National College of Ireland

Enhancing Remaining Useful Life (RUL) Prediction through CNN-Based Fusion Architectures

MSc Research Project
Data Analytics

Zin Win Phyo
Student ID: X23402849

School of Computing
National College of Ireland

Supervisor: Hamilton Niculescu

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Zin Win Phyo
Student ID:	X23402849
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Hamilton Niculescu
Submission Due Date:	16/08/2025
Project Title:	Enhancing Remaining Useful Life (RUL) Prediction through CNN-Based Fusion Architectures
Word Count:	7769
Page Count:	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Zin Win Phyo
Date:	11th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Remaining Useful Life (RUL) Prediction through CNN-Based Fusion Architectures

Zin Win Phyo
X23402849

Abstract

Predictive Maintenance (PdM) enables proactive detection of equipment degradation and reduction of downtime and operation costs. Remaining Useful Life (RUL) estimation is central to PdM using sensor-derived time-series data to predict the time until an equipment fails. Although Convolutional Neural Networks (CNNs) are capable of learning the spatial features effectively, their limited capacity to capture long-term temporal dependencies can restrict the prediction accuracy. This study evaluated three CNN-based fusion architecture, which are CNN+Transformer Encoder, Multi-Scale CNN, and CNN+ Autoencoder on the NASA CMAPSS dataset. This was done using a unified experimental framework to ensure consistent preprocessing, training, and evaluation. Model performance was evaluated in terms of predictive accuracy, computational efficiency, and model complexity. Results indicate that every fusion model outperform the baseline standalone CNN model, with CNN + Autoencoder delivering the highest accuracy, Multi-Scale CNN performing a good trade-off between performance and efficiency, and CNN + Transformer enhancing temporal modelling capabilities at the expense of greater computational demand. These findings provide practical guidance for selecting CNN-based fusion strategies to implement the PdM system on the real-world industrial deployment, balancing predictive capability with resource requirements.

Keywords: Predictive Maintenance (PdM), Remaining Useful Life (RUL), Convolutional Neural Networks, Transformer Encoder, Multi-Scale CNN, Autoencoder

1 Introduction

1.1 Background

The evolution of Industry 4.0 and Industrial Internet of Things (IIoT) have led the industrial systems more interconnected, data-driven and intelligent (Kang et al.; 2016; Qin et al.; 2016). These changes have transformed the maintenance and reliability management in the manufacturing sector and other industries significantly. The most crucial component of this transformation is the shift from traditional reactive or scheduled maintenance to PdM which is a proactive approach that aims the prevention of equipment breakdowns prior to their occurrence (Lee et al.; 2014). This not only reduces unplanned downtown and maintenance costs but also improves operational efficiency and safety in complex industrial environments (Lee et al.; 2015).

RUL estimation serves as a foundation for PdM, and this involves estimating the remaining operational life of a machine or component (Si et al.; 2011). Effective prediction of RUL will allow to schedule the interventions at the most appropriately calculated time, which will prevent the unnecessary servicing as well as the unexpected breakdowns (Lei et al.; 2020). This goal can be achieved with the help of modern sensor technologies, which allow continuous monitoring of machine conditions and generate large volumes of multivariate time-series data capturing wear and degradation patterns (Lee et al.; 2015).

To effectively analyze this complex sensor data, researchers have adopted to use of deep learning techniques. Particularly, CNNs have been widely used as they are good at extracting spatial features without the need of extensive manual feature engineering (LeCun et al.; 2015). However, CNNs have weakness in their ability when it comes to capturing temporal dependencies especially the long-term patterns and degradation trends (Li et al.; 2018). This limitation becomes a significant bottleneck in the modelling of the dynamic behavior of machinery operation in variable conditions.

To address this gap, there has been recent research on hybrid architectures that combine CNNs with other deep-learning models that are capable of learning temporal structures. These hybrid models aim to improve the robustness and accuracy of RUL prediction. However, there is a lack of comparative research that evaluates the fusion strategies under a standardized experimental setup despite the growing number of studies (Lei et al.; 2018; Fink et al.; 2020).

Most of the existing studies analyze one of the architectures independently, using different datasets, preprocessing techniques, or evaluation metrics, thus preventing to draw reliable conclusions for the comparison (Bousdekis et al.; 2018). This study seeks to address this gap by performing a comparison side-by-side of three CNN-based fusion models with a common experimental pipeline. By doing so, the outcomes of this study led to a clearer understanding of their strengths and trade-offs. Moreover, this provides practical guidance for selecting appropriate model for real-world predictive maintenance applications.

1.2 Importance of the Study

Since the industries are adopting data-driven solutions for the maintenance, the choice of deep-learning models to use in RUL prediction becomes increasingly important. The real-world scenario require precision, and they must also require models that remain computationally efficient. The hybrid CNNs architectures have demonstrated new possibilities to capture both spatial and temporal patterns in sensor data (Zhang et al.; 2019). However, without a standard benchmark or direct comparison, it remains unclear which fusion strategy is actually performing better than the others under different conditions (Lei et al.; 2018).

This study is important because it directly bridge that gap by comparing three widely used CNN-based fusion technologies, CNN + Transformer Encoder, Multi-Scale CNN Fusion, and CNN + Autoencoder. These three models are trained and evaluated on the same environment and their performance is measured using the dataset of Commercial Modular

Aero-Propulsion System Simulation (CMAPSS) which was created by NASA Prognostics Center of Excellence, a benchmark dataset for RUL prediction studies. The aim of this research is to get the assessment of their predictive performance, computational efficiency and trade-offs. The results are expected to guide both academic researchers and industrial practitioners in making informed decisions when choosing model architectures for predictive maintenance systems. Ultimately, this work contributes to more reliable and efficient RUL estimation, , which is part of the overall trend to Industry 4.0 (Qin et al.; 2016).

1.3 Research Question and Objectives

This study is guided by the following research question:

"What is the impact of different fusion strategies for CNN-based models (CNN + Transformer Encoder, Multi-Scale CNN Fusion, and CNN + Autoencoder) on predictive maintenance performance compared to standalone CNN models?"

The objective of this study is to evaluate and compare the performance of three CNN-based fusion strategies when applied to predict RUL. The three models are trained under the same experimental conditions with NASA CMAPSS benchmark dataset. The evaluation focuses on predictive accuracy, computational effectiveness, and model complexity. The results are aimed to provide actionable insights to use in real-world predictive maintenance applications.

1.4 Limitations and Assumptions

While this study aims to compare CNN-based fusion approaches to RUL estimation, several limitations must be acknowledged. First, the assessment relies exclusively on NASA CMAPSS dataset, which is a well-known dataset that addresses flight engine degradation during simulation. As it only represents simulated but not real-world conditions, its results may not be easily applied to other industrial situations or equipment.

Second, this research narrows down to three particular fusion architectures and does not consider other deep-learning models and traditional machine learning models. These alternatives might provide alternative perspectives. Additionally, the study assesses the computational efficiency and resource requirements by the model complexity and training time, without deployment on embedded devices. These limitations suggest that the findings are valuable for benchmarking within a controlled environment. However, further research is required to validate in making concrete conclusions for real-world environments.

1.5 Structure of the Report

The rest of this report is organized as follows:

- **Related Work** presents a critical analysis of related work on CNN-based fusion models for RUL prediction.

- **Methodology and Design Implementation** outlines the datasets, preprocessing steps, model architectures, and evaluation strategy.
- **Evaluation & Results** discusses the experimental outcomes with comparative performance insights.
- **Conclusion and Future Work** summarizes the findings and proposes directions for future research.

2 Related Work

In the last several years, RUL prediction has become a focus in PdM research, primarily due to the availability of sensor data for research studies. Traditional statistical and machine learning models are unable to manage the complicated degradation patterns of modern machinery, particularly when operating conditions varies.

To address these challenges, deep learning has become a powerful alternative as they can learn high-dimensional, nonlinear patterns directly from raw sensor data. CNNs have gained popularity due to their capability at extracting spatial features and working with multivariate time series data with limited preprocessing. However, standalone CNNs have a limitation in capturing capture long-range temporal dependencies, which are critical in modeling degradation over time. This has led to an increasing interest in hybrid models that combine CNNs with other architectures to incorporate temporal modeling capabilities and improve prediction accuracy.

2.1 CNN-Based Models for RUL Prediction

CNNs has become the backbone of many RUL prediction models due to their computational efficiency and the capability in local feature extraction. Zhang et al. (2019) showed that CNN-based models can learn sensor patterns from turbofan engine datasets better than the conventional feature engineering pipeline. Still, Du et al. (2023) described that standalone CNNs struggle to model long-sequence dependencies, particularly when degradation signals are gradual or noisy.

To mitigate this, many researchers have investigated architectural improvements and fusion methods which remain the benefits of CNNs while improving their temporal awareness. Wang et al. (2021) proposed a multiscale convolutional attention network that improved the prediction accuracy by integrating convolutional layers with attention mechanisms, which highlights the need of temporal fusion. These developments mark a shift in RUL research towards hybrid deep-learning models which are the focus of this study.

2.2 CNN + Transformer Encoder Architectures

The recent developments in deep learning have revealed that the combinations of CNNs and Transformer Encoders can enhance RUL prediction with a significant contribution of both spatial and temporal feature learning. Transformer-based models which are originally developed for natural language processing, have gained popularity in time-series forecasting due to their self-attention mechanism that allows capturing long-range

dependencies.

Zhang et al. (2022) introduced the Dual-Aspect Self-Attention Transformer (DAST), a parallel processing algorithm that considers both the sensor and time dimensions, demonstrating superior performance on turbofan engine datasets compared to recurrent architectures. The findings indicate that incorporating self-attention to CNN-based pipeline can improve accuracy and interpretability. Qin et al. (2022) proposed the Temporal Deep Degradation Network (TDDN), that combines CNNs and attention mechanism to extract the features from long sequences while maintaining robustness in noisy environments. However, both studies require significant computational resources and large datasets to train effectively.

Further, Du et al. (2023) introduced the Trans-Lighter framework, a lightweight hybrid of CNNs and Transformer Encoders that are optimised for resource-limited industrial environments. Their work emphasizes model simplification while preserving predictive performance, addressing a key challenge in deploying attention-based systems at scale. Similarly, Wang et al. (2023) explored a CNN Transformer hybrid on multivariate sensor streams, and they found that it led to better RUL prediction accuracy compared with standalone CNNs. In general, these studies indicate the effectiveness of CNN Transformer fusion, but also indicate trade-offs in model complexity and training expenses.

Although existing literature shows several promising results, many previous studies evaluate only a single model architecture without comparing it to alternative fusion strategies under consistent conditions. The current paper fills that gap by comparing CNN + transformer encoder models to other CNN-based fusion approaches in controlled settings through the NASA CMAPSS dataset.

2.3 Multi-Scale CNN Fusion Approaches

Multi-scale feature extraction has been emerged as a useful approach to RUL prediction, which allows capturing degradation patterns occurring at different temporal resolutions. Unlike standard CNNs which use fixed kernel sizes, multi-scale architectures utilize parallel convolutional filters of varying sizes. This model enables the extraction of both fine-grained local features and border global patterns thus enhancing modelling of machinery behaviors that degrades over time.

Wang et al. (2021) proposed Multi-Scale Convolutional Neural Network (MSCNN) which integrates multiple convolutional branches with varying receptive fields are used to simultaneously capture local anomalies and long-term degradation. The MSCNN had a higher accuracy of RUL estimation when tested on bearing datasets compared with traditional single-scale CNNs. This concept was expanded by Zhou and Wang (2024) with an Adaptive Multi-Scale Feature Fusion (AMFF) framework which combines multi-scale feature extraction. Through a dynamic weighting of the input of each feature scale, the AMFF system demonstrated improved generalization across various operation regimes.

Elizar et al. (2022) conducted a systematic review of the multi-scale deep learning approaches, concluding that in spite of multi-scale designs offering significant gains in feature richness and robustness, they also introduced higher computational complexity and require more careful tuning. However, in the majority of comparative studies multi-scale architectures are either overlooked or evaluated in isolation.

The current study combines the Multi-Scale CNN Fusion in a unified framework with other CNN-based fusion methods. This methodical comparison analyses the performance of the Multi-Scale CNN Fusion in terms of predictive accuracy, computational efficiency and scalability, using the NASA CMAPSS dataset.

2.4 CNN + Autoencoder Architectures

Autoencoders have been widely adopted in the field of RUL prediction due to their ability to learn compressed representations of input data, which can help reduce noise and improve generalization. When combined with CNNs, these architectures obtain the benefit from both spatial feature extraction and unsupervised pretraining, making them suitable for complex sensor data in industrial environments.

Fathi et al. (2021) proposed a convolutional autoencoder-based framework for predictive maintenance of robotic systems with robust anomaly detection and RUL prediction performance even in the absence of explicit run-to-failure sequences. The use of CNN + Autoencoder pipelines was expanded to hybrid pipelines by Abdelli et al. (2022), who used CNNs with attention-based gated recurrent units (GRUs) and autoencoders for monitoring the health of semiconductor lasers to improve model robustness. Even though they used GRUs, the autoencoder-based representation served as the core of the initial feature extractor, highlighting its critical role in the model performance.

Other studies, in particular, Cheng et al. (2020), have explored hybrid autoencoder-based RUL prediction systems where encoder-decoder framework enabled temporal degradation tracking across sensor modalities. However, CNN + Autoencoder architectures can easily be restricted in terms of modeling long-range sequential dependencies unless paired with other sequence-aware modules. Therefore, they are best suited in cases where denoising, dimensionality reduction or anomaly detection are critical.

The current study uses CNN + Autoencoder as one of the three fusion models to be studied, whose performance will be compared to that of CNN + Transformer and Multi-Scale CNN models to understand its trade-offs in terms of predictive power, efficiency, and complexity.

2.5 Summary of Literature Review

In the reviewed literature, it can be seen that hybrid deep learning architectures have improved RUL prediction. Such hybrid architectures outperform standalone CNNs by incorporating mechanisms that capture long-term temporal dependencies, manage varying signal resolutions, and learn compressed feature representations that are robust to noise.

However, there is still a critical gap to the existing model comparison and evaluation scheme. The majority of the recent works focus on one architecture only and present performance results based on different preprocessing procedures, datasets and evaluation metrics. The lack of standardization makes it difficult to draw reliable conclusion about the effectiveness of different fusion strategies.

In order to address these, the current study systematically implements and evaluates three CNN-based fusion models, including CNN + Transformer Encoder, Multi-Scale CNN Fusion, and CNN + Autoencoder, in a unified experimental framework. The same preprocessing pipelines and evaluation metrics are used to train and test all the models on NASA CMAPSS dataset. This is a controlled experimental setup that allows a fair, parallel comparison of predictive performance, computational efficiency and model complexity. By doing so, the conclusions drawn explain the strengths and limitations of each architecture and provide practical guidance for model selection to be used in real-world predictive maintenance systems.

3 Methodology & Design Implementation

This study involves a structured, comparative experimental methodology to evaluate the effectiveness of CNN-based fusion strategies for RUL estimation, using the NASA CMAPSS dataset. It is also based on the Knowledge Discovery in Databases (KDD) process that provides as a well-established framework to extract useful knowledge out of large and complex datasets over the sequential phases of data selection, preprocessing, transformation, modeling and evaluation.

The methodology process starts with identifying and selecting relevant data from the CMAPSS dataset. This step focuses on selection of the most suitable engine subsets that can be used to estimate the RUL estimation based on practical relevance. Data preparation is the second step, and the raw multivariate sensor readings are pre-processed into a structured format for supervised learning. This consists of normalization, temporal segmentation and label constructions, which is used to capture temporal degradation patterns. These tasks play a critical role in helping the models to capture temporal degradation patterns effectively and will be discussed in details in the later sections.

The core of this study lies in the comparative design and evaluation of multiple deep learning models on RUL prediction. A baseline CNN model is implemented as a reference point, in comparison with which three CNN-fusion architectures are evaluated. These models are trained and tested using consistent configurations and under controlled experimental conditions to ensure fairness and reproducibility. This enables a systematic comparison of model performance and allows to explore questions of which fusion-strategy is relatively more effective.

The final step of the methodology involves a systematic evaluation of all the models using appropriate quantitative metrics. It is designed to assess the predictive accuracy and reliability of each model in estimating RUL. By implementing a consistent evaluation process across all models, this study ensures a fair comparison between the baseline and the other three fusion-based CNN models. The insights derived from this comparison

study inform conclusions regarding the practical effectiveness of each modeling strategies with the context of data-driven predictive maintenance.

3.1 Dataset Description

This study is based on CMAPSS dataset which was created by NASA's Prognostics Center of Excellence[1]. The dataset is one of the widely recognized benchmarks for data-driven prognostics. It is commonly used in the development and evaluation of RUL prediction models. It simulates the degradation of aircraft turbofan engines in different operational conditions and fault modes, to provide realistic run-to-failure time-series data for predictive maintenance research.

CMAPSS is comprised of 4 sub-datasets such as FD001, FD002, FD003, and FD004 based on the different sets of operational settings and fault complexities. To conduct this study, only FD001 will be used, since it contains a single operating condition and a single fault mode. This controlled setup simplifies the design of the experiment and enables clearer comparative analysis of the evaluated deep learning models.

The FD001 subset of CMAPSS data is a run-to-failure multivariate time-series data which represents the degradation of aircraft engines. Each engine instance records operational setting and sensor measurements from initial operation until its failure. The dataset contains 3 operational settings and 21 sensor readings, recorded at different cycle length across engines, which make it well-suited for modelling temporal degradation patterns under a controlled scenario.

Column headings are not provided in the original dataset. For better consistency and easier processing in this study, the operational settings and sensor measurements will be assigned to as generic column names (e.g. setting_1, sensor_1, etc.). But to facilitate Exploratory Data Analysis (EDA) and better interpretability, descriptive labels of the sensors were referenced from Saxena et al. (2008) . These descriptive labels are not used in modeling but they are applied in the EDA in order to enhance their visualization and interpretation. A summary of the assigned column names can be found in Table 1., and the corresponding sensor description used during EDA are provided as a supplementary dictionary.

[1]Dataset available at
`https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository`

Sensor Name	Description
sensor_1	Fan Inlet Temperature (°R)
sensor_2	LPC Outlet Temperature (°R)
sensor_3	HPC Outlet Temperature (°R)
sensor_4	LPT Outlet Temperature (°R)
sensor_5	Fan Inlet Pressure (psia)
sensor_6	Bypass-Duct Pressure (psia)
sensor_7	HPC Outlet Pressure (psia)
sensor_8	Physical Fan Speed (rpm)
sensor_9	Physical Core Speed (rpm)
sensor_10	Engine Pressure Ratio (P50/P2)
sensor_11	HPC Outlet Static Pressure (psia)
sensor_12	Ratio of Fuel Flow to Ps30 (pps/psia)
sensor_13	Corrected Fan Speed (rpm)
sensor_14	Corrected Core Speed (rpm)
sensor_15	Bypass Ratio
sensor_16	Burner Fuel-Air Ratio
sensor_17	Bleed Enthalpy
sensor_18	Required Fan Speed
sensor_19	Required Fan Conversion Speed
sensor_20	HPT Cool Air Flow
sensor_21	LPT Cool Air Flow

Table 1: Sensor Dictionary

The life cycle of the respective engine is terminated by a failure event, but the dataset does not provide RUL labels. These are generated by counting down from the last cycle of each engine to zero. The approach converts the dataset into supervised learning format in which each time step is associated with corresponding RUL value. This structure enables the predictive models to learn the temporal relationship between sensor patterns and the proximity to engine failure.

3.2 Exploratory Data Analysis (EDA)

EDA was carried out to get a good insight into the structure and characteristics of the CMAPSS FD001 dataset before the preprocessing and implementation of the models. The main purpose was to evaluate the behavior of engine degradation and distributions of sensor signals in order to make informed decisions for feature selection and model designs.

The first analysis of engine cycle lengths showed that there were a lot of difference regarding the operational lifespan of each unit. Some of the engines had less than 150 cycles, meanwhile others had more than 300 cycles. This supports the use of sequence-based modeling techniques and justifies why the time series are segmented into fixed-length input windows. The distribution of engine lifespans is shown in Figure 1.

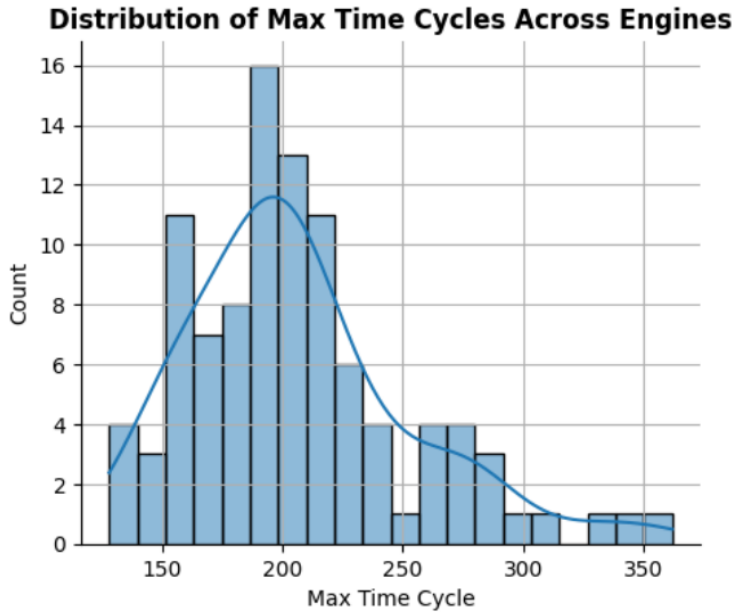
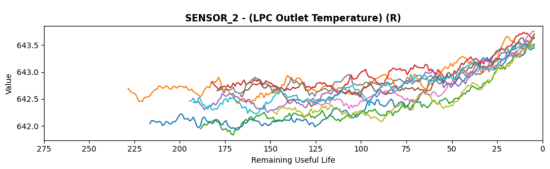
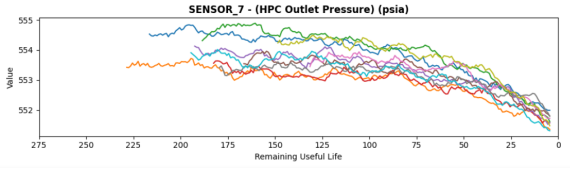


Figure 1: Distribution of Engine Lifespans

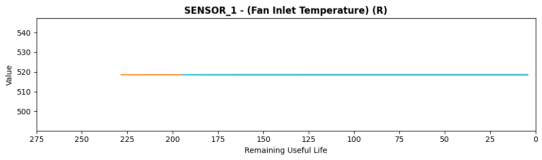
Boxplots and histograms at sensor level were analyzed for the range, variability and stability. This helped in detecting sensors with consistently narrow value ranges, limited fluctuation, or abnormal outliers.To complement it, the time-series plots of individual engines were created to visualize how the sensor readings changed over time. Some sensors demonstrated clear temporal degradation patterns while others remained flat and constant. Examples of these sensor trends are illustrated in Figures 2.These findings were used to make early decision on preliminary feature selection and dimensionality reduction. The sensor dictionary used to support interpretability during this phase is provided in Table 1.



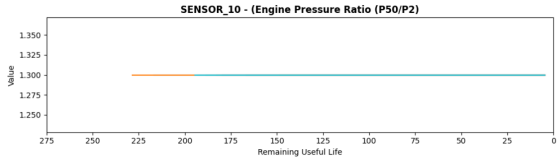
(a) Sensor 2 – Degradation Trend



(b) Sensor 7 – Degradation Trend



(c) Sensor 1 – Constant Signal



(d) Sensor 10 – Constant Signal

Figure 2: Sensor behavior over time: (a–b) showing degradation patterns; (c–d) showing constant signals.

The dataset did not contain any missing value. Moreover, correlation analysis between sensors revealed moderate to strong linear relationships between specific sensor

pairs, which is an indicator of potential redundancy and also used as regularization in the modeling phase.

Based on the results of EDA, some early decisions were made regarding feature engineering. The sensors with low variability or minimal contribution to degradation trends were marked for exclusion in subsequent preprocessing steps. The insights obtained through distributional and temporal patterns helped to simplify the input space and guided subsequent model input design, minimizing noise and focusing attention on more informative signals.

3.3 Data Preprocessing and Transformation

To ensure the consistent and meaningful input across all models, preprocessing pipeline was applied to the CMAPSS FD001 dataset. This process involved feature selection, label generation, normalization, and time-series windowing by using Python and its libraries such as Pandas, NumPy, and Scikit-learn.

In the first stage, constant and non-informative sensors were eliminated. This choice was supported by correlation analysis conducted during EDA. Not all of the sensors with significant variation were capitalized. Only those sensors demonstrating meaningful variation or relevance to degradation trends were retained.

RUL label was generated by subtracting each engine's current cycle number from its final recorded cycle. It gave a decreasing value of RUL at each time slot ending at zero at failure. All the continuous features were then normalized using Min-Max to the range of [0,1] to make the features comparable with one another regardless of their scale and facilitate more stable and efficient model training.

Sliding windows approach was used to segment the time-series data. Each window comprised 30 consecutive time steps to be used as input features, and the RUL value in the final time step was used as corresponding target label. This format will allow the deep learning models to learn from short-term temporal and remaining compatible with the convolutional network structure.

The decision to use of 30-step window was based on prior literatures. The past study on the RUL prediction based on the NASA CMAPSS dataset have chosen window lengths of 20 - 50 time steps because this range balances the ability to capture short-term degradation patterns without overburdening the model (Li et al.; 2018). This study also confirmed that a 30-step window provided more stable convergence and lower validation error than 20 or 50-step. Thus, the chosen window size reflects a compromise between model performance, computational efficiency, and temporal resolution.

A reproducible and fair comparison was conducted across all models by providing the same preprocessed dataset as input to all model variants. Pre-processing pipeline was encapsulated into re-usable data loading functions so that the same data loading instruction could be used during training, validation, and evaluation phases to ensure consistency across all models.

3.4 Model Architecture Design

All the architectures introduced in the study are based on the standardized experimental framework, such as input-output structure and are trained under identical conditions. The only difference between them is how each model processes the same input sequences to extract temporal and spatial features. In this section, the architectural design of all models is outlined with an emphasis on how variations in the structure affect their effectiveness in extracting the degradation patterns, which are applicable in RUL prediction.

3.4.1 Baseline CNN Model

The baseline model used in the study is one-dimensional CNN model. We use this model as the reference point to evaluate the impact of the proposed CNN-based fusion strategies. Although the main goal of this research is to experiment with more advanced architectural settings, this baseline model serves as a meaningful benchmark to access whether the fusion-based enhancements actually contribute to the measurable improvements in RUL prediction process.

In order to ensure that the baseline model was well-optimized and a fair point of comparison, a comprehensive Grid Search was performed in order to tune the key hyper-parameters of the baseline model. It was done systematically exploring combinations of convolutional filter sizes, kernel dimensions, dropout rates, dense layer size and learning rate. The best performing configuration identified during this search was then reused across all fusion models to isolate the effects of architectural changes while controlling for model capacity.

The finalized architecture of the baseline model consists of two convolutional blocks, each followed by pooling and dropout layers for regularization. The convolutional layers are designed to capture localized temporal patterns from the input sequences. The resulting feature maps are then flattened and passed through a fully connected dense layer. The final output layer is a single unit layer without activation, and it is suited to continuous regression tasks. This model can then learn the short-term temporal dependencies effectively from the sliding window segments of multivariate sensor data.

The final configuration selected through Grid Search included two convolutional layers with 128 filters each, a kernel size of 5, a dropout rate of 0.3, and 128 dense units. To train the model, the Adam optimizer was employed, and its learning rate was set to 0.0005. These parameters were obtained through a constrained Grid Search, where the scope of hyperparameters was intentionally limited due to computational resource constraints and system limitations. The search space was informed by values where were commonly used in previous deep learning studies for RUL prediction.

Table 2 contains a summary of these values, and they were passed to all the subsequent fusion models in order to be consistent and fair in the evaluation.

Hyperparameter	Grid Search Options	Selected Value
Number of Filters (Layer 1)	64, 128	128
Number of Filters (Layer 2)	128, 256	128
Dense Layer Units	100, 128	128
Dropout Rate	0.3, 0.5	0.3
Learning Rate	0.001, 0.0005	0.0005
Kernel Size	3, 5	5

Table 2: Selected Hyperparameters for Baseline CNN Model

3.4.2 CNN + Transformer Encoder

This architecture is an expansion of the baseline CNN model by incorporating Transformer Encoder layer to improve temporal modeling capabilities. Although CNNs can be used to extract the localized spatial patterns, they are limited in their ability to capture the long-range temporal dependencies, that are essential in modeling the degradation trends in RUL prediction. The Transformer Encoder overcomes this weakness by incorporating self-attention mechanisms that allow the model to weigh and integrate information across all time steps in the input sequence.

In this architecture, the initial convolutional blocks are retained to perform the local feature extraction. The output feature maps of CNN layers are then fed through Transformer Encoder layer, which contains multi-head self-attention, layer normalization, and residual connection. This allows the model to learn both spatial and global temporal relationships in the same architecture.

The output of Transformer Encoder is flattened and fed to a dense layer, and followed by a single unit output layer for regression. Using the strengths of both convolution and attention mechanisms, this hybrid model will enhance the predictive accuracy compared to the baseline standalone CNN model, especially in capturing complex and long-term dependencies in multivariate time-series sensor data.

All non-Transformer components of this model (preprocessing steps, core training hyperparameters such as convolutional filter sizes, kernel size, dropout rate, dense units, and learning rate) were inherited directly from the optimized baseline CNN configuration. This consistency provides a fair and controlled comparison. This isolates the effect of the Transformer-based fusion strategy. Transformer Encoder block itself was configured with with a head size of 64, 2 attention heads, a feed-forward dimension of 128, and a dropout rate of 0.3. These values were selected through a targeted Grid Search as shown in the following Table 3.

The scope of the parameter lists was decided by commonly use configurations in time-series foresting models. It was kept limited in the consideration of computational resource constraints, balancing model complexity and feasibility. The Transformer Encoder block was applied after CNN feature extraction to improve temporal modeling through global self-attention. The chosen configuration demonstrated strong performance on the validation set while maintaining a manageable number of parameters and training cost.

Hyperparameter	Selected Value	Grid Search Options	Remarks
Transformer Head Size	64	32, 64	Tuned via Grid Search
Transformer Attention Heads	2	2, 4	Tuned via Grid Search
Feed-Forward Dimension	128	128, 256	Tuned via Grid Search
Transformer Dropout Rate	0.3	0.3, 0.4	Tuned via Grid Search
Convolutional Filters (Layer 1 & 2)	128, 128	64, 128 $\rightarrow$ 128; 128, 256 $\rightarrow$ 128	Inherited from Baseline CNN
Kernel Size	5	3, 5	Inherited from Baseline CNN
Dropout Rate	0.3	0.3, 0.5	Inherited from Baseline CNN
Dense Layer Units	128	100, 128	Inherited from Baseline CNN
Learning Rate	0.0005	0.001, 0.0005	Inherited from Baseline CNN

Table 3: Selected Hyperparameters for CNN + Transformer Model

3.4.3 Multi-Scale CNN

Multi-Scale CNN architecture enhances the baseline model by adding parallel convolutional branches that different kernel sizes in order to extract the temporal patterns in multiple resolutions. This approach is useful in RUL prediction, where degradation signals can evolve over both short and long time scales. The architecture particularly enhances the ability of the model to extract complementary features form input sequence by allowing the model to process information at different receptive fields simultaneously.

The input to this model is passed through multiple convolutional branches in parallel with different kernel sizes. Outputs of these branches are concatenated and fed into a common dense layer, and then followed by a single-unit output layer for regression. Through this fusion strategy, the network learns multi-resolution patterns without depending on recurrent or attention-based mechanism.

To optimize this architecture, Grid Search was used to tune some key hyperparameters specific to this multi-scale design. The number of filters in each convolutional branch, dropout rate, and dense layer size were changed across a predefined search space. The final model consisted of 64 filters, dropout rate of 0.2, and 100 dense units is turned out to be the best configuration. The search space was guided by commonly used values in other CNN-based RUL prediction studies and was kept limited in the consideration of computational resource constraints. Despite the limited search space, this approach ensured the model remained efficient and capable to learn meaningful representations at different temporal resolutions.

To guarantee a fair comparison, all other settings, such as preprocessing, training procedures and learning rate, were kept consistent with the baseline CNN. Table 4 gives a summary of selected values and search ranges for this architecture.

Hyperparameter	Selected Value	Grid Search Options	Remarks
Convolutional Filters (per branch)	64	32, 64	Tuned via Grid Search
Dropout Rate	0.2	0.2, 0.3	Tuned via Grid Search
Dense Layer Units	100	64, 100	Tuned via Grid Search
Kernel Sizes (fixed)	3, 5, 7	Fixed	Fixed across branches
Learning Rate	0.001	0.001, 0.0005	Inherited from Baseline CNN

Table 4: Selected Hyperparameters for Multi-Scale CNN Model

3.4.4 CNN + Autoencoder

CNN + Autoencoder architecture integrates unsupervised feature compression with supervised RUL prediction by combining Time Distributed Autoencoder with convolutional layers. This architecture aims to reduce input dimensionality, denoise the signal, and extract latent temporal representations before applying CNN-based regression. This fusion strategy can facilitate better generalization and robustness in RUL prediction particularly when dealing with high-dimensional multivariate sensor data.

At first, the input time-series window is passed through Autoencoder implemented with Time Distributed fully connected layers. The encoder compresses each time step into a lower dimensional latent representation, which is then run through a series of 1D convolutional layers. Only during training. the decoder part of the Autoencoder is used to minimize reconstruction loss while only the encoder output is retained for downstream prediction. The extracted features are flattened after the convolutional processing, and then passed through a dense layer and a final single-unit output layer.

To optimize key hyperparameters such as latent encoding dimension, dropout rate, and the number of dense units in the final regression layer, a dedicated Grid Search was conducted. The encoder and decoder of the Autoencoder were implemented using Time Distributed Dense layers in order to keep the temporal order of the input sequence. During training, the model jointly minimizes both reconstruction loss (for Autoencoder) and regression loss (for RUL prediction), though only the output of the encoder is used during inference.

This final configuration was selected by the Grid Search and it consisted of a latent dimension of 32, dropout rate of 0.3, and 100 units in the dense prediction layer. The search space was constrained by computational resource limitations and informed by values which are commonly used in prior studies for time-series prediction. Although the search space was limited, this strategy ensured the model efficiency and a balance between training cost and efficiency.

The rest of the parameters, such as learning rate, convolutional filter sizes and preprocessing steps, were kept identical with the optimized baseline CNN to provide consistency. The Table 5 gives a summary of hyperparameters values and its search range.

This hybrid model benefits from the ability of the Autoencoder to perform dimensionality reduction and noise filter, followed by CNN's strength in capturing short-term

temporal patterns. Together, all these components will enhance the model's ability to generalize across different degradation behaviors and sensor dynamics.

Hyperparameter	**Selected Value**	**Grid Search Options**	**Remarks**
Encoding Dimension (Latent Size)	16	8, 16, 32	Tuned via Grid Search
Convolutional Filters (Layer 1 & 2)	128, 128	64, 128 → 128; 128, 256 → 128	Inherited from Baseline CNN
Kernel Size	5	3, 5	Inherited from Baseline CNN
Dropout Rate	0.3	0.3, 0.5	Inherited from Baseline CNN
Dense Layer Units	128	100, 128	Inherited from Baseline CNN
Learning Rate	0.0005	0.001, 0.0005	Inherited from Baseline CNN

Table 5: Selected Hyperparameters for CNN + Autoencoder Model

3.5 Model Training Strategy

To guarantee reproducibility and fairness in comparing all model architectures, a standardised training and evaluation framework was implemented. All models were developed in Python using the TensorFlow/Keras deep learning framework, along with supporting libraries such as NumPy, Pandas, and scikit-learn.

These models were trained using the preprocessed CMAPSS, fix-length sliding window sequences as the input and corresponding RUL targets. The primary training objective was to minimize prediction error of RUL estimation. To achieve this, the Mean Squared Error (MSE) loss function was applied due to its suitability for continuous regression tasks. And, Adam optimiser was used due to the adaptive learning rate capabilities. Learning rates were determined using Grid Search and they remained unchanged in during the final training.

During the Grid Search phase, early stopping was implemented in order to prevent over-fitting and to reduce redundant computation when evaluating hyperparameter configurations. However, early stopping was not applied in final model training. Instead, all the models were trained for a fixed 50 epochs to ensure identical training conditions and to eliminate any bias due to different training periods. This consistent epoch setting allowed performance differences to be attributed solely to the architectural variation rather than differences in convergence behavior.

Batch size of 64 was applied consistently across all experiments. The same random seed was fixed for dataset splitting and weight initialisation to ensure the reproducibility. All non-architectural parameters, such as preprocessing steps, training hyperparameters and data division splits, were kept identical to maintain fairness across models. Architectural variations and their respective hyperparameters, determined via Grid Search, were the only thing that are allowed to vary between models.

3.6 Result Evaluation Strategy

The evaluation of all models in this study considered both predictive accuracy and computational complexity. It will provide a balanced assessment of both performance and practicality. In this way, the comparison between the architectures did not only involve their ability to make accurate predictions, but also their computational demands.

Predictive evaluation was assessed on a held-out validation set, which used three standard regression metrics to attest their accuracy. The metrics were chosen to reflect the complementary aspects of model performance, its overall accuracy as well as sensitivity to larger errors, and the proportion of variance in RUL prediction explained by the model. All metrics were calculated with scikit-learn's regression evaluation functions and results were recorded after the fixed 30-epoch training schedule to maintain the evaluation consistency in all models.

Besides accuracy, computational complexity was also analyzed to understand the trade-offs between performance and efficiency of the models. This included comparing the number of trainable parameters, model depth and total training time. These indicators provided an insight into the resource requirements demanded by each architecture, and also their suitability for large scale or time sensitive applications.

To complement the quantitative evaluation, visual diagnostic methods were also employed in order to have better insights into model behaviour. These included residual error distributions and actual vs predicted scatter plots. Such visualization supported the interpreting the numerical results and provided further context for accessing each model's generalization capability. This integrated evaluation framework ensured that the comparative analysis was both comprehensive and methodologically rigorous, enabling well supported conclusions on the relative merits of each architecture.

Besides technical evaluation, interpretability and ethical considerations are the key to successful implementation of predictive maintenance systems. By improving model transparency with feature attribution or attention-weight visualization, domain experts may better appreciate and understand the factors that guide RUL predictions, which can boost confidence in model outputs. There are also ethical concerns to put into consideration, such as the secure handling of sensitive sensor data information, the accountability of inaccurate predictions and the potential workforce implications of AI-driven automation. By addressing these factors, the models would not only be able to improve the confidence of the models but also be adapted in the real-world industrial settings where the reliability and transparency will be very crucial.

4 Evaluation & Results

This section presents the findings of the comparative analysis between the baseline CNN with the three proposed CNN-based fusion architectures. The evaluation framed, explained in Section 3.6, incorporates both predictive performance metrics and model complexity indicators to give a balanced assessment of each architecture's effectiveness and suitability for RUL prediction in the predictive maintenance applications.

4.1 Model Performance Analysis

The predictive accuracy of each architecture was evaluated on the validation set using three standard regressions metrics as shown in Table 6. All those metrics collectively capture various aspects of prediction quality, allowing a more comprehensive model comparison.

Model	**MAE**	**RMSE**	R^2 **Score**
Baseline CNN	30.9578	45.4137	0.4609
CNN + Transformer	28.0855	35.6437	0.6679
Multi-Scale CNN	22.7333	32.0161	0.7321
CNN + Autoencoder	19.8020	30.2217	0.7613

Table 6: Performance Comparison of All Models

The baseline CNN has MAE of 30.9578, RMSE of 45.4137, and R^2 of 0.4609, providing the reference benchmark for comparison. Each of the three fusion based models delivered a noticeable improvement over the baseline across all metrics, which confirms the advantage of augmenting CNN architectures with additional feature extraction and representation learning mechanisms.

In percentage terms, the CNN + Transformer Encoder lowered the MAE by 9.29% and RMSE by 21.53% compared to the baseline, and increased the R^2 coefficient from 0.4609 to 0.6679. The Multi-Scale CNN provided greater improvement, reporting a 26.60% reduction in MAE and a 29.53% decrease in RMSE, with a corresponding R^2 of 0.7321. The CNN + Autoencoder attained the most substantial improvement, reducing MAE by 36.06% and RMSE by 33.47%, while returning the highest R^2 score of 0.7613.

These findings demonstrate that the integration of the fusion mechanisms into CNN architectures can significantly enhance the ability of RUL prediction. The degree of improvement can differ in fusion strategies, and the autoencoder based approach yielding the most consistent and best performance gains.

4.2 Model Complexity Analysis

In addition to predictive performance, model complexity was valuated in order to estimate the computational cost of with each architecture. The total number of trainable parameters, architectural depth, and total training time were chosen as the complexity indicators. These results are presented in Table 7.

Model	**Trainable Params**	**Depth (Layers)**	**Training Time (s)**
CNN Baseline	159,489	9	189.0768
CNN + Transformer	242,817	20	440.4743
Multi-Scale CNN	181,129	15	201.5635
CNN + Autoencoder	183,353	11	193.0066

Table 7: Model Complexity Comparison

Among the fusion architectures, Multi-Scale CNN was the most computationally efficient and needs the smallest parameter count and developed in the shortest training time, while still providing significant accuracy gains. The CNN + Autoencoder, although it had a higher parameter count than the baseline, the model had a reasonable training time when compared to its accuracy improvement. Conversely, the CNN + Transformer Encoder was the most computationally expensive with the highest parameter count, deepest architecture, and longest training time, reflecting the overhead introduced by the multi head attention mechanism.

4.3 Visualization Analysis

Besides the numerical evaluation presented earlier, visual assessments were used to further determine the predictive behaviour and learning attributes of the assessed models. The predicted versus actual RUL scatter plots Figure 3 shows both CNN + Transformer and Multi-Scale CNN achieve tighter clustering along the diagonal compared to the baseline CNN model. CNN + Autoencoder demonstrates closest alignment proving the more accurate predictive accuracy as discussed earlier. Residual error distributions, Figure 4 further supports these observations. CNN + Autoencoder demonstrated the narrowest and the most centered distribution compared to other models, indicating balanced and low magnitude prediction errors.

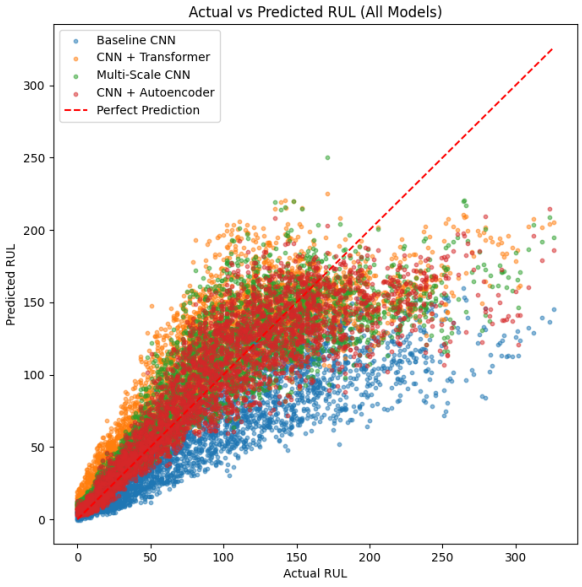


Figure 3: Comparison between Predicted vs Actual RUL values for all models.

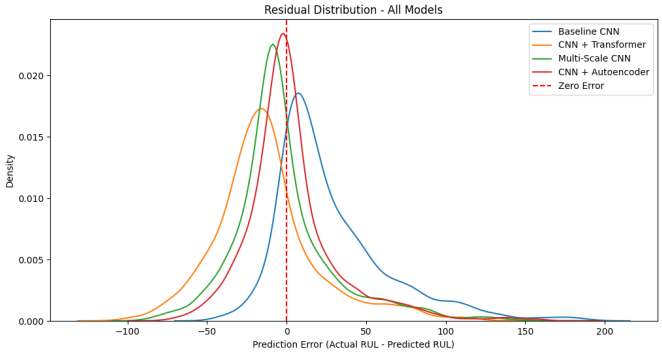


Figure 4: Residual error distribution for all models.

The convergence patterns of training and validation loss across all models (Figure 5) show that all architectures converge rapidly within just the initial five epochs, with CNN + Autoencoder displaying the lowest final loss values, reflecting strong generalization capacity. The similar trend is observed for the training and validation MAE curves (Figure 6). CNN + Autoencoder achieve the lowest validation MAE values and also maintaining consistently lower errors during training. But CNN + Transformer and the Baseline model exhibit higher final loss values and greater variability in their validation curves, suggesting less consistent generalisation.

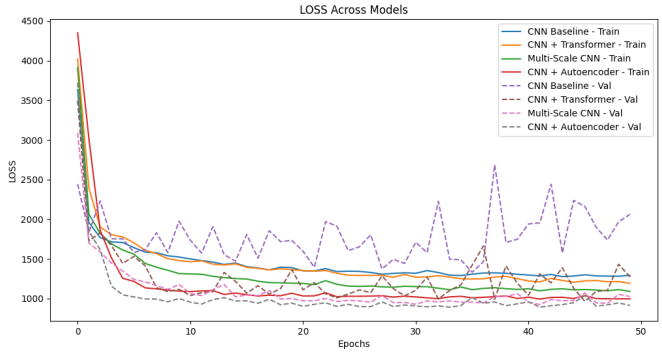


Figure 5: Training and validation loss curves for all models.

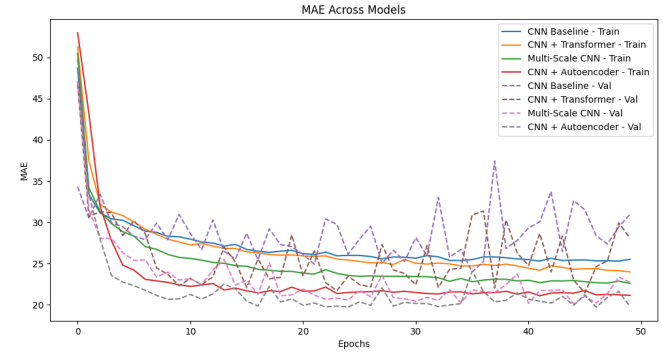


Figure 6: Training and validation MAE curves for all models.

In general, the visualization-based analysis supported the results of Sections 4.1 and 4.2 by verifying that the fusion-based models consistently outperform to the baseline CNN in all aspects related to the predictive alignment, error distribution, and convergence stability, with CNN + Autoencoder exhibiting the most accurate and robust performance in all evaluated dimensions.

4.4 Comparative Analysis and Interpretation of Results

The comparative evaluation between the baseline CNN and three CNN-based fusion architectures has shown that incorporation additional feature extraction and representation learning can significantly improve performance of RUL prediction. The numerical measures in Section 4.1 reveal that all fusion architecture outperformed the baseline CNN in all the metrics. Among these, CNN + Autoencoder showed the highest accuracy. This improvement attributed to the ability of the Autoencoder to reduce noise and compress high-dimensional sensor data, allowing to focus on the most informative features.

Multi-Scale CNN delivered competitive results and it also pointed out a good compromise between predictive performance and computational efficiency. The model successfully captured the short-term and long-term degradation patterns by extracting features at multiple temporal resolutions. Additionally, it has the smallest parameter count and shortest training time, which is beneficial in cases with limited computing resources.

CNN + Transformer Encoder has shown improved long-range temporal modeling through its self-attention mechanism. The model also outperforms the baseline model in all evaluated metrics. However, this improvement came at the cost of significantly higher computational complexity and longer training time. This trade-off makes it a less optimal solution in a latency-sensitive or resource-constrained environments.

As shown in Table 8, performance, complexity, and key findings for each model are summarized in order to provide a clear comparison of trade-offs. This comparative perspective reinforces the importance of considering both predictive accuracy and computational feasibility when choosing architectures to any practical predictive maintenance applications.

Model	**MAE** $\downarrow$	**RMSE** $\downarrow$	$R^2 \uparrow$	**Parameters**	**Training Time (s)**	**Key Observations**
CNN Baseline	30.9578	45.4137	0.4609	159,489	189.08	Reference benchmark
CNN + Transformer	28.0855	35.6437	0.6679	242,817	440.47	Improved long-range temporal modeling; highest computational cost
Multi-Scale CNN	22.7333	32.0161	0.7321	181,129	201.56	Strong accuracy with moderate complexity; efficient training
CNN + Autoencoder	19.8020	30.2217	0.7613	183,353	193.01	Best overall accuracy; balanced performance and complexity

Table 8: Performance, complexity, and observations for baseline CNN and proposed fusion models.

The findings are also consistent with broader findings in the predictive maintenance

literature, where the improvement of predictive accuracy often come with the cost of increased computational complexity. The superior performance of the CNN + Autoencoder aligns with studies emphasizing the value of denoising and feature compression, while the efficiency of the Multi-Scale CNN reflects prior work advocating for architectures that balance performance and scalability in industrial contexts. Similarly, the trade-offs found with the CNN + Transformer correspond to the challenges reported in deploying attention-based systems in real-time or resource-constrained environments. By situating the comparative results of this within these wider discussion, the results of the study do not merely confirm prior knowledge, but also give useful guidelines on the choice of architectures that most effectively match the computational and operational realities of Industry 4.0 implementation.

5 Conclusion and Future Work

This study examined the impact of three fusion techniques of CNN-based models: CNN + Transformer Encoder, Multi-Scale CNN Fusion, and CNN + Autoencoder on RUL predictions. The paper explored these fusion architectures in a controlled experimental process against the baseline standalone CNN and evaluated predictive accuracy, computational complexity and convergence behaviour.

This research was guided by the question:

"What is the impact of different fusion strategies for CNN-based models (CNN + Transformer Encoder, Multi-Scale CNN Fusion, and CNN + Autoencoder) on predictive maintenance performance compared to standalone CNN models?"

The findings of this study provide a comprehensive answer to this question by highlighting the trade-offs among accuracy, efficiency, and complexity for each fusion approach.

The outcomes established that fusion-based enhancements can substantially improve CNN performance to predict RUL. All three fusion architectures performed better than the baseline CNN across all key performance metrics. CNN + Autoencoder achieved the best predictive accuracy, benefiting from its ability to denoise high-dimensional sensor data prior to convolution feature extraction. The Multi-Scale CNN delivered good trade-off between efficiency and accuracy at lowest computational cost and shortest training time. The CNN + Transformer Encoder improved long-range temporal modelling capabilities through self-attention mechanisms but the model came with the highest computational cost that may limit its use in latency-sensitive or resource-constrained environments.

These findings carry significant implications both to researchers and practitioners in the field of predictive maintenance. This study enables making more informed decision in choosing the architectures based on their deployment constraints by identifying the strengths and trade-offs of the various fusion strategies. The insights from this comparative analysis also provides practical guidance on choosing the right model architecture depending on the trade-off between prediction accuracy, computation

efficiency, and possibility of deployment in real-world applications.

Besides emphasizing the advantages of each fusion strategy, it is also important to consider how these models could be scaled and implemented in the real-life industrial setting. Although the proposed CNN fusion architectures show high predictive performance under the controlled experimental settings, scalability and implementation in the real industrial settings need additional consideration. Industrial environments often involve heterogeneous sensor networks, variable data quality, and constraints imposed by edge devices with limited computing capabilities. The implementation of these models at scale therefore requires solutions including model compression, distributed training, and optimization for real-time inference. It is important to address these factors to ensure that the demonstrated improvements translate into robust, efficient and cost-efficient solutions to Industry 4.0 large-scale predictive maintenance.

5.1 Limitations

Despite the promising findings, a number of limitations need to be acknowledged. This evaluation was done with respect to NASA CMAPSS dataset which simulates turbofan engine degradation. This dataset provides a controlled and widely recognised benchmark, but the outcome might not completely generalise to more complicated operational. In real-world industrial scenarios, the environmental variance, various fault modes and diverse operating conditions can substantially influence the predictive performance. But those may not be adequately represented in a simulated dataset.

Moreover, this research was conducted on three specific CNN-based fusion architectures. This may leave other potentially valuable hybrid designs unexplored. Other architectures like CNN – GNN (Graph Neural Network) hybrids or other adaptive attention-based architectures may offer complementary strengths, and they could potentially provide further improvements in RUL prediction accuracy and robustness.

Lastly, the evaluation has been carried out using a fixed set of hyperparameters which are determined through grid search for each model. Although this approach provided fair and consistent comparison between architectures, it may not represent the most optimal configuration for each model. There could be alternative tuning strategies which lead to further performance improvements.

5.2 Future Work

With reference to the findings of this research, there are several directions to be pursued for future research. One of them would be to validate the proposed fusion architectures on a wider range of datasets, especially those based on real industrial settings. Although the NASA CMAPSS dataset is a well-controlled benchmark for RUL predication, the real industrial data can be more challenging due to inconsistent sampling rate, missing or incomplete readings and measurement noise generated by extreme operating conditions. It would be essential to adapt the current fusion architecture to cope with these complexities.

Future research could also explore alternative fusion strategies, for example, integrating CNNs with Graph Neural Networks to better capture relational dependencies between sensors. In addition, the use of temporal convolutional transformer hybrids could be considered to model multi-scale patterns and long-range dependencies. These approaches may contribute towards the advancement of temporal modelling capabilities of fusion-based RUL prediction frameworks.

Another future opportunity is the optimisation of these models for deployment in resource limited or real-time industrial settings. Predictive maintenance systems often operate on embedded devices or edge computing systems, where computational resources, memory capacity and power usage are limited. The research of existing compression and acceleration methods, as quantisation, pruning, or knowledge distillation, could significantly reduce computational demands without compromising predictive power.

By doing those, future studies can further improve the balance between the predictive accuracy, computational efficiency and robustness. Achieving this balance will be essential to the practical implementation of the deep learning-based predictive maintenance solutions in diverse and dynamic industrial environments.

References

Abdelli, K., Grieser, H. and Pachnicke, S. (2022). A machine learning-based framework for predictive maintenance of semiconductor laser for optical communication, *Journal of Lightwave Technology* **40**(14): 4698–4708.
URL: *http://dx.doi.org/10.1109/JLT.2022.3163579*

Bousdekis, A., Magoutas, B., Apostolou, D. and Mentzas, G. (2018). Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance, *Journal of Intelligent Manufacturing* **29**(6): 1303–1316.

Cheng, C., Ma, G., Zhang, Y., Sun, M., Teng, F., Ding, H. and Yuan, Y. (2020). A deep learning-based remaining useful life prediction approach for bearings, *IEEE/ASME Transactions on Mechatronics* **25**(3): 1243–1254.
URL: *http://dx.doi.org/10.1109/TMECH.2020.2971503*

Du, N. H., Long, N. H., Ha, K. N., Hoang, N. V., Huong, T. T. and Tran, K. P. (2023). Trans-lighter: A light-weight federated learning-based architecture for remaining useful lifetime prediction, *Computers in Industry* **148**: 103888.
URL: *https://www.sciencedirect.com/science/article/pii/S0166361523000386*

Elizar, E., Zulkifley, M. A., Muharar, R., Zaman, M. H. M. and Mustaza, S. M. (2022). A review on multiscale-deep-learning applications, *Sensors* **22**(19).
URL: *https://www.mdpi.com/1424-8220/22/19/7384*

Fathi, K., van de Venn, H. W. and Honegger, M. (2021). Predictive maintenance: An autoencoder anomaly-based approach for a 3 dof delta robot, *Sensors* **21**(21).
URL: *https://www.mdpi.com/1424-8220/21/21/6979*

Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W.-J. and Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications, *Engineering Applications of Artificial Intelligence* **92**: 103678.
URL: *https://www.sciencedirect.com/science/article/pii/S0952197620301184*

Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J., Kim, B. H. and Do Noh, S. (2016). Smart manufacturing: Past research, present findings, and future directions, *International Journal of Precision Engineering and Manufacturing-Green Technology* **3**(1): 111–128.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning, *Nature* **521**(7553): 436–444.

Lee, J., Bagheri, B. and Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems, *Manufacturing Letters* **3**: 18–23.
URL: *https://www.sciencedirect.com/science/article/pii/S221384631400025X*

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L. and Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—reviews, methodology and applications, *Mechanical Systems and Signal Processing* **42**(1): 314–334.
URL: *https://www.sciencedirect.com/science/article/pii/S0888327013002860*

Lei, Y., Li, N., Guo, L., Li, N., Yan, T. and Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to rul prediction, *Mechanical Systems and Signal Processing* **104**: 799–834.
URL: *https://www.sciencedirect.com/science/article/pii/S0888327017305988*

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N. and Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap, *Mechanical Systems and Signal Processing* **138**: 106587.
URL: *https://www.sciencedirect.com/science/article/pii/S0888327019308088*

Li, X., Ding, Q. and Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks, *Reliability Engineering System Safety* **172**: 1–11.
URL: *https://www.sciencedirect.com/science/article/pii/S0951832017307779*

Qin, J., Liu, Y. and Grosvenor, R. (2016). A categorical framework of manufacturing for industry 4.0 and beyond, *Procedia CIRP* **52**: 173–178. The Sixth International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2016).
URL: *https://www.sciencedirect.com/science/article/pii/S221282711630854X*

Qin, Y., Cai, N., Gao, C., Zhang, Y., Cheng, Y. and Chen, X. (2022). Remaining useful life prediction using temporal deep degradation network for complex machinery with attention-based feature extraction.
URL: *https://arxiv.org/abs/2202.10916*

Saxena, A., Goebel, K., Simon, D. and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation, *2008 International Conference on Prognostics and Health Management*, pp. 1–9.

Si, X.-S., Wang, W., Hu, C.-H. and Zhou, D.-H. (2011). Remaining useful life estimation – a review on the statistical data driven approaches, *European Journal of Operational Research* **213**(1): 1–14.
URL: *https://www.sciencedirect.com/science/article/pii/S0377221710007903*

Wang, B., Lei, Y., Li, N. and Wang, W. (2021). Multiscale convolutional attention network for predicting remaining useful life of machinery, *IEEE Transactions on Industrial Electronics* **68**(8): 7496–7504.

Wang, H., Zhang, W., Yang, D. and Xiang, Y. (2023). Deep-learning-enabled predictive maintenance in industrial internet of things: Methods, applications, and challenges, *IEEE Systems Journal* **17**(2): 2602–2615.

Zhang, W., Yang, D. and Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey, *IEEE Systems Journal* **13**(3): 2213–2227.

Zhang, Z., Song, W. and Li, Q. (2022). Dual-aspect self-attention based on transformer for remaining useful life prediction, *IEEE Transactions on Instrumentation and Measurement* **71**: 1–11.
URL: *http://dx.doi.org/10.1109/TIM.2022.3160561*

Zhou, L. and Wang, H. (2024). An adaptive multi-scale feature fusion and adaptive mixture-of-experts multi-task model for industrial equipment health status assessment and remaining useful life prediction, *Reliability Engineering System Safety* **248**: 110190.
URL: *https://www.sciencedirect.com/science/article/pii/S0951832024002631*