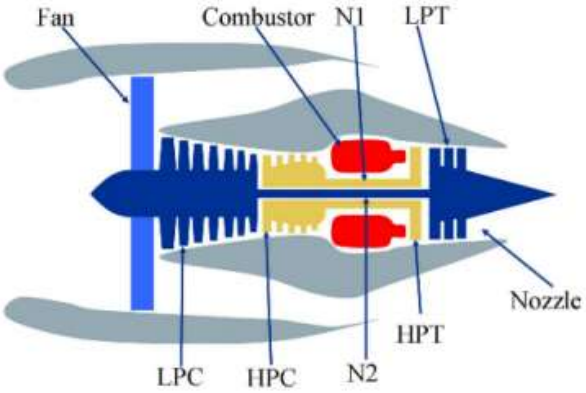
REPORT

A. Problem Definition:

Predictive maintenance is important part of applications especially require costly engine maintenance such as power plant, aircraft, automotive, factory automation etc. It uses predictive tools for maintenance actions if necessary via allows early detection of failures. Recently, with significant improvements in deep learning area and the amount of data extracted from production processes has increased exponentially, data-driven approaches for remaining useful life studies have been attracting a lot of attention. However, a major challenge in data-driven prognostics is that it is often impossible to obtain a large number of samples for failure progression, which is costly and labor demanding. For this reason, we used machine learning approach on turbofan engine dataset to solve this problem.

Data Overview

The degradation data of the turbofan engine will be used in this work was simulated by C-MAPSS developed by NASA [17]. A simplified diagram of the simulation engine is shown in Fig. 1.



The main components includes: fan, low pressure compressor (LPC), high pressure compressor (HPC), combustor, high pressure turbine (HPT), low pressure turbine (LPT), and nozzle. C-MAPSS was developed on MATLAB software and Simulink environments. It includes many editable input parameters that allow the user to enter specific values, such as Fuel flow, Fan flow modifier, and Fan pressure-ratio modifier, etc. The input to the C-MAPSS contains 14 factors that affect the degradation of the turbofan engine, and the output of the simulation model represents the health condition of the turbofan engine. A description of the 21 simulation outputs of C-MAPSS is shown in Table 1.

The legend of column 5 'Trend' represents the degradation trend of the output, where ↑ indicates that the parameter is ascending with time, ↓ indicates that the parameter is descending with time, and ~ indicates that the parameter is irregular with time. The C-MAPSS dataset can be divided into four sub datasets according to different operating conditions and fault modes. A description of four sub-datasets is given in Table 2.

	Symbol	Description	Units	Trend
1	T2	Total Temperature at fan inlet	°R	~
2	T24	Total temperature at LPC outlet	°R	↑
3	T30	Total temperature at HPC outlet	°R	↑
4	T50	Total temperature LPT outlet	°R	↑
5	P2	Pressure at fan inlet	psia	~
6	P15	Total pressure in bypass-duct	psia	~
7	P30	Total pressure at HPC outlet	psia	↓
8	Nf	Physical fan speed	rpm	↑
9	Nc	Physical core speed	rpm	↑
10	Epr	Engine pressure ratio	--	~
11	Ps30	Static pressure at HPC outlet	psia	↑
12	Phi	Ratio of fuel flow to Ps30	pps/psi	↓
13	NRf	Corrected fan speed	rpm	↑
14	NRc	Corrected core speed	rpm	↓
15	BPR	Bypass ratio	--	↑
16	farB	Burner fuel-air ratio	--	~
17	htBleed	Bleed enthalpy	--	↑
18	NF_dmd	Demanded fan speed	rpm	~
19	PCNR_dmd	Demanded corrected fan speed	rpm	~
20	W31	HPT coolant bleed	lbm/s	↓
21	W32	LPT coolant bleed	lbm/s	↓

Table 1: 21 sensor outputs of simulation engine running.

Sub-datasets	FD001	FD002	FD003	FD004
Engines in training set	100	260	100	249
Engines in test set	100	259	100	248
Max/min cycles for training	362/128	378/128	525/145	543/128
Max/min cycles for test	303/31	367/21	475/38	486/19
Operating condition	1	6	1	6
Fault modes	1	1	2	2
TW length	30	21	36	18
Training samples	17731	48558	21120	56815
Test samples	100	259	100	248

Table 2: Description of the C-MAPSS dataset.

Each sub-dataset contains training data, test data, and the actual RUL corresponding to the test data. The training data contains all the engine data from a certain health state to the fault, while the test data is a piece of data before the engine running fault. Moreover, the training and test data respectively contain a certain number of engines with different initial health states. Due to the different initial health states of the engines, the running cycles of different engines in the same database are different. Taking the FD001 database as an example, the test dataset contains 100 engines, with a maximum running cycle of 303 and a minimum running cycle of 31.

For our case we only use first dataset that include total 100 engines data.

B. Data Expolaration

Data Distribution

For data analysis we used pandas library in python. For each sensors and settings features, we plot histogram, boxplot, time series data and scatter plot with label-regression label (RUL)-

Detailed and plots of them are given following:

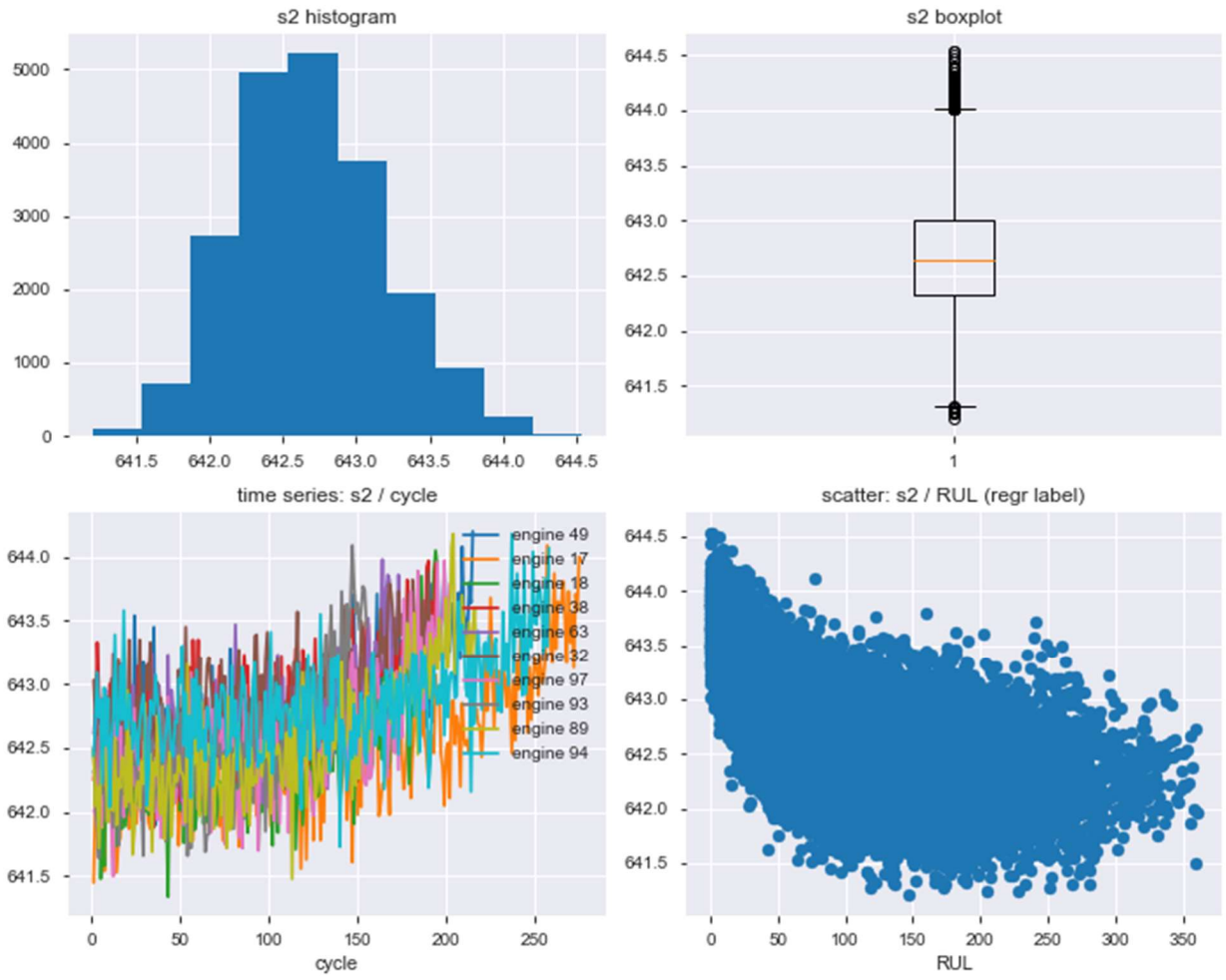


Fig-1: Sensor 2 distribution info graph

As we see for above graph, sensor2 has normal distribution in histogram with has some outliers as seen in box plot. However, in third figure, we see sensor2 time series data with cycles for 10 different engines and their values are as pass time increase. Also scatter plots of sensor2 are given with time remaining useful life of engine with give some information to us for correlation-negative-.

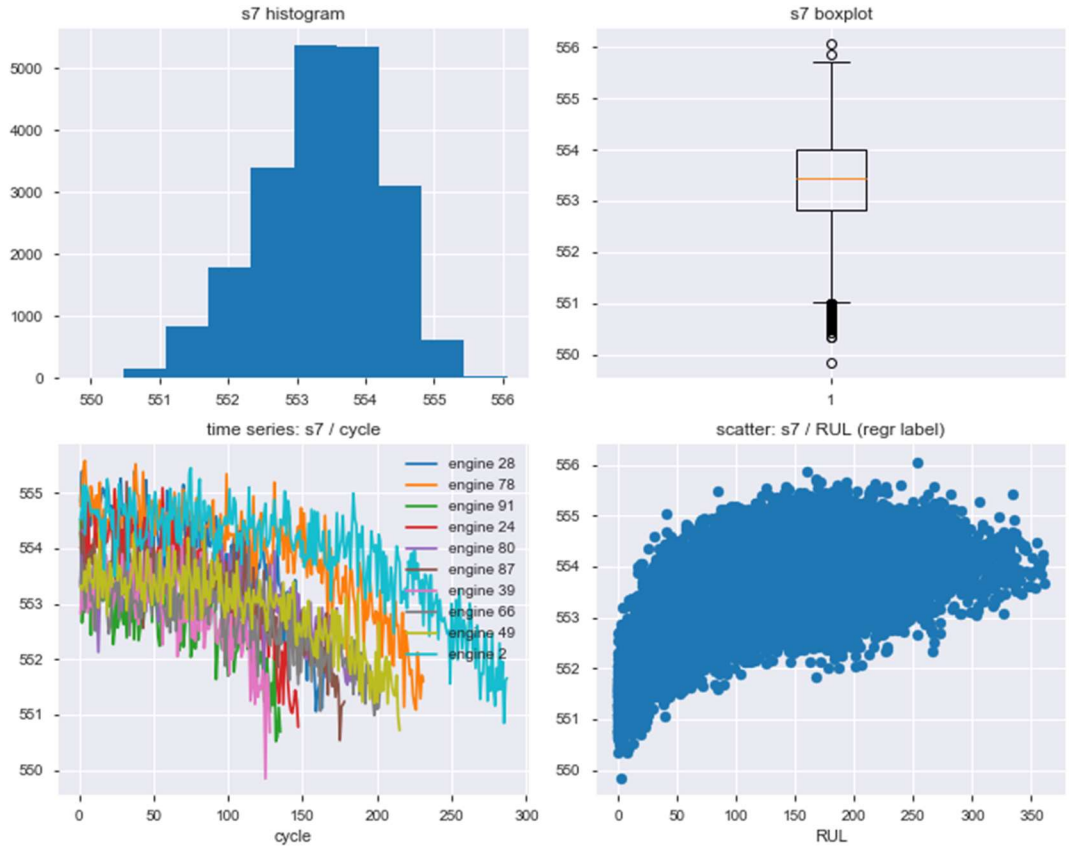


Fig-2: Sensor 7 distribution info graph

As we see for above graph, sensor7 has normal distribution in histogram with has some outliers as seen in box plot. However, in third figure, we see sensor7 time series data with cycles for 10 different engines and their values are as pass time decrease. Also scatter plots of sensor7 are given with time remaining useful life of engine with give some information to us for correlation-positive-

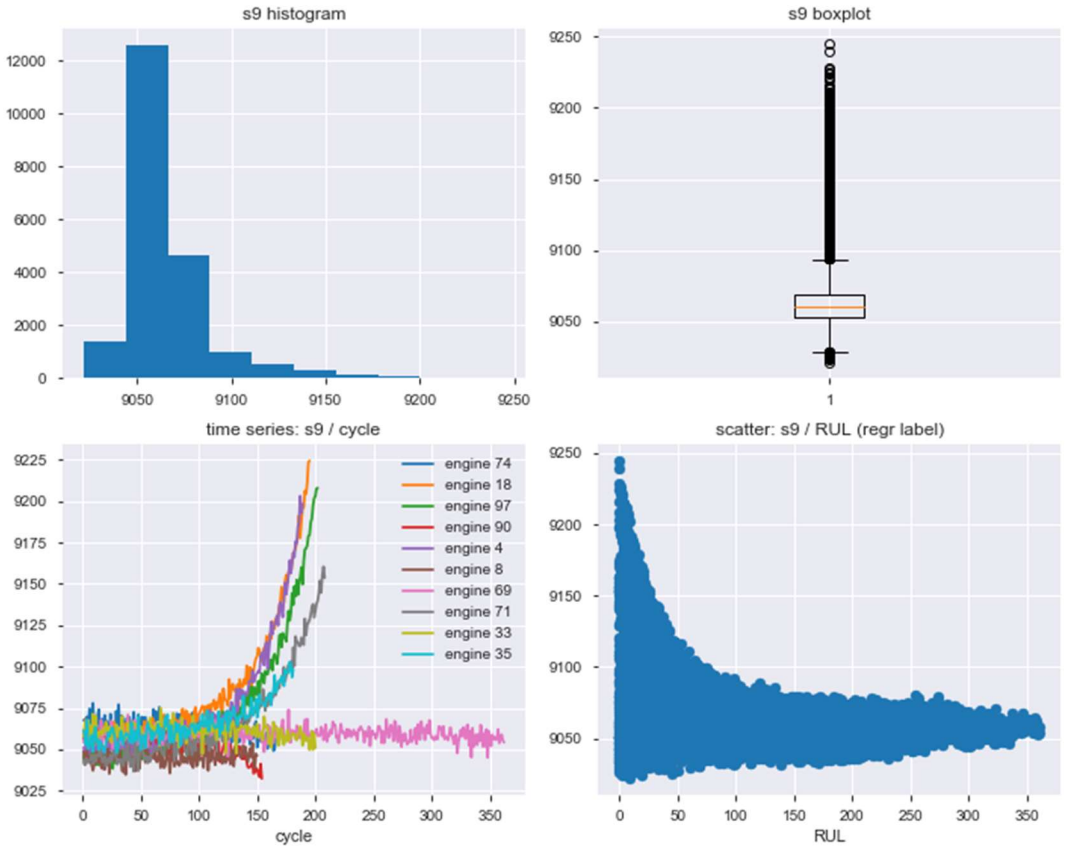


Fig-3: Sensor 9 distribution info graph

As we see for above graph, sensor9 has skewed left distribution in histogram with has some large outliers as seen in box plot. However, in third figure, we see sensor9 time series data with cycles for 10 different engines and their values are as pass time decrease with some of them decrease. Also scatter plots of sensor9 are given with time remaining useful life of engine with give some information to us for correlation-not negative or positive-

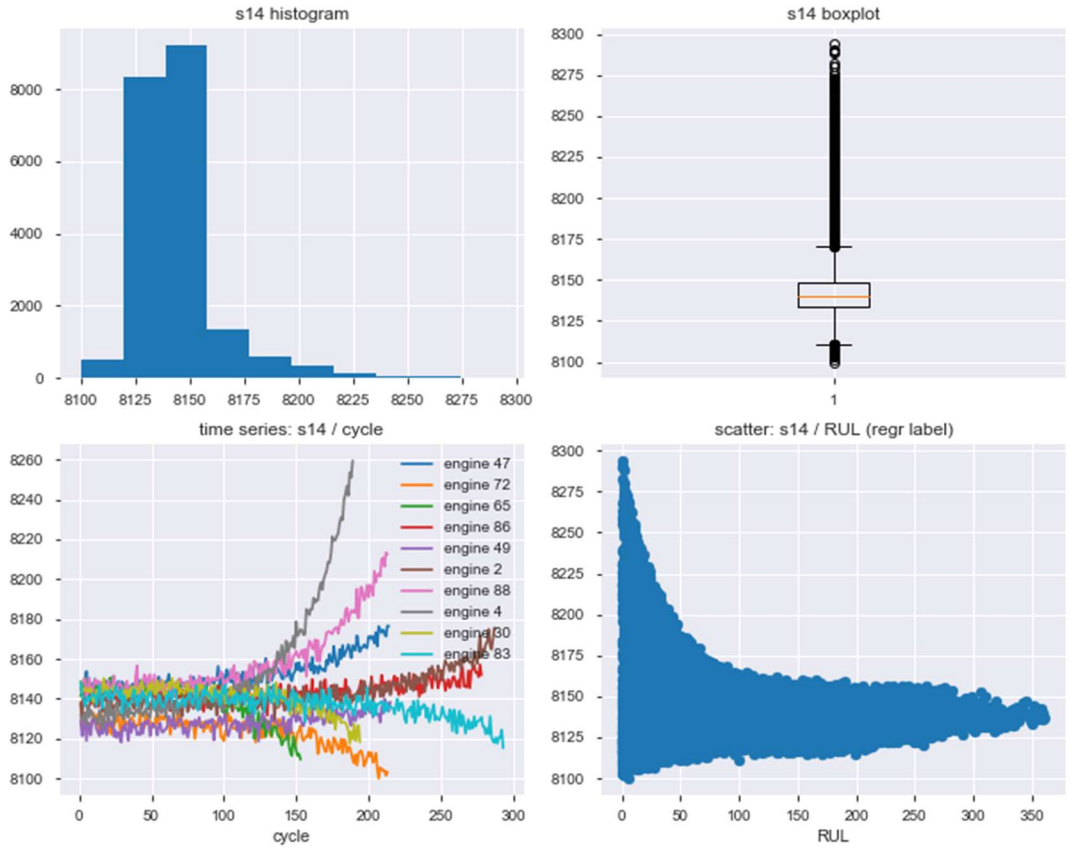


Fig-4: Sensor 14 distribution info graph

As we see for above graph, sensor14 has skewed left distribution in histogram with has some large outliers as seen in box plot. However, in third figure, we see sensor14 time series data with cycles for 10 different engines and their values are as pass time decrease with some of them decrease. Also scatter plots of sensor14 are given with time remaining useful life of engine with give some information to us for correlation-not negative or positive-

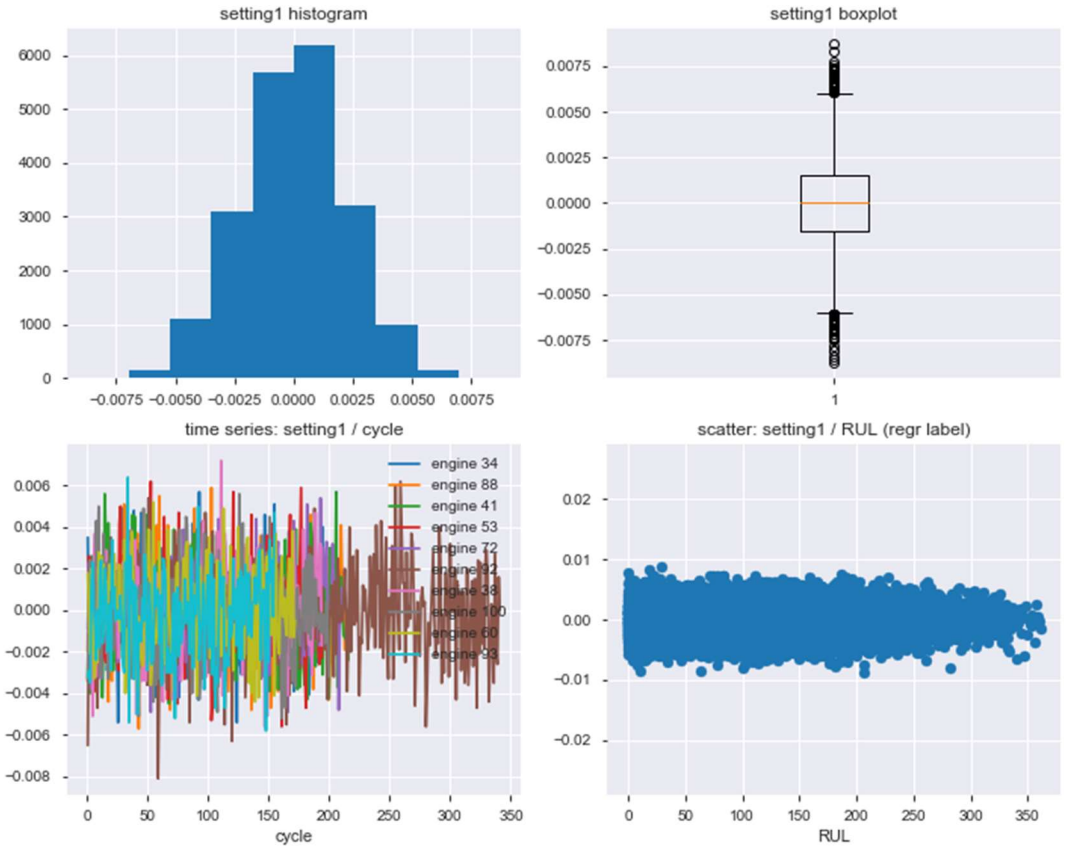


Fig-5: Setting 1 distribution info graph

As we see for above graph, setting1 has normal distribution in histogram with has some large outliers as seen in box plot. However, in third figure, we see setting1 time series data with cycles for 10 different engines and their values are as pass time not decrease with some of them not decrease. Also scatter plots of setting1 are given with time remaining useful life of engine with give some information to us for correlation-not negative or positive-

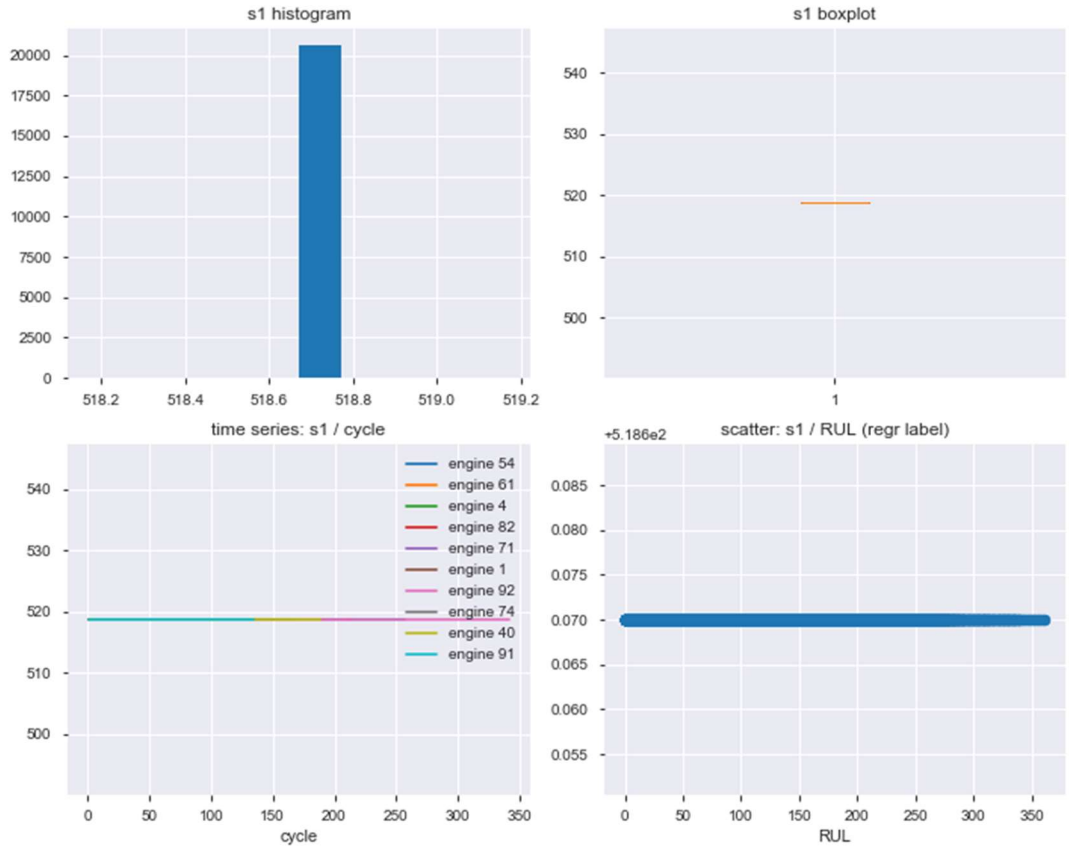


Fig-6: Sensor 1 distribution info graph

As we see for above graph, sensor1 has unique value.

As general data distributions of other sensors are is similar one of above figures. So, we only show some of them in here. However, some of them include only one unique value as sensor1 data, we are remove them from our data for training and they are sensor1, sensor5, sensor10, sensor16, sensor18, sensor19, setting3.

Also, we plot time series data individually for each sensor data with 10 random engine data for each them. Detailes are given following:

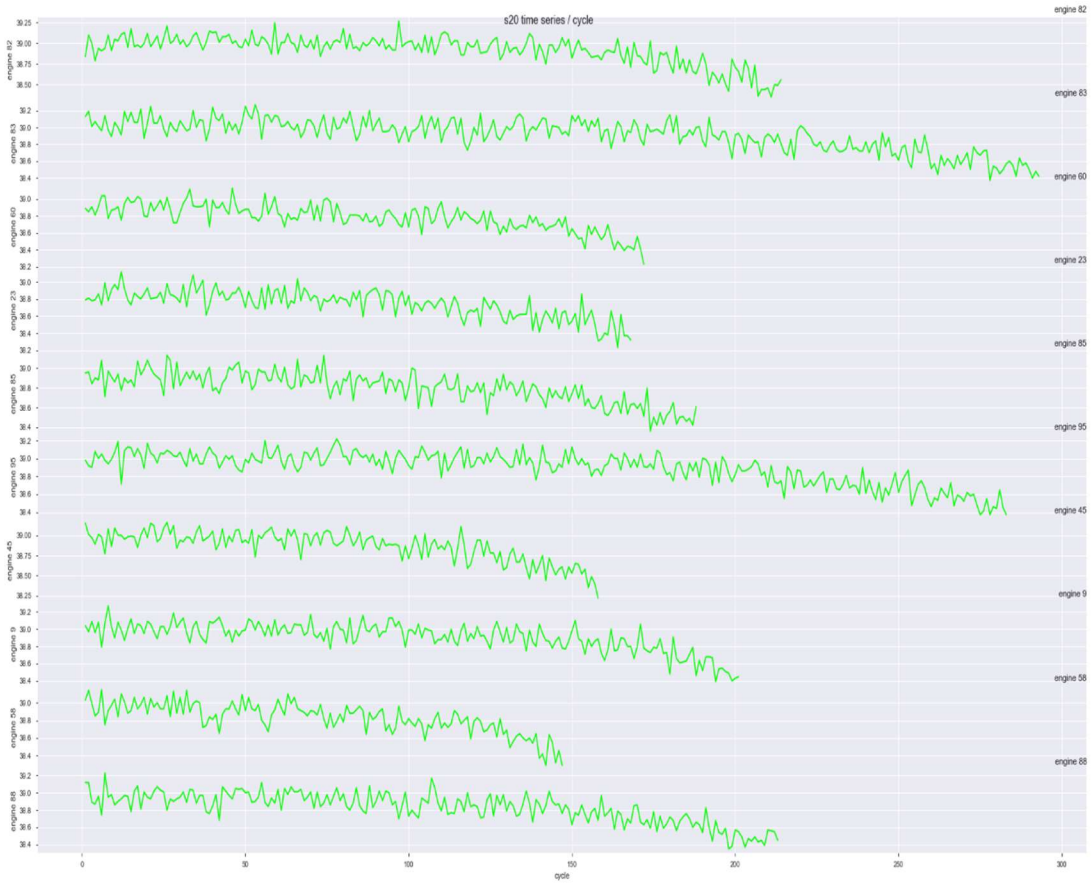


Fig-7: Sensor20 time series data info graph

As we see for above graph, sensor20 for each engine has similar trend and decreases as time pass over.

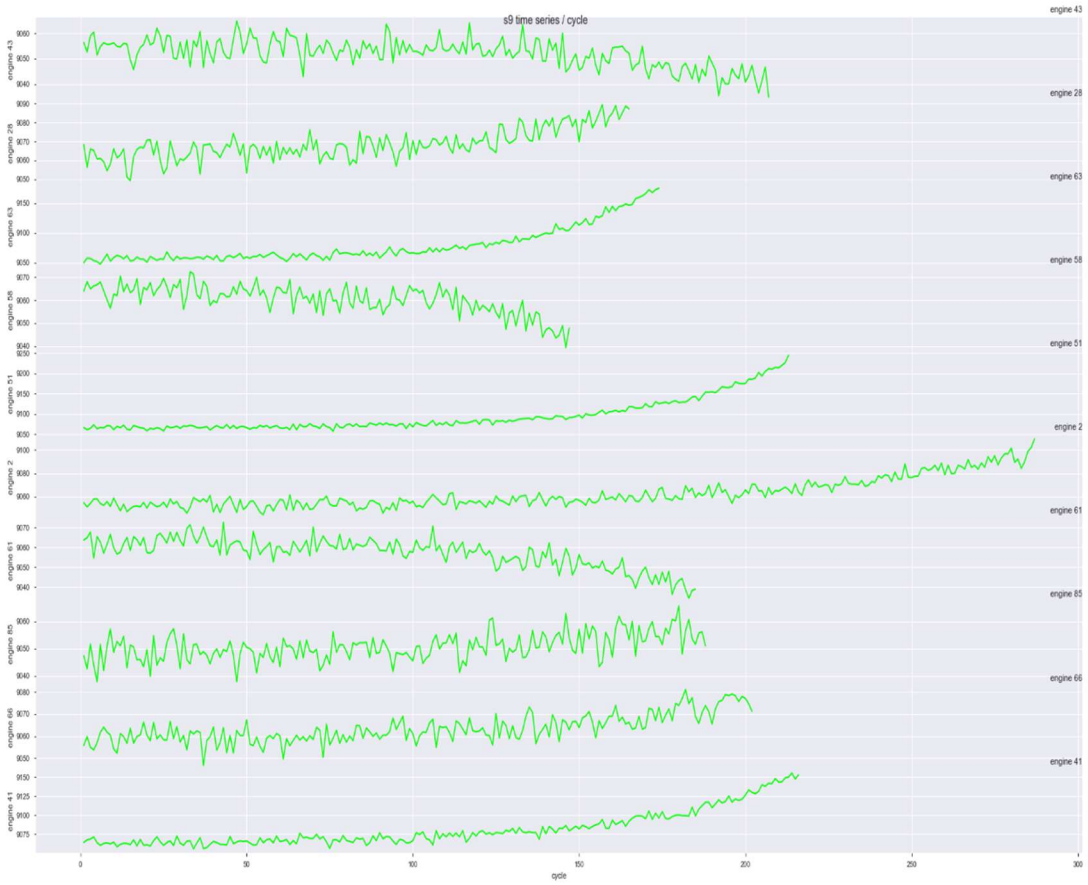


Fig-8: Sensor9 time series data info graph

As we see for above graph, sensor9 for each engine has different trend and decreases or increases as time pass over.

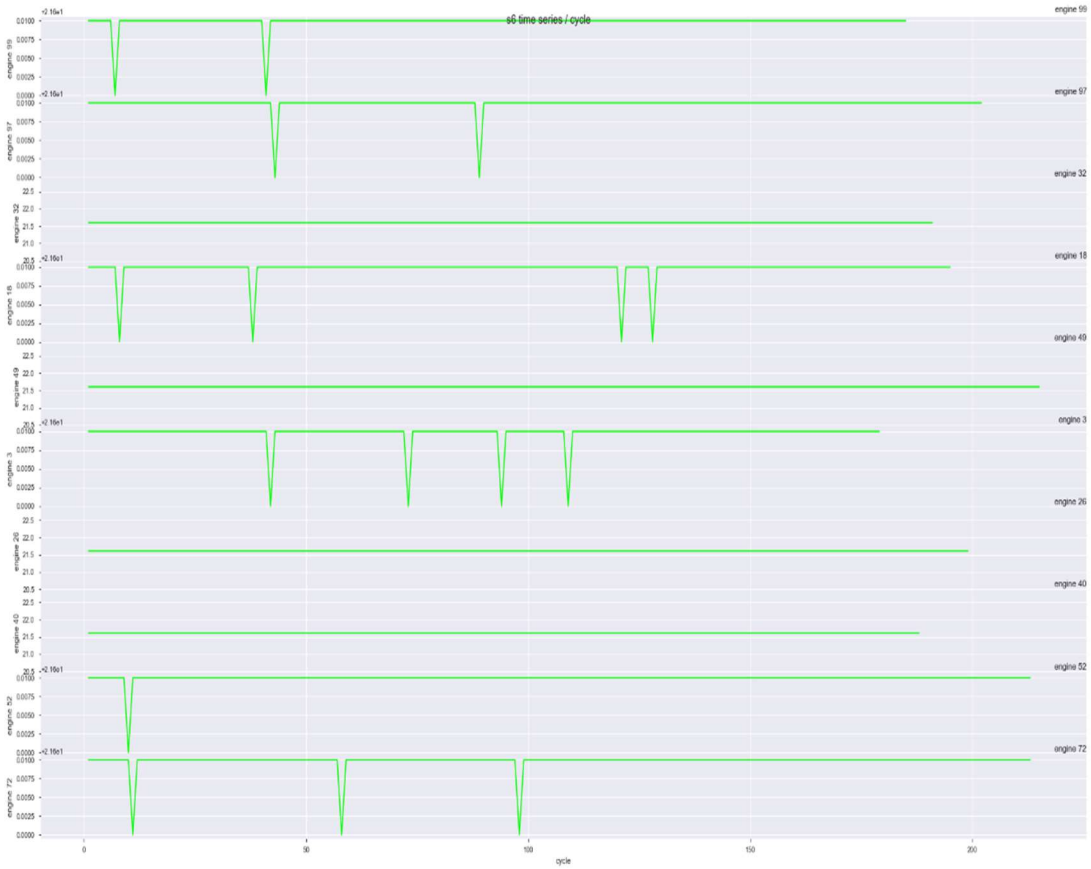


Fig-9: Sensor6 time series data info graph

As we see for above graph, sensor6 for each engine has similar almost the same trend and decreases in some time.

Other data has similar trends. We only take here different samples.

Train Data

For our data, we can apply binary classification, multiclass classification or regression analysis. For this purpose, we should generate labels.

- **RUL -** for regression – how much time to before failure of engine – as cycle –
- **BINC**-binary classification – last 30 cycle before failure as 1, otherwise 0
- **MCC**-multiclass classification – last 15 cycle before failure as 2, last [16-30] cycle before failure as 1, otherwise 0

Also, our classification labels rates as following:

Record #/% for each class-binary classification- :
0 17531
1 3100
Name: BINC, dtype: int64

Negative samples = 85%

Positive samples = 15%

Record #/% for each class-multi class classification- :
0 17531
2 1600
1 1500
Name: MCC, dtype: int64

Class 0 samples = 85%

Class 1 samples = 7%

Class 2 samples = 8%

Binary classification Results

Metrics/ Algorithm	Logistic Regression	Decision Tree	Random Forest	Gradient Boosting	Bagging Classifier
Accuracy	0.9846	0.9766	0.9865	0.9870	0.9853
f1_score	0.6479	0.5513	0.6955	0.7128	0.6595

Table -1 Binary classification results

As we see Random Forest give the best results. Also confusion matrix is given flowing:

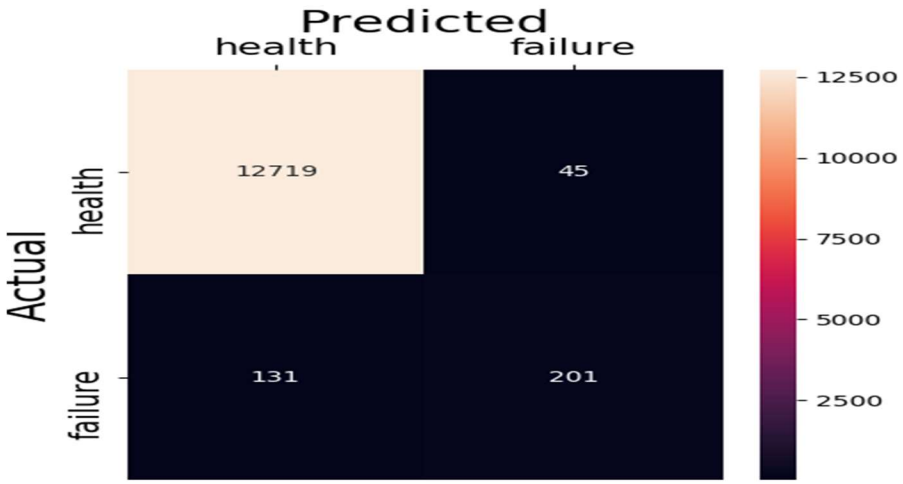


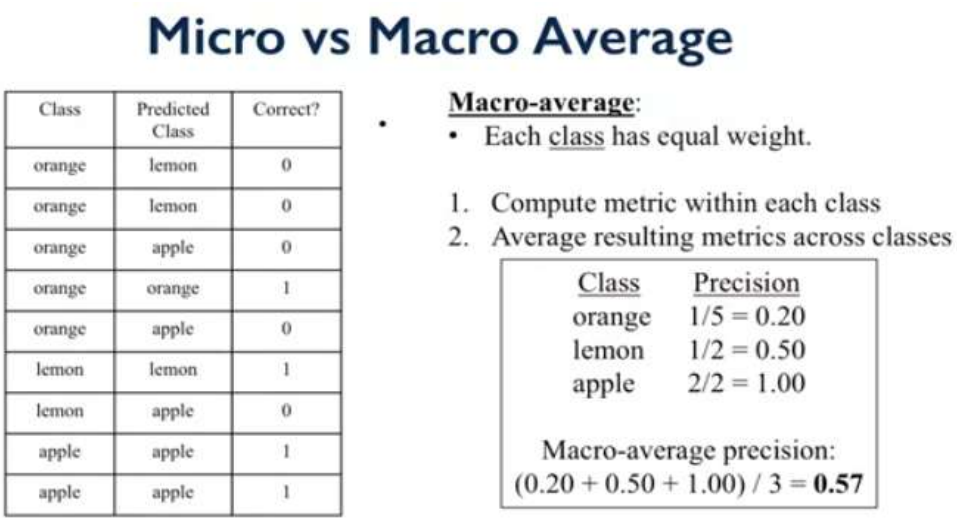
Fig-10 Confusion matrix of Random Forest

Multiclass classification Results

Metrics/ Algorithm	Logistic Regression	Decision Tree	Random Forest	Gradient Boosting	Bagging Classifier
f1_macro	0.6028	0.6304	0.7153	0.7227	0.6556
f1_micro	0.9793	0.9748	0.9835	0.9835	0.9809

Table -2 Multiclass classification results

As we see Random Forest give the best results. Also scores calculations are given following:



For our case labels 0, 1, and 2 can be considered as orange, apple and lemon and macro, and micro average calculated.

Regression analysis

Metrics/ Algorithm	Logistic Regression	Decision Tree Reg	Random Forest Reg	Gradient Boosting R.	Bagging Classifier R.
RMSE	0.1608	0.1669	0.1160	0.1175	0.1214
r2_score	0.3223	0.2703	0.6472	0.6381	0.6141

Table -3 Regression results

As we see Random Forest give the best results.