Comparative Analysis of Predictive Maintenance Models for Turbofan Engines Using CNN, LSTM, and Reinforcement Learning.

Mjema Christina John

Abstract— **This project presents a comprehensive comparative study of machine learning models for predictive maintenance using NASA's turbofan engine degradation dataset. The dataset comprises a multivariate time series collected from a fleet of engines, where each engine undergoes degradation leading to eventual failure. The primary objective is to predict the remaining useful life (RUL) of engines based on operational settings and sensor measurements. Various machine learning models including Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Reinforcement Learning (RL) are evaluated for their predictive performance. Through this study, insights are provided into the efficacy of different neural network architectures for RUL prediction in industrial settings.**

I. PROJECT OVERVIEW

The project aims to develop predictive maintenance models capable of forecasting the remaining useful life of industrial components. The ability to predict RUL enables organizations to schedule maintenance activities efficiently, reducing downtime and maintenance costs while ensuring operational continuity. The project focuses on leveraging machine learning algorithms to analyze sensor data and historical maintenance records to predict when the failure of NASA's turbo engines is likely to occur.

II. INTRODUCTION

Predictive maintenance has emerged as a crucial strategy in industrial operations, aiming to mitigate the risks associated with unexpected equipment failures and optimize maintenance schedules. Traditionally, maintenance practices have been largely reactive or based on fixed schedules, leading to inefficiencies, increased downtime, and higher maintenance costs.

In industrial settings such as aerospace, automotive, and manufacturing, where downtime can have significant financial implications, predictive maintenance models play a pivotal role in ensuring operational continuity and maximizing asset utilization. These models leverage various machine learning techniques, ranging from traditional statistical methods to sophisticated deep learning architectures, to analyze complex patterns in sensor data and predict potential failures before they occur.

Maintenance policies play a crucial role in ensuring the reliability and efficiency of industrial equipment. In the literature, four main categories of maintenance policies have been identified: run-to-failure (R2F), preventive maintenance (PM), condition-based maintenance (CBM), and predictive maintenance (PdM) [1]. R2F involves operating equipment until failure occurs, making it the simplest but costliest approach. PM involves periodic maintenance based on a schedule, while CBM monitors equipment conditions and triggers maintenance based on specific degradation indicators. PdM, also known as statistical-based maintenance, utilizes predictive analytics to determine maintenance actions, performing maintenance only when necessary. Notably, CBM and PdM are often treated as interchangeable in literature [1], [2].

Various modeling approaches have been explored for maintenance policy analysis and reliability engineering. These include Markov chains, Petri nets, fault tree analysis, and the analytic hierarchy process [3]. Additionally, quantitative methods such as heuristic methods, simulation techniques, and analytical methods have been proposed.

Advancements in data analytics and machine learning have revolutionized the field of predictive maintenance. By harnessing the power of historical data, sensor measurements, and advanced algorithms, organizations can now forecast the remaining useful life (RUL) of critical components with unprecedented accuracy. This proactive approach enables timely interventions, allowing maintenance activities to be scheduled precisely when needed, thus minimizing downtime and reducing operational costs. Machine learning-based approaches have gained prominence in predictive maintenance due to their ability to handle high-dimensional datasets effectively. Supervised approaches leverage failure information in the dataset, while unsupervised approaches rely solely on process information [4].

While machine learning has been extensively applied in fault diagnosis, predicting the remaining useful life (RUL) of a machine presents unique challenges. RUL serves as a risk indicator for preventive maintenance, indicating the time a machine can operate before failure. However, acquiring run-to-failure datasets for RUL modeling is challenging.

The availability of turbo engine run-to-failure datasets provided by NASA has spurred research in RUL prediction. Various methods, including machine learning and statistical approaches, have been explored for RUL prediction using these datasets [5].

Kang et al. [13] proposed a Multi-Layer Perceptron (MLP) for automating the prediction of equipment failure in continuous production lines, with a focus on predicting the remaining useful life (RUL) of components using the NASA turbo engine datasets.

Recent advancements in deep learning have led to the development of sophisticated RUL prediction models. Techniques such as multi-scale deep convolutional neural networks (MS-DCNN), recurrent convolutional neural networks

(RCNN), and generative adversarial networks (GAN) have shown promise in improving prediction accuracy [6], [7], [8], [9], [10], [11], [12].

In this study, we delve into the performance evaluation of deep and reinforcement learning models: Reinforcement Learning (RL), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), for RUL prediction. By conducting a comparative analysis of these models using NASA's turbofan engine degradation dataset, we aim to provide insights into their predictive capabilities and identify the most effective approach for predictive maintenance in industrial applications.

III. PROBLEM STATEMENT

Predictive maintenance is essential for preventing unexpected equipment failures and minimizing downtime in industrial settings. However, traditional maintenance strategies often rely on fixed schedules or reactive approaches, leading to inefficiencies and increased costs. The challenge lies in developing accurate models that can forecast the remaining useful life of components based on sensor data and historical maintenance records. These models must account for the complex relationships between sensor readings, maintenance actions, and component degradation over time.

IV. METHODOLOGY

Three neural network architectures, namely Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN), are implemented and evaluated for their predictive performance in estimating the remaining useful life (RUL) of turbofan engines. The dataset is preprocessed to handle missing values, scale the features, and engineer relevant features. The neural network models are trained using the preprocessed data and optimized using suitable hyperparameters.

Data Collection and Preprocessing Historical data regarding the operation of jet engines is collected, comprising operational settings, sensor measurements, and corresponding remaining useful life (RUL) information.

Data Source: The dataset used in this study is sourced from NASA's turbofan engine degradation dataset, comprising multivariate time series data from a fleet of engines. The dataset comprises 100 training trajectories and 100 test trajectories. These trajectories represent a scenario where engines operate under a single condition (Sea Level) and experience one fault mode (HPC Degradation). Each trajectory represents a multivariate time series capturing operational settings, sensor measurements, and the progression of engine degradation. The goal of the analysis is to predict the remaining operational cycles before failure (Remaining Useful Life, RUL) for engines in the test set.

Features Description for Data Set

1) **Unit number**: Identification number of the engine.
2) **Time (cycles)**: Number of cycles elapsed.
3) **Operational setting 1**: First operational setting.
4) **Operational setting 2**: Second operational setting.
5) **Operational setting 3**: Third operational setting.
6) **Sensor measurement 1** to **Sensor measurement 26**: Various sensor measurements capturing different aspects of engine performance and health. These measurements are taken during each operational cycle and provide insights into the condition of the engine.

Preprocessing:

Dropping irrelevant columns such as sensor-measurement22' and 'sensor-measurement23'. Removing constant columns like 'op-setting-3' and specific sensor measurements.

RUL Calculation: the RUL for both testing and training data is calculated and the results are appended as a column labeled 'RUL' in the dataset.

For CNN and LSTM models the dataset with appended 'RUL' is loaded. Data preprocessing involves scaling the input features using MinMaxScaler to ensure they fall within a certain range (0 to 1).

A. Convolutional Neural Network (CNN) Approach

A Convolutional Neural Network (CNN) is a type of deep learning model commonly used for analyzing visual imagery. It has also been adapted to sequential data. It is useful in extracting features and patterns from sensor data related to engine maintenance and performance. By leveraging convolutional layers, CNNs can learn hierarchical representations of the data, aiding in predicting the Remaining Useful Life (RUL) of turbo engines.

Model Architecture:

The CNN model shown in figure 1 is constructed using Keras Sequential API. The model consists of two Conv1D layers with 64 filters each and a kernel size of 3, followed by MaxPooling1D layers to reduce the spatial dimensions. After flattening the output, a dense layer with 100 units and ReLU activation is added, followed by a dropout layer to prevent overfitting. Finally, a dense layer with a single unit and linear activation is used as the output layer.

Compilation:

The model is compiled using the Adam optimizer and Mean Squared Error (MSE) as the loss function. Data Preparation for CNN:

The training data is prepared for CNN by creating sequences of length 24 (sequence-length) from the input data. Each sequence represents a window of time steps, and the RUL (remaining useful life) at the last time step of each sequence is used as the target variable. Training:

The CNN model is trained using the prepared training data with 50 epochs and a batch size of 32. Evaluation:

The trained model is evaluated using the prepared test data. Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) are computed to assess the performance of the model on the test data.

B. Long Short-Term Memory (LSTM) Approach

A Long Short-Term Memory (LSTM) network is a type of recurrent neural network (RNN) architecture capable of learning long-term dependencies in sequential data. LSTM networks can be used to analyze the temporal aspect of

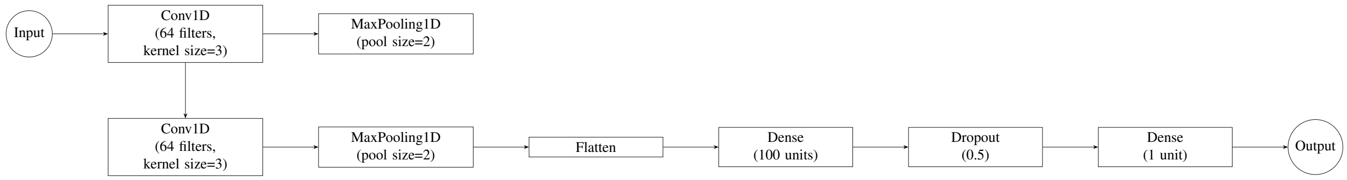

Fig. 1. CNN Network Architecture

engine data, enabling the Remaining Useful Life (RUL) prediction. They are suitable for processing sequential data, such as historical data related to engine operations, allowing the model to capture patterns and trends over time. LSTM networks are well-suited for time series forecasting tasks because they can retain information over extended periods. Utilizing LSTM networks, the model can effectively predict the lifespan of turbo engines based on the available data.

Model Architecture:

The LSTM neural network model, as shown in Figure 2, is constructed using the Keras Sequential API. The model comprises an LSTM layer with 100 units, followed by a dropout layer to prevent overfitting. Subsequently, a dense layer with 50 units and ReLU activation is added, followed by another dropout layer. Finally, a dense layer with a single unit and linear activation serves as the output layer.

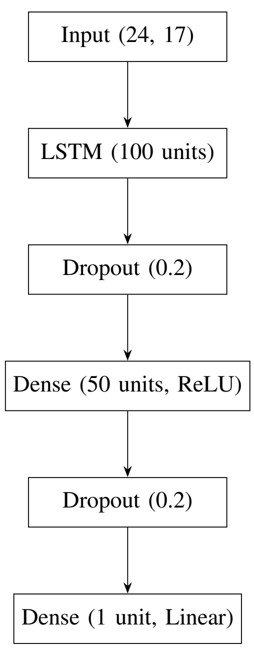

Fig. 2. LSTM Network Architecture

Compilation: The model is compiled using the Adam optimizer and Mean Squared Error (MSE) as the loss function.

Data Preparation for LSTM: The training data is prepared for LSTM by creating sequences of length 24 from the input data. Each sequence represents a window of 24 time steps, and the RUL (remaining useful life) at the last time step of each sequence is used as the target variable.

Training: The LSTM model is trained using the prepared training data with 50 epochs and a batch size of 32.

Evaluation: The trained model is evaluated using the prepared test data. Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) are computed to assess the performance of the model on the test data.

C. Reinforcement Learning (RL) Approach

RL is a subfield of machine learning where an agent learns to make sequential decisions by interacting with an environment. The agent learns to maximize cumulative rewards over time through a trial-and-error process. RL problems are typically modeled as Markov Decision Processes (MDPs), characterized by states, actions, transition probabilities, and rewards.

RL-Based RUL Predictor Architecture:

The RL-based RUL Predictor architecture, depicted in Figure 3, comprises two essential components: the QLearningRULPredictor and the RULPredictorRL. The QLearningRULPredictor implements the Q-learning algorithm, utilizing a Q-table to facilitate action selection based on an exploration-exploitation strategy. Meanwhile, the RULPredictorRL serves as the predictive maintenance environment, emulating engine operations for both training and testing the RL agent's performance. Throughout the training phase, the RL agent interacts with this environment, iteratively selecting actions and updating Q-values to discern an optimal policy for predicting the Remaining Useful Life (RUL) of engines. This framework empowers the agent to make judicious maintenance decisions, effectively balancing exploration and exploitation to optimize long-term rewards.

Q-Learning Algorithm The Q-learning algorithm is employed to train the RL agent. By maintaining a Q-table, each entry denotes the expected cumulative reward for a specific action taken from a particular state. This iterative learning process encompasses exploration and exploitation, wherein the agent updates Q-values based on observed rewards and state transitions, thereby refining its decision-making capabilities.

Predictive Maintenance Environment The predictive maintenance environment mirrors real-world engine operations, encompassing states that encapsulate various engine conditions such as operational settings, sensor measurements, and the current time cycle. The action space is delineated into two distinct actions: performing maintenance or continuing operation without intervention.

State Representation: The state representation includes operational settings, sensor measurements, and the current time cycle, forming the state space.

Action Space: The action space comprises two actions:

Input (State)

↓

QLearningRULPredictor

↓

Step

↓

Output (Action)

Fig. 3. Architecture of the RL-based RUL Predictor

performing maintenance or continuing operation without maintenance.

Reward Scheme: Maintenance actions are incentivized through a reward scheme predicated on the inverse relationship with the remaining useful life. By rewarding maintenance actions when the remaining useful life is diminished, the agent is steered towards prioritizing interventions that maximize the engines' operational longevity effectively.

Training Phases: During the training phase, the RL agent engages with the environment, dynamically selecting actions based on an exploration-exploitation strategy. Episodes culminate either when a pre-defined number of time steps is reached or when specific termination conditions are met, such as the completion of all engine cycles or reaching a maximum number of episodes.

Testing Phase After the training phase, the trained RL agent's efficacy is evaluated using a distinct set of test data. Here, the agent applies its learned policy to forecast the Remaining Useful Life (RUL) of engines, with predictive accuracy gauged by the aggregated rewards amassed during testing.

Evaluation Phase The predictive model's efficacy is scrutinized based on the total reward accrued during the testing phase. Higher cumulative rewards denote superior predictive performance, indicative of the agent's adeptness in managing maintenance decisions to optimize the engines' remaining useful life.

V. RESULTS

In this section, we compare the performance of two different neural network architectures: Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), in estimating the remaining useful life (RUL) of turbofan engines. We also present the performance of the Reinforcement Learning (RL) model in predictive maintenance of the turbo engine.

Convolutional Neural Network (CNN)

The CNN model achieved an MSE of 0.0001539, an MAE of 0.01136, and an R2 value of 0.9935 as shown in Table I. While the CNN model's performance is slightly lower than the LSTM model, it still demonstrates strong predictive capabilities. CNNs are well-suited for processing

TABLE I

PERFORMANCE COMPARISON OF CNN AND LSTM MODELS

Model	**MSE**	**MAE**	**R2**
CNN	0.0001539	0.01136	0.9935
LSTM	0.0000124	0.0027	0.9995

spatial data, making them effective in capturing patterns in sensor measurements. Additionally, CNNs offer relatively simpler architectures compared to LSTMs, making them easier to implement and interpret. The model achieves a relatively low MSE of 0.00015386, indicating good predictive performance. The MAE is 0.01136, which suggests that, on average, the model's predictions are close to the actual RUL values. The high R-squared value of 0.9935 indicates that the model explains a large proportion of the variance in the RUL values. Overall, the CNN-based approach demonstrates strong performance in predicting the remaining useful life of components based on the provided dataset.

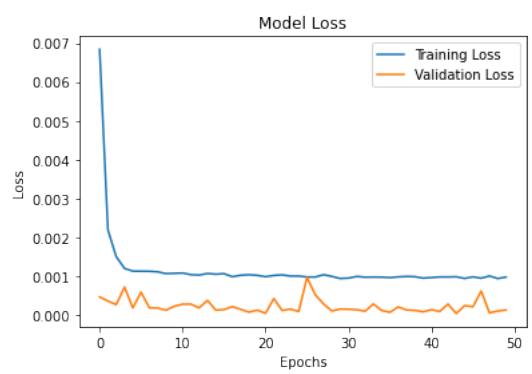

Fig. 4. Evolution of Training and Validation Losses for the CNN.

Long Short-Term Memory (LSTM)

The LSTM model achieved an MSE of 0.0000124, an MAE of 0.0027, and an R2 value of 0.9995. The LSTM model demonstrates high accuracy in predicting RUL see Table I. LSTMs excel in capturing temporal dependencies in sequential data, making them well-suited for time-series prediction tasks like RUL estimation. Additionally, LSTMs offer better interpretability compared to DRL, as their internal state dynamics can be analyzed to understand the model's decision-making process. The model achieves a very low MSE of 1.2401e-05, indicating good predictive performance. The MAE is 0.0027, which suggests that, on average, the model's predictions are close to the actual RUL values. The high R-squared value of 0.9995 indicates that the model explains a large proportion of the variance in the RUL values. Overall, the LSTM-based approach demonstrates strong performance in predicting the remaining useful life of components based on the provided dataset.

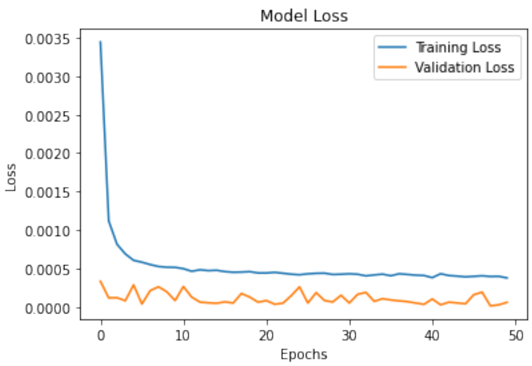

Fig. 5. Evolution of Training and Validation Losses for the LSTM Model.

Reinforcement Learning (RL)

The RL model achieved a total reward of 2.75 during the training process with 1000 episodes 6. The total accumulated reward increased to 11.02 when the model was trained with 4000 episodes 7. While this metric indicates the model's performance, it's essential to interpret it within the context of the specific reinforcement learning environment and reward function used.

The total reward metric reflects how well the RL agent performed in maximizing its cumulative reward while interacting with the environment. In this context the reward is inversely proportional to the RUL, at each time step the agent needs to select an action based on the current state of the engine. The efficiency of maintenance actions taken based on the observations, or the overall cost savings achieved by optimizing maintenance schedules is indicated by the total cumulative reward at the end of the process.

Discussion CNN and LSTM models demonstrate strong predictive performance, with the LSTM model outperforming the CNN model in terms of MSE and MAE. However, both models achieve high R-squared values, indicating their ability to explain a large proportion of the variance in the remaining useful life (RUL) values.

The CNN model's performance is slightly lower than the LSTM model, but it still demonstrates strong predictive capabilities. CNNs are well-suited for processing spatial data, making them effective in capturing patterns in sensor measurements. Additionally, CNNs offer relatively simpler architectures compared to LSTMs, making them easier to implement and interpret.

On the other hand, LSTMs excel in capturing temporal dependencies in sequential data, making them well-suited for

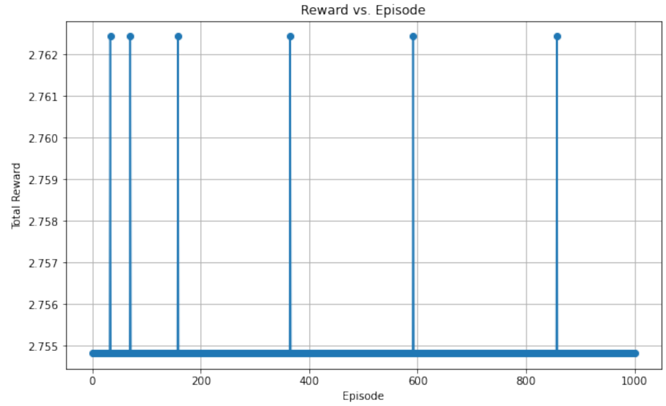

Fig. 6. Relationship between Episodes and Total Rewards during Training.

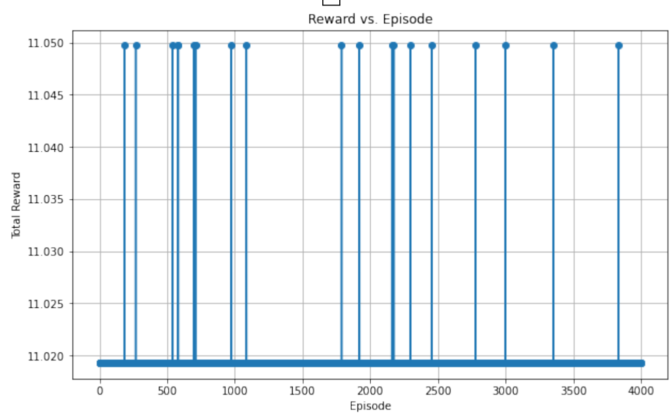

Fig. 7. Relationship between Episodes and Total Rewards during Training.

time-series prediction tasks like RUL estimation. Additionally, LSTMs offer better interpretability compared to CNNs, as their internal state dynamics can be analyzed to understand the model's decision-making process.

A. Training and Validation Loss

Figures 4 and 5 show the training and validation loss for the CNN and LSTM models respectively. The patterns observed in the plots are discussed below.

Steep Decline in Training Loss: The initial steep decline in the training loss indicates that both the CNN and LSTM models are effectively learning from the training data. During the early epochs, the models rapidly adjust their parameters to minimize the discrepancy between predicted and actual values. This phase is characterized by significant improvements in the model's ability to capture patterns and features in the data.

Slow Decrease Approaching Convergence: As training progresses, the rate of decrease in the training loss slows down for both models. This phenomenon is expected as the models fine-tune their parameters to further optimize their performance. Approaching convergence, the models reach a point where further adjustments lead to diminishing returns in terms of reducing the training loss. This phase signifies that the models have learned the underlying patterns in the data to a large extent.

Validation Loss Below Training Loss Curve: One interesting observation is that the validation loss consistently starts below the curve of the training loss for both models. This suggests that, initially, the models generalize well to unseen data, as indicated by the lower loss on the validation dataset compared to the training dataset. This behavior demonstrates the models' ability to capture meaningful patterns that are consistent across different datasets.

Validation Loss Fluctuations: Throughout the training process, the validation loss fluctuates around a certain level, without showing a significant decreasing trend. These fluctuations may arise due to various factors such as the inherent noise in the validation data, model complexity, and hyperparameter settings. While fluctuations in validation loss are common, it's essential to monitor them to ensure that the model's performance is stable and not deteriorating.

Generalization: The consistent gap between the training and validation loss indicates that the models are generalizing reasonably well to unseen data. If the validation loss starts to increase while the training loss continues to decrease, it suggests that the models may be memorizing noise in the training data rather than learning meaningful patterns. Regularization techniques such as dropout and early stopping can help mitigate overfitting and improve generalization performance.

B. RL-Based RUL Predictor Reward Versus Episode Graph

The plot of the rewards per episode in figures 6 and 7 illustrate the progression of total rewards accumulated by the reinforcement learning agent over successive episodes of training. The blue lines represents the total reward obtained by the agent at each episode The plot demonstrates the agent's learning dynamics and its ability to improve performance over time.

The total number of episodes used to train the predictor positively affects the total accumulated reward. The plot shows a horizontal line where the total reward is 2.75 for most episodes with a few spikes to when training is done in 1000 episodes. A similar pattern is observed when the agent is trained in 4000 episodes, the accumulated reward at the end of most episodes is 11.02 with spikes to 11.05. The total reward accumulated by the agent is around the same value for most episodes except for a few. This suggests that as the agent does not gain more experience.

However, achieving a perfect linear relationship where x=y may not always be feasible or desirable in practice. In most real-world scenarios, the relationship between episodes and rewards is more complex, influenced by factors such as the learning dynamics of the environment, the effectiveness of the agent's policy, and the task's difficulty.

Therefore, while observing a line where x=y indicates steady improvement in the agent's performance, it's essential to consider other factors such as the convergence behavior, the learned policy's stability, and the trained agent's overall effectiveness in achieving the desired objectives.

1) Limitations with the RL-Based RUL Predictor: Translating the predictive maintenance problem into a reinforcement learning environment is challenging. For this project, only two actions were selected and the state was set to the engine settings from the dataset. This is simple and may not necessarily capture the complex relationship between engine data and the RUL.

Interpreting the total reward requires a deeper understanding of the reinforcement learning framework, including the environment, state space, action space, and reward structure. While a higher total reward generally indicates better performance, it's crucial to validate the model's predictions using standard evaluation metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) value.

VI. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In addition to the ongoing efforts to enhance predictive models for turbofan engine maintenance, there's potential for extending this study's applicability to predicting the Remaining Useful Life (RUL) of oil wells. This refers to how long the well will take before seizing production given its current state.

Exploring ensemble learning techniques, feature selection methods, and advanced deep learning architectures could further refine the predictive performance of models for oil well maintenance prediction. Moreover, assessing the scalability of these models to large-scale industrial datasets and their deployment in real-world scenarios is crucial.

Future work could involve adapting and evaluating the developed models for predicting the RUL of oil wells. Comparing the predictions of the RL model with actual RUL values and employing standard evaluation metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) value can help assess its accuracy in this new domain. Additionally, exploring different reward functions and hyperparameter settings could optimize the performance of these models specifically for oil well maintenance prediction.

While Reinforcement Learning (RL) holds promise in predictive maintenance applications, its effectiveness and scalability depend on various factors. Further research and experimentation are necessary to fully leverage the potential of RL in estimating the Remaining Useful Life of oil wells, contributing to advancements in both the oil and gas industry and predictive maintenance methodologies.

REFERENCES

[1] Susto G.A., Beghi A., De Luca C. A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. IEEE Trans. Semicond. Manuf. 2012;25:638–649. doi: 10.1109/TSM.2012.2209131.

[2] Mobley R.K. An Introduction to Predictive Maintenance. Elsevier; Amsterdam, The Netherlands: 2002.

[3] Angius A., Colledani M., Yemane A. Impact of condition based maintenance policies on the service level of multi-stage manufacturing systems. Control. Eng. Pract. 2018;76:65–78. doi: 10.1016/j.conengprac.2018.04.011.

[4] Susto G.A., Schirru A., Pampuri S., McLoone S., Beghi A. Machine learning for predictive maintenance: A multiple classifier approach. IEEE Trans. Ind. Inform. 2014;11:812–820. doi: 10.1109/TII.2014.2349359.

[5] Ramasso E., Saxena A. Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. Int. J. Progn. Health Manag. 2014;5:1–15.

[6] Li H., Zhao W., Zhang Y., Zio E. Remaining useful life prediction using multi-scale deep convolutional neural network. Appl. Soft Comput. 2020;89:106113. doi: 10.1016/j.asoc.2020.106113.

[7] Hou M., Pi D., Li B. Similarity-Based Deep Learning Approach for Remaining Useful Life Prediction. Measurement. 2020;159:107788. doi: 10.1016/j.measurement.2020.107788.

[8] Cheng H., Kong X., Chen G., Wang Q., Wang R. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. Measurement. 2020;168:108286. doi: 10.1016/j.measurement.2020.108286.

[9] Wang B., Lei Y., Yan T., Li N., Guo L. Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. Neurocomputing. 2020;379:117–129. doi: 10.1016/j.neucom.2019.10.064.

[10] Chen Y., Peng G., Zhu Z., Li S. A novel deep learning method based on attention mechanism for bearing remaining useful life prediction. Appl. Soft Comput. 2020;86:105919. doi: 10.1016/j.asoc.2019.105919.

[11] Wu J., Hu K., Cheng Y., Zhu H., Shao X., Wang Y. Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. ISA Trans. 2020;97:241–250. doi: 10.1016/j.isatra.2019.07.004.

[12] Li X., Zhang W., Ma H., Luo Z., Li X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. Knowl. Based Syst. 2020;197:105843. doi: 10.1016/j.knosys.2020.105843.

[13] Kang, Ziqiu, Cagatay Catal, and Bedir Tekinerdogan. "Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks." Sensors 21.3 (2021): 932.