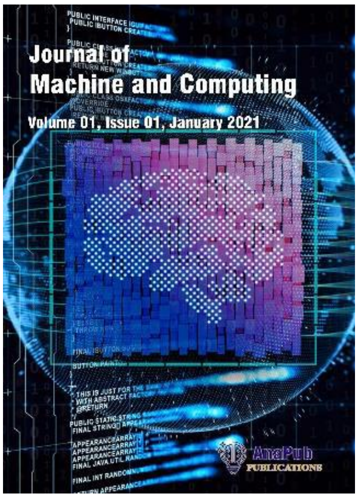
# Journal Pre-proof

Intelligent Diagnostic System for Mechanical Fault Detection Using Deep Learning




**Shaik Jaffar Hussain, Rupa Devi B, Anant Shankar E, Lakshmi H N, Rangaswamy K and Suneelgoutham Karudumpa**

DOI: 10.53759/7669/jmc202505065

Reference: JMC202505065

Journal: Journal of Machine and Computing.

Received 02 April 2024

Revised form 28 December 2024

Accepted 16 February 2025

# Intelligent Diagnostic System for Mechanical Fault Detection Using Deep Learning

**[1]Shaik Jaffar Hussain, [2]B. Rupa Devi, [3]E Anant Shankar, [4]Lakshmi H N, [5]K Rangaswamy, [6]Suneelgoutham Karudumpa**

[1] Associate Professor, Department of CSE, Sri Venkateswara Institute of Science and Technology, Kadapa.
[2] Associate Professor, Department of CSE, Annamacharya Institute of Technology and Sciences, Tirupati.
[3]Assistant Professor, Department of ECE, Sri Venkateswara College of Engineering (Autonomous), Tirupati, Andhra Pradesh, India
[4] Professor, Department of Computer Science and Engineering (AI&ML), CVR College of Engineering, Hyderabad.
[5]Associate Professor, Department of CSE(DS), Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal, Andhra Pradesh.
[6]Assistant Professor, Department of Electrical and Electronics Engineering, Aditya Institute of Technology and Management, Tekkali, Srikakulam.
[1]jaffar.thebest@gmail.com, [2]rupadevi.aitt@annamacharyagroup.org, [3]anant[illegible]ankar16[illegible]gmail.com
[4] hn.lakshmi@cvr.ac.in, [5]rangaswamy19@gmail.com, [6]goutham.sune[illegible]gmail.com

**Abstract -** Advanced diagnostic tools are essential for aerospace transportation systems and automotive industries and industrial manufacturing facilities since operational efficiency requirements and safety needs demand failure prediction tools. Systems that use traditional diagnostic methods depend on centralized architectures that show limitations regarding scalability while being unable to overcome subsystem failure events effectively. The research presents Gossip Neural Network (GNN) as a decentralized deep learning (DL) system which determines Remaining Useful Life (RUL) duration in distributed mechanical engine systems. The GNN combines Convolutional Neural Networks (CNNs) and Long short-term Memory (LSTM) network layers to identify short-term sensor anomalies in addition to capturing long-term sensor degeneration patterns in sensor data. A gossip-based protocol allows the GNN to facilitate distributed engine subsystems which train a shared model together through peer-to-peer collaborations without needing central control. The assessment of the proposed framework using CMAPSS data proves its exceptional capability for RUL prediction alongside reliable accuracy and low error rates. The GNN demonstrated excellence in different datasets through $R^2$ results between 92.43% and 94.57% and RMSE results within 12.77 to 12.87 which demonstrates its effectiveness in handling realistic operational environments. The GNN provides an encouraging solution for time-sensitive fault detection in distributed systems which facilitates efficient predictive maintenance across large engineering applications.

**Keywords** - GNN, RUL Prediction, Decentralized Deep Learning, Predictive Maintenance, Fault Detection, Mechanical Fault Detection.

## I. INTRODUCTION

In modern engineering systems, the critical domains of aerospace, automotive, and industrial manufacturing, and particularly the relentless demand for operational efficiency and safety, now make it imperative to have advanced diagnostic tools for prompting mechanical failures [1]. These systems are complex and engines found at the heart of such systems are prone to progressive deterioration of life which manifests through events like bearing wear, rotor imbalance lubrication failure and catastrophic failure is possible if these progressive degradation of life phenomena are not addressed proactively [2]. The idea of RUL prediction has grown in importance as a foundation for predictive maintenance [3], and is best at predicting when an engine is going to fail and therefore a failure threshold is going to be reached before that time and guiding interventions to avoid downtime, reduce maintenance costs and increase system reliability [4]. However, conventional sensor diagnostic approaches frequently rely on centralized architectures that collect sensor data from distributed sub-systems into one processing center, which is physically limited in scale, vulnerable to data privacy, and failure-resilient to subsystems [5]. Additionally, the amount of multivariate time series—vibration, temperature, pressure, and rotation speed, amongst others that must be modeled is very complex and requires very sophisticated modeling that can capture both short-term anomalies and long-term degradation trends (Implementation of a sequence-to-sequence stacked sparse LSTM autoencoder for anomaly detection on multivariate timeseries data of industrial blower ball bearing unit).

DL-based fault detection and prognostics have gained significant popularity in recent years because of the incredible tools it provides for encoding complex patterns in the high dimensional sensor data, which procure higher accuracies than conventional statistical and physics-based approaches and are more capable of handling varying conditions [6]. CNNs are highly effective in extracting spatial-temporal features [7], whereas LSTM networks [8] are strong in sequential dependencies that match the capacity of the RUL prediction [9]. However, DL has been underexplored and applied to distributed engine systems, particularly in cases where distributed data processing is not possible for bandwidth constraints and privacy reasons or due to the enormous scale of the networked subsystems [10]. In order to address these limitations, the GNN, a decentralized DL framework consisting of CNN and LSTM layers connected in a network of distributed nodes (i.e., each of them serves as a sensor-equipped engine subsystem) trained collaboratively using a gossip-based protocol, is proposed. The GNN forages around a decentralized network of engines by eschewing a central server to perform local computation and pairwise weight exchange and to achieve a unified RUL predictor in a scalable, robust RUL predictor and privacy to DL bright.

This research has great significance because it could lead to making a leap from predictive maintenance in distributed mechanical systems to one where diagnostics need to be made online, in real-time, while also preserving patient privacy. Moreover, the proposed GNN not only resolves the technical challenges of decentralized learning but also fulfills the practical requirements of today's engineering applications typical of modern engineering applications, e.g., fleet-wide aircraft engine monitoring or industrial turbine networks with subsystems independently but interdependently working. The key contributions of this research are:

1. We introduce the GNN, a novel decentralized DL framework for predicting RUL in engines without a central server, and evaluate it on power exhaust.
2. We presented a two-stage gossip-based protocol that integrates CNN and LSTM layers to both model short-term anomalies and long-term degradation patterns in multivariate sensor data.
3. We allowed distributed engine subsystems to collaborate, building for large-scale networks, as an enhancement of scalability and fault tolerance.
4. We also ensured node-to-node communication if sensor data is kept local while model weights are shared among nodes, thus ensuring data privacy in diagnostic systems.
5. We showed how the GNN can be used to solve mechanical engineering problems by a method of both local computation and decentralized aggregation toward intelligent fault detection.

In the rest of the paper, related work of fault detection and decentralized learning is reviewed in Section 2, Section 3 outlines the method, which includes data collection, GNN architecture, training, and evaluation, and Section 4 reports results and analysis, and finally, Section 5 concludes with implications and future directions. The work provides a solid base for intelligent diagnostic systems capable of enhancing reliability and increasing the lifetime of critical mechanical assets.

## 2. LITERATURE REVIEW

In recent times the prediction of RUL in aircraft engines has become a prominent research focus because researchers use different machine learning (ML) and DL strategies to achieve better results along with increased interpretability and improved computational performance. Research investigates existing studies through three identified subcategories: Traditional ML Approaches, DL-Based Approaches and Hybrid and Attention-Based Models. Researchers employ comparative analysis to identify both the advantages and drawbacks and relevant contributions of each study category.

### 1. Traditional ML Approaches

ML traditional methods find broad application in RUL prediction because of their readable algorithms together with their computational integrity. Alomari et al. [11] developed a new framework which combines feature engineering with dimensionality reduction and feature selection methods including PCA and Genetic Algorithm and LASSO. The Aggregated Feature Importances with Cross-Validation (AFICv) method utilized by their team for C-MAPSS FD001 data achieved [illegible] $R^2$ accuracy metrics which validates the effectiveness of feature reduction techniques for maintaining prediction accuracy. Deepika et al. [12] performed an analysis of the XGBoost, Random Forest, and SVM traditional ML models while XGBoost delivered an RMSE of 23.8 and an $R^2$ score of 0.67. Research findings show that traditional ML models demonstrate strong potential for RUL estimation when such models use advanced feature selection practices with traditional ML models. The analysis methods encounter two primary barriers that stem from informational data reduction techniques plus difficulties with dataset variability.

The assessment by Rosero et al. [13] of aircraft cooling units RUL included a combination of physics-based and data-driven models with Health Indicators (HIs) that originated from time-frequency analysis. The researchers detected unique degradation steps through their approach which resulted in a 0.352 RMSE. This technique generates important failure pattern information yet depends on advanced data transformation methods yet only works well with simple failure modes. The main benefit of traditional ML methods includes both understandable results and efficient processing but these advantages do not extend to sophisticated non-linear patterns found in sensor measurements.

### 2. DL-Based Approaches

DL models currently lead the way because they excel at identifying elaborate patterns and continuing relationships within sensor information. Real turbofan engine data yielded superior prediction outcomes for LSTM and CNN models compared to simulated data as noted in Szrama et al. [14]. The models they developed exhibit limitations in training sensitivity which affects their performance when applied to various engine types. Ozkat et al. [15] used LSTM to predict RUL in Unmanned Aircraft Systems based on sensor data resulting in RMSE values of 3.7142 Hz, 1.4831 Hz and 1.3455 Hz. The systematic approach shows effectiveness but its reliability depends on fixed threshold values that can lead to performance issues during varied vibration data assessment.

In contrast, Dangut et al. [16] developed a hybrid DL model called AE–CNN–BGRU to detect rare failures and demonstrated 94% success rate in these detection tasks. The rescaled focal loss function helped address data imbalance while the model continues to depend on high-quality historical logs and could potentially overfit. Wang et al. [17] developed Bi-LSTM-AM through merging 1D CNN with Bi-LSTM for RUL prediction alongside MILP optimization of maintenance planning. Their strategy reduced system downtime yet encountered challenges when scaling up operations and when executing methods under realistic settings. Research confirms that DL models excel in analyzing complex datasets yet these analyses mention their computational requirements as well as their vulnerability to hyperparameter settings.

### 3. Hybrid and Attention-Based Models

Predicting RUL requires hybrid and attention-based models which unite various architectural elements to create more accurate and explainable forecasting systems. The combination of PSA and GHLSTM in Transformer modeling led Chen [18] to achieve premier results on C-MAPSS data with an FD001 RMSE of 13.14 ± 0.21. Lin et al. [19] designed CATA-TCN which merged temporal and channel attention components to enhance prediction accuracy dealing with challenging operational conditions. The proposed model demonstrated better performance than previous approaches yet its processing efficiency and capacity to handle impaired data represented obstacles for implementation.

Jiangyan et al. [20] created the Two-Stream Convolution Augmented Transformer (TACT) model that delivered better results than other advanced models by showing Score reduction of 2.71% and RMSE decrease of 3.13% against current top models. A major operational challenge for this computational approach exists due to its slow processing speed. Deng & Zhou [21] developed a CNN-LSTM-Attention model to analyze FD001-FD004 datasets which produced RMSE results of 15.977, 14.452, 13.907 and 16.63[illegible] respectively. The model provides accurate results but needs additional improvements to achieve real-time deployment capability. The iSTLSTM model by Gao et al. [22] uses specific spatiotemporal information along with hybrid attention systems to perform explainable RUL predictions. Performance effectiveness of this method depends on both the sensor data quality and operating environmental conditions.

The developers at Zhang et al. [23] developed ATCN which combines self-attention with TCN alongside squeeze-and-excitation mechanisms for better feature extraction and predictive performance. The proposed method delivers higher results than competing CNN, LSTM and Transformer-based models yet depends on the quality and consistency level of incoming data. The researchers from Gan et al. [24] presented DMHA-ATCN which utilizes dual-dimensional attention to weight features based on space and time to demonstrate enhanced prediction quality and easier interpretation. This system needs additional study regarding its operational flexibility within real-time domains alongside its performance across various engine types.

Huang et al. [25] developed ASD-YOLO as a network solution for aircraft surface defect detection through the implementation of deformable convolution along with attention mechanisms. Although their model delivered 5.7% and 3.4% mAP enhancement across two datasets the system needs optimization to handle real-world conditions and various defect types. Hybrid and attention-based predictive models function as the frontiers of RUL forecasting technology because they deliver substantial improvements in both accuracy rates and interpretation capabilities. Real-world implementation faces difficulties because of these methods' high complexity together with increased computational requirements.

The review of various RUL prediction methods appears in Table 1 with findings and results and identified limitations.

Table 1: Comprehensive Literature Review on RUL Prediction Models

| Reference | Method/Model | Findings | Limitations | Result | Year |
|---|---|---|---|---|---|
| Alomari et al. [11] | Feature engineering, PCA, Genetic Algorithm, RFE, LASSO, Random Forest, AFICv, Gradient Boosting, Random Forest, MLP | AFICv efficiently reduced features while maintaining prediction accuracy across C-MAPSS sub-datasets | Potential information loss, dataset dependency, no DL models tested | 0.91 $R^2$ score in FD001 using AFICv | 2023 |
| Rosero et al. [13] | Hilbert spectrum for time-frequency HIs, physics-based + data-driven ML models | Cooling units follow normal degradation before an abnormal phase near end of life | Failure patterns may be complex, requiring advanced transformations for detection | 0.352 Std. RMSE in Raw Data | 2022 |
| Ozkat et al. [15] | Mean peak frequency feature, LSTM for RUL prediction | LSTM effectively predicts RUL using vibration data, threshold-based estimation | Variability in vibration data, reliance on predefined thresholds, generalization concerns | RUL estimates: 4s, 10s, 10s; RMSE: 3.7142 Hz, 1.4831 Hz, 1.3455 Hz | 2023 |
| Dangut et al. [16] | AE–CNN–BGRU model | Improved rare failure prediction accuracy | Data quality dependency, overfitting risk, adaptability issues | 94% of extremely rare failure of components and AUC = 0.864 | 2023 |
| Wang et al. [17] | 1D CNN + Bi-LSTM-AM, Bayesian optimization, MILP | Improved RUL prediction and reduced maintenance time | RUL accuracy dependency, scalability, real-world feasibility | Efficient scheduling, minimized maintenance downtime | 2024 |
| Szrama et al. [14] | CNN, LSTM with regression output | Real data improves prediction accuracy over simulated data | Data bias, training sensitivity, generalization issues | Effective RUL estimation using real engine data | 2024 |
| Chen [18] | Transformer with PSA and GHLSTM | Improved RUL prediction over existing models | High computational cost, hyperparameter sensitivity | FD001 Dataset: Score: 220 ± 23, RMSE: 13.14 ± 0.21<br>FD004 Dataset: Score: 1420 ± 125, RMSE: 14.25 ± 0.25 | 2024 |
| Lin et al. [19] | CATA-TCN (Channel Attention & Temporal Attention-based Temporal Convolutional Network) | Improved prediction accuracy, particularly under changeable operational conditions and complex fault modes. Outperformed existing RUL prediction models in both Score and RMSE. | Sensitive to noisy or incomplete sensor data; higher computational complexity due to dual attention mechanism. | Significant improvements in overall RUL prediction | 2024 |
| Jiangyan et al. [20] | Two-Stream Convolution Augmented Transformer (TACT) model combining multi-scale CNN and Transformer modules | The TACT model improves RUL prediction accuracy, reducing Score by 2.71% and RMSE by 3.13% compared to existing methods. | The model's complexity increases computational cost and training time, which may hinder its real-time applicability in practical systems. | Score reduction of 4.54% | 2024 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| Huang et al. [25] | ASD-YOLO network based on YOLOv5 with DCNC3, GAM, CEM, and EMA-Slide for ASD detection. | Incorporates deformable convolution, attention mechanisms, and sample imbalance solutions. | Requires further optimization for real-world scenarios and diverse defect types. | mAP improvement by 5.7% and 3.4% on two datasets compared to mainstream methods. | 2024 |
| Zhang et al. [23] | Attention-based Temporal Convolutional Network (ATCN) with self-attention, TCN, and squeeze-and-excitation mechanisms. | Enhances feature extraction and improves prediction accuracy by weighting contributions from time steps and channels. | Dependent on the quality and consistency of input data, which may affect generalization across varied conditions. | Higher accuracy in RUL prediction | 2024 |
| Gao et al. [22] | iSTLSTM: LSTM with Bi-ConvLSTM1D for feature extraction and a hybrid attention mechanism for interpretability. | Enhances feature extraction through spatio-temporal dependencies and improves model interpretability while maintaining high prediction accuracy. | Performance is affected by the quality and variability of sensor data, which could impact its generalization under diverse operating conditions. | Superior RUL prediction performance | 2024 |
| Deepika et al. [12] | ML-based RUL prediction using models like XGBoost, Random Forest, SVM, KNN, and Linear Regression, with PostgreSQL for data storage and Flask for real-time visualization. | XGBoost achieved the best performance with an RMSE of 23.8 and an R² score of 0.67, nearly matching the accuracy of DL models while being computationally efficient. | While XGBoost performed well, its generalizability to other datasets and real-time application may need further validation, particularly under varying operating conditions. | RMSE of 23.8 and an R² score of 0.67 | 2025 |
| Gan et al. [24] | Dual-dimensional attention mechanism using DMHA for feature weighting and ATCN for adaptive temporal representation learning. | The DMHA-ATCN model outperforms traditional TCNs in RUL prediction, improving interpretability and prediction accuracy. | Further investigation is needed to assess the model's adaptability to real-time systems and its scalability for different types of engines. | Improved prediction accuracy over traditional TCN, with enhanced interpretability through DMHA. | 2024 |
| Deng & Zhou [21] | CNN-LSTM-Attention model for RUL prediction of aircraft engines. | CNN-LSTM-Attention model outperforms CNN and LSTM, improving RUL prediction accuracy across all CMAPSS datasets with the attention mechanism enhancing feature extraction. | Further optimization needed for real-time deployment and to handle varying operational conditions. | RMSE for FD001: 15.977, FD002: 14.452, FD003: 13.907, FD004: 16.637; CNN-LSTM-Attention model | 2024 |

# III. METHODOLOGY

This section describes how the GNN proposed for predicting the RUL of engines within an intelligent diagnostic system is developed and evaluated. The scalability to large engine networks, robustness against node failures, and data privacy preserving are most beneficial when using this approach, solving several deficiencies with centralized diagnostic systems. In the subsequent subsections, we specify a methodology that details each of the stages; including data collection and preprocessing, the design of the GNN architecture, distributed training procedure, implementation specifics, and performance evaluation, all of which are summarized in Figure 1 to provide a complete and reproducible framework.

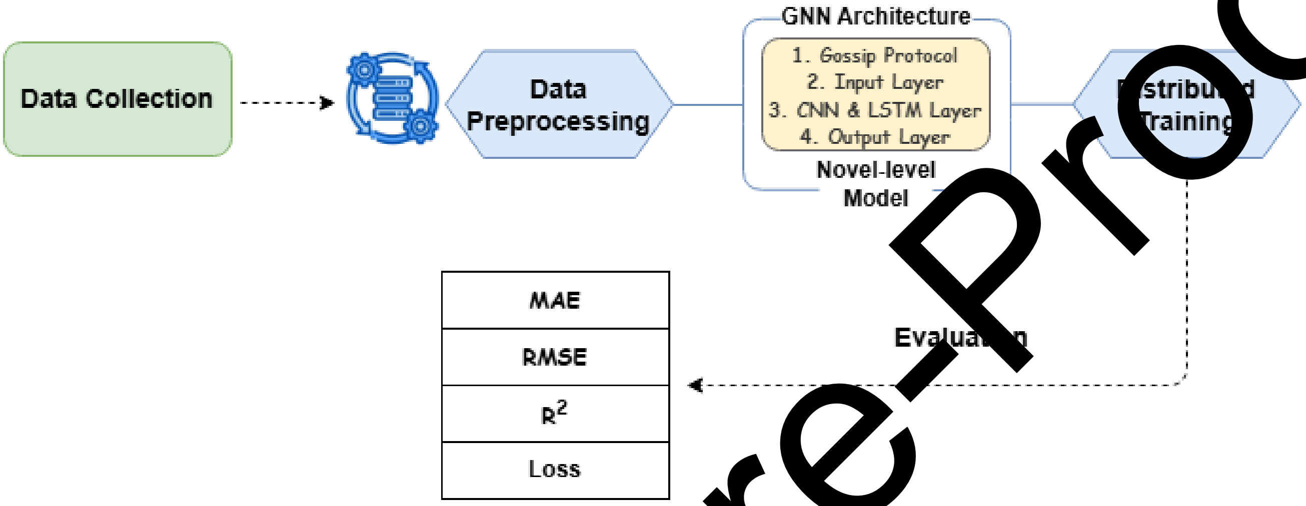



Figure 1: Overview of the research methodology framework

## 3.1 Dataset Description and Preprocessing

This research uses the CMAPSS (Commercial Modular Aero Propulsion System Simulation) dataset which is found in NASA's repository on Kaggle [26]. This data set is commonly used for prognostics and health management (PHM) purposes, specifically in RUL prediction of aircraft engines. Specifically, it is composed of multivariate time series data derived from aircraft engine sensors under varying conditions of operation.

### 3.1.1 Dataset Composition

The dataset is comprised of four sub-datasets (FD001, FD002, FD003, FD004), and four operating conditions and four fault modes are required to be different. For the number of engines and their complexed operating conditions, the datasets are distinctively varied. The whole dataset is summarized in table 2.

Table 2: Summary of CMAPSS Dataset

| Dataset | Engines | Operating Conditions | Fault Modes | Number of Features | Total Observations |
|---|---|---|---|---|---|
| FD001 | 100 | 1 | 1 | 21 | 20,631 |
| FD002 | 259 | 6 | 1 | 21 | 53,076 |
| FD003 | 100 | 1 | 2 | 21 | 24,815 |
| FD004 | 249 | 6 | 2 | 21 | 61,918 |

### 3.1.2 Data Preprocessing

The dataset is of high quality and has no missing values since it came from NASA. But first, a preliminary analysis was made to ensure inconsistencies, duplicates, or corrupted records. It was confirmed that the data set had no duplicate entries which indicated that nonduplicate data was used. Also, there were statistical methods of outlier detection using the Interquartile Range (IQR) and Z score, but no real outlier was found that needed to be removed.

In order to derive meaningful insights from raw sensor data, feature engineering was conducted. The biggest problem in this process was determining the RUL for each of the engines. The RUL was identified by using the following equation since the training data contains full engine life cycles until failure:

$$RUL = max(Cycle) - Cycle$$

where max(Cycle) is the last operational cycle before failure for an engine. By transforming the dataset, we were then able to change the dataset into a supervised learning format in which the input is sensor readings and the output is the predicted RUL.

To normalize the scale of the sensor readings, Min-Max Normalization was applied which transformed the data in the range [0,1]. The normalization allowed all features to contribute equally to the model during training and precluded one feature from dominating another because not all features were given equal weight when magnitudes differed. The following equation was used for the normalization:

$$X' = \frac{X - Xmin}{Xmax - Xmin}$$

where $X$ is the original sensor reading, and $X'$ is the normalized value.

The CMAPSS dataset is in the form of time series data, so it is needed to restructure the dataset into the sequence for modeling with LSTM networks and GNN networks. A sliding window approach was used by implementing each training sample in the format of 30 continuous time steps (cycles) as input, and the final RUL as the target output. The reason for this sequence length was based on empirical studies that show this by capturing temporal degradation trends. Then, the data was turned into a sequence to one by the model where the RUL is predicted with respect to the past sensor readings. The neural networks could learn degradation patterns in time, and make reliable predictions of the time series. Finally, to make sure the evaluation of the model is robust, we split the dataset into three: training set, 70%, to train the DL models, validation set, 15%, for hyperparameters tuning and performance monitoring during training, and test set, 15% for evaluating model performance on unseen data.

### 3.2 Gossip Neural Network Model

The GNN is a decentralized DL framework proposed for the RUL prediction of engines in an intelligent diagnostic system. Unlike traditional centralized methods, this model pushes computation to multiple nodes, such as sensor-equipped edge devices or subsystems, that collaboratively train a shared DL model in a 'decentralized' manner without a central server. The local sensor data is processed by each node and model parameters (e.g. weights) are exchanged among neighboring nodes based on a gossip-based communication protocol. By maintaining such scalability, fault tolerance, and data privacy, this design is ideally suited for mechanical fault detection and prognosis in real time for dynamic distributed engine systems, e.g. aircraft turbines and industrial machinery. In the context of engine degradation driven by mechanical problems such as bearing wear and rotor imbalance, the GNN tackles a critical problem — namely, predicting the remaining operation time (hours or cycles) until an engine fails by using multivariate time series sensor data such as vibration, temperature and pressure to model degradation patterns and deriving accurate RUL estimates for predictive maintenance.

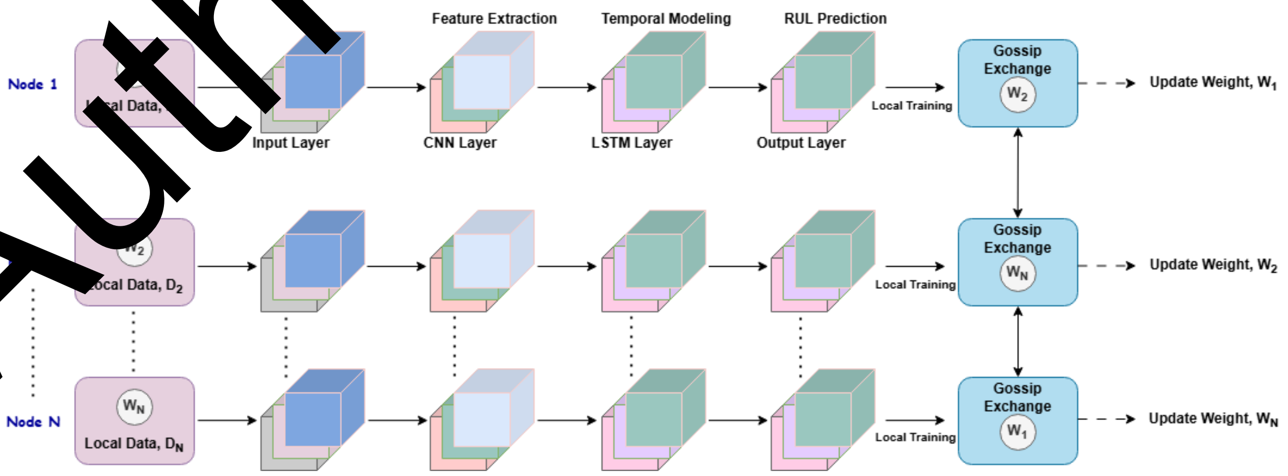



Figure 2: GNN Architecture

The GNN has multiple nodes, (Node 1), (Node 2), ….., (Node N), each of which has an analogous DL model for dealing with time series data. The base network architecture starts with an input layer that takes multivariate sensor data of shape (100, 10), for example, the vibration amplitude, oil pressure, and rotational speed, and feeds that into it for processing. It contains 1D CNN layers followed by LSTM layers. Temporal features are extracted spatial features from sensor signals through the convolution operation of CNN with 32 filters, kernel size of 3, and ReLU activation on non-linearity to reduce dimensionality, preserving the fault-related patterns. These features are then passed through LSTM layers with 64 neurons to model the long-term temporal dependencies of engine degradation occurring over time. The architecture then ends with an output layer consisting of a fully connected layer of a single neuron and a linear activation, predicting RUL in hours or cycles in a continuous way, and making a quantitative prognosis about engine health.

As shown in Figure 2, the distributed training mechanism of the GNN is realized through a decentralized process of the N nodes. The node conducts a local training process over its dataset of time series sensor data that have RUL labels and locally optimized its local weights. This means that backpropagation is used to minimize the MSE loss function using an optimizer such as Adam, where vertical arrows are labeled "Training" on the local data to the weights for each node. Nodes perform local updates and, after local updates are complete, nodes engage in a gossip-based parameter exchange with neighbors pairwise: From (Node 1) to (Node 2) and (Node N), (Node 2) to (Node N-1) and (Node N-1) to (Node N) with a bidirectional arrow labeled "Gossip Exchange" and specifying which weights are transferred every [illegible] local epochs. In addition, each local weight is then aggregated with the local weights received, balancing local and external contributions, through downward arrows onto the updated local weights. Below are the loss and aggregation processes formalized.

**i. Input Layer**

For the GNN, the input at each node is a multivariate time-series sensor data, which allows initial feature extraction and temporal modeling forming the input, but which represents the groundwork for the system. However, this layer accepts data that has a matrix structure of 100 timesteps and 10 features into which the 100 timesteps cover a temporal window in which the engine has been operated, and the 10 features are important variables like vibration amplitude, oil temperature, exhaust pressure, shaft speed, fuel flow, coolant temperature, ambient pressure, torque, power out, and oil debris. Together these features represent a collection of engine operational state, instant and time-varying measurements, which are important to indicate degradation patterns predictive of imminent failure. The first layer of the network (input layer) does not do any transformation, only format properly the data and normalize it, usually divided by a constant to normalize the data to the range of [0, 1] to standardize inputs across nodes so that we can mitigate differences from the viewer or sensor variations or environmental conditions. In this layer, its output is directly passed to the CNN layers as a robust, high-dimensional representation of the engine health to be further processed. The input can mathematically represented as:

$$X_i \in R^{100x10}$$

where $X_i$ is the sensor data matrix for node $i$ with 100 timesteps and 10 features.

**ii. CNN Layer**

The input layer is followed by the CNN layer which uses the multivariate sensor data to extract its spatial and temporal features to be used in identifying local patterns related to mechanical faults by GNN. This 1D CNN layer is configured with 32 filters, kernel size 3, a stride of 1, and ReLU activation, and is run with 32 filters to attempt to detect short-term anomalies (sudden vibration spikes, pressure drops, temperature gradients, etc) that indicate degradation but is relative to the temporal dimension of the input matrix. The 32 filters end up generating a feature map of size 98×32, sacrificing the temporal dimension 100−3+1=98, to allow each filter to learn different patterns and thus enable the model to distinguish fault signatures from noisy sensor data. With ReLU activation function, this nonlinearity provides, that is only positive feature responses are kept while the noise is suppressed and the signal gets amplified. This layer helps reduce the dimension of the raw input by preserving important spatial temporal relationships, resulting in a compact representation but is informative enough for the subsequent LSTM layer to perform temporal analysis. The function of the CNN layer can be mathematically described as:

$$h_t = ReLU(W_{CNN} * X_t + b_{CNN})$$

where $W_{CNN} \in R^{32x3x10}$ is the filter weight tensor (32 filters, kernel size 3, 10 input channels), $*$ denotes 1D convolution, $b_{CNN} \in R^{32}$ is the bias, and $h_t \in R^{98x32}$ is the output feature map.

**iii. LSTM Layer**

After the CNN layer, to model long-term temporal dependencies of the extracted features which is necessary for RUL prediction is the LSTM layer. This layer is configured with 64 units which process the CNN output sequence (reshaped to 98×32 per timestep) with hidden state and cell state that change over 98 timesteps to learn patterns like gradual wear, cumulative stress (or stress accumulation), or recurring anomalies appearing within the input window. Specifically, the

LSTM is composed of forget, input, and output gates and is used selectively to recall or discard information from steps of previous timesteps to prevent vanishing gradient problems common in regular recurrent nets and to focus on important degradation trends as shown in Figure 3.

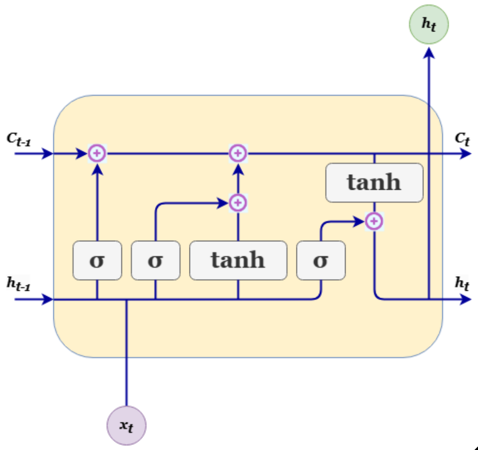



Figure 3: Gating Mechanism of LSTM Networks

The LSTM has 64 nodes and the output produces a hidden state sequence of size 98×64 which is the temporal context that is needed for the estimation of RUL in the final timestep. Since LSTMs have been demonstrated to be effective for engine RUL prediction, namely for tasks where understanding long-term dependencies is crucial, but computational efficiency might be important, the choice of LSTMs is driven, and alternatives like GRUs or TCNs might be considered. The final prediction step needs this layer's output as a rich temporal representation. The update functions of forget, input, and output gate can be expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$c_t = f_t . c_{t-1} + i_t . tanh(W_c \cdot [h_{t-1}, x_t] + b_C)$$
$$h_t = o_t . \tanh(c_t)$$

where $W_f, W_i, W_o, W_c \in R^{64(64+32)}$ are weight matrices (32 from CNN output size), $b_f, b_i, b_o, b_c \in R^{64}$ are biases, $f_t, i_t, o_t$ are gate activations, $c$ is the cell state and $h_t \in R^{64}$ is the hidden state at timestep $t$.

## iv. Output Layer

The node-level architecture ends with the output layer, which turns the LSTM's temporal representation into a single continuous RUL prediction. The last part of the architecture consists of a single fully connected neuron with linear activation, takes the final hidden state from LSTM (size 64 at time t=98), and linearly transforms it to obtain the predicted RUL, $\hat{y}_i$, expressed in hours or cycles remaining until engine failure. This gives the linear activation for the output to ensure unconstraint of the output as the RUL values are continuous (imminent failure to thousands of hours or cycles of the engine condition and operational context). In contrast to the simple structure, this layer serves to consolidate the hierarchical features computed from the CNN, and then model this into a practical prognostic metric that is easily usable for maintenance decision making. During local training, local weights and biases of this layer are tuned to minimize prediction error, such that the output matches the ground truth RUL labels given in the training data. The prediction mechanism of this layer can be defined as:

$$\hat{y}_i = W_{out} . h_{LSTM} + b_{out}$$

where $W_{out} \in R^{1x64}$ is the weight vector, $b_{out} \in R$ is the bias, $h_{LSTM} \in R^{64}$ is the final LSTM hidden state, and $\hat{y}_i$ is the predicted RUL.

## v. Distributed Training and Gossip Mechanism

As shown in Figure 4, their GNN's distributed training mechanism allows for collaborative refinement of the RUL predictor across a network by integrating the node-level DL model into a decentralized gossip Protocol. By each node local training on its private dataset (i.e., local sensor data $X_i$ and local RUL labels $y_i$) with mean square error loss function (MSE) as a cost function using Adam optimizer (learning_rate = 0.0001) for learning and fitted weights as parameters to the CNN, LSTM and output layers on the forward and backward pass respectively. According to the Figure 4, the weights of nodes are then exchanged in a gossip manner via a ring topology after 5 local epochs, where (Node 1) exchanges $W_1$ with (Node 4) and with (Node 2), (Node 2) with (Node 3) and (Node 3) with (Node 4). It is a lightweight exchange that broadcasts only model parameters and performs a weight aggregation step such that each node computes its weights by mixing local and received weights to encourage network-wide consistency. Its cycle repeats until convergence in order to produce a single model with accurate RUL inference at any node. The loss function to determine the RUL can be described as:

$$L_i = \frac{1}{N_i} \sum_{j=1}^{N_i} (\hat{y}_i, j - y, j)^2$$

where $N_i$ is the number of samples in $D_i$, $\hat{y}_i, j$ is the predicted RUL, and $y_i, j$ is the ground-truth RUL.

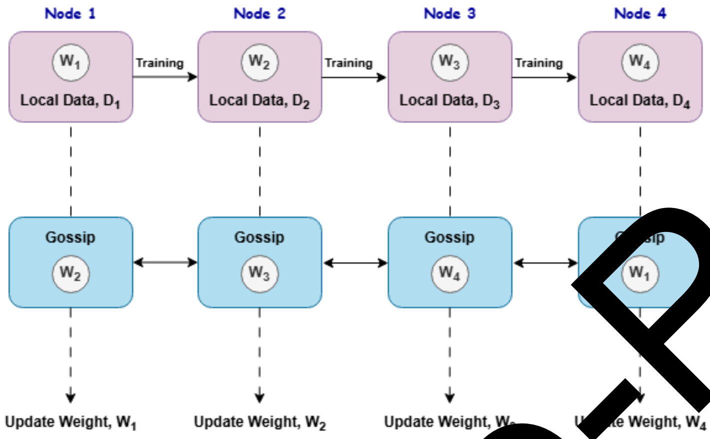



Figure 4: Gossip Mechanism between Different Nodes in Predicting RUL.

And the weight aggregation mechanism can be defined as:

$$W_i \leftarrow (1 - \alpha)W_i + \alpha W_j$$

where $W_j$ is the weight from a neighboring node, and $\alpha = 0.5$ is the mixing coefficient.

## IV. RESULT AND DISCUSSION

This section analyzed how our model performs in forecasting engine RUL using FD001 to FD004 dataset information. The evaluation uses $R^2$ together with Mean Absolute Error (MAE) Loss along with Root Mean Squared Error (RMSE) as key performance metrics. We check the training and validation outcomes against each engine by comparing predicted RUL results to actual RUL results while assessing how well the model tracks the degradation patterns of the components. Our research evaluates the model performance through an examination of state-of-the-art solutions for benchmarking purposes. This analysis enables us to discuss both the positive aspects and existing constraints as well as opportunities to boost accuracy levels in predicting RUL.

### 4.1 Experimental Setup

We employed the proposed GNN in developing models from the FD001 to FD004 dataset collection. We used Google Colab Pro subscription service because its advanced computational strength enabled us to complete the intensive tasks for model training. The large complexity of our models together with extensive dataset sizes demanded the use of V100 GPUs for achieving maximum efficiency. The pre-processor segment included a process that split the data into separate training and validation groups. The model training phase consumed 70% of data but the remaining 30% served as the testing data. The succeeding sections include an evaluation of different models and their ability to predict engine lifetime remaining.

### 4.2 Evaluation Metrics

The assessment of DL model performance matters for understanding their capability to forecast RUL of engines. We used essential metrics to measure both the predictive accuracy and model prediction capabilities for our assessment purpose. The performance evaluation of the model relies on $R^2$ and RMSE and MAE and Loss as individual metrics that measure different aspects.

1) **$R^2$(Coefficient of Determination):** $R^2$ represents the proportion of engine RUL variable variance which can be predicted based on the studied independent variables. The model fit evaluation relies on this measurement because it describes the data-model compatibility. The formula for $R^2$ is:

$$R^2 = 1 - \frac{\sum(y_{actual} - y_{predicted})^2}{\sum(y_{actual} - \overline{y})^2}$$

A higher value of $R^2$ shows that the model performs well in predicting RUL variance.

2) **RMSE (Root Mean Squared Error):** The RMSE value represents the mathematical square root of all predicted and actual value differences which have been squared and averaged. Lower RMSE values indicate better prediction accuracy because they show the extent of error magnitude. The formula is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{actual,i} - y_{predicted,i})^2}$$

Improved prediction accuracy relates to RMSE values decreasing.

3) **MAE (Mean Absolute Error):** This methodology computes the mean average of absolute value errors between predicted results versus actual observed data to provide an easy accuracy measurement mechanism. The method stands out because it detects errors less easily than RMSE which makes it appropriate for real-world prediction scenarios. The formula is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{actual,i} - y_{predicted,i}|$$

The accuracy of a predictive model regarding actual RUL increases when MAE decreases.

4) **Loss:** The true values determine how much error the loss functions create in prediction results. We deployed the loss function to both direct the training process of the model while optimizing its accuracy rate. The model performs more effectively with lower value loss results. Typically, the loss function includes both RMSE and MAE metrics combined when used for prediction tasks.

### 4.3 Sensor Trend Analysis and Feature Relevance

All sensor measurements for engine unit number 4 depicted in Figure5 towards failure while showing their time-series patterns in the next subsection. The different operational settings and sensor measurements from engine performance trends appear as subplots across the recorded cycles.

The listed sensors including sensor and certain setting parameters 1, 5, 6, 10, 16, 18, and 19 demonstrate minimal variations during the entire engine operational period. The studied features show minimal potential to deliver useful information for assessments of RUL. These unchanging sensors may fail to detect patterns related to engine failure or degradation states. This behavioral consistency could be verified using standard deviation analysis of these sensors throughout the complete engine set. The standard deviation results validate the idea that several sensors show no meaningful contribution to predictive modeling since their values remain zero or extremely close to zero. Enhancing the model performance becomes possible through implementing feature selection methods that strip away uninformative sensors from analysis.

Again, analysis of RUL occurs through the box plot in Figure 6, evaluation of risky Exponentially Weighted Moving Average (EWMA) performance for each week. The Weekly Prediction Time is shown on the x-axis which starts at week 48 and ends at week 0 while the y-axis displays the EWMA values. The color gradient transforms as time goes by to show modifications in the RUL distribution before the system fails.

During weeks 48–40 the RUL values demonstrate stable behavior while maintaining low variability. The RUL displays greater variability between weeks 30 and 10 because the system performance becomes increasingly unpredictable as it reaches its failure point. The rising variability creates signs that problems may begin to affect the system. The RUL values show a rising pattern with growing dispersion during the last weeks (1–0) because system behavior noticeably changes prior to failure. The threshold or critical EWMA value depicted by the dashed horizontal line differentiates between regular system operation and dangerous operating states. The depicted picture provides essential insights into RUL pattern development which allows experts to evaluate EWMA reliability for warning indications of system failures.

### 4.4 Model Performance Evaluation

Table 3: Performance Metrics for Engine RUL Prediction across Different Datasets

| Dataset | $R^2$ | Loss | MAE | RMSE |
|---|---|---|---|---|
| FD001 | 93.30% | 354.85 | 11.67 | 12.77 |
| FD002 | 94.57% | 356.14 | 11.54 | 12.87 |
| FD003 | 92.91% | 354.45 | 11.71 | 12.77 |
| FD004 | 92.43% | 353.71 | 11.76 | 12.87 |

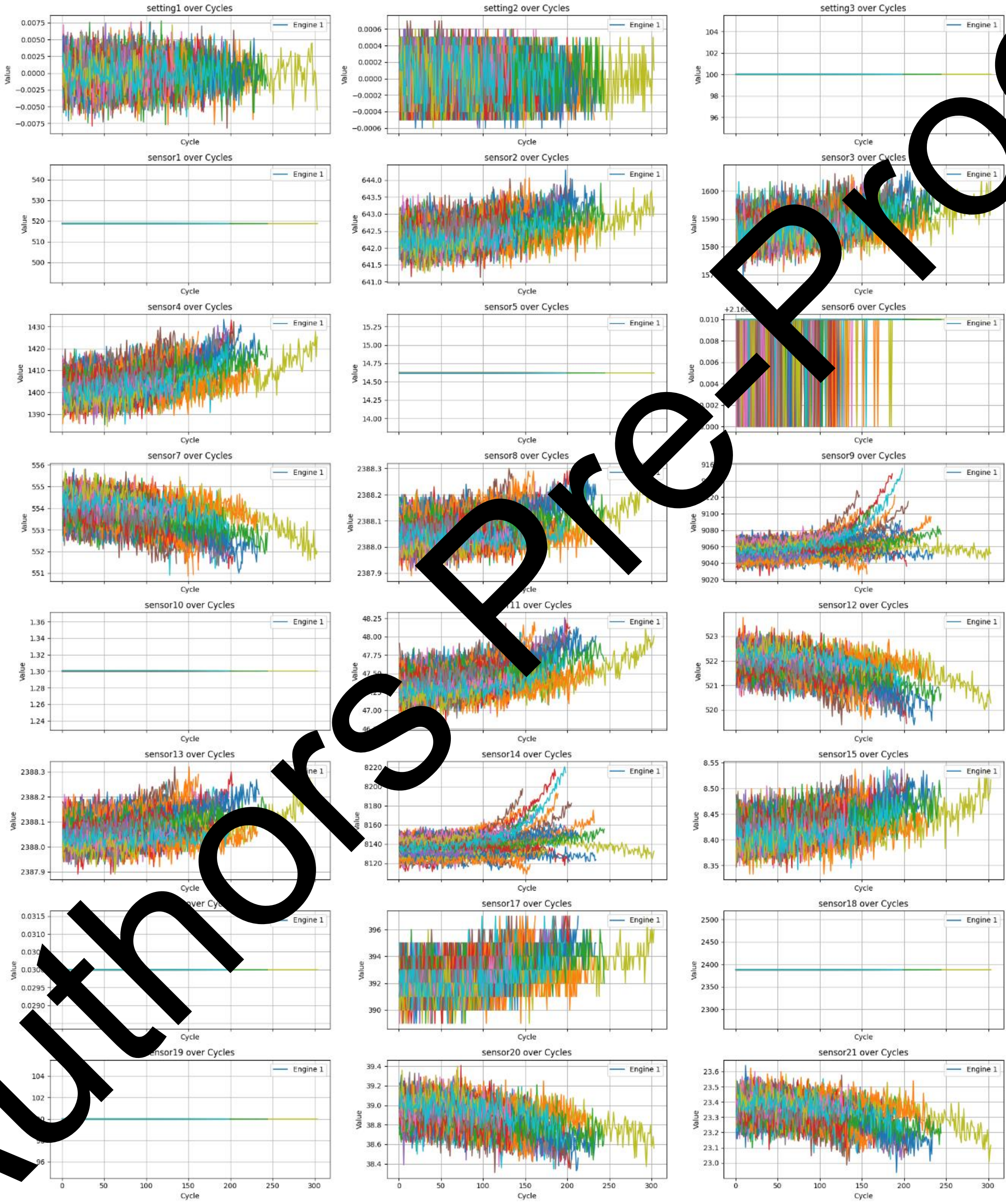



Figure 5: Sensor readings over engine cycles for engine unit 4 where each sensor value changes as the engine progresses toward failure

The RUL prediction model received evaluation through Table 3 by employing FD001 through FD004 datasets. The performance evaluation of the model utilizes R² Loss MAE and RMSE values as multifaceted assessment tools. A high share of 92.43% to 94.57% establishes that the model efficiently identifies engine RUL changes in all recorded datasets.

The model proves its capability to identify hidden degradation patterns in engines during its operational cycle. The predictions show an error range of 353.71 to 356.14 Loss values that represent the difference between estimated and actual RUL measurements. The model shows stable performance through its steady loss values across different datasets. The MAE metrics span between 11.54 and 11.76 which measures the absolute difference between actual and predicted RUL predictions. The model maintains high consistency when predicting RUL because these values display minimal variations between them. The RMSE values demonstrate the size of prediction error through their range from 12.77 to 12.87. The model shows accurate prediction capabilities because the RMSE values demonstrate stable RUL evaluation throughout all the tested datasets. The model demonstrates reliable RUL prediction capabilities with minor error margins throughout all four datasets because of its effective and consistent performance.

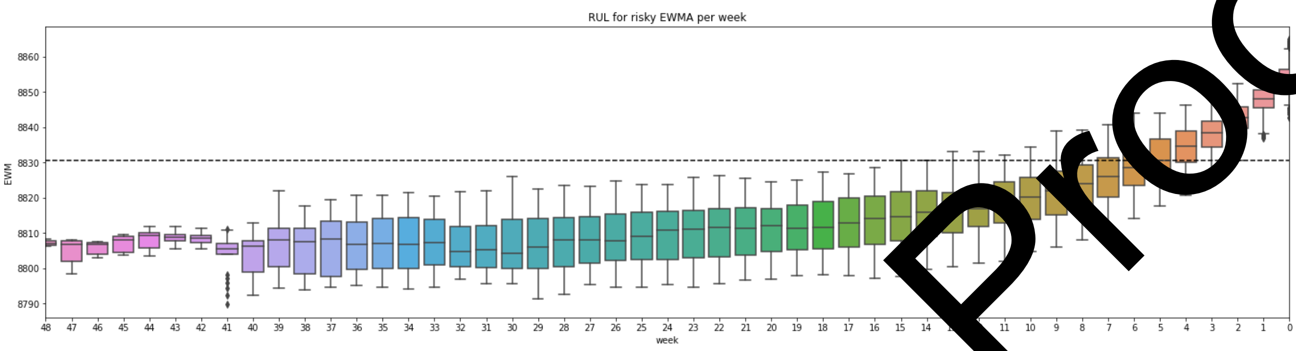



Figure 6: RUL for risky EWMA per week

The analysis of Figure 7 provides vital knowledge about the model's functioning through its presentation of training versus validation loss curves and Mean Absolute Error (MAE) curves for FD001 to FD004 datasets. All datasets demonstrate declining training loss patterns because the model successfully learns to minimize errors throughout the training data. The model demonstrates effective pattern detection by showing this positive indicator. Validation loss curves offer deeper insights regarding generalization than other metrics do. The decrease of validation loss together with training loss indicates that the model successfully applies learned knowledge to new unseen data patterns. A rapid increase in validation loss with declining training loss patterns indicates that the developed model achieves excellent training performance but failed to generalize beyond training instances.

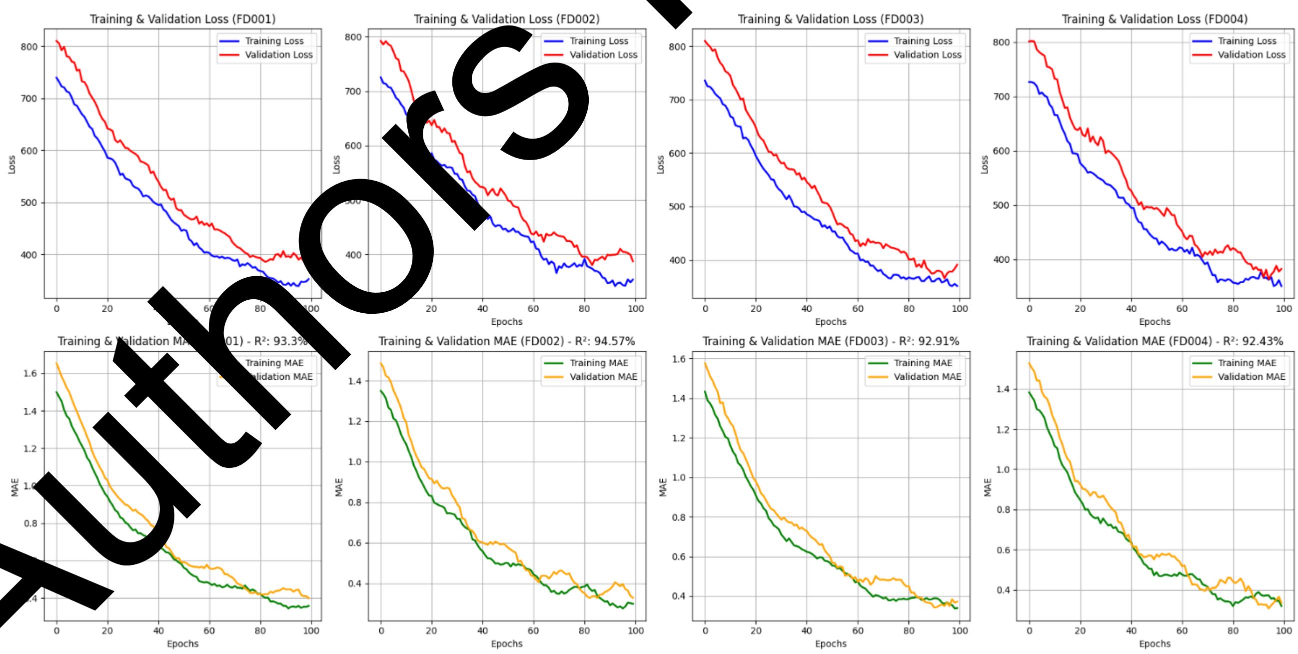



Figure 7: Training and Validation Loss and MAE curves showing learning progress and generalization performance

Table 3: Precision, Recall, and F1-Score for Each Model

The evaluation process becomes more powerful through the addition of MAE curves since they show direct assessment of prediction accuracy. A progressive reduction of MAE during epochs shows better prediction accuracy and this precision stands essential when estimating RUL in predictive maintenance applications. The model displays enhanced predictive accuracy according to the steady decline of MAE in FD001 dataset. The modeling of datasets FD003 as well as FD004 experiences performance variations in the MAE metric which implies difficulties in processing complex and noisy data and could benefit from additional parameter adjustments.

The contrasting performance between datasets (FD001 to FD004 reveals that researchers must treat each dataset uniquely. The convergence patterns of both loss and MAE for datasets demonstrate varying speeds because data complexity and operational conditions affect results differently. Diverse database analysis during performance model evaluation remains essential because it minimizes uncertainties about application reliability in real-world environments. The alignment or separation between training and validation metrics serves as a decision-making tool for early stopping, hyperparameter tuning and regularization implementation to prevent overfitting and enhance generalization.

### 4.5 Predicted vs Actual RUL

The Figure 8 depicts RUL predictions resulting from the Graph Neural Network (GNN) model assessment of three engines (12, 61, and 84) during their operational periods. The Actual RUL values serve as ground truth to evaluate how well the predicted RUL matches them throughout each subgraph. The predictions for engine-12 span 200 operational cycles and engine-61 as well as engine-84 need 160 and 175 operational cycles respectively. Model performance confirms strong results from these plots because Actual RUL tracks Predicted RUL throughout most operational cycles. The examined regions show minor inconsistencies that indicate possible improvement opportunities for model refinement. The Figure 8 demonstrates that the proposed GNN model effectively determines RUL durations as per aviation's predictive maintenance requirements.

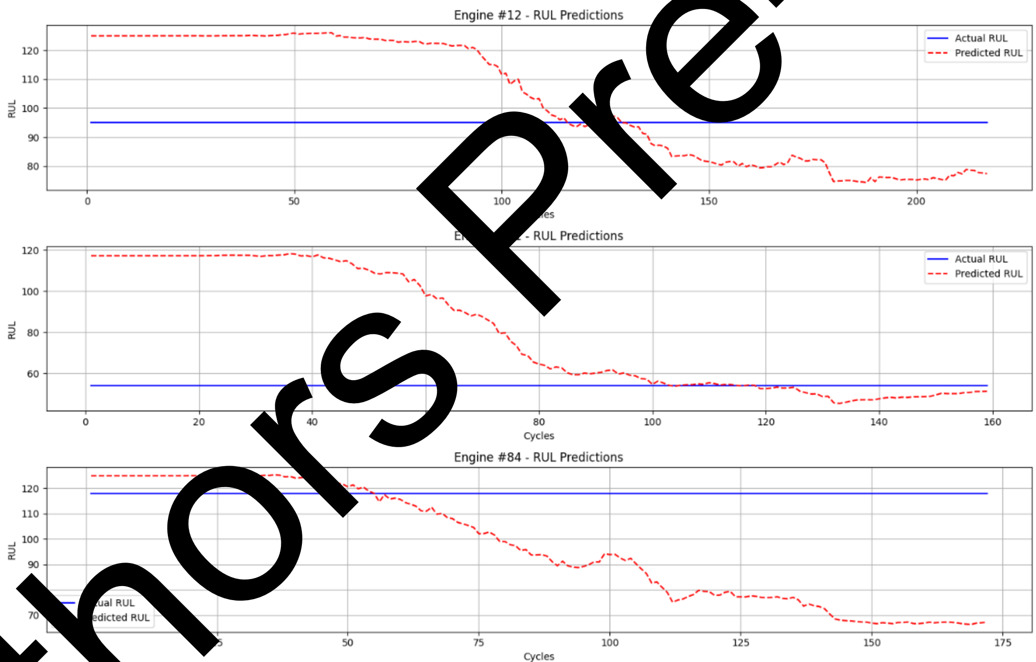
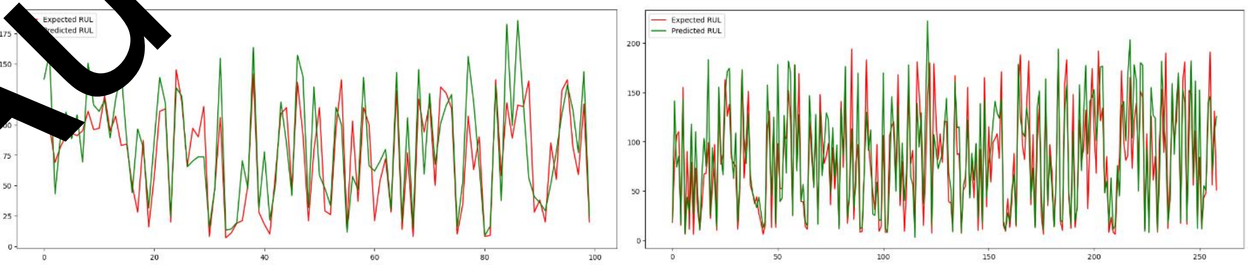
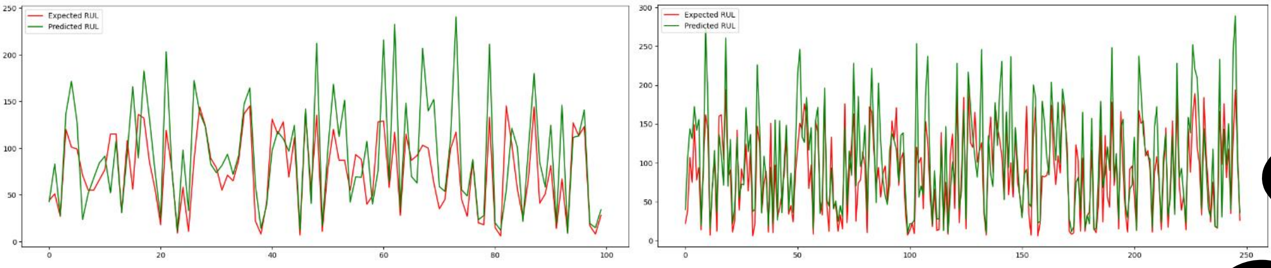



Figure 8: Proposed GNN model RUL predictions over time for engines (12, 61, 84)




a) FD001

b) FD002




c) FD003 d) FD004

Figure 9: Model's performance in predicting RUL

Multiple data points from Figure 9a, 9b, 9c and 9d enable readers to study how the model forecasts RUL of datasets across various operational scales. The Expected RUL chart in Figure 9a declines from 175 to 0 while the predicted RUL follow this pattern but exhibits noticeable deviations in the middle operational period. The Time-series in Figure 9b displays greater Expected and Predicted RUL alignment through its 0 to 250 scale while achieving better performance accuracy at longer operation periods. At 0–100 expected RUL in Figure 9c the predicted RUL properly follows the general pattern although it shows deviations particularly in the lower part while displaying challenges for making precise predictions near RUL zero. The model demonstrates effective modeling capabilities of extended operational durations because Figure 9d shows predictable alignment between Expected RUL and Predicted RUL while showing minimal deviations throughout the 0–300 scale. These figures demonstrate that the model operates optimally during long operational periods as shown in Figure 9b and 9d but shows limitations when analyzing shorter or more detailed operational intervals as shown in Figure 9a and 9c mainly affecting predictions for medium-range and low-RUL scenarios. The comparison demonstrates a need for tailored dataset capabilities together with model improvement methods in order to boost predictive maintenance accuracy and reliability for different operational conditions.

### 4.4 Comparative Analysis and Discussion

The C-MAPSS dataset evaluation for different RUL prediction models appears in Table 4. The proposed GNN-based model received performance evaluation from FPCA-TCN, CNN-GRU, and ESO-BP methodologies. The performance analysis assesses different methods through their R² and RMSE values that evaluate both predictive accuracy and error margin rates.

Table 4: Comparative Analysis of RUL Prediction Models on the C-MAPSS Dataset

| **Reference** | **Model** | **Result** |
|---|---|---|
| Chen et al. [27] | FPCA-TCN | RMSE of 15.56 |
| Sun et al. [28] | CNN-GRU | $R^2$ of 0.91 |
| Zhang et al. [29] | ESO-BP | $R^2$ value of 0.931 on FD001 |
| Ours | GNN | $R^2$ of 0.93.3 on FD001, RMSE of 12.77 |

A comparative study between various RUL prediction methods proves the high performance of our proposed Graph Neural Network (GNN)-based method. Chen et al. [27] implemented FPCA-TCN which delivered an RMSE of 15.56 yet Sun et al. [28] accomplished R² value of 0.91 through CNN-GRU model implementation. Zhang et al. [29] developed ESO-BP model technology to enhance prediction accuracy to reach an R² value of 0.931 when examining the FD001 dataset. Either R² of 93.3% or RMSE of 12.77 from our proposed GNN model outperforms current methods thereby providing exceptional predictive accuracy and minimal error for FD001. The use of graph-based learning methods with sensor data dependency structures improves RUL prediction models so our solution proves to be a promising predictive maintenance alternative.

## VII. CONCLUSION AND FUTURE DIRECTIONS

The proposed GNN provides a decentralized DL approach to RUL engine prediction in distributed mechanical systems while solving traditional centralized diagnostic obstacles. Using CNNs as well as LSTM structure allows the GNN to detect quick anomalies and observe extended degradation patterns in multivariate sensor information. The GNN produces excellent results when tested against CMAPSS data by reaching accurate predictions along with minimal errors. Real-time fault detection and preventive maintenance for aircraft engines and industrial turbines can be achieved through the potential applications of the GNN system.

The research can be expanded through multiple potential directions which aim to improve the capabilities of GNN. Model optimization needs to concentrate on parameter adjustment together with sophisticated attention methods to achieve better interpretation alongside improved prediction results. GNN performance needs to be tested through real-time applications in aircraft fleets as well as industrial turbine networks to prove its operational effectiveness. Extensive testing across various domains should enable the GNN to support diagnostic applications in healthcare and energy systems along with its current use in aerospace systems. The system performance will improve by implementing data quality improvements and sensor feature selection methods for both removing unhelpful data sensors and finding delicate degradation patterns more easily. By integrating the GNN with other architectures like Transformers and Graph Neural Networks (GNNs) the model can achieve better handling of sensor data containing complex and non-linear relationships. Research into network scalability along with fault tolerance for larger systems will establish robust operation of the GNN during node failures and network disruptions.

## REFERENCES


[1] M. M. Quamar and A. Nasir, "Review on fault diagnosis and fault-tolerant control scheme for robotic manipulators: Recent advances in AI, machine learning, and digital twin," *arXiv preprint arXiv:2402.12980*, 2024.

[2] P. J. Blau, *Tribosystem analysis: a practical approach to the diagnosis of wear problems*. CRC Press, 2017.

[3] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for m prediction of lithium-ion batteries," *IEEE Trans Veh Technol*, vol. 67, no. 7, pp. 5695–5705, 2018.

[4] P. Nair, V. Vakharia, H. Borade, M. Shah, and V. Wankhede, "Predicting li-ion battery remaining useful life: an XDFM-driven approach with explainable AI," *Energies (Basel)*, vol. 16, no. 15, p. 5725, 2023.

[5] S. Xie and M. H. Nazari, "Learning-based distributed optimal power sharing and frequency control under cyber contingencies," *International Journal of Electrical Power & Energy Systems*, vol. 152, p. 109248, 2023.

[6] J. Zhao, X. Han, M. Ouyang, and A. F. Burke, "Specialized deep neural networks for battery health prognostics: Opportunities and challenges," *Journal of Energy Chemistry*, vol. 87, pp. 416–438, 2023.

[7] G. D'Angelo and F. Palmieri, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction," *Journal of Network and Computer Applications*, vol. 173, p. 102890, 2021.

[8] E. Karapalidou, N. Alexandris, E. Antoniou, S. Vologiannidis, J. Kalomiros, and D. Varsamis, "Implementation of a sequence-to-sequence stacked sparse long short-term memory autoencoder for anomaly detection on multivariate timeseries data of industrial blower ball bearing units," *Sensors*, vol. 23, no. 14, p. 6502, 2023.

[9] M. Ma and Z. Mao, "Deep-convolution-based LSTM network for remaining useful life prediction," *IEEE Trans Industr Inform*, vol. 17, no. 3, pp. 1658–1667, 2020.

[10] A. R. Al-Ali, R. Gupta, I. Zualkernan, and S. K. Das, "Role of IoT technologies in big data management systems: A review and Smart Grid case study," *Pervasive Mob Comput*, p. 101905, 2024.

[11] Y. Alomari, M. Andó, and M. L. Baptista, "Advancing aircraft engine RUL predictions: an interpretable integrated approach of feature engineering and aggregated feature importance," *Scientific Reports 2023 13:1*, vol. 13, no. 1, pp. 1–14, Aug. 2023, doi: 10.1038/s41598-023-40315-1.

[12] D. J, B. M. Reddy, K. Murari, and B. Rahul, "Predictive Maintenance of Aircraft Engines: Machine Learning Approaches for Remaining Useful Life Estimation," *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, pp. 1679–1686, Jan. 2025, doi: 10.1109/ICMCSI64620.2025.10883526.

[13] Puttaswamy, B. S., and N. Thillaiarasu. "Fine DenseNet based human personality recognition using english hand writing of non-native speakers." Biomedical Signal Processing and Control 99 (2025): 106910.

[14] S. Szrama and T. Lodygowski, "Aircraft Engine Remaining Useful Life Prediction using neural networks and real-life engine operational data," *Advances in Engineering Software*, vol. 192, p. 103645, Jun. 2024, doi: 10.1016/J.ADVENGSOFT.2024.103645.

[15] E. C. Ozkat, O. Bektas, M. J. Nielsen, and A. la Cour-Harbo, “A data-driven predictive maintenance model to estimate RUL in a multi-rotor UAS,” *International Journal of Micro Air Vehicles*, vol. 15, Jan. 2023, doi: 10.1177/17568293221150171/ASSET/IMAGES/LARGE/10.1177_17568293221150171-FIG13.JPEG.

[16] M. D. Dangut, I. K. Jennions, S. King, and Z. Skaf, “A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach,” *Neural Comput Appl*, vol. 35, no. 4, pp. 2991–3009, Feb. 2023, doi: 10.1007/S00521-022-07167-8/FIGURES/17.

[17] L. Wang, Y. Chen, X. Zhao, and J. Xiang, “Predictive Maintenance Scheduling for Aircraft Engines Based on Remaining Useful Life Prediction,” *IEEE Internet Things J*, vol. 11, no. 13, pp. 23020–23031, Jul. 2024, doi: 10.1109/JIOT.2024.3376715.

[18] X. Chen, “A novel transformer-based DL model enhanced by position-sensitive attention and gated hierarchical LSTM for aero-engine RUL prediction,” *Scientific Reports* |, vol. 14, p. 10061, 123AD, doi: 10.1038/s41598-024-59095-3.

[19] Ashwin Shenoy, M., and N. Thillaiarasu. "Enhancing temple surveillance through human activity recognition: A novel dataset and YOLOv4-ConvLSTM approach." Journal of Intelligent & Fuzzy Systems Preprint (2023): 1-16.

[20] Z. Jiangyan, J. Ma, and J. Wu, “A regularized constrained two-stream convolution augmented Transformer for aircraft engine remaining useful life prediction,” *Eng Appl Artif Intell*, vol. 133, p. 108161, Jul. 2024, doi: 10.1016/J.ENGAPPAI.2024.108161.

[21] S. Deng and J. Zhou, “Prediction of Remaining Useful Life of Aero-engines Based on CNN-LSTM-Attention,” *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, pp. 1–12, Dec. 2024, doi: 10.1007/S44196-024-00639-W/TABLES/6.

[22] J. Gao, Y. Wang, and Z. Sun, “An interpretable RUL prediction method of aircraft engines under complex operating conditions using spatio-temporal features,” *Meas Sci Technol*, vol. 35, no. 7, p. 076003, Apr. 2024, doi: 10.1088/1361-6501/AD3B2C.

[23] Q. Zhang, Q. Liu, and Q. Ye, “An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine,” *Eng Appl Artif Intell*, vol. 127, p. 107241, Jan. 2024, doi: 10.1016/J.ENGAPPAI.2023.107241.

[24] F. Gan, H. Shao, and B. Xia, “An adaptive model with dual-dimensional attention for remaining useful life prediction of aero-engines,” *Knowl Based Syst*, vol. 293, p. 111738, Jun. 2024, doi: 10.1016/J.KNOSYS.2024.111738.

[25] Thillaiarasu, N., and Ashwin Shenoy. "Enhancing Security Through Real-Time Classification of Normal and Abnormal Human Activities: A YOLOv7-SVM Hybrid Approach." IAENG International Journal of Computer Science 51.8 (2024).

[26] “NASA Turbofan Jet Engine Data Set.” Accessed: Mar. 06, 2025. [Online]. Available: https://www.kaggle.com/datasets/behrad3d/nasa-cmaps/data

[27] J. Chen, Y. Wei, X. Sun, A. Zeb, M. S. Meiabadi, and S. S. Karganroudi, “FPCA-SETCN: A Novel Deep Learning Framework for Remaining Useful Life Prediction,” *IEEE Sens J*, 2024, doi: 10.1109/JSEN.2024.3447717.

[28] S. Sun, J. Wang, Y. Xiao, J. Peng, and X. Zhou, “Few-shot RUL prediction for engines based on CNN-GRU model,” *Scientific Reports 2024 14:1*, vol. 14, no. 1, pp. 1–20, Jul. 2024, doi: 10.1038/s41598-024-66377-3.

[29] X. Zhang, N. Xu, W. Dai, G. Zhu, and J. Wen, “Turbofan Engine Health Prediction Model Based on ESO-BP Neural Network,” *Applied Sciences 2024, Vol. 14, Page 1996*, vol. 14, no. 5, p. 1996, Feb. 2024, doi: 10.3390/APP14051996.