Deep Learning Models for Multivariate Time-Series Forecasting

Submitted in partial fulfillment of the

requirements of CS F266

By

Harshit Agrawal

ID No. 2018A8PS0484P

Under the supervision of:

Dr. Poonam Goyal.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

May 2021

Declaration of Authorship

I, Harshit Agrawal, declare that this Undergraduate project titled, ‘Deep Learning Models for Multivariate Time-Series Forecasting’ and the work presented in it are my own. I confirm that:

- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given.
- I have acknowledged all main sources of help.

Signature – Harshit Agrawal

Date – 13-05-2021

Certificate

This is to certify that the project titled, "Deep Learning Models for Multivariate Time-Series Forecasting" and submitted by Harshit Agrawal ID No. 2018A8PS0484P in partial fulfillment of the requirements for CS F266 embodies the work done by him under my supervision.

Supervisor
Dr. Poonam Goyal
Associate Professor,
BITS Pilani, Pilani Campus,

Date:

<u>Acknowledgement</u>

The successful completion of this project required a lot of support and guidance, and I am extremely privileged to have obtained this all through the process.

I would like to extend my deepest gratitude to my project mentor, **Dr. Poonam Goyal**, for providing me with the opportunity to work with her on this project, and her guidance and mentorship during the same.

I would like to extend my gratitude to all other members of the organization, who have been highly cooperative during the program, and have worked very hard for making this program a very fruitful one in terms of the experience gained by the students.

At last, special thanks to my family and friends, whose timely support was always a constant source of motivation for me, and to whom I will always remain indebted for their unspoken yet valuable encouragement.

Contents

1. ABSTRACT

Time-Series Forecasting problems contain a complex mixture of inputs including static covariates, observed inputs only in the past, and known inputs into future. Given the diversity of time-series problems across various domains, numerous neural network design choices have emerged. Over the years, literature has been enriched with numerous models for Time-Series Forecasting such as 1D Convolutional Neural Networks to Long-Short-Term Memory Networks and sequence-to-sequence architectures. In recent years, research has proposed several deep learning (DL) approaches to provide reliable remaining useful life (RUL) predictions in Prognostics and Health Management (PHM) applications. However, the available approaches do not consider the heterogeneity of the prognostics data and are often not very efficient in utilizing the temporal correlation in sensor data. In this work, we propose an Encoder-Decoder architecture followed by the use of Multi-Head Self Attention mechanism, while taking into consideration the variety of inputs, to learn both the short-term temporal features and long-term dependencies respectively in sensor data to accurately predict the RUL of aircraft engines. Since the sensor data can be complex and relatively simple in different datasets, we also employ Gated Linear Units (GLUs) to make our architecture self-adjusting to variety of complexities in datasets. Our work shows significant improvements in RUL estimation over the already available approaches for Commercial Modular Aero-Propulsion System Simulation (CMAPPS) and New Commercial Modular Aero-Propulsion System Simulation (NCMAPPS) datasets.

2. INTRODUCTION

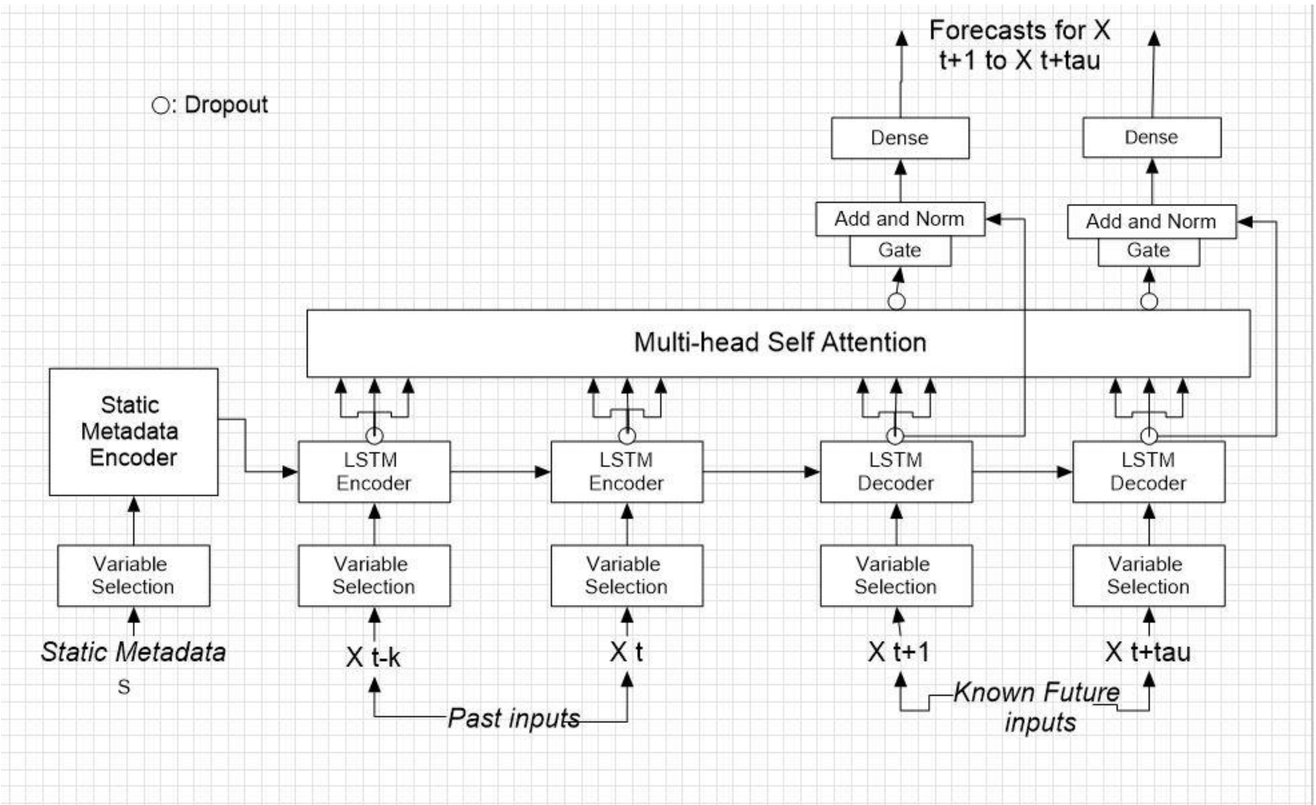
The remaining useful life (RUL) is a technical term used to describe the progression of faults in Prognostics and Health Management (PHM) applications. RUL predictions allow us to prevent critical failures. Various techniques for Multi-Horizon Multivariate Time-Series Forecasting have been summarised in [1]. Deep Learning techniques, such as Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM), have shown rapid developments and outperformed traditional prognosis algorithms in RUL predictions for turbofan engine degradation [2] [3]. However, prognostics data applications commonly have access to a variety of data sources, including known information about the future, other exogenous time series and static metadata, without any prior knowledge on how they interact. This heterogeneity of data sources together with little information about their interactions makes

RUL prediction particularly challenging. The current architectures fail to take into account the above heterogeneity and often not very efficient in utilizing the temporal correlation in sensor data. In this work, we distinguish our sensor data into numerous categories viz. Past Observed Inputs, Known Future Inputs and Static Metadata [4] . We then employ a sequence-to-sequence network to capture the short-term temporal correlations and a Multi-head Self-Attention layer [5] to learn long term dependencies in our data. We also make use of Gated Linear Units (GLUs) [6] to skip over the unused part of architectures when dealing with simpler data. Variable selection network are also utilized from [4] to minimize the contribution of irrelevant features in our timestamps.

3. Related Work

The C-MAPSS dataset has been extensively used to evaluate several DL approaches to RUL predictions. This section reviews the most recent studies applied on the C-MAPSS dataset. The selected studies either utilize a Convolutional Neural Network (CNN), a Deep Belief Network (DBN) or Long-Short Term Memory (LSTM) in the proposed deep architecture. Yuan et al. proposed an LSTM approach for several different faults [7] . The proposed approach was compared with traditional RNN, Gated Recurrent Unit LSTM (GRU-LSTM) and AdaBoost-LSTM. It showed improved performance in all cases. Another LSTM approach was provided by Zheng et al. [8] . The proposed approach provides RUL predictions using two LSTM layers, two Feed-forward Neural Network (FNN) layers, and an output layer. The LSTM layers were able to reveal hidden patterns in the C-MAPSS dataset and achieved higher accuracy compared to the Hidden Markov Model or traditional RNN. A similar study was provided by Wu et al. In this study, an LSTM was combined with a dynamic difference method in order to extract new features from several operating conditions before the training procedure. These features contain important degradation information, which improves the LSTM to better control the underlying physical process. The proposed approach showed enhanced performance compared to traditional RNN and GRU-LSTM.

4. <u>Proposed Approach</u>



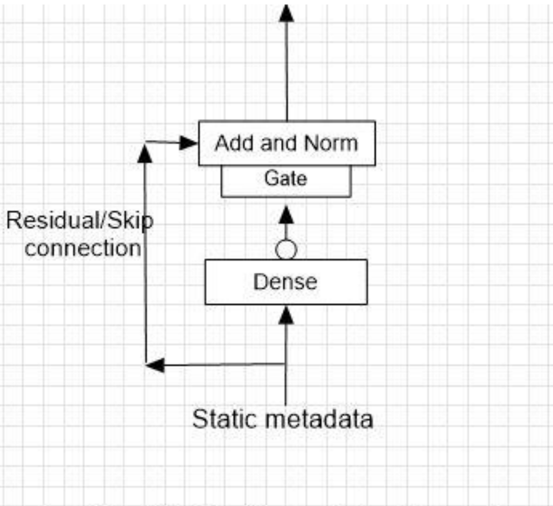
Our Transformer Architecture for RUL prediction

The major components of our architecture are as follows:

4.1. <u>Variable Selection Networks:</u> When multiple variables may be available, their relevance and specific contribution to the output are unknown. We use the variable selection network from [4] to achieve instance-wise variable selection applied to both static covariates and time-dependent covariates. Beyond providing insights into which variables are most significant for the prediction problem, variable selection also allows to remove any unnecessary noisy inputs which could negatively impact performance.

4.2. <u>Gated Linear Units:</u> We have used Gated Linear Units (GLU) from [6] to suppress any parts of architecture that are not required for a given dataset.

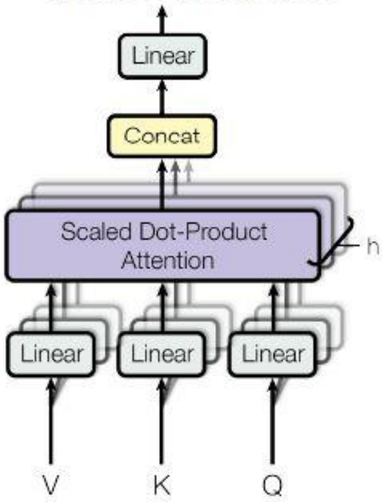
4.3. Static Covariate Encoders: We have designed our model to incorporate static features into our network. The static covariate encoder process the static features in our data and feed it to sequence-to-sequence layer.



Inside Static Covariate Encoder

4.4. A ***Sequence-to-Sequence layer (Encoder-Decoder)*** to learn the local temporal relations and features.

4.5. Multi-Head Self Attention : This work also employs MultiHead Self Attention [5] to learn long term dependencies across various time steps.



5. Datasets used

We evaluate our proposed approach on **C-MAPPS and NCMAPPS datasets**.

The aircraft engine dataset is generated by NASA using the commercial modular aero-propulsion system simulation tool. A total of 26 columns of data are provided, including the unit number, operating cycle number, three kinds of operational settings, and 21 sensor signals. We have used the subset FD002 of the C-MAPPS to evaluate our model. The engines in subset FD002 have six operational modes that switch over time. These engines degenerate from random initial states and have different lifespans. In the training set, the 260 engines contains sensor observations of the entire lifetime, and all data can be used as training samples. In the testing set, the sensor observations of the 259 engine units end at an unknown moment before failure occurs. The prediction task is to estimate the remaining operating cycle number of each aircraft engine in the testing set.

The proposed methodology is also demonstrated and evaluated on a synthetic dataset with run-to-failure degradation trajectories of a small fleet comprising nine turbofan engines with unknown and different initial health conditions. The dataset was generated with the Commercial Modular Aero-Propulsion System Simulation (NC-MAPSS) dynamical model. Real flight conditions as recorded onboard of commercial jets were taken as input to the NC-MAPSS model. We use DS02 subset of the NCMAPPS dataset. It contains N=6 training units and M=3 test units. We have reduced the training units owing to computational limitations and used training units 16,18,20 and test units 11,14,15.

16	0.77M	16	63	HPT+LPT
18	0.89M	17	71	HPT+LPT
20	0.77M	17	66	HPT+LPT
Test Dataset - $\mathcal{D}_{T*}$				
Unit (u)	m_j	t_s	t_{EOL}	Failure Mode
11	0.66M	19	59	HPT+LPT
14	0.16M	36	76	HPT+LPT
15	0.43M	24	67	HPT+LPT

6. Results and Discussion

We used quantile loss functions to train our model. We used q=0.1 , 0.5 to formulate our quantile loss function. As a result, we obtained 2 types of outputs: P90 output and p50 output.

Further to evaluate our results, we used the s-score function and Root Mean Square Function (RMSE).

$$S = \begin{cases} \sum_{i=1}^{n} e^{(-\frac{d_i}{13})} - 1, & \text{for} \quad d_i < 0 \\ \sum_{i=1}^{n} e^{(-\frac{d_i}{10})} - 1, & \text{for} \quad d_i \geq 0 \end{cases}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}$$

6.1. CMAPPS (FD002 dataset):

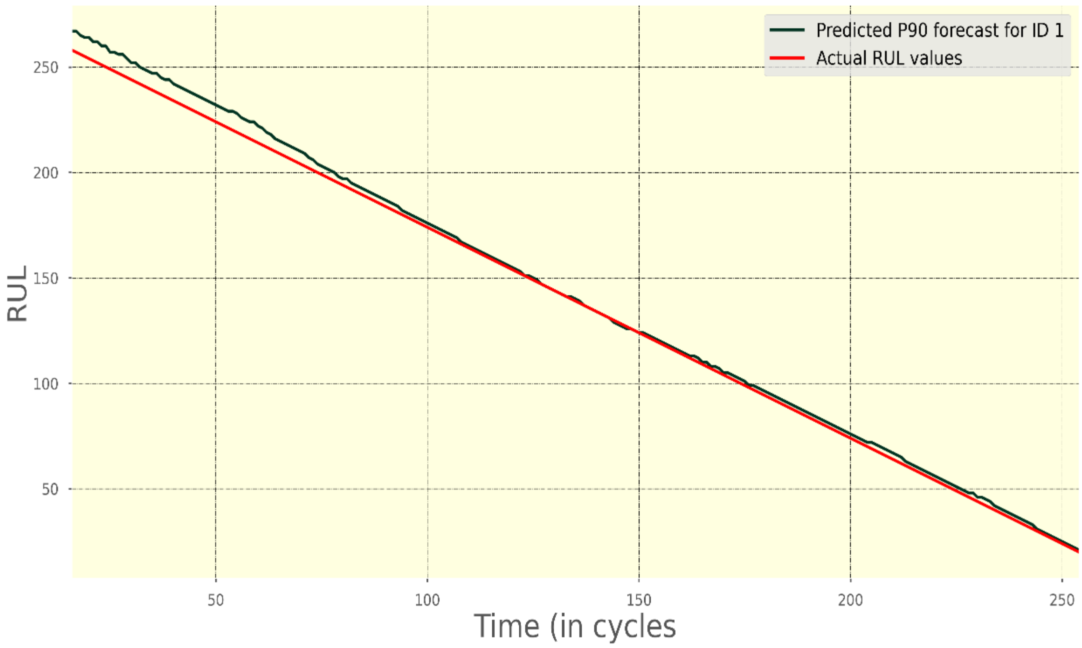
We evaluated the proposed approach on FD002 subset of CMAPPD dataset. We employed hyperparameter tuning via random search to find the best parameters. The hyperparameter tuning took 9 hours to complete on the paid version of Google Colab. Following were the hyperparameters obtained :

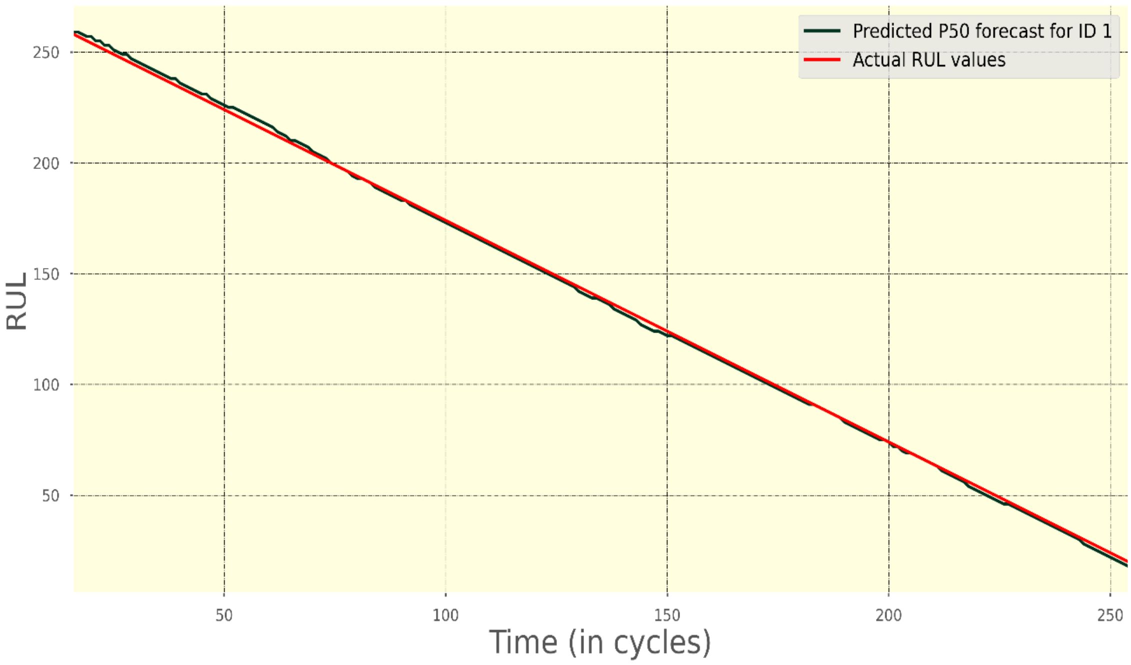
Dropout rate	0.2
Minibatch Size	64
No.of heads	1
Hidden Layer Size	320
Stack Size	1
Learning rate	0.001

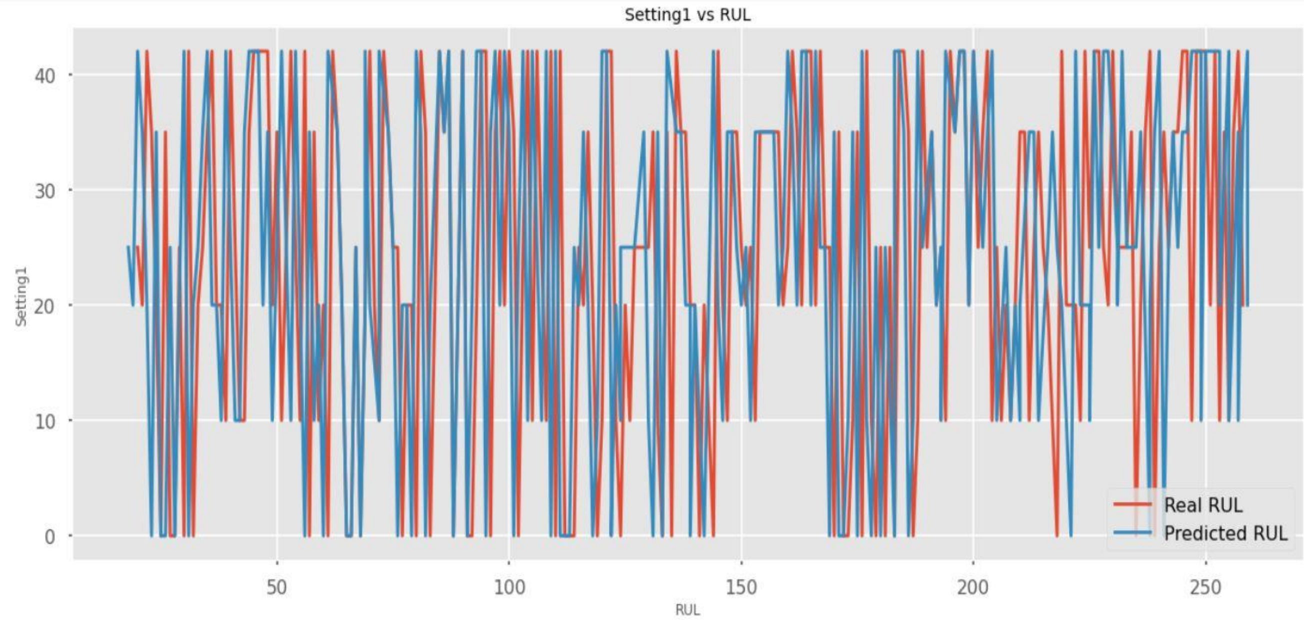
Apart from these hyperparameters, following fixed parameters were used:

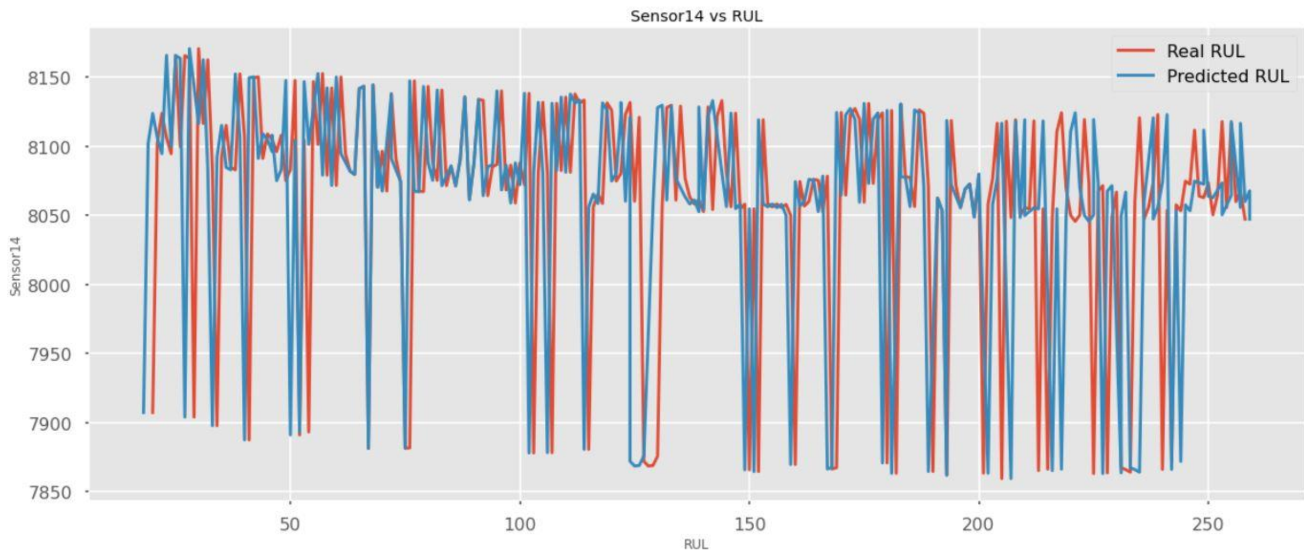
Total Time Steps : 20, Encoder size : 17, Early Stopping Patience : 3, epochs: 20.

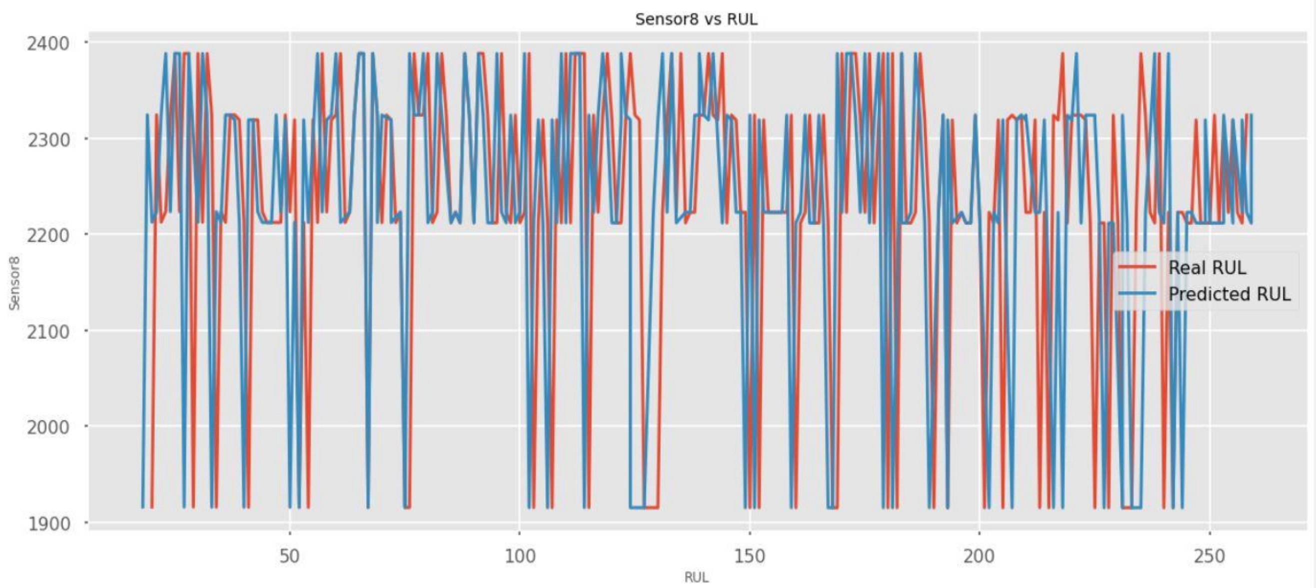
The resultant RUL VS cycle curve, RUL vs Setting 1 , RUL vs Sensor 14 , RUL vs Sensor 8 for engine ID 1 :






Setting1 vs RUL
Setting1
RUL
Real RUL
Predicted RUL


Sensor14 vs RUL
Sensor14
RUL
Real RUL
Predicted RUL


Sensor8 vs RUL
Sensor8
RUL
Real RUL
Predicted RUL

The evaluation metrics for CMAPPS using our proposed approach and comparing with other State-Of-The-Art approaches:

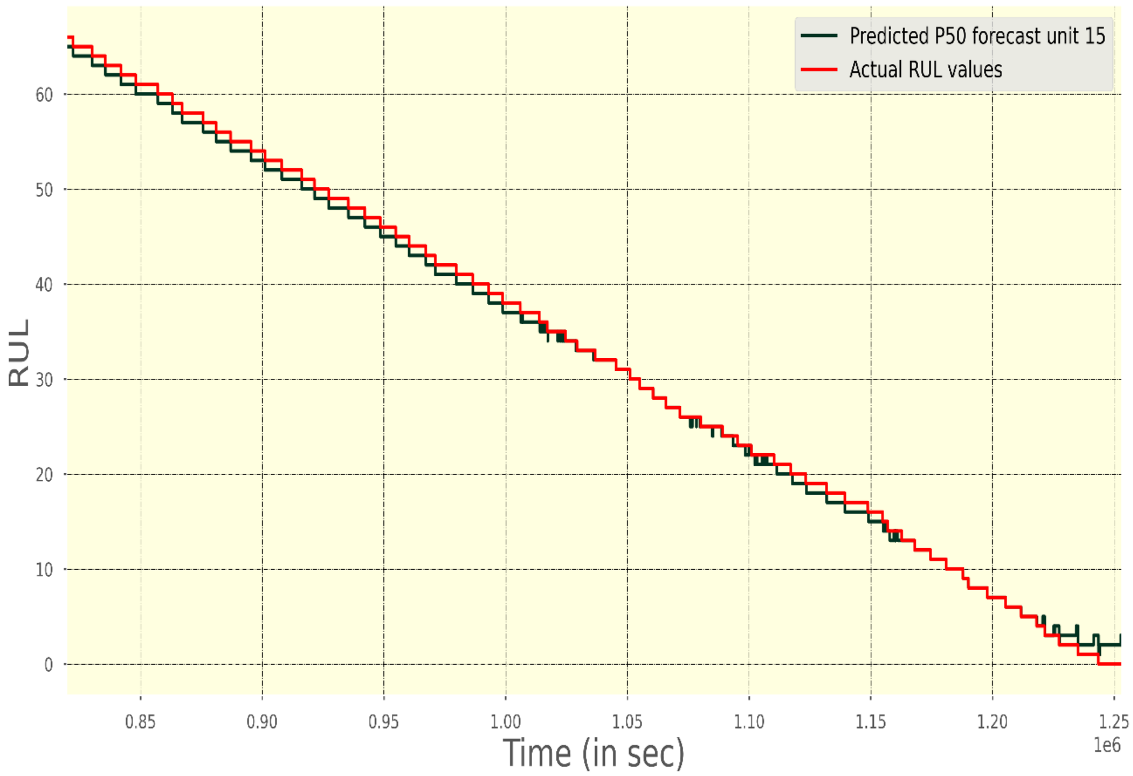
	RMSE loss	**S-score**
TFT	On P90 forecast: **3.761801748** On P50 forecast: **2.957496175**	On P90 forecast : **11022.8158** On P50 forecast: **5599.595934**
LSTM+FNN	**22.49**	**4450**
Semi Supervised setup	**22.73**	**3336**
Proposed Approach	On P90 forecast: **3.56417406** On P50 forecast: **1.573298**	On P90 forecast : **9713.307115** On P50 forecast: **3223.651315**

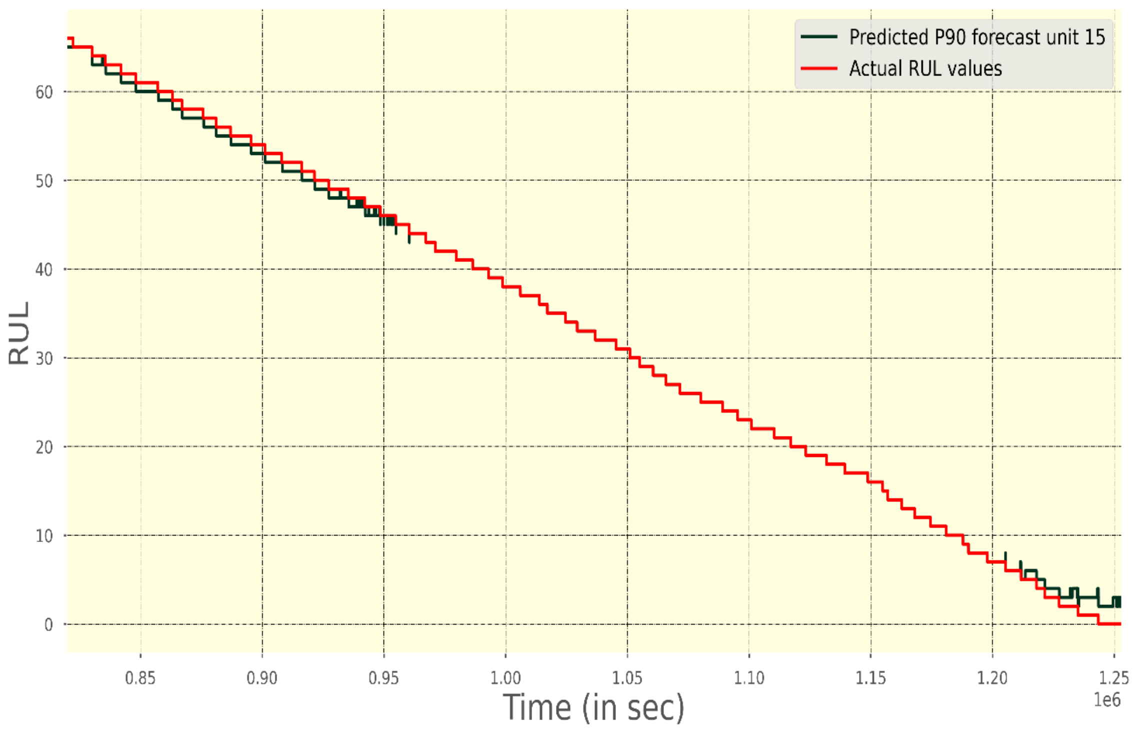
6.2 NCMAPPS dataset:

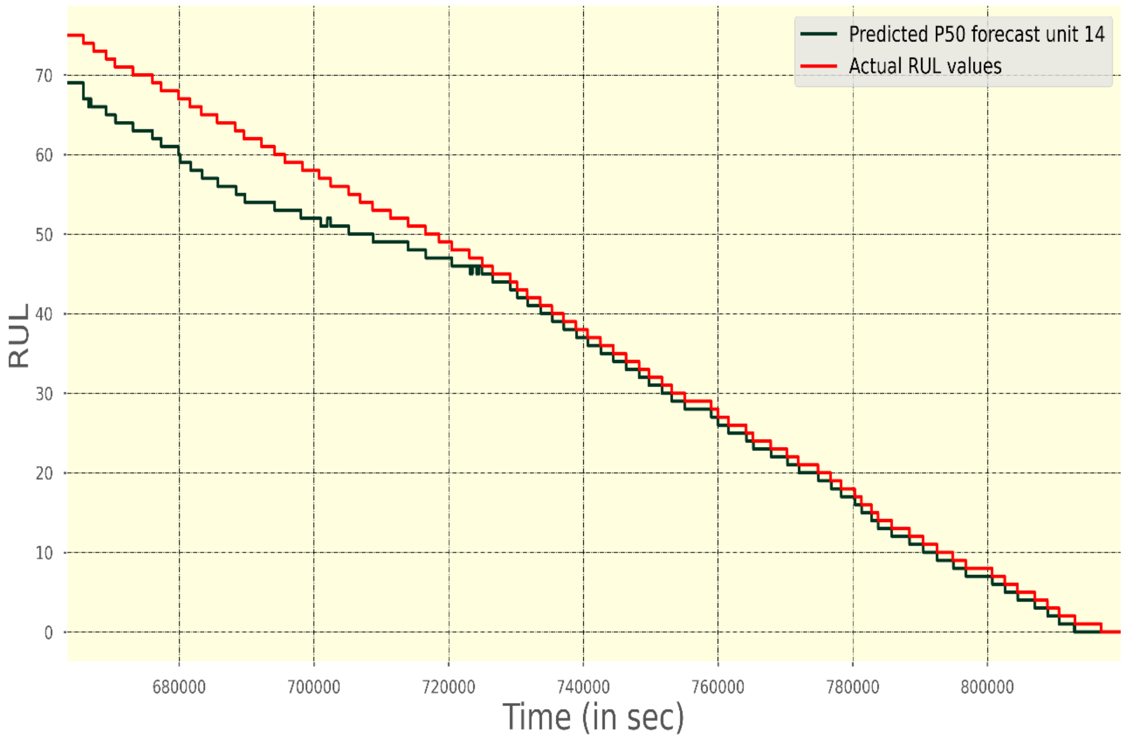
We evaluated the proposed approach on DS02 subset of NCMAPPS dataset [9] .

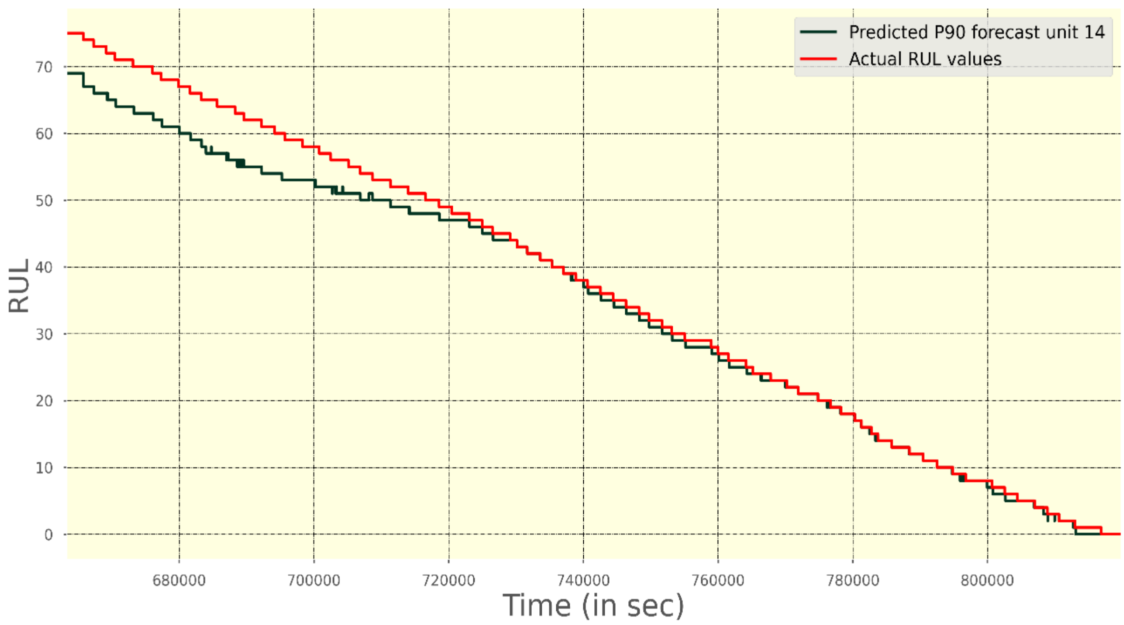
The following fixed parameters were used: Total Time Steps : 150, Encoder size : 145, Early Stopping Patience : 2, epochs: 6.

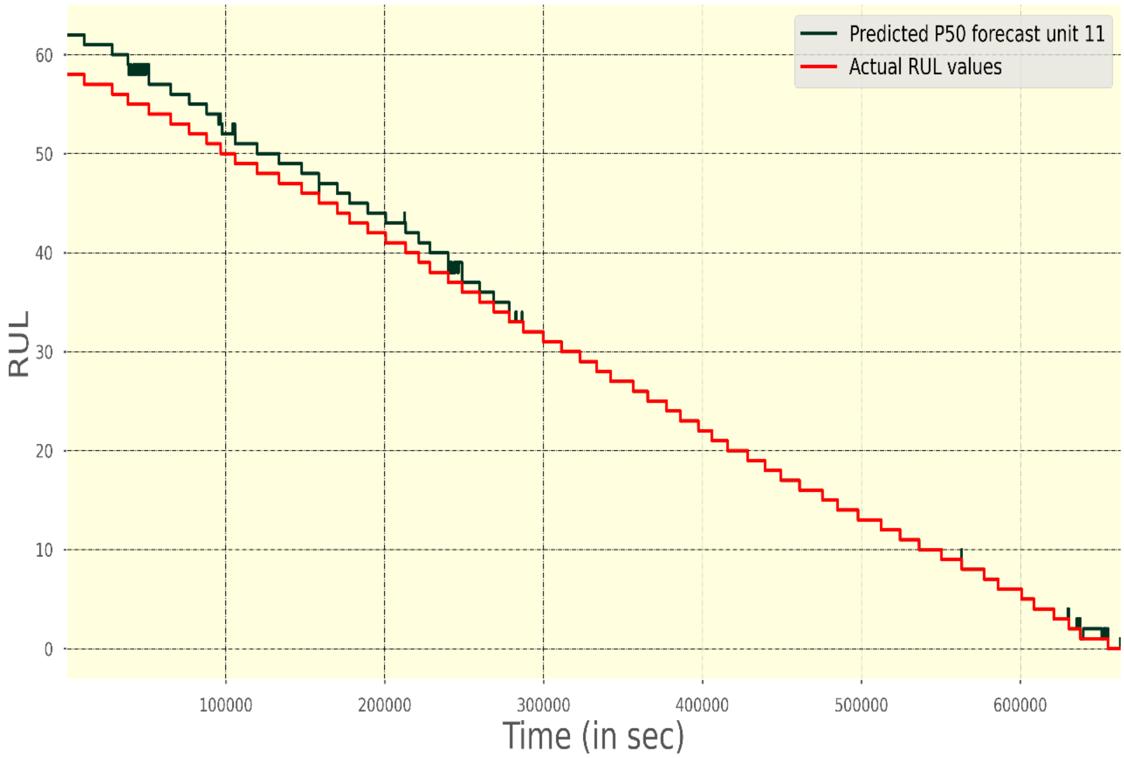
Following hyperparameters were used:
Dropout rate : 0.2, Minibatch size : 256, No.of heads : 4, Learning rate : 0.001, Hidden Layer Size: 150

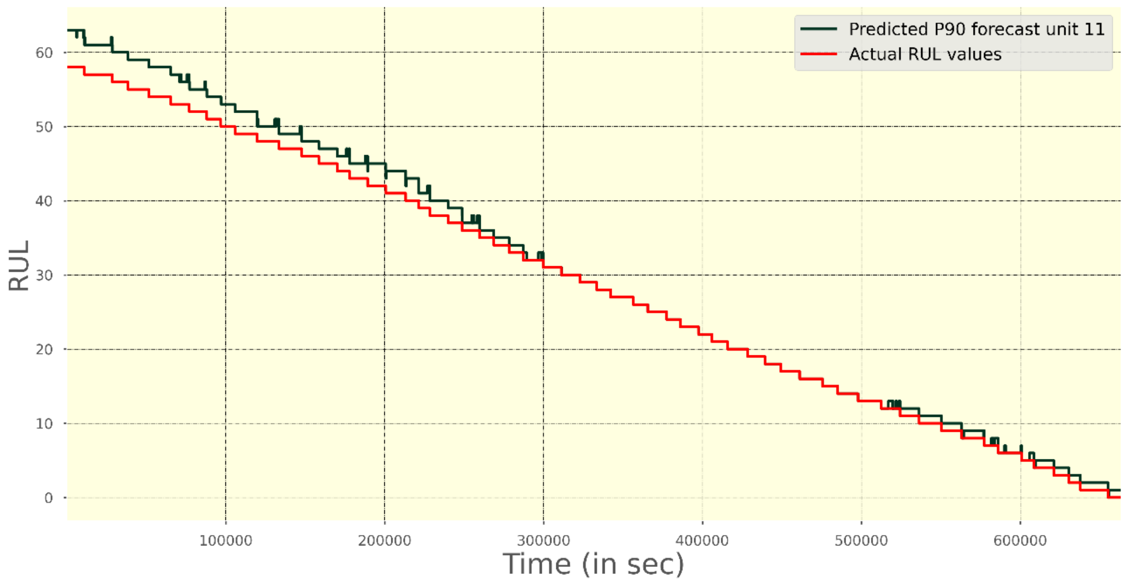

Predicted P50 forecast unit 15
Actual RUL values
RUL
60
50
40
30
20
10
0
0.85
0.90
0.95
1.00
1.05
1.10
1.15
1.20
1.25
1e6
Time (in sec)


Predicted P90 forecast unit 15
Actual RUL values
RUL
60
50
40
30
20
10
0
0.85
0.90
0.95
1.00
1.05
1.10
1.15
1.20
1.25
1e6
Time (in sec)


Predicted P50 forecast unit 14
Actual RUL values
RUL
Time (in sec)
70
60
50
40
30
20
10
0
680000
700000
720000
740000
760000
780000
800000


Predicted P90 forecast unit 14
Actual RUL values
RUL
Time (in sec)
70
60
50
40
30
20
10
0
680000
700000
720000
740000
760000
780000
800000


Predicted P50 forecast unit 11
Actual RUL values
RUL
Time (in sec)
0
10
20
30
40
50
60
100000
200000
300000
400000
500000
600000


Predicted P90 forecast unit 11
Actual RUL values
RUL
Time (in sec)
0
10
20
30
40
50
60
100000
200000
300000
400000
500000
600000

The evaluation metrics for NCMAPPS using our proposed approach and comparing with other State-Of-The-Art approaches:

	RMSE loss	**S-score**
TFT	On P90 forecast**: 8.0413879** On P50 forecast: **7.8239517**	On P90 forecast :**1413676.1969** On P50 forecast: **1352889.7026**
Physics based+ Deep Learning [10]	**4.22**	**43000**
Proposed Approach	On P90 forecast: **3.56417406** On P50 forecast: **1.573298**	On P90 forecast : **17481.60528** On P50 forecast: **9200.8525**

7. Conclusions

The proposed architecture has showed very promising results on prognostics datasets. We used the observed , known and static inputs and utilized our proposed model to predict the RUL of the engines . Our approach outperforms other methods as shown. Future work include working on Variable Selection Networks and potentially employing backcasting to further improve our architecture.

8. References

[1] Lim, B. and Zohren, S., 2020. Time series forecasting with deep learning: A survey. *arXiv preprint arXiv:2004.13408*.

[2] Deng, Kunyuan, Xiaoyong Zhang, Yijun Cheng, Zhiyong Zheng, Fu Jiang, Weirong Liu, and Jun Peng. "A remaining useful life prediction method with long-short term feature processing for aircraft engines." *Applied Soft Computing* 93 (2020): 106344.

[3] Ellefsen, A.L., Bjørlykhaug, E., Æsøy, V., Ushakov, S. and Zhang, H., 2019. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, *183*, pp.240-251.

[4] Lim, B., Arik, S.O., Loeff, N. and Pfister, T., 2019. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*.

[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

[6] Y. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, in: ICML, 2017.

[7] Yuan M, Wu Y, Lin L. Fault diagnosis and remaining useful life estimation of aero-engine using LSTM neural network. IEEE International Conference on Aircraft Utility Systems (AUS). IEEE; 2016. p. 135-140

[8] Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. 2017 IEEE International Conference on Prognostics and Health Management (ICPHM). IEEE; 2017. p. 88-95

[9] Chao, M.A., Kulkarni, C., Goebel, K. and Fink, O., 2021. Aircraft Engine Run-to-Failure Dataset under Real Flight Conditions for Prognostics and Diagnostics. *Data*, *6*(1), p.5.

[10] Chao, M.A., Kulkarni, C., Goebel, K. and Fink, O., 2020. Fusing physics-based and deep learning models for prognostics. *arXiv preprint arXiv:2003.00732*.