Development of a Remaining Useful Life Estimation Method Using Transformer and a Reweighting Technique

Gyeongho Kim[1] **Sunghoon Lim**[1,2]

[1]Department of Industrial Engineering, Ulsan National Institute of Science and Technology (UNIST)
[2]Industry Intelligentization Institute, Ulsan National Institute of Science and Technology (UNIST)

Introduction

- **Abstract**:

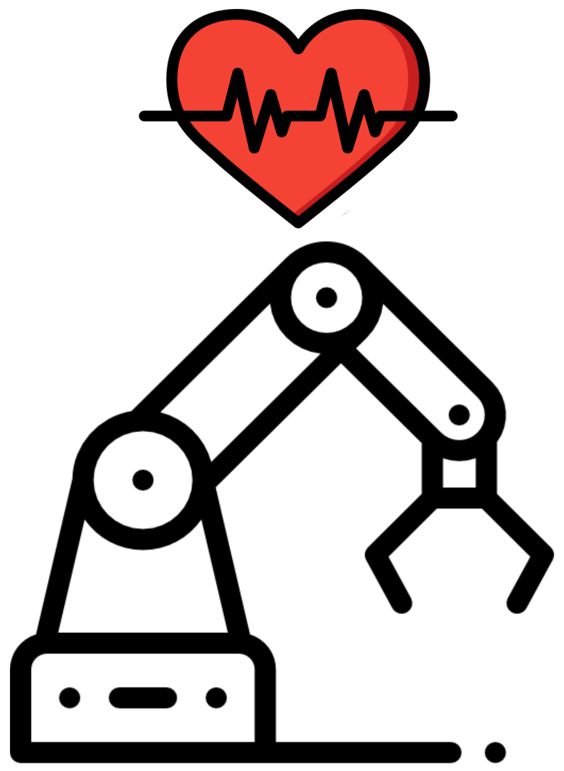
  - Monitoring and estimating the remaining useful life (RUL) of industrial machinery are important tasks in the field of prognostics and health management (PHM). Based on current health status and real-time sensor measurements, RUL is estimated to provide time left for potential failures so that repair and replacement can be planned in advance to improve the productivity and efficiency of machinery operations. Considering the nature of machinery operation time, there exists a data imbalance problem in the existing RUL estimation dataset, such as the C-MAPSS dataset. As the data imbalance problem can have detrimental effects on the performance of RUL estimation methods, this research addresses the problem by proposing an adaptive reweighting technique that rebalances the effects of samples with different ground-truth RULs. In addition, this research proposes a novel estimation method that uses a transformer architecture, which is effective in the C-MAPSS dataset. For subsets FD001 and FD003, the proposed method shows the state-of-the-art estimation performance with the highest RMSE and score values of 11.39, 194.08, and 11.29, 144.02, respectively.

- **Keywords**:
  - Industrial Applications, Prognostics and Health Management, Remaining Useful Life Estimation, Deep Learning

CONTENTS

Introduction

- Prognostics and health management (PHM)
  - Industrial machinery, electronics
  - Automated maintenance
  - Increased efficiency, cost saving



- Remaining useful life (RUL) estimation
  - Accurate prediction of remaining life cycles
  - Key aspects:
    - Using raw measurements from multiple sensors
    - Feature selection
    - Feature extraction, fusion

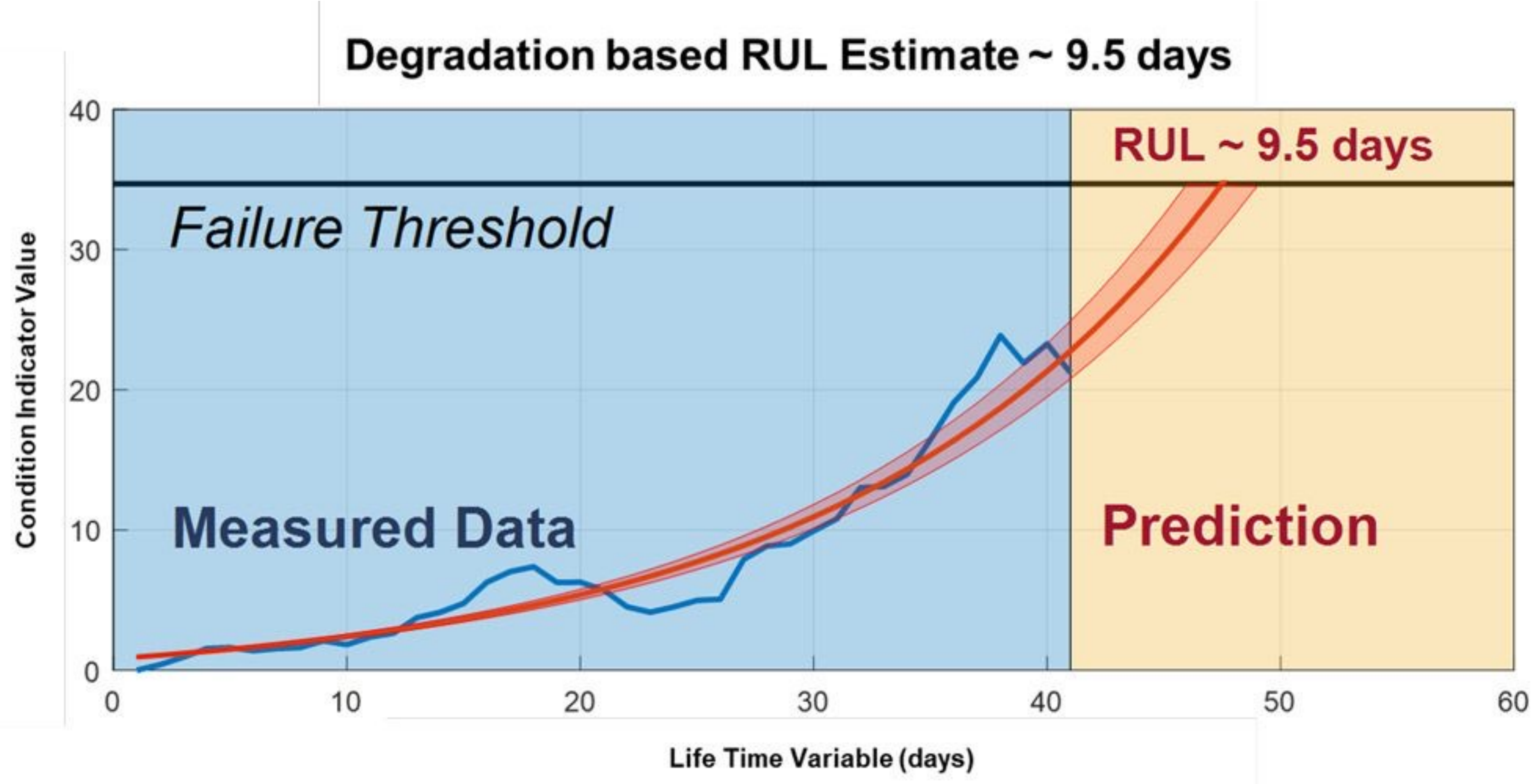


Figure from: https://kr.mathworks.com/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance.html

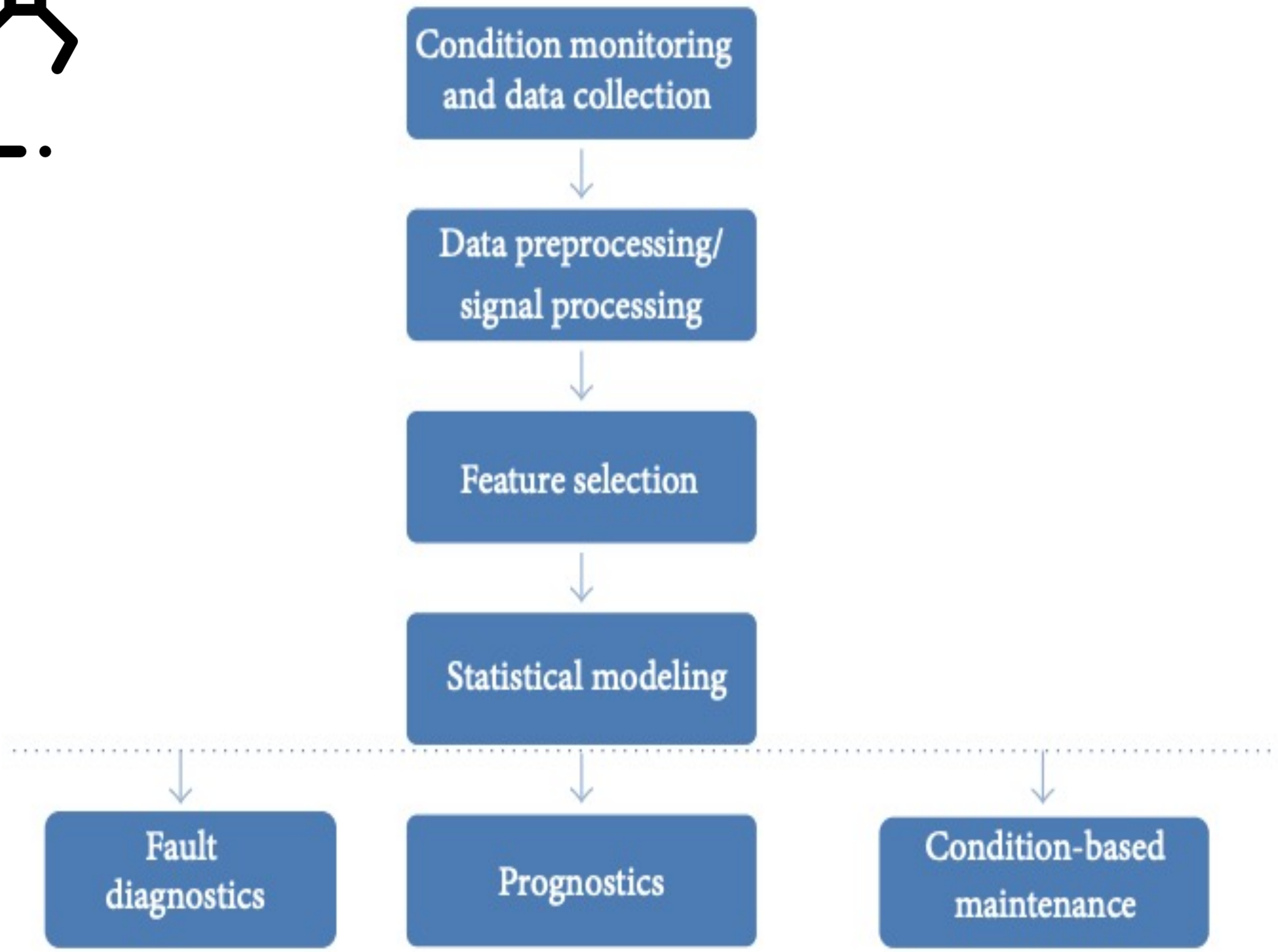
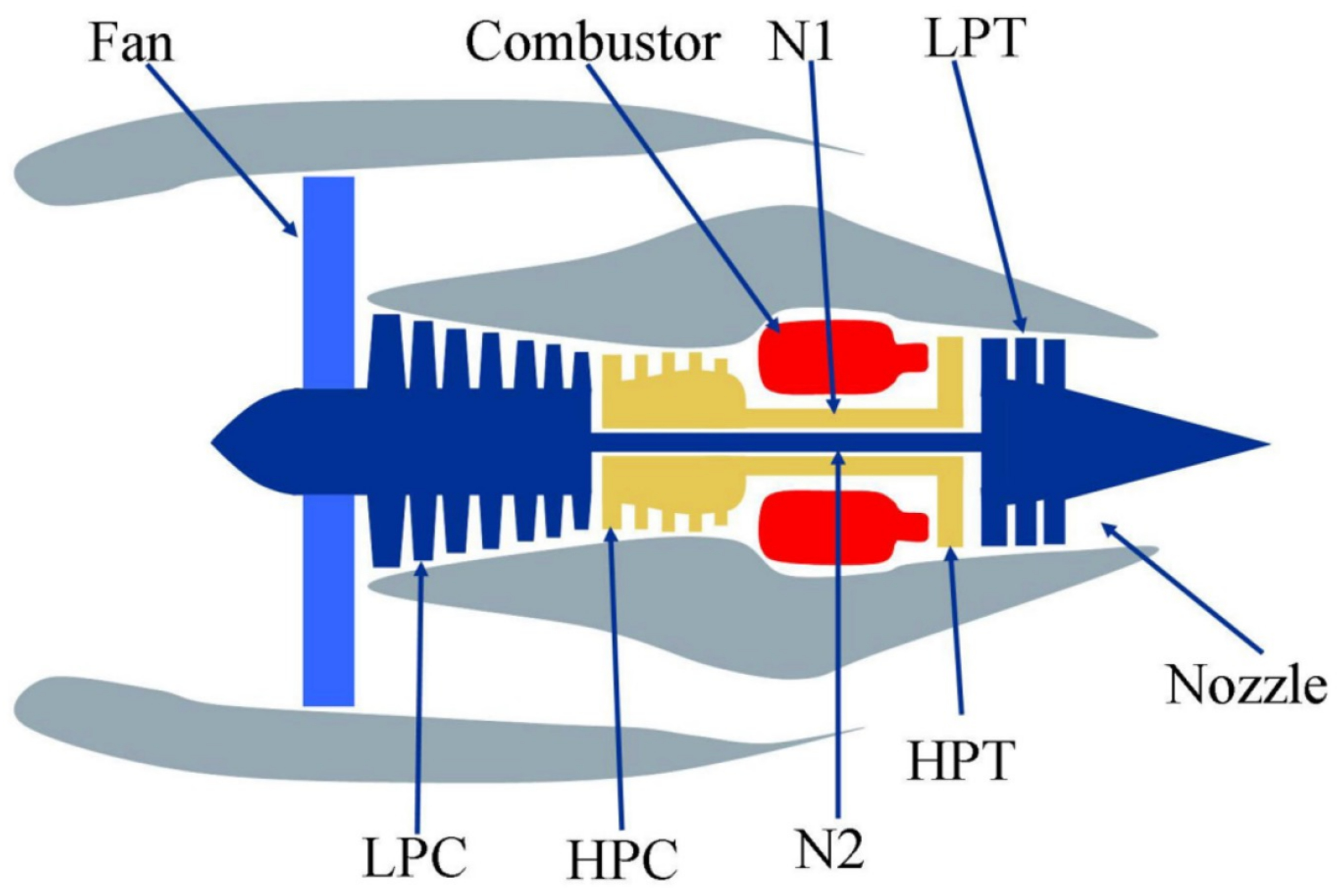


Figure from Tsui et al. (2015)

Introduction

- Existing RUL estimation approaches
  - Physical model-based approaches
    - Domain expertise, human error effects, complexity of parameter modeling
  - Data-driven approaches
    - Information from sensor measurements
    - Statistical model-based approaches → Machine learning (ML) & deep learning (DL) applications

  - Benchmark dataset: commercial modular aero-propulsion system simulation (C-MAPSS)
    - NASA's Ames Research Center



Existing Works

- Conventional RUL estimation approaches
  - Physical model-based approaches
  - Stochastic process-based approaches
    - Particle filter (PF), Kalman filter (KF)

- ML-based approaches
  - Feature extraction (e.g., Wavelet transform) → supervised prediction models (e.g., SVM)

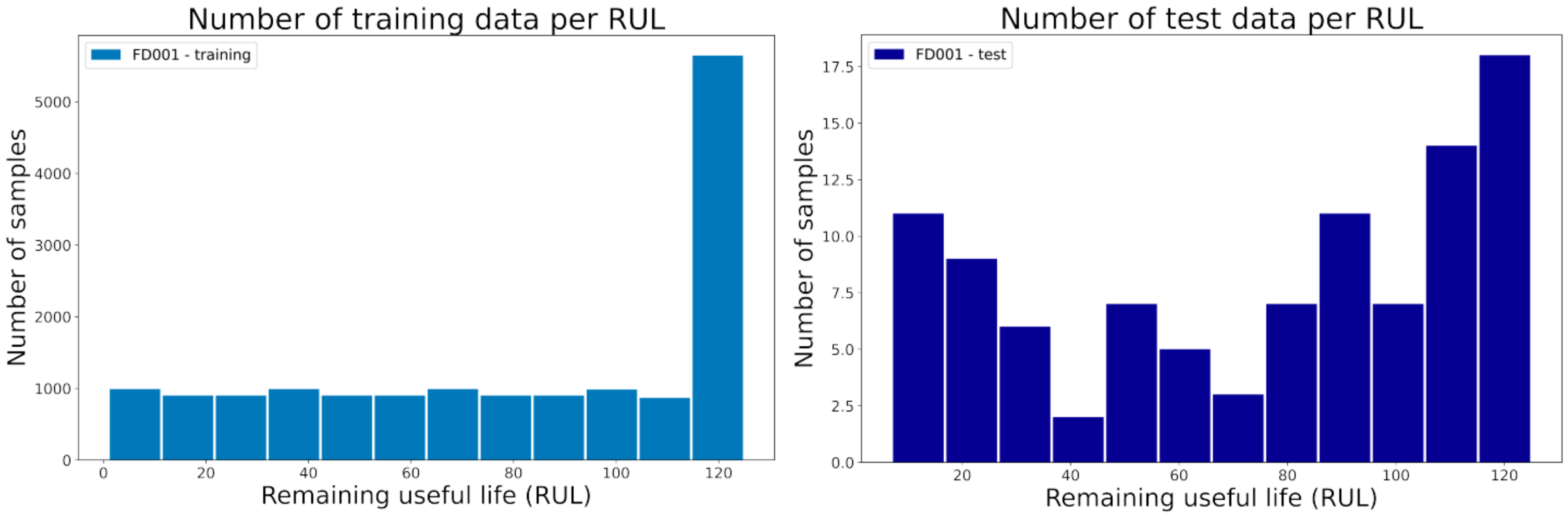
- DL-based approaches
  - Multivariate time series → supervised regression problems
  - High expressive power
    - RNN-based approaches
      - LSTM, GRU, Bidirectional, etc.
    - CNN-based approaches
      - 1D, 2D, multi-scale, residual, etc.
    - Transformer-based approaches
      - Not quite studied yet

- Hybrid DL-based approaches
  - Multiple models/techniques used in a hybrid manner

Proposed Method

- Existing problems of RUL estimation
  - Complex input data from sensor measurements
    - Multivariate time-series data

  - Unique data imbalance
    - Not quite studied yet
    - Most of data collected during *normal* operation time (i.e., before onset of degradation)
    - *#sample* of maximum RUL >> others (i.e., *#sample* of smaller RULs)



Proposed Method

- **Transformer-based estimation model**

- Transformer
  - Based on architecture proposed in Vaswani et al. (2017)
  - Short-, long-term dependencies & high expressive power
  - An encoder architecture is mainly employed

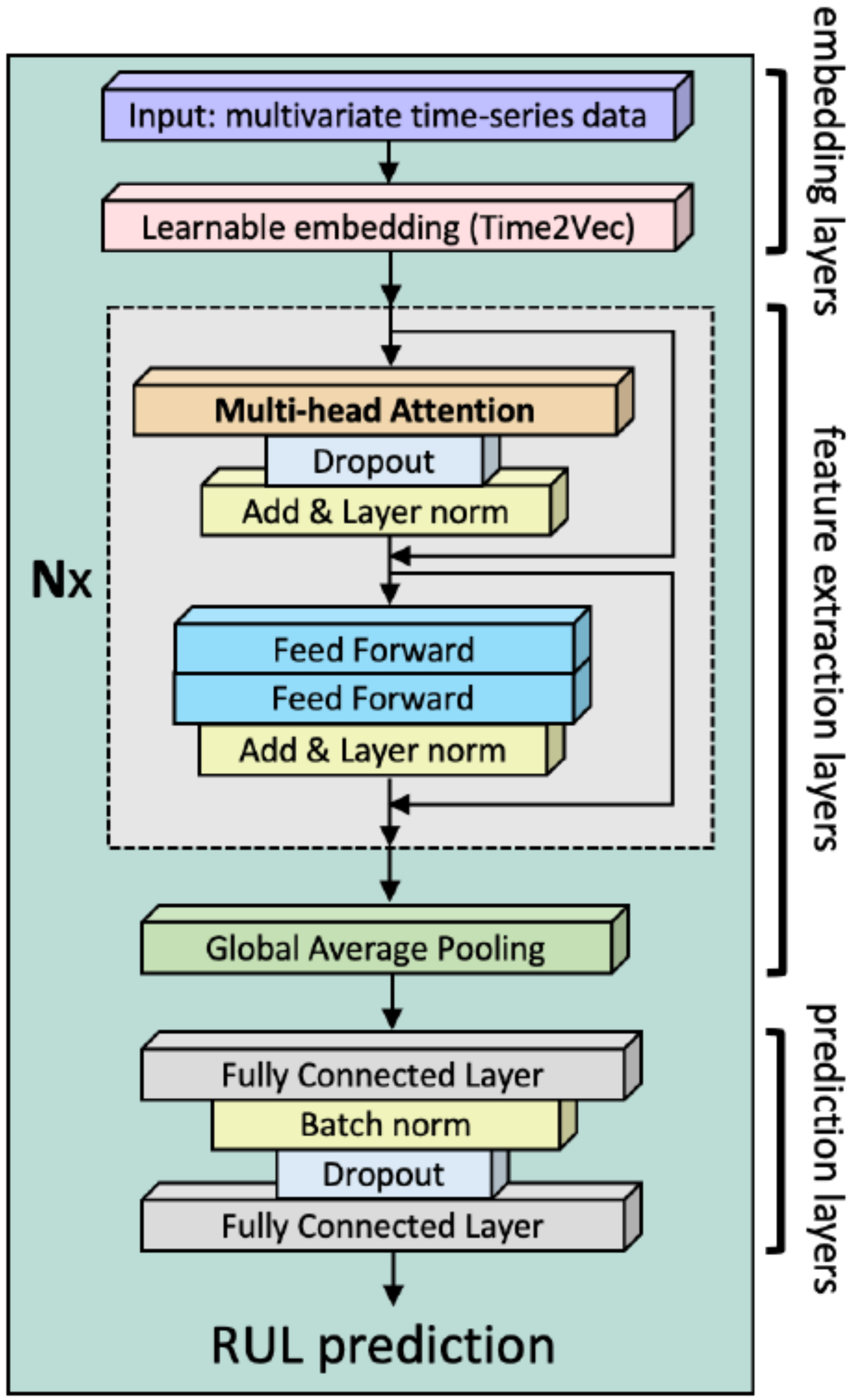
- Proposed model
  - 1) embedding layers
    - positional embedding → Time2Vec encoding

$$Time2Vec(t)[i] = \begin{cases} w_i + b_i, & for\ i = 0. \\ \mathcal{F}(w_i + b_i), & for\ i \geq 1. \end{cases}$$

where:

t: timestamp
i: variable index
$\mathcal{F}$: periodic function (e.g., the sine function)
w, b: learnable parameters

  - 2) feature extraction layers
  - 3) prediction layers



FIRST IN CHANGE

Proposed Method

- Proposed model
  - 1) embedding layers
  - 2) feature extraction layers
    - Multi-head attention blocks
    - Residual connection, layer normalization

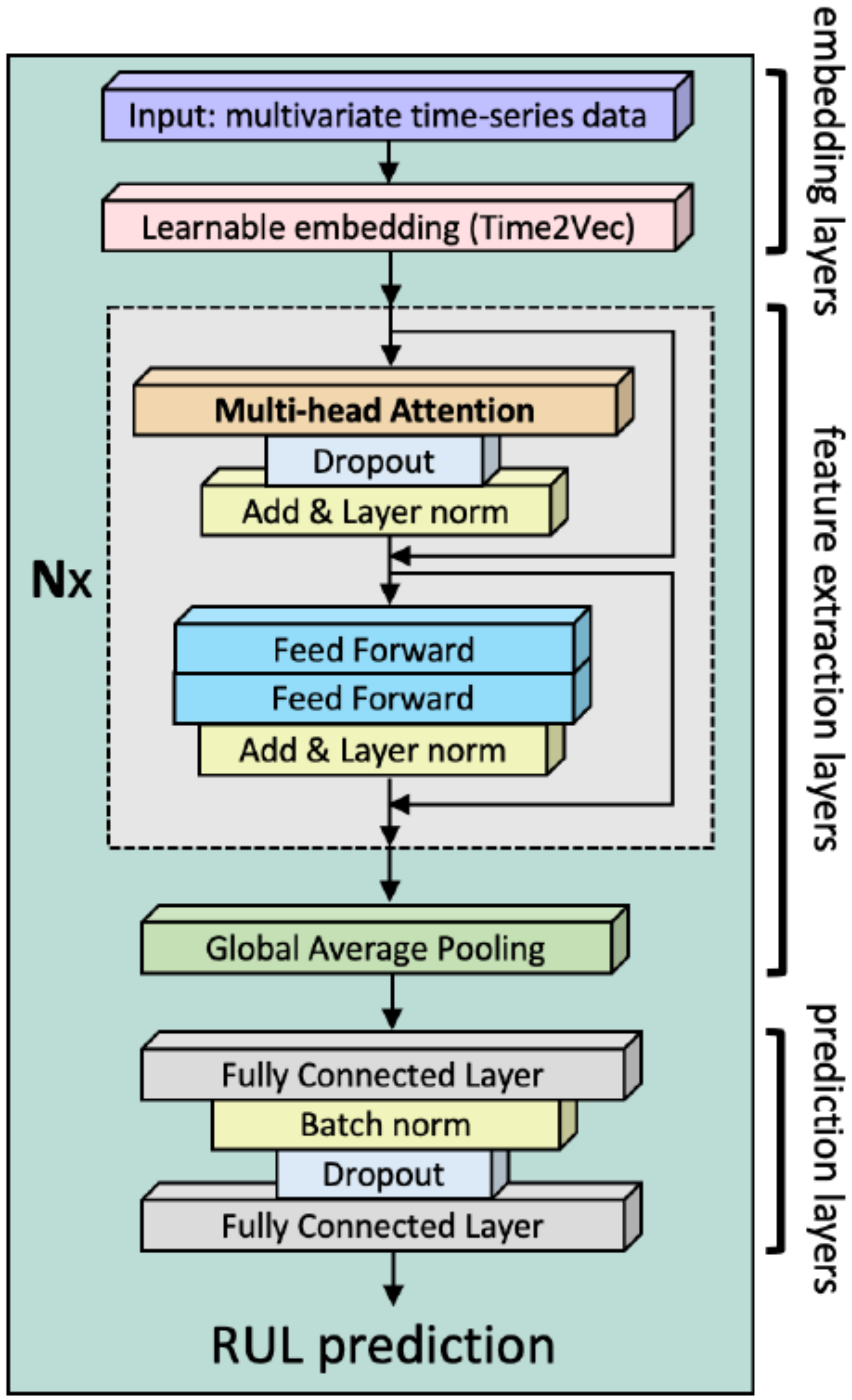
$$h_i = softmax(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}})VW_i^V$$

$$MultiHead(Q,K,V) = [h_1, ..., h_H]W^O$$

where:

Q: query
K: key
V: value
W^Q, W^K, W^V, W^O: learnable parameters

  - 3) prediction layers
    - Two FC layers
    - Feature representation → final RUL prediction



Proposed Method

- **Adaptive RUL-wise reweighting**

- To tackle unique data imbalance problems
  - Prone to overfitting to data w/ maximum & higher RUL
  - Related to: long-tailed recognition, imbalance
  - Possible strategies: under/over-sampling, augmentation, etc.

- Sample-wise reweighting + Rebalancing
  - Prevent from overfitting majority samples (w/ higher RULs)

- Adaptively arranging weights by RULs
  - During training, every step (e.g., epoch)

- RUL follows a continuous distribution
  - c.f., image classification (every independent class)
  - Kernel smoothing → maintain continuity of target (RUL)

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}}exp(-\frac{(x-x_i)^2}{2t^2})$$

where:

x: data point
t: kernel bandwidth

FIRST IN CHANGE

Proposed Method

- Adaptive RUL-wise reweighting

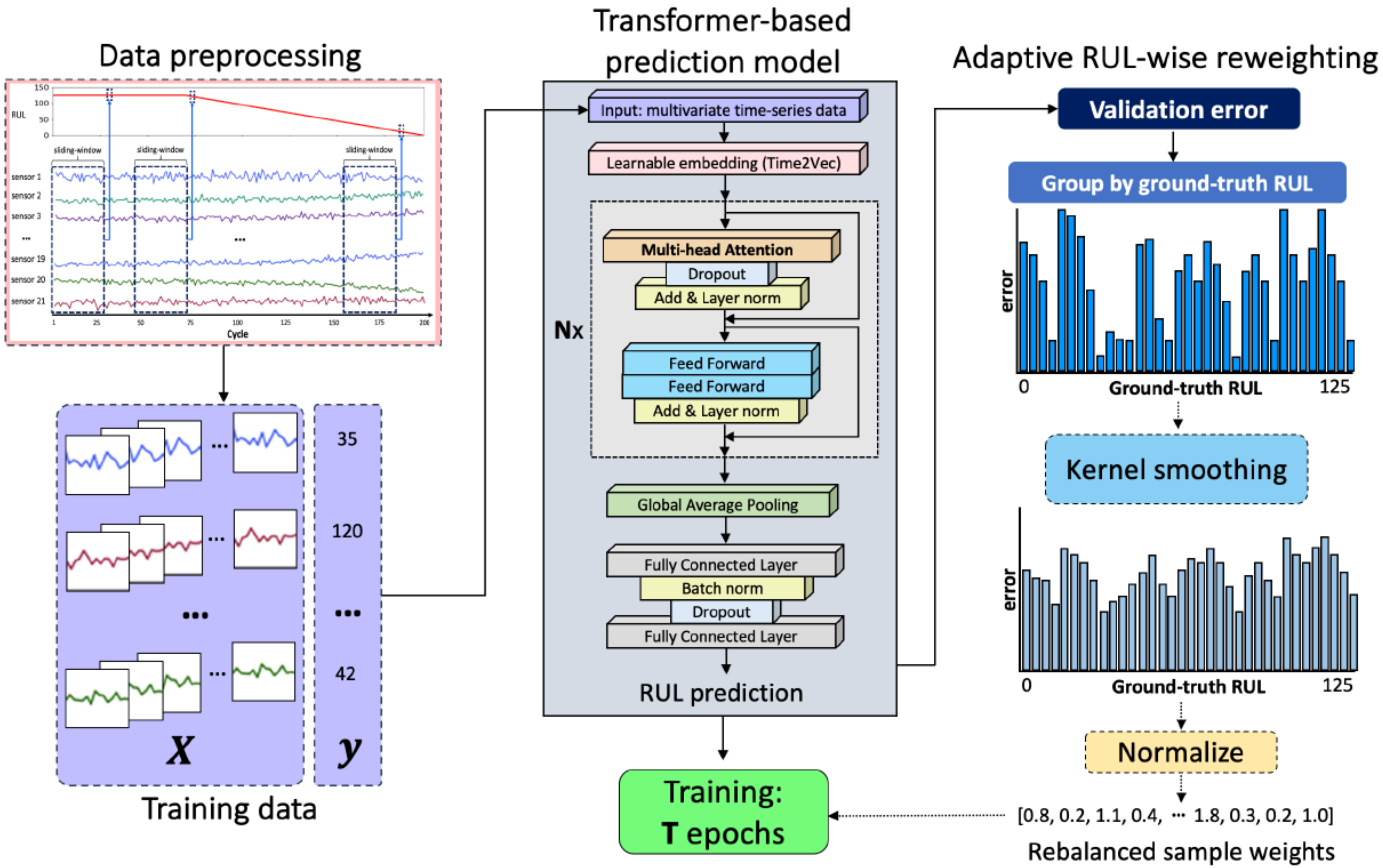
Algorithm 1 Adaptive RUL-wise Reweighting (ARR).

Input: training dataset $D_{train} = \{(x_i, y_i);\ i \in (1, ..., n)\}$, validation dataset $D_{valid} = \{(x_j, y_j);\ j \in (1, ..., m)\}$, total training epochs T, RUL weights C, Gaussian kernel smoother GKS, estimation model f_θ, loss function $\mathcal{L}$

Initialize f_θ ▷ Initialization of learnable parameters θ
Initialize $C : [c_1, c_2, ..., c_{125}] = [1, 1, ..., 1]$ ▷ RUL weights having 125 values initialized with 1
for $epoch = 1$ **to** $t - 1$ **do** ▷ Warm up training
  $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i)$
  $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}(\theta)$
end for
for $epoch = t$ **to** T **do** ▷ Apply reweighting after warm up
  Initialize $E : [e_1, e_2, ..., e_{125}]$
  for $j = 1$ **to** m **do** ▷ Calculate an RUL-wise error using a validation set
    e_{y_j}.append($\mathcal{L}(f_\theta(x_j), y_j)$)
  end for
  for $rul = 1$ **to** 125 **do** ▷ Average validation errors by RUL
    $c_{rul} \leftarrow sum(e_{rul})/|e_{rul}|$
  end for
  $C \leftarrow GKS(C)$ ▷ Apply Gaussian kernel smoothing on an error distribution
  $C \leftarrow Normalize(C)$ ▷ Normalize a smoothed distribution to use as weights
  $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i) \cdot C[y_i]$ ▷ Use adaptive weights during training
  $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}(\theta)$
end for

Proposed Method

- Overview of the proposed RUL estimation method



Experiments

- RUL estimation benchmark dataset: C-MAPSS
  - 21 sensor measurements

A DESCRIPTION OF THE VARIABLES IN THE C-MAPSS DATASET.

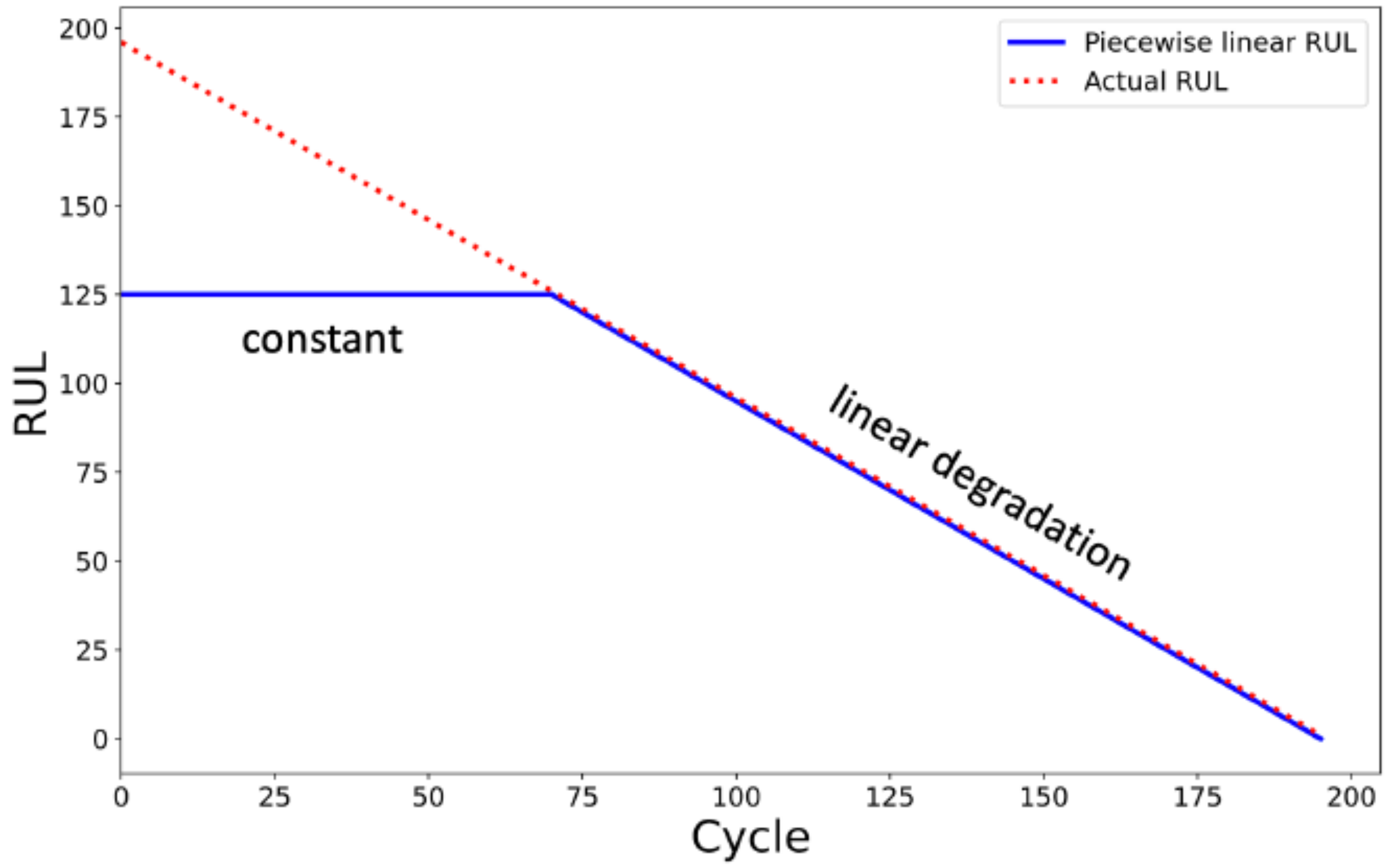
Variable	Description	Unit
Sensor 1	Total temperature at fan inlet	°R
Sensor 2	Total temperature at LPC outlet	°R
Sensor 3	Total temperature at HPC outlet	°R
Sensor 4	Total temperature at LPT outlet	°R
Sensor 5	Pressure at fan inlet	*psia*
Sensor 6	Total pressure in bypass-duct	*psia*
Sensor 7	Total pressure at HPC outlet	*psia*
Sensor 8	Speed of physical fan	*rpm*
Sensor 9	Speed of physical core	*rpm*
Sensor 10	Ratio of engine pressure (P50/P2)	-
Sensor 11	Static pressure at HPC outlet	*psia*
Sensor 12	Ratio of fuel flow to Ps30	*pps/psi*
Sensor 13	Corrected fan speed	*rpm*
Sensor 14	Corrected core speed	*rpm*
Sensor 15	Bypass ratio	-
Sensor 16	Ratio of fuel-air at burner	-
Sensor 17	Bleed enthalpy	-
Sensor 18	Demanded fan speed	*rpm*
Sensor 19	Demanded corrected fan speed	*rpm*
Sensor 20	HPT collant bleed	*lbm/s*
Sensor 21	LPT collant bleed	*lbm/s*

  - 4 subsets (FD001, FD002, FD003, FD004)

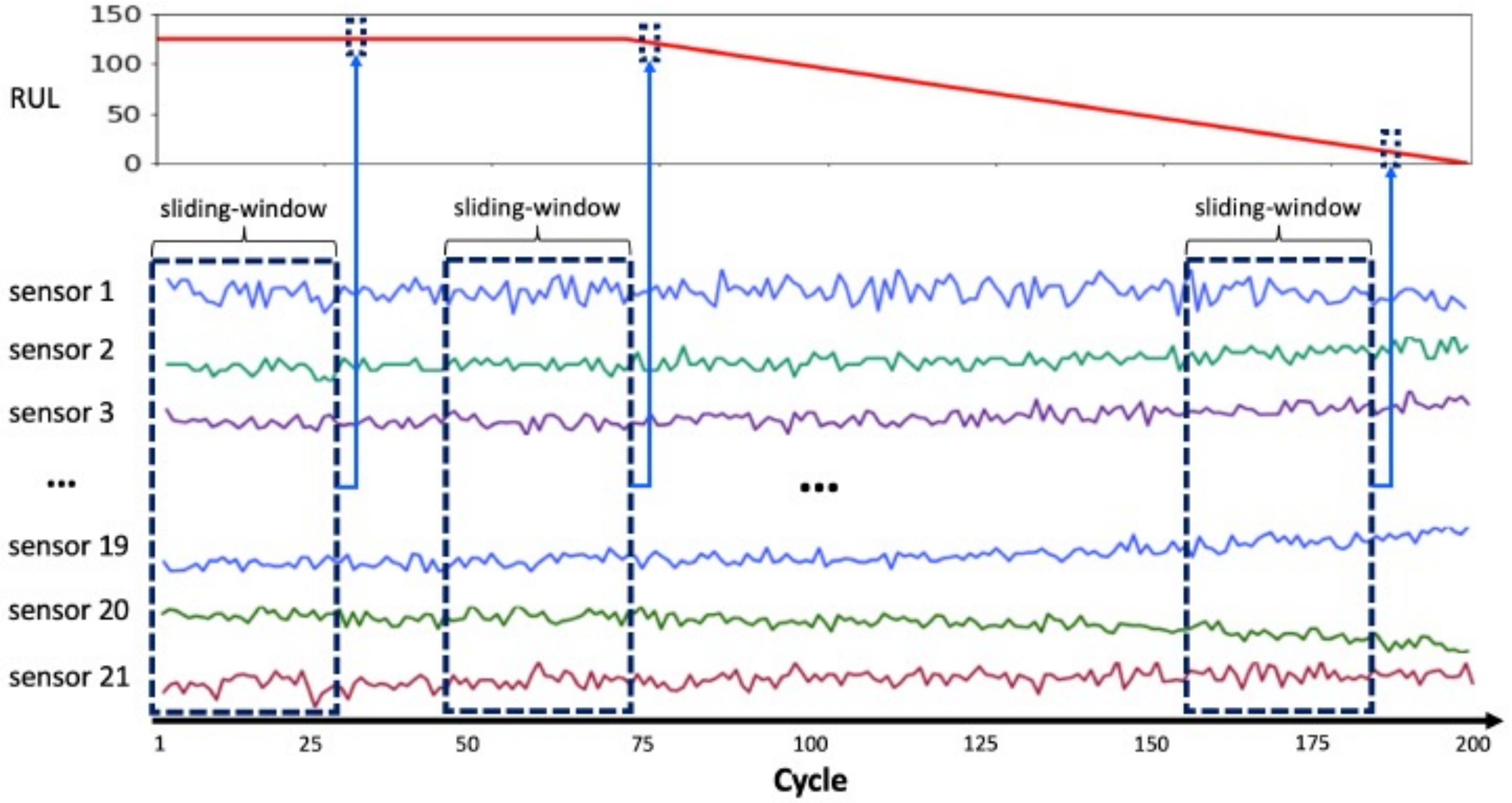
DETAILED INFORMATION ON THE C-MAPSS DATASET.

Variable	Subset			
	FD001	FD002	FD003	FD004
Number of engine units in the training set	100	260	100	248
Number of engine units in the test set	100	259	100	249
Operation conditions	1	6	1	6
Fault modes	1	1	2	2

- Data preprocessing
  - Piecewise linear RUL transformation (max=125)



  - Sliding-window preprocessing



FIRST IN CHANGE

Experiments

- Evaluation metrics
  - Root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}.$$

  - Score function (Saxena et al., 2008)

$$score = \sum_{i=1}^{N} s_i, \quad s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1, & for\ d_i < 0. \\ e^{\frac{d_i}{10}} - 1, & for\ d_i \geq 0. \end{cases}$$

where:

y_i = ground-truth RUL
$\hat{y}_i$ = predicted RUL
d_i = $\hat{y}_i$ - y_i: difference between predicted and true RULs

- Implementation
  - Loss function: mean squared error (MSE)
  - Optimizer: Adam
  - Regularization & more: early stopping, layer normalization, He initialization

Results and Discussion

- Transformer-based estimation model
  - Various configurations of hyperparameters
    - #attention block (N), #head (h), #dimensionality of feed-forward layers (d_{ff}), #dimensionality of key (d_k)

RESULTS OF THE ABLATION EXPERIMENT USING THE PROPOSED TRANSFORMER-BASED MODEL.

Type	N	h	d_{ff}	d_k	FD001		FD002		FD003		FD004	
					RMSE	*Score*	RMSE	*Score*	RMSE	*Score*	RMSE	*Score*
base	6	6	6	64	**12.37**±0.29	**238.54**±3.38	19.57±0.26	4792.06±125.55	12.13±0.27	245.80±51.77	21.74±0.08	4936.79±292.95
(A)	2	-	-	-	12.59±0.05	255.68±23.59	19.34±0.44	4202.85±101.59	12.31±0.18	262.80±54.78	21.91±0.35	5547.87±118.56
	4	-	-	-	13.25±0.46	325.96±47.26	19.21±0.52	4328.87±76.23	13.13±0.81	319.59±49.24	21.75±0.30	6453.47±239.13
	8	-	-	-	12.79±0.42	295.16±48.26	19.50±0.20	5271.34±118.27	13.73±0.88	640.28±37.02	22.59±0.41	6087.83±72.23
(B)	-	2	-	-	12.55±0.25	254.86±27.35	20.11±0.14	5772.45±61.56	13.10±0.87	423.23±272.85	22.21±0.70	6200.26±390.43
	-	4	-	-	13.78±1.53	366.11±31.38	19.30±0.51	4676.25±139.80	13.54±2.62	650.71±54.96	21.96±0.26	6571.54±126.08
	-	8	-	-	12.96±0.20	290.36±34.59	19.34±0.34	5319.92±99.68	12.66±0.31	342.96±81.52	22.40±0.30	7264.96±88.69
(C)	-	-	2	-	12.85±0.28	299.942±35.19	**19.09**±0.43	**4056.67**±138.01	12.51±0.36	267.66±33.87	22.09±0.26	5992.65±136.64
	-	-	4	-	12.74±0.58	281.15±44.20	19.36±0.40	4458.63±115.30	13.17±1.02	506.47±135.03	21.84±0.08	4749.66±65.76
	-	-	8	-	12.46±0.36	261.692±42.62	19.54±0.46	5523.61±106.88	13.19±0.68	369.74±106.11	**21.50**±0.21	**4306.08**±64.59
	-	-	10	-	12.92±0.23	304.21±32.17	19.28±0.30	5171.73±127.75	13.27±0.65	502.11±216.34	21.90±0.11	5464.94±169.23
(D)	-	-	-	8	12.90±0.48	290.154±30.19	19.87±0.08	6042.43±87.37	12.16±0.66	269.61±66.53	22.33±0.24	7044.58±113.29
	-	-	-	16	12.96±0.44	284.13±35.37	19.58±0.30	6291.63±169.84	**11.94**±0.71	**238.64**±44.12	22.27±0.42	6360.03±148.10
	-	-	-	32	13.18±0.34	313.54±32.25	19.10±0.20	5210.58±64.58	13.03±0.45	328.95±38.60	21.90±0.42	5205.14±560.33

Results and Discussion

- Adaptive RUL-wise reweighting (ARR)
  - Using the best configurations, warm-up epochs=50

THE EFFECTS OF ARR. WHERE WARM-UP EPOCHS = 50.

Subset	N	h	d_{ff}	d_k	ARR usage	Metric	
						RMSE	*Score*
FD001	6	6	6	64	✗	12.37±0.29	238.54±3.38
					✓	**11.39**±0.40	**194.08**±12.96
FD002	6	6	2	64	✗	19.09±0.43	4056.67±138.01
					✓	**19.01**±0.09	**3234.99**±171.27
FD003	6	6	6	16	✗	11.94±0.71	238.64±44.12
					✓	**11.29**±0.19	**144.02**±18.18
FD004	6	6	8	64	✗	21.50±0.21	4306.08±64.59
					✓	**21.41**±0.15	**4048.13**±295.17

- Various warm-up epochs (10, 50, 100)

EFFECTS OF ARR WITH VARIOUS WARM-UP EPOCHS DURING TRAINING.

Subset	ARR usage	Warm-up epoch	Metric	
			RMSE	*Score*
FD001	✓	10	11.58±0.21	196.99±7.21
		50	**11.39**±0.40	**194.08**±12.96
		100	12.02±0.33	235.10±35.16
FD002	✓	10	19.04±0.36	3752.91±73.01
		50	**19.01**±0.09	**3234.99**±171.27
		100	19.10±0.09	3591.41±46.63
FD003	✓	10	11.69±0.20	207.43±43.88
		50	**11.29**±0.19	**144.02**±18.18
		100	11.85±0.45	230.09±10.20
FD004	✓	10	21.46±0.30	3628.19±45.88
		50	21.41±0.15	4048.13±295.17
		100	**21.39**±0.13	**3560.02**±42.54

Results and Discussion

- Comparison with existing methods in the literature

COMPARISON OF ESTIMATION PERFORMANCE WITH EXISTING METHODS.

Method	Number of sensors	Maximum RUL	Size of sliding-window	FD001		FD002		FD003		FD004	
				RMSE	*Score*	RMSE	*Score*	RMSE	*Score*	RMSE	*Score*
DLSTM [49]	13	130	N/A	14.72	262.00	29.00	6953.00	17.72	452.00	33.43	15069.00
RBM-LSTM [36]	14	130	N/A	12.56	231.00	22.73	3366.00	12.10	251.00	22.66	2840.00
MS-DCNN [38]	14	125	30/20/30/15	11.44	196.22	19.35	3747.00	11.67	241.89	22.22	4844.00
CNN [37]	21	130	15	18.45	1286.70	30.29	13570.00	19.82	1596.20	29.16	7886.40
DCNN [40]	14	125	30	12.61	272.70	22.36	10412.00	12.64	284.10	23.31	12466.00
RNN-AE [33]	14	115-190	15	13.58	228.00	19.59	2650.00	19.16	1727.00	22.15	2901.00
BiLSTM [32]	14	125	N/A	13.65	295.00	23.18	4130.00	13.74	317.00	24.86	5430.00
HDNN [51], [52]	14	125	30	13.02	245.00	**15.24**	1282.42	12.22	287.72	**18.16**	1527.42
Deep LSTM [30]	21	130	10	16.14	338.00	24.49	4450.00	16.18	852.00	28.17	5550.00
CNN-RNN [50]	21	N/A	31	16.89	820.67	30.97	15917.00	17.82	950.94	29.73	7212.20
RecCNN [39]	14	125	30	12.16	212.48	20.85	2087.77	12.01	180.76	24.97	3400.44
GP [22]	21	125	N/A	32.61	1250.00	68.54	70100.00	49.89	3510.00	63.93	10900.00
LSTMBS [74]	21	130	31/21/38/19	14.89	481.10	26.86	7982.00	15.11	493.40	27.11	5200.00
LSTM-MSCNN [53]	14	125	30	12.75	281.00	22.46	5170.00	11.35	278.00	24.10	4790.00
MT-CNN [48]	21	125	30	12.48	224.00	19.77	2023.00	12.11	334.00	19.98	2097.00
DACL [54]	21	130	32/22/39/19	14.41	288.00	28.52	12004.00	14.36	321.00	26.81	6594.00
DAST [45]	14	125	40/60/40/60	11.43	203.15	15.25	**924.96**	11.32	154.92	18.36	**1490.72**
Ours (proposed)	14	125	30/20/30/15	**11.39**	**194.08**	19.01	3234.99	**11.29**	**144.02**	21.39	3560.02

Conclusion and Future Works

- Importance of RUL estimation in the PHM domain
    - Increasing complexity of the problem
- Proposed method
    - 1) Transformer-based estimation model
    - 2) Adaptive RUL-wise reweighting (ARR) technique
- Effectiveness of the proposed method
    - Comparable estimation performance of the transformer-based model
    - Improved performance when using ARR
- Future works
    - Uncertainty quantification
    - Active learning + manufacturing/industrial machinery
    - DL-based anomaly detection

References

- Tsui, Kwok L., Nan Chen, Qiang Zhou, Yizhen Hai, and Wenbin Wang. "Prognostics and health management: A review on data driven approaches." Mathematical Problems in Engineering 2015 (2015).
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- Saxena, Abhinav, Kai Goebel, Don Simon, and Neil Eklund. "Damage propagation modeling for aircraft engine run-to-failure simulation." In 2008 international conference on prognostics and health management, pp. 1-9. IEEE, 2008.

- **This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korea Government (MSIT) under Grant 2021R1F1A1046416.**

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY 2007

Industrial Intelligence Laboratory (IIL)

Gyeongho Kim

Contact: kkh0608@unist.ac.kr

FIRST IN CHANGE