Krafthack pitch

Martin Tveten & Per August Moen

08/03/2022

Where to find the code

Our code is written into an R package which can be installed as such: You can install Krafthack2022 from github with:

```
# install.packages("devtools")
devtools::install_github("Tveten/Krafthack2022")
```

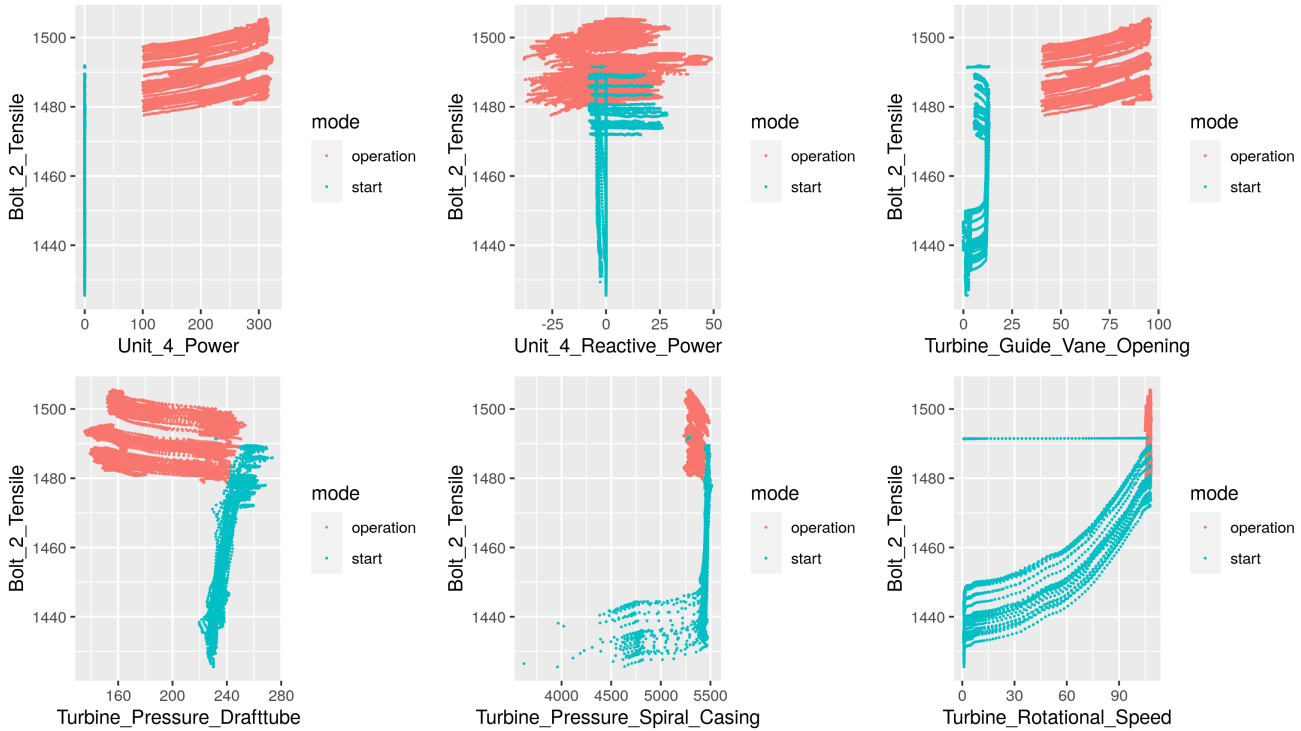
The code can also be found here.

The code that produces the predictions is found in ../inst/data_prediction.R.

Overview

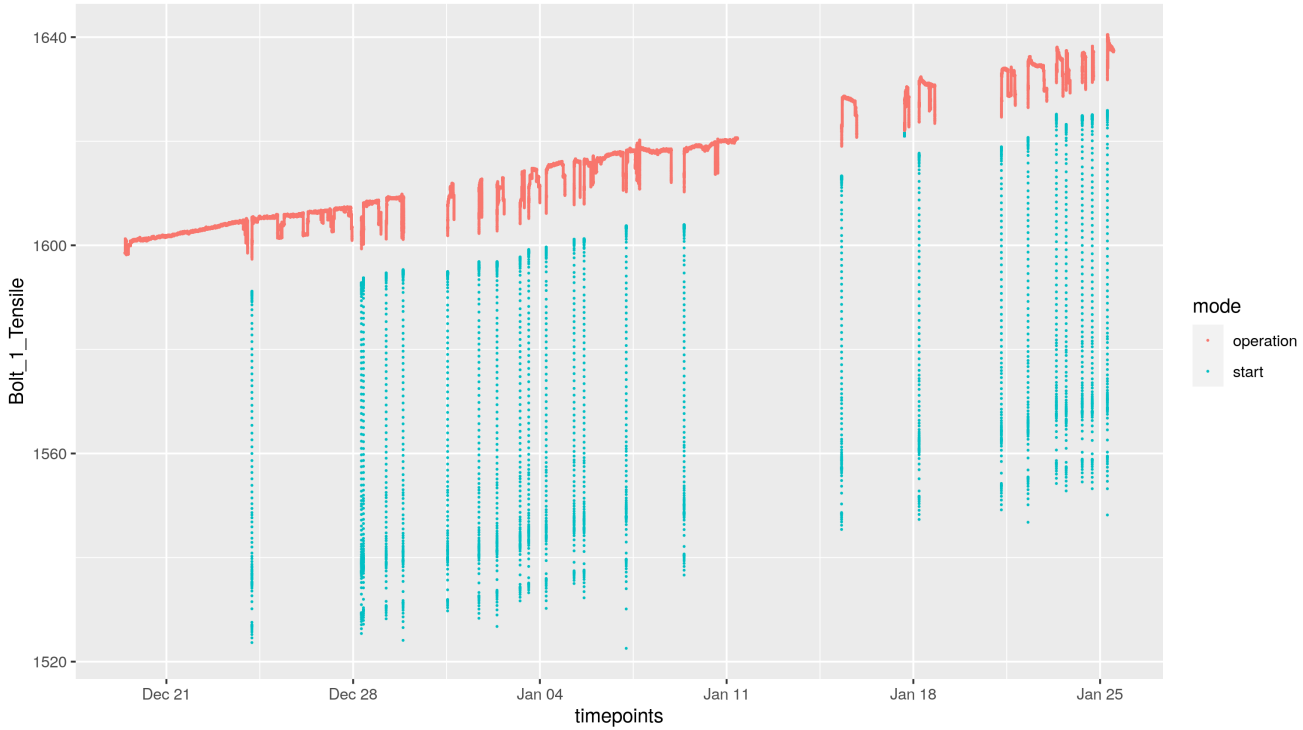
We have fitted separate a linear (ridge regression) model to each bolt tensil using all possible interactions of the features. Our motivation for this choice of model is two-fold: firstly, we are curious as to how well such a simple model can perform; secondly, we find many linear relationships in the data, suggesting that a linear model can be a reasonable assumption.

Data findings – highly correlated covariates



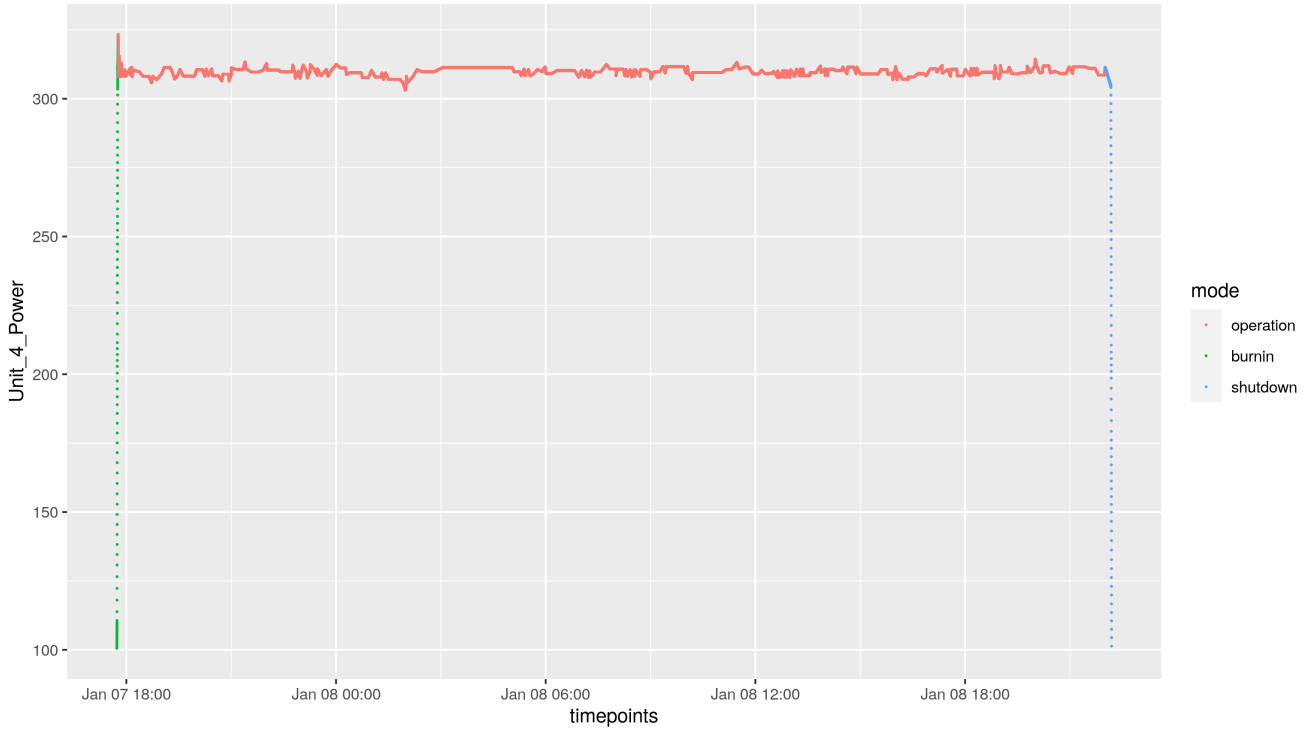
We find that most, if not all, of the features are highly correlated. For instance, the feature “Reactive power” is essentially just noise, and we choose to remove it from the model.

Data findings – tensile trend

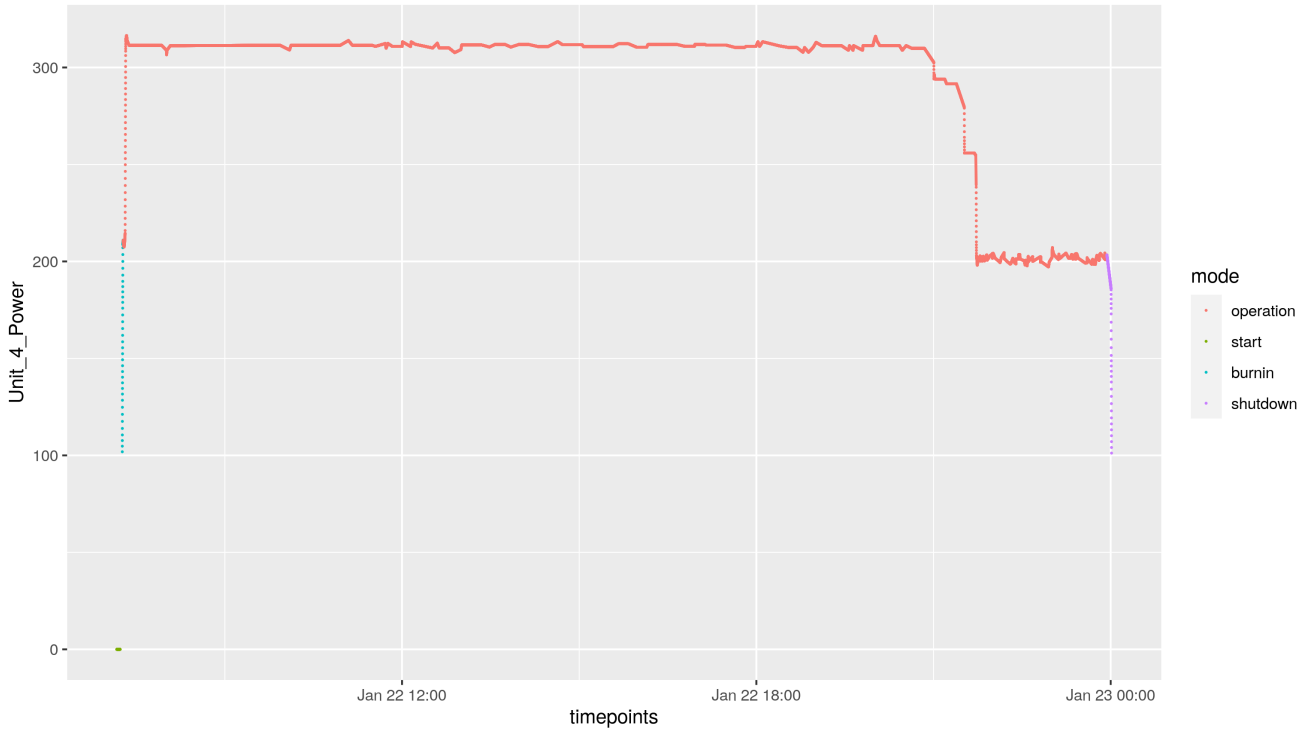


We find that there is a clear linear relationship between time and bolt tensiles. This suggests that a linear model can, although simple, be a reasonable model choice.

Data findings – Bump in some tensils after burnin



Augmentations – burn-in and shutdown



We have chosen to augment the data set by introducing two new modes: “burnin” and “shutdown”.

Augmentations – bump feature

In addition, we have introduced a new feature f given by

$$f = \frac{1}{t^{1/4}},$$

where t is the time since the last burnin period ended. f is zero whenever the mode is not “operational”.

Model considerations

We have used a simple linear ridge regression model for each bolt separately, with features being all possible interactions between the following variables:

- timepoints
- Unit_4_Power
- Turbine_Guide_Vane_Opening
- Turbine_Pressure_Drafttube
- Turbine_Pressure_Spiral_Casing
- Turbine_Rotational_Speed
- mode = (“start”, “burnin”, “operation”, “shutdown”)

In addition we have included the previously defined feature f and its interactions with the remaining features.

We use the “glmnet” package train the model.

Model considerations

The (ridge) linear model is simple, with its only flexibility coming from the vast number of interactions.

- Crossplots reveal that many relationships are linear or near-linear, and adding transformations of features has not improved prediction.
- The ridge penalty reduces risk of overfitting. Chosen by cross-validation.
- We

Performance testing

The data are clearly non-stationary. Validation regime:

- Splitting into consecutive blocks of alternating training and validation sets.
- Size of blocks: 200 observations.
- I.e.: 50% training and 50% validation with an approximately uniform distribution.
- Measure mean percentage error on validation data.

Putting it into production

- Predicting new data points is very fast (linear combination).
- Simple and fast training.
  - 10-15 minutes in single CPU core.
  - Easy to parallelise.
- Easy to understand and learn from the model.
  - Can be verified against system experts.

Work flow for anomaly detection

1. Get sensor readings at time t.
2. Predict tensile value at time t.
3. Apply unsupervised anomaly detection method on prediction - observation.

Things we would have done if we had more time

- Smooth the predicted values, as they are quite noisy.
- Fit the ridge regression/linear regression using robust models.
- Randomise the performance testing.
- Hyper-parameter tuning based on cross-validation with MPE as performance metric.
- Perform anomaly detection by comparing the predicted values versus the observed values. There are several methods for this, such as CAPA-CC, OCD and Inspect.

Scalability and transferability

- The current model is quite general and can be used in many domains.
  - Learn normal behaviour on training data.
  - Look for anomalies in residuals.
- Even more so if:
  - Re-training as new data comes in.
  - Outlier-robust regression model/loss function. As anomalies can enter the data without being given too much weight in the normal behaviour model.