus on the critical points in the data, and RUL estimation was made by transferring the information to the following layers with the LSTM layers.

1.2. Contributions

It is made to estimate RUL on the time series like aero-engines data in this study. The time series data were considered with different network structures in the literature. RNN, LSTM, and Gated Recurrent Unit (GRU) structures were used in these studies. Against previous works, these networks fetched good results due to dynamic structure. However, there was no further improvement in results due to limited and noisy data. The Self-Attention Layer has been added to the models, resulting in better results. In addition, the formation of more robust structures in the models was provided with different techniques. Although Residual Network (ResNet) layers have been used in different studies, it has been observed that it causes overfitting in the datasets in the experiments. In the previously mentioned literature, they have obtained state-of-the-art results by using each in different models. However, robust, deep structures focused on relevant points in the time domain have not been used together. Also, this study determined this gap in the literature, and network models were developed on better estimations.

The main contribution of this study to the literature is as follows:

1) With the Self-Attention layer, it has been ensured to focus on the crucial points in the time series. Thus, the proposed model detected the vital points for a good estimation of time series data in each sensor signal.
2) By adding the noise layer, it is planned that the proposed model will operate more robustly and make more accurate estimations. The Gaussian Noise layer is just used in the training process. Moreover, the model has become more robust in the noise layer.
3) Using the ResNet structure ensures that the backpropagation can reach the first layer. Thus, the network structure can be made deeper.
4) Also, the Convolution layer is used with the stride technique instead of the pooling layer to avoid losing information. Thus, while feature extraction was performed, dimension reduction was also made together.

Self-Attention and ResNet structure were wielded as cascaded, and better results were obtained with estimation performance compared to other studies. However, the proposed model has limitations. One of them, the model input, is created with a constant input shape. Also, the dataset used in the study has sub-datasets, and fixed input data is not created with the same pre-processing steps. For this reason, sub-datasets with the same input form are used. In addition, model structure, hyperparameters, and weighting coefficients should be changed to obtain better results in each sub-dataset.

The rest of this study is organized as follows: In the second part, the offered Cascaded Self-Attention ResNet Network (CSARN) model and the methods used in the model are explained. In addition, the C-MAPSS dataset on which the proposed model is tested is detailed, and the pre-processing steps for RUL estimation are explained. In the third part, the experimental work carried out is explained. The results obtained with the experimental study are reported and graphed. It is also compared with other studies in the literature. The last part summarizes all the work, and information about future work is given.

2. Materials and Methods

This section presents the proposed model for RUL estimation and the studied dataset. Within the scope of this

study, a self-attention-based model is created. Since the dataset has a noisy structure, a Gaussian Noise layer has been added to the beginning of the model. Then, it is transferred to the following layers by detecting the most critical points in the time domain with the Self-Attention layer. Both automatic feature extraction is provided with the ResNet layers, and training is improved with the skip connection structure. An FC structure is used in the last layers, and RUL estimation is made with the last layer.

The C-MAPSS dataset produced by NASA Ames Research Lab is introduced. The sensor information that they contain in the dataset is given in detail. The pre-processing steps on the dataset are explained before training the proposed model. The selection of the sensors in the dataset, the normalization process, the time windowing technique details applied to the dataset, and the target data prepared according to the degradation model are explained. Finally, the metrics to evaluate the predictions made on the test dataset with the trained model are presented.

2.1. Proposed Method

2.1.1. Self-Attention Module

CNN shows remarkable performance in automatic feature extraction. Convolution operation, by its nature, provides translational equivariance and translational invariance features in operations [26]. Convolution, introduced by LeCun et al. for the first time in neural networks, achieved great success, especially in image classification and object detection. By scanning with the filters created with fixed lengths on the input data, the features are automatically extracted for all the data. As we move from the first CNN layer to the last CNN layer in deep networks, the complexity of the extracted features increases and gets closer to the desired estimations. The translational equivariance feature of convolution operation provides valuable information in local data, but capturing global dependencies in time series signals is not feasible.

The Attention mechanism, developed and frequently used recently, shows a significant impact where it is used [27]. It was used in natural language processing (NLP) first and showed great success in its field. The Attention mechanism ensures that the models are focused on the critical regions in sequential data. Unlike the attention mechanism, the Self-Attention mechanism transfers the necessary information from a single content to the following layers. In the Self-Attention structure shown in Figure 1, a single content is entered into the block. With the help of this input data Convolution layer, three different contents are created, and the block output is focused on the most critical parts. Within the scope of this study, it was ensured that the Self-Attention mechanism was influential throughout the time domain in the dataset used. The crucial parts in the time domain are transferred to the following layers at the output of the block. By focusing on essential regions, regression performance would be improved with the help of Self-Attention mechanisms.

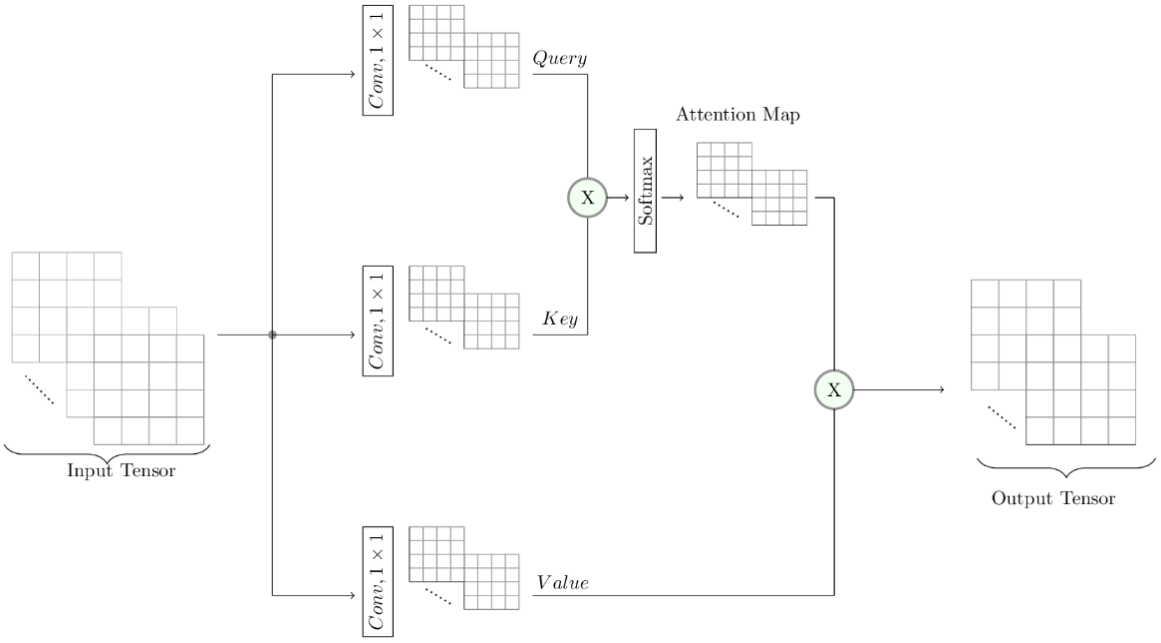

Figure 1. Self-Attention layer in the proposed model

The Self-Attention layer used in this study is shown in Figure 1. Input data is divided into three tensors with Convolution layers: Query, Key, and Value. The resulting Query and Key tensors are multiplied by the Hadamard product and passed through the softmax activation function across the time domain. Thus, it is ensured that it focuses on the most appropriate points in the time domain for our regression problem. Output data is procured by multiplying the same dimensional Value Tensor with the attention map. The equation expressing the Self-Attention layer is shown below [27].

$$A_M = softmax(Query, Key)Value \quad (1)$$

2.1.2. ResNet Module

More complex features are extracted as the network structure becomes deeper in deep learning models. Adding more layers increases the number of model parameters, but the models' performance is generally improved. However, vanishing/exploding gradient problems arise as the network structures get deep. In addition, while the model coefficients are updated with the backpropagation algorithm during the training, updating the first layers with deepening becomes more challenging. Residual Convolution Block has been proposed in the ResNet model to fix the reported update problem [28]. With this structure, it is aimed to spread the loss effect in the output layer to the first layers. The ResNet Block structure used in our study is shown in Figure 2. As shown in the structure, while feature extraction continues from one branch, a skip connection is established from the other branch. Thus, even if the network structure continues to deepen, the gradient effect is aimed at reaching the first layers through the second path.

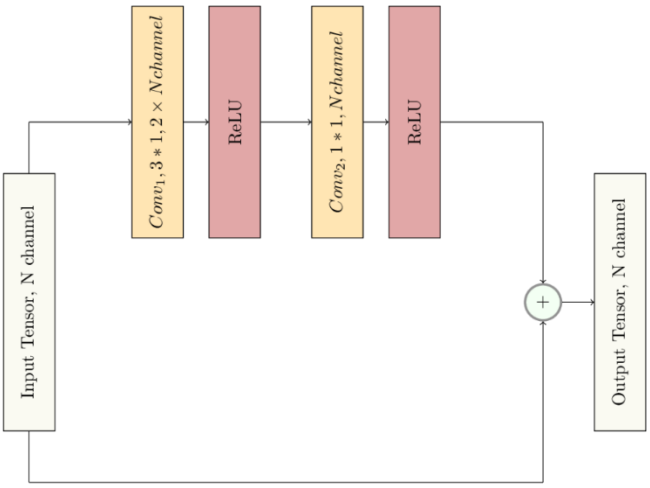

Figure 2. ResNet block in the proposed model

2.1.3. Cascade Self-Attention ResNet Network (CSARN)

Cascaded Self-Attention ResNet Network (CSARN) consists of four different layers connected in series. These are Convolution, Self-Attention, ResNet, and FC layers, respectively. The proposed model structure is shown in Figure 3. First, the Gaussian Noise layer, which has zero mean and 0.01 standard deviation, was added to the proposed model. This layer is only wielded during model training. Then, the Convolution layer with ten filters with a filter size of 11x1 is connected to the model in series. In addition, the number of filter numbers was kept constant for all Convolution layers, except the convolutional layer before the Flatten layer used in the model. After the first Convolution layer, the Self-Attention layer was added. Thus, it is ensured that the relevant points in the input data are focused and transferred to the subsequent layers. The proposed ResNet layer and the Convolution layer were added to the model twice after the Self-Attention layer. In the Convolution layer, size reduction was achieved by taking stride set 2 throughout the time domain.

Moreover, in this and the following Convolution layers, the filters are used with a filter size is 7x1. After the last Convolution layer, the structure is connected to the FC layers with the Flatten layer. There are dropout layers with a dropout rate of 0.3 between the FC1 and FC2 layers, as shown in Figure 3, and between the FC2 and FC3 layers. There have 140 and 90 neurons in the FC 2 and 3 layers, respectively. The FC layers are finally connected to the output layer with a single neuron. 'ReLu' is used as the activation function in all model layers. In addition, the regularization parameter in all layers in the model was set as 0.0002.

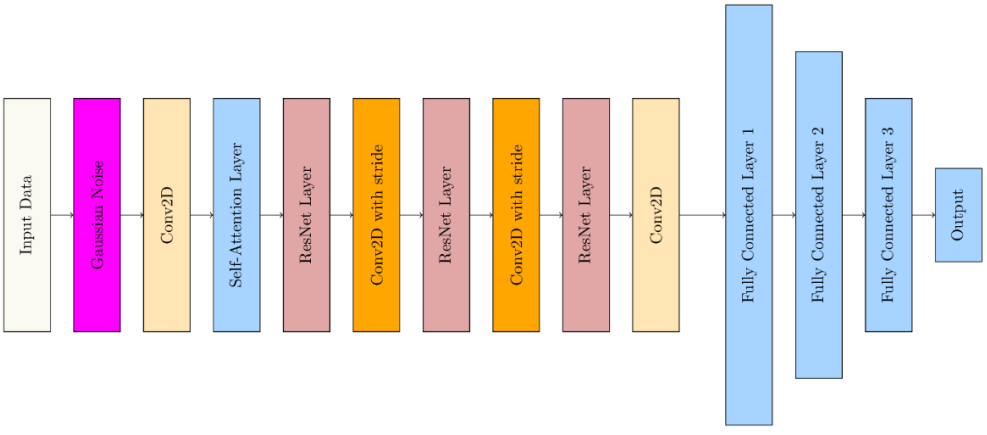

Figure 3. Proposed model for RUL prediction

2.2. Experimental Configuration

2.2.1. Dataset Description

In this paper, the proposed model is evaluated on NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) turbofan engine dataset [29]. This dataset contains synthetic data generated on MATLAB Simulink environment by NASA Ames Research Lab. The aero-engine model while creating the dataset is shown in Figure 4.

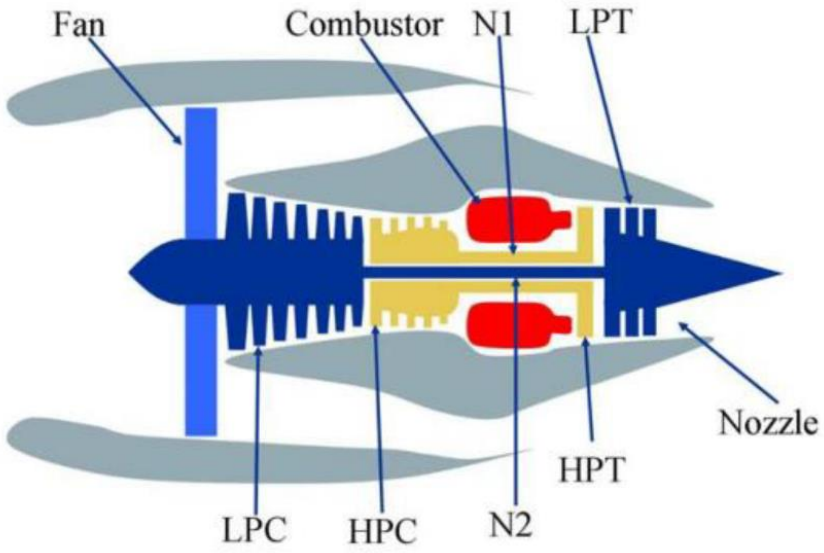

Figure 4. C-MAPSS turbofan engine [29]

The dataset consists of 4 sub-datasets, each with different conditions and failure modes. These sub-datasets were generated at six different operational conditions by operating the engines at altitudes between sea level and 40000 ft, in the speed range 0 to 0.84 Mach, and throttle resolver angle between 20 and 100. Also, RUL scenarios on the datasets were simulated on the deterioration of two of the engine's five rotating elements (Fan, LPC, HPC, HPT, and LPT) [29]. Input parameter values in each motor simulation in the dataset are changed to ensure that they have different RUL values. The engines in the training set were run to time that occurs break down. However, in the test dataset, the engines were run from the start to a specific time. The details of C-MAPSS are listed in Table 1.

Table 1. Details of the C-MAPSS dataset

Dataset	NASA C-MAPSS			
	FD001	FD002	FD003	FD004
Train sets	100	260	100	249
Test sets	100	259	100	248
Operating Conditions	1	6	1	5
Fault Conditions	HPC	HPC	HPC, Fan	HPC, Fan
Training Samples	20631	53759	24720	61249
Min/Max Cycles for Train set	128 / 362	128 / 378	145 / 525	128 / 543
Min/max cycles for Test set	31 / 303	21 / 367	38 / 475	19 / 486

The simulated engines were followed with 58 different sensor data, but a dataset was created with 21 of them. In addition, three sensor data measuring environmental conditions have been added to the dataset. The sensor data and their units are listed in Table 2, respectively.

Table 2. All sensors in the C-MAPSS dataset

Symbol	Description	Units
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°*R*
T50	Total temperature at LPT outlet	°*R*
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/p2)	-
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps / psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass ratio	-
farB	Burner fuel-air ratio	-
htBleed	Bleed Enthalpy	-
Nf_dmd	Demanded fan speed	rpm
PCNFR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm / s
W32	LPT coolant bleed	lbm / s

2.2.2. Normalization

The C-MAPSS dataset comprises 26 columns with engine ID, time cycles, three operational conditions, and 21 sensor values. Each of the 21 sensors that are comprised of the C-MAPSS dataset does not change over time. While some sensor values in the dataset decrease towards the end of the engine's life, some increase. Data collected from seven sensors numbered 1, 5, 6, 10, 16, 18, and 19 remain constant over time. These sensor signals are reported not to provide any helpful information by studies in the literature, but they are used in the scope of this work [30].
The data are normalized before training the model we propose. The z-score normalization process was used to ensure that the trained model converges fast and that the features with higher values than other sensor values do not dominate and adversely affect the model's training. Equation 2 shows the z-score normalization process.

$$x_{norm}^{i,j} = \frac{x^{i,j} - \mu^j}{\sigma^j} \quad \forall i,j \tag{2}$$

The μ^j value in Equation 2 shows the mean of the feature, and the σ^j shows its standard deviation of the feature.

2.2.3. Time Windowing

Data collected from sensors as time series must be preprocessed to train, especially to capture the associative connections in the time domain. The time windowing technique is used to capture these associative connections. Thus, the dataset is workable for dynamic and static networks. The sub-datasets in Table 1 have engines with different data lengths. In these sub-datasets, window sizes were determined according to the engine with the lowest data in order to be used in each engine training and test dataset. The FD001 and FD003 sub-datasets are used because the window sizes could be 30 in this study. As the engines in the FD002 and FD004 sub-datasets can have less than 30 samples, they did not use in this study. This windowing process is shown in Figure 4. In this study, the stride process in time windowing was set one. In the time windowing process shown in Figure 4, the status of the

features from the raw dataset after normalized and shifted is shown.

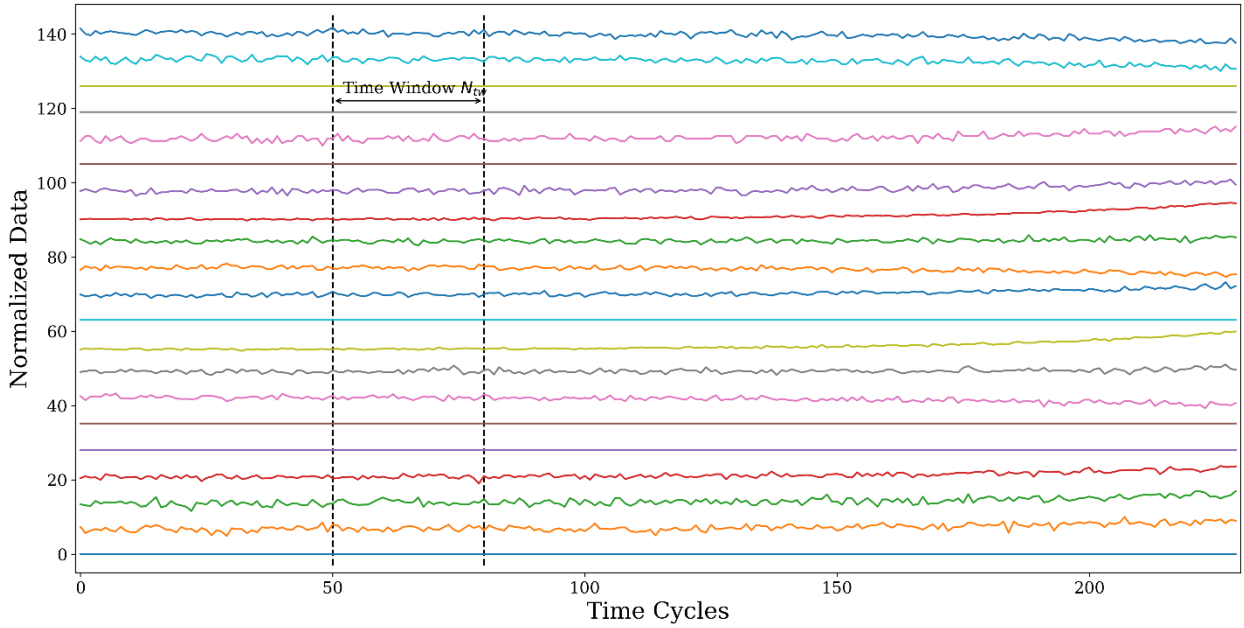

Figure 4. Processing of time windowing with selected features

2.2.4. RUL Labeling

In real-world applications, it is thought that the life of the system studied will end linearly. It is expected that the life of the system or its part will decrease with each operating cycle. However, when the collected data is examined, there is not much change in the initial sensor values of the systems. Sensor values will be changed towards the end of their life compared to the start time. For this reason, Heimes et al. presented a piece-wise degradation model [31]. Linear and piece-wise degradation models are shown in Figure 5 together.

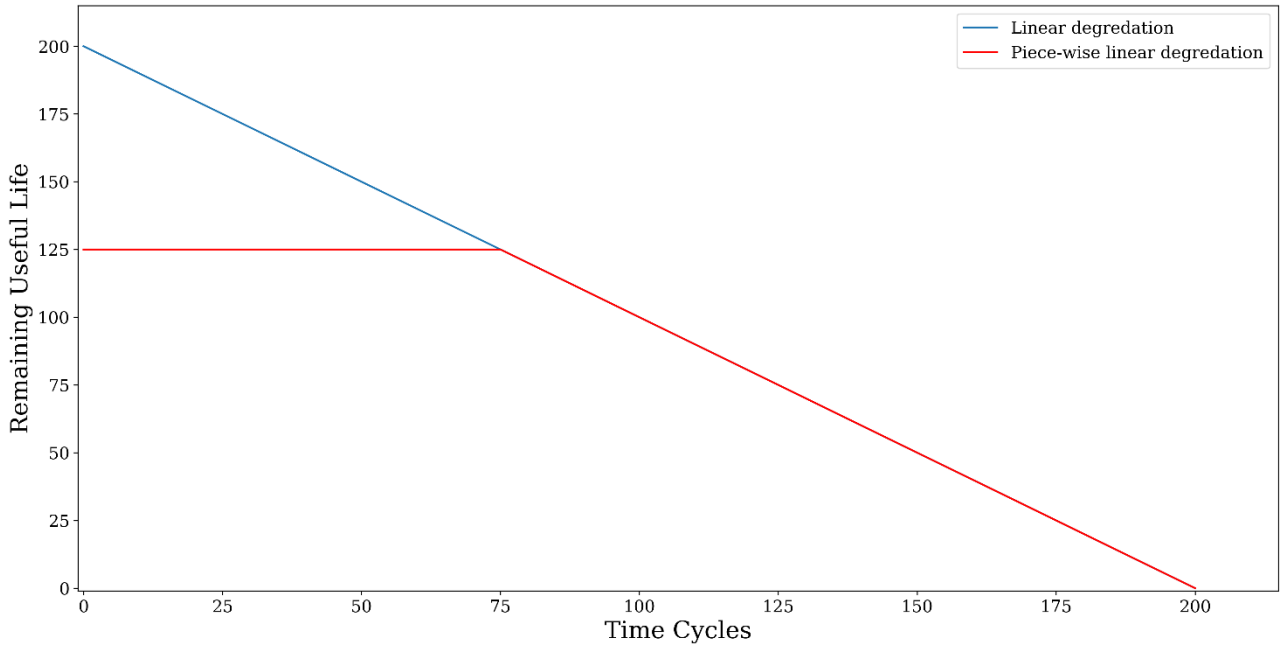

Figure 5. Piece-wise linear degradation function

In the piece-wise degradation model shown in Figure 5, the aero-engines were initially considered healthy. The R_{early} value was determined according to the minimum number of samples in the training datasets. In the C-MAPSS

summary shown in Table 1, the minimum number of samples in the training data appears to be 128. In the literature, it is set between 120-130. Within the scope of this study, the R_{early} value was set as 125.

2.2.5. Evaluation Metrics

All developed predictive models are evaluated with two different functions in the literature. These functions are RMSE and Score functions. Also, Mean Absolute Percentage Error (MAPE) evaluation metric is used in the prediction problems. The RMSE and MAPE functions are shown in equations 4 and 5, respectively. Also, RMSE is widely used in prognostic and health management problems.

$$d_i = RUL_{predicted} - RUL_{actual} \tag{3}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \tag{4}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|RUL_{actual} - RUL_{predicted}\right|}{RUL_{actual}} \tag{5}$$

In equations 3 and 4, the d_i value is shown as the difference between the estimated RUL and the actual RUL value. In addition, the score function was suggested to compare the models on the Data Challenge at the 2008 PHM conference [31]. In the score function expressed in equations 6 and 7, the errors obtained as positive and negative produce different values. While the errors obtained as negative are more tolerable, the errors with positive values create higher scores as they can cause catastrophic failures.

$$S_i = \begin{cases} e^{-\frac{d_i}{13}} - 1, & for\ d_i < 0 \\ e^{\frac{d_i}{10}} - 1, & otherwise \end{cases} \tag{6}$$

$$Score = \sum_{i=1}^{N} S_i \tag{7}$$

3. Result and Discussion

In this section, the details of the proposed model and experimental work on the dataset are given. The procedures and methods of the experiments are explained. The results obtained with the proposed model are reported. In addition, the performances on the dataset in RUL estimation are given. Lastly, a comparative table of the results obtained with the proposed system and the other studies in the literature is presented.

3.1. Experiments

In the scope of this paper, the FD001 and the FD003 sub-datasets are selected from the C-MAPSS dataset. A new dataset is created, taking 21 sensor data that monitor aero-engines. Then, the dataset is normalized with the z-score normalization method and is made suitable for training. In order to make more accurate estimations by utilizing historical data, a time windowing technique has been applied to the dataset. The data was processed by determining a 30-length fixed time window (N_{tw}). Thus, each input data is sized as N_{tw} x N_{ft} with 21 selected sensors (N_{ft}). These processes have been applied to both the training and test datasets. RUL labeling was done with a piece-wise

linear degradation model for the values to be estimated during the training. Thus, the necessary preparations for training the proposed CSARN model were made, and then the hyperparameter settings were made.

In the CSARN model, Self-Attention layers and ResNet layers are formed by cascade connection. Each layer and its hyperparameters in the model are given in Table 3. The model coefficients are adjusted to be optimized with adam optimizer [32]. The batch size was determined as 32 during model training. The random validation split was set as 20% for model validation during training. The model is set to train for 250 epochs. Early stopping was used to prevent overfitting. Model training is stopped 20 epochs after the training loss, and validation loss values start to diverge. Thus, model training was stopped beforehand to prevent overfitting before reaching the maximum number of epochs.

Table 3.The details of the CSARN model

Layer Name	Input Size	Output Size	Description
Gaussian Layer	30x21x1	30x21x1	Std value=0.01
Conv 1	30x21x1	30x21x10	11x1,10 filter
Self-Attention	30x21x10	30x21x10	
ResNet Layer 1	30x21x10	30x21x10	
Conv 2	30x21x10	15x21x10	7x1,10 filter, stride 2x1
ResNet Layer 2	15x21x10	15x21x10	
Conv 3	15x21x10	8x21x10	7x1,10 filter, stride 2x1
ResNet Layer 3	8x21x10	8x21x10	
Conv 4	8x21x10	2x21x5	7x1,5 filter
Flatten	2x21x5	210	
Dropout + FC	210	140	Dropout rate=0.3
Dropout + FC	140	90	Dropout rate=0.3
FC	90	1	

Also, the loss function is proposed, like the score function from evaluation metrics. The proposed loss function is shown in equation 7.

$$loss = 5 \times \begin{cases} -d_i \times e^{-\frac{d_i}{13}} - 1, & for\ d_i < 0 \\ d_i \times e^{\frac{d_i}{10}} - 1, & otherwise \end{cases} \quad (8)$$

In the above equation, d_i represents the difference between the predicted RUL and the actual RUL value. The trained model was used for RUL estimation in the test dataset. As with the score function, a loss function that is designed could tolerate negative values.

3.2. Experimental Analysis and Results

In this section, the performance of the proposed CSARN model on the FD001 and FD003 datasets has been examined. Model training was done with the training set of each sub-dataset and tested on its test dataset. There are 100 motors in the test datasets. In the data of these engines, data has been collected from the beginning, and at a point before the end of the engine's life, the data acquisition from the engine was stopped. As of this point, the RUL of the engine has been estimated. The estimation values obtained were also interpreted and examined according to the evaluation metrics used in the literature.

The estimations obtained for each engine in the FD001 test dataset are shown in Figure 6 from smallest to largest for better visualization. In addition, the actual RUL values and the RUL labeling part suitable for the piece-wise linear degradation model are given together. As seen in the figure, the estimations made towards the end of the life of the motors give more precise results. In addition, it is seen that the variance of the results obtained in the middle part of the estimated values is higher than in other parts.

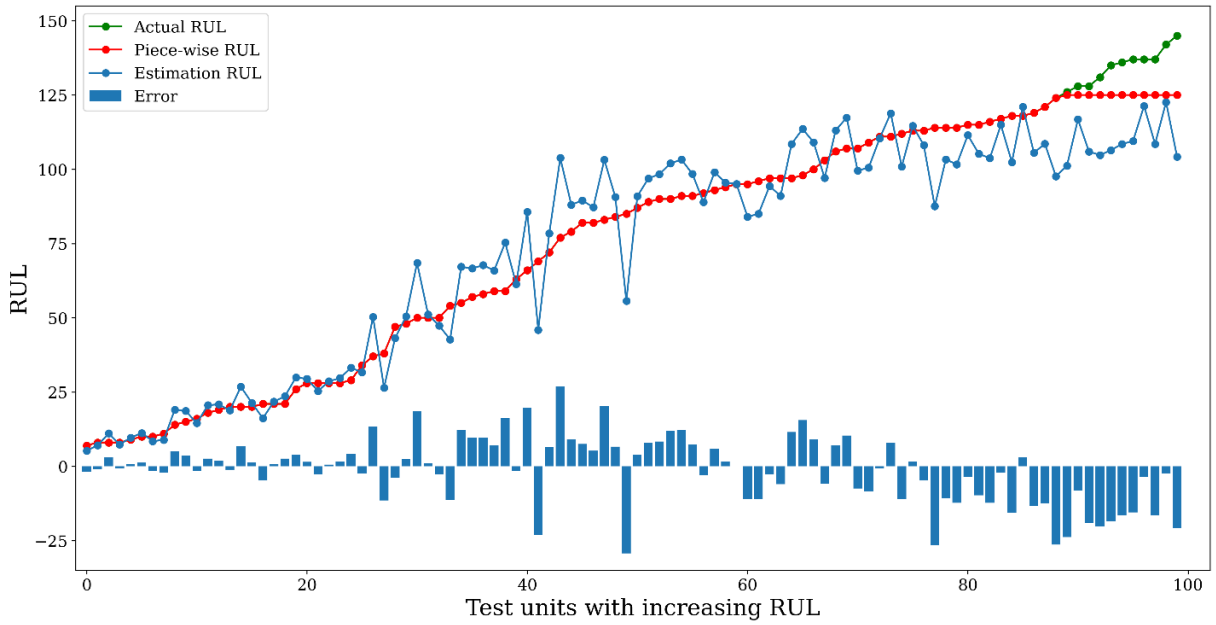

Figure 6. RUL estimation of all engines in the FD001 test dataset

In the same way, the engine's lives are predicted for the FD003 test dataset and are shown in Figure 7. The graph shows the estimated RUL values, piece-wise RUL values, and actual RUL values. The obtained results with high score values are expected with the same model because the FD003 dataset has two failure modes. As in the FD001 dataset, it is seen that the variance is higher in the middle part of the RUL values.

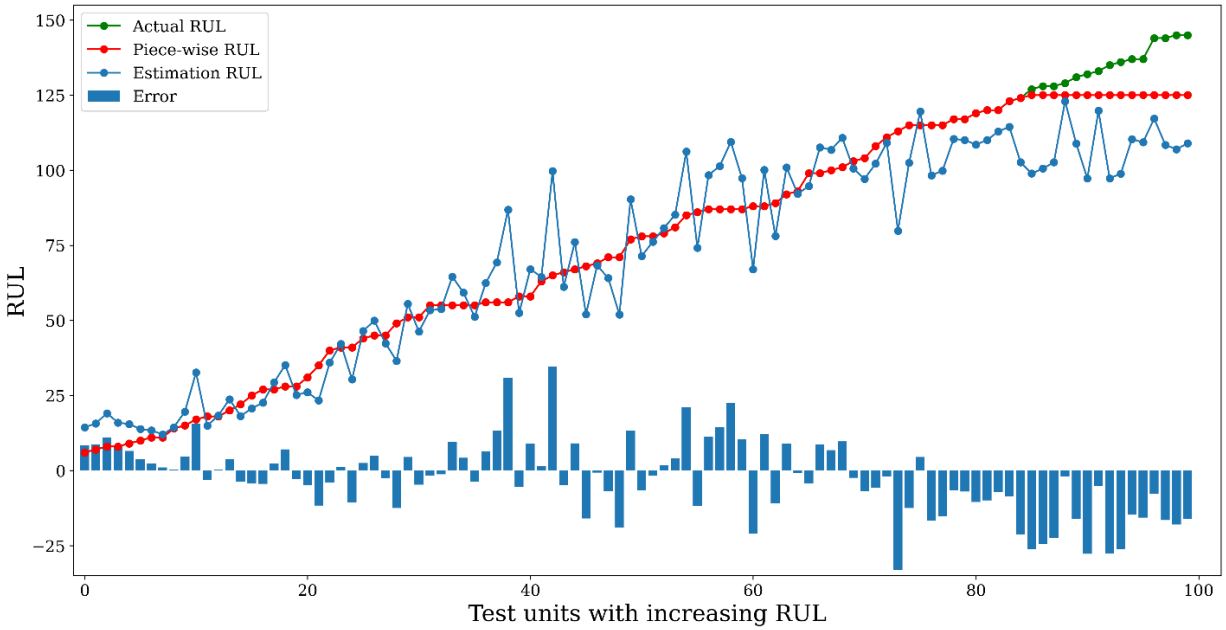

Figure 7. RUL estimation of all engines in the FD003 test dataset

The evaluation results made on the sub-datasets used with the proposed model are given in Table 4. The evaluations were obtained with the RMSE and Score functions used in the literature. In the FD001 and FD003 sub-datasets, RMSE values were obtained as 11.017 and 12.629, respectively. Likewise, when the Score function is evaluated, it is seen that the results obtained are 157.19 and 218.6. Also, the obtained MAPE results are 0.127 and 0.193, respectively. When the dataset is examined under a single operating condition but in two different failure modes, the results obtained with the same model are higher. This situation shows that the data obtained from the system has become more complex, and the degradation model has become harder to predict.

Table 4. The obtained results via the proposed model

	FD001			FD003		
	RMSE	Score	MAPE	RMSE	Score	MAPE
CSARN	11.017	157.19	0.127	12.629	218.6	0.193

To illustrate the effect of the model on the motors in the training and test dataset, real and estimated RUL plots of randomly selected motors from the FD001 are shown in Figure 8. Likewise, random motors were selected in the FD003 dataset and are shown in Figure 9.

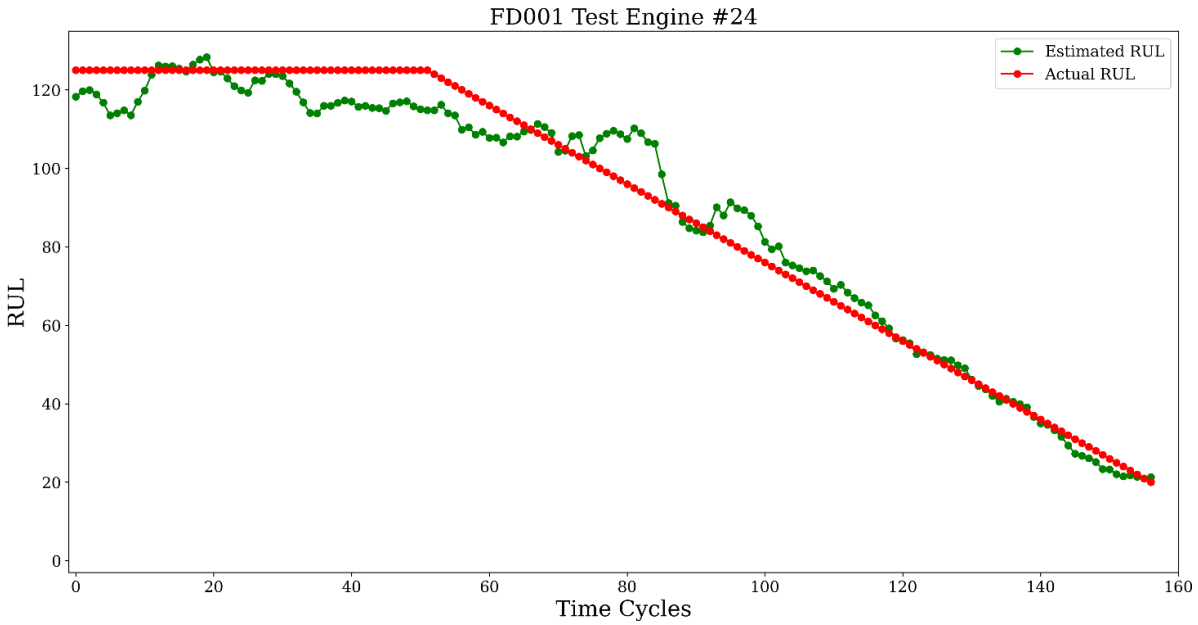

Figure 8. RUL prediction results in each time cycles engine no 24

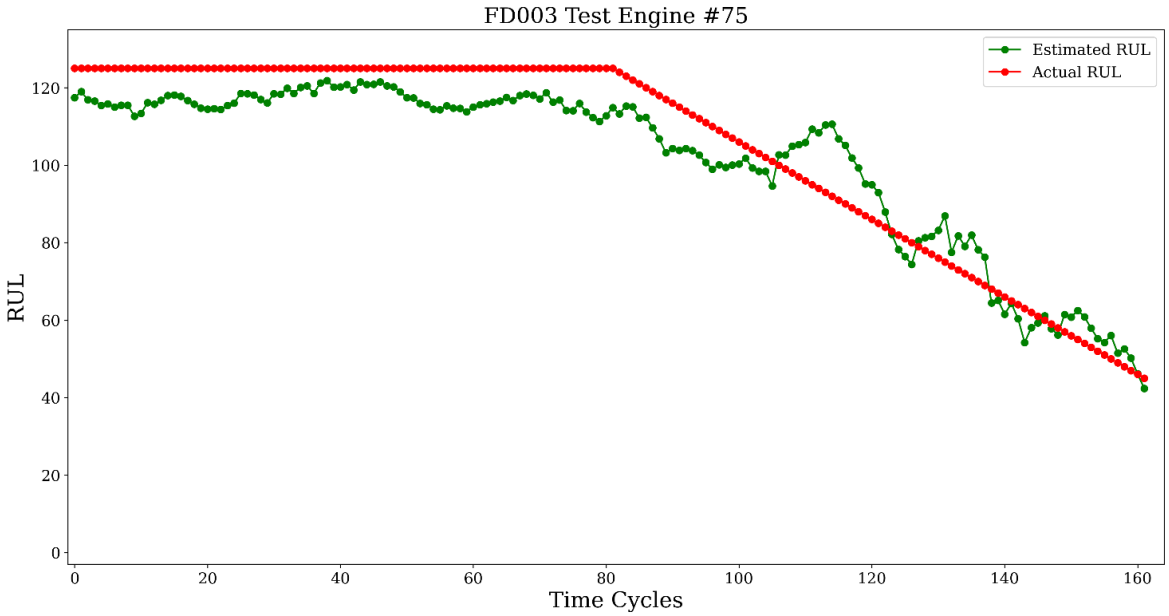

Figure 9. RUL prediction results in each time cycles engine no 75

3.3. Comparison with other studies

In this section, experimental studies and other state-of-the-art methods in the literature are compared. The results obtained in the comparison are given in chronological order. The results of the proposed CSARN method and other methods are shown in Table 5.

Table 5. Comparison the CSARN model and the state-of-the-art models in literature

Methods	Year	FD001		FD003	
		RMSE	Score	RMSE	Score
MODBNE [11]	2016	15,04	334,23	12,51	421,91
D-LSTM [13]	2017	16,14	338	16,18	852
HDNN [18]	2019	13,017	245	12,22	287,72
CNNTW [33]	2019	12,18	224,16	15,67	1279,85
CapsNet [17]	2019	12,58	276,34	11,71	283,81
DAG [20]	2019	11,96	229	12,46	535
NPBLSTM [19]	2020	12,321	238,34	**11,364**	226,482
ATS2S [22]	2020	12,63	243	11,44	263
CNN-BiLSTM [34]	2021	12,13	174	11,96	242
CNN+ATT [24]	2021	11,48	198	12,31	251
LSTM-MLSA [30]	2021	11,567	252,86	12,134	370,39
Bi-LSTM-Two Stream [35]	2022	11,96	206,33	13,41	223,36
BiGRU-TSAM [36]	2022	12,56	213,35	12,45	232,86
CSARN	2022	**11,017**	**157.19**	12,629	**218,6**

As can be seen in Table 5, the proposed model gave the best results in the evaluation of score function in the FD001 and FD003 sub-datasets compared to other models in the literature. However, in the FD003 sub-dataset, obtained the RMSE value is not the best, but it is a fair result. Especially in the score function, much better success was achieved compared to other studies. In the proposed model, the Self-Attention layer and the loss function that we have proposed have been effective.

4. Conclusion

In this study, the effect of the cascaded structure of Self-Attention and ResNet layers on RUL estimation was investigated. It was focused on important points with the Self-Attention layer for better estimation in the time domain. The model was established as an end-to-end structure that can do automatic feature extraction with Convolution layers. In addition, with the ResNet structure integrated into the model, the effect of the loss value from the last layer to the first layer is effectively spread. Different actions were taken to prevent overfitting within the study's scope and in the tested models. First of all, Gaussian noise was added by modeling the data. Thus, it was ensured that the model works more robustly or produces results in noisy data. In addition, it was tried to prevent the model from memorizing during training by adding dropout layers between FC layers. However, the weight coefficients in each model layer were multiplied by a regularization coefficient to prevent overfitting. Finally, while training the model, it was ensured that the training process was stopped without memorizing the proposed model with the early stopping mechanism. The performance of the proposed model was tested on the FD001 and FD003 sub-datasets of the C-MAPSS dataset. The best score value was obtained in these sub-datasets. In addition, the lowest RMSE value was found in the FD001 sub-dataset. These sub-datasets have been chosen because training can be done without structural changes in the proposed model.

Compared to other state-of-the-art models in the literature, successful results were obtained on RUL estimation. In future studies, the proposed model's extension to other sub-datasets and its performance in other test datasets will be examined within the scope of this study. Also, on the second loss function, which merges with the proposed loss function because of the success on the score function, will be worked on. In addition, the contributions of other dynamic network structures (LSTM, GRU) found in the literature to the proposed model structure will be investigated in future studies.

References

[1] Xu, J., Wang, Y., & Xu, L. (2013). PHM-oriented integrated fusion prognostics for aircraft engines based on sensor data. IEEE Sensors Journal, 14(4), 1124-1132.

[2] Nandi, S., Toliyat, H. A., & Li, X. (2005). Condition monitoring and fault diagnosis of electrical motors—A review. IEEE transactions on energy conversion, 20(4), 719-729.

[3] Kara, A. (2021). A data-driven approach based on deep neural networks for lithium-ion battery prognostics. Neural Computing and Applications, 33(20), 13525-13538.

[4] Park, J., & Jung, W. (2015). A systematic framework to investigate the coverage of abnormal operating procedures in nuclear power plants. Reliability Engineering & System Safety, 138, 21-30.

[5] Hou, G., Xu, S., Zhou, N., Yang, L., & Fu, Q. (2020). Remaining useful life estimation using deep convolutional generative adversarial networks based on an autoencoder scheme. Computational Intelligence and Neuroscience, 2020.

[6] Li, H., Zhao, W., Zhang, Y., & Zio, E. (2020). Remaining useful life prediction using multi-scale deep convolutional neural network. *Applied Soft Computing*, *89*, 106113.

[7] Correia, J. A., De Jesus, A. M., & Fernández-Canteli, A. (2012). A procedure to derive probabilistic fatigue crack propagation data. *International Journal of Structural Integrity*.

[8] Liao, L. (2013). Discovering prognostic features using genetic programming in remaining useful life prediction. *IEEE Transactions on Industrial Electronics*, *61*(5), 2464-2472.

[9] Li, N., Lei, Y., Lin, J., & Ding, S. X. (2015). An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Industrial Electronics*, *62*(12), 7762-7773.

[10] Sateesh Babu, G., Zhao, P., & Li, X. L. (2016, April). Deep convolutional neural network based regression approach for estimation of remaining useful life. In International conference on database systems for advanced applications (pp. 214-228). Springer, Cham.

[11] Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2016). Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE transactions on neural networks and learning systems, 28(10), 2306-2318.

[12] Li, X., Ding, Q., & Sun, J. Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety, 172, 1-11.

[13] Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017, June). Long short-term memory network for remaining useful life estimation. In 2017 IEEE international conference on prognostics and health management (ICPHM) (pp. 88-95). IEEE.

[14] Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing, 275, 167-179.

[15] Yu, W., Kim, I. Y., & Mechefske, C. (2019). Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mechanical Systems and Signal Processing, 129, 764-780

[16] Wang, J., Wen, G., Yang, S., & Liu, Y. (2018, October). Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network. In 2018 Prognostics and System Health Management Conference (PHM-Chongqing) (pp. 1037-1042). IEEE.

[17] Ruiz-Tagle Palazuelos, A., Droguett, E. L., & Pascual, R. (2020). A novel deep capsule neural network for remaining useful life estimation. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 234(1), 151-167.

[18] Al-Dulaimi, A., Zabihi, S., Asif, A., & Mohammadi, A. (2019). A multimodal and hybrid deep neural network model for remaining useful life estimation. Computers in industry, 108, 186-196.

[19] Al-Dulaimi, A., Zabihi, S., Asif, A., & Mohammed, A. (2020). NBLSTM: Noisy and hybrid convolutional neural network and BLSTM-Based deep architecture for remaining useful life estimation. Journal of Computing and Information Science in Engineering, 20(2), 021012.

[20] Li, J., Li, X., & He, D. (2019). A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction. IEEE Access, 7, 75464-75475.

[21] Song, Y., Shi, G., Chen, L., Huang, X., & Xia, T. (2018). Remaining useful life prediction of turbofan engine using hybrid model based on autoencoder and bidirectional long short-term memory. Journal of Shanghai Jiaotong University (Science), 23(1), 85-94.

[22] Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Yan, R., & Li, X. (2021). Attention-based sequence to sequence model for machine remaining useful life prediction. Neurocomputing, 466, 58-68.

[23] Liu, L., Song, X., & Zhou, Z. (2022). Aircraft engine remaining useful life estimation via a double attention-

based data-driven architecture. Reliability Engineering & System Safety, 221, 108330.

[24] Tan, W. M., & Teo, T. H. (2021). Remaining useful life prediction using temporal convolution with attention. Ai, 2(1), 48-70.

[25] Xia, J., Feng, Y., Lu, C., Fei, C., & Xue, X. (2021). LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems. Engineering Failure Analysis, 125, 105385.

[26] Zhao, H., Jia, J., & Koltun, V. (2020). Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10076-10085).

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[29] Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, October). Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 international conference on prognostics and health management (pp. 1-9). IEEE.

[30] Xia, J., Feng, Y., Lu, C., Fei, C., & Xue, X. (2021). LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems. Engineering Failure Analysis, 125, 105385.

[31] Heimes, F. O. (2008, October). Recurrent neural networks for remaining useful life estimation. In 2008 international conference on prognostics and health management (pp. 1-6). IEEE.

[32] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[33] Yang, H., Zhao, F., Jiang, G., Sun, Z., & Mei, X. (2019). A novel deep learning approach for machinery prognostics based on time windows. Applied Sciences, 9(22), 4813.

[34] Song, J. W., Park, Y. I., Hong, J. J., Kim, S. G., & Kang, S. J. (2021, May). Attention-based bidirectional LSTM-CNN model for remaining useful life estimation. In 2021 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.

[35] Jin, R., Chen, Z., Wu, K., Wu, M., Li, X., & Yan, R. (2022). Bi-LSTM-Based Two-Stream Network for Machine Remaining Useful Life Prediction. IEEE Transactions on Instrumentation and Measurement, 71, 1-10.

[36] Zhang, J., Jiang, Y., Wu, S., Li, X., Luo, H., & Yin, S. (2022). Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. Reliability Engineering & System Safety, 221, 108297.