
REMAINING USEFUL LIFE PREDICTION OF TURBOFAN ENGINE USING ATTENTION AUGMENTED LSTM BASED NETWORK

**Oluwafemi Daniel Oluwapelumi¹, Samuel David Iyaghigba^{1,*}, Marcus Aul Maafan¹,
Sadiq Thomas², Abbas Abdullahi¹**

¹Air Force Institute of Technology Kaduna

²Nile University of Nigeria Abuja/Dept of Computer Engineering

Email: danieloluwafemi58@gmail.com, samiyagh004@yahoo.com*, Aulmarcus6@gmail.com,
Sadiqthomas@nileuniversity.edu.ng, netlikora@gmail.com

***Corresponding author:** Samuel David Iyaghigba

DOI: <https://doi.org/10.67203/sp.ICESID.2026>

ABSTRACT

Aircraft systems are complex networks of components that support the entire mission and functionality of an aircraft. These complex engineered systems play fundamental roles in providing critical services, particularly regarding air transportation and energy needs. Among these, turbofan engines are the primary power source. As the usage of these complex engineering systems expands, they are increasingly subjected to harsh operational environments. These conditions inevitably make them susceptible to mechanical faults that, if not detected early, can result in catastrophic failure. Notably, the turbofan engine has accounted for 60% of airplane failures during flight. Therefore, Prognostics and Health Management technology utilizing deep learning is essential for predicting the Remaining Useful Life of these systems. This study explores a Long Short-Term Memory network integrated with a sliding-window approach and a Temporal Attention mechanism to estimate engine operational lifespan. Unlike existing literature reliant on legacy benchmarks, this model is trained and validated on a high-fidelity dataset generated using AGTF30, a NASA-developed turbofan simulation environment; because RUL is measured in seconds rather than operating cycles, comparison against C-MAPPS-based literature is limited to relative RMSE rather than direct numerical validation. By extracting temporal degradation features from raw multivariate sensor data, the framework achieves an overall Root Mean Square Error of 249.0 seconds and, crucially, a near-failure RMSE of 39.3 seconds, demonstrating highly reliable predictive capability during critical terminal degradation phases.

Keywords: Turbofan Engine; Predictive Maintenance; Remaining Useful Life; LSTM Model; Deep learning approach

1. INTRODUCTION

Complex engineered systems play a fundamental role in critical services such as energy, transport, and aerospace, underpinning the proper functioning of modern societies and economies. As these systems see wider use, they are increasingly subjected to harsh operational environments on a daily basis (Saxena et al., 2008). Left undetected, the compounding effect of minor mechanical faults can result in catastrophic failure, and for turbofan engines, the primary power source for an aircraft, the consequences of undetected degradation are particularly severe: as these engines accumulate flight hours, internal components gradually suffer wear, corrosion, and fatigue, leading to performance degradation and shortened lifespan.

In modern industrial operations, the accurate prediction of Remaining Useful Life (RUL) for critical machinery such as turbofan engines is of paramount concern (Thakkar & Chaoui, 2022). The turbofan is a core propulsion component of an aircraft; its stability and reliable operation are critical to flight safety (Jung et al., 2025). A turbofan engine is an air-breathing propulsive device and the predominant propulsion system in modern commercial and military aviation. Operating on the Brayton thermodynamic cycle, it consists of a gas turbine that adds kinetic energy to the air passing through it by burning fuel, and a ducted fan powered by that turbine to force air rearward and produce thrust (Chapman & Litt, 2017). The cost associated with machinery maintenance can be substantial, particularly in civil aviation, where aero-engine maintenance contributes significantly to overall operating expenses (Fu et al., 2023). RUL prediction involves estimating the time remaining before a

component reaches a failure threshold, given its historical and current operational state. Accurate RUL prediction lets manufacturers and operators reduce costs, improve reliability, and optimize maintenance strategies, maximizing service life while minimizing downtime and improving safety (Fu et al., 2023).

RUL is the amount of time a machine or asset is likely to operate before it requires repair or replacement. Depending on the system, this period can be represented in days, kilometers, cycles, or another relevant quantity. RUL prediction gives early warning of failure and forms a central pillar of Prognostics and Health Management (PHM), allowing engineers to schedule maintenance, optimize operating efficiency, and avoid unplanned downtime.

RUL prediction methods fall into two broad categories: model-based and data-driven. Model-based methods construct a model from mechanical or thermodynamic principles, a process that takes considerable time and depends heavily on prior expert knowledge, and even then, prediction accuracy cannot be guaranteed given the complexity and uncertainty of real engine systems. Data-driven approaches instead learn estimation models from historical run-to-failure data, avoiding the limitations of physical models and expert-built knowledge bases while benefiting from lower computational cost. Deep learning has become the dominant data-driven direction, offering strong results on RUL-related tasks, though mining complex time-series data effectively remains difficult given that turbofan sensor data is non-linear, time-variant, large in scale, and high in dimension. Long Short-Term Memory (LSTM) networks, a specialized form of Recurrent Neural Network, have proven particularly effective for this kind of temporal data. Convolutional Neural Networks saw earlier adoption but struggled to model long-term continuous degradation across multiple flight cycles (Li et al., 2018).

Previous RUL research has relied heavily on the public NASA C-MAPSS benchmark, which is widely adopted. However, it originates from a simplified thermodynamic model and does not capture the characteristics of modern geared turbofan architectures. This study addresses that gap by generating a run-to-failure dataset from the NASA AGTF30 geared turbofan dynamic simulation model, which is, to the authors' knowledge, the first application of this model for supervised RUL dataset generation. An LSTM network incorporating a sliding-window approach and a single-head temporal attention mechanism is then developed and evaluated on this dataset to predict the remaining useful life of the turbofan engine.

The specific contributions of this work are as follows. First, a run-to-failure dataset is generated from the NASA AGTF30 simulation, capturing multivariate sensor readings, including turbine exit temperature, pressure, and flow (T45, P45, W45) and high-pressure compressor exit conditions (T25, P25), across full degradation trajectories. Second, an LSTM network is developed incorporating a single-head temporal attention mechanism that computes an alignment score per timestep via a tanh-activated dense layer followed by a softmax-weighted sum, trained using a sliding-window approach to convert the raw multivariate sequence into fixed-length windows suitable for temporal modeling. Third, the model is evaluated using standard prognostic metrics, including Root Mean Square Error (RMSE) and an asymmetric scoring function that penalizes late RUL predictions more heavily than early ones, reflecting the safety-critical nature of aviation maintenance. This contribution aligns with the Intelligent Systems, Automation and Control Engineering sub-theme of ICESID 2026 and supports sustainable aviation development in Nigeria by advancing local capacity in predictive maintenance at AFIT, enabling reduced unscheduled downtime, extended engine life, and lower material waste and operational cost.

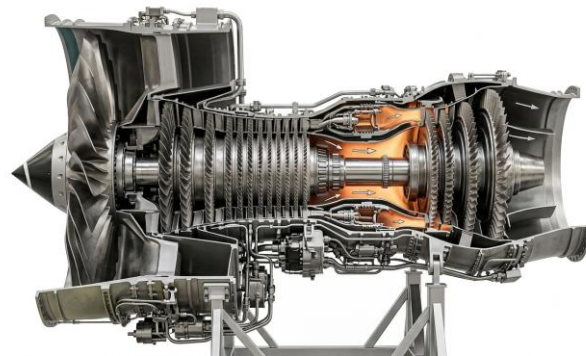


Figure 1: Typical Turbofan engine - Author own figure

The remainder of this paper is organized as follows. Section 2 reviews related work on RUL prediction methods and identifies the gap this study addresses. Section 3 describes the methodology, including dataset generation, feature selection, and model architecture. Section 4 presents the results and discusses model performance across seeds, engine life stages and its limitations. Section 5 summarizes the key findings. Section 6 outlines directions for future work, and Section 7 concludes the paper.

2. LITERATURE REVIEW

Prognostics, defined as the estimation of remaining useful component life, is central to systems health management (Vichare & Pecht, 2006). The inception of PHM can be traced back to the 1980s, when the Civil Aviation Authority (CAA) of the United Kingdom initiated its use with the aim of mitigating helicopter accidents. It later transitioned into health and usage monitoring systems (HUMS) by the late 1990s, enabling measurement of both the health condition and performance of helicopters (Fu et al., 2023).

In the early days of aircraft engines, defects were identified and repaired through Corrective Maintenance (CM). This approach was later replaced by Preventive Maintenance (PM), in which inspections and tasks are scheduled according to the probability of failure for specific components and usage conditions (Jardine et al., 2006). While Preventive Maintenance presented a significant step forward by introducing a degree of anticipation into maintenance planning, it remains constrained by its reliance on fixed time intervals or usage thresholds rather than the actual condition of the equipment. Components may be replaced prematurely, resulting in unnecessary cost and wasted resources, or failures may still occur between scheduled maintenance windows (Jardine et al., 2006). These limitations gave rise to a more mature model known as Condition-Based Monitoring (CBM), in which maintenance decisions are driven by the real-time health state of the system as gathered from sensor data, operational parameters, and observable degradation indicators. Building on CBM, Predictive Maintenance (PdM) extends this concept by forecasting when failure is likely to occur rather than simply estimating current health.

Remaining Useful Life (RUL) prediction methods are broadly categorized into two paradigms: physics-based and data-driven, as shown in Figure 2. Physics-based methods leverage mathematical models to represent system degradation over time by simulating the underlying failure mechanisms through well-established physical laws (Cubillo et al., 2016). These can be broadly divided into physical failure models and state-space models. Physical failure models describe degradation directly using governing equations derived from material science and mechanics, such as Archard's law for adhesive wear and the Paris-Erdogan law for fatigue crack growth. State-space models instead apply filtering techniques such as Kalman Filters and Particle Filters to track the evolving health state of a system in real time, updating predictions as new observations become available (Cubillo et al., 2016). Despite these strengths, physics-based models carry significant limitations that restrict their use in complex real-world systems, particularly gas turbine engines. They require precise and comprehensive knowledge of the physics governing each degradation mode, and developing and validating such models demands considerable domain expertise and time, making them difficult to scale across engine types or fault modes. This computational and developmental cost is amplified for real-time applications (Cubillo et al., 2016).

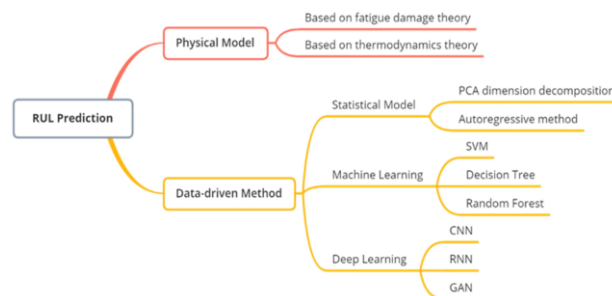


Figure 2: Classification of remaining useful life prediction methods (adapted from Xie et al., 2021).

Data-driven methodologies instead leverage historical data to evaluate current system health and predict RUL (Ensarioğlu et al., 2023). Rather than directly modeling the physical complexities of

wear and mechanical deterioration, data-driven approaches employ artificial intelligence to identify failure patterns and extract critical features from large datasets, revealing degradation characteristics by converting raw sensor history into usable information. Early data-driven research focused on traditional machine learning approaches such as Support Vector Machines, K-means clustering, and reinforcement learning (Ensarioğlu et al., 2023). While effective on small datasets, these approaches struggled in turbofan prognostics for two main reasons. First, manual feature engineering requires researchers to create and select features based on their own understanding of the data and statistical intuition, which becomes difficult to scale across large or high-dimensional datasets (H. Liu et al., 2021). Second, these methods lack temporal memory: they treat each sensor snapshot as an independent event, ignoring the fact that an engine's health at cycle 50 is directly dependent on its condition at cycle 49.

To address these limitations, researchers turned to deep learning, which uses multilayered neural networks for tasks such as classification, regression, and representation learning. Common architectures include fully connected networks, deep belief networks, recurrent neural networks, convolutional neural networks, generative adversarial networks, and transformers. Convolutional Neural Networks (CNNs) were initially adapted from image processing to extract spatial features from multi-sensor grids. While CNNs are effective at detecting local patterns, such as a sudden pressure rise, they struggle to model long-term, continuous degradation across hundreds of flight cycles, since they lack a recursive structure (Ensarioğlu et al., 2023). This was why Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks were introduced. RNNs (Hochreiter & Schmidhuber, 1997; J. Zhang et al., 2018) are a family of neural networks specialized for processing a sequence of values, capable of capturing temporal dependencies within input samples. RNNs, especially those built from LSTM cells and Gated Recurrent Units (GRUs) (Dey & Salem, 2017; Yuan et al., 2016), and attention (Vaswani et al., 2017) have been widely used for prognostics of components such as lithium-ion batteries (Y. Zhang et al., 2018), gears, fuel cells, and turbofan engines (Cubillo et al., 2016; Fu et al., 2023; Saxena et al., 2008; Y. Zhang et al., 2018). However, standard RNNs suffer from the vanishing gradient problem, gradually forgetting early health-state information as more sequential data is introduced, which makes long-term dependencies difficult to learn. This limitation gave rise to more advanced recurrent architectures, notably LSTM networks (Hochreiter & Schmidhuber, 1997; J. Zhang et al., 2018) and GRUs (Dey & Salem, 2017). Since aero-engine data is inherently multivariate and time-dependent, this has contributed to the widespread adoption of LSTM models in RUL research (Hochreiter & Schmidhuber, 1997). Ensarioğlu et al. (2023) demonstrated that RNNs are efficient at analyzing the time-series characteristics of turbofan sensor data.

The LSTM addresses the vanishing gradient problem through three gating mechanisms that regulate the flow of information across time steps. The forget gate determines what historical information is discarded and is given as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

The input gate controls what new information is written to the cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

And the output gate determines what information is passed to the next hidden state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

Solanki et al. (2025) proposed a model combining pre-processing techniques with an LSTM network, introducing an improved piecewise linear degradation model to determine the initial point of deterioration and assign RUL target labels, evaluated on the C-MAPSS dataset. Dida et al. (2025) developed a prediction model combining CNN, LSTM, and attention mechanisms, using CNN to extract local features from sensor data while the LSTM-attention network analyzed temporal characteristics to forecast engine life. More recently, Transformer-based architectures have introduced an alternative to recurrence, replacing sequential processing with self-attention mechanisms that allow the model to weigh different parts of a sequence simultaneously, making them theoretically powerful for long-range dependency modeling (H. Liu et al., 2021; Vaswani et al., 2017). Recent work has applied hierarchical, multi-stage attention Transformers to turbofan RUL prediction with strong reported accuracy, demonstrating that the architecture race in this field has continued well past LSTM-based methods (Fan et al., 2024; L. Liu et al., 2022), demonstrating that the architecture race in this field has continued well

past LSTM-based methods. What has not moved, however, is the benchmark itself: this Transformer-based literature, like the LSTM-based work reviewed above, remains confined almost exclusively to C-MAPSS. The dataset, not the model architecture, is the persistent bottleneck this study addresses.

The quality of the simulation environment from which training data is generated is a primary determinant of whether a prognostic model will generalize to real-world application. The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, developed by NASA's Ames Research Center (Saxena et al., 2008), remains the most widely used benchmark in the turbofan RUL literature. C-MAPSS simulates a two-spool turbofan engine based on the thermodynamic architecture of the CFM56, modeling degradation in the HPC and HPT modules across six combinations of altitude, throttle resolver angle, and Mach number. The dataset is divided into four subsets, FD001 through FD004, each representing a different combination of operating conditions and fault modes (Saxena et al., 2008), with 21 sensor measurements recorded per flight cycle covering temperatures, pressures, shaft speeds, and efficiencies. C-MAPSS has been essential to the field, providing a public benchmark that has allowed researchers to contribute meaningfully to turbofan prognostics. It carries structural limitations, however, that narrow the conclusions that can be drawn from evaluations conducted on it. First, C-MAPSS employs a quasi-steady-state thermodynamic model, meaning operating conditions transition between predefined clusters rather than evolving continuously, concealing the transient dynamics that occur within a single flight cycle. High-fidelity telemetry from platforms such as AGTF30, by contrast, captures the rich, non-linear behavior seen during throttle excursions, altitude transitions, and rapid acceleration events (Chapman & Litt, 2017). Second, C-MAPSS models a dual-spool turbofan that does not reflect the variable-geometry features, gear reduction system, or closed-loop control behavior of next-generation geared turbofan engines.

To address the limitations of steady-state datasets such as C-MAPSS, the research community has turned to high-fidelity simulation environments capable of generating more realistic engine data. The Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) is the most prominent example of this shift (Lavelle et al., 2014). Unlike earlier algebraic frameworks, T-MATS takes a modular, physics-based approach in which fan speeds, pressures, and temperatures evolve continuously in response to transient input (Chapman & Litt, 2017), using a library of component blocks for compressors, turbines, combustors, nozzles, ducts, and control actuators that can be assembled into different propulsion system configurations. The AGTF30 engine model, built within the T-MATS environment, represents a 30,000 lbf class geared turbofan (Chapman & Litt, 2017; Lavelle et al., 2014). This architecture introduces complexities absent from legacy benchmarks, most notably a gear reduction system that decouples the fan from the low-pressure spool, allowing both to operate at their respective aerodynamically optimal speeds. AGTF30 also runs within a closed-loop FADEC simulation, with a multivariable controller regulating thrust and surge margin. These features make AGTF30 a considerably more demanding environment for prognostic models, requiring them to differentiate between control-induced transients and genuine degradation signatures, a distinction that meaningfully strengthens a model's potential for real-world generalization (Chapman & Litt, 2017).

Taken together, the literature reviewed here reveals a consistent pattern: model architectures for turbofan RUL prediction, from early machine learning through LSTM and now Transformers, have advanced considerably, but almost entirely within the confines of C-MAPSS, a benchmark whose steady-state, dual-spool architecture no longer represents the modern engines it is meant to stand in for. Whether the properties that make these architectures effective on that older benchmark transfer to the transient, closed-loop, gear-coupled dynamics of a modern geared turbofan is a question the existing literature has not answered, largely because almost no one has tested it under those conditions. This gap, is not a shortage of capable model architectures but a shortage of realistic conditions to evaluate them against, is what this study addresses directly, by generating a fault-seeded dataset from the T-MATS/AGTF30 environment and evaluating an attention-augmented LSTM against it.

3. IMPLEMENTATION OF METHODOLOGY

This section describes the four-stage engineering pipeline used to generate, prepare, and model the dataset: dataset generation, data preprocessing, model architecture, training and optimization. The final two stages are addressed in the sections that follow.

3.1 Dataset Generation

A synthetic dataset was generated using the AGTF30 (Advanced Geared Turbofan, 30,000 lbf thrust class) simulation model implemented within the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS), a NASA Glenn Research Center open-source MATLAB/Simulink library. The AGTF30 simulation environment is shown in Figure 3. Unlike empirical datasets, this simulation environment solves full thermodynamic cycle equations at each timestep, providing physically consistent sensor readings across all degradation scenarios. Degradation is modeled as continuous wear of the High-Pressure Compressor (HPC) and High-Pressure Turbine (HPT), represented by adjustments to each component's efficiency and flow capacity scalars. To the authors' knowledge, this represents the first published application of the AGTF30 dynamic simulation model for generating a run-to-failure RUL training dataset.

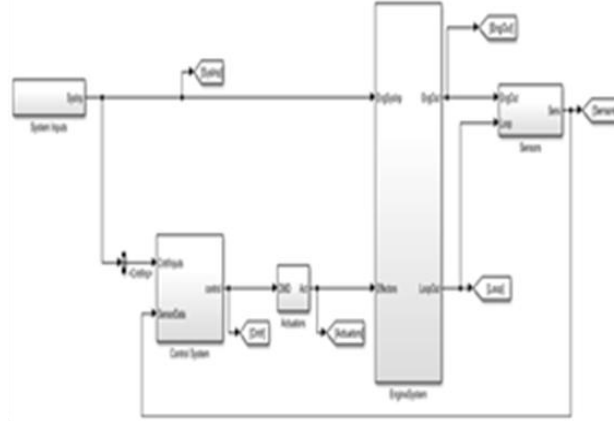


Figure 3: AGTF30 Simulink model.

Table 1: Mission Profile Parameters

Phase	Start (s)	End (s)	PLA (deg)	Altitude (ft)	Description
Ground Idle	0	100	40	0	Engine stabilizes at low power
Power-Up	100	500	40–80	0	Throttle advances to cruise setting
Cruise	500	2000	80	0	Steady cruise, fault detection window

Engine degradation is introduced by scaling the efficiency and flow capacity parameters of the HPC and HPT blocks within the T-MATS model. For each engine unit, four independent degradation factors are drawn from a uniform random distribution over the range [0.88, 1.00] using a fixed seed (rng (42)): one each for HPC efficiency, HPC flow capacity, HPT efficiency, and HPT flow capacity. Each factor is multiplied against its corresponding healthy design-point value to produce the degraded parameter used for that engine's simulation run, yielding four statistically independent degradation trajectories per engine rather than a single composite severity level. This randomized, per-unit approach follows the precedent set by the original C-MAPSS benchmark, in which each engine begins with an independently varying degree of initial wear (Saxena et al., 2008), and reflects the manufacturing and operational variability seen across a real engine fleet rather than a small set of discrete, repeatable severity levels. The degradation floor of 0.88 was set through boundary testing to ensure numerical stability across the full 2,001-second simulation.

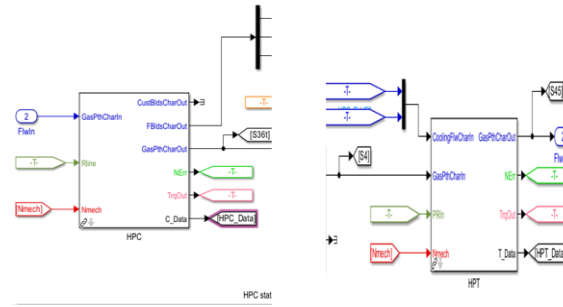


Figure 4: HPC and HPT block in NASA AGTF30 environment

Table 2: Component Degradation Factor Ranges

Component	Parameter	Healthy Value	Degradation Range
HPC	η (efficiency scalar)	0.994014	0.88–1.00
HPC	W_c (flow capacity scalar)	0.132800	0.88–1.00
HPT	η (efficiency scalar)	0.998392	0.88–1.00
HPT	W_c (flow capacity scalar)	0.195100	0.88–1.00

The AGTF30 model includes a Full Authority Digital Engine Control (FADEC) fuel controller that regulates fuel flow to hold exhaust gas temperature (T45) at a constant 2,241 R under steady-state cruise. As a result, T45 remains largely invariant across degradation levels. Degradation instead manifests primarily in total pressure at the HPT exit (P45) and mass flow rate at station 45 (W45), both of which drop measurably with compressor and turbine wear. HP shaft rotational speed (Nh) and thrust-specific fuel consumption (TSFC) were also evaluated as candidate signals but excluded, Nh showed only moderate sensitivity and TSFC changed negligibly across degradation levels. HPC inlet temperature (T25) and pressure (P25) were retained as upstream thermodynamic state indicators. Table 3 summarizes all candidate signals and which were retained as model inputs. This pattern, pressure and flow as primary degradation indicators rather than EGT alone, is consistent with real turbofan health-monitoring practice.

Table 3: *Candidate Sensor Signals and Degradation Sensitivity*

Signal	Description	Healthy (cruise)	Sensitivity	Retained
P45	Total pressure at HPT exit (psia)	~134.2	High	Yes
W45	Mass flow rate at station 45 (lbm/s)	~66.9	High	Yes
Nh	HP shaft rotational speed (RPM)	~21,150	Medium	No
T45	Exhaust gas temperature (R)	2241.0	Low (FADEC-held)	Yes
TSFC	Thrust-specific fuel consumption	~0.174	Low	No
T25	HPC inlet temperature (R)	~518	Medium	Yes
P25	HPC inlet pressure (psia)	~14.7	Medium	Yes

Simulations were logged at 1 Hz, yielding 2,001 rows per engine and 240,120 rows across the full 120-engine fleet. Each row contains the columns listed in Table 4. While 120 unique engine trajectories appear small, the sliding window approach with window 100 generates approximately 159,000 training samples, so the number of training sequences exceeds the number of model parameters. Generalization across trajectories is still limited to a single simulation environment with degradation factors in [0.88, 1.00], which is discussed in Section 4.7."

Table 4: Dataset Column Description

Column	Type	Description
unit_id	Integer	Engine identifier (1–120)
Time	Float	Elapsed simulation time (s)
T45	Float	HPT exit total temperature (R)
P45	Float	HPT exit total pressure (psia)
W45	Float	Mass flow rate at station 45 (lbm/s)
T25	Float	HPC inlet total temperature (R)
P25	Float	HPC inlet total pressure (psia)
HPC_eff	Float	HPC degradation factor applied (metadata only)
HPT_eff	Float	HPT degradation factor applied (metadata only)
RUL	Float	Remaining Useful Life label (s)
censored	Integer	0 = failed, 1 = did not reach failure threshold

3.2 Data Preprocessing

Out Of the seven candidate signals, five (T45, P45, W45, T25, P25) were retained as model inputs; they showed measurable, non-trivial change across the fleet as degradation progressed, while Nh and TSFC did not. The HPC and HPT degradation factors themselves (HPC_eff, HPT_eff) were excluded from model inputs entirely to prevent data leakage, since these internal efficiency scalars are not observable in a real deployment scenario and would artificially inflate accuracy if included.

Raw sensor values alone proved insufficient, since a single-timestep reading gives no indication of whether an engine is approaching failure or simply sitting in steady-state cruise; two engines at an identical T45 value might be on entirely different trajectories. To capture this, four rolling features were computed from the two most degradation-sensitive channels, P45 and W45: a ten-step rolling mean (P45_mean10, W45_mean10), smoothing single-timestep noise, and a ten-step finite-difference slope (P45_slope10, W45_slope10), capturing whether each channel is rising, falling, or flat over the preceding ten seconds, the most direct indicator of an active degradation event.

The 120 engines were divided into training (84), validation (18), and test (18) subsets using a stratified split by engine health score rather than a random shuffle. This ensures each subset contains a proportional mix of healthy and degraded units (Table 5).

Table 5: Train-Validation-Test Split

Subset	Engine Units	Count	Failed	Censored
Training	1–84	84 (70%)	59	25
Validation	85–102	18 (15%)	9	9
Test	103–120	18 (15%)	10	8
Total	1–120	120	78	42

All five input features were scaled to the range [0, 1] using min-max normalization, preventing features with large absolute values, such as T45 at approximately 2,241 R, from numerically dominating the gradient relative to features with small absolute values, such as P25:

$$x_{\text{norm}} = (x - x_{\min}) / (x_{\max} - x_{\min}) \quad (4)$$

where $x_{\min}$ and $x_{\max}$ are computed exclusively from the training set and applied identically to the validation and test sets, avoiding data leakage from statistics computed on the full dataset before splitting.

Failure is defined as the first timestep during the cruise phase ($t = 500\text{s}$ to $t = 2,000\text{s}$) at which P45 drops below 75% of its healthy cruise baseline of 134.215 psia ($P45_{\text{fail}} = 100.66$ psia).

$$P45_{\text{fail}} = 0.75 \times 134.215 = 100.66 \text{ psia}$$

Restricting this check to the cruise phase prevents the naturally lower P45 at ground idle from being misclassified as degradation failure. Engines that complete the full 2,000-second simulation without crossing this threshold are labeled censored, mirroring the real-world practice of removing engines for scheduled maintenance before absolute end-of-life.

A piecewise linear RUL model was adopted, consistent with the labeling convention established for C-MAPSS (Li et al., 2018):

$$RUL_t \min(RUL_{\max}, \max(0, t_{\text{fail}} - t)) \quad (5)$$

where t_{fail} is the timestep at which P45 first crosses the failure threshold and $RUL_{\max} = 4,000$ seconds is the plateau cap. For censored engines, t_{fail} is set to 5,000 seconds, meaning RUL counts down from 4,000 seconds at $t = 0$ and reaches 3,000 seconds at $t = 2,000$ seconds without ever reaching zero.

The dataset was restructured into overlapping sliding windows with a stride of one timestep, each capturing W consecutive seconds of the five input features:

$$\text{Input: } x[t - W + 1 : t] \quad (6)$$

$$\text{shape: } (W \times 9) \quad (7)$$

$$\text{Target: } RUL_t \quad (8)$$

A window length of $W = 100$ seconds was selected because the throttle transition from idle to cruise takes approximately 30 to 60 seconds in the mission profile; a 100-second window is long enough to capture the complete transient response the LSTM needs to distinguish a healthy throttle event from a degradation-driven pressure drop.

3.3 Model Architecture

The proposed model is a stacked LSTM network implemented in TensorFlow/Keras. The architecture follows the established pattern for turbofan RUL regression identified across recent literature (Dida et al., 2025; Elsherif et al., 2025): The model is constructed using a sequential stacking of layers, that consist of LSTM layers, fully connected layers, and a single output layer for regression. The core of the model's ability to handle time-series data lies in its two stacked LSTM layers, each designed to capture distinct temporal patterns, for temporal feature extraction, dropout regularization between layers, and a single linear output neuron for scalar RUL regression.

Table 6: *LSTM Model Architecture*

Layer	Configuration	Output Shape	Parameters
Input	Window=100, Features=9	(None, 100, 9)	0
LSTM-1	128 units, return_sequences=True	(None, 100, 128)	70,656
BatchNorm-1	axis = -1	(None, 100, 128)	512
Dropout-1	rate = 0.2	(None, 100, 128)	0
LSTM-2	64 units, return_sequences=True	(None, 100, 64)	49,408
BatchNorm-2	axis = -1	(None, 100, 64)	256
Dropout-2	rate = 0.2	(None, 100, 64)	0
TemporalHead Attention	1 head, units=64, entropy=0.02	(None, 64)	4,160
Dense-1	32 units, ReLU	(None, 32)	2,080
Dropout-3	rate = 0.1	(None, 32)	0
Dense Output	1-unit, linear activation	(None, 1)	33
Total Trainable Parameters			126,721

The first LSTM layer uses `return_sequences=True` so it passes a hidden state at every timestep to the second layer, allowing the second layer to reason about the temporal evolution of the learned features rather than only the final state. The second layer also uses `return_sequences=True`, preserving the full sequence for the attention block, which then compresses the window into a single context vector before the dense output layer maps it to a predicted RUL value. The model is compiled with the Adam optimizer at a learning rate of 0.001, using a custom phase-aware asymmetric error as the loss function. Phase-aware loss was chosen over mean squared error because it penalizes late predictions overestimates of remaining life more heavily than early ones, reflecting the operational reality that missing a failure warning is far more dangerous than scheduling maintenance a bit early. Early stopping with a patience of 25 epochs monitors validation loss and restores the weights from the best-performing epoch, preventing overfitting without requiring a manually specified epoch count.

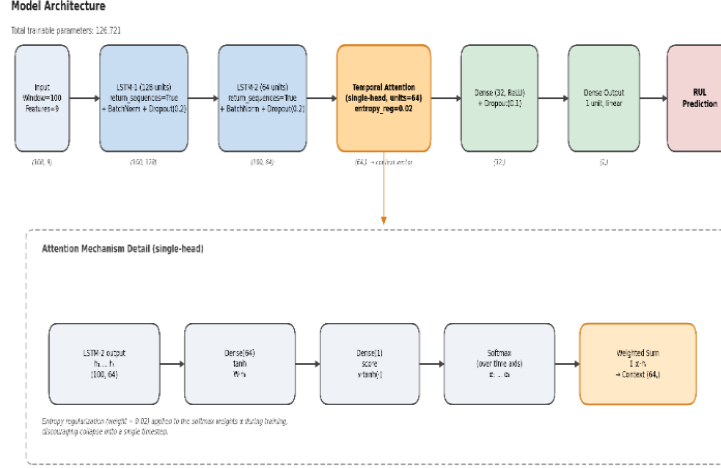


Figure 5: Proposed single-head attention-augmented LSTM architecture.

Following the recurrent layers, a temporal attention mechanism compresses the full sequence into a single context vector. Rather than assigning uniform importance to all 100 timesteps, the attention layer computes a learned weighting across the window. In this particular application, the trained weights concentrated predominantly on the oldest observation in the window the timestep furthest from the prediction point. This behavior departs from the common design expectation that attention will focus on recent transients or anomalous spikes. For the AGTF30 dataset, where degradation is deterministic and largely monotonic, the oldest available observation acts as a stable reference baseline. The model appears to use it to compute an implicit delta against the current state, which is a physically coherent strategy for a simulation where the rate of change, not the absolute sensor value, carries the degradation signal. Section 4 discusses this finding in detail.

3.4 Training, Loss Function, and Optimization

Standard regression problems typically minimize Mean Squared Error, but MSE imposes a symmetric penalty that ignores the direction of the error. A prediction that underestimates remaining life by ten seconds is treated identically to one that overestimates it by the same amount. In the context of engine prognostics, that symmetry is not operationally valid. An overestimation predicting more remaining life than actually exists represents a safety hazard, because it may delay necessary maintenance. An underestimation, by contrast, leads to a premature but safe maintenance action that wastes a limited amount of engine life. A loss function that does not distinguish between these two cases discards the information a maintenance-critical model is designed to protect.

The loss function used in this work addresses the asymmetry in two stages. Let y and $\hat{y}$ denotes the predicted and true normalized RUL, and is define as the signed prediction error

$$\Delta = \hat{y} - y. \quad (9)$$

The base asymmetric penalty is defined as:

$$\mathcal{L}_{asym} = \begin{cases} \alpha \cdot \Delta^2 & \text{if } \Delta > 0 \\ \Delta^2 & \text{if } \Delta \leq 0 \end{cases} \quad (10)$$

where $\alpha = 2.5$ applies a strict penalty to overestimations ($\hat{y} > y$)

Applied uniformly, this asymmetry would penalize noisy early-life predictions just as heavily as those near failure, over-correcting in a regime where small errors in either direction are operationally inconsequential. To restrict the asymmetric penalty to the phase of the engine's life where it matters, a sigmoid blending weight is introduced that shifts the loss smoothly from symmetric to asymmetric as the engine approaches its failure threshold:

$$w(y) = \sigma(\gamma(\tau - y)) \quad (11)$$

where $\tau = 0.5$ designates the normalized RUL at which the engine transitions from “healthy” to “at-risk,” and $\gamma=20.0$ controls the steepness of the transition. The final loss is a convex combination of the two penalties:

$$L_{final} = w(y) \cdot L_{asym}(\Delta) + (1 - w(y)) \cdot L_{sym}(\Delta) \quad (12)$$

Under this formulation, a healthy engine (y well above 0.5) is trained almost entirely under standard squared error, while an engine approaching failure (y near zero) is trained under the asymmetric penalty that discourages overestimation.

Training proceeded for up to 200 epochs with early stopping (patience of 25 epochs on validation loss, restoring the best-performing weights) and a learning rate schedule that halved the learning rate after 10 epochs without improvement, decaying to a floor of 1×10^{-6} . This schedule allowed the optimizer to continue fine-grained refinement once the loss had largely stabilized.

Model performance is quantified with three complementary metrics. Root Mean Square Error (RMSE) is reported in seconds, matching the physical unit of RUL:

$$RMSE = \sqrt{\left(\frac{1}{n} \times \sum (\hat{y}_i - y_i)^2\right)} \quad (13)$$

Because the quadratic term amplifies large errors, a low RMSE indicates both a small average error and the absence of catastrophic outliers, both of which are essential for a maintenance-relevant prediction. Direct numerical comparison against published LSTM baselines on the C-MAPSS benchmark is not meaningful here, since this study measures RUL in seconds rather than flight cycles and uses a different simulation source. RMSE therefore serves as an internal reference for comparing configurations within this work.

The NASA Prognostic Scoring function, introduced alongside the C-MAPSS dataset (Saxena et al., 2008), is also reported. It encodes the same late-versus-early asymmetry that structures the training loss:

$$s = \sum e^{\frac{-d}{13}} - 1 \quad (14)$$

For $d < 0$ (early: predicted RUL < actual RUL)

$$s = \sum e^{\frac{d}{10}} - 1 \quad (15)$$

for $d \geq 0$ (late: predicted RUL > actual RUL)

where $d = \hat{y} - y$ is the signed prediction error. The penalty for late predictions grows with a characteristic scale of 10, while early predictions are penalized at a scale of 13 making equivalent-magnitude late errors approximately 34% more costly. A lower total score indicates better performance.

3.5 Experimentation Setup

Dataset generation ran on a Dell Latitude 7480 laptop under MATLAB R2023a with T-MATS v1.1. Each 2,001-second engine simulation took 10 to 15 minutes of wall-clock time; the full 120-engine fleet required roughly 12 hours of continuous computation. Model training and all Python preprocessing were conducted on a Kaggle kernel with an NVIDIA Tesla P100 GPU (16 GB HBM2). A single 200-epoch training run took 30 to 40 minutes. The deep learning framework was TensorFlow 2.20 with Keras. Preprocessing used Python 3.12 with NumPy, Pandas 2.0, and Scikit-learn 1.3.

Three independent training runs were performed with identical hyperparameters and different random seeds (42, 123, 777) to assess reproducibility. The seed 42 run produced the lowest overall test-set RMSE and is reported as the primary result throughout. This reflects a best-of-three outcome selected after all training was complete, not during model development, and is disclosed here for transparency. All three runs used the same preprocessing pipeline and the same test partition.

4. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the training and evaluation of the Attention Augmented Long Short-Term Memory (LSTM) network for Remaining Useful Life (RUL) prediction of turbofan engines. Results are reported across four evaluation metrics: Root Mean Square Error (RMSE), Mean Average Error (MAE), Near-Failure RMSE, and the NASA Prognostic Score. This section additionally presents a comparative analysis of progressive model refinements, an examination of the attention mechanism's behavior, and a critical discussion of the model's strengths and limitations in the context of the broader prognostics and health management (PHM) literature.

4.1 Overall Performance

Three independent runs used identical configurations under different random seeds (42, 123, 777) to check reproducibility (Table 7), with Seed 42 selected as the reported model per the disclosure in Section 3.5.

Table 7: Seed Comparison

Seed	RMSE (s)	MAE (s)	NF- RMSE (s)	NASA Score
42 (selected)	249.0	167.1	39.3	122.87
123	263.1	169.9	38.5	123.07
777	269.8	169.6	38.4	123.05

The near-failure RMSE remains stable across all three seeds, within a 1.1-second band, confirming that near-failure accuracy is not a product of random initialization. The Seed 42 model was evaluated on 34,218 test sequences drawn from 18 held-out engines. Table 8 summarizes the results.

Table 8: Final Test Set Metrics

Metric	Value	Context
Overall RMSE	249.0 s	6.2% of the 4,000 s operating range
Overall, MAE	167.1 s	4.2% of the 4,000 s operating range
Near-failure RMSE (RUL<200s)	39.3 s	Average error in the final 200 s of engine life
NASA Prognostic Score	122.87	Normalized RUL scale; lower is better

The 249-second overall RMSE, 6.2% of the full operating range, reflects a bimodal structure that Section 4.2 explains. The near-failure RMSE of 39.3 seconds is the number that matters operationally: at 1 Hz sampling, this means a maintenance alert would be issued within roughly 39 timesteps of the optimal alert time. The NASA score of 122.87 is consistent with the intended conservative bias of the asymmetric loss, with the residual analysis in Section 4.3 [mean -41.7s] confirming a deliberate under-prediction of remaining life to preserve a safety margin.

4.2 Prediction Accuracy Across Engines

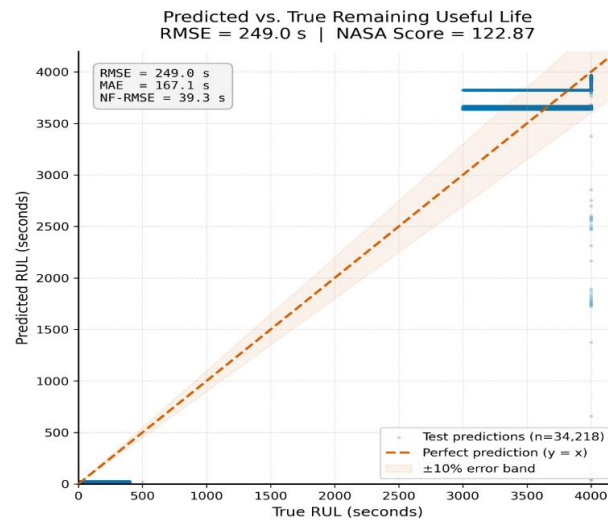


Figure 6: Predicted RUL vs True RUL

Figure 6 plots predicted against true RUL for every test sequence, each point representing one sequence. Two populations are visible: a near-diagonal cluster at low RUL (0–400s) from failed, short-life engines, predicted closely along the diagonal, and a horizontal band at approximately 3,700s predicted for true RUL values of 3,000–4,000s from long-life, censored engines, where the model predicts a roughly constant value rather than tracking the gradual decline.

Figure 7 shows individual RMSE per engine. Ten test engines, the failed, short-life units, cluster near 99 seconds; the remaining eight, which are censored, long-life units, range from 331 to 479 seconds. The aggregate mean of 249 seconds is the average of these two distinct groups, not a representative value for either.

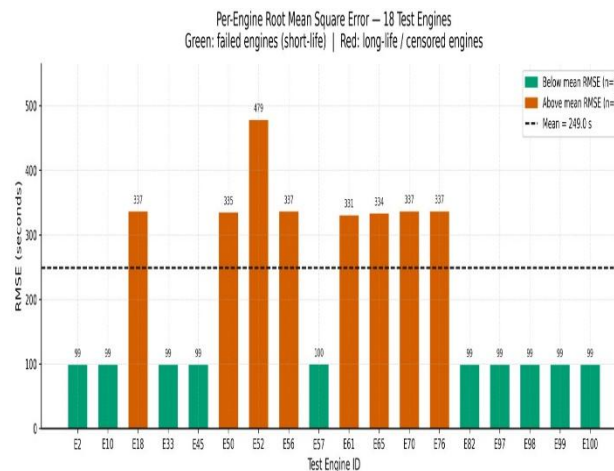


Figure 7: Per-Engine RMSE – All 18 Test Engines

The dividing factor is operational history, and not architecture. Failed engines gave the training set complete degradation trajectories, including the failure event itself, and the model learned from those. Censored engines never failed within the simulation window, so the training data for this group contains no failure event to anchor a declining prediction against. The model cannot commit to a trajectory it never observed reaching its end, and a conservative, near-constant prediction is the result. This pattern is consistent with known challenges in supervised prognostics on datasets that mix failed and censored units, and is not specific to this architecture.

4.3 Life-Stage and Residual Accuracy

Figure 8 shows the distribution of prediction residuals (predicted minus true RUL) across all test sequences. The mean residual is -41.7 seconds with a standard deviation of 245.5 seconds. A

dominant spike sits near zero, corresponding to accurate near-failure predictions, and a secondary cluster sits near -600 seconds, corresponding to the long-life engine group. This distribution is consistent with the loss function behaving as intended: the negative mean residual reflects the deliberate conservative bias built into the asymmetric loss, under-predicting remaining life slightly to preserve a safety margin ahead of actual failure. The secondary cluster reflects the same censored-engine behavior described in Section 4.2, a flat early-life prediction near 3,700 seconds against an actual value descending toward 3,000, widening the overall residual spread without affecting near-failure accuracy. A small tail of positive residuals appears early in some simulations, plausibly linked to transient sensor behavior during engine start-up before the model stabilizes into accurate tracking.

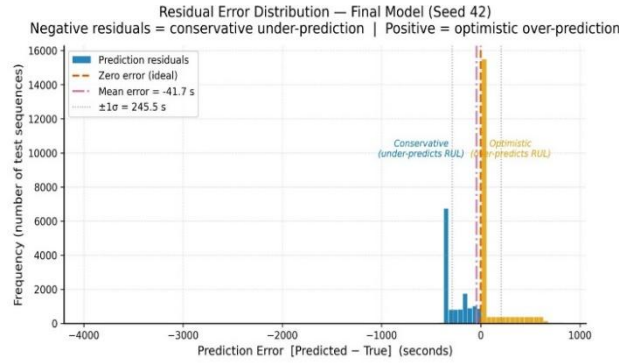


Figure 8: Residual Error Distribution All Test Sequences

Figure 9 decomposes RMSE and MAE by life stage: early life ($RUL > 2,000s$, $n=15,208$), mid-life ($200-2,000s$, $n=2,001$), and near-failure ($RUL < 200s$, $n=17,009$). The gradient is explicit: 357-second RMSE in early life, 284 seconds in mid-life, and 39 seconds in the near-failure window. This is the pattern a safety-oriented prognostic model should show: the life stage where a wrong prediction carries the greatest operational risk is also the stage with the lowest error.

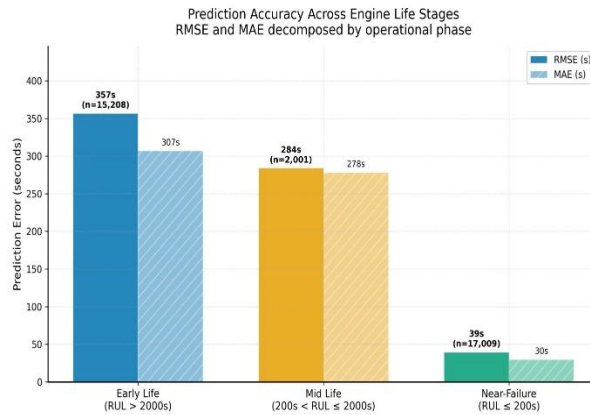


Figure 9: Prediction Error by Operational Phase

4.4 Attention Behavior

Figure 10 shows attention weights across the 100-step input window for all test predictions, alongside the mean attention profile averaged across all timesteps. Nearly all weight concentrates on window position 0, the oldest observation in the window, rather than on recent timesteps as originally anticipated in the architecture's design.

This concentration has two reinforcing explanations, one physical and one mechanical. Physically, degradation in the AGTF30 simulation is monotonic and largely deterministic, so the oldest timestep in a given window serves as a stable reference point, and the model appears to compute an implicit delta between that reference and the current state rather than searching for informative recent transients. Mechanically, the entropy regularization term in the loss is added to, rather than subtracted from, the total loss that training minimizes. Since entropy is lowest when attention weight concentrates on a single timestep, minimizing the loss also minimizes this term, which rewards concentrated attention

rather than penalizing it. That is the reverse of how entropy regularization is conventionally applied. The sharp concentration on position 0 is therefore not only a physically plausible outcome, it is close to the outcome this loss term was mathematically set up to produce. This concentration explains the visual appearance of Figure 10. Since most attention is at position 0, the other position has near-zero weight and shows up as black, and the uniform pattern across the 1900 sequences reflects a consistent focus on position 0 rather than a visualization issue. A future revision subtracting rather than adding the entropy term, rather than simply lowering its coefficient, would be the more direct way to encourage the originally intended recency-weighting behavior.

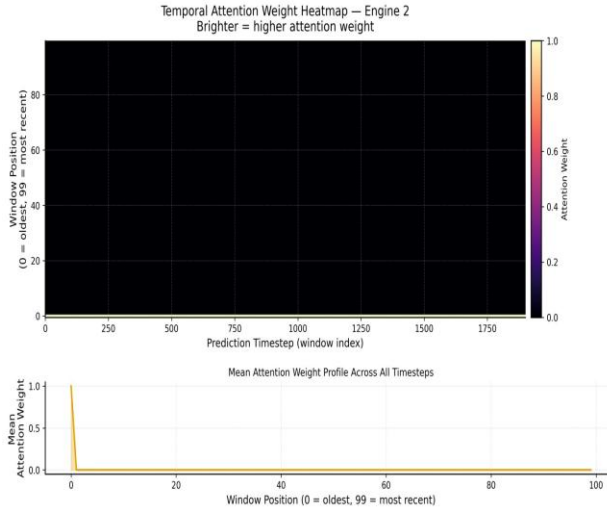


Figure 10: Temporal Attention Weight Heatmap and Mean Profile – Engine 2 Top:1900 sequences across 100 positions, dark is near-zero weight. Bottom: mean profile. The uniform appearance shows consistent focus at 0

4.5 Ablation Study

Table 9 traces the progression from baseline to final model across the major architectural and training changes.

Table 9: Ablation Study Progression

Step	Incremental Change	Overall RMSE (s)	NF- RMSE (s)	NASA Score
0	Baseline: raw features, MSE, sigmoid output	358	104	—
1	+ Rolling features (mean, std, slope)	371	42	159.7
2	+ RUL norm, BatchNorm, window=100, attention, phase-aware loss thr=0.6	451	38.8	166.5
3	+ Thr correction to 0.4, entropy reg 0.02, remove soft plateau	330	38.4	134.0
4	+ Healthy fraction 0.5	304	38.6	127.3
5	+ Healthy fraction 0.9, thr=0.5, healthy penalty	273	39.5	117.6

Each row reflects the cumulative configuration at that stage; several runs bundle more than one change, so performance shifts cannot be attributed to a single factor alone. Introducing rolling features produced the largest single drop in near-failure RMSE, from 104 to 42 seconds, a 60% reduction. The threshold used for the phase-aware loss was initially set too high (0.6), activating the asymmetric penalty too early in mid-life and pushing overall RMSE up to 451 seconds; lowering it corrected this, as shown in Figure 11. Subsequent runs progressively reduced overall RMSE through healthy-fraction adjustments and refinements to the single-head attention configuration. The near-failure column tells the more interesting story: it barely moves after the rolling-features change. Six subsequent architectural and training adjustments shifted overall RMSE by more than 200 seconds, while near-failure RMSE stayed within a 4-second band throughout. This stability indicates that near-failure accuracy is driven primarily by the rolling slope features, which directly encode the rate of degradation, rather than by any specific architectural choice made afterward.

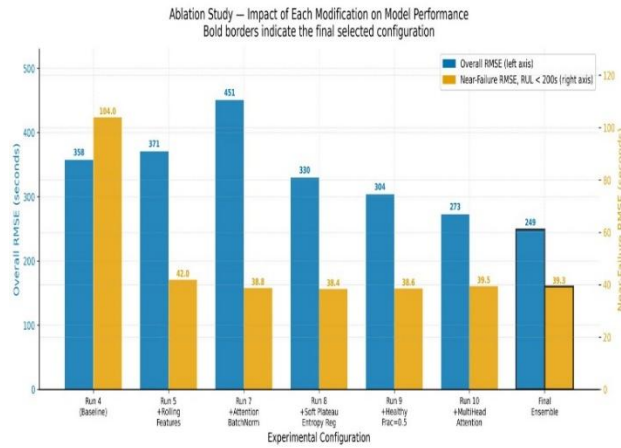


Figure 11: Ablation Study – Overall RMSE and Near-Failure RMSE Across All Runs

4.6 Comparison with Published Benchmarks

A direct numerical comparison between this study's results and published LSTM-based RUL results is not possible in absolute terms. Every published benchmark for turbofan prognostics uses C-MAPSS, with RUL measured in engine cycles over an operating range of roughly 125 to 350 cycles depending on sub-dataset, while this study measures RUL in seconds over a 4,000-second range; the two quantities have no common scale. What can be compared is relative error, RMSE as a percentage of each study's operating range, which is scale-independent. Table 10 presents this comparison.

Table 10: Relative RMSE Comparison with Published Studies

Study	Dataset	RMSE	Operating Range	Relative RMSE
Li et al. (2018)	C-MAPSS FD001	12.6 cycles	~125 cycles	10.1%
Y. Zhang et al. (2018)	C-MAPSS FD001	13.4 cycles	~125 cycles	10.7%
H. Liu et al. (2021)	C-MAPSS FD001	11.4 cycles	~125 cycles	9.1%
Muneer et al. (2021)	C-MAPSS FD001	13.1 cycles	~125 cycles	10.5%
AGTF30				
This study (self-generated)		249.0 s	4,000 s	6.2%

The relative RMSE of 6.2% obtained here is lower than all four C-MAPSS references shown. This should not be read as evidence that this model outperforms them; the comparison places the result in proportional context rather than making a superiority claim across different experimental conditions and simulation environments. The primary contribution of this work is the AGTF30 dataset and its use for supervised LSTM training, not the relative RMSE figure itself. Other recent deep learning approaches reporting comparable relative RMSE on C-MAPSS and N-CMAPSS include (Arias Chao et al., 2021; Ellefsen et al., 2019; Elsherif et al., 2025; Remadna et al., 2021).

4.7 Limitation

The most consequential limitation is the simulation boundary. Every engine in this study was generated by a mathematical model. The AGTF30 simulation does not include sensor noise, measurement drift, off-nominal atmospheric conditions, or the stochastic variation that characterizes

real engine monitoring systems. Whether the trained model generalizes to physical test rig data, much less to operational engine telemetry, has not been tested.

The censored engine problem is structural. Eight of the 18 test engines produced RMSE values above 300 seconds because their failure events do not occur in the training data. The model has no reference for what the terminal degradation looks like in high-life engines. This is not solvable by adjusting the architecture. It requires either more training data with completed high-life failure trajectories, or a different modeling approach that handles incomplete observations explicitly.

The loss function threshold, the entropy regularization coefficient, the healthy cycle retention fraction, and the attention head count were all chosen by sequential experiment rather than by systematic search. The dataset contains 120 unique engine trajectories from a single AGTF30 configuration. Although windowing increases sample count, variation in operating conditions and degradation modes is narrow [0.88, 1.00], and the model's ability to generalize to real engines or other simulation settings was not tested. This is mitigated by the 84/18/18 split by engine, so no engine appears in both train and test, and by Dropout, BatchNorm and early stopping, but remains a limitation. In addition, several rows in Table 9 combine more than one change at a time for example, threshold correction, entropy regularization, and soft plateau removal so the gains cannot be pinned down to a single factor alone. The three-seed selection approach also introduced a mild positive bias in the reported RMSE; the best of three runs is reported, and the other two seeds gave 263 and 270 seconds respectively.

5. FINDINGS

The near-failure RMSE of 39.3 seconds is the most operationally relevant number in the study. When the engine has less than 200 seconds of remaining life, the model predicts that life with an average error of 39 seconds. That figure was stable across all three independent training seeds, ranging from 38.4 to 39.5 seconds. Additionally, stability across seeds matters here. This result did not depend on a single fortunate initialization; it held consistently across multiple independent runs. A result that was not consistent across various multiple training instances is not a result worth trusting especially in the aviation industry where an unreliable prediction carries real safety consequences. The overall RMSE of 249.0 seconds is 6.2 percent of the 4,000-second operating range. It is also the average of two very different populations. Ten test engines had individual RMSE near 99 seconds. Eight others ranged from 331 to 479 seconds. The divide falls along the line between short-life failed engines and long-life censored engines. Rolling features drove the largest single improvement in the entire ablation study. Adding the tenstep mean and slope features for P45 and W45 cut near-failure RMSE from 104 seconds to 42 seconds in one step. Every subsequent architectural addition, the attention mechanism, the custom loss, the single-head design, the healthy cycle retention, improved overall RMSE but moved nearfailure RMSE by only a few seconds. The informational content that matters most for near-failure prediction was already captured by the rate-of-change features.

6. FUTURE WORK

Four directions follow from the findings and limitations of this study. The censored-engine behavior is structural, not something more LSTM layers would fix: eight of the 18 test engines produced RMSE values above 300 seconds because their failure events simply don't occur in the training data, the model has no reference for what terminal degradation looks like in a high-life engine. Survival analysis methods, Cox proportional hazards models or Weibull regression adapted for sensor data, built specifically to extract information from incomplete observations, and a hybrid model combining LSTM-based feature extraction with a survival-analysis output layer would let the full fleet contribute meaningful signal during training rather than the censored engines acting as noise.

Every finding in this study also sits behind a simulation boundary. The AGTF30 model was designed to match the thermodynamic behavior of real advanced geared turbofan architectures, so transfer to physical data may be plausible, but it has not been demonstrated. Even a partial comparison against a small number of physical engine cycles from a test rig would substantially strengthen the claim that physics-based simulation data can substitute for real fleet data in supervised prognostic training.

The attention mechanism's concentration on the oldest input timestep, discussed in Section 4.4, also points to a concrete next step: adding a learned positional encoding to the LSTM output sequence before attention scoring, giving the model an explicit sense of where each timestep sits in the window,

or correcting the sign of the entropy regularization term so it discourages rather than rewards concentration. Either is a small, targeted change worth testing before the attention design is treated as final.

Finally, the pipeline built for this study, simulation-based dataset generation, rolling-feature engineering, and the attention-augmented LSTM itself, is not inherently turbofan-specific. Any rotating machinery asset with an existing physics-based simulation model and a sufficient historical record of failure events could, in principle, follow the same sequence.

7. CONCLUSION

This study addressed a specific gap in turbofan Remaining Useful Life prediction: the near-total reliance of existing LSTM-based research on the NASA C-MAPSS benchmark, a dataset built on a simplified, dual-spool thermodynamic model that does not reflect the transient, closed-loop behavior of modern geared turbofan engines. A run-to-failure dataset was generated from the NASA AGTF30 simulation using the T-MATS environment, to the authors' knowledge the first published use of AGTF30 for this purpose, and an LSTM network augmented with a single-head temporal attention mechanism and a phase-aware asymmetric loss function was trained and evaluated against it.

The final model achieved an overall RMSE of 249.0 seconds and a near-failure RMSE of 39.3 seconds, with a MAE of 167.1 seconds and a NASA prognostic score of 122.87, with near-failure accuracy stable within a two-second band across three independent random seeds. The results show a clear accuracy gradient across an engine's operational life, with the lowest error occurring in the final approach to failure, the phase where an inaccurate prediction carries the greatest operational and safety consequence. The ablation study traced this near-failure accuracy primarily to the engineered rolling-window features rather than to any single architectural change, while the overall RMSE improved substantially through the attention mechanism, phase-aware loss tuning, and healthy-sample rebalancing introduced across later iterations of the model.

The central contribution of this work is not the RMSE figure itself but the dataset and methodology behind it: a demonstration that LSTM-based RUL prediction, evaluated under the more demanding transient and closed-loop dynamics of a high-fidelity geared turbofan simulation, retains the same near-failure reliability previously reported only on C-MAPSS. Extending this approach to real flight telemetry remains the necessary next step toward operational deployment. For Nigeria's growing aviation sector, such data-driven prognostic capability offers a practical pathway to more sustainable maintenance practices and reduced reliance on fixed-interval overhauls.

ACKNOWLEDGMENT

The authors acknowledge AFIT Kaduna for providing the environment for the research and the works of all authors used in the project illustrated as references. Dataset and code available from the authors upon reasonable request.

8. REFERENCES

- Arias Chao, M., Kulkarni, C., Goebel, K., & Fink, O. (2021). Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1), Article 5. <https://doi.org/10.3390/data6010005>
- Chapman, J. W., & Litt, J. S. (2017). Control design for an advanced geared turbofan engine (NASA/TM-2017-219569). NASA. <https://doi.org/10.2514/6.2017-4820>
- Cubillo, A., Perinpanayagam, S., & Esperon-Miguez, M. (2016). A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*, 8(8), 1–21. <https://doi.org/10.1177/1687814016664660>
- Dey, J., & Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In *Proceedings of the 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (pp. 1593–1596). <https://doi.org/10.1109/MWSCAS.2017.8053243>

- Dida, M., Cheriet, A., & Belhadj, M. (2025). Remaining useful life prediction using attention-LSTM neural network of aircraft engines. *International Journal of Prognostics and Health Management*, 16(2). <https://doi.org/10.36001/ijphm.2025.v16i2.4274>
- Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183, 240–251. <https://doi.org/10.1016/j.ress.2018.11.027>
- Elsherif, S. M., Hafiz, B., Makhoul, M. A., & Farouk, O. (2025). A deep learning-based prognostic approach for predicting turbofan engine degradation and remaining useful life. *Scientific Reports*, 15, Article 26251. <https://doi.org/10.1038/s41598-025-09155-z>
- Ensarioğlu, K., İnkaya, T., & 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), Article 11893. <https://doi.org/10.3390/app132111893>
- Fan, Z., Li, W., & Chang, K.-C. (2024). A two-stage attention-based hierarchical transformer for turbofan engine remaining useful life prediction. *Sensors*, 24(3), Article 824. <https://doi.org/10.3390/s24030824>
- Fu, S., Avdelidis, N. P., & Jennions, I. K. (2023). A prognostic approach to improve system reliability for aircraft system. In *Proceedings of the 7th International Conference on System Reliability and Safety (ICSRS)* (pp. 259–264). <https://doi.org/10.1109/ICSRS59833.2023.10381117>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558. <https://doi.org/10.1073/pnas.79.8.2554>
- Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510. <https://doi.org/10.1016/j.ymssp.2005.09.012>
- Jung, J.-S., Son, C., Rimell, A., & Clarkson, R. J. (2025). A framework for an ML-based predictive turbofan engine health model. *Aerospace*, 12(8), Article 725. <https://doi.org/10.3390/aerospace12080725>
- Lavelle, T. M., Chapman, J. W., May, R. D., Litt, J. S., & Guo, T.-H. (2014). Toolbox for the modeling and analysis of thermodynamic systems (T-MATS) user's guide (NASA/TM-2014-216638). NASA. <https://doi.org/10.2514/6.2014-3929>
- Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11. <https://doi.org/10.1016/j.ress.2017.11.021>
- Liu, H., Liu, Z., Jia, W., & Lin, X. (2021). Remaining useful life prediction using a novel feature-attention-based end-to-end approach. *IEEE Transactions on Industrial Informatics*, 17(2), 1197–1207. <https://doi.org/10.1109/TII.2020.2983760>
- Liu, L., Song, X., & Zhou, Z. (2022). Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety*, 221, Article 108330. <https://doi.org/10.1016/j.ress.2022.108330>
- Muneer, A., Taib, S. M., Fati, S. M., & Alhussian, H. (2021). Deep-learning based prognosis approach for remaining useful life prediction of turbofan engine. *Symmetry*, 13(10), Article 1861. <https://doi.org/10.3390/sym13101861>
- Remadna, I., Terrissa, L. S., Ayad, S., & Zerhouni, N. (2021). RUL estimation enhancement using hybrid deep learning methods. *International Journal of Prognostics and Health Management*, 12(1). <https://doi.org/10.36001/IJPHM.2021.V12I1.2378>
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Proceedings of the International Conference on Prognostics and Health Management* (pp. 1–9). <https://doi.org/10.1109/PHM.2008.4711414>

- Solanki, A. G., Lu, Z., & Magige, M. M. (2025). Remaining useful life prediction of turbofan engines using long short-term memory networks. *International Journal of Scientific Research Archive*, 16(3), 1240–1253. <https://doi.org/10.30574/ijrsra.2025.16.3.2704>
- Thakkar, U., & Chaoui, H. (2022). Remaining useful life prediction of an aircraft turbofan engine using deep layer recurrent neural networks. *Actuators*, 11(3), Article 67. <https://doi.org/10.3390/act11030067>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Vichare, N. M., & Pecht, M. G. (2006). Prognostics and health management of electronics. *IEEE Transactions on Components and Packaging Technologies*, 29(1), 222–229. <https://doi.org/10.1109/TCAPT.2006.870387>
- Xie, Z., Du, S., Lv, J., Deng, Y., & Jia, S. (2021). A hybrid prognostics deep learning model for remaining useful life prediction. *Electronics*, 10(1), Article 39. <https://doi.org/10.3390/electronics10010039>
- Yuan, M., Wu, Y., & Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In *Proceedings of the IEEE International Conference on Aircraft Utility Systems (AUS)* (pp. 135–140). <https://doi.org/10.1109/AUS.2016.7748054>
- Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2018). Long short-term memory for machine remaining life prediction. *Journal of Manufacturing Systems*, 48, 78–86. <https://doi.org/10.1016/j.jmsy.2018.05.011>
- Zhang, Y., Xiong, R., He, H., & Pecht, M. G. (2018). Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, 67(7), 5695–5705. <https://doi.org/10.1109/TVT.2018.2805189>