A Comparative Study of CNN-Based Predictive Maintenance Techniques on C-MAPSS Dataset

İlyas Çiçek
Electrical and Electronics Engineering
Kayseri University
38280 Kayseri, Türkiye
ilyascicek@kayseri.edu.tr

Engin Cemal Mengüç
Electrical and Electronics Engineering
Kayseri University
38280 Kayseri, Türkiye
enginmenguc@kayseri.edu.tr

Nurettin Acır
Electronics Engineering
Turkish Air Force Academy
National Defence University
34149 İstanbul, Türkiye
nacir@hho.msu.edu.tr

2025 IEEE 7th Symposium on Computers & Informatics (ISCI) | 979-8-3315-1554-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISCI65687.2025.11167805

***Abstract*—In this study, convolutional neural network (CNNs) based predictive maintenance (PdM) techniques are performed on the well-known jet engine data called the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). We first employ the one-dimensional CNN (1D-CNN) for predicting the remaining useful life (RUL) of the jet engines. Then, to further improve the prediction performance, we combine the long short-term memory (LSTM), bidirectional LSTM (BiLSTM), and gated recurrent unit (GRU) layers with the 1D-CNN. The performances of all CNN architectures are comparatively evaluated in terms of root mean square error (RMSE), training time, and correlation coefficient metrics. The simulation results show that the integration of LSTM, BiLSTM, and GRU layers with 1D-CNN improves the RUL prediction performance in terms of both correlation coefficient and total RMSE, compared to the classical 1D-CNN model.**

***Index Terms*—Predictive Maintenance, Convolutional Neural Network, LSTM, BiLSTM, GRU**

I. Introduction

With the transition of modern industry into the era of Industry 4.0, industrial environments have become increasingly interconnected and intelligent [1]. One of the most significant consequences of this transformation is the generation of large amounts of data by machines, sensors, and other industrial devices. These massive data streams containing valuable information can lead to substantial improvements in operational efficiency, reliability, and decision-making processes if they are properly analyzed [2]. Advanced data analytics techniques enable the extraction of meaningful patterns and trends from these complex datasets [3]. In particular, predictive analytics has emerged as a powerful tool to prevent equipment failures, allowing for proactive maintenance and minimizing unexpected downtime [4]. This capability not only detects the effectiveness of the single machine but also optimizes the overall performance of the rest of the system. As a result, big data analytics plays a crucial role in driving smarter, safer, and more sustainable industrial systems.

To address the challenges caused by unplanned downtime and inefficient equipment usage, one of the most effective solutions is predictive maintenance (PdM). Unlike traditional maintenance strategies, the PdM relies on real-time data and intelligent analytics to predict when a piece of equipment is likely to fail [5]. The PdM evaluates data collected from sensors, machine logs, and other industrial sources to monitor the current condition of assets. By applying machine learning algorithms and statistical models, the PdMs can detect early signs of wear, anomalies, or performance degradation. This also enables maintenance teams to intervene at the optimal time—neither too early, which can lead to unnecessary costs, nor too late, which can result in unexpected breakdowns [6]. As a result, predictive maintenance not only improves equipment performance but also reduces operational costs, extends asset life cycles, and improves safety across different industrial environments. It stands as a cornerstone technology in the realization of Industry 4.0's vision of smarter, data-driven manufacturing [7]. Furthermore, the PdM is widely adopted across various industries due to its potential to reduce costs and prevent failures. In manufacturing, it can be used as continuous production by monitoring machinery such as CNC machines, conveyors, and robotic arms [8]–[10]. In the automotive and aerospace sectors, PdM is used to track the health of critical components such as engines, brakes, and avionics systems to enhance safety and reliability [11], [12]. Energy and utility companies rely on PdM to monitor infrastructure such as wind turbines, transformers, and pipelines, where equipment failure can lead to significant service disruptions [13]–[15]. Even in healthcare, predictive approaches are increasingly applied to maintain sophisticated medical equipment, ensuring that lifesaving devices operate without interruption [16].

Convolutional Neural Networks (CNNs) have gained significant attention in predictive maintenance applications due to their ability to automatically extract hierarchical and spatial features from raw sensor data. In recent years, numerous studies have employed 1D-CNNs to process time-series signals, such as vibration [17], temperature [18], or pressure data [19], for the estimation of Remaining Useful Life (RUL) or early fault detection. These models are particularly effective in capturing local temporal patterns and reducing the need for manual feature engineering. CNN-based approaches have demonstrated promising results in various industrial settings, including aerospace [20], manufacturing [21], and energy systems [22]. However, despite their strengths, conventional CNNs may struggle to capture long-term temporal dependencies in sequential data, which has motivated the integration of CNNs with recurrent neural networks (RNNs) to further

enhance predictive performance.

In this study, CNN-based PdM is applied to the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset [23], which contains multivariate time-series data simulating engine degradation under various operating conditions and failure modes. The main goal here is to estimate the RUL of engines, that a critical metric for preventing in-flight failures and optimizing maintenance schedules, by using conventional 1D-CNN and its modified versions with Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Gated Recurrent Unit (GRU) layers [24]–[26]. To this end, the 1D-CNN is first employed to extract spatial features from the sensor data. Unfortunately, since temporal dependencies play a key role in understanding degradation patterns, the 1-DCNN may be insufficient for predicting the RUL. Thus, in this study, its performance is improved by integrating recurrent layers such as the LSTM, BiLSTM, and GRU layers after the convolutional layers to better capture sequential dynamics and long-term dependencies within the sensor data. Throughout this study, the architectures are called 1D-CNN, 1D-CNN+LSTM, 1D-CNN+BiLSTM, and 1D-CNN+GRU. Their RUL performances are assessed in terms of the root mean square error (RMSE), training time, and correlation coefficient metrics. Our simulation results indicate that the inclusion of recurrent layers in the 1D-CNN architecture significantly improves prediction accuracy. This study also highlights that modeling the spatial and temporal characteristics of sensor data together in industrial applications is significant for predictive maintenance tasks.

II. Overview of CNN, LSTM, BiLSTM, and GRU

This section presents a brief overview of the CNN, LSTM, BiLSTM, and GRU architectures utilized in this study.

A. Convolutional Neural Network

CNNs were first introduced by Yann LeCun in the late 1980s [27]. LeCun's work laid the foundation for modern deep learning approaches in computer vision. Although CNN-like structures existed earlier, LeCun's contribution was pivotal in combining convolutional layers with backpropagation training and practical implementation on real-world data. CNNs have become the cornerstone of many modern machine learning systems, particularly in areas requiring spatial data understanding. Some of the applications are image classification, object detection, time series analysis, and so on [28]–[30]. The CNN is a type of deep, feedforward neural network that uses convolution layers instead of or in addition to fully connected layers. The central operation is the convolution, which applies a kernel (or filter) to all the input data to create feature maps. The generalized form of convolution storage used to generate a single output feature map for the 1D-CNN architecture can be defined as $y_i = (x * w)_i + b_i = \sum_{k=0}^{K-1} w_k x_{i+k} + b_i$, where y_i, x_i, w_i, and b_i are the output signal, input signal (or feature map), kernel, and bias at time index i, respectively.

B. Long Short Term Memory

The LSTM is a special form of RNN to better represent long-term dependencies in sequential data [24]. Traditional RNNs have difficulty retaining features across time-consuming sequences because of the vanishing gradient problem. LSTMs overcome this problem by introducing a memory cell and three key gates that regulate the flow of information. These gates allow the network to decide what information to keep, discard, or output at each time step, making LSTMs more effective for learning patterns across longer time intervals. As a result, LSTMs are more robust than basic RNNs when handling time-series data with complex temporal relationships. In Fig. 1, a schematic overview of the LSTM structure is provided, where X_t, C_t, and h_t represent the input, cell state, and hidden state at time index t, respectively. Also, $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and tangent hyperbolic functions, respectively. The LSTM cell can be described by [31]:

$$i_t = \sigma(W_{xi}x_t + U_{hi}h_{t-1} + b_i) \quad (1)$$

$$g_t = \tanh(W_{xg}x_t + U_{hg}h_{t-1} + b_g) \quad (2)$$

$$f_t = \sigma(W_{xf}x_t + U_{hf}h_{t-1} + b_f) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + U_{ho}h_{t-1} + b_o) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t x \odot g_t \quad (5)$$

$$h_t = \tanh(C_t) \odot o_t \quad (6)$$

where i_t, f_t, o_t, and $\odot$ are the input, forget, and output gates, and the multiplication of elements, respectively. Also, W, U, and b represent the feedforward weights, recurrent weights, and biases, respectively.

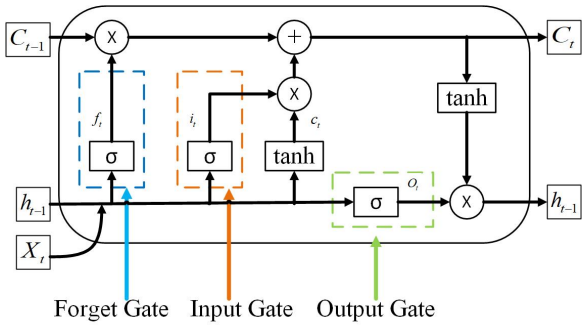


Fig. 1: Structure of a single LSTM cell [31]

C. Bidirectional Long Short Term Memory

The BiLSTM is an extension of the standard LSTM, which simultaneously processes input data considering both forward and backward directions [25]. While traditional LSTM only considers past information (from left to right in a data sequence), BiLSTMs also incorporate future context by processing the sequence in reverse simultaneously. This dual perspective enables the network to capture more comprehensive temporal relationships within the data, making BiLSTMs particularly effective in tasks where the entire sequence context is important, such as time series prediction or natural language processing. Despite being computationally complex, BiLSTM

often enhances the performance in many engineering applications due to its richer sequence understanding. The BiLSTM structure is given in Fig. 2, where x_t and y_t are the input and output signals at time index t, respectively. The mathematical model of the BiLSTM is given as

$$y_t = W_{\vec{h}_y}\vec{h}_t + W_{\overleftarrow{h}_y}\overleftarrow{h}_t + b_y \tag{7}$$

$$\vec{h}_t = \mathcal{H}(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \tag{8}$$

$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t) + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \tag{9}$$

where $\vec{h}$, $\overleftarrow{h}$, and $\mathcal{H}$ represent the forward hidden layer, the backwards hidden layer, and Hilbert space, respectively [31].

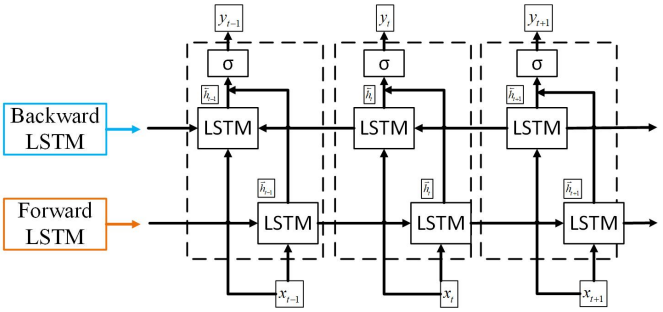


Fig. 2: Internal structure of BiLSTM cell [31]

D. Gated Recurrent Unit

The GRU is designed to model sequential data by using fewer parameters, which is known as a simplified version of LSTM [26]. The GRU has two gates called the forget and input gates. This structure makes them a single update gate and uses a reset gate to control how new input is combined with previous memory. The GRU differs from the LSTM in that it stores all information in the hidden state, without using a separate memory cell. Owing to its streamlined structure, the GRU trains more efficiently and with lower computational cost, yet remains effective in modeling temporal dependencies. Due to their comparable performance to the LSTM, GRUs are widely adopted in various time-series and sequence modeling applications. The general layout of the GRU structure is demonstrated in Fig. 3. Also, the equations of the GRU are given by

$$z_t = \sigma(W_{xi}x_t + U_{hi}h_{t-1} + b_i) \tag{10}$$

$$r_t = \sigma(W_{xf}x_t + U_{hf}h_{t-1} + b_f) \tag{11}$$

$$\hat{h}_t = \tanh(W_{xg}x_t + U_{hg}(r_t \odot h_{t-1}) + b_g) \tag{12}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{13}$$

where z_t and r_t represent the update and reset gates [31].

III. Proposed PdM based on CNN Architectures

The overall architecture of the proposed PdM framework based on the 1D-CNN, 1D-CNN+LSTM, 1D-CNN+BiLSTM, and 1D-CNN+GRU models is illustrated in Fig. 4. As depicted, the proposed PdM approach comprises three primary

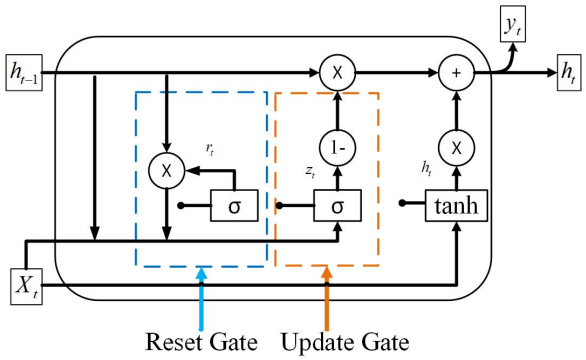


Fig. 3: Structure of a single GRU cell [31]

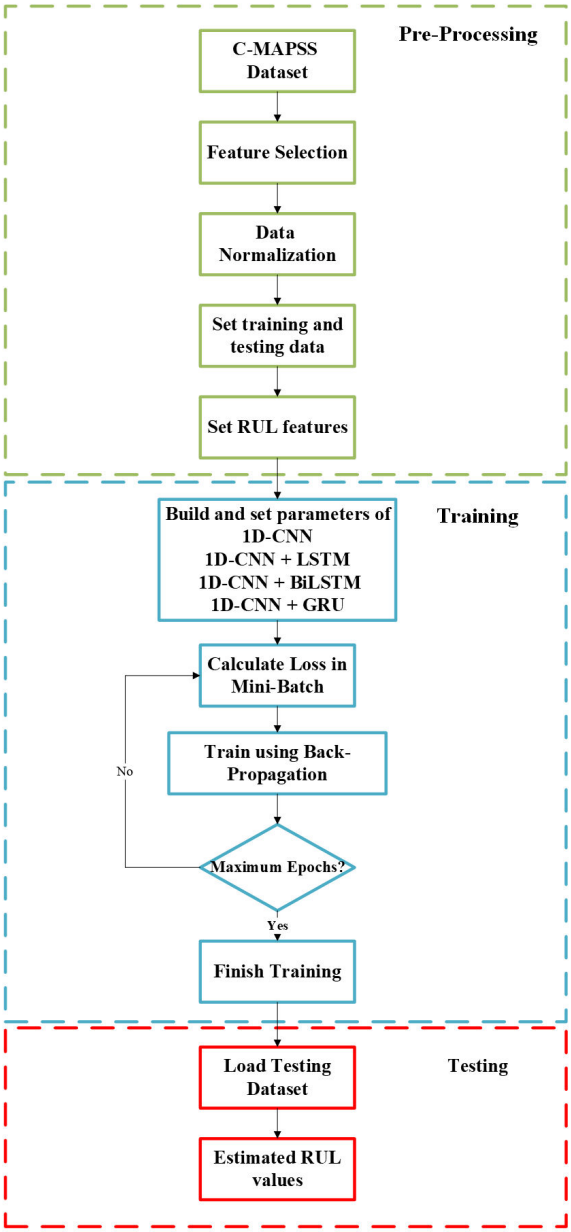


Fig. 4: Overall architecture of proposed PdM framework.

TABLE I: Information of C-MAPSS dataset

Dataset	FD001	FD002	FD003	FD004
Number of engine for training	100	260	100	249
Number of engine for testing	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2
Training samples	17,731	48,819	21,82	57522
Testing samples	100	259	100	248

TABLE II: Hyperparameters and simulation settings

Parameter	Value	Parameter	Value
Filter number	10	Time seq dimension of FD001	30
Filter Length	10	Neurons in fully-connected layer	100
Number of Features	14	Dropout rate	0.5
Constant RUL value	125	Convolution Layers	5
Batch size	512	Epoch number	250

stages: pre-processing, training, and testing. These stages are summarized as follows.

In the pre-processing stage, similar to [32], the corresponding raw data from the C-MAPSS dataset is first subjected to a series of steps, including feature selection, normalization, and partitioning into training and testing datasets. It should be noted that we used the FD001 subset from the C-MAPSS dataset summarized in Table I. This subset contains data from 100 engines for training and testing procedures and is characterized by a single operating condition and a single fault mode. Additionally, each instance in the FD001 includes measurements from 21 distinct sensors that capture various engine parameters. At the feature selection, we use the prognosability MATLAB command to measure the variability of condition indicators at failure, thus eliminating unnecessary sensor data from the training data. Then, the normalization is applied to scale the input features to a common range, thereby enhancing the convergence rate and stability of the training process. Finally, to enable standardized RUL prediction, the RUL targets are defined accordingly.

In the training stage, the processed data from the previous stage is used to train four deep learning architectures: 1D-CNN, 1D-CNN+LSTM, 1D-CNN+BiLSTM, and 1D-CNN+GRU. The model architectures are constructed and their hyperparameters are configured as detailed in Table II. Training is carried out using a mini-batch strategy, wherein the loss is computed iteratively and minimized through backpropagation. This process is repeated over a predefined number of epochs, at which point the model training is considered complete.

Finally, in the testing stage, the trained CNN-based models are evaluated using a separate test dataset. This phase yields the predicted RUL values, which serve as indicators of the model's predictive capability. To assess performance, the predicted RUL values are compared against the actual RUL values using evaluation metrics such as the RMSE and the correlation coefficient.

IV. Simulation Results and Discussion

In this section, we will compare the performances of the proposed PdM structures based on 1D-CNN, 1D-CNN+LSTM, 1D-CNN+BiLSTM, and 1D-CNN+GRU on the C-MAPSS dataset in terms of RMSE, training time, and correlation coefficient. The C-MAPSS dataset, developed by NASA, serves as a benchmark to simulate the degradation of aircraft engines and is widely used in the field of prognostics and health management (PHM) [23]. The C-MAPSS is included in the NASA Prognostics Data Repository and has become a widely adopted resource in machine learning research, particularly for predictive maintenance and RUL prediction. It is generated through a high-fidelity simulation of turbofan engines operating under various conditions and fault modes. The CMAPSS dataset is separated into four subsets, which are FD001, FD002, FD003, and FD004, respectively. Their properties are summarized in Table I. As mentioned previously, we used the FD001 subset to measure the performance of the proposed PdM structures in this study. All experiments were conducted on a PC equipped with an Intel Core i5-11500T processor (1.5 GHz) and 8 GB of RAM, using the MATLAB R2024b environment. Also, as in [32], the hyperparameters and simulation settings are given in Table II.

To measure the RUL prediction performances of the designed network architectures, we used the RMSE and correlation coefficient (CC), given by

$$CC = \frac{cov\,(d, y)}{\sigma_d \sigma_y}, \tag{14}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{K} (d_i - y_i)^2}{K}} \tag{15}$$

where $cov(\cdot)$ denotes the covariance function, while σ_d and σ_y represent the standard deviations of the actual (d) and predicted (y) RUL values, respectively. Moreover, in (2), d_i and y_i correspond to the actual and predicted RUL at time index i.

Table III presents the average testing correlation coefficients, testing RMSE values, and training times of the network architectures across all engines. Here, Fig. 5 shows a comparison of the RUL prediction of the corresponding architectures tested on engine data for randomly selected engines. These values were obtained by averaging the results of 10 independent simulation runs. Based on the results summarized in Table III and illustrated in Fig. 5, the following observations can be made.

- Firstly, the correlation coefficient values significantly increased when the new layers, such as LSTM, BiLSTM, and GRU, were added to the conventional 1D-CNN. Especially, the 1D-CNN+BiLSTM structure produced the best value among all other network structures.
- In terms of RMSE values, the 1D-CNN+GRU surprisingly yields the best overall performance. This is primarily because, when examining the RMSE values across all engines individually, the 1D-CNN+GRU consistently maintained RMSE values within a narrow range of 15 to 17. In contrast, although other architectures achieved

TABLE III: Simulation results in terms of testing correlation coefficients, testing RMSE values, and training times

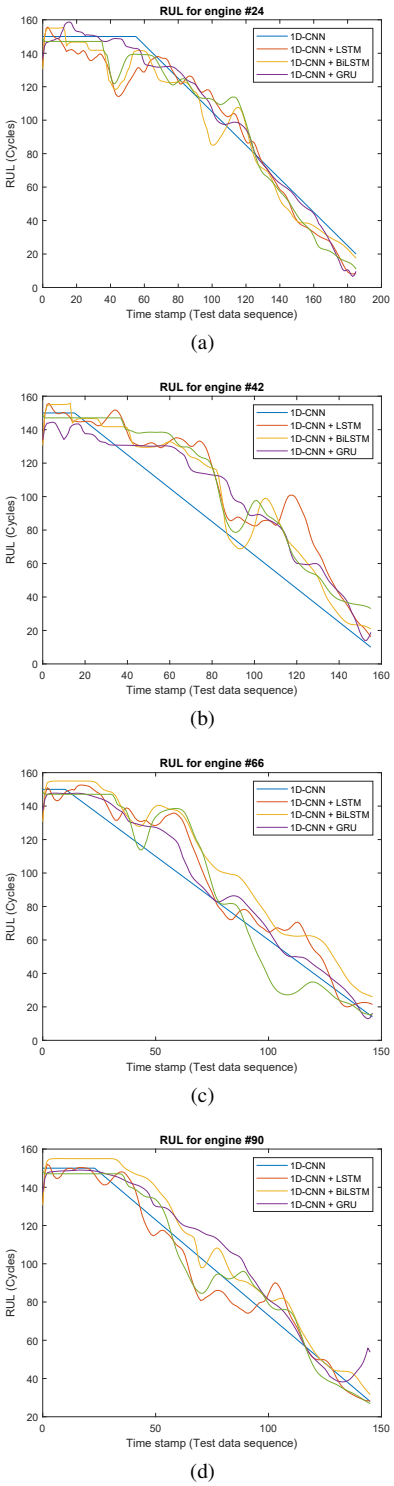
Network Architecture	Testing Correlation Coefficient	Testing RMSE	Training Time (sec)
1D-CNN	0.7336	17.47	158.01
1D-CNN + LSTM	0.7975	17.55	146.88
1D-CNN + BiLSTM	0.8494	19	185.32
1D-CNN + GRU	0.7962	16.69	186.95

better performance than the 1D-CNN+GRU on most engines, they also exhibited significantly larger deviations in some cases, which adversely affected their overall average performance. It should be noted that, as observed from the RMSE and correlation coefficient results of the 1D-CNN + BiLSTM model, a higher correlation does not necessarily imply higher prediction accuracy. This is because the correlation coefficient primarily captures the trend alignment between the predicted and actual values. However, during the prediction process, significant errors may still arise due to bias or scale mismatches, which can lead to a high RMSE despite a strong correlation.

- Furthermore, the training times across all architectures were broadly similar, which indicates no serious computational advantage for any particular architecture.
- As illustrated in Fig. 5, the proposed PdM architectures demonstrated a serious ability to accurately predict the true RUL. It is worth noting that these performances could be further enhanced through the use of alternative network architectures, deeper or more specialized layers, and advanced feature selection strategies.

V. Conclusion

In this study, we have applied 1D-CNN-based PdM methods to the C-MAPSS dataset for RUL prediction. By enhancing 1D-CNN with LSTM, BiLSTM, and GRU layers, it has achieved high accuracy in capturing temporal dependencies in sensor data. The results have shown that the proposed 1D-CNN-based PdM methods using the LSTM, BiLSTM, and GRU layers have significantly outperformed the conventional 1D-CNN in terms of RUL prediction, RMSE, and correlation. Overall, this study has shown that combining spatial and temporal modeling is essential for effective predictive maintenance. However, it is worth noting that although the proposed PDM architectures were trained on a specific dataset and demonstrate satisfactory performance, their predictive accuracy may deteriorate when confronted with noisy or incomplete data in real-world production environments. Moreover, since they were optimized for a particular component type and failure scenario, their robustness and generalizability to other machines or systems may be limited. In our future work, considering the aforementioned limitations, the performance of the proposed models could be further enhanced by exploring deeper or more specialized architectures, incorporating attention mechanisms, and integrating advanced feature selection techniques.



(a) (b) (c) (d)

Fig. 5: RUL Predictions of randomly selected FD001 engines.

REFERENCES

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, pp. 239–242, 2014.

[2] M. Khan, X. Wu, X. Xu, and W. Dou, "Big data challenges and opportunities in the hype of industry 4.0," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[3] N. Elgendy and A. Elragal, "Big data analytics: a literature review paper," in *Advances in Data Mining. Applications and Theoretical Aspects: 14th Industrial Conference, ICDM 2014, St. Petersburg, Russia, July 16-20, 2014. Proceedings 14*. Springer, 2014, pp. 214–227.

[4] S. Abiteboul and M. Stonebraker, "Predictive analytics for dynamic workload management in cloud computing."

[5] H. Zheng, A. R. Paiva, and C. S. Gurciullo, "Advancing from predictive maintenance to intelligent maintenance with ai and iiot," *arXiv preprint arXiv:2009.00351*, 2020.

[6] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, "Intelligent prognostics tools and e-maintenance," *Computers in industry*, vol. 57, no. 6, pp. 476–489, 2006.

[7] T. Zonta, C. A. Da Costa, R. da Rosa Righi, M. J. de Lima, E. S. Da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers & Industrial Engineering*, vol. 150, p. 106889, 2020.

[8] W. Luo, T. Hu, Y. Ye, C. Zhang, and Y. Wei, "A hybrid predictive maintenance approach for cnc machine tool driven by digital twin," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101974, 2020.

[9] V. Pokhriyal, V. Prajwal, V. Kannan, D. K. Mahto, and V. Acharya, "Predictive maintenance for an industrial robotic arm using lora technology," in *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. IEEE, 2024, pp. 1–9.

[10] K. S. Kiangala and Z. Wang, "Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts," *The International Journal of Advanced Manufacturing Technology*, vol. 97, pp. 3251–3271, 2018.

[11] F. Arena, M. Collotta, L. Luca, M. Ruggieri, and F. G. Termine, "Predictive maintenance in the automotive sector: A literature review," *Mathematical and Computational Applications*, vol. 27, no. 1, p. 2, 2021.

[12] C. E. Dibsdale, *Aerospace Predictive Maintenance*. SAE International, 2020.

[13] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, and S. Trujillo, "Real-time predictive maintenance for wind turbines using big data frameworks," in *2017 ieee international conference on prognostics and health management (icphm)*. IEEE, 2017, pp. 70–77.

[14] H. de Faria Jr, J. G. S. Costa, and J. L. M. Olivas, "A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis," *Renewable and sustainable energy reviews*, vol. 46, pp. 201–209, 2015.

[15] S. Timashev and A. Bushinskaya, "Practical methodology of predictive maintenance for pipelines," in *International Pipeline Conference*, vol. 44205, 2010, pp. 329–338.

[16] A. Shamayleh, M. Awad, and J. Farhat, "Iot based predictive maintenance management of medical equipment," *Journal of medical systems*, vol. 44, no. 4, p. 72, 2020.

[17] A. Rihi, S. Baïna, F.-Z. Mhada, E. El Bachari, H. Tagemouati, M. Guerboub, I. Benzakour, K. Baïna, and E. H. Abdelwahed, "Innovative predictive maintenance for mining grinding mills: from lstm-based vibration forecasting to pixel-based mfcc image and cnn," *The International Journal of Advanced Manufacturing Technology*, vol. 135, no. 3, pp. 1271–1289, 2024.

[18] C. Gianoglio, E. Ragusa, P. Gastaldo, F. Gallesi, and F. Guastavino, "Online predictive maintenance monitoring adopting convolutional neural networks," *Energies*, vol. 14, no. 15, p. 4711, 2021.

[19] C. König and A. M. Helmi, "Sensitivity analysis of sensors in a hydraulic condition monitoring system using cnn models," *Sensors*, vol. 20, no. 11, p. 3307, 2020.

[20] D. Pebrianti, Z. Khalani, L. Bayuaji *et al.*, "Predictive maintenance in aerospace industry using convolutional neural network," in *2024 9th International Conference on Mechatronics Engineering (ICOM)*. IEEE, 2024, pp. 157–162.

[21] S. Al-Said, O. Findik, B. Assanova, S. Sharmukhanbet, and N. Baitemirova, "Enhancing predictive maintenance in manufacturing: A cnn-lstm hybrid approach for reliable component failure prediction," in *Technology-Driven Business Innovation: Unleashing the Digital Advantage, Volume 1*. Springer, 2024, pp. 137–153.

[22] K. P. Punitha, "Iot-powered robust anomaly detection and cnn-enabled predictive maintenance to enhance solar pv system performance," *IoT for Smart Grid: Revolutionizing Electrical Engineering*, pp. 243–255, 2025.

[23] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 international conference on prognostics and health management*. IEEE, 2008, pp. 1–9.

[24] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[28] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. Prudhvith, and N. Kumar, "Image classification using svm and cnn," in *2020 International conference on computer science, engineering and applications (ICCSEA)*. IEEE, 2020, pp. 1–5.

[29] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, and J. M. Z. Maningo, "Object detection using convolutional neural networks," in *TENCON 2018-2018 IEEE region 10 conference*. IEEE, 2018, pp. 2023–2027.

[30] L. Sadouk, "Cnn approaches for time series," *Time Series Analysis: Data, Methods, and Applications*, p. 57, 2019.

[31] K. Patra, R. N. Sethi, and D. K. Behera, "Anomaly detection in rotating machinery using autoencoders based on bidirectional lstm and gru neural networks," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 4, pp. 1637–1653, 2022.

[32] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.