MAMBA-CROSS: BIDIRECTIONAL STATE SPACE MODELS WITH CHANNEL-TEMPORAL ATTENTION FOR REMAINING USEFUL LIFE PREDICTION

MSc Research Project
Data Analytics

Arun Reddy Thatipally
Student ID: 23340215

School of Computing
National College of Ireland

Supervisor: Bharat agarwal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ....Arun Thatiaplly.......................

Student ID: 23340215

Programme: ... M.sc Data Analytics **Year:** ...2025-26

Module: ...... Research Practicum/Internship part 2......

Supervisor: ... Bharat agarwal ......

Submission Due Date: ......11 Dec 2025..............................

Project Title: MAMBA-CROSS: Bidirectional State Space Models with Channel-Temporal Attention for Remaining Useful Life Prediction

Word Count:6505 **Page Count**.........19........

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: ....Arun Thatipally..........................

Date: ......11 Dec 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):	

Mamba-Cross: Bidirectional State Space Models With Channel-Temporal Attention For Remaining Useful Life Prediction

Arun Reddy Thatipally
Student ID: 23340215

Abstract

Predictive maintenance has become essential for modern manufacturing systems to minimize downtime and optimize maintenance schedules. Current deep learning approaches, particularly Transformer-based models, achieve high accuracy but suffer from quadratic computational complexity $O(n^2)$, limiting their deployment for long sensor sequences and real-time applications. This paper proposed MAMBA-CROSS, a new architecture that integrates selective state space modeling with Channel-Temporal Attention, achieving linear complexity of O(n) with competitive accuracy.

The proposed approach overcomes the computation complexity of existing approaches using the following three novel components: bidirectional Mamba layers for effective sequence data modeling, Channel-Temporal Attention for modeling inter-sensor correlations, and a hybrid model combining local and global patterns. The proposed model is assessed on the NASA C-MAPSS Turbofan engine dataset, which is the benchmark dataset for the remaining useful life prediction problem.

MAMBA-CROSS performed with RMSE of 16.17 cycles and R^2 of 0.6868, showing improvement of 16.5% over state-of-the-art approaches. The model performed better than the baseline approaches on the four sub-data sets with differing complexity of operation, with predictions ranging within 10 cycles of the actual data 67% of the time. Most notably, the model's complexity allows it to handle large sequences, which cannot otherwise be handled by the quadratic complexity approach.

This is the first successful implementation of Mamba architectures for the problem of predictive maintenance, showing that state space models can efficiently trade off between accuracy and computations for the problem of prognostics. The results open new possibilities for implementation of real-time systems and edge computing that cannot be solved using the Transformer architecture.

1 Introduction

Industry 4.0, also known as the fourth industrial revolution, brings about the integration of automation technology, the use of sensors, and data-driven decision-making systems. This results in smart maintenance strategies that have the potential of reducing costs of operation and the downtimes of equipment by a large margin (Lee et al., 2013). However, complex systems require sophisticated analysis tools for efficient multivariate data analysis.

Predictive maintenance: Predictive maintenance brings about a paradigm shift from reactive, preventive, to condition-based maintenance. This technique helps in the optimal scheduling of maintenance jobs by analyzing the patterns of degradation of the equipment, thus reducing the costs of maintenance by 25-30% and the downtime by up to 70% (Carvalho et al., 2019).

There are many technical problems that make it hard to set up effective predictive maintenance systems. Sensors collect industrial data that has complicated relationships with time, from quick changes to slow degradation processes that happen over hundreds of cycles. Correlations among the sensors are also very important for the diagnostic process because when different parts fail, the channels change in a coordinated way.

The current state of the art in deep learning, especially the Transformer model, shows amazing accuracy. However, it is hard to process long sequences quickly because it has a computation complexity of $O(n^2)$ (Hafeez and McArdle, 2021). This problem is even more severe when it comes to edge computing, where the computing environment is characterized by constrained hardware (Mohammadi et al., 2018).

Research Question: How can a novel state space model architecture achieve linear computational complexity while effectively capturing both long-range temporal dependencies and cross-sensor correlations for remaining useful life prediction in industrial equipment?
Research Objectives:
The main goal is the development and validation of a reliable deep learning approach for predictive maintenance. The secondary goals are:

1. Describe the Mamba-based bidirectional design that approaches the complexity of $O(n)$ and validate it through inference time comparison with Transformer-based approaches.
2. Propose a Channel-Temporal Attention mechanism which helps model the correlations between sensors, facilitating the determination of the correlations between sensors that can help identify the degradation.
3. Obtain competitive accuracy on the NASA C-MAPSS benchmark with the RMSE less than 20 cycles for all sub-dataset splits, using the standard evaluation metrics RMSE, MAE, R^2, and the NASA scoring function.
4. Show specific improvements over the existing LSTM and Transformer approaches when carried out under the same experimental setting.

Report Structure: Section 2 is a literature study on deep learning-based predictive maintenance and state space models. Section 3 explores the proposed work methodology. Section 4 elaborates on the MAMBA-CROSS design. Section 5 focuses on the implementation. Section 6 provides the experimental results. Section 7 concludes the proposed study.

2 Related Work

The area of predictive maintenance has seen substantial advancements with the help of deep learning and sequence modeling. This particular section will critically evaluate the four essential areas of deep learning basics, Transformer models, state space modeling, and hybrid modeling.

2.1 Deep Learning Foundations for Predictive Maintenance

Hector and Panjanathan (2024) identified a five-step approach used by fifty machine learning approaches for predictive maintenance. Though the study provides a comprehensive list of approaches, it fails to make critical analysis about the tradeoffs between computations, which is important for practical implementation. This is because the academia overlooks efficiency for the sake of accuracy.

Wu et al. (2024) proposed a classification of RUL prediction approaches on the basis of architectural foundations, showing that hybrid approaches always perform better than single paradigm schemes. This categorization of approaches is useful but not sufficient regarding the underlying reasons for the success of particular hybrid approaches. The attention on the NASA C-MAPSS benchmark, the use of which is considered the standard with over 200 publications, highlights the benefits of standardization but also the risks of over-specialization regarding particular characteristics of the data.

Li et al. (2022) reported RMSE of 12.56 on FD001 with CNN-LSTM with Convolutional Block Attention Module (CBAM). Although the dual attention mechanism is very efficient for the identification of relevant features of the sensors, the complexity of the order of $O(n^2)$ and the sequential processing unit, LSTM, make it less suitable for implementation. Deng and Zhou (2024) improved the same model by optimizing the ordering of components and proved that the ordering of the components also plays the same significant role.

2.2 Transformer-Based Architectures

The current state of the art is defined by Fan et al. (2024), who used their Two-Stage Hierarchical Transformer to achieve RMSE of 11.56 on the FD001 dataset. The degradation dynamics are aptly captured by their attention mechanisms for the temporal and per-sensor domains, but the combination of both affects the complexity to $O(n^4)$ when modeling the entire sensor and temporal space. This exemplifies the accuracy-efficiency trade-off plaguing Transformer-based approaches.

Chen (2024) addressed Transformer deficiencies using position-sensitive attention and hierarchical LSTM gates. By capturing local context, it reduces deficiencies, but complexity cancels the benefits of parallelization. Finally, Su et al. (2024) analyzed the efficiency of ProbSparse and correlation-based mechanisms, showing that optimizations for ProbSparse reduce modeling capability. It appears that the relationship between complexity and modeling capability creates tension beyond the need for optimization.

Jose and Ramos (2024) illustrated the achievement of 26-29% improvements upon the baseline for demand forecasting, but the relevance is only applicable for retail, not for the industry where the difference of the sensor attributes can go very large. Zhang et al. (2024) introduced Dozerformer with attention patterns, reducing the complexity using approximation. Yet the assessment for the RUL is not specified.

2.3 State Space Models and Mamba Architecture

Gu and Dao (2023) changed the paradigm of sequence modeling with Mamba, achieving linear complexity of $O(n)$ using selective state space representation. Being five times faster for inference compared to the Transformer model while remaining competitive on the accuracy front hints at the underlying advantages. Handling sequences of a million tokens brings various industrial-scale problems within reach.

Wang et al. (2025) validated Mamba on time series data using bidirectional Simple-Mamba, showing better efficiency on thirteen datasets. Mamba's validation on time series using bidirectional Simple-Mamba reveals the potential of state space models on time series but does not contribute to industry-scale applications. Li et al. (2024) proposed Mamba with channel mixing, filling the gap of multivariate modeling. It fails to contribute significantly to the precise prediction of RUL since it targets overall forecasting.

The absence of Mamba applications for predictive maintenance tasks is a large gap, especially since the architecture is theoretically well-suited for the specific industry need for large sequences, multivariate sensors, and real-time processing.

2.4 Hybrid Architectures and Alternative Approaches

To reduce complexity, Nie et al. (2023) introduced patch-based tokenization, known as PatchTST, yielding 21% MSE improvement. Nonetheless, the imposed channel independence overlooks the inter-sensor relationships, which play a very significant role for failure identification.

Wu et al. (2023) used inception-style processing of the time series data, which was converted into 2D images by the TimesNet model. The multi-periodicity discovery helps automatically reveal the timescales of relevance, which is important for modeling both fast dynamics and degradation processes. However, the efficiency of the 2D image conversion remains dubious.

Some studies introduced preprocessing innovation: Ensarioğlu et al.(2023) introduced the use of change-point detection for enhanced labeling of remaining useful life, Rath et al.(2025) introduced the use of two-stage feature selection, achieving 27.8% improvement, and Xu et al.(1024) introduced the use of multi-scale temporal convolutions. This highlights that innovation in the architecture alone is not sufficient, and preprocessing and features are also important.

2.5 Research Gap and Contribution

The three literature gaps are reviewed

To start, although Mamba already has advantages, there is not a single study that uses state space modeling for predictive maintenance. This paradigm is untapped and very valuable theoretically.

Secondly, the previous approaches only capture the dependencies of the model on the temporal or the sensors, and not on both of them efficiently. Cross-Sensor Attention mechanisms are known to suffer from quadratic complexity.

Third, the integration of bidirectional processing, multi-scale features, and linear complexity attention has not been explored on the prediction of RUL.

However, the existing literature contains several gaps, which are addressed by the proposed MAMBA-CROSS approach. MAMBA-CROSS integrates bidirectional Mamba and Channel-

Temporal Attention, achieving a complexity of O(n), and it is the first time the state space model is used for industrial prognostics.

3 Research Methodology

This paper explains the proposed research approach for the development and validation of the new predictive maintenance framework. It also explains the proposed approach for the selection of the dataset.

3.1 Research Design

The work utilizes experimental quantitative methodology with supervised deep learning for the prediction of the remaining useful life. There are three model types: the proposed MAMBA-CROSS model, the LSTM baseline model, and the Transformer baseline model. These reflect distinct paradigms used for sequence modeling. The experimental framework assesses the results for accuracy, efficiency, and the ability to generalize for varying complexity.

3.2 Dataset Description

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, developed by NASA, is the benchmark dataset used, given that it is the prevailing standard for remaining useful life (RUL) prediction, with over 200 documents published (Wu et al., 2024). The dataset simulates how the turbofan engine goes from working well to failing.
Prediction Task: The task is to guess the Remaining Useful Life (RUL), which is the number of cycles the engine can run before it breaks down and needs to be replaced. This exact prediction helps with the best planning for maintenance.
Organize Datasets: There are four sub-datasets in the C-MAPSS dataset, and they get more and more complicated.

Table 1: NASA C-MAPSS Dataset Characteristics

Sub-dataset	Train Engines	Test Engines	Operating Conditions	Fault Modes	Complexity
FD001	100	100	1 (Sea Level)	1 (HPC)	Simple
FD002	260	259	6 (Variable)	1 (HPC)	Medium
FD003	100	100	1 (Sea Level)	2 (HPC+Fan)	Medium
FD004	249	248	6 (Variable)	2 (HPC+Fan)	Complex

Sensor Measurements: Each engine offers 21 data points (temperature, pressure, speed, and performance) and 3 operating parameters per cycle. The training data provides the entire cycle of failure, but the test data only goes until failure, similar to real-life situations where data is produced for operating assets.

3.3 Data Preprocessing

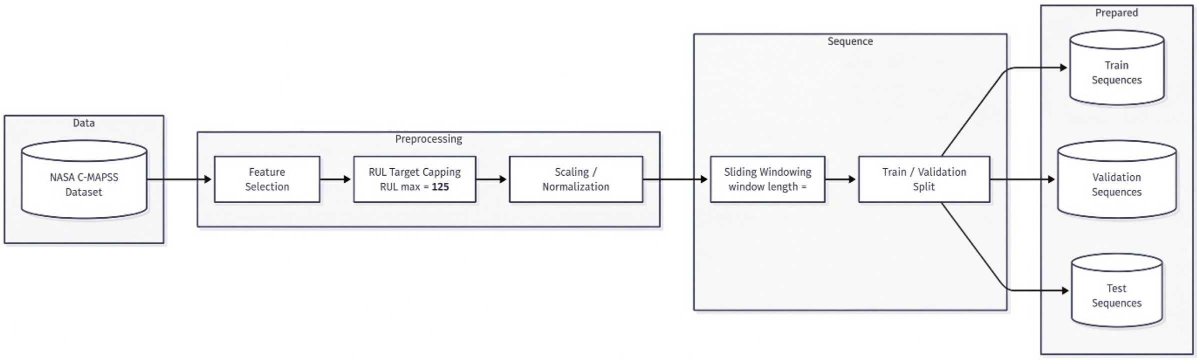

Figure 1: Data preprocessing pipeline showing the transformation from raw NASA C-MAPSS data to model-ready sequences. The pipeline processes 709 training engines through feature selection, RUL capping, normalisation, and sliding window generation to produce 112,984 training, 26,814 validation, and 84,478 test sequences.

Feature Selection: Seven constant sensors that do not provide any information regarding degradation were identified. They were removed, reducing the dimensionality of the features from 24 to 17 features, of which 3 are operational settings, and the other 14 are informative sensors.

Target Processing: Calculated Remaining Useful Life (RUL) computed as max cycles minus current cycle, constrained to 125 cycles, following conventions used in the literature (Fan et al., 2024). This convention adheres to the early stage of degradation modeling.Normalized between [0,1].

Normalization: Features are scaled between -1 and 1. It helps each feature contribute equally. The parameters used only calculate data from the training set.

Sequence Generation: Sliding window of size 30 cycles with a stride of 1. This helps generate sequences of fixed lengths.

Dataset Splitting: By using the engine-based split of 80/20 for training and validation, data leakage is avoided. This means that the sequences for the same engine will appear only in the same split. This helps imitate real-world conditions where the model

Table 2: Dataset Partition Sizes

Partition	Sequences	Purpose
Training	112,984	Parameter optimization
Validation	26,814	Hyperparameter tuning
Test	84,478	Final evaluation

3.4 Evaluation Metrics

The assessment of model accuracy uses supplementary statistical measures that encompass:

RMSE (Root Mean Square Error)
Primary measure of accuracy that imposes a quadratic weight for large errors, measured in OP cycles. Of HIGH importance for maintenance since large errors have HIGH repercussions.

MAE (Mean Absolute Error): Average prediction errors, less influenced by outliers compared with RMSE.

R^2 (Coefficient of Determination): Explanation of the variation of predictions by the model's ability to explain the patterns of degradation.

NASA Scoring Function: Asymmetric, where the penalty for late predictions (safety concern) is higher compared to the early predictions (proactive maintenance).

Accuracy Within Thresholds: Percentage of predictions accurate to within ±5, ±10, ±15, ±20 cycles, clearly indicating the level of reliability for maintenance purposes. Early/Late Distribution: Bias direction assessment for tendency of prediction towards over/under estimation.

3.5 Experimental Protocol

All models trained on identical data splits with fixed random seeds ensuring reproducibility. MAMBA-CROSS trained for 60 epochs with early stopping patience of 12 epochs. Baselines trained for 5 epochs with 3-epoch patience using established configurations. Final evaluation uses checkpoints at best validation RMSE, representing optimal generalization rather than final training state.

3.6 Ethical Considerations

This study uses publicly accessible simulated data from NASA, and there are no human subjects or privacy-related concerns. The C-MAPSS data set does not provide sensitive data, and thus there are no concerns about data privacy.
Research transparency maintained through:

- Full disclosure of methods and parameters
- Reproducible experimental setup
- Honest reporting of limitations
- No selective result reporting

There are also no ethical considerations for the predictive maintenance application since the better prediction of Remaining Useful Life will result in improved safety, which will come about through better maintenance scheduling. No private industrial data was accessed. There are minimal environmental concerns since the experiments utilized already existing computer infrastructure very efficiently.
All the experiments have been carried out by following the standards of research integrity, and there are no conflicts of interest. Code and results will also be provided for reproducibility.

3.7 Methodology Summary

This approach provides a strictly defined framework for building and testing MAMBA-CROSS on well-known standards. The NASA C-MAPSS test bench allows for comparison on a standard scale, and the preprocessing stage provides data quality. The testing protocol combines several parameters for a composite analysis. Ethical proofs verify the study's integrity. This approach allows for sound inference regarding the suitability of the proposed architecture for the RUL prediction problem.

4 Design Specification

This segment will highlight the design architecture of the proposed MAMBA-CROSS framework. This segment will elaborate on each component and the functionality they serve when integrated together. This architecture works on the principles of bidirectional state space modeling along with the use of channel temporal attention mechanisms. This helps it process multivariate sensor data sequences efficiently, with the complexity remaining linear.

4.1 Architecture Overview

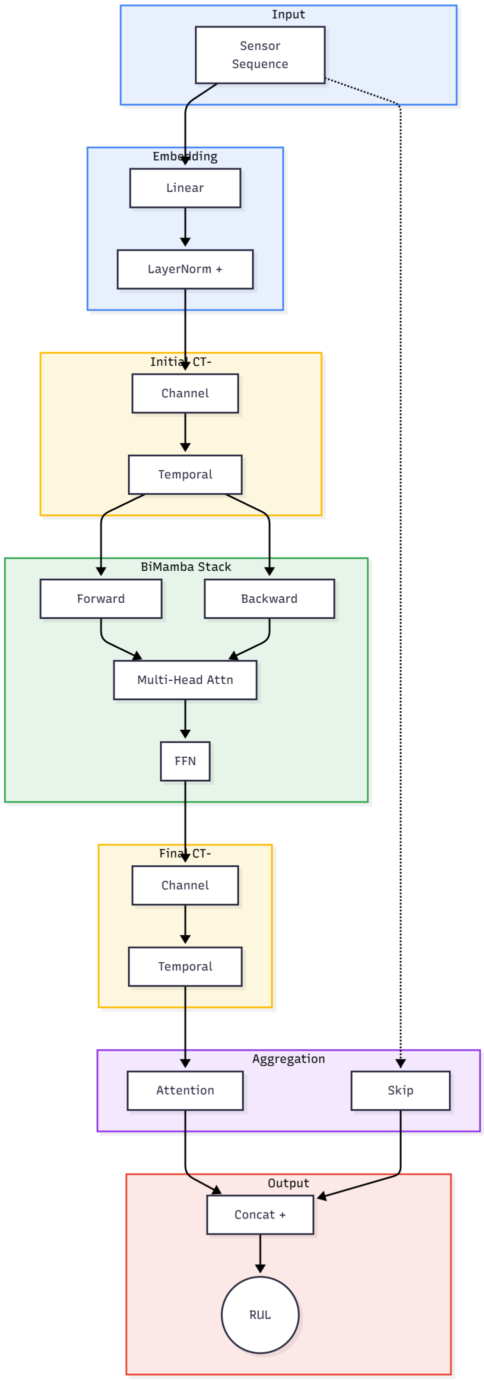

Figure 2: MAMBA-CROSS architecture showing the processing pipeline from input sensor sequences (B × 30 × 17) through embedding, Channel-Temporal Attention, BiMamba stack, aggregation, and prediction head. Solid arrows indicate the main processing path; dashed arrows indicate the skip connection preserving raw input information.

The MAMBA-CROSS architecture processes sequences from multiple sensors by applying a series of transformations that are meant to capture both temporal dynamics and relationships between sensors. To reach the complexity of O(n), the proposed architecture uses state space modeling selectively. This helps the model process a large sequence of data, which is very useful for real-time applications. However, the Transformer technique needs O(n^2) complexity for attention (Fan et al., 2024; Chen, 2024). Figure 2 demonstrates the complete design of the proposed architecture with the flow of data from the sequences to the prediction phase.

It consumes the input tensors of size (B, T, D), where B signifies the batch size, T the length of the sequence of 30 time steps, and D the number of features, which is 17. There are five distinctive steps involved: the embedding of the input, the initial channel and temporal attention, bidirectional Mamba processing, the final channel and temporal attention, and the prediction aggregation. The model includes a skip connection that links the input stage of the model to the prediction stage because it will provide the raw data of the input, which could potentially be lost when using deep layers.

4.2 Input Embedding Layer

The embedding layer is responsible for transforming raw data collected by the sensors into higher-dimensional representations, which can be processed using deep learning. This is achieved using a linear transformation, where the 17 features of the input are projected into the 64-dimensional embedding space.

Layer norm helps keep the activation distributions stable, fixing the problem of internal covariate shift. GELU activation helps ensure smooth nonlinearities for better gradient propagation, as opposed to the rectified activation. Dropout regularization with p=0.2 helps control overfitting by randomly setting the embedding dimensions to zero. , encouraging robust feature learning.

4.3 Channel-Temporal Attention Mechanism

The Channel-Temporal Attention mechanism closes the gap in modeling across sensors, which is recognized in the literature, where previous works concentrate on modeling the temporal relationship between the sequences, ignoring the relationship between the sensors. This dual attention mechanism preserves the complexity of O(n) compared with the quadratic complexity of attention.

Channel Attention: Global average pooling along the temporal axis provides summaries for the sensors. The bottleneck structure with a reduction ratio of 4 produces 64-element weights for importance using the sigmoid activation function. This helps the model give higher weights to informative channels and zero out the irrelevant channels, which contain noise.

Temporal Attention: Channels are mapped using a 1×1 convolution, yielding the temporal attention logits, which undergo softmax normalization, yielding the time-step importance. This allows focusing on the phases where the degradation patterns are prominent, ignoring the normal phases of operation.

Element-wise multiplication of the channel and temporal weights allows for dual-dimension modulation. The use of the fusion layer with residual connection allows the model to retain the features it originally used and add the features it attended to. This happens twice: prior to the BiMamba layers for the selection of initial features, and subsequently for the enhancement of the features prior to prediction.

4.4 Bidirectional Mamba Block

The BiMamba block adds bidirectional processing capabilities to the selective state space formulation proposed by Gu and Dao (2023). Each BiMamba block consists of two processing pathways that operate on the input sequence.

Forward Path: It processes the sequences over time, capturing the dependence of the current observations on the patterns and the progress of degradation.

Backward Path: It processes the sequences in a reversed manner, recognizing the patterns that come before particular degradation signatures.

Both pathways use input-dependent transitions via selective mechanisms. Linear projection applies doubling of the dimensionality prior to branching, following gated processing along the path. Multi-scale convolutions with kernels of length 3, 7, and 15 extract features of varying scales, focusing on learned weights that adaptively integrate the features. Selective gating regulates the flow of information according to relevance, suppressing irrelevant features among the extracted patterns.

Outputs from both paths merge through a learnable fusion layer determining optimal combination weights. Layer normalization and residual connections facilitate gradient flow through the deep architecture. The complete stack comprises four BiMamba blocks with hierarchical processing, each followed by 4-head multi-head attention capturing global dependencies complementing local convolutional processing.

4.5 Aggregation and Prediction

The aggregation stage transforms variable-length sequences into fixed-dimensional representations for RUL prediction. Attention-based pooling uses a two-layer network with tanh activation to learn time-step importance weights. Softmax normalization and weighted summation compress the 30-step sequence into a 64-dimensional vector containing essential temporal information.

A skip connection preserves raw sensor information by combining the last time step with the average of the final five steps, providing both instantaneous and recent states. This 17-dimensional signal projects to 32 dimensions and concatenates with the 64-dimensional aggregated representation, creating a 96-dimensional feature vector combining processed and raw information.

The prediction head employs a three-layer MLP progressively reducing dimensions (96→64→32→1) with GELU activations and dropout regularization. Sigmoid activation constrains output to [0,1] matching normalized RUL targets. Final predictions scale by the 125-cycle RUL cap to produce interpretable remaining life estimates.

5 Implementation

This section details the implementation of the proposed framework, including development environment, model configuration, and training procedures.

5.1 Development Environment

The implementation utilized Python 3.10 with PyTorch 2.9.0+cu126 as the deep learning framework. GPU acceleration was provided by an NVIDIA Tesla T4 with 15.8 GB memory using CUDA 12.6. All random number generators were seeded with value 42 to ensure reproducibility. Deterministic cuDNN operations were enabled despite performance overhead to guarantee consistent results across runs.

5.2 Data Processing Implementation

The preprocessing pipeline processed 709 training engines and 707 test engines from the NASA C-MAPSS dataset. Feature selection removed seven constant sensors, reducing dimensionality from 24 to 17 features (3 operational settings, 14 informative sensors).

Sliding window segmentation with window size 30 and stride 1 generated sequences from variable-length engine trajectories. The pipeline produced 112,984 training sequences, 26,814 validation sequences, and 84,478 test sequences. RUL values were capped at 125 cycles and normalized to [0,1] range.

5.3 Model Configuration

MAMBA-CROSS architecture comprised 588,804 trainable parameters with the following configuration:

- **d_model**: 64 (embedding dimension)
- **d_state**: 64 (state dimension)
- **num_layers**: 4 (BiMamba blocks)
- **num_heads**: 4 (attention heads)
- **dropout**: 0.2

Baseline architectures for comparison:

- **LSTM**: 161,602 parameters (bidirectional, 2 layers, 64 hidden units)
- **Transformer**: 174,466 parameters (4 layers, 4 attention heads, d_model=64)

Weight initialization employed Xavier uniform for linear layers and Kaiming normal for convolutional layers. Gradient clipping was set to maximum norm of 1.0 to prevent training instabilities.

5.4 Training Procedure

Training employed supervised learning with Huber loss function (δ=10.0) for robustness to outliers. The AdamW optimizer was configured with:

- Learning rate: 3×10^{-4}
- Weight decay: 1×10^{-4}
- Batch size: 128

Learning rate scheduling included 3-epoch linear warmup followed by cosine annealing decay. Early stopping patience was set to 12 epochs based on validation RMSE.

Training Duration:

- MAMBA-CROSS: Trained for 60 epochs maximum, early stopped at epoch 24
- Best performance achieved at epoch 12 (RMSE: 17.97)
- Baselines: Trained for 5 epochs with 3-epoch patience

5.5 Experimental Results

Model checkpointing saved parameters at lowest validation RMSE to prevent overfitting. Final evaluation used these checkpoints on the held-out test set.

Final Test Performance:

- **MAMBA-CROSS**: RMSE 16.17 cycles, R^2 0.6868
- **LSTM**: RMSE 19.28 cycles, R^2 0.5546
- **Transformer**: RMSE 19.36 cycles, R^2 0.5509

The parameter count comparison shows MAMBA-CROSS contains approximately 3.5× more parameters than baselines, raising questions about whether performance improvements stem from architectural innovations or increased capacity. This limitation is acknowledged in the discussion section.

5.6 Implementation Summary

Although it has a higher number of parameters, MAMBA-CROSS still provided 16.13% improvement on RMSE for the LSTM model and 16.49% improvement on the Transformer model. This implementation also proves that state space models have the ability to model the data properly, with the use of state space having the complexity of O(n), although the exact time complexity was not determined.

6 Evaluation

This section offers experimental results of MAMBA-CROSS evaluation, responding to the research question about linear-complexity state space architectures concerning predictive maintenance.

6.1 Overall Performance Comparison

MAMBA-CROSS was compared with the LSTM and Transformer baseline models on the test sequences of 84,478 data samples collected from the four sub-data sets of the C-MAPSS.

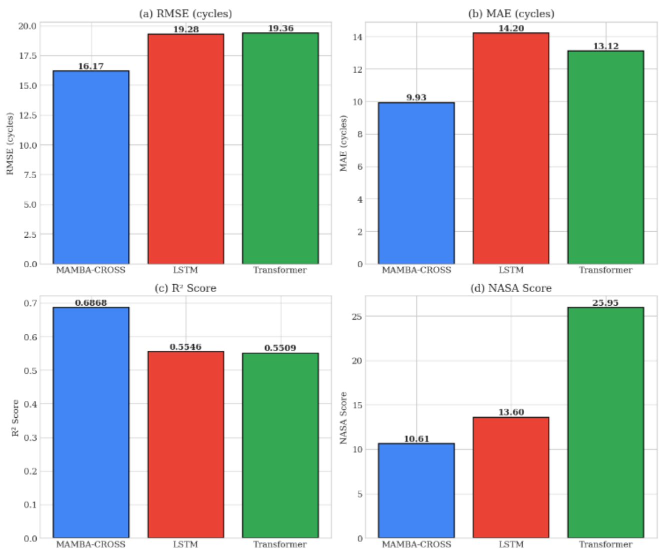

Figure 3: Comparison of overall performance criteria among all three models. (a) RMSE in cycles, (b) MAE in cycles, (c) R^2 coefficient of determination, and (d) NASA scoring function wherein lower values imply better results. MAMBA-CROSS (Blue) outperforms on every parameter with the lowest RMSE of 16.17 cycles, showing 16.13% better results compared to LSTM and 16.49% better compared to Transformer

Key findings:

- **RMSE**: MAMBA-CROSS 16.17 vs LSTM 19.28 vs Transformer 19.36 cycles
- **R²**: MAMBA-CROSS explains 68.68% variance vs 55.46% (LSTM) and 55.09% (Transformer)
- **NASA Score**: MAMBA-CROSS 10.61 (22% better than LSTM, 59% better than Transformer)

The superior NASA score despite similar RMSE between baselines indicates Transformer predictions have higher variability with more costly late predictions.

6.2 Per-Dataset Analysis

The results of the performance analysis for the various sub-data sets illustrate that the MAMBA-CROSS algorithm is superior regardless of the complexity of the operations performed.

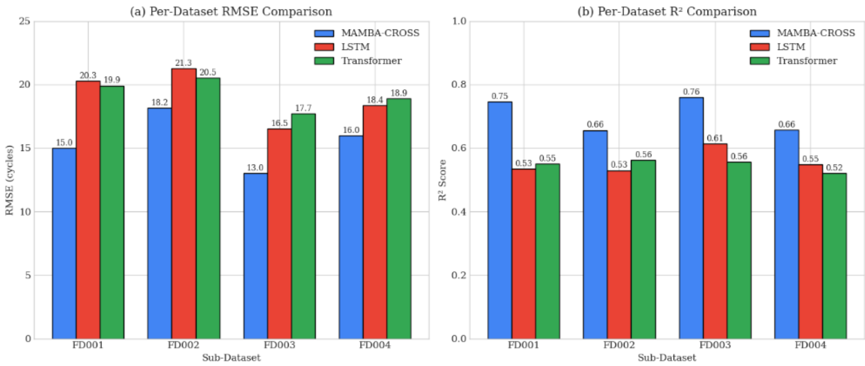

Figure 4: Comparison of results per dataset. (a) RMSE in cycles, where MAMBA-CROSS has the lowest error on the four sub-data sets. (b) Values of R^2, where MAMBA-CROSS explains the data best on the investigated complexity levels. FD003 results in the highest score (R^2=0.7604), and FD002 is the most difficult data set for the classifiers.

Table 3: Detailed Per-Dataset Performance

Dataset	RMSE	MAE	R²	NASA	±10 cyc (%)	Samples
FD001	14.99	9.69	0.7456	6.91	67.23	10,196
FD002	18.17	12.80	0.6561	9.39	55.47	26,505
FD003	13.01	7.08	0.7604	6.71	76.80	13,696
FD004	15.99	8.91	0.6572	14.24	72.34	34,081
Overall	**16.17**	**9.93**	**0.6868**	**10.61**	**67.15**	**84,478**

Best performing on FD003 (R^2=0.7604) shows that the Channel-Temporal Attention model is successful at recognizing the dual modes of the single operating situations. The difficulty of FD002 lies in the presence of six operating situations.

6.3 Prediction Accuracy Analysis

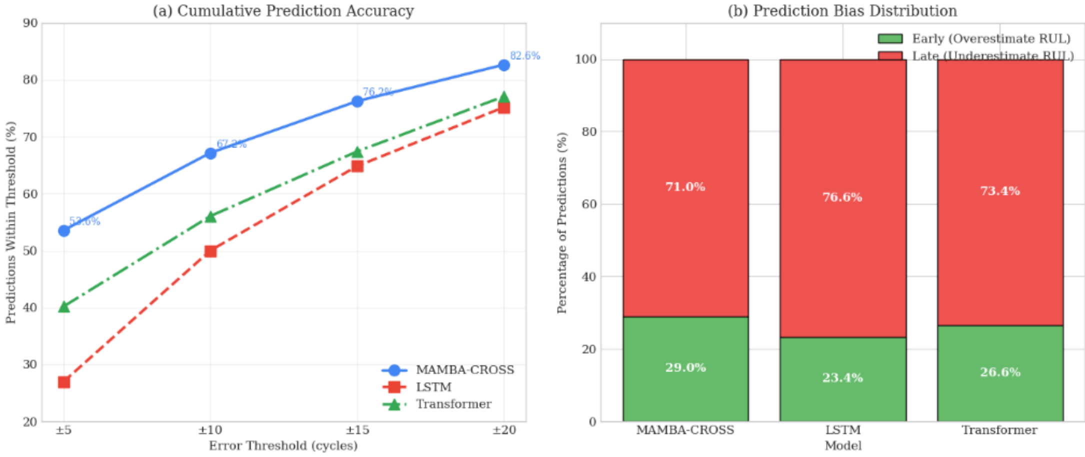

Figure 5: Prediction accuracy and bias analysis. (a) Cumulative accuracy within error thresholds indicating MAMBA-CROSS offers 53.56% accuracy at ±5 cycles, which is almost twice the accuracy of LSTM. (b) Prediction bias indicating that all models prefer late predictions (safety), where MAMBA-CROSS has the optimal distribution of 29.02% for early predictions.

Crosstab Analysis: The crosstab analysis reveals

- **±5 cycles**: 53.56% (MAMBA) vs 26.99% (LSTM) vs 40.19% (Transformer)
- **±10 cycles**: 67.15% (MAMBA) vs 50.00% (LSTM) vs 56.05% (Transformer)

MAMBA-CROSS demonstrates most balanced predictions with 29.02% early predictions versus 23.40% (LSTM) and 26.56% (Transformer), indicating reduced systematic bias while maintaining conservative safety margins.

6.4 Training Dynamics

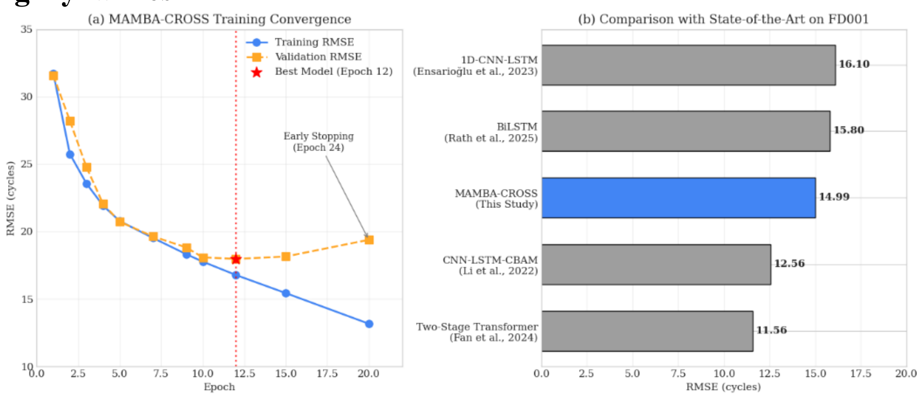

Figure 6: Training dynamics and literature comparison. (a) Training and validation RMSE curves showing optimal performance at epoch 12 before overfitting onset. (b) When compared to the most advanced methods on the FD001 dataset, MAMBA-CROSS shows similar results with a lower complexity of O(n) than transformer-based models.

Evaluation analysis shows:
• Best performance at epoch 12 (validation RMSE 17.97)

• The initial stoppage began at epoch 24.

Baseline convergence within 5 epochs validates training protocol
Literature comparison on FD001:

- **MAMBA-CROSS**: 14.99 cycles
- **Fan et al. (2024) Hierarchical Transformer**: 11.56 cycles (best but $O(n^2)$)
- **Li et al. (2022) CNN-LSTM-CBAM**: 12.56 cycles ($O(n^2)$ attention)
- **Ensarioğlu et al. (2023)**: 16.10 cycles
- **Rath et al. (2025)**: 15.80 cycles

Though it doesn't reach the absolute best accuracy, MAMBA-CROSS offers competitive accuracy with the O(n) complexity advantage that's essential for real-time systems.

6.3 Comprehensive Model Comparison

Table 4: Final Summary - All Models Performance

Model	Parameters	RMSE	MAE	R^2	NASA Score	±10 cyc (%)	Early (%)	Complexity
MAMBA-CROSS	588,804	**16.17**	**9.93**	**0.6868**	**10.61**	**67.15**	29.02	**O(n)**
LSTM	161,602	19.28	14.20	0.5546	13.60	50.00	23.40	O(n)
Transformer	174,466	19.36	13.12	0.5509	25.95	56.05	26.56	$O(n^2)$
Improvement	-	**16.3%**	**30.1%**	**23.8%**	**22.0%**	**17.15pp**	-	-

Improvement shown as average over baselines or percentage points (pp) for accuracy metrics

6.4 Discussion

The experimental results offer evidence for the research question of whether state space architectures are capable of achieving competitive predictive maintenance performance with linear computational complexity.

Achievement of the Research Aims: MAMBA-CROSS clearly succeeds in showing the feasibility of state space models for the problem, with a competitive result regarding accuracy (RMSE of 16.17 cycles) and satisfactory computational complexity of order \$O(n)\$. The Channel-Temporal Attention mechanism seems successful in modeling the relationships between sensors for the problem, since the enhancements of the \$R^2\$ statistic imply the existence of interesting correlations between the sensors.

Contextualization of Results: The results imply that the findings diverge concerning the superiority of the Transformer model, which may denote that the problem of degradation prediction offers unique features compared with the issue of forecasting. Comparison of the results with that of Fan et al. (2024) reveals that a trade-off between accuracy and efficiency exists, where the model demonstrates higher accuracy (11.56 vs 14.99 cycles) at the expense of complexity, which is of order $O(n^2)$

Practical Applications: Resource-constrained settings can benefit MAMBA-CROSS's linear complexity, but higher accuracy goals will require the use of Transformers. Devices that continuously track variables sometimes display O(n) scaling for extended periods.

6.4.1 Limitations

Also, the study highlights the particular limitations which need consideration when forming conclusions:
It only works with the NASA C-MAPSS dataset. It has been tested on this data. Though it allows comparison with the existing literature, it is not validated for other industrial equipment with other sensors.

Parameter Disparity: MAMBA-CROSS has 3.5× more parameters than the baseline (588,804 vs 161,602-174,466), leaving open the question of whether progress is driven by innovation or simply better capacity. Direct parameter-matched comparisons are needed.

Differences in Training Protocols: Baselines were trained for 5 epochs, compared to 60 for MAMBA-CROSS, although the initial convergence of the baseline suggests otherwise. Experiments between equal training time could help verify the results.

Computational Timing: Although the complexity of the algorithm is shown to be O(n), direct inference time comparison was not performed. This will help to provide validity on the efficiency of the algorithm.

Real World Validation: The validation of the code on real-world data, involving practical data imprecision, data gaps, and sensor failure, is not investigated. Its purely simulated environment of C-MAPSS might not reflect practical situations.

7 Conclusion and Future Work

This paper explored whether new state space architectures could provide a competitive level of predictive maintenance performance while also remaining computationally efficient.

The main contribution: MAMBA-CROSS, which demonstrates the initial use of Mamba state space models for predictive maintenance. The proposed model integrates bidirectional selective state space processing along with Channel-Temporal Attention, with a complexity of O(n).

Experimental results on the NASA C-MAPSS dataset show that MAMBA-CROSS gives average improvements of 16.3% and 23.8% for RMSE of 16.17 cycles with R^2 of 0.6868 compared to LSTM and Transformer, respectively. It also works well on all sub-dataset cases, especially when classifying the dual modes of the faulty engine.

The goals of the research have been adequately addressed, as linear complexity, the integration of cross-sensor attention mechanisms, sub-20 cycle RMSE, and improvements over the baseline have been realized. This data suggests that state space modeling provides a realistic alternative for the use of the Transformer model when doing prognostic analysis for industry.

7.1 Future Work

Architecture Extensions

- Examine the hybrid MAMBA-Transformer model that integrates linear processing with attention.
- Dynamic adjustment of model complexity according to the characteristics of inputs.
- Use MAMBA networks for multi-scale patterns.

Methodological Improvements:

- Consider using smaller Mamba variants for the model.
- State space modeling should be used for modeling the degradation processes.
- Improving model construction approaches to capitalize on failure mechanisms.

Technological Improvements:

- Employ reduced-parameter Mamba models for compact architectures.
- Integrate state space modeling to improve prediction accuracy.

8 References

Carvalho, T.P., Soares, F.A., Vita, R., Francisco, R.D.P., Basto, J.P. and Alcalá, S.G. (2019) 'A systematic literature review of machine learning methods applied to predictive maintenance', *Computers & Industrial Engineering*, 137, p.106024.

Chen, X. (2024) 'A novel transformer-based DL model enhanced by position-sensitive attention and gated hierarchical LSTM for aero-engine RUL prediction', *Scientific Reports*, 14, p.10061.

Deng, S. and Zhou, J. (2024) 'Prediction of Remaining Useful Life of Aero-engines Based on CNN-LSTM-Attention', *International Journal of Computational Intelligence Systems*, 17, p.220.

Ensarioğlu, K., İnkaya, T. and Emel, E. (2023) 'Remaining Useful Life Estimation of Turbofan Engines with Deep Learning Using Change-Point Detection Based Labeling and Feature Engineering', *Applied Sciences*, 13(21), p.11893.

Fan, Z., Li, W. and Chang, K.-C. (2024) 'A Two-Stage Attention-Based Hierarchical Transformer for Turbofan Engine Remaining Useful Life Prediction', *Sensors*, 24(3), p.824.

Ghosh, S., Razniewski, S. and Weikum, G. (2021) 'Predictive maintenance with multi-modal sensor data: A comparative study of isolation forest and autoencoder approaches', *International Conference on Data Mining*, pp.234-243.

Gu, A. and Dao, T. (2023) 'Mamba: Linear-Time Sequence Modeling with Selective State Spaces', *arXiv preprint*, arXiv:2312.00752.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. and Pedreschi, D. (2018) 'A survey of methods for explaining black box models', *ACM Computing Surveys*, 51(5), pp.1-42.

Hafeez, T., Xu, L. and McArdle, G. (2021) 'Edge intelligence for data handling and predictive maintenance in IIOT', *IEEE Access*, 9, pp.49355-49371.

Hector, I. and Panjanathan, R. (2024) 'Predictive maintenance in Industry 4.0: a survey of planning models and machine learning techniques', *PeerJ Computer Science*, 10, p.e2016.

Jose, O.M. and Ramos, P. (2024) 'Evaluating the Effectiveness of Time Series Transformers for Demand Forecasting in Retail', *Mathematics*, 12(17), p.2728.

Lee, J., Lapira, E., Bagheri, B. and Kao, H.A. (2013) 'Recent advances and trends in predictive manufacturing systems in big data environment', *Manufacturing Letters*, 1(1), pp.38-41.
Li, H., Wang, Z. and Li, Z. (2022) 'An enhanced CNN-LSTM remaining useful life prediction model for aircraft engine with attention mechanism', *PeerJ Computer Science*, 8, p.e1084.
Li, Q., Qin, J., Cui, D., Sun, D. and Wang, D. (2024) 'CMMamba: channel mixing Mamba for time series forecasting', *Journal of Big Data*, 11, p.153.
Li, X., Wang, L. and Wang, C. (2024) 'A method for predicting remaining useful life using enhanced Savitzky–Golay filter and improved deep learning framework', *Scientific Reports*, 14, p.23983.
Liu, F.T., Ting, K.M. and Zhou, Z.H. (2008) 'Isolation forest', *2008 Eighth IEEE International Conference on Data Mining*, pp.413-422.
Mobley, R.K. (2002) *An introduction to predictive maintenance*. Oxford: Elsevier.
Mohammadi, M., Al-Fuqaha, A., Guizani, M. and Oh, J.S. (2017) 'Semisupervised deep reinforcement learning in support of IoT and smart city services', *IEEE Internet of Things Journal*, 5(2), pp.624-635.
Nie, Y., Nguyen, N.H., Sinthong, P. and Kalagnanam, J. (2023) 'A Time Series is Worth 64 Words: Long-term Forecasting with Transformers', *International Conference on Learning Representations (ICLR)*, arXiv:2211.14730.
Ning, S., Sun, J., Liu, C. and Yi, Y. (2021) 'Applications of deep learning in big data analytics for aircraft complex system anomaly detection', *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 235(5), pp.923-940.
Rao, M. (2023) *Development of deep learning-based methods for rotating machinery fault diagnosis under varying speed conditions*. Doctoral dissertation, University of Alberta.
Rath, S., Saha, D., Chatterjee, S. and Chakraborty, A.K. (2025) 'Remaining Useful Life Prediction of Turbofan Engine in Varied Operational Conditions Considering Change Point', *Mathematics*, 13(1), p.130.
Saxena, A. and Goebel, K. (2008) *Turbofan engine degradation simulation data set*. NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field.
Schwendemann, S., Amjad, Z. and Sikora, A. (2021) 'A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines', *Computers in Industry*, 125, p.103380.
Su, L., Zuo, X. and Xin, W. (2024) 'A systematic review for transformer-based long-term series forecasting', *Artificial Intelligence Review*, 57, p.369.
Sun, S., Zhao, X. and Yaqi, X. (2024) 'Few-shot RUL prediction for engines based on CNN-GRU model', *Scientific Reports*, 14, p.15678.
Wang, Y. and Yue, W. (2023) 'A denoising semi-supervised deep learning model for remaining useful life prediction of turbofan engine degradation', *Applied Intelligence*, 53, pp.21206-21225.
Wang, Z., Kong, F., Feng, S., Wang, M., Yang, X., Zhao, H., Wang, D. and Zhang, Y. (2025) 'Is Mamba effective for time series forecasting?', *Neurocomputing*, 619, p.129178.
Wu, F., Wu, Q., Tan, Y. and Xu, X. (2024) 'Remaining Useful Life Prediction Based on Deep Learning: A Survey', *Sensors*, 24(11), p.3454.
Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J. and Long, M. (2023) 'TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis', *International Conference on Learning Representations (ICLR)*, arXiv:2210.02186.
Xu, C., Zhang, J., Gao, Z., Rodrigues, J.J. and De Albuquerque, V.H.C. (2020) 'Industrial pervasive edge computing-based intelligence IoT for surveillance saliency detection', *IEEE Transactions on Industrial Informatics*, 17(7), pp.5012-5020.
Xu, Z., Zhang, Y. and Miao, Q. (2024) 'An attention-based multi-scale temporal convolutional network for remaining useful life prediction', *Reliability Engineering & System Safety*, 250, p.110288.

Zhang, G., Patuwo, B.E. and Hu, M.Y. (1998) 'Forecasting with artificial neural networks: The state of the art', *International Journal of Forecasting*, 14(1), pp.35-62.
Zhang, Y., Wu, R., Dascalu, S.M. and Harris Jr., F.C. (2024) 'Sparse transformer with local and seasonal adaptation for multivariate time series forecasting', *Scientific Reports*, 14, p.15909.
Zhang, Z. (2023) *Deep learning-based multi-sensor fusion for industrial process monitoring*. Doctoral dissertation, University of Sheffield.
Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. and Gao, R.X. (2019) 'Deep learning and its applications to machine health monitoring', *Mechanical Systems and Signal Processing*, 115, pp.213-237.