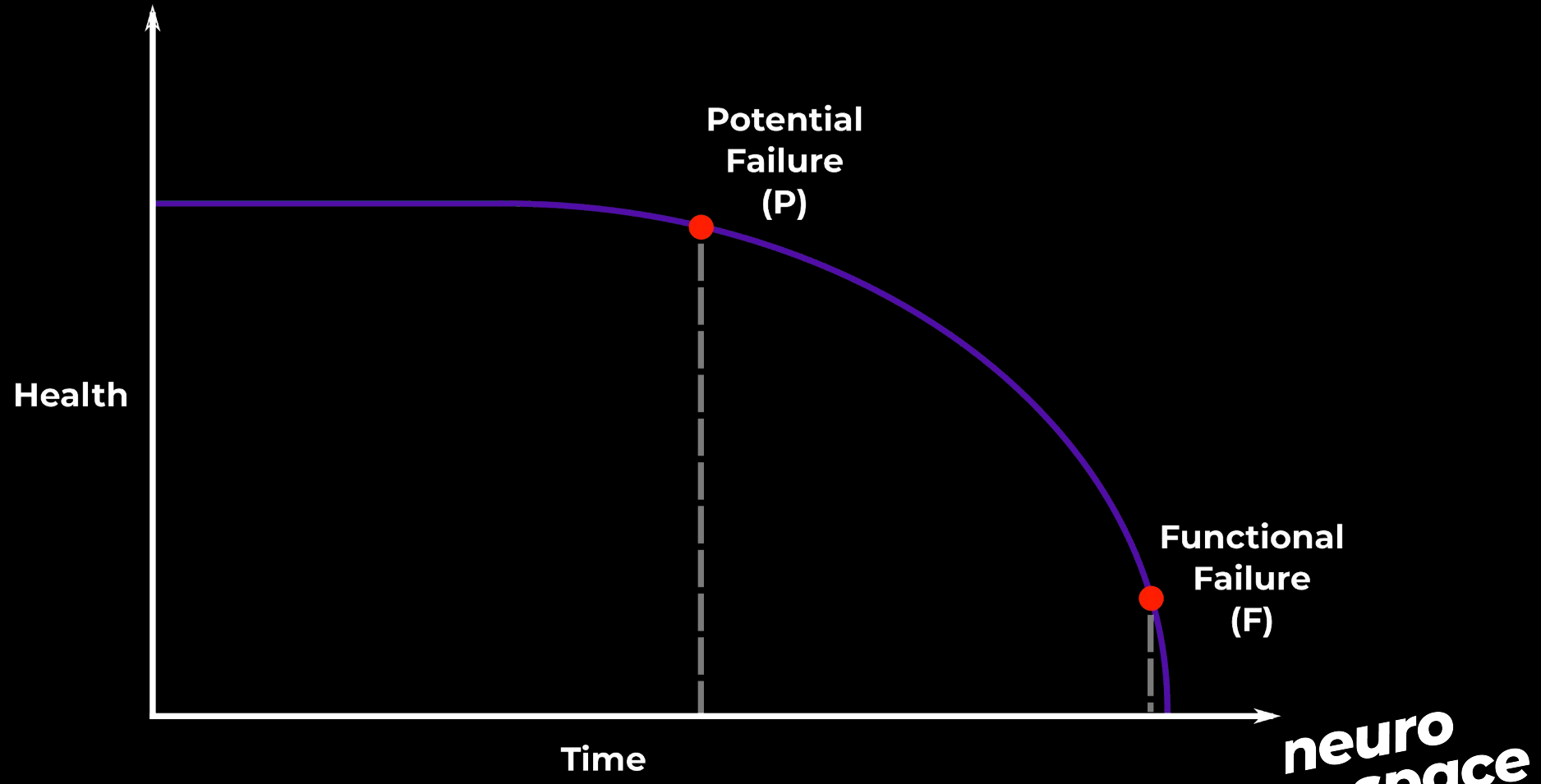
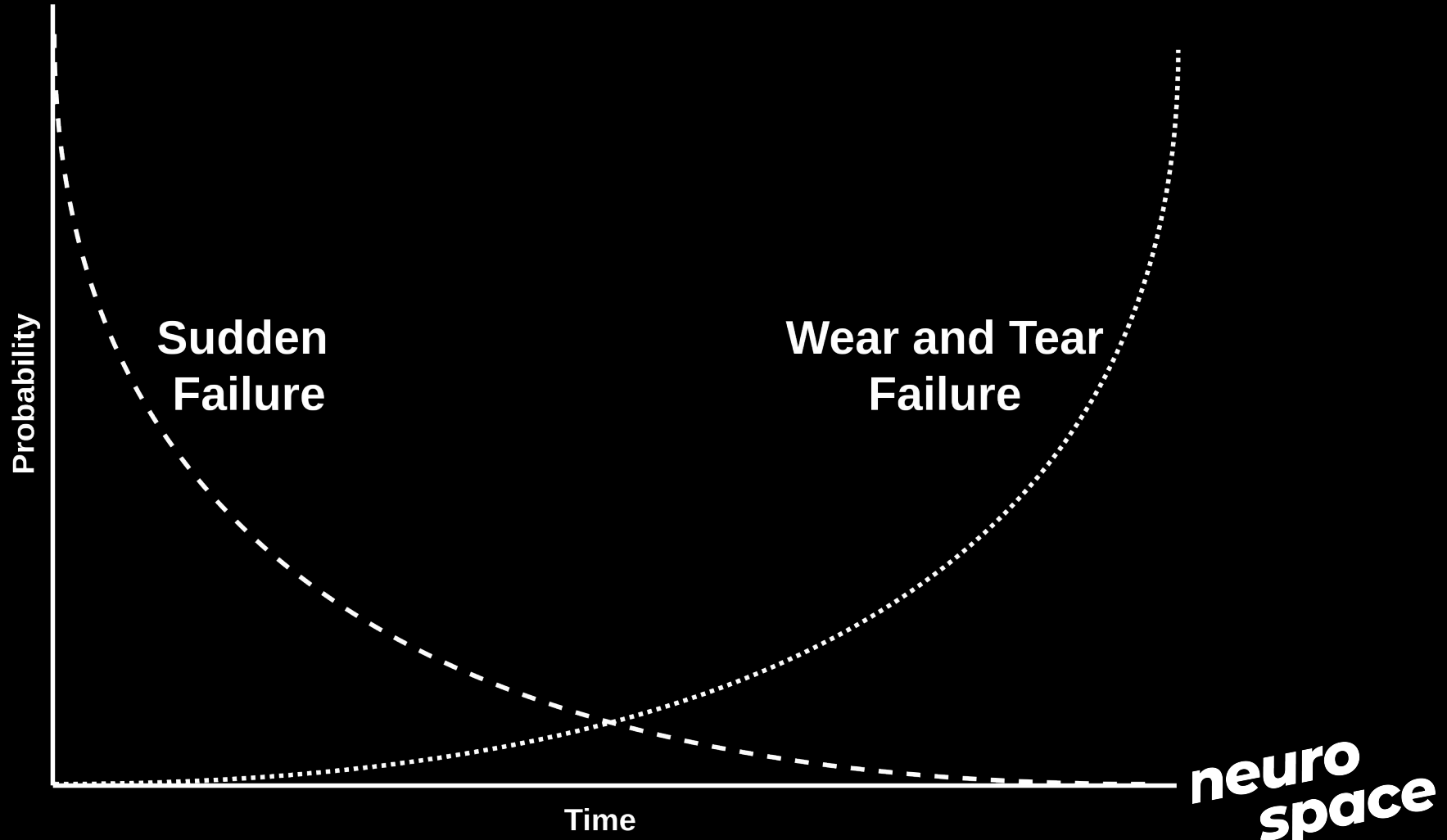
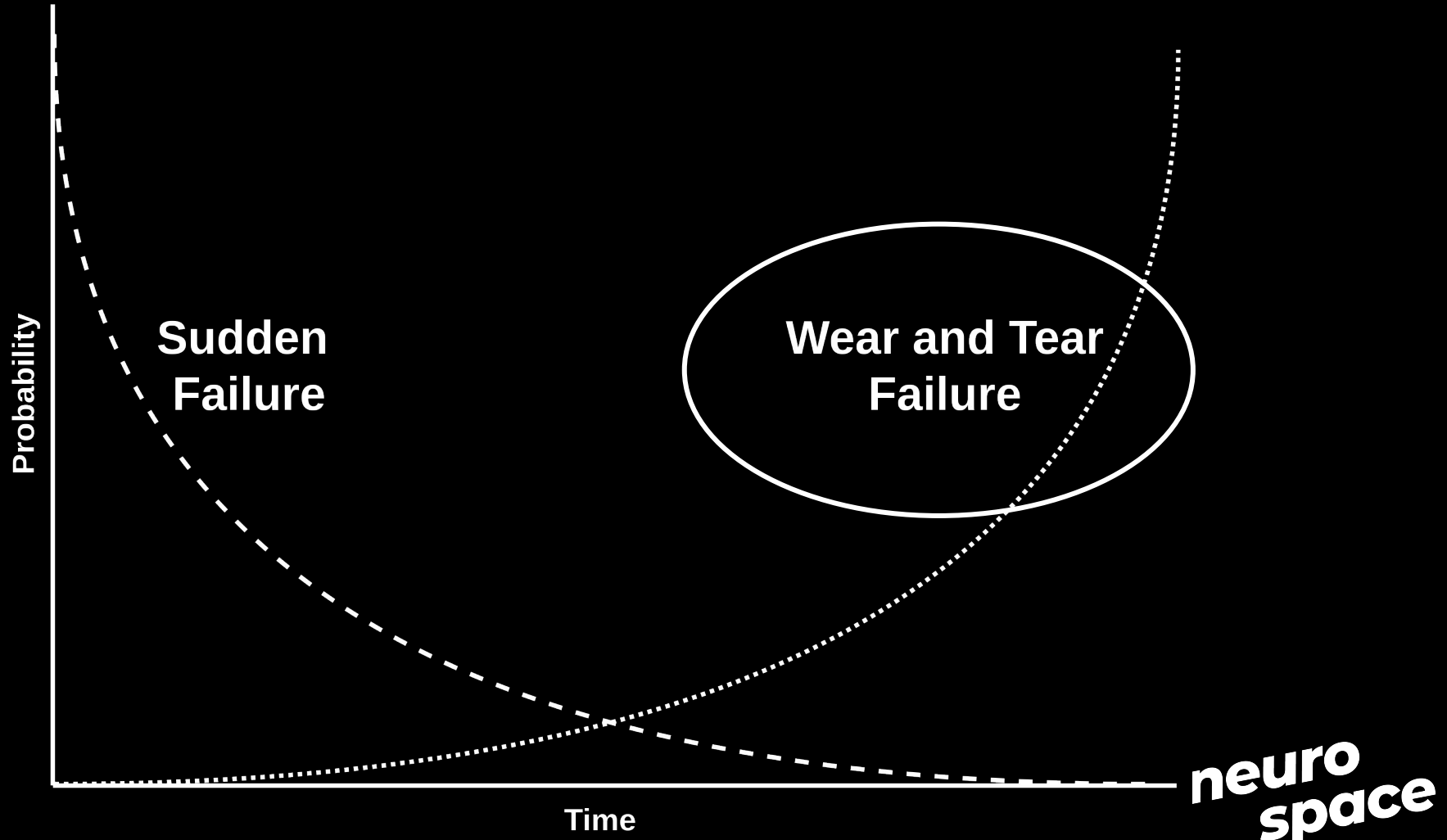
Predictive maintenance

neuro space


Potential
Failure
(P)
Health
Functional
Failure
(F)
Time
neuro
space


Probability
Sudden Failure
Wear and Tear Failure
Time
neuro space


Probability
Sudden Failure
Wear and Tear Failure
Time
neuro space

challenge

So far, little (no) evidence exist that this is possible for small datasets (<70)

We need to give value as fast as possible, thus we wish to create a solid model with only 7 breakdowns

neuro space

Predictive maintenance on a water pump

neuro space

introduction to the dataset

n	220,320	100%
n_normal	205,836	approx. 93.42%
n_recovering	14,477	approx. 6.57%
n_broken	7	approx. 0.003%

52 sensors

neuro space

introduction to the dataset

n	220,320	100%
n_normal	205,836	approx. 93.42%
~~**n_recovering**~~	~~14,477~~	~~approx. 6.57%~~
n_broken	7	approx. 0.003%

neuro space

time between failures (TBF)

1	12 days
2	5 days
3	31 days
4	6 days
5	35 days
6	9 days
7	17.5 days

neuro space

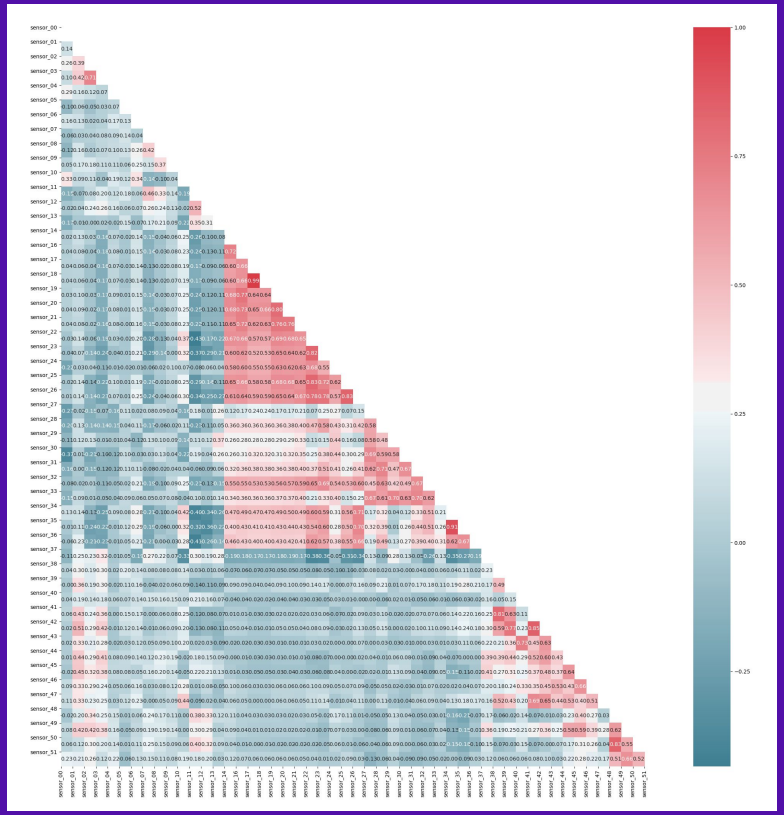
time between failures (TBF)

1	12 days
2	5 days
3	31 days
4	6 days
5	35 days
6	9 days
7	17.5 days

$\bar{X}$ = 16.5 days

neuro space

heatmap (correlation)



neuro space

approach

We use Python, Scikit-Learn, Talos, Pandas, Numpy, Tensorflow (Keras), Scipy, and Statsmodels

neuro space

approach

In all trials, the data is preprocessed the same way:

1. Split in train (3.5 breakdowns), validation (0.5 breakdown), and test (2 breakdowns)
2. Scaled by the help of StandardScaler
3. Noise reduced by the help of Principal Component Analysis, n_components=10
4. Those models that were hypertuned, has the option to use MSE or MAE

neuro space

Testing four different algorithms

neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	0.05777	0.0669	0.0497
Random Forest Regression	5.3386e-05	0.0430	0.0431

neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	**0.05777**	**0.0669**	**0.0497**
Random Forest Regression	5.3386e-05	0.0430	0.0431

neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	**0.0715973**	**0.041526**	**0.1066**
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	0.05777	0.0669	0.0497
Random Forest Regression	5.3386e-05	0.0430	0.0431

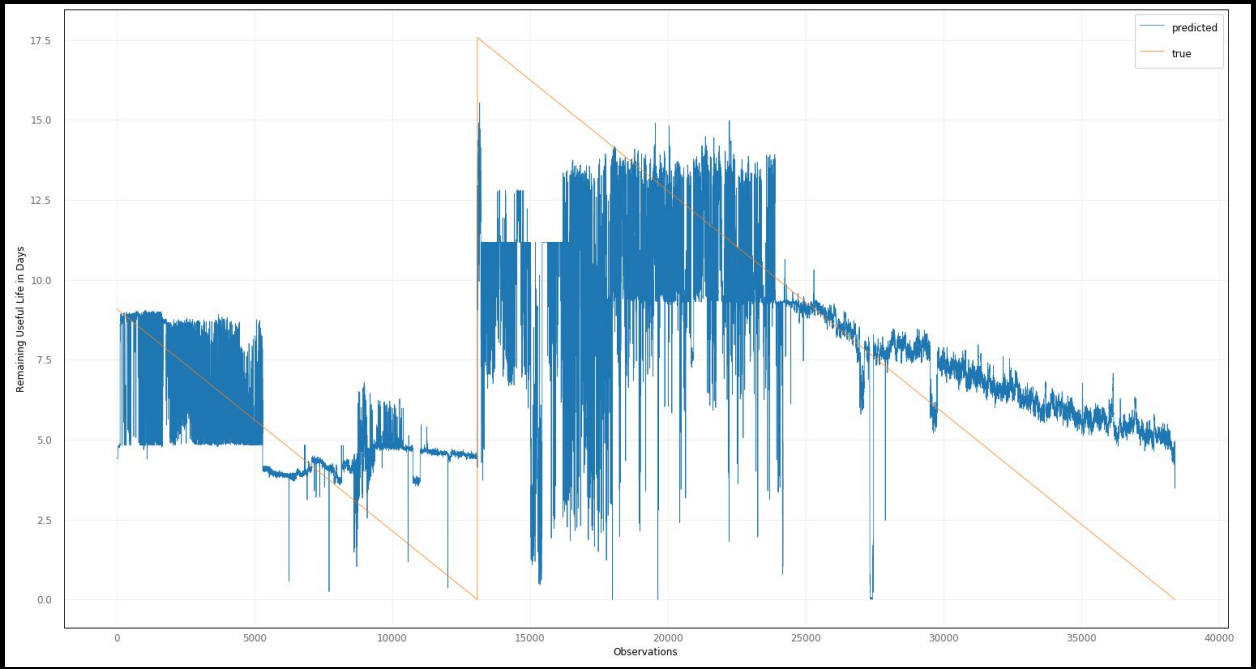
neuro space

neural network with Talos

```
103  p = {'activation1':[relu, elu, tanh, sigmoid],
104       'activation2':[relu, elu, tanh, sigmoid],
105       'activation3': [sigmoid],
106       'optimizer': ['Adam', "RMSprop"],
107       'losses': ['mse', 'mae'],
108       'first_hidden_layer': [12, 10, 8, 6],
109       'second_hidden_layer': [2, 4, 6],
110       'batch_size': [100, 1000],
111       'epochs': [10, 15, 25, 50]}
112
```

neuro space

neural network with Talos



neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	**1.871**	**N.C**	**0.0422**
Long-Short-Term-Memory	0.05777	0.0669	0.0497
Random Forest Regression	5.3386e-05	0.0430	0.0431

neuro space

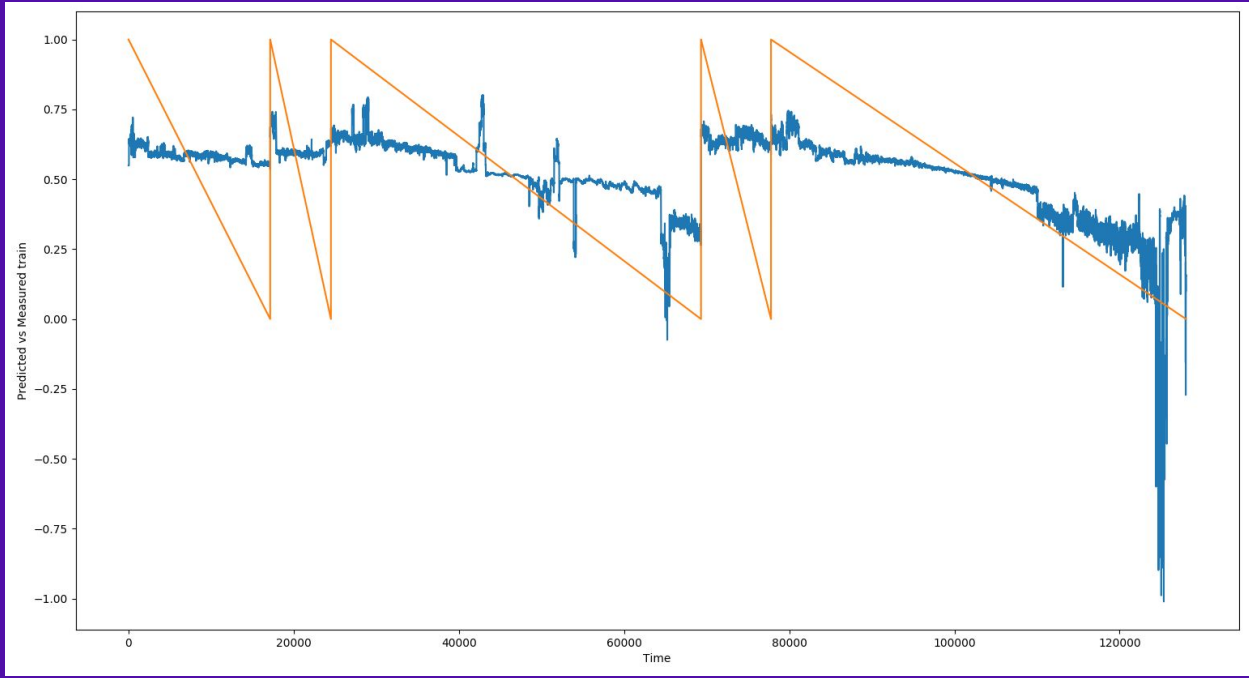
support vector regression

GridSearchCV for hypertune optimization

According to Scikit-learn SVR is hard to scale when dataset is large (n>10,000)

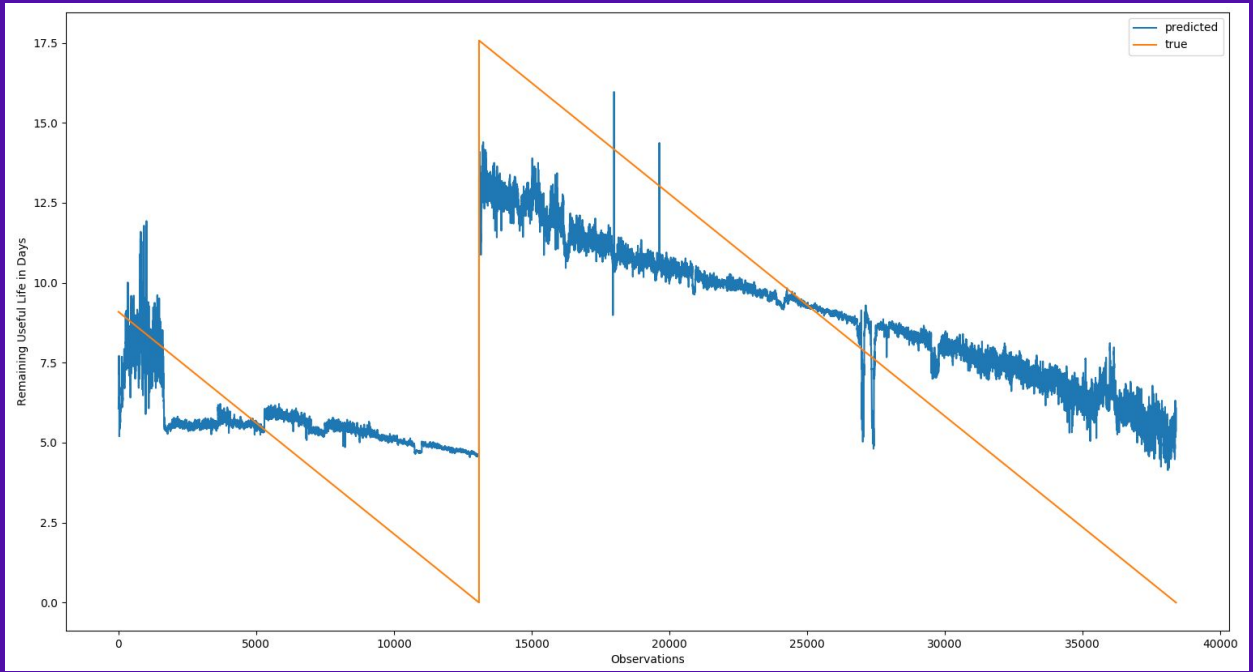
neuro space

support vector regression (train+validation)



neuro space

support vector regression (test)



neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	**0.05777**	**0.0669**	**0.0497**
Random Forest Regression	5.3386e-05	0.0430	0.0431

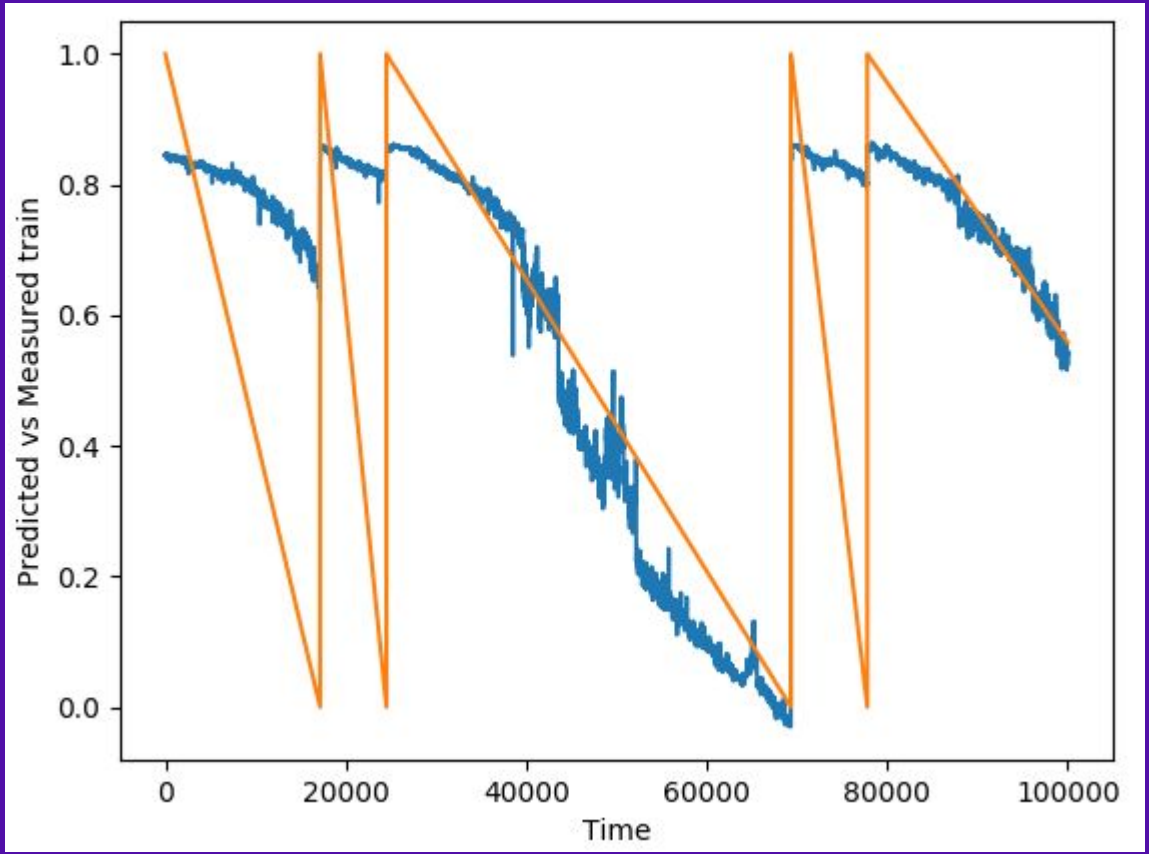
neuro space

long-short-term-memory

We used basic optimization principles of trial and error

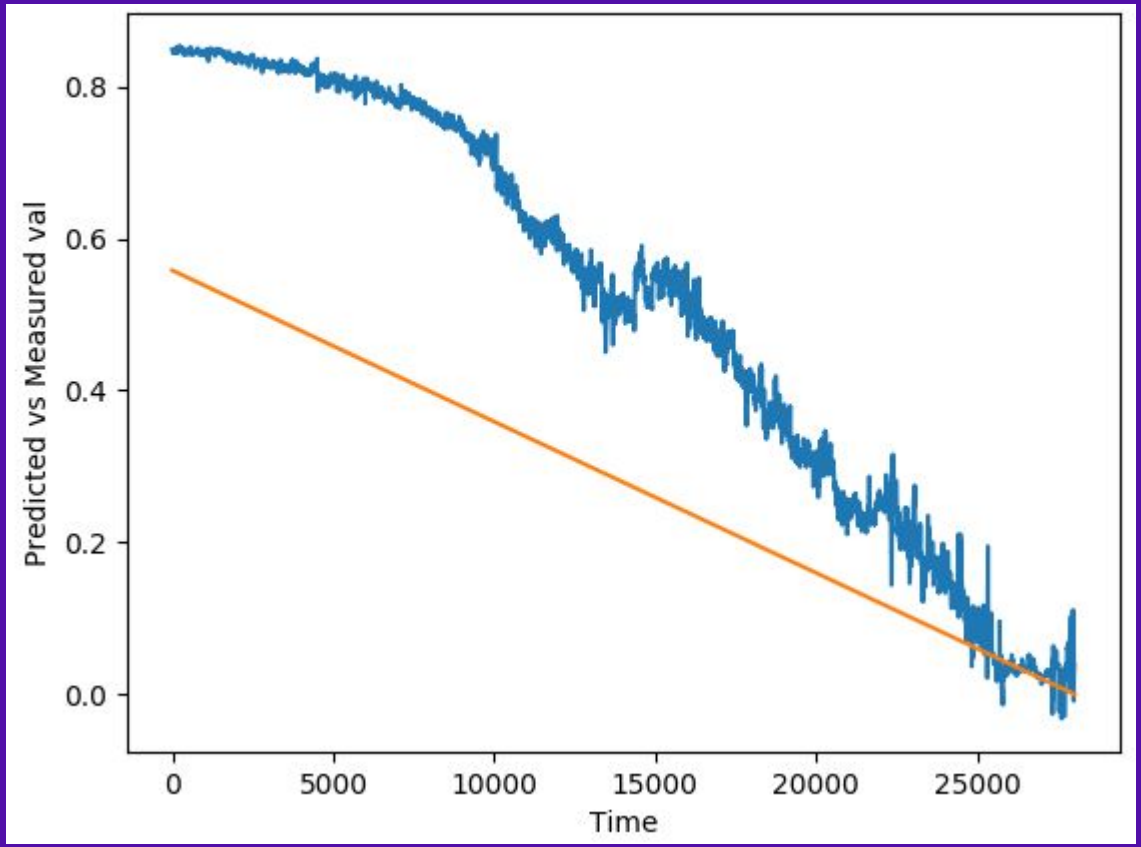
neuro space

long-short-term-memory (train)



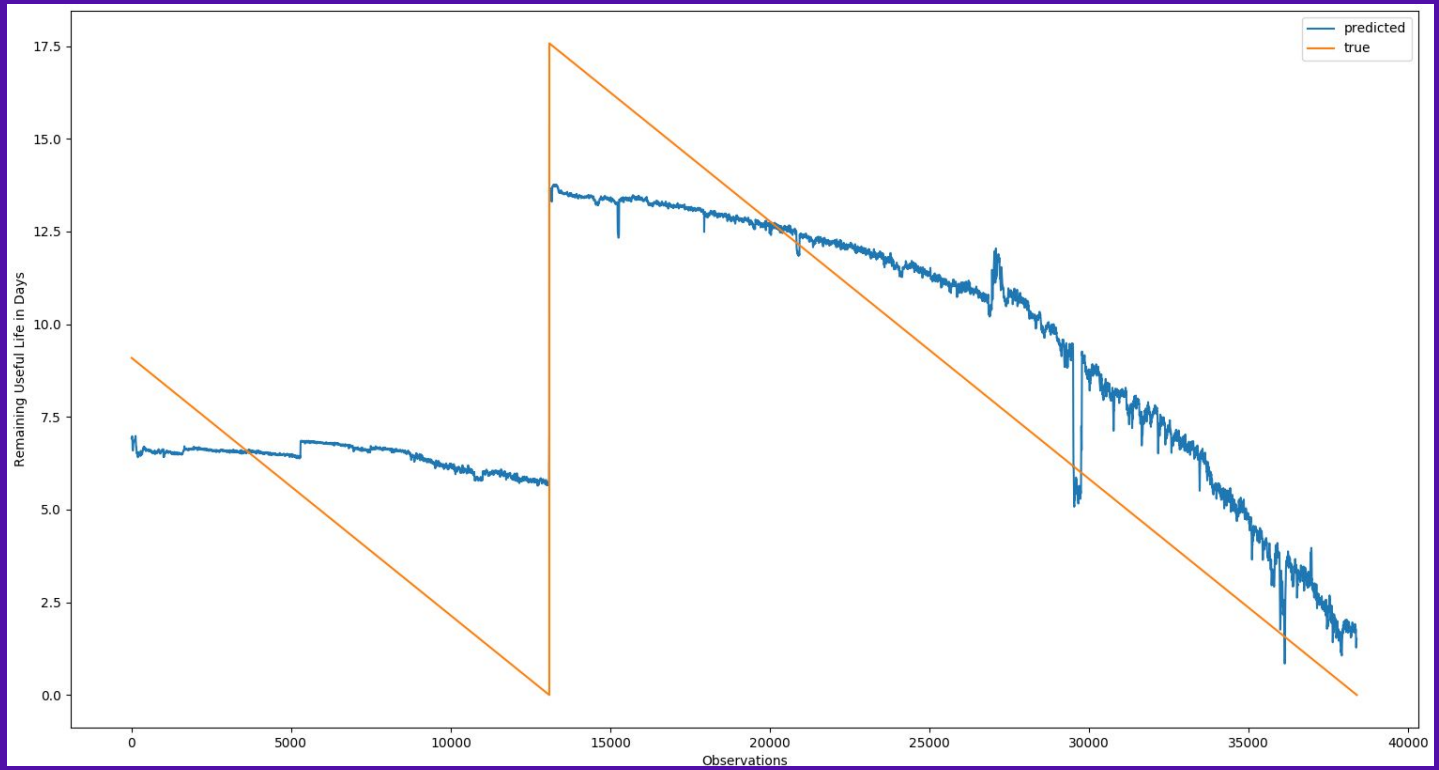
neuro space

long-short-term-memory (validation)



neuro space

long-short-term-memory (test)



neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	0.05777	0.0669	0.0497
Random Forest Regression	5.3386e-05	0.0430	0.0431

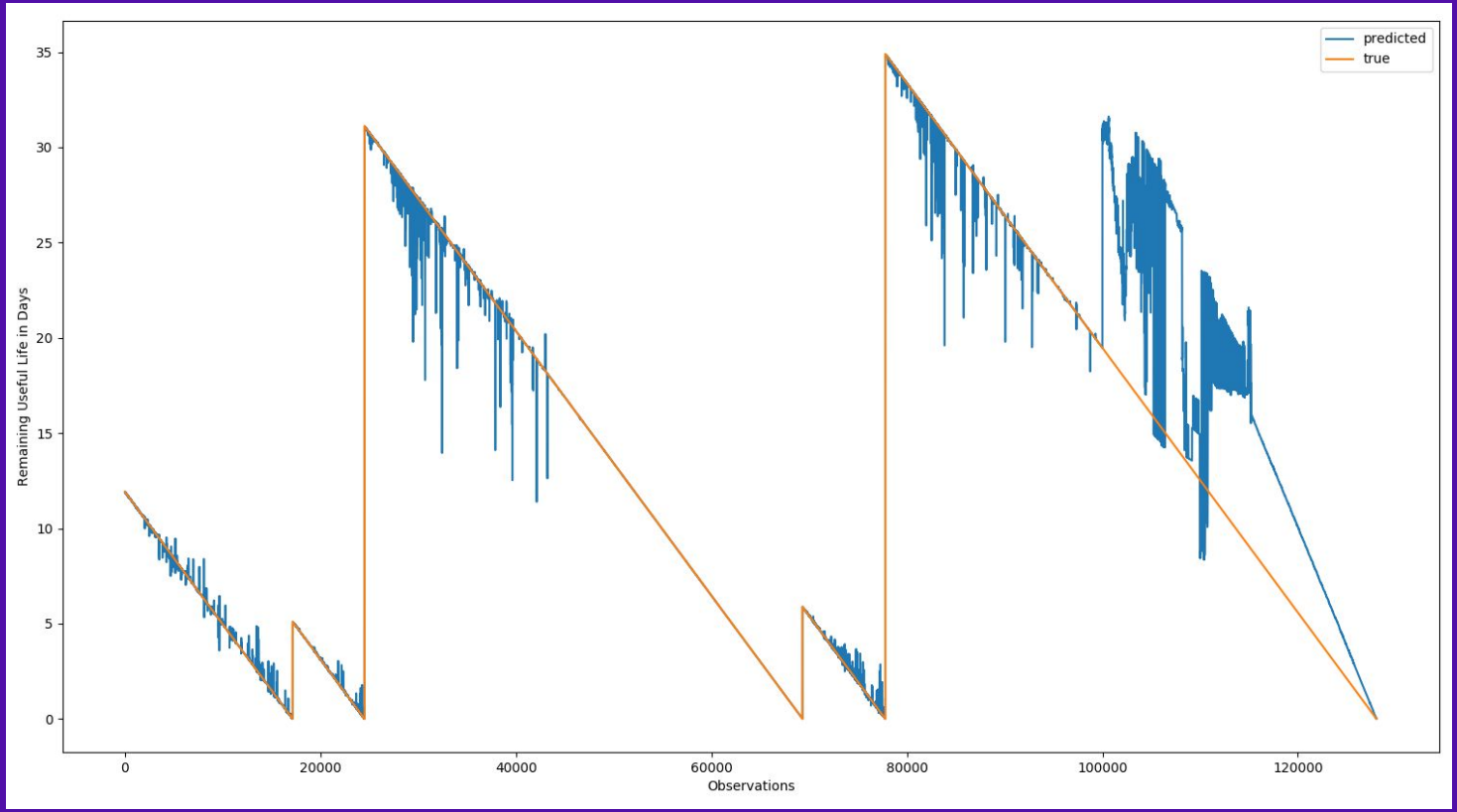
neuro space

random forest regression

We used GridSearchCV for parameter Hypertuning

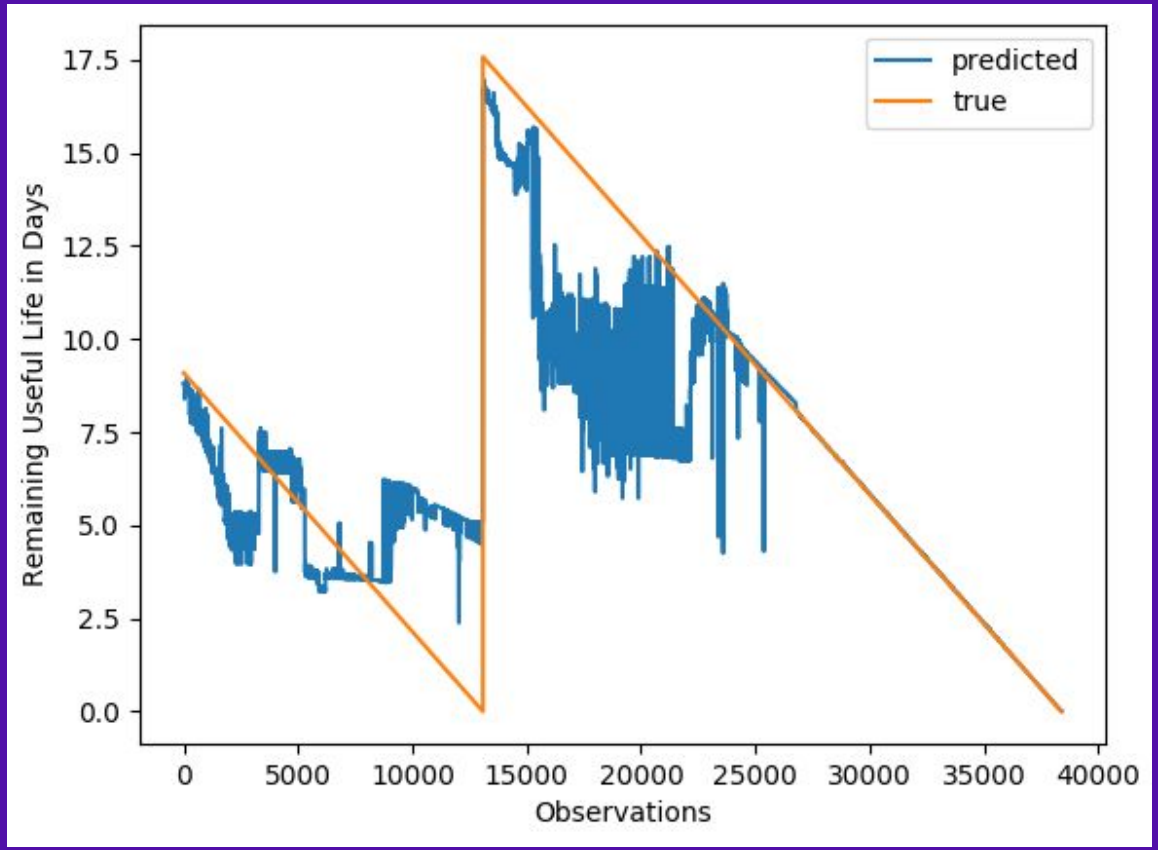
neuro space

random forest regression (train and validation)



neuro space

random forest regression (test)



neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	0.05777	0.0669	0.0497
Random Forest Regression	5.3386e-05	0.0430	0.0431

neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	**0.05777**	**0.0669**	**0.0497**
Random Forest Regression	5.3386e-05	0.0430	0.0431

neuro space

summary

Machine learning model	Train loss	Validation loss	Test loss
Dense Neural Network	0.0715973	0.041526	0.1066
Support Vector Regression	1.871	N.C	0.0422
Long-Short-Term-Memory	0.05777	0.0669	0.0497
Random Forest Regression	**5.3386e-05**	**0.0430**	**0.0431**

neuro space