Prediction of Remaining Useful Life(RUL) for Aircraft Engine Using Neural Network Models

Ying Liao
Jian Guo
Cameron Cortines
Pham Van Vung
Sevgi Arca
Faranak Abri

Outline

1. Introduction & Motivation
2. Data exploration
3. Data preprocessing
4. Sensor selection
5. Neural Network models
6. Conclusion and future work

Introduction

- Estimation of the *Remaining Useful Life* (RUL) of a set of unspecified components
- Data-driven approach
- Data consists of multivariate time series for each unit
- Data provided was split into training and test sets beforehand
- Test data components were not run to failure

Motivation

RUL

- The amount of time a component can be expected to continue operating.
- High importance in PHM
- Learn directly from the data

MSE (Mean Square Error):

$$\frac{\sum_{u=1}^{n}\sum_{c=1}^{m_u}\left(R_{true}-R_{estimated}\right)^2}{\sum_{u=1}^{n} m_u}$$

Data exploration

- ➢ C-MAPSS
- ➢ Clustering of operational conditions
- ➢ MDS for visualization

C-MAPSS

- ➢ Input: Health related parameters

  Fuel flow, 13 component-related health parameters and etc.

- ➢ Output: sensor measurement(temperature; pressure; speed)

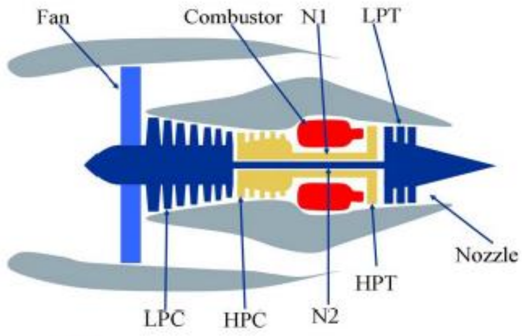


Figure 1. Simplified diagram of engine simulated in C-MAPSS

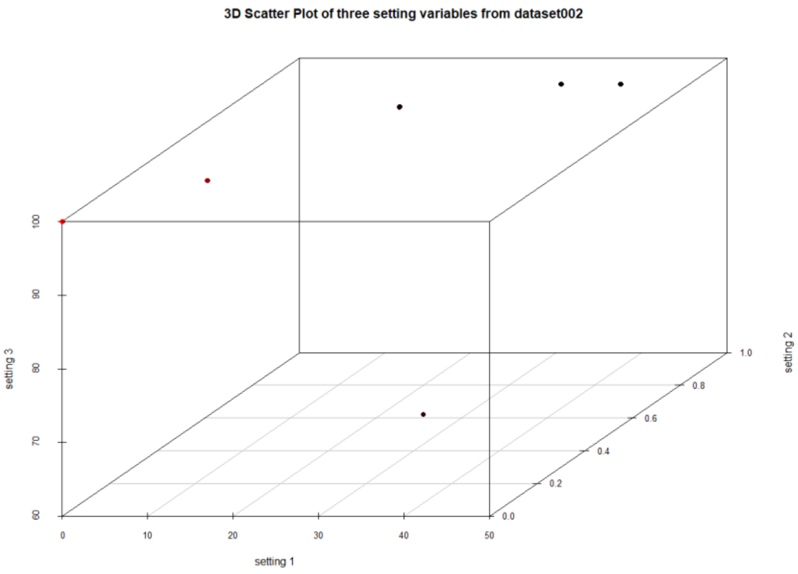
Name
Fuel flow
Fan efficiency modifier
Fan flow modifier
Fan pressure-ratio modifier
LPC efficiency modifier
LPC flow modifier
LPC pressure-ratio modifier
HPC efficiency modifier
HPC flow modifier
HPC pressure-ratio modifier
HPT efficiency modifier
HPT flow modifier
LPT efficiency modifier
HPT flow modifier

Operational conditions

1. Altitude: 0~0.84
2. Mach number: 20~100
3. Throttle resolver angle(TRA): 0~42

Unit	Cycle	Altitude	Mach #	TRA	Sensor 1	…	Sensor 21
1	1	34.9983	0.84	100	449.44	…	8.8071
1	2	41.9982	0.8408	100	445	…	6.2665
…	…	…	…	…	…	…	…
1	149	42.0017	0.8414	100	445	…	6.2285
2	1						
2	2						
…	…						

K-means for clustering



1. Make initial guesses for the means m1, m2,...,mk
2. Until there is no change in any mean

   a) assign each data point to the cluster whose mean is the nearest;

   b) calculate the mean of each cluster;

   c) for i from 1 to k:

      Replace mi with the mean of all examples for Cluster i

1. End until

Visualization of whole dataset--Sammon Mapping $\Re^d \rightarrow \Re^q$

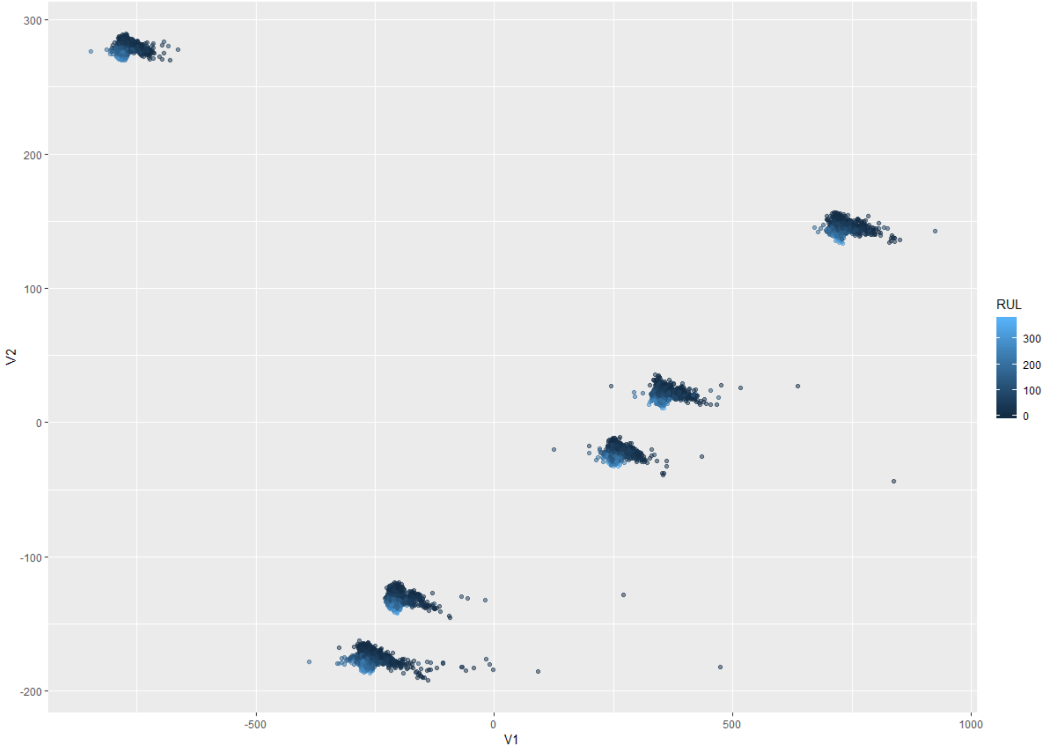
Sammon's stress: $$E = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j}^{n} \frac{(d^*_{ij} - d_{ij})^2}{\boxed{d^*_{ij}}}$$

Small distance→ high weight

Distance between point i and point j in the original space: d^*_{ij}

Distance between their projections: d_{ij}

Sammon mapping



Train Dataset #2: 53759 instances

Randomly choose 8000 instances in the dataset for visualization.

Normalization

Standard normalization:

$$N\left(x^d\right) = \frac{x^d - \mu^d}{\sigma^d}, \forall d$$

Modified normalization:

$$N\left(x^{(m,d)}\right) = \frac{x^{(m,d)} - \mu^{(m,d)}}{\sigma^{(m,d)}}, \forall m, d$$

Time Representation

The quickest and simplest method:

consider each time point independently →

predict at each time step

An alternative method:

Phase space embedding representation →

- a sequence of instances is generated using a fixed length sliding window
- Drawback: curse of dimentionality

Adding Useful Features

Adding extra seven **useful** features containing:

cls :

6 clusters based on operational conditions

mode1 ~ mode6:

Total of the number of cycles spent in each mode since the beginning of the series.

mode1	mode2	mode3	mode4	mode5	mode6	cls	unit	cycle	setting1	setting2	setting3	feature1	...	feature 21
0	0	1	0	0	0	3	1	1	34.9983	0.84	100	449.44		1358.61
0	0	2	0	0	0	3	1	2	41.9982	0.8408	100	445		1353.22
0	0	2	0	0	1	6	1	3	24.9988	0.6218	60	462.54		1256.76
0	0	3	0	0	1	3	1	4	42.0077	0.8416	100	445		1354.03
0	0	3	0	0	2	6	1	5	25.0005	0.6203	60	462.54		1257.71
0	0	3	1	0	2	4	1	6	25.0045	0.6205	60	462.54		1266.38
0	0	4	1	0	2	3	1	7	42.0043	0.8409	100	445		1347.45
0	1	4	1	0	2	2	1	8	20.002	0.7002	100	491.19		1481.69
0	1	5	1	0	2	3	1	9	41.9995	0.8407	100	445		1348.23
0	1	6	1	0	2	3	1	10	42.0011	0.84	100	445		1356.4
0	1	7	1	0	2	3	1	11	42.0029	0.84	100	445		1352.72
0	1	7	1	1	2	5	1	12	0.0015	0.001	100	518.67		1585.52
0	2	7	1	1	2	2	1	13	20.0003	0.7	100	491.19		1488.74
0	2	8	1	1	2	3	1	14	42.002	0.8407	100	445		1352.69
0	3	8	1	1	2	2	1	15	10.0038	0.2513	100	489.05		1501.72
...	...	...												

Sensor/feature Selection

- 21 sensors/features : feature1 to feature 21
- Goal: select the most valuable features for training
- Methods: Univariate feature selection and PCA

Univariate feature selection

- A filter method

- Use SelectKBest : scores the features using a function: χ^2 (Chi-squared) test function
- Then remove all but the k highest scoring features that have maximum relevance/dependence with the target variable.

Select K Best using Chi-squared test

1. Compute Chi-square:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

O: the observed(actual) value *E: the expected value*

1. Compute p value using Chi-square using Chi-Square Distribution Calculator

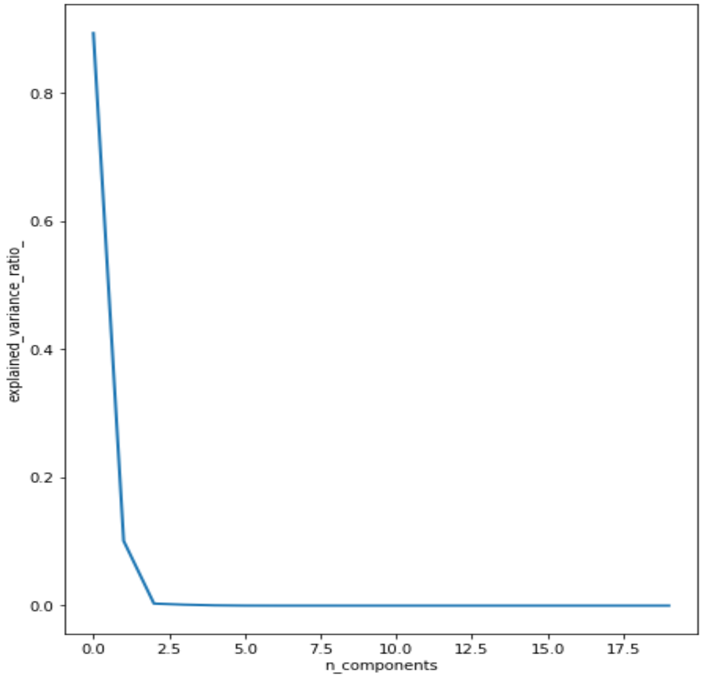
P value : the probability that the variables are **independent.**

1. **$p < 0.05$** is the usual test for dependence.
2. Pick k features that has **p** value less than 0.05
3. Result: feature 3, 4, 7, 9, 12, 18 are selected

PCA- Feature Extraction

- Use normalized training dataset
- Perform PCA on 21 features
- 2 components represent most of

  the features



PCA- Feature Extraction

- Each principal component is a linear combination of the original features

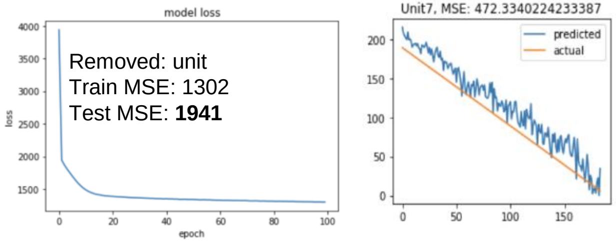
$$\mathrm{PC}^j = \beta_1^j X_1 + \beta_2^j X_2 + \cdots + \beta_n^j X_n$$

X_i: original features

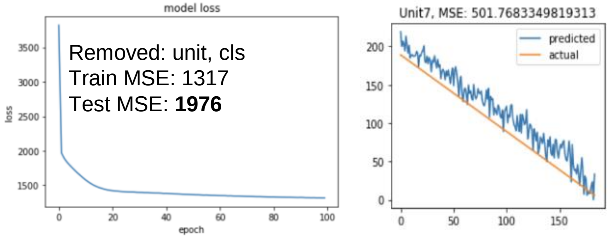
β_i: corresponding weights/coefficients

- To find what features contribute the most in the components, we just need to find the features with the highest weights/coefficients (absolute value)
- Result: feature 1, 6, 7, 8, 12, 18 are selected.

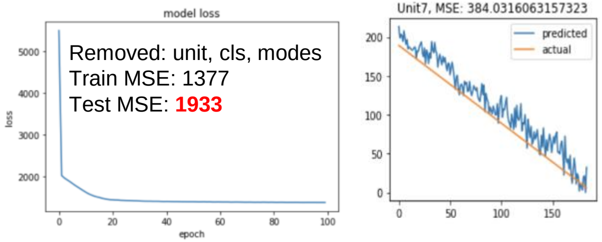
MLPs on original data



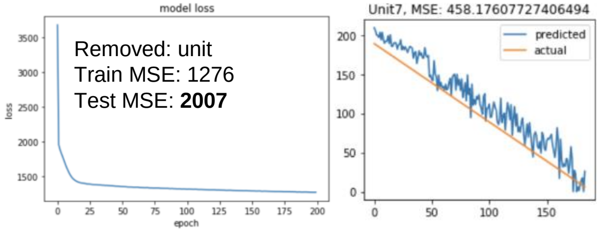
Removed: unit
Train MSE: 1302
Test MSE: **1941**



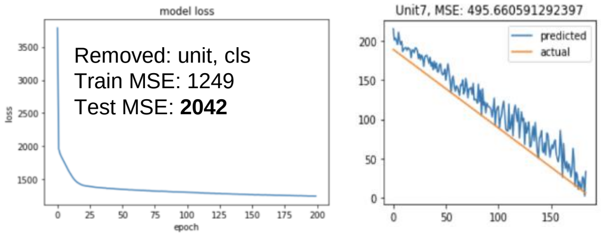
Removed: unit, cls
Train MSE: 1317
Test MSE: **1976**



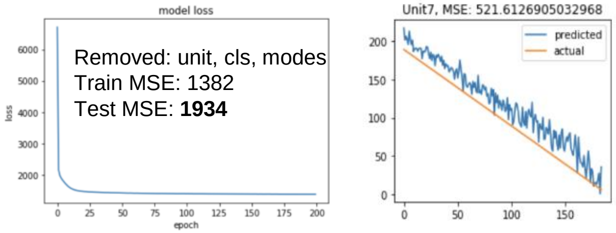
Removed: unit, cls, modes
Train MSE: 1377
Test MSE: **1933**



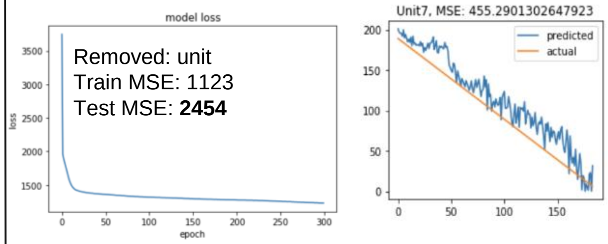
Removed: unit
Train MSE: 1276
Test MSE: **2007**



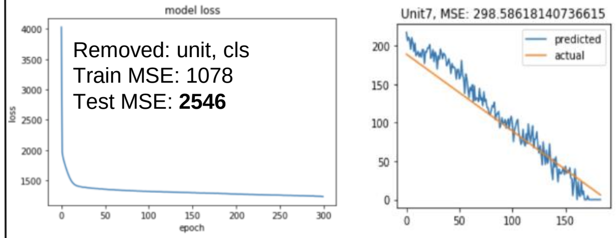
Removed: unit, cls
Train MSE: 1249
Test MSE: **2042**



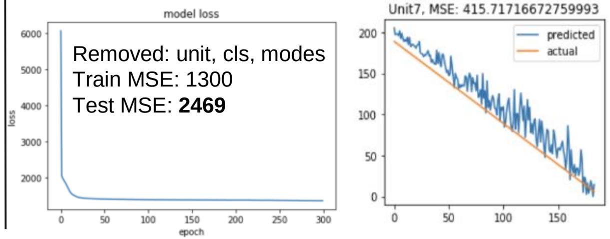
Removed: unit, cls, modes
Train MSE: 1382
Test MSE: **1934**



Removed: unit
Train MSE: 1123
Test MSE: **2454**

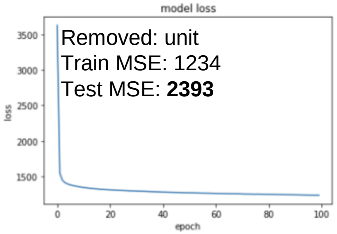


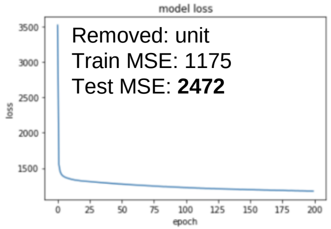
Removed: unit, cls
Train MSE: 1078
Test MSE: **2546**

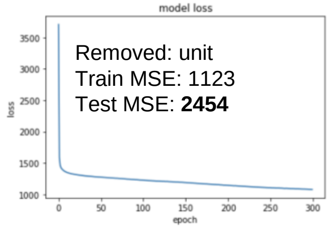


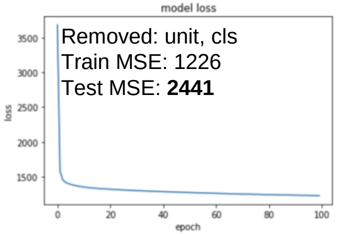
Removed: unit, cls, modes
Train MSE: 1300
Test MSE: **2469**

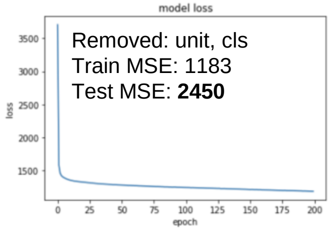
MLPs with normalized (cls) data

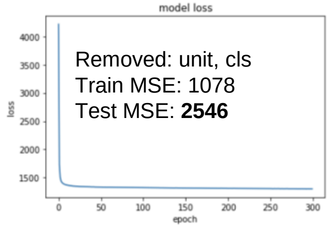


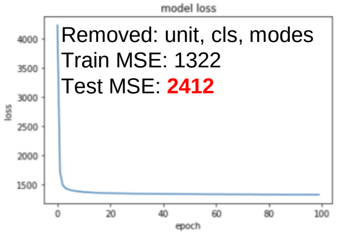


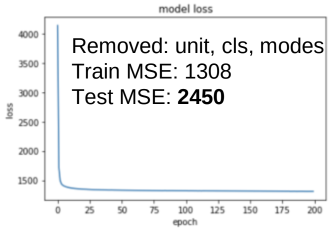


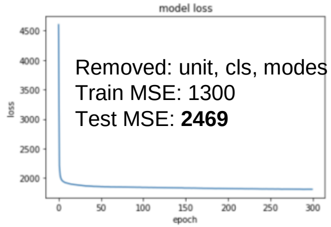




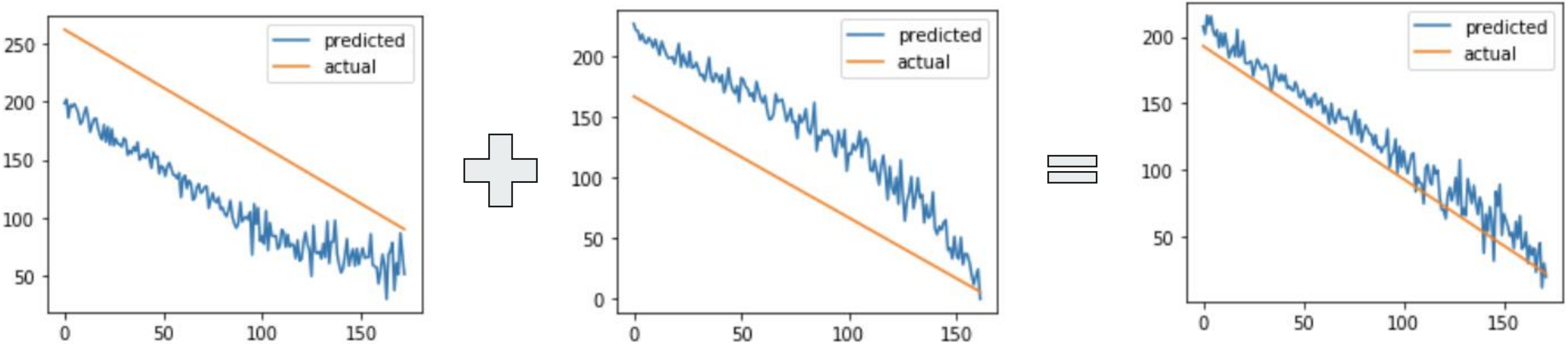




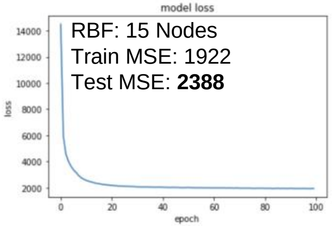


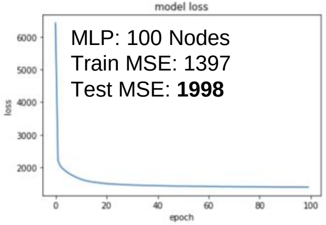


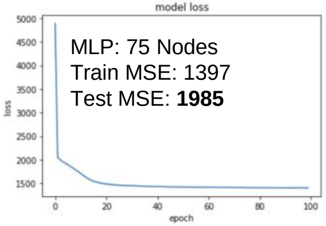
Combining models?



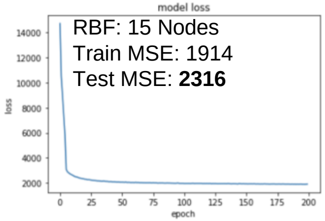
MLPs + RBF

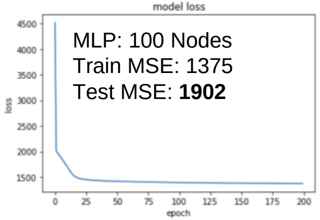


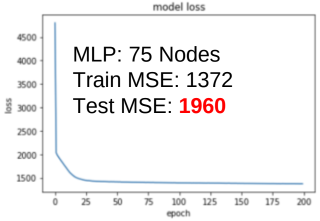




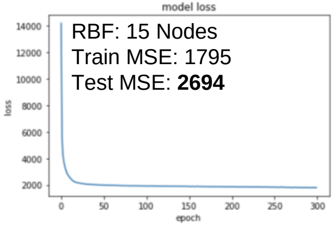
2040

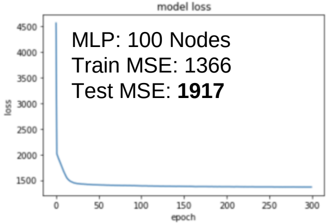


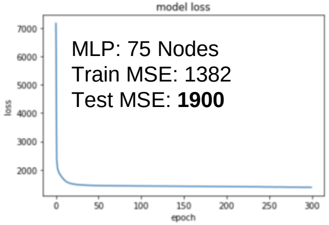




1972

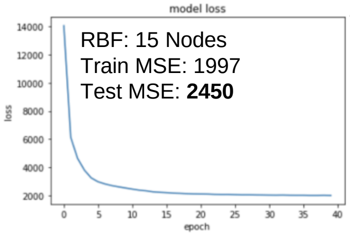


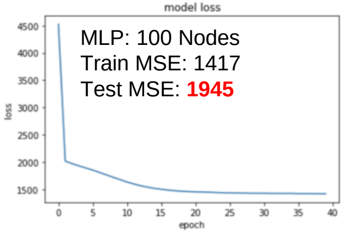


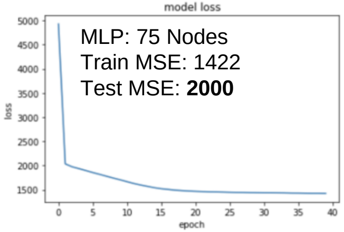


2055

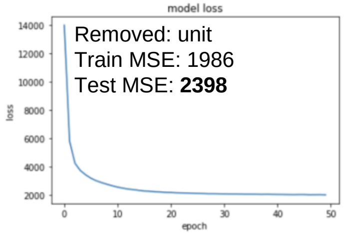
Fewer epochs?

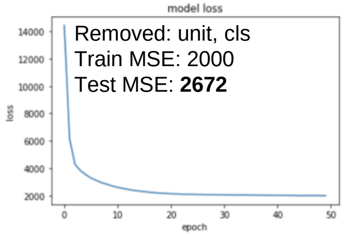


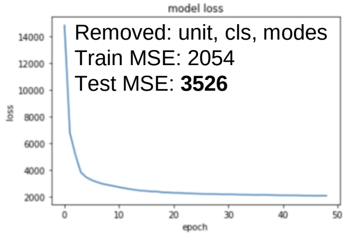




2036







2728

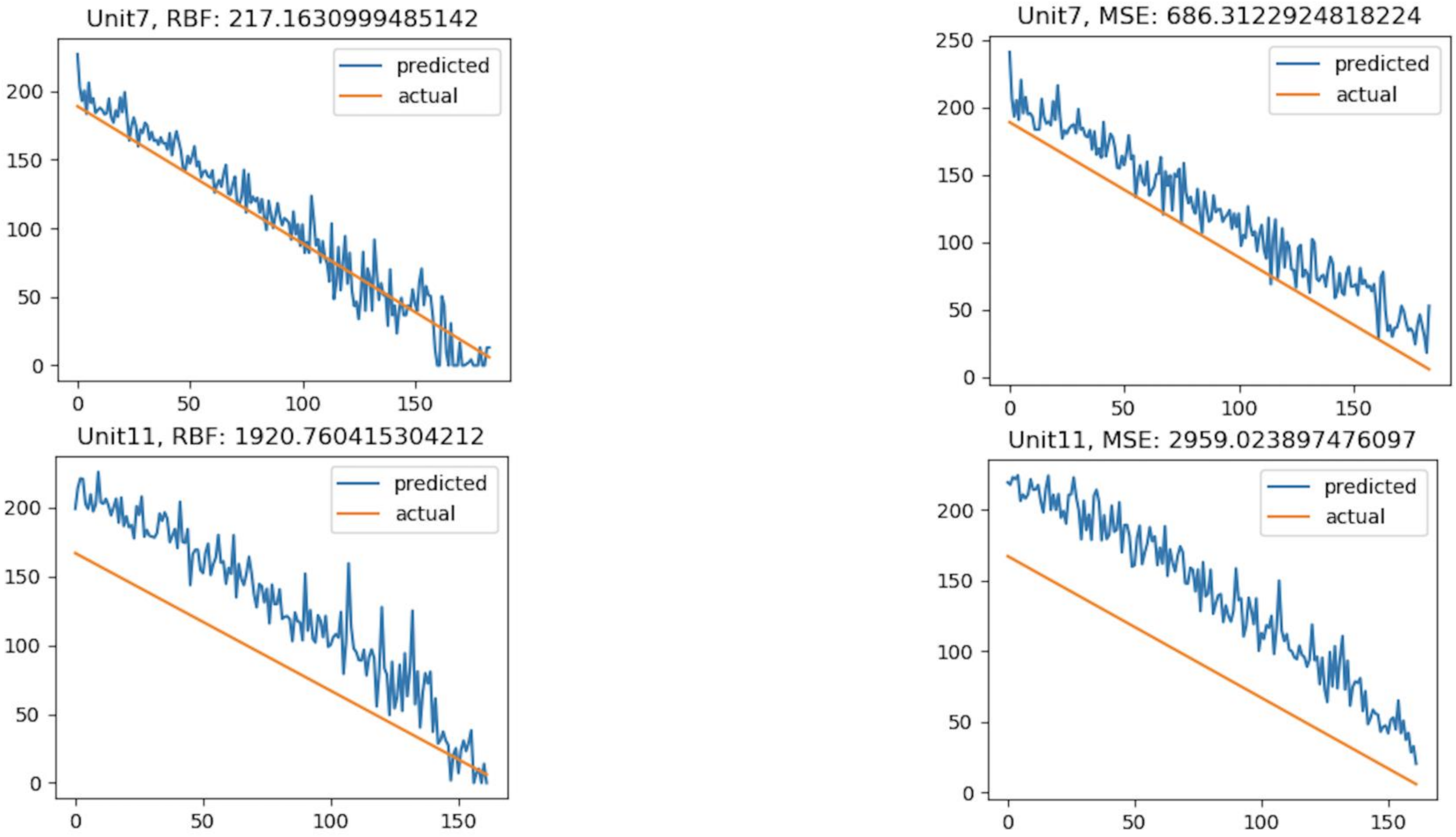
Experiments with Feature Selection

100 nodes, 100 epocs, 1 hidden layer

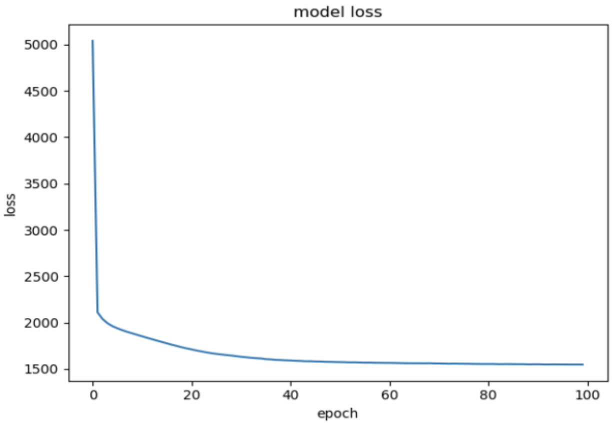
Selected sensors:

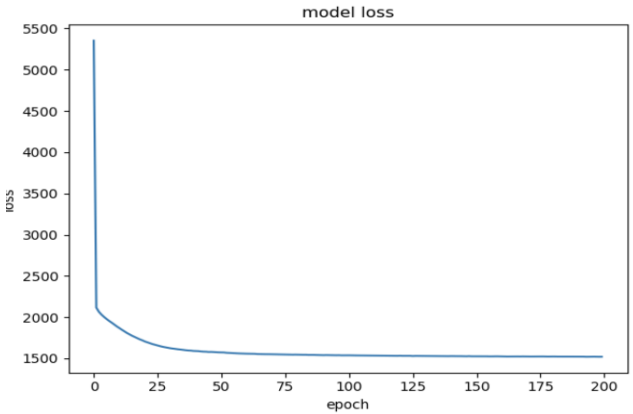
- chi squared: 3, 4, 7, 9, 12, 18
- PCA: 1, 6, 7, 8, 12, 18

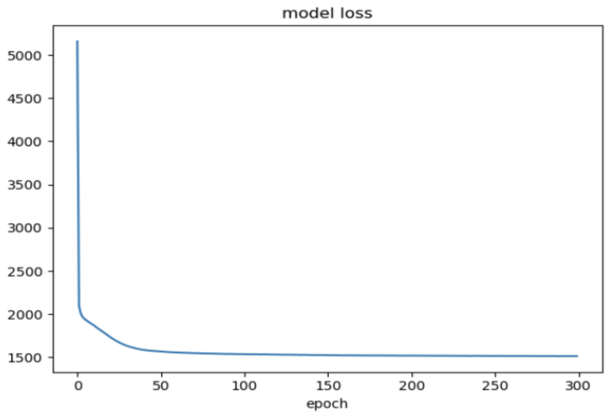
Chi-Squared		PCA	
MLP	**RBF**	**MLP**	**RBF**
2121	2107	2543	2538


Unit7, RBF: 217.1630999485142
predicted
actual
0
50
100
150
200
Unit7, MSE: 686.3122924818224
predicted
actual
250
Unit11, RBF: 1920.760415304212
predicted
actual
Unit11, MSE: 2959.023897476097
predicted
actual

- 100 epocs, chi squared selected sensors, other features on RBF
- 200 epocs, chi squared selected sensors, other features on RBF
- 300 epocs, chi squared selected sensors, other features on RBF







Best result: settings, cycle and chi squared selected sensors with 100 epocs

100 nodes, 100 epocs, 1 hidden layer

Chi-Squared		PCA		No Selection	
MLP	RBF	MLP	RBF	MLP	RBF
2121	2107	2543	2538	1977	1924

Conclusion & Future Work

- Look for other feature selection methods
- Experiment with other models

References

- Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity based prognostics approach for remaining useful life estimation of engineered systems. In Int. conf. on prognostics and health management (p. 1-6).
- Zhao, D., P., R. G., & Willett. (2011). Comparison of data reduction techniques based on SVM classifier and SVR performance. In Proc. spie, signal and data processing of small targets (Vol. 8137, p. 1-15).
- Ramasso, E. (2014a). Investigating computational geometry for failure prognostics. Int. Journal on Prognostics and Health Management. (submitted)
- Ramasso, E. and A. Saxena. *Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset*. in *Annual Conference of the Prognostics and Health Management Society 2014.*2014.

References

- Orr, M. J. (1996). Introduction to radial basis function networks.
- L. Peel, “Data driven prognostics using a Kalman filter ensemble of neural network models,” in Proc. Int. Conf. Prognost. Health Management, 2008, pp. 1–6.
- https://stackoverflow.com/questions/22984335/recovering-features-names-of-explained-variance-ratio-in-pca-with-sklearn
- https://www.kaggle.com/kanncaa1/feature-selection-and-data-visualization
- https://www.datacamp.com/community/tutorials/feature-selection-python
- https://www.mathsisfun.com/data/chi-square-test.html
- https://github.com/PetraVidnerova/rbf_keras
- https://www.datacamp.com/community/tutorials/deep-learning-python

References

Table 1. Description of the five turbofan degradation datasets available from NASA repository.

Datasets		#Fault Modes	#Conditions	#Train Units	#Test Units
Turbofan data from NASA repository	#1	1	1	100	100
	#2	1	6	260	259
	#3	2	1	100	100
	#4	2	6	249	248
PHM2008 Data Challenge	#5T	1	6	218	218
	#5V	1	6	218	435