Interpretable RUL Prediction via Hybrid Deep Evidential clustering for Aero-Engine Health Monitoring

Mohamed Ali Ben Azzouna
Universite de Tunis, ISG de Tunis
SMART Lab
Tunis, Tunisia
mohamedali.benazzouna@outlook.com

Safa Ben Ayed
CESI LINEACT, UR7527,
93 Bd de la Seine, Nanterre, 92000,
Ile-de-France, France
sbenayed@cesi.fr

Lilia Rejeb
Universite de Tunis, ISG de Tunis
SMART Lab
Tunis, Tunisie
lilia.rejeb@isg.rnu.tn

***Abstract*—Remaining Useful Life (RUL) prediction is critical for predictive maintenance in safety-critical systems such as aerospace engines. While deep learning models achieve high predictive accuracy, they often lack interpretability and reliable uncertainty estimation. This paper proposes the Hybrid Deep Evidential Clustering (HDEC) framework to address both challenges. A CNN–LSTM–GRU backbone first extracts degradation features from multivariate time-series data. These features are then clustered using NN-EVCLUS, an evidential clustering approach based on Dempster–Shafer theory, which groups engines according to their degradation stage. A dedicated RUL predictor is trained for each cluster to enable specialized and interpretable predictions. Engines with uncertain cluster membership are handled through soft memberships, allowing RUL estimation as a weighted combination of cluster-specific predictors instead of hard assignment. Experiments on the NASA C-MAPSS dataset demonstrate that HDEC improves predictive performance while providing well-calibrated uncertainty estimates and interpretable degradation-regime assignments.**

***Index Terms*—Remaining Useful Life, Evidential Clustering, Dempster-Shafer Theory, Deep Learning, Prognostics and Health Management, Uncertainty Quantification**

I. INTRODUCTION

In aerospace and industrial systems, unplanned component failures can lead to severe safety risks and significant economic losses. Predictive Maintenance (PdM) addresses this challenge by estimating the Remaining Useful Life (RUL) of a component, which refers to the number of operational cycles remaining before a predefined failure threshold is reached, thereby enabling timely and cost-effective interventions [1].

RUL prediction for turbofan engines remains challenging due to nonlinear degradation dynamics, high-dimensional sensor measurements, and varying operating conditions. Data-driven deep learning approaches have become dominant for this problem [2]. Hybrid architectures combining Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) networks have shown strong performance on the NASA C-MAPSS benchmark [3]–[6]. CNNs capture local spatial correlations among sensor channels, while LSTM and GRU layers model long-term temporal dependencies in degradation trajectories.

Despite their strong predictive performance, three key limitations remain. First, most deep learning models produce deterministic point estimates without explicitly quantifying uncertainty, preventing assessment of prediction reliability in safety-critical applications. Second, existing approaches typically rely on a single global model and assume homogeneous engine behavior at inference time, neglecting structural heterogeneity across operating regimes. Third, current methods lack intrinsic interpretability. While post-hoc techniques identify influential features, they do not explain the degradation regime associated with a prediction.

Prior work addresses these limitations only partially. Evidential Deep Learning (EDL) methods [7] provide uncertainty estimates but do not capture structural heterogeneity across engines. Bayesian deep learning [8] estimate predictive distributions but is computationally expensive and lacks explicit health-regime decomposition. Regime-aware approaches [9] identify operating clusters through unsupervised preprocessing, improving robustness to distribution shifts, but do not propagate uncertainty into RUL estimates. Attention-based models [10] and evidential similarity approaches [11] improve either accuracy or interpretability, but provide neither per-engine uncertainty nor intrinsic regime assignments. Particle-filter methods [12] propagate output uncertainty without clustering engines into degradation regimes. Although belief function theory [13], [14] has been formalized for Prognostics and Health Management (PHM) decision-making under uncertainty [15], its integration with deep feature extraction for joint uncertainty quantification and regime discovery remains largely unexplored.

To address these gaps, we propose HDEC, a Hybrid Deep Evidential Clustering framework. HDEC integrates a CNN–LSTM–GRU feature extractor with NN-EVCLUS, a Dempster–Shafer-based Siamese network [16], to assign each engine a mass function over K learned clusters. Each cluster represents a distinct degradation regime and is associated with a dedicated RUL predictor. For engines with ambiguous cluster membership, HDEC produces predictions through an uncertainty-weighted combination of cluster-specific models.

HDEC offers three advantages. First, it provides per-engine uncertainty estimates for reliability-aware decision-making. Second, it captures engine heterogeneity across health stages through cluster-specific modeling. Third, it delivers intrinsic interpretability by linking predictions to learned degradation regimes without relying on post-hoc explanation methods.

The remainder of this paper is organized as follows. Section II presents the proposed HDEC framework. Section III describes the experimental setup and results. Section IV concludes the paper.

II. The HDEC Framework

HDEC operates in two stages. In Stage 1, a deep feature extraction backbone is trained to encode sensor windows into compact representations. In Stage 2, an evidential clustering model partitions these representations into K clusters, where each cluster groups engines exhibiting similar health-stage characteristics along a common degradation progression. A dedicated RUL predictor is then trained for each cluster, allowing each model to specialize in a specific type of degradation dynamics.

At inference time, engines with uncertain cluster membership are not forced into a single cluster. Instead, their RUL estimate is calculated as a weighted combination of the predictions from all cluster-specific models, with weights determined by the engine's degree of affiliation to each cluster. This two-stage pipeline, from preprocessing and feature extraction to clustering and RUL estimation, enables simultaneous accuracy, uncertainty quantification, and interpretable degradation-regime assignments.

A. Dataset and Preprocessing

The NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) benchmark [17] is a widely used dataset for turbofan engine degradation, comprising four subsets (FD001–FD004) that vary in the number of operating conditions (1 or 6) and fault modes (1 or 2), with training trajectories ranging from 100 to 249 engines per subset.

Experiments are conducted on NASA's C-MAPSS FD004, the most operationally complex subset, containing six operational conditions and two fault modes. Its multi-condition heterogeneity makes it a suitable benchmark for evaluating uncertainty estimation and health-regime partitioning.

Of the 21 sensor channels, seven with near-zero variance are removed. RUL labels are clipped at $R_{\mathrm{thr}} = 130$ cycles [18], meaning any true RUL exceeding this threshold is set to 130. Formally:

$$R_{\mathrm{label}} = \begin{cases} R_{\mathrm{real}}, & \text{if } R_{\mathrm{real}} \leq 130 \\ 130, & \text{if } R_{\mathrm{real}} > 130 \end{cases} \tag{1}$$

This clipping prevents extremely high RUL values from skewing training. Sensor readings are normalized per operational condition using min-max scaling to account for differences in engine speed and load. Measurement noise is reduced with exponential smoothing ($\alpha = 0.3$). Time series are segmented into overlapping windows of 40 cycles with a stride of 1. Sequences shorter than 40 cycles are zero-padded to maintain consistent input dimensions. Preliminary validation experiments indicated that 40-cycle windows outperform the commonly used 30-cycle windows [18], resulting in 51,538 training windows.

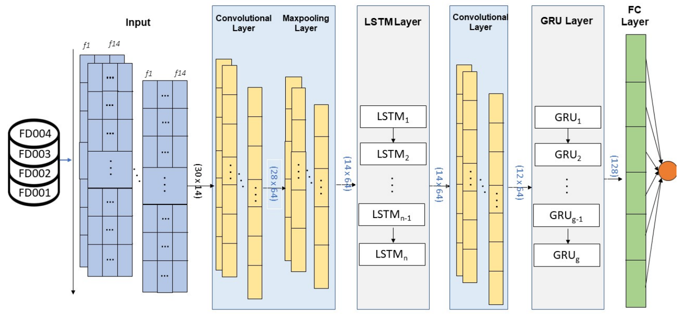

Fig. 1. Hybrid CNN-LSTM-GRU architecture for feature extraction

B. CNN-LSTM-GRU Feature Extraction Backbone

Each 40-cycle sensor window is processed through a cascade of spatial and temporal layers as shown in Fig. 1. A first 1D-CNN with max-pooling extracts localized temporal patterns and cross-channel correlations at each time step by applying convolution along the time dimension [18]. A LSTM layer captures long-range temporal dependencies such as gradual degradation trends [4]. A second 1D-CNN recovers spatial structure at a higher level of abstraction, followed by a GRU layer that models short-term temporal dynamics with fewer parameters [19].

The final GRU output at the last time step forms a 64-dimensional feature vector $\mathbf{f} \in \mathbb{R}^{64}$, which is passed to NN-EVCLUS for clustering. All layers use 64 filters or units (kernel size 3, ELU activation). Two fully connected layers (128→64 units) map features to the RUL estimate. The backbone is trained with the Adam optimizer (batch size 64, 13 epochs). Both batch size and epoch count were determined via engine-level 5-fold cross-validation to prevent data leakage.

C. Evidential Clustering with NN-EVCLUS

The core novelty of HDEC lies in integrating NN-EVCLUS into the RUL pipeline, producing per-engine mass vectors over K learned cluster hypotheses. The pipeline begins with the CNN-LSTM-GRU backbone, which extracts 64-dimensional feature vectors from 40-cycle sensor windows. These features are then processed by NN-EVCLUS, a Dempster–Shafer based evidential clustering network that maps each vector to a mass function over K clusters, where K denotes the number of latent clusters defined in the model. Each cluster represents a distinct degradation regime learned independently of predefined operating condition labels. Finally, each cluster has a dedicated RUL predictor that generates the predicted remaining useful life.

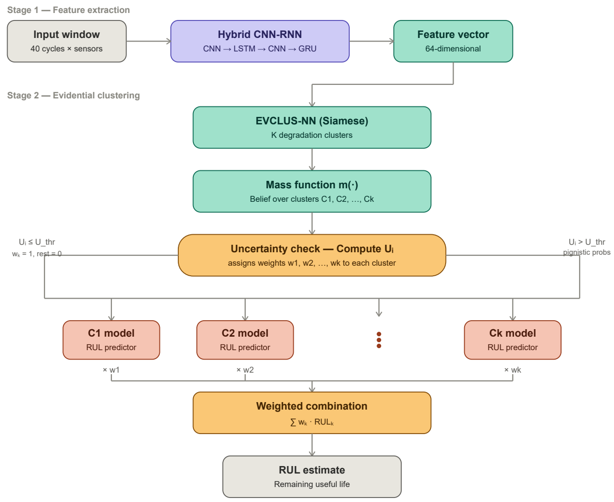

Fig. 2. HDEC Architecture Overview

As shown in Fig. 2, engines with ambiguous cluster membership i.e., whose mass is distributed across multiple clusters are not forced into a single cluster. Instead, their RUL prediction is obtained by blending the outputs of all cluster-specific predictors using pignistic probabilities, which convert the belief masses into a probabilistic estimate.

Each cluster-specific predictor is trained exclusively on the engine time windows assigned to its respective cluster through a hard assignment strategy, whereby each window is allocated to the cluster associated with the highest pignistic probability, $\arg\max_k \mathrm{BetP}(C_k)$. This enables each predictor to specialize in the degradation dynamics and temporal patterns associated with the corresponding health state.

1) Belief Function Theory: Dempster-Shafer Theory (DST) [13], [14] generalises probability theory by allowing belief mass to be distributed over *sets* of hypotheses rather than individual events, making it naturally suited to represent epistemic uncertainty. Let $\Omega = \{C_1, \ldots, C_K\}$ denote the frame of discernment, where each element corresponds to a distinct degradation cluster. A Basic Belief Assignment (BBA) $m : 2^\Omega \to [0, 1]$ distributes unit masses across all subsets of Ω, called focal sets, as given in Eq. (2).

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{2}$$

The mass on the singleton $\{C_k\}$ expresses strong belief that an engine belongs to cluster C_k; mass on a non-singleton $A \subsetneq \Omega$ captures genuine uncertainty about which cluster applies. To convert BBAs into probabilities for downstream prediction, we apply the pignistic transformation [20], as defined in Eq. (3).

$$BetP(C_k) = \sum_{\substack{A \subseteq \Omega \\ A \neq \emptyset}} \frac{m(A)\, \mathbf{1}_{C_k \in A}}{|A|} \tag{3}$$

where $\mathbf{1}_A(C_k) = 1$ if $C_k \in A$, and 0 otherwise.

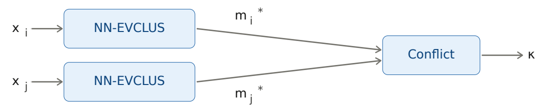

Fig. 3. NN-EVCLUS Siamese architecture.

2) NN-EVCLUS Formulation: NN-EVCLUS [16] is a Siamese neural network that maps feature vectors to mass vectors. The network is trained to minimize the discrepancy between the pairwise cluster conflict and the transformed feature dissimilarity. The conflict between engines i and j is defined as in Eq. (4):

$$\kappa_{ij}(\theta) = \mathbf{m}_i^\top \mathbf{C}\, \mathbf{m}_j, \tag{4}$$

where θ denotes the parameters of the NN-EVCLUS model, and $\mathbf{m}{*}i$ and $\mathbf{m}{*}j$ are the BBAs of engines i and j. The matrix entry $C*qr = \mathbf{1}(F_q \cap F_r = \emptyset)$ encodes conflict between focal sets F_q and F_r, where F_q and F_r are subsets of Ω referred to as focal sets. The transformed feature dissimilarity $\delta^* * ij$ is defined as a scaled Euclidean distance between the feature vectors of engines i and j, and is used in the loss formulation in Eq. (5).

$$\mathcal{L}(\theta) = \frac{1}{n(n-1)} \sum_{i<j} \left(\kappa_{ij}(\theta) - \delta^*_{ij}\right)^2 \tag{5}$$

where n is the number of engines in the training batch. Training uses mini-batch stochastic gradient descent, where $s = 30$ randomly sampled pairs of engines are used per epoch to compute the loss. The Siamese structure of NN-EVCLUS is illustrated in Fig. 3.

3) Uncertainty score and inference pipeline.: For each test engine i, the uncertainty score is defined as:

$$U_i = 1 - \max_k m(\{C_k\}) \tag{6}$$

where $m(\{C_k\})$ is the belief mass assigned to cluster C_k. The term $\max_k m(\{C_k\})$ represents the highest confidence assigned to any singleton focal set. When this value is high, the engine is strongly associated with one cluster, indicating low uncertainty. Conversely, when the mass is distributed across multiple clusters, the maximum value decreases, leading to higher uncertainty.

Engines with $U_i > 0.6$ are considered uncertain, where the threshold is determined via validation on the training set. At test time, if $U_i \leq 0.6$, the prediction from the most confident cluster is used. Otherwise, predictions from all K clusters are combined using pignistic probabilities:

$$\hat{y}_i = \sum_{k=1}^{K} p^{\mathrm{bet}}(C_k) \cdot \hat{y}_{k,i} \tag{7}$$

where $\hat{y}_{k,i}$ is the RUL prediction for engine i produced by the model associated with cluster C_k. The number of clusters

TABLE I
RMSE AND S-SCORE COMPARISON ON FD004

Method	RMSE	S-score
DAG [3]	22.43	3370
CNN-BDGRU [5]	18.29	2304
LK-CNN-Markov [22]	13.00	3805
CiRNN [23]	16.16	803
Backbone (CNN-LSTM-GRU)	15.90	1898
HDEC-CM	16.34	1370
HDEC-EP	**15.03**	**990**

$K = 3$ is selected via grid search over $K \in \{2, \dots, 5\}$. Lower values reduce cluster separability, while higher values lead to over-partitioning and degraded prediction performance.

III. EXPERIMENTS AND RESULTS

A. Experimental Setup and Metrics

All models were implemented in PyTorch and evaluated with 5-fold engine-level cross-validation on an NVIDIA RTX 4070 Ti GPU. Two inference strategies are evaluated: HDEC-CM (Crisp Membership), which assigns each engine to its most confident cluster, and HDEC-EP (Evidential Pooling), which computes RUL as a pignistic-weighted combination of all cluster-specific predictors for engines with uncertain membership ($U_i > 0.6$).

Root Mean Squared Error (RMSE) quantifies overall prediction accuracy, penalizing large errors quadratically. The asymmetric S-score [21] penalizes late predictions (underestimating RUL) more heavily than early ones, reflecting the higher operational cost of missed failures, as defined in Eq. (8):

$$\mathrm{S} = \begin{cases} \sum_{i=1}^{n} \left(e^{-d_i/13} - 1 \right), & d_i < 0 \\ \sum_{i=1}^{n} \left(e^{d_i/10} - 1 \right), & d_i \geq 0 \end{cases} \quad \text{with} \quad d_i = \hat{y}_i - y_i. \tag{8}$$

Reporting both metrics is important: a model could have low RMSE yet systematically over-predict RUL (i.e., late predictions), resulting in high S-score and increased operational risk.

B. Baseline Comparison

Table I presents the RMSE and S-score of HDEC against four established baselines and the standalone backbone on FD004. HDEC-CM (RMSE 16.34) outperforms DAG [3] by 26.9% and CNN-BDGRU [5] by 10.7%. HDEC-EP further reduces RMSE to 15.03, confirming that the blending strategy adds quantitative value beyond hard cluster assignment.

Three observations in Table I deserve explicit attention. First, HDEC-CM (RMSE 16.34) is marginally worse than the standalone backbone (RMSE 15.90) in raw accuracy. This is not a failure of clustering: it reflects the fact that cluster-specific models are trained on engine subsets rather than the full training set, reducing each predictor's sample size. The accuracy deficit is recovered and surpassed by the HDEC-EP (15.03), and more importantly, the HDEC-CM already achieves a strong S-score among all methods (1370 vs. 1898 for the backbone), meaning the clustering introduces a beneficial conservative bias. The system tends to predict RUL slightly earlier rather than later, which is the operationally safer error direction. Second, the S-score improvement from HDEC-CM to HDEC-EP (1370→990) is proportionally larger than the RMSE improvement (16.34→15.03), indicating that the weighted blending is particularly effective at correcting the late-prediction errors that the asymmetric S-score penalizes most heavily. Third, HDEC-EP achieves competitive RMSE while delivering the best S-score among all compared methods except CiRNN [23] (803 vs. 990). However, CiRNN provides neither uncertainty quantification nor health-regime partitioning. LK-CNN-Markov [22] achieves a lower RMSE (13.00) but at the cost of a dramatically worse S-score (3805 vs. 990), indicating systematic late-prediction bias that is operationally dangerous. HDEC-EP is the only method that combines competitive accuracy, a strong S-score, explicit per-engine uncertainty signals, and interpretable health-regime assignments.

TABLE II
HDEC PER-CLUSTER RESULTS ON FD004 ($K = 3$)

	Cluster 1	Cluster 2	Cluster 3
Engines (test)	113	65	70
RMSE	17.64	24.72	**10.71**
Confident ($U_i \leq 0.6$)	103	14	64
Uncertain ($U_i > 0.6$)	10	51	6
Median U_i	0.16	0.65	0.35

C. Cluster Structure and Physical Interpretation

As shown in Fig. 4, Clusters 1 and 3 demonstrate high confidence, with median U_i of 0.16 and 0.35 respectively, and the majority of their engines remain below the flagging threshold. Cluster 2 differs fundamentally: median $U_i = 0.65$ exceeds the threshold, with 51 of 65 engines (78.5%) flagged. This concentration of uncertainty in Cluster 2 is consistent with the evidential framework: NN-EVCLUS distributes belief mass across multiple clusters when feature vectors fall in ambiguous regions of the representation space, naturally producing higher uncertainty scores for engines that cannot be confidently assigned to any single stable regime. NN-EVCLUS assigns higher uncertainty to Cluster 2 engines because their degradation patterns overlap multiple regimes, leading to distributed belief mass across clusters rather than a single confident assignment. Prediction accuracy in Fig. 4 (right) mirrors the uncertainty pattern precisely. Cluster 3 achieves RMSE 10.71 substantially below the dataset baseline and 37% better than the standalone backbone. Cluster 1 records moderate performance (RMSE = 17.64), consistent with engines in a stable but less distinctive operating regime. Cluster 2 shows the highest error (RMSE = 24.72), confirming that flagged engines are also the hardest to predict.

The physical interpretation of unsupervised cluster assignments is striking. **Cluster 3** represents late-stage degradation, where rotor speed, temperature, and pressure ratio

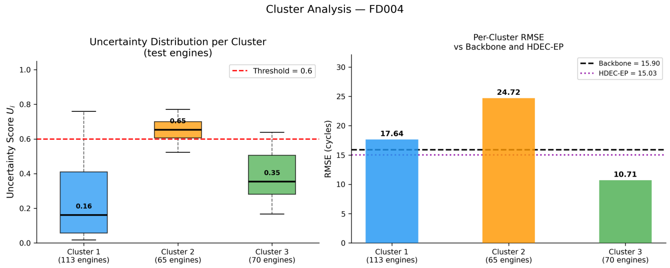

Fig. 4. Cluster-wise uncertainty distribution and prediction accuracy on FD004

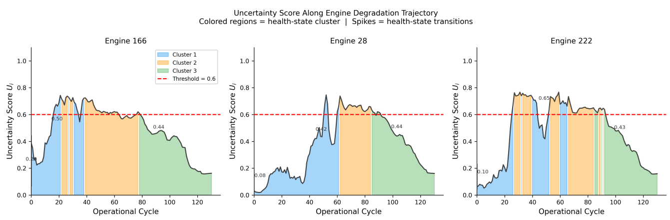

Fig. 5. Uncertainty scores U_i along three engine degradation trajectories.

trend strongly and monotonically toward end-of-life. These consistent, narrowing signatures enable accurate prediction because late-stage degradation follows a convergent pattern across engines. The model trained on Cluster 3 leverages this consistency for reliable remaining-life estimation.

Cluster 1 captures engines in earlier, stable operation with clear but uniform signatures. This yields moderate accuracy (RMSE = 17.64) and high confidence (median $U_i = 0.16$).

Cluster 2 represents the operationally challenging regime: engines actively transitioning between stable operation and active degradation. At this mid-life transition point, overlapping degradation processes produce genuinely ambiguous signatures that cannot be attributed to any single stable regime. NN-EVCLUS responds appropriately by distributing belief across clusters rather than committing to one representing real epistemic uncertainty, not a modeling artifact.

D. Uncertainty Trajectory Validation

The per-engine uncertainty score U_i is more than a static test-time label. It reflects the actual evolution of the engine health. Fig. 5 shows U_i along three representative engine degradation trajectories from the training set. These trajectories were chosen to illustrate the typical uncertainty evolution pattern observed across the fleet. The background color indicates cluster assignment at each window. We use the training set here because it contains complete run-to-failure trajectories, which are not available in the truncated test set.

In all three engines, U_i follows a consistent and physically meaningful pattern. It stays low during the initial stable operating phase (Cluster 1), rises sharply as the engine enters the mid-life transition (Cluster 2), and then decreases again as the engine stabilizes into the late-degradation regime (Cluster 3). Cluster assignments shift in alignment with the U_i spikes. This confirms that the uncertainty signal tracks real structural

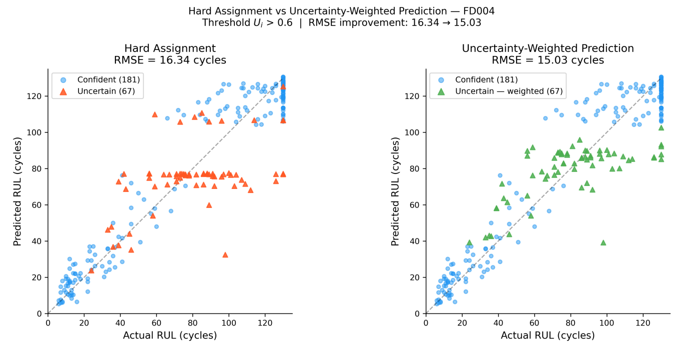

Fig. 6. Comparison of Actual and Predicted RUL under HDEC-CM and HDEC-EP.

changes in the degradation process, not noise or arbitrary model variation.

This temporal validation shows that U_i can serve as a *dynamic health-transition indicator*. A rising U_i signals that an engine is entering a new degradation phase, precisely when maintenance attention is most needed.

E. Impact of Uncertainty-Weighted Prediction

Fig. 6 compares the prediction scatter for all 248 FD004 test engines under two strategies: HDEC-CM (RMSE 16.34) and HDEC-EP (RMSE 15.03). Confident engines (circles, n=181) behave similarly under both strategies, showing that weighted blending does not degrade well-assigned engines. The main difference appears in the 67 uncertain engines (triangles).

Under HDEC-CM, the 67 uncertain engines (orange triangles) collapse into a near-flat band around 40–60 predicted cycles. Their actual RUL ranges from near zero to 120 cycles. This flat-band effect occurs because mid-transition engines are forced into a single cluster, even though their features are split between two regimes. The assigned cluster model applies a regime-specific mapping that does not fit these engines, compressing their predicted RUL toward the cluster's central tendency.

HDEC-CM (green triangles, right panel) mitigates this effect. Predictions move closer to the ideal diagonal by combining cluster-specific outputs using pignistic probability weights associated with each cluster. This correction is partial, as expected: mid-transition engines are inherently difficult to predict, and blending cannot recover information absent from the sensor signal.

Fig. 7 quantifies per-engine improvements for the 67 uncertain engines. Of these, 40 (59.7%) saw reduced error under the weighted strategy, with a mean reduction of 1.9 cycles. The remaining 40.3% showed no improvement, which is expected due to the genuine unpredictability of mid-transition engines. Improvement is not uniform: Cluster 2 engines (squares, n=51) show the most consistent gains, while the remaining 16 engines from Clusters 1 and 3 show mixed results. This pattern confirms that blending is most effective for engines in the operationally ambiguous mid-life transition and does not harm engines where HDEC-CM already works well.

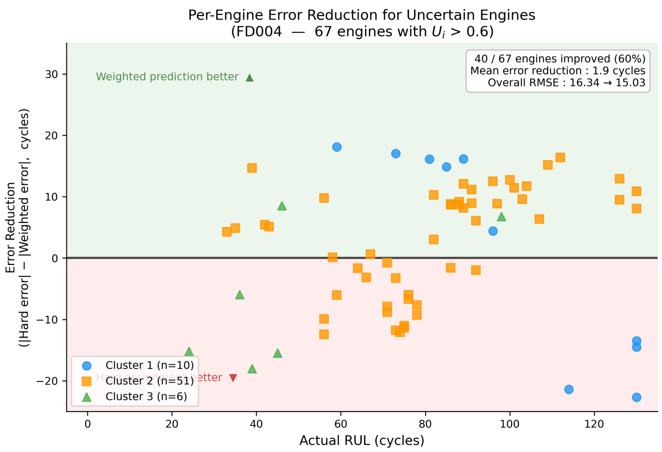

Fig. 7. Impact of HDEC-EP on Uncertain Engine RUL Errors

Beyond prediction accuracy, the U_i signal provides additional information on engine health-state uncertainty. For the 67 flagged engines (27.0% of the test fleet), deterministic RUL estimates may not fully reflect the ambiguity associated with transitional degradation regimes.

This guidance comes directly from the scalar U_i, which is produced at no extra cost during the NN-EVCLUS forward pass. Unlike a standard confidence interval, which only communicates statistical uncertainty, a value of $U_i > 0.6$ indicates that an engine is actively transitioning between degradation regimes. This is a qualitatively different signal, because it highlights engines that are entering a new phase of wear and may require attention sooner.

IV. Conclusion

This paper presents HDEC, a framework combining deep feature extraction and evidential clustering for interpretable, uncertainty-aware RUL prediction in aerospace engines. HDEC identifies transitional health states through per-engine uncertainty estimates and cluster-specific degradation modeling.

Experiments on C-MAPSS FD004 show that HDEC achieves the best combination of predictive accuracy and uncertainty-aware metrics. By jointly coupling clustering with uncertainty quantification, it surpasses regime-aware preprocessing [9] and attention-based methods [10]. Analysis reveals coherent degradation regimes: late-stage engines are predicted with high confidence, while transitional engines exhibit higher uncertainty, confirming U_i as a meaningful degradation-phase indicator. Limitations include evaluation on simulated data and fixed offline cluster counts.

Future work will investigate transformer- and Mamba-based backbones, online cluster adaptation, and validation on real-world datasets.

References

[1] Y. Lei, N. Li, S. Gontarz, J. Lin, S. Radkowski, and J. Dybala, "A model-based method for remaining useful life prediction of machinery," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1314–1326, 2016.

[2] B. A. Ture, A. Akbulut, A. H. Zaim, and C. Catal, "Stacking-based ensemble learning for remaining useful life estimation," *Soft Computing*, pp. 1–13, 2023.

[3] J. Li, X. Li, and D. He, "A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction," *IEEE Access*, vol. 7, pp. 75 464–75 475, 2019.

[4] M. Xia, X. Zheng, M. Imran, and M. Shoaib, "Data-driven prognosis method using hybrid deep recurrent neural network," *Applied Soft Computing*, vol. 93, p. 106351, 2020.

[5] J. Zhang, J. Tian, M. Li, J. I. Leon, L. G. Franquelo, H. Luo, and S. Yin, "A parallel hybrid neural network with integration of spatial and temporal features for remaining useful life prediction in prognostics," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2022.

[6] W. Chen, C. Liu, Q. Chen, and P. Wu, "Multi-scale memory-enhanced method for predicting the remaining useful life of aircraft engines," *Neural Computing and Applications*, vol. 35, no. 3, pp. 2225–2241, 2023.

[7] S. B. Ayed, R. S. Broujeny, and R. T. Hamza, "Remaining useful life prediction with uncertainty quantification using evidential deep learning," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 15, no. 1, pp. 37–55, 2025.

[8] L. Zhuang, A. Xu, and X.-L. Wang, "A prognostic driven predictive maintenance framework based on Bayesian deep learning," *Reliability Engineering & System Safety*, vol. 234, p. 109181, 2023.

[9] A. Aasi, S. E. Bajgani, and O. Mahdavi, "Condition-aware ai for predictive maintenance: Dual-attention cnn–gru with per-regime scaling," *Expert Systems with Applications*, p. 131582, 2026.

[10] C. Liu, L. Zhang, J. Niu, R. Yao, and C. Wu, "Intelligent prognostics of machining tools based on adaptive variational mode decomposition and deep learning method with attention mechanism," *Neurocomputing*, vol. 417, pp. 239–254, 2020.

[11] F. Cannarile, P. Baraldi, and E. Zio, "An evidential similarity-based regression method for the prediction of equipment remaining useful life in presence of incomplete degradation trajectories," *Fuzzy Sets and Systems*, vol. 367, pp. 36–50, 2019.

[12] K. Peng, R. Jiao, J. Dong, and Y. Pi, "A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter," *Neurocomputing*, vol. 361, pp. 19–28, 2019.

[13] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *The Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 325–339, 1967.

[14] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.

[15] S. B. Ayed, M. B. Khalifa, and S. Ouchani, "Modeling distributed and flexible phm framework based on the belief function theory," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2024, pp. 160–173.

[16] T. Denoeux, "NN-EVCLUS: Neural network-based evidential clustering," *Information Sciences*, vol. 572, pp. 297–330, 2021.

[17] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management (PHM08)*. Denver, CO, USA: IEEE, 2008, pp. 1–9.

[18] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[19] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.

[20] P. Smets, "The transferable belief model for quantified belief representation," in *Quantified Representation of Uncertainty and Imprecision*. Springer, 1998, pp. 267–301.

[21] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, "Metrics for evaluating performance of prognostic techniques," in *2008 international conference on prognostics and health management*. IEEE, 2008, pp. 1–17.

[22] Y. Wang, C. Su, P. Wang, J. Zhen, and D. Wang, "A hybrid large-kernel cnn and markov feature framework for remaining useful life prediction," *Machines*, vol. 14, no. 1, p. 57, 2026.

[23] R. Dutta Baruah and M. Munoz-Organero, "Recurrent neural network with contextual attention for prognostics," *Evolving Systems*, vol. 16, no. 4, p. 117, 2025.