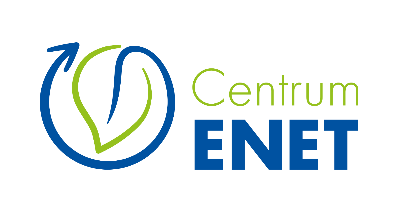


VSB – ENET Centre

Power Line Fault Detection

Using Machine Learning Classification Models to Detect Partial Discharge Faults in Covered Conductors in the Real Environment

Prepared For: Technical University of Ostrava - ENET Centre



15 March 2019 // Revision 1.0

Jeffrey A. Egan

chief engineer @ dragonaur.io



Contents

[Problem Overview 3](#_Toc535766020)

[Data Exploration 3](#_Toc535766021)

[Signal Processing 4](#_Toc535766022)

[Discrete Wavelet Transforms 4](#_Toc535766023)

[Removing Sinusoidal Trend Data 4](#_Toc535766024)

[Extraction of Signal Features 4](#_Toc535766025)

[Feature Matrix 4](#_Toc535766026)

[Prototype and Survey Classification Models 4](#_Toc535766027)

[k-Nearest Neighbors 4](#_Toc535766028)

[Support Vector Machine 4](#_Toc535766029)

[Random Forest 5](#_Toc535766030)

[Final Model Selection and Refinement 5](#_Toc535766031)

[Conclusions 5](#_Toc535766032)

# Problem Overview

Medium voltage overhead power lines run for hundreds of miles to supply power to cities. These great distances make it expensive to manually inspect the lines for damage that doesn't immediately lead to a power outage, such as a tree branch hitting the line or a flaw in the insulator. These modes of damage lead to a phenomenon known as partial discharge — an electrical discharge which does not bridge the electrodes between an insulation system completely. Partial discharges slowly damage the power line, so left unrepaired they will eventually lead to a power outage or start a fire.

Your challenge is to detect partial discharge patterns in signals acquired from these power lines with a new meter designed at the [ENET Centre](http://cenet.vsb.cz/en/) at [VŠB](https://www.vsb.cz/en/university/who-we-are/). Effective classifiers using this data will make it possible to continuously monitor power lines for faults.

ENET Centre researches and develops renewable energy resources with the goal of reducing or eliminating harmful environmental impacts. Their efforts focus on developing technology solutions around transportation and processing of energy raw materials.

By developing a solution to detect partial discharge you’ll help reduce maintenance costs, and prevent power outages.

Submissions are evaluated on the [Matthews correlation coefficient](https://en.wikipedia.org/wiki/Matthews_correlation_coefficient) (MCC) between the predicted and the observed response. The MCC is given by:

where *TP* is the number of true positives, *TN* the number of true negatives, *FP* the number of false positives, and *FN* the number of false negatives.

# Data Exploration

Faults in electric transmission lines can lead to a destructive phenomenon called [partial discharge](https://en.wikipedia.org/wiki/Partial_discharge). If left alone, partial discharges can damage equipment to the point that it stops functioning entirely. Your challenge is to detect partial discharges so that repairs can be made before any lasting harm occurs.

Each signal contains 800,000 measurements of a power line's voltage, taken over 20 milliseconds. As the underlying electric grid operates at 50 Hz, this means each signal covers a single complete grid cycle. The grid itself operates on a [3-phase power scheme](https://en.wikipedia.org/wiki/Three-phase_electric_power), and all three phases are measured simultaneously.

Thus a sample rate of 40 Msps (Million samples per second) for the data provided in the competition.

**metadata\_[train/test].csv**

id\_measurement: the ID code for a trio of signals recorded at the same time.

signal\_id: the foreign key for the signal data. Each signal ID is unique across both train and test, so the first ID in train is '0' but the first ID in test is '8712'.

phase: the phase ID code within the signal trio. The phases may or may not all be impacted by a fault on the line.

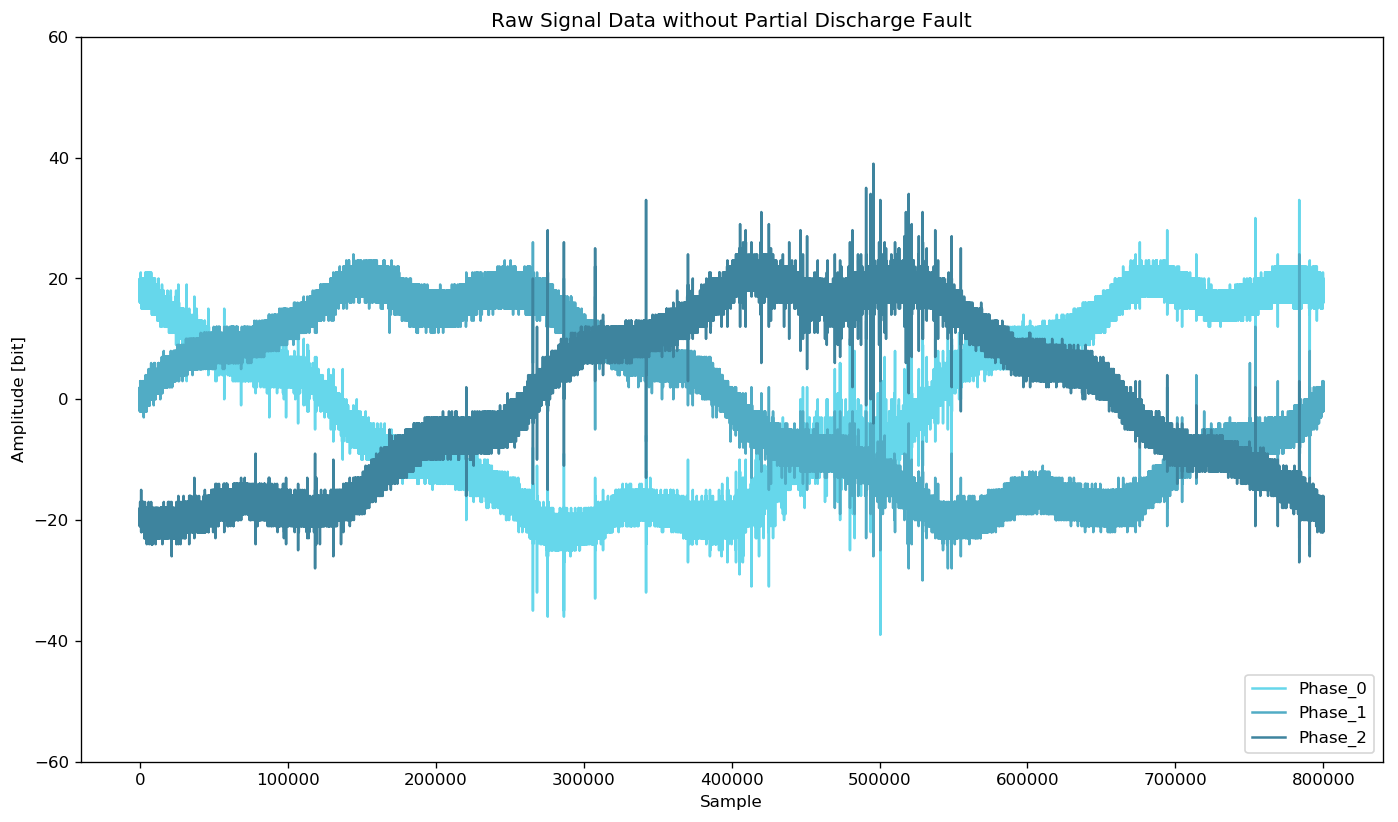
target: 0 if the power line is undamaged, 1 if there is a fault.

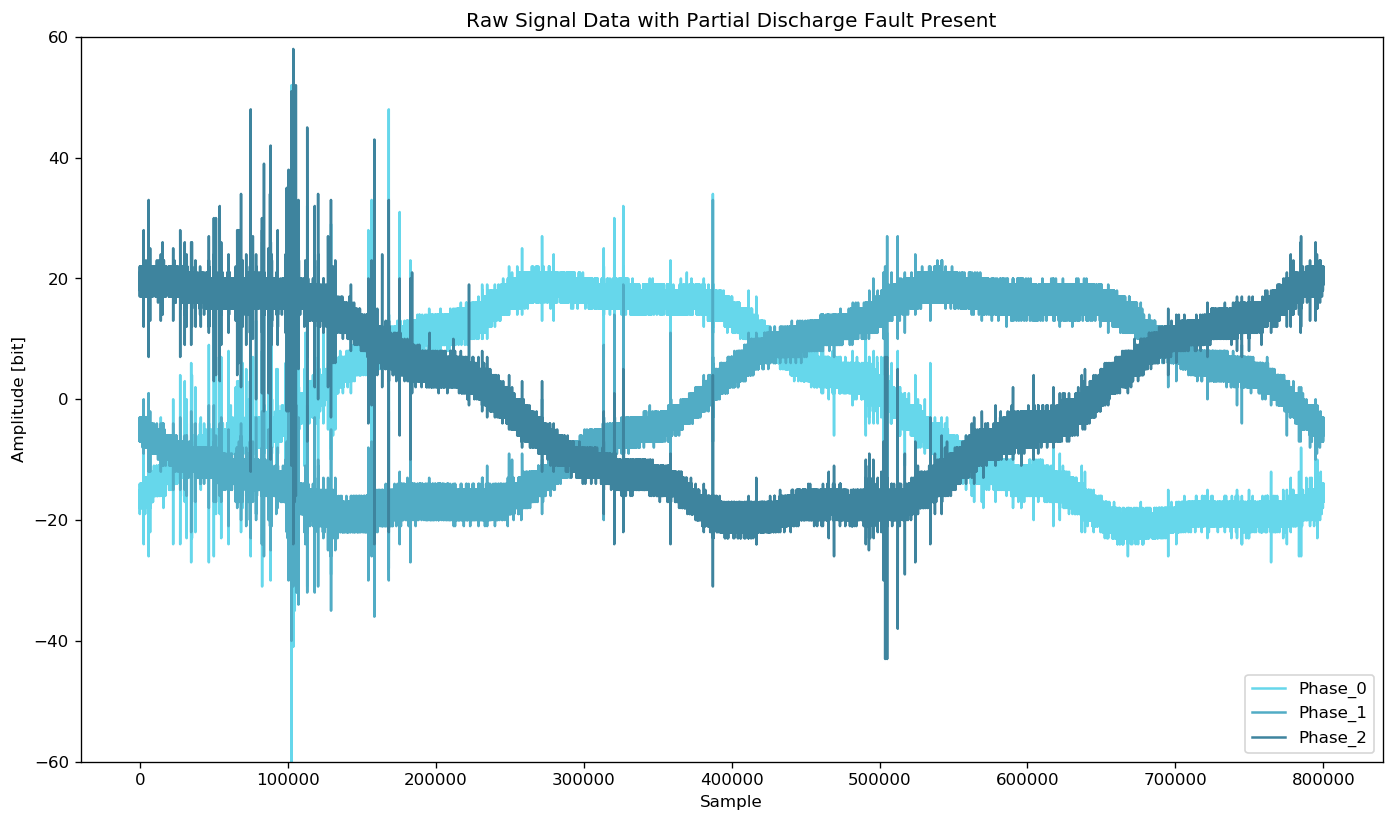
**[train/test].parquet** - The signal data. Each column contains one signal; 800,000 int8 measurements as exported with pyarrow.parquet version 0.11. Please note that this is different than our usual data orientation of one row per observation; the switch makes it possible loading a subset of the signals efficiently. If you haven't worked with [Apache Parquet](https://parquet.apache.org/) before, please refer to either [the Python data loading starter kernel](https://www.kaggle.com/sohier/reading-the-data-with-python).

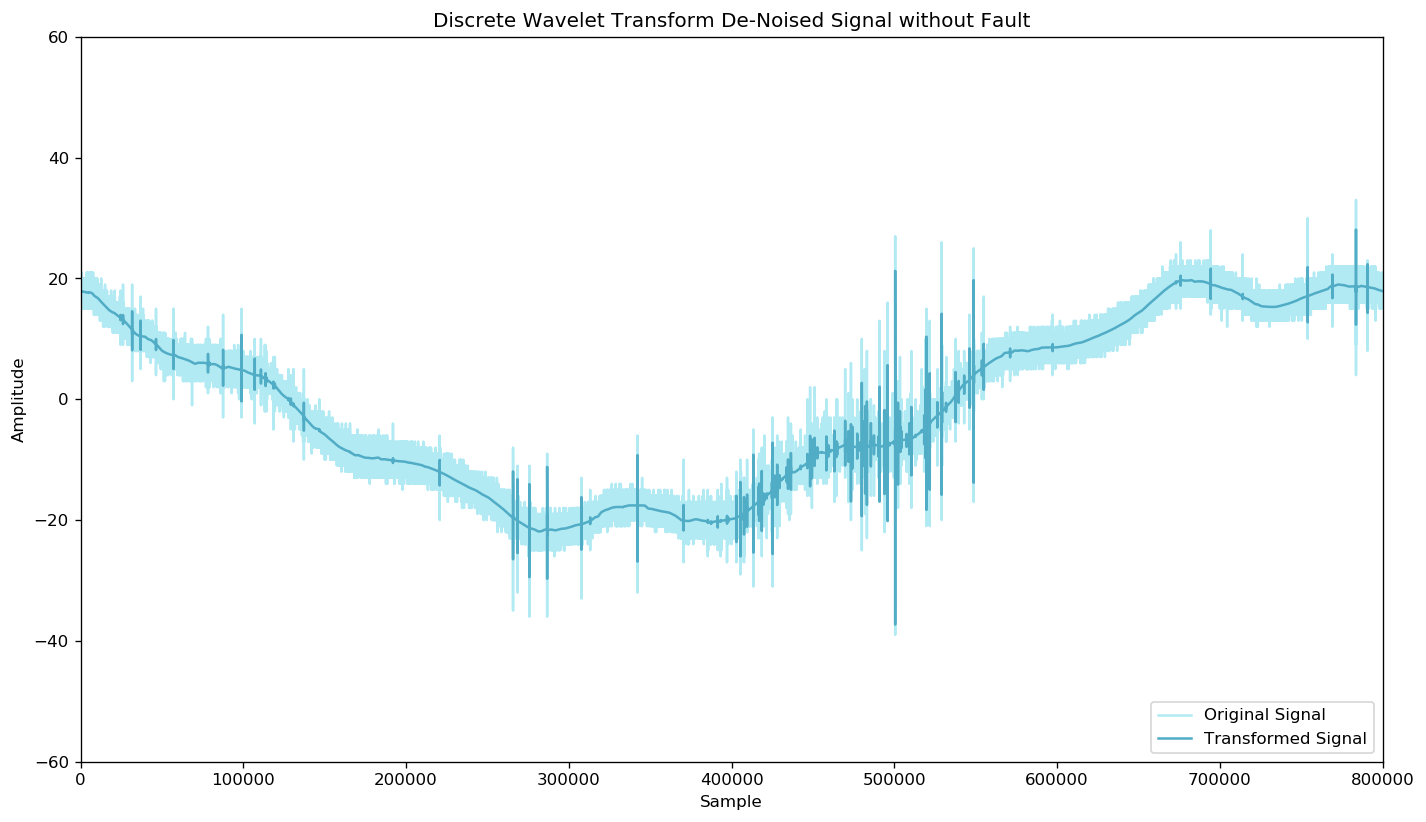
**sample\_submission.csv**: a valid sample submission.

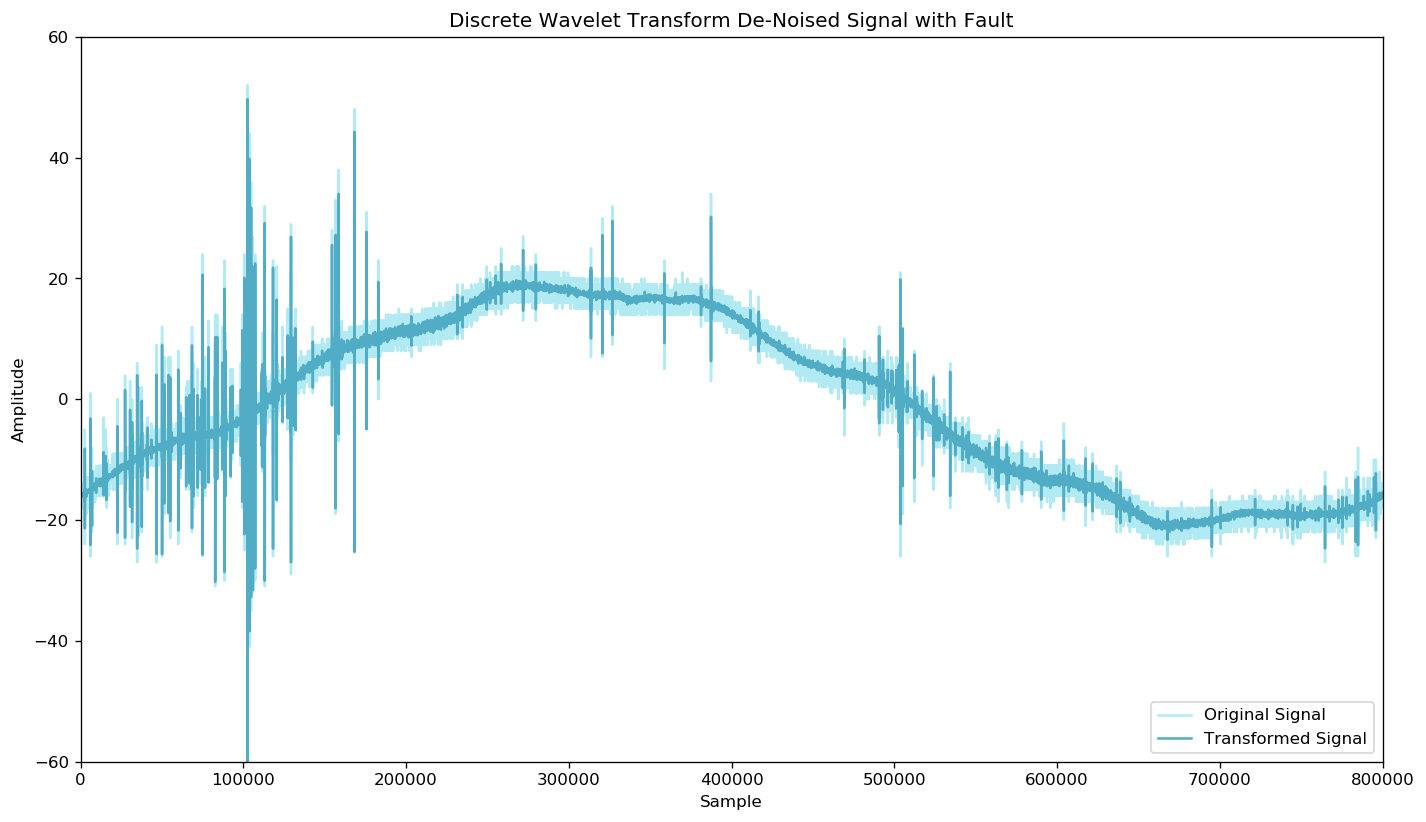
Given the data, supervised learning, binary classification,

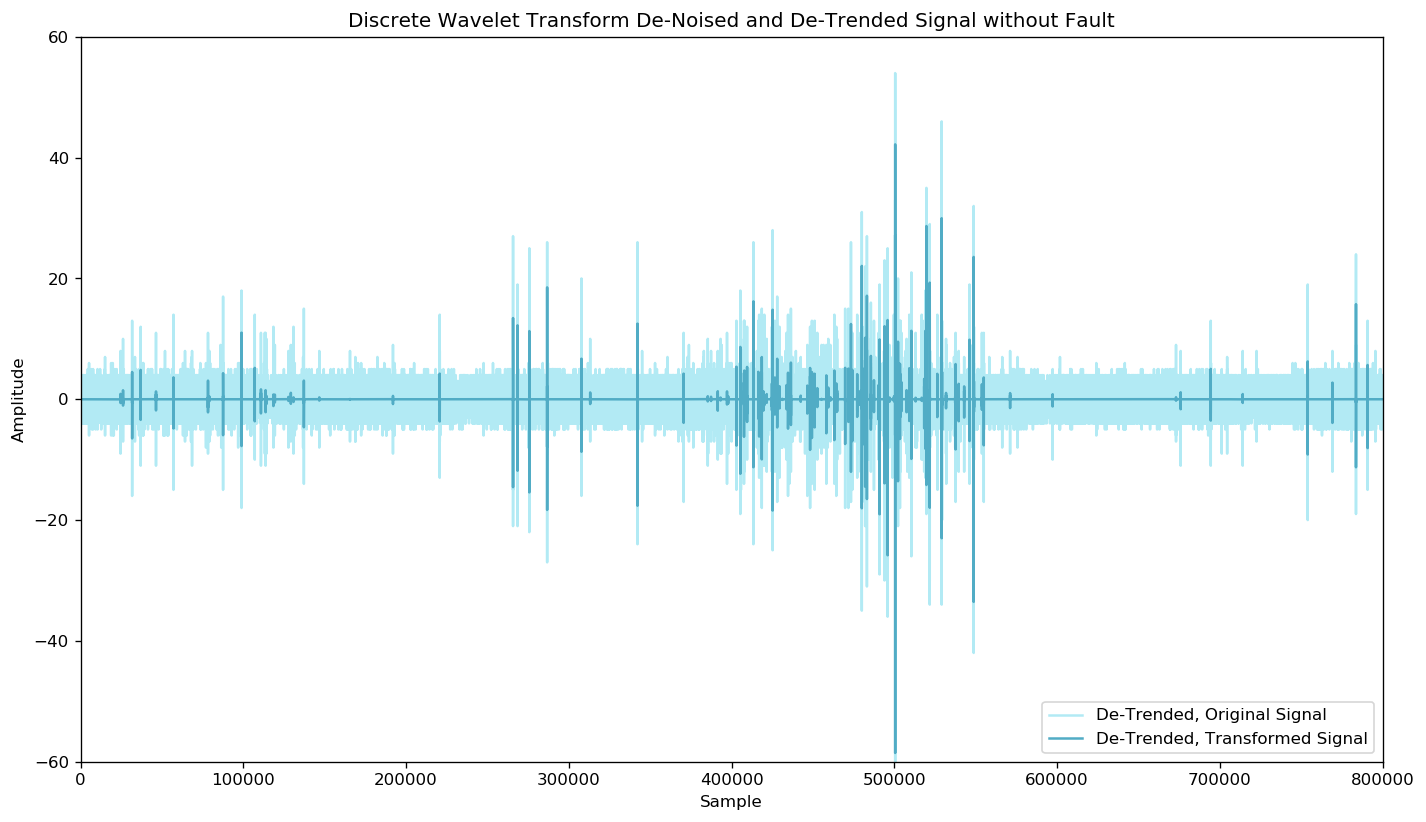
Plot raw signals

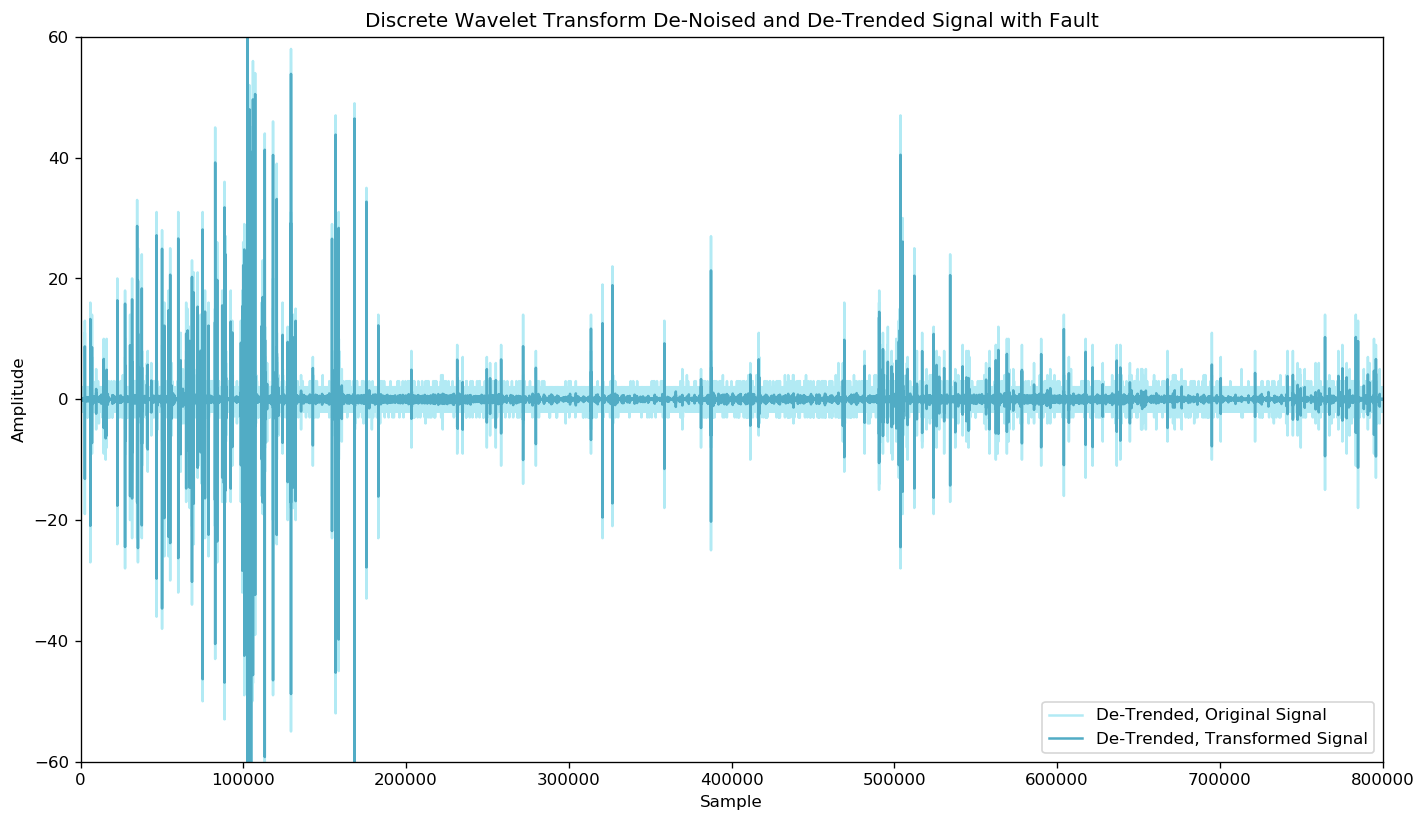












Imbalanced Data - % samples with fault

## Signal Processing

Pipeline

### Discrete Wavelet Transforms

### Removing Sinusoidal Trend Data

### Extraction of Signal Features

## Feature Matrix

# Prototype and Survey Classification Models

As previously stated, this problem requires a model capable of binary classification. Three classification models were surveyed that might be suitable for the task:

* k-Nearest Neighbors
* Support Vector Machine
* Random Forest

In the following sub-sections, this report offers a brief introduction to each model type, a prototyped solution developed in Python, and an initial investigation into results and viability.

At the conclusion of this section, the best performing model is identified and selected for further refinement. Model classification performance is being judged using the sponsor’s scoring function: Matthews Correlation Coefficient.

## k-Nearest Neighbors Classifier

The [k-nearest neighbors](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm) (k-NN) algorithm is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors where *k* is generally a small, positive, and odd integer value.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/*d*, where *d* is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.

## Support Vector Machine

A [Support Vector Machine](https://en.wikipedia.org/wiki/Support-vector_machine) (SVM) is a supervised machine learning model with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples with noted category, a SVM training algorithm builds a model that will assign new examples to one category or the other, making is a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

## Random Forest

[Random Forests](https://en.wikipedia.org/wiki/Random_forest) are an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of overfitting to their training set.

Random Forests’ resilience to overfitting the training set is an important trait to leverage for this project where the training data is highly imbalanced between the two classes.

# Final Model Selection and Refinement

Submission File

For each signal in the test set, you must predict a binary prediction for the target variable. The file should contain a header and have the following format:

signal\_id,target  
0,0  
1,1  
2,0  
etc.

# Conclusions

SLTM model