



# PREDICTIVE MAINTENANCE USING LONG SHORT-TERM MEMORY ALGORITHM ON SENSOR DATA

Suma Shruthika  
 PSG College of Technology  
 Coimbatore, Tamilnadu, India

**Abstract**— Think of a complex system with very expensive parts. We can't risk running into failure as it will be extremely costly to repair highly damaged parts. But more importantly, it's a safety issue. This is why numerous organizations attempt to avoid failure beforehand by performing regular inspections on their equipment. One big challenge is to determine when to do maintenance. Since we don't know when failure will occur, we have to be conservative in our planning. LSTM can be used to predict the remaining useful life. But if we schedule maintenance very early, we will end up wasting machine life that is still usable, and this will add up to our costs. However, if we can predict when machine failure will occur, we can schedule maintenance right before it. Recurrent Neural Networks can predict when this machine failure is bound to happen. Predictive maintenance lets us estimate time to failure. It also pinpoints problems in complex machinery and helps us identify what parts need to be fixed. This way, we can minimize downtime and maximize equipment lifetime.

**Keywords**— Predictive Maintenance · LSTM · Recurrent Neural Networks

## I. INTRODUCTION

An LSTM is built in order to predict when an in-service machine will fail and remaining useful life of aircraft engines so that the maintenance can be planned in advance. We can employ simulated aircraft engine run-to-failure events to demonstrate the predictive maintenance modeling process. Predictive Maintenance is also a domain where data is collected over time to monitor the state of an asset with the goal of finding patterns to predict failures which can also benefit from certain deep learning algorithms. Since they are very good at learning from sequences, LSTM networks among all the deep learning methods can be employed for predictive maintenance.

## II. PROPOSED ALGORITHM

We apply deep learning in predictive maintenance domain and uses a simple scenario where only one data source (sensor values) is used to make predictions.

### A. Data Ingestion –

Upon ingesting the training, test and ground truth datasets, we can observe that the training data comprises numerous multivariate time series having "cycle" as the time unit, together with 21 sensor readings for each cycle. The testing data has the same data schema as the training data. The main difference will be that the data does not be able to denote when the failure is bound to happen. Lastly, the ground truth data gives the number of remaining working cycles for the engines in the testing data.

id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21	
0	1	0.632184	0.750000	0.0	0.0	0.545181	0.310661	0.269413	0.0	0.220588	0.132160	0.308965	0.0	0.333333	0.0	0.0	0.558140	0.561834								
1	2	0.344828	0.250000	0.0	0.0	0.150602	0.379551	0.222316	0.0	0.264706	0.204768	0.213159	0.0	0.416667	0.0	0.0	0.682171	0.686827								
2	3	0.517241	0.583333	0.0	0.0	0.376506	0.346632	0.322248	0.0	0.220588	0.155640	0.458638	0.0	0.416667	0.0	0.0	0.728682	0.721348								
3	4	0.741379	0.500000	0.0	0.0	0.370482	0.285154	0.408001	0.0	0.250000	0.170090	0.257022	0.0	0.250000	0.0	0.0	0.666667	0.662110								
4	5	0.580460	0.500000	0.0	0.0	0.391566	0.352082	0.332039	0.0	0.220588	0.152751	0.300885	0.0	0.166667	0.0	0.0	0.658915	0.716377								

Fig. 1. Ground truth data set

### B. Data Modelling –

These machine learning models which are traditional in predictive maintenance are built on feature engineering which is used for manual construction of apt features employing domain expertise in addition to related methods. Models are difficult to be used again as feature engineering is particular about the scenario of the problem and the data which is available differs for every business. Possibly the most captivating feature of using deep learning in the domain of predictive maintenance is the reason that these networks can automatically extract the right features from the data, eliminating the need for manual feature engineering.

When using LSTMs in the time-series domain, one important parameter to pick is the sequence length which is the window for LSTMs to look back. This may be viewed as similar to picking window\_size = 5 cycles for calculating the rolling features which are rolling mean and rolling standard deviation



for 21 sensor values. The idea of using LSTMs is to let the model extract abstract features out of the sequence of sensor values in the window rather than engineering those manually. The expectation is that if there is a pattern in these sensor values within the window prior to failure, the pattern should be encoded by the LSTM. One critical advantage of LSTMs is their ability to remember from long-term sequences (window sizes) which is hard to achieve by traditional feature engineering. For example, computing rolling averages over a window size of 50 cycles may lead to loss of information due to smoothing and abstracting of values over such a long period, instead, using all 50 values as input may provide better results. While feature engineering over large window sizes may not make sense, LSTMs are able to use larger window sizes and use all the information in the window as input. Below, we illustrate the approach.

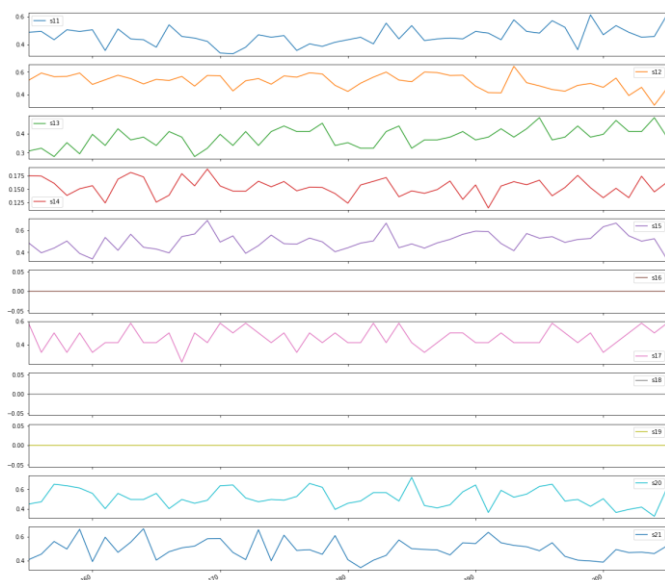


Fig. 2. Sensor Data Plot for Sensor ID 3 Prior to a Failure Point

### C. LSTM Network –

Next, we build a deep network. The initial layer constitutes an LSTM layer of 100 units accompanied by another LSTM layer having 50 units. The last layer is a dense output layer having a single unit and sigmoid activation as it is a binary classification problem.

Next, we look at the performance on the test data. In the final cycle data for every engine id present in the test data will be kept for the purpose of testing. In order to compare the results to the template, we pick the last sequence for each id in the test data.

	Accuracy	Precision	Recall	F1-score
<b>LSTM</b>	0.978495	0.960000	0.96	0.960000
<b>Template Best Model</b>	0.940000	0.952381	0.80	0.869565

Fig. 3. Test Result Comparison

### III. EXPERIMENT AND RESULT

This research paper covers the basics of using deep learning in predictive maintenance and many predictive maintenance problems usually involve a variety of data sources that needs to be taken into account when applying deep learning in this domain. Additionally, it is important to tune the models for the right parameters such as window size.

### IV. REFERENCE

- [1] Adichie, J. N. [1967], “Estimates of regression parameters based on rank tests,” *Ann. Math. Stat.*, **38**, 894–904.
- [2] Aitkin, M. A. [1974], “Simultaneous inference and the choice of variable subsets,” *Technometrics* **16**, 221–227.
- [3] Allen, D. M. [1971], “Mean square error of prediction as a criterion for selecting variables,” *Technometrics*, **13**, 469–475.
- [4] Allen, D. M. [1974], “The relationship between variable selection and data augmentation and a method for prediction,” *Technometrics*, **16**, 125–127.
- [5] Andrews, D. F. [1974], “A robust method for multiple linear regression,” *Technometrics*, **16**, 523–531.
- [6] Neter, J., W. Wassermann and M. J. Kutner 1983. Applied linear statistical models. Richard D. Irwin Inc. Homewood, Illinois.
- [7] Prairie, Y. T. 1996. Evaluating the predictive power of regression models. *Can. J. Fish. Aquat. Sci* **53**: 490-492.