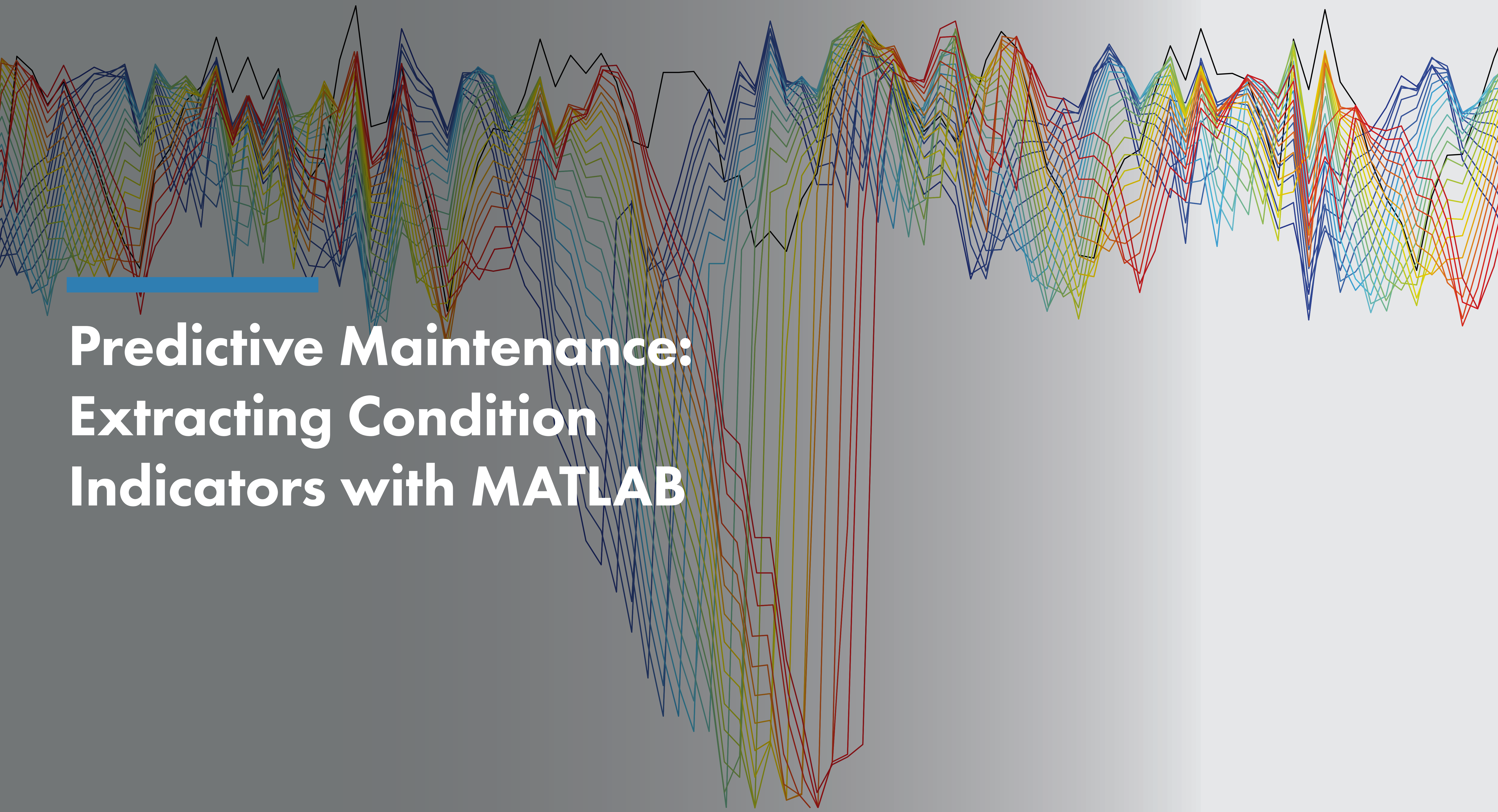
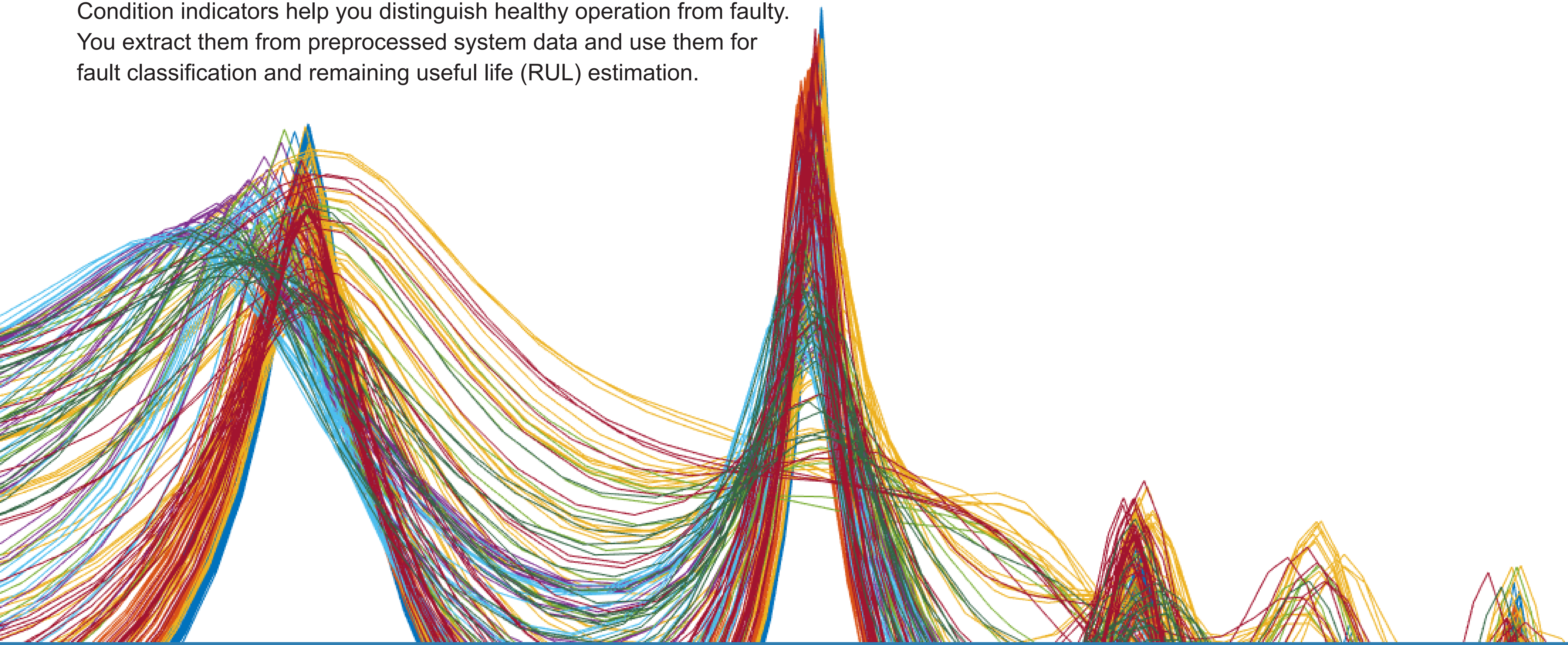


Predictive Maintenance: Extracting Condition Indicators with MATLAB

MathWorks®

What Is a Condition Indicator?

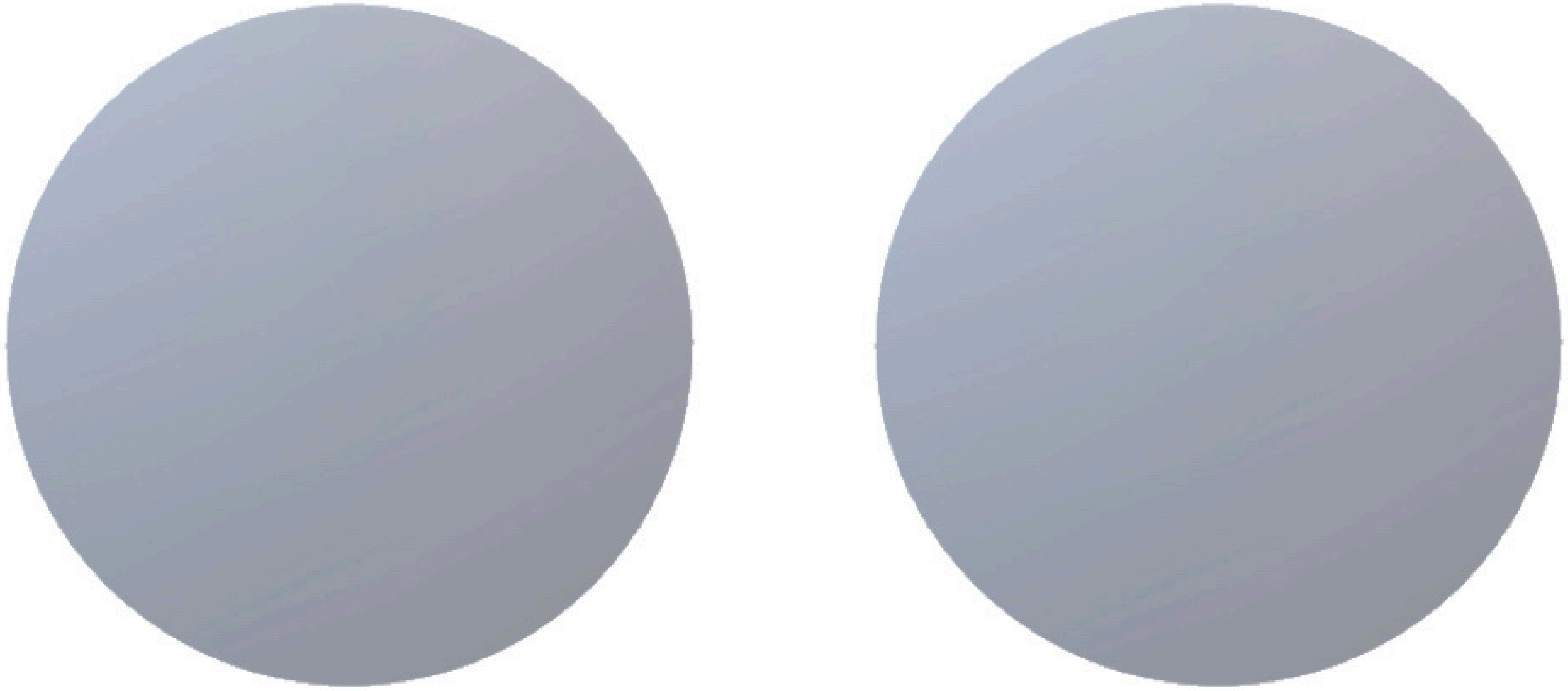
A key step in predictive maintenance algorithm development is identifying *condition indicators*: features in your system data whose behavior changes in a predictable way as the system degrades. Condition indicators help you distinguish healthy operation from faulty. You extract them from preprocessed system data and use them for fault classification and remaining useful life (RUL) estimation.



A Visual Exercise

Let’s start with a visual exercise to understand how condition indicators work. What’s the difference between these two shapes?

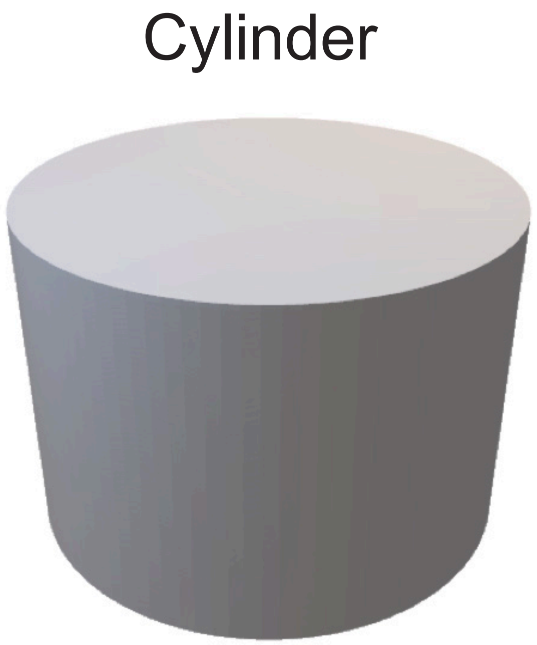
Circles



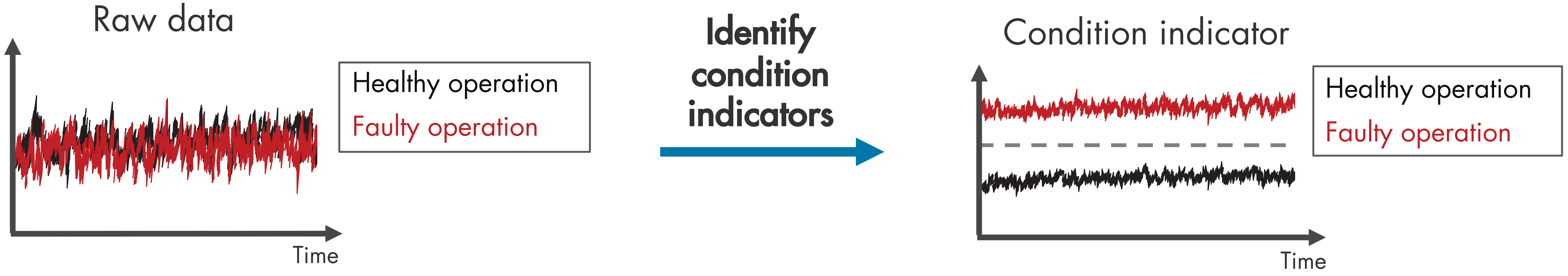
It looks like there’s no significant difference because the two circles look almost the same.

A Visual Exercise - Continued

On the previous page, the shapes looked the same because you were looking at them from a certain angle, the top view. However, if you change your perspective, you can clearly see the differences between the two shapes and can identify them as a cone and a cylinder.

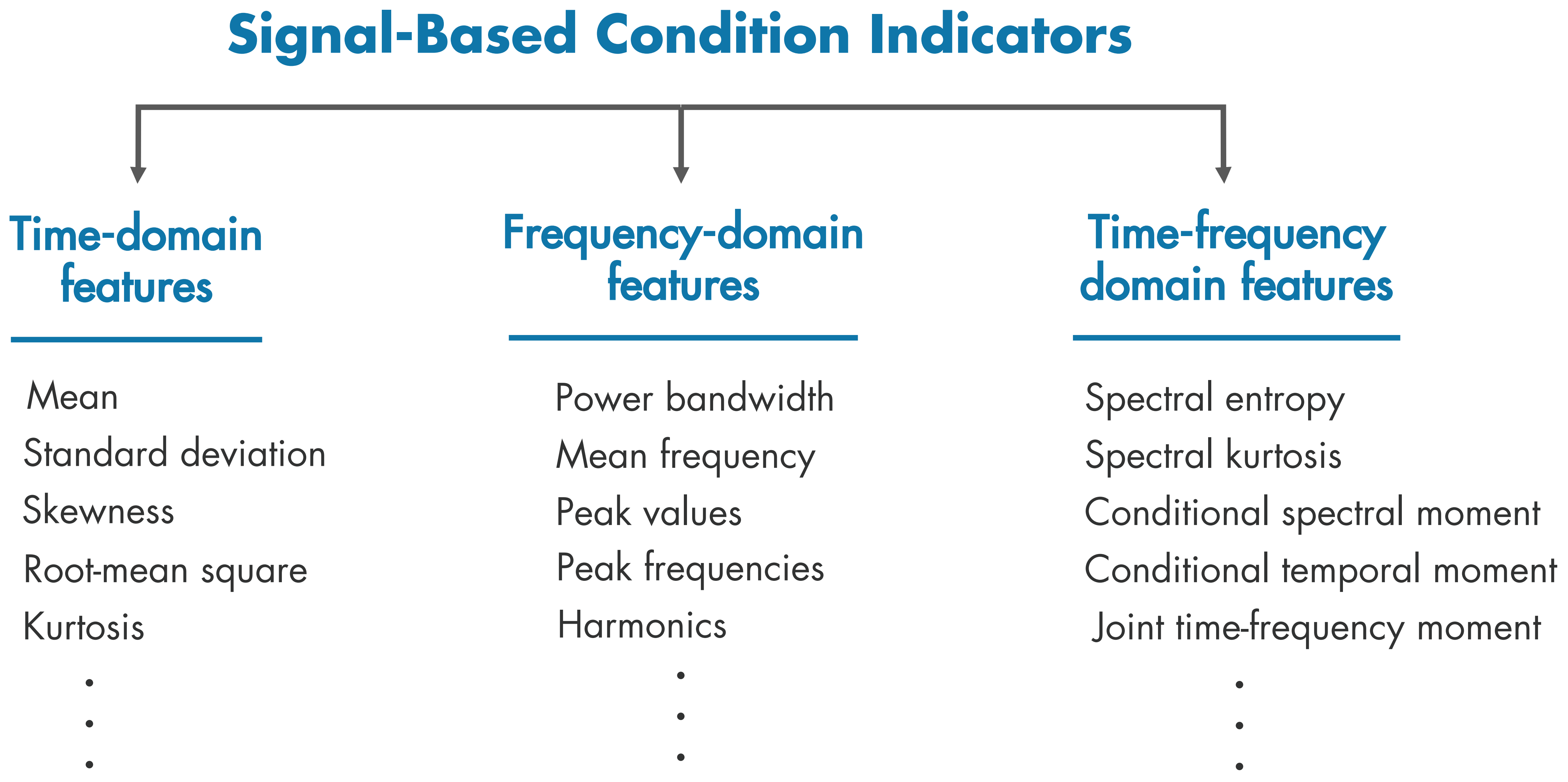


Similarly, when you look at raw measurement data from your machine, it's hard to tell healthy operation from faulty. But, using condition indicators, you're able to look at the data from a different perspective that helps you discriminate between healthy and faulty operation.



Feature Extraction Using Signal-Based Methods

You can derive condition indicators from data by using time, frequency, and time-frequency domain features:



Signal-Based Condition Indicators

Time-domain features

- Mean
- Standard deviation
- Skewness
- Root-mean square
- Kurtosis
- ⋮

Frequency-domain features

- Power bandwidth
- Mean frequency
- Peak values
- Peak frequencies
- Harmonics
- ⋮

Time-frequency domain features

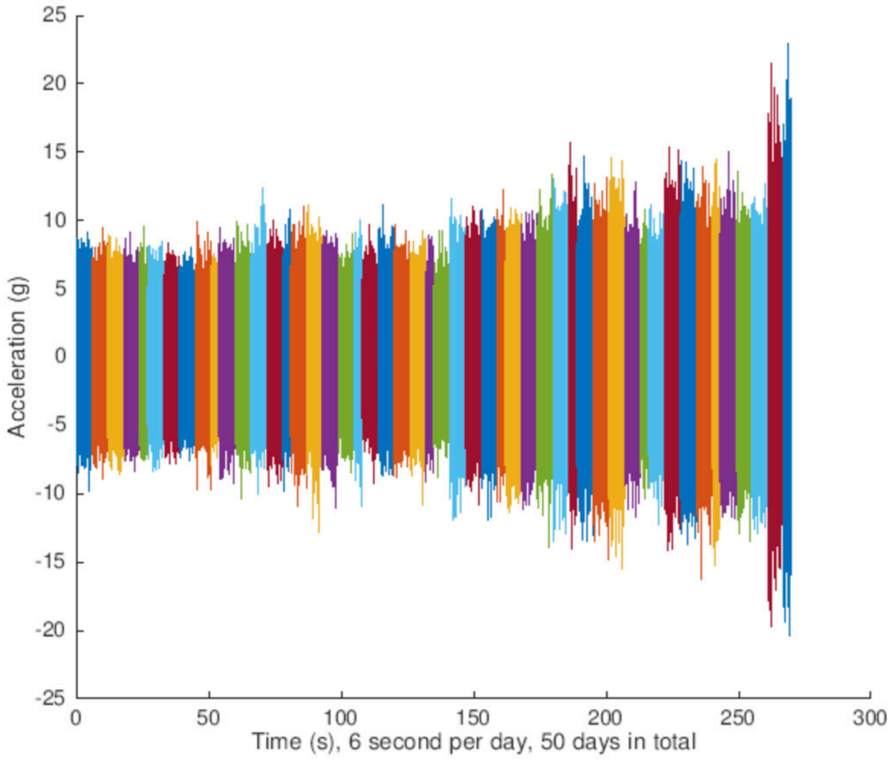
- Spectral entropy
- Spectral kurtosis
- Conditional spectral moment
- Conditional temporal moment
- Joint time-frequency moment
- ⋮

Feature Extraction Using Signal-Based Methods - Continued

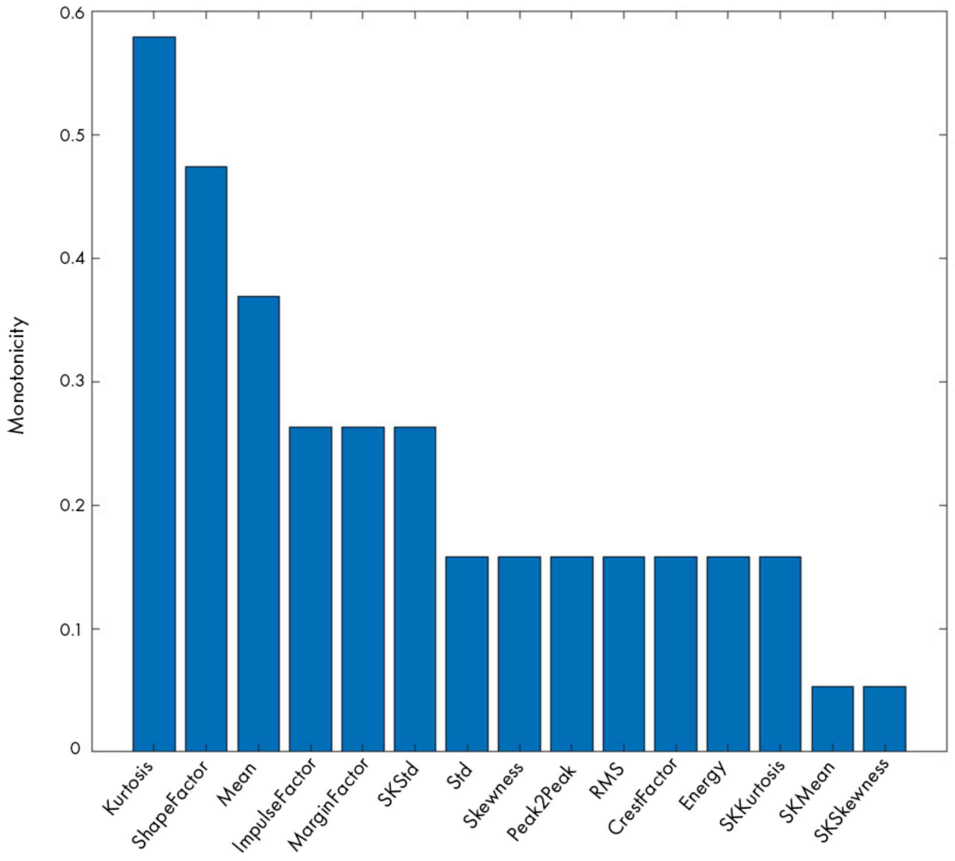
Time-Domain Features

For some systems, simple statistical features of time signals can serve as condition indicators, distinguishing faulty conditions from healthy. For example, the average value of a particular signal or its standard deviation might change as system health degrades. You can also use higher-order moments of the signal such as skewness and kurtosis. With such features, you can try to identify threshold values that distinguish healthy operation from faulty operation, or look for abrupt changes in the value that mark changes in system state.

Predictive Maintenance Toolbox™, an add-on product for MATLAB®, contains additional functions for computing time-domain features, demonstrated in the example *Turbine High-Speed Bearing Prognosis*.



The changing trend in time-domain vibration signals shows the degradation of a wind turbine high-speed shaft over 50 consecutive days.



The monotonicity graph shows the feature importance ranking for different time-domain and frequency-domain features.

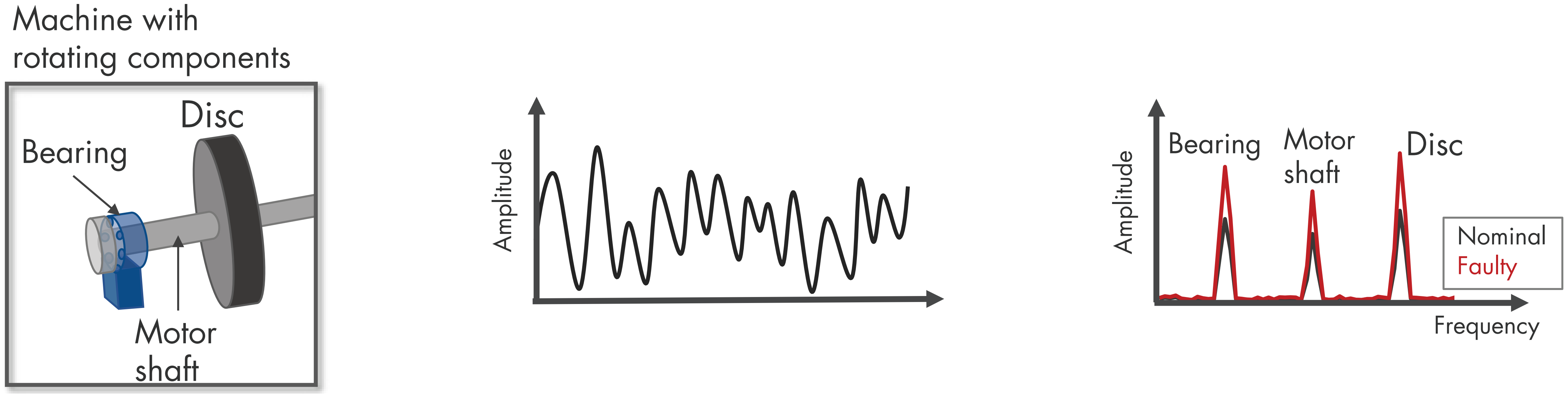
Feature Extraction Using Signal-Based Methods - Continued

Frequency-Domain Features

Sometimes, time-domain features alone are not sufficient to act as condition indicators, so you'll want to look at frequency-domain features as well.

Take a machine with rotational components and three vibration sources: bearing, motor shaft, and disc. If you look at the vibration data from the machine in the time domain, you see the combined effect of all the vibrations from these different rotating components. But by analyzing the data in the frequency domain, you can isolate different sources of vibration, as seen in the second plot. The peak amplitudes, and how much they change from nominal values, can indicate the severity of the faults.

More information on calculating frequency-domain condition indicators can be found in the example *Condition Monitoring and Prognostics Using Vibration Signals.*



When analyzing the vibration data from the machine in the time domain, the bearing, motor shaft, and disc all affect vibration amplitude. Using frequency-domain analysis, you can distinguish between different sources of vibration.

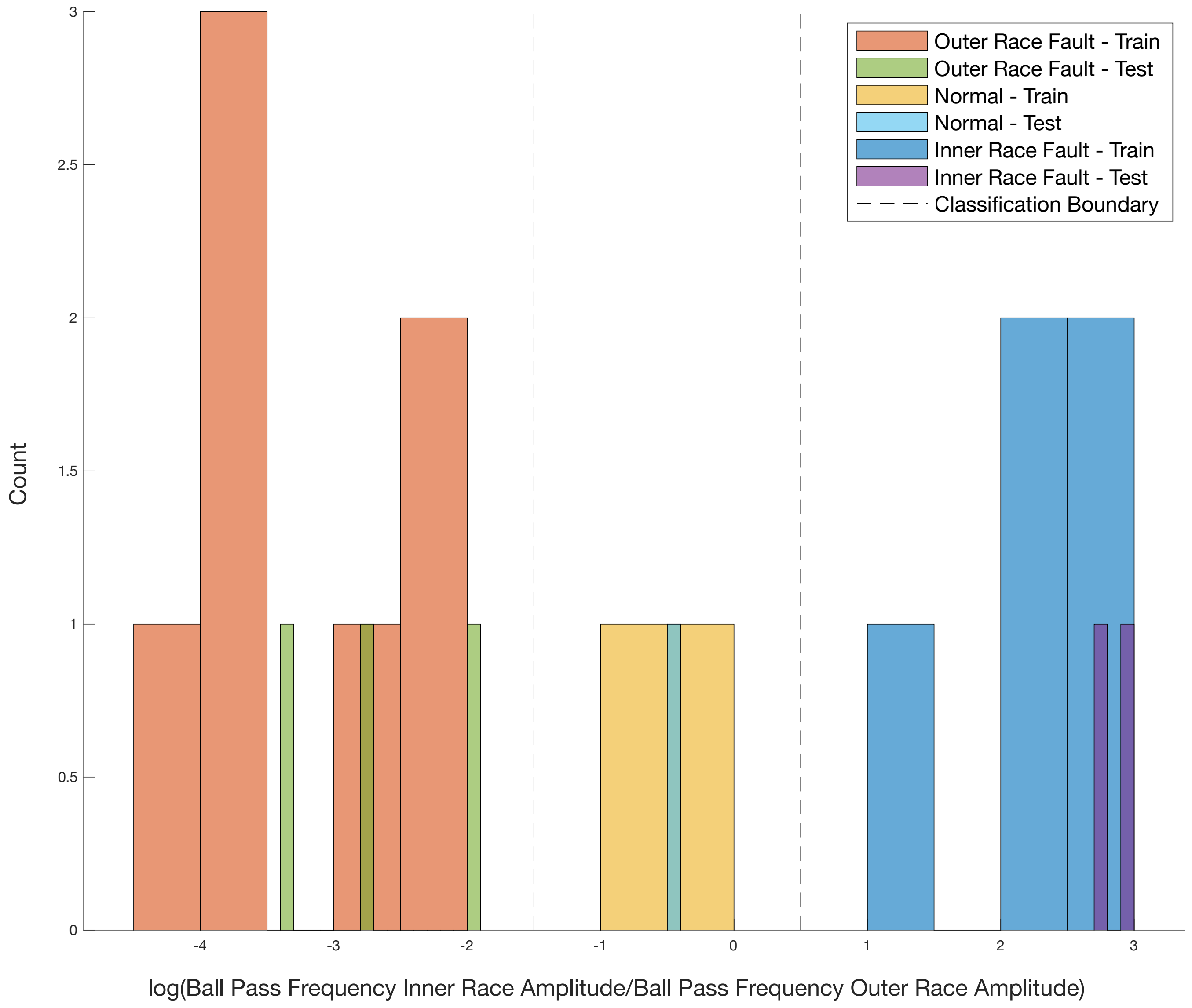
Feature Extraction Using Signal-Based Methods - Continued

Time-Frequency Domain Features

Another way to extract features is to perform time-frequency spectral analysis on the data, which helps characterize changes in the spectral content of a signal over time. Time-frequency domain condition indicators include features such as spectral kurtosis and spectral entropy.

The *Rolling Element Bearing Fault Diagnosis* example shows how to use kurtogram, spectral kurtosis, and envelope spectrum to identify different types of faults in rolling element bearings.

» *Learn more about time, frequency, and time-frequency domain features*

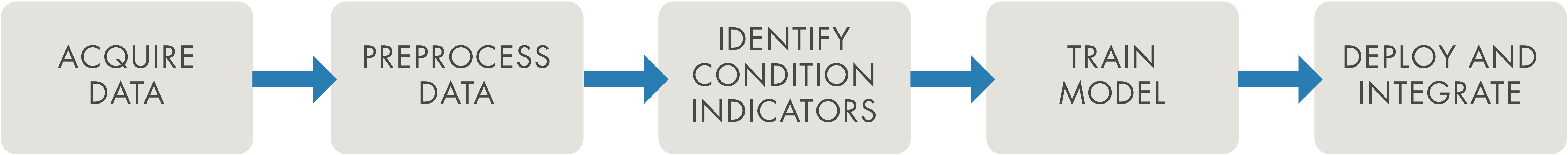


The histogram shows a clear separation between the three bearing conditions. The log ratio between the bandpass frequency inner and outer race amplitudes is a valid feature to classify bearing faults.

Predictive Maintenance Workflow

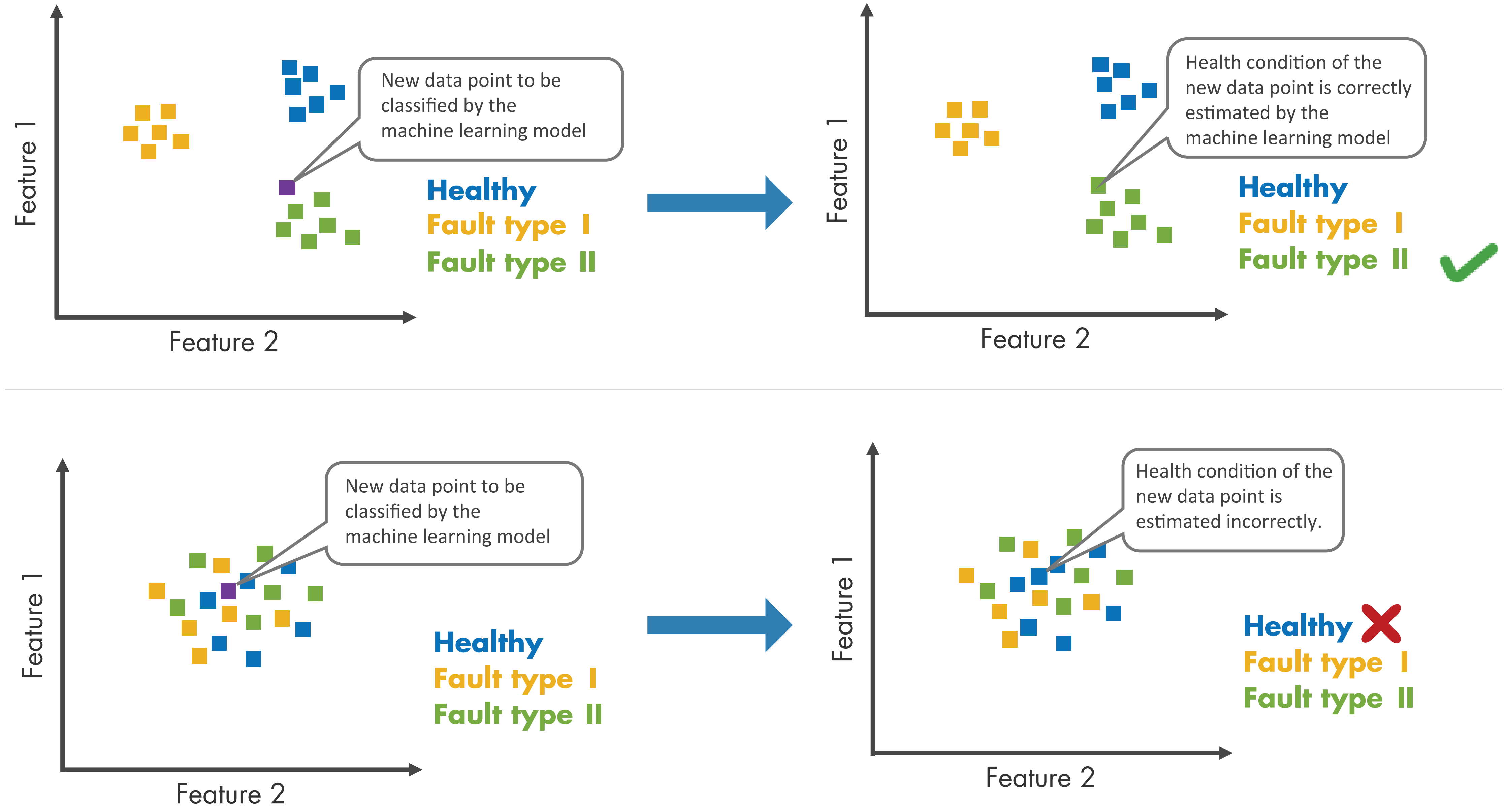
Now that you've looked at what condition indicators are, it's time to take a look at the steps involved in extracting the features for identifying condition indicators. Designing a predictive maintenance algorithm starts with collecting data from your machine under different operating conditions and fault states. The raw data is then preprocessed for cleaning it up and bringing it into a form from which condition indicators can be extracted. Feature extraction looks for features that are distinctive, meaning they uniquely define healthy operation and different fault types. These features will be your *condition indicators.*

Using the extracted features, you can train a machine learning model for fault classification and remaining useful life estimation. You can then deploy your algorithm and integrate it into your systems for machine monitoring and maintenance.


ACQUIRE DATA → PREPROCESS DATA → IDENTIFY CONDITION INDICATORS → TRAIN MODEL → DEPLOY AND INTEGRATE

What Are Distinctive Features and Why Are They Important?

Once you identify some useful features, you can use them to train a machine learning model. If the selected set of features is distinctive, the model can correctly estimate the machine's current condition when you feed new data from the machine to the model.



New data point to be classified by the machine learning model

Health condition of the new data point is correctly estimated by the machine learning model

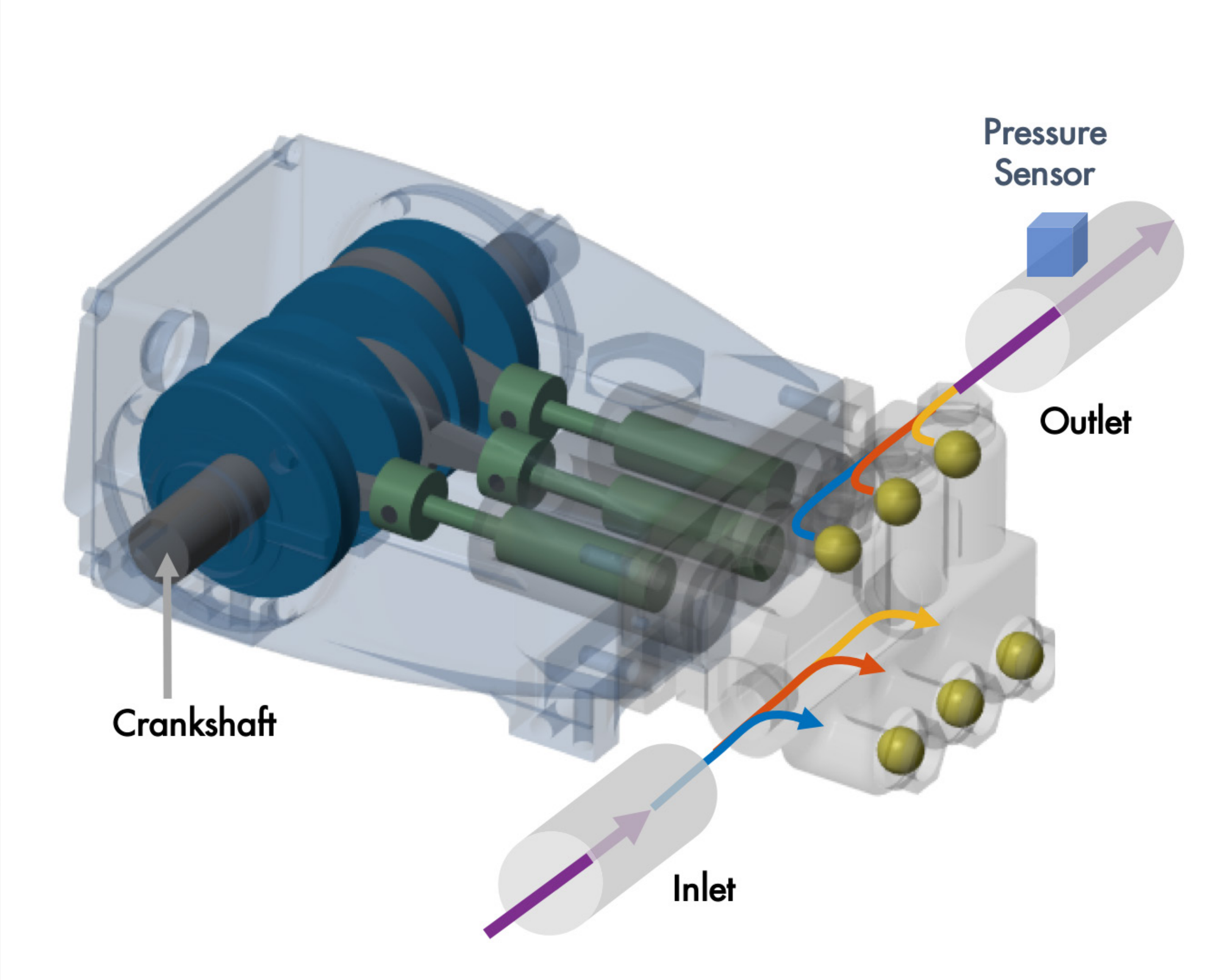
New data point to be classified by the machine learning model

Health condition of the new data point is estimated incorrectly.

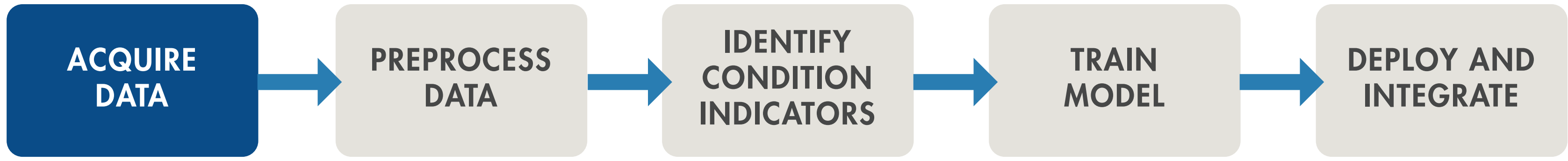
Example: The Triplex Pump

This example uses a triplex pump to demonstrate the workflow. The pump has a motor that turns the crankshaft that in turn drives three plungers. The fluid gets sucked into the inlet and discharged through the outlet, where the pressure is measured by a sensor. Faults that can develop in such a pump include:

- Seal leakage
- Blocked inlet
- Worn bearing

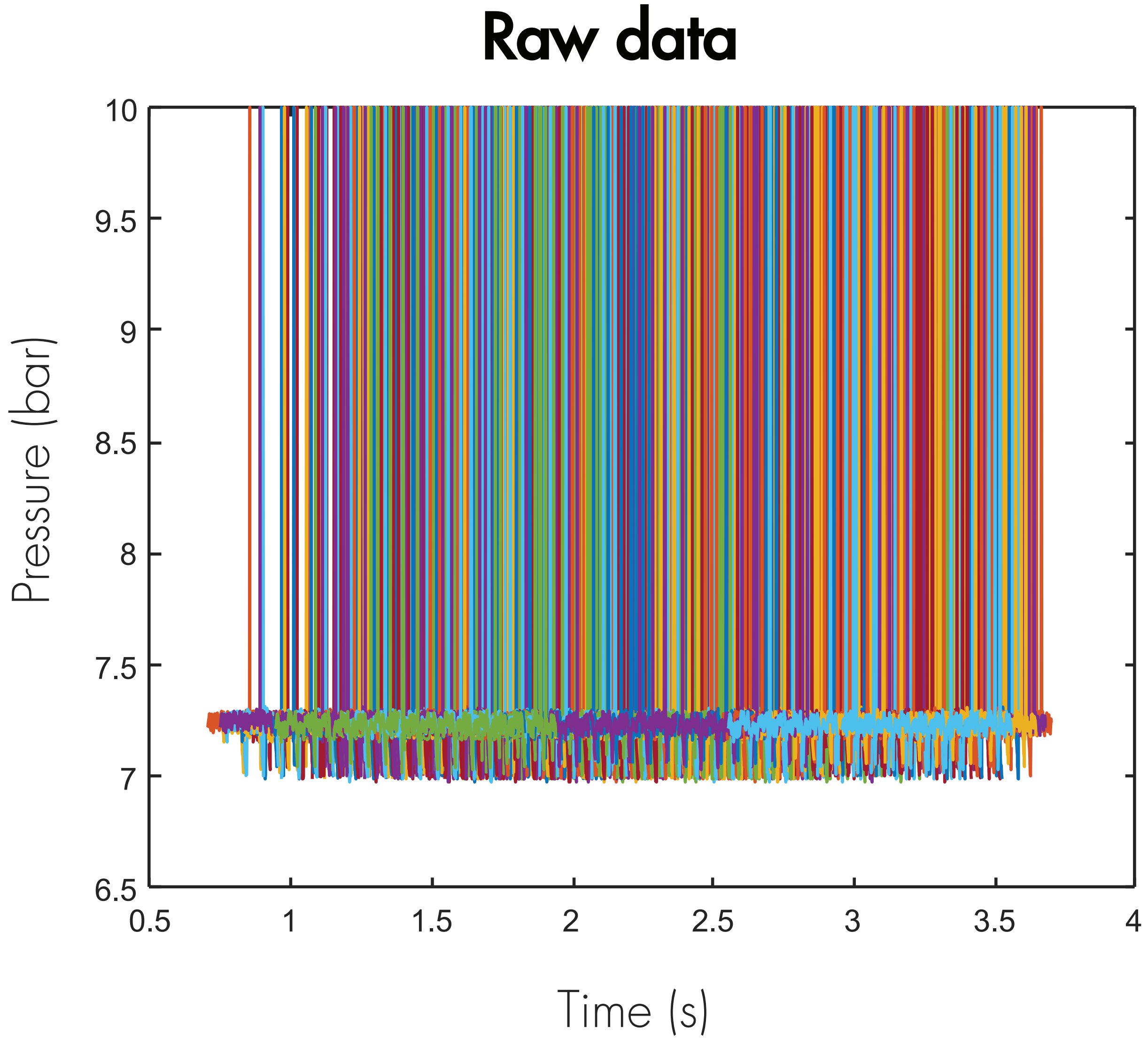


Acquiring Data



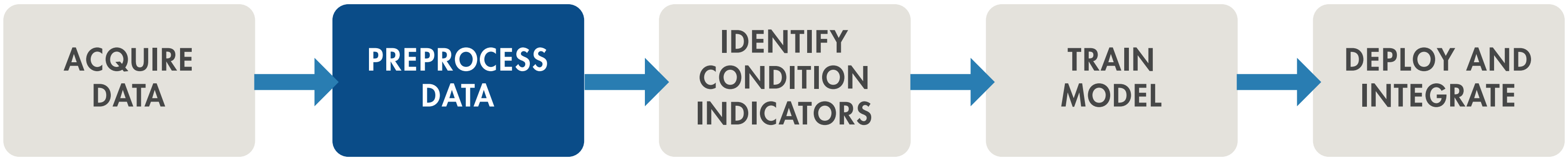
This pressure data includes one-second-long measurements taken at steady state from normal operation, all three fault types, and also their combinations:

- Healthy
- Blocked inlet
- Worn bearing
- Seal leakage
- Blocked inlet, worn bearing
- Seal leakage, worn bearing
- Seal leakage, blocked inlet

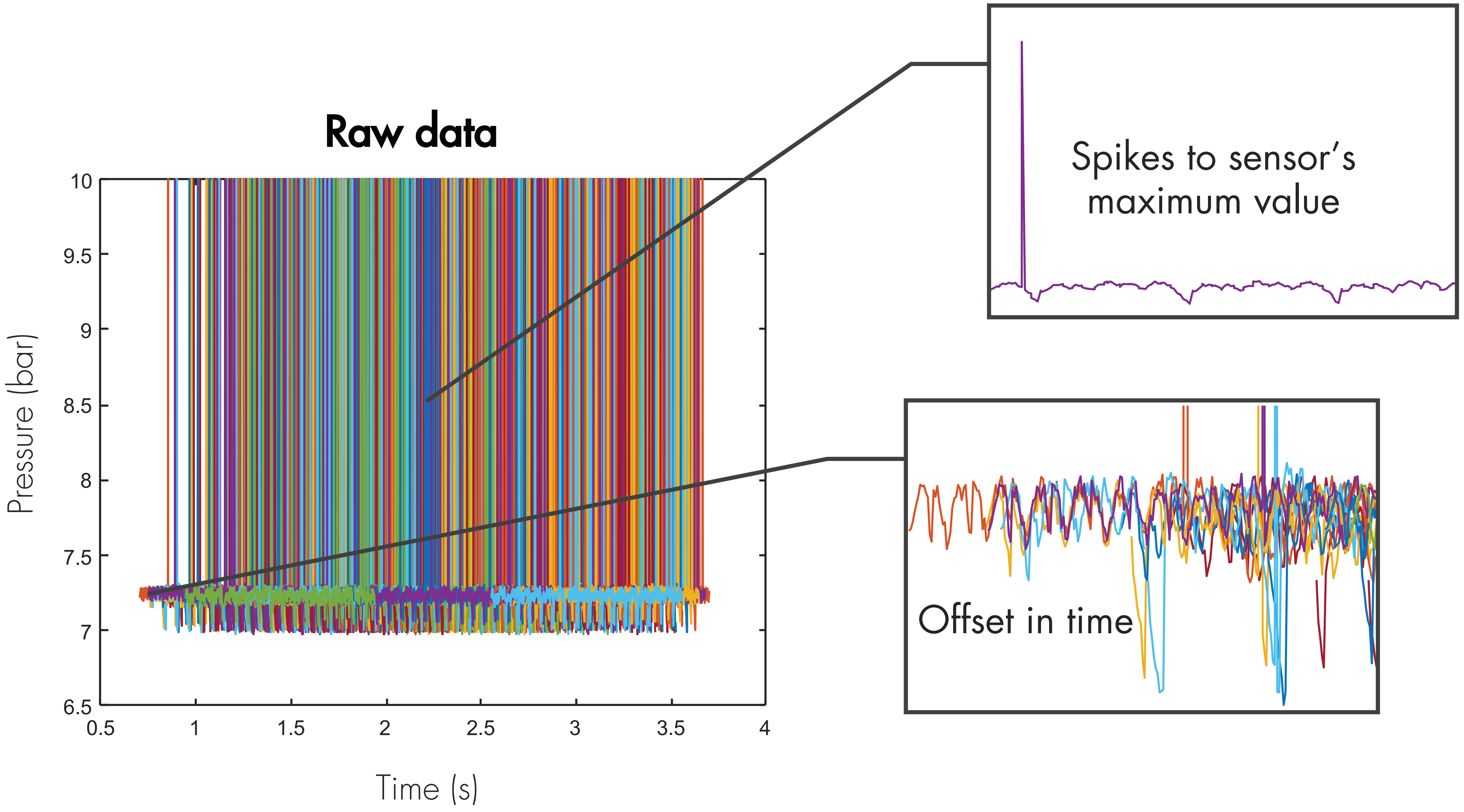


Plot of the pressure data that has been collected from the triplex pump.

Preprocessing Data



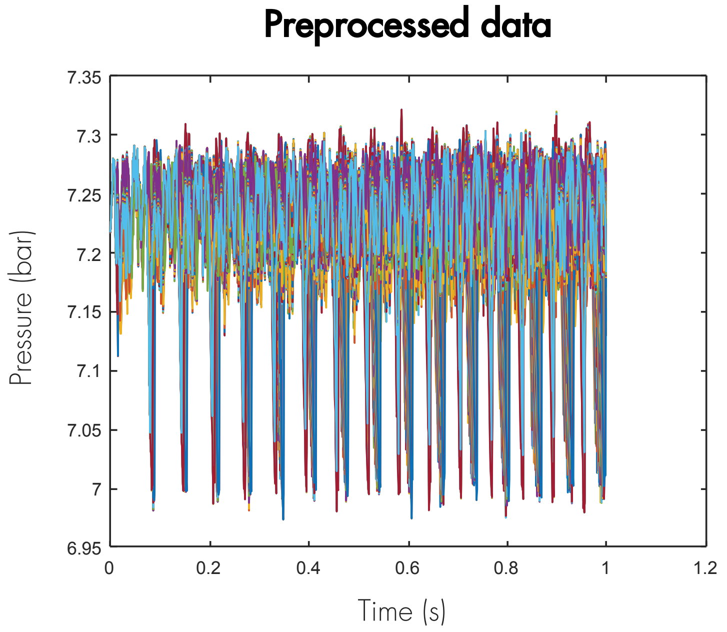
You need to bring the data into a usable form to extract condition indicators. The raw data is noisy and has spikes up to the sensor's maximum value. It's also offset in time even though the durations of the measurements are the same.



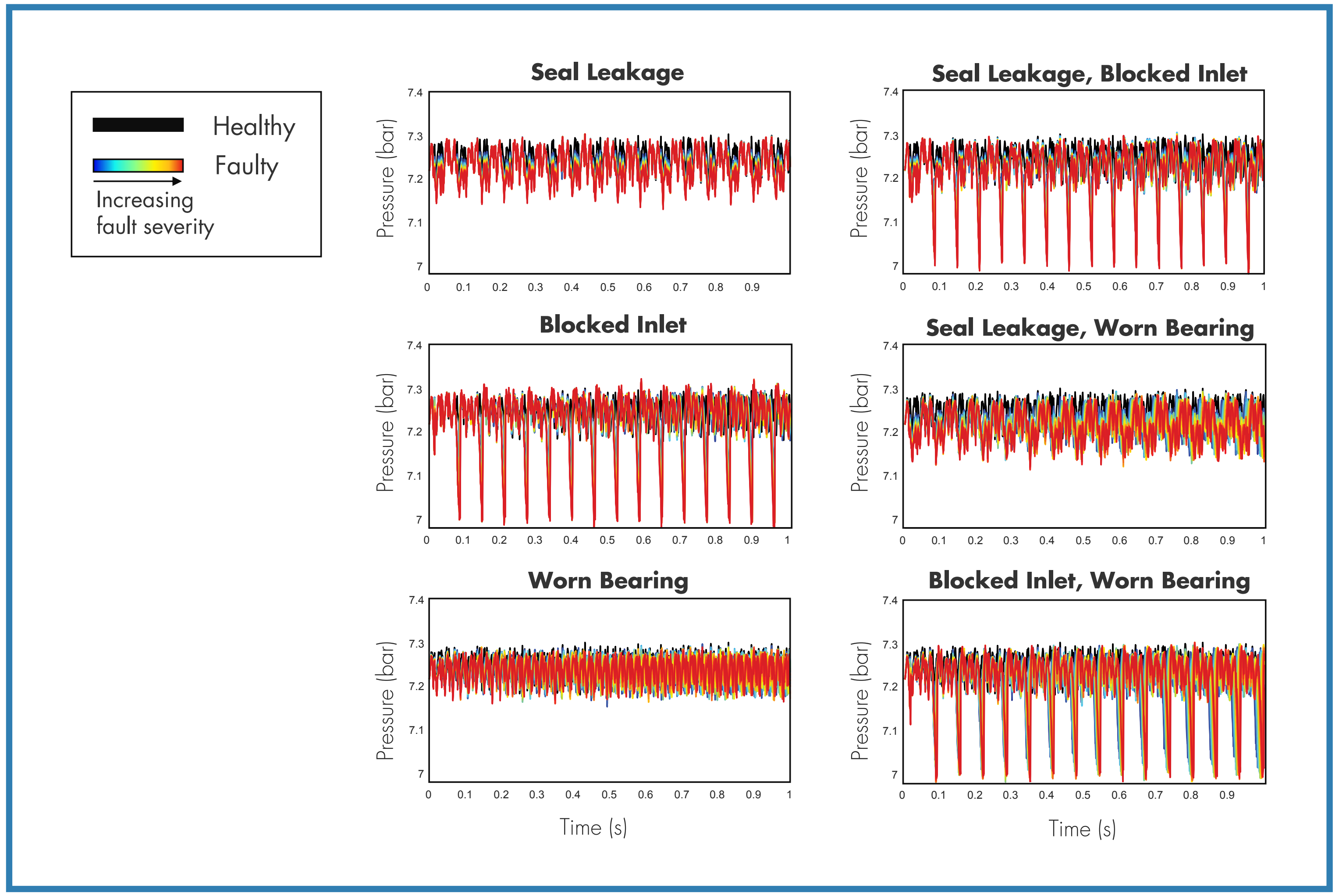
Preprocessing Data - Continued

MATLAB has *functions* to help smooth, denoise, and perform other preprocessing techniques on signal data.

The processed data includes all the healthy and faulty conditions. In order to investigate different fault types and their combinations, you can plot them individually. The first thing to notice on these plots is the cyclical behavior of the time-domain pressure signal. Next, take a look at a plot of a shorter time period on the next page to see what's happening in each cycle more clearly.



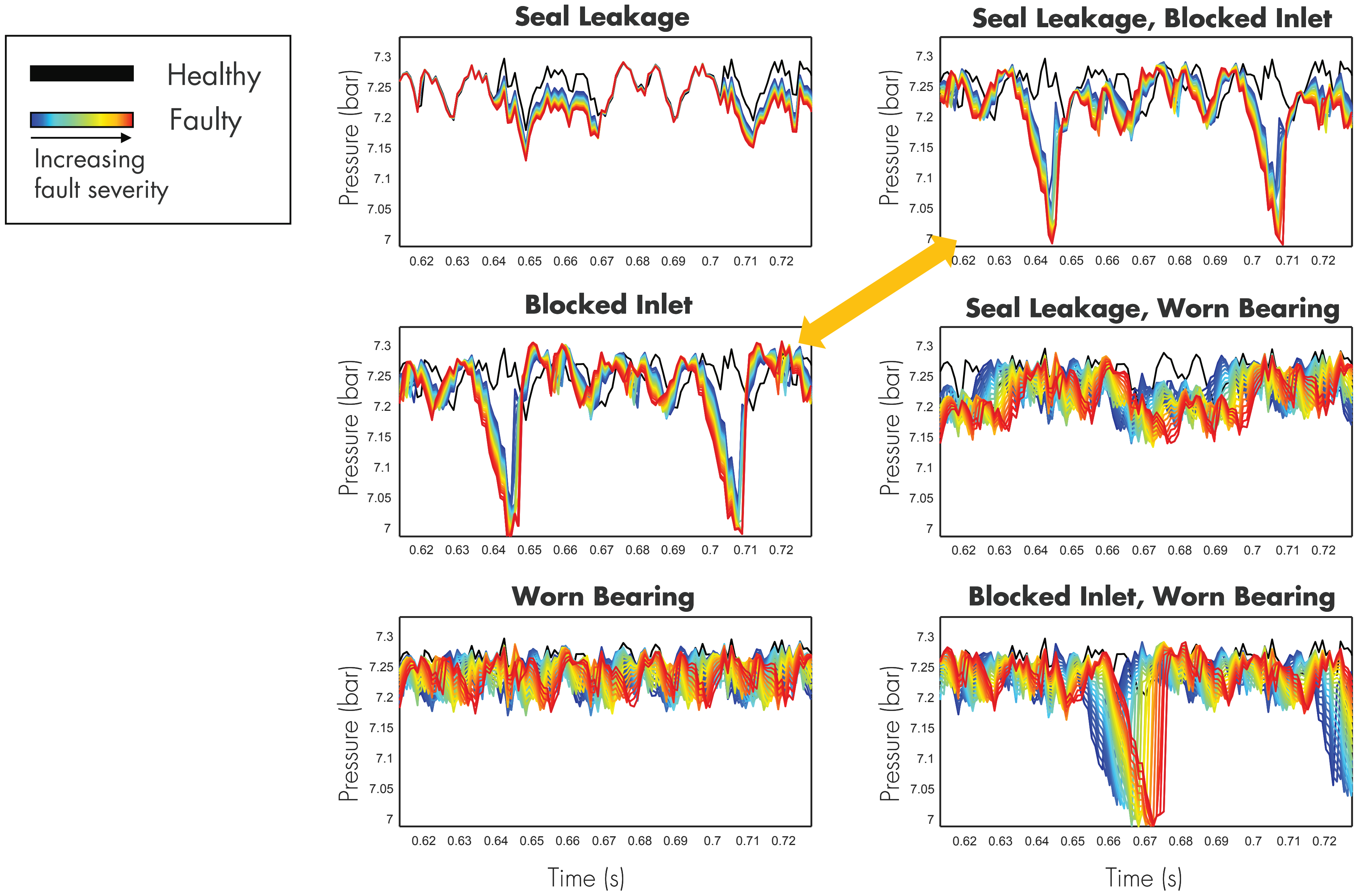
Graph Data Mapped to 6 Plots



Preprocessing Data - Continued

These plots provide a more detailed look at the pressure signal for different types of faults. The change in the pressure data as the pump deteriorates is reflected with changing colors from dark blue to red indicating increasing fault severity. Now the question is: Can you distinguish the black line, the healthy operation, from the rest of the data on each plot? And, can you identify the unique differences between each set of colored lines? Notice how the pressure data looks very similar for the “seal leakage, blocked inlet” and “blocked inlet” faults.

Now let’s look at some of the time-domain features to identify condition indicators to help you distinguish between fault types.



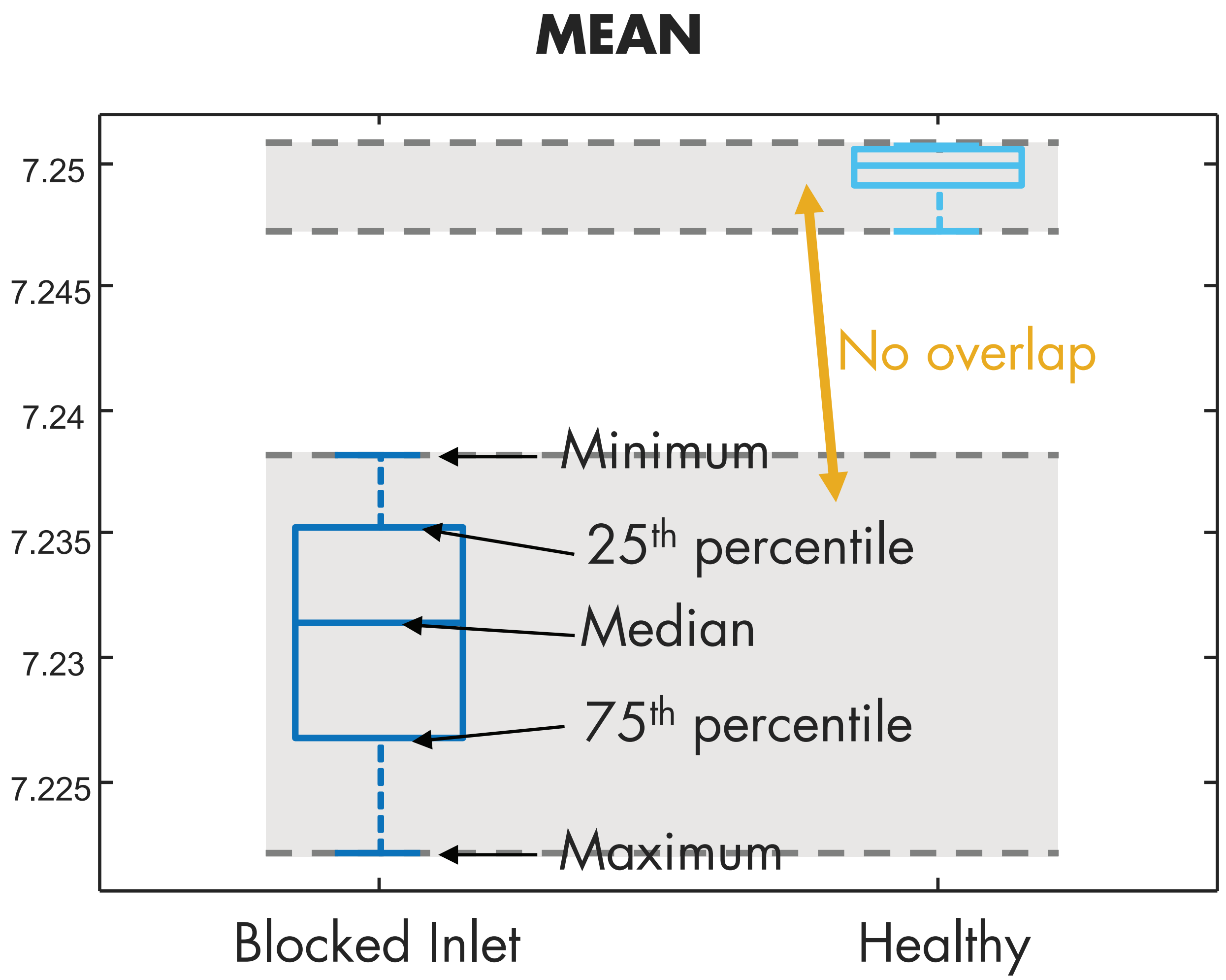
Using Time-Domain Features for Identifying Condition Indicators

ACQUIRE DATA
PREPROCESS DATA
IDENTIFY CONDITION INDICATORS
TRAIN MODEL
DEPLOY AND INTEGRATE

Use trial and error to see how each of the following set of common time-domain features performs: *Mean*, *Variance*, *Skewness*, and *Kurtosis.*

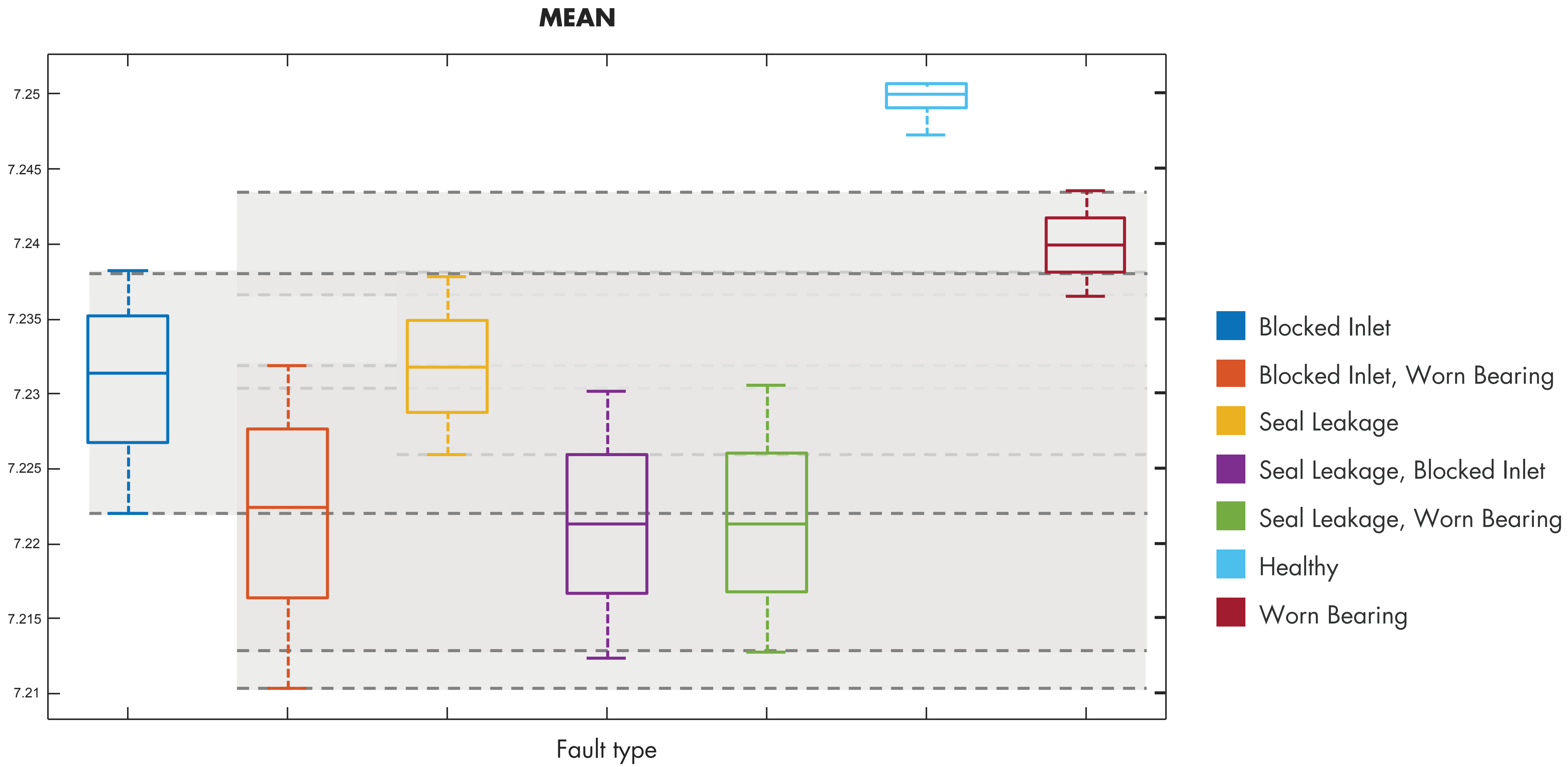
One way to understand if these condition indicators can differentiate between types of faults is to investigate them using a boxplot. First, plot a single feature, such as the mean, for the healthy condition and blocked inlet fault.

The boxes don't overlap in the plot. This means there's a difference between these data groups. By using the mean of the pressure data, you can easily distinguish the blocked inlet fault from healthy condition.



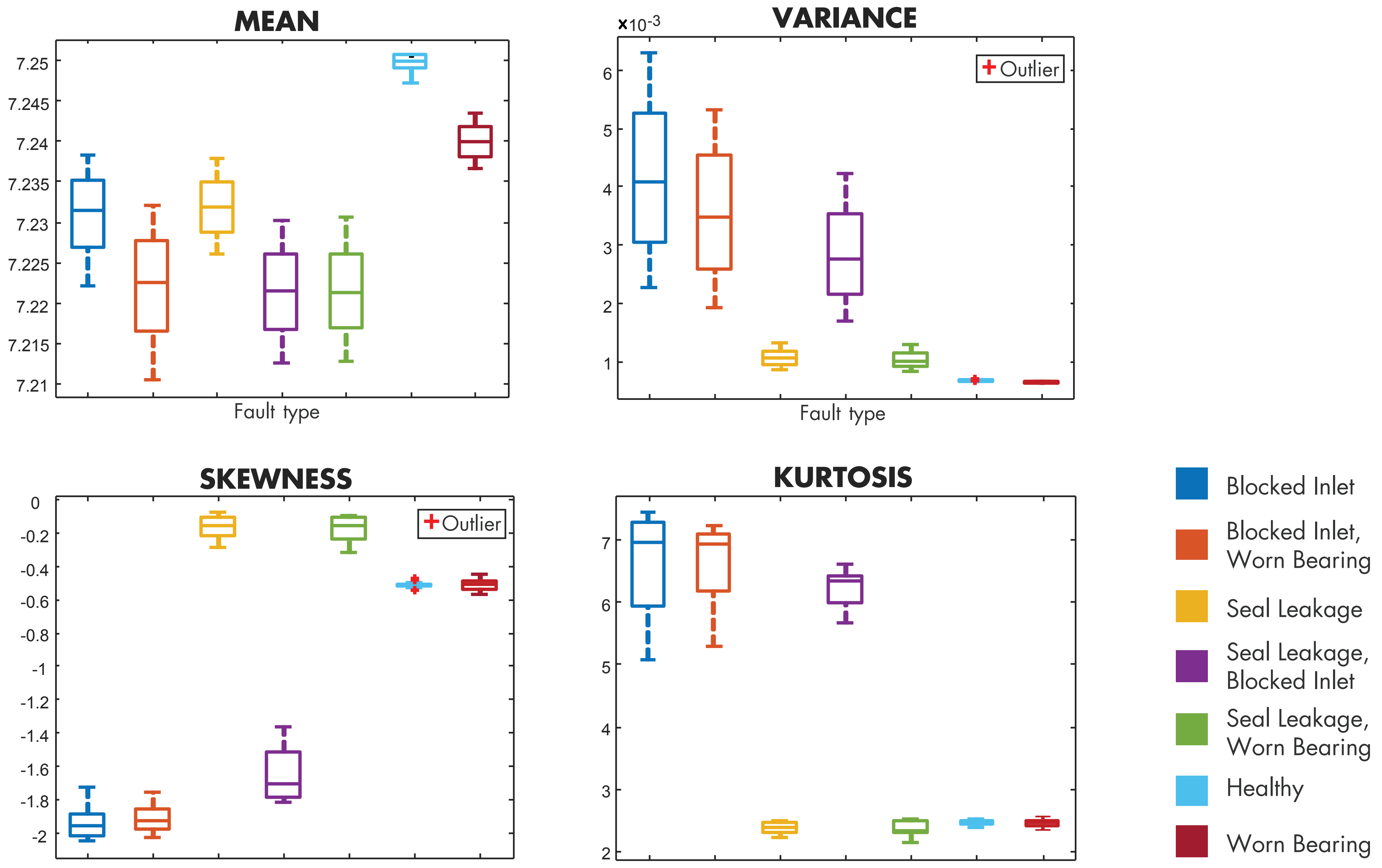
Using Time-Domain Features for Identifying Condition Indicators - Continued

Things change when you add the datasets for other fault types as well. You're not able to distinguish between all the fault types as some of them overlap. Due to this overlapping, the mean on its own is not enough to set fault types apart.



Using Time-Domain Features for Identifying Condition Indicators - Continued

If you try this with other features as well, you end up at the same conclusion: A single condition indicator is not sufficient to classify the faulty behavior, especially when you have multiple faults.



MEAN

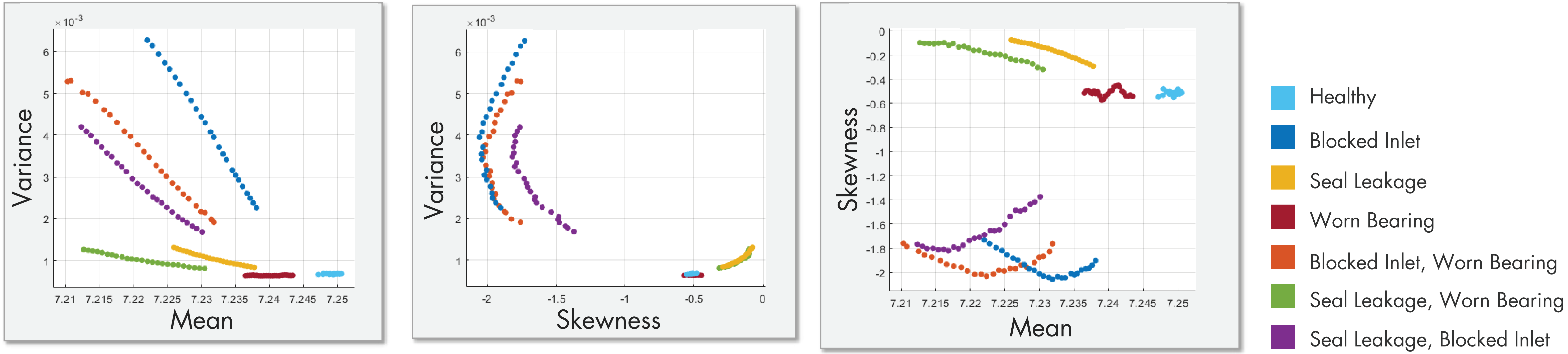
VARIANCE

SKEWNESS

KURTOSIS

Using Time-Domain Features for Identifying Condition Indicators - Continued

Below is a scatter plot of a combination of the features: mean, variance, and skewness. Notice how well the variance versus mean plot distinguishes between different types of faults.



You can immediately see that two condition indicators are better than one for separating different faults. You can try different pairs of features to see which ones are better at classifying faults.

Using Time-Domain Features for Identifying Condition Indicators - Continued

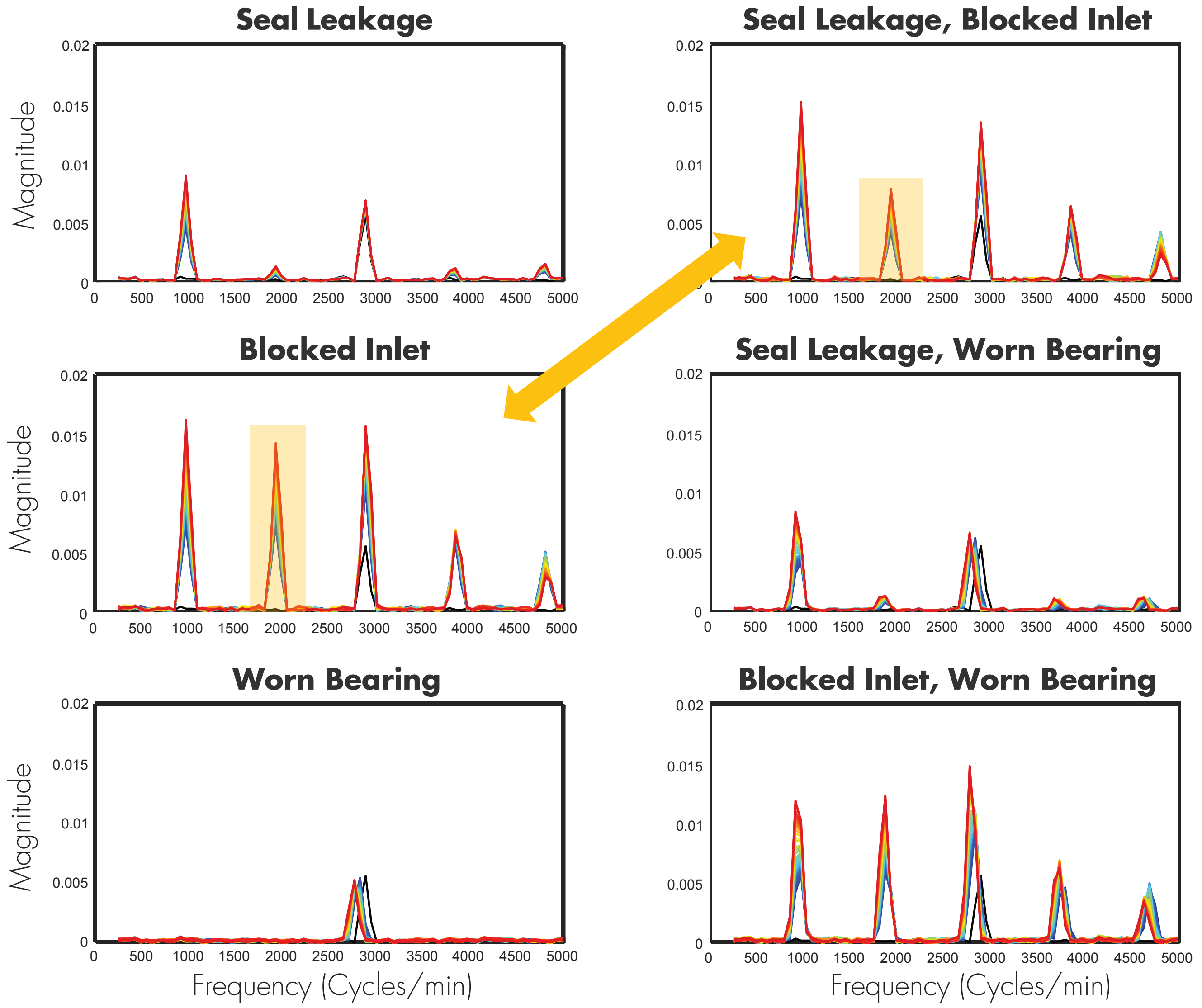
Frequency-domain analysis is important in analyzing periodic data and data acquired from a machine with rotating components; let's see if you can extract additional features by analyzing your data in the frequency domain.

What differentiates these plots from each other are the peaks and the peak frequencies, so they can serve as condition indicators. With time-domain features, it was hard to distinguish between the "seal leakage, blocked inlet" and "blocked inlet" faults because of the similarity of the datasets. By looking at the data in the frequency domain, you can see that the peak values at the highlighted frequency range will help you successfully separate these two faults.

In MATLAB, you can use **`fft`** function to compute the fast Fourier transform of a signal and analyze it in the frequency domain. You can then use the **`findpeaks`** function to extract the peaks and peak frequencies from the FFT signal.

In summary, the features you should use to train a machine learning model in this example include:

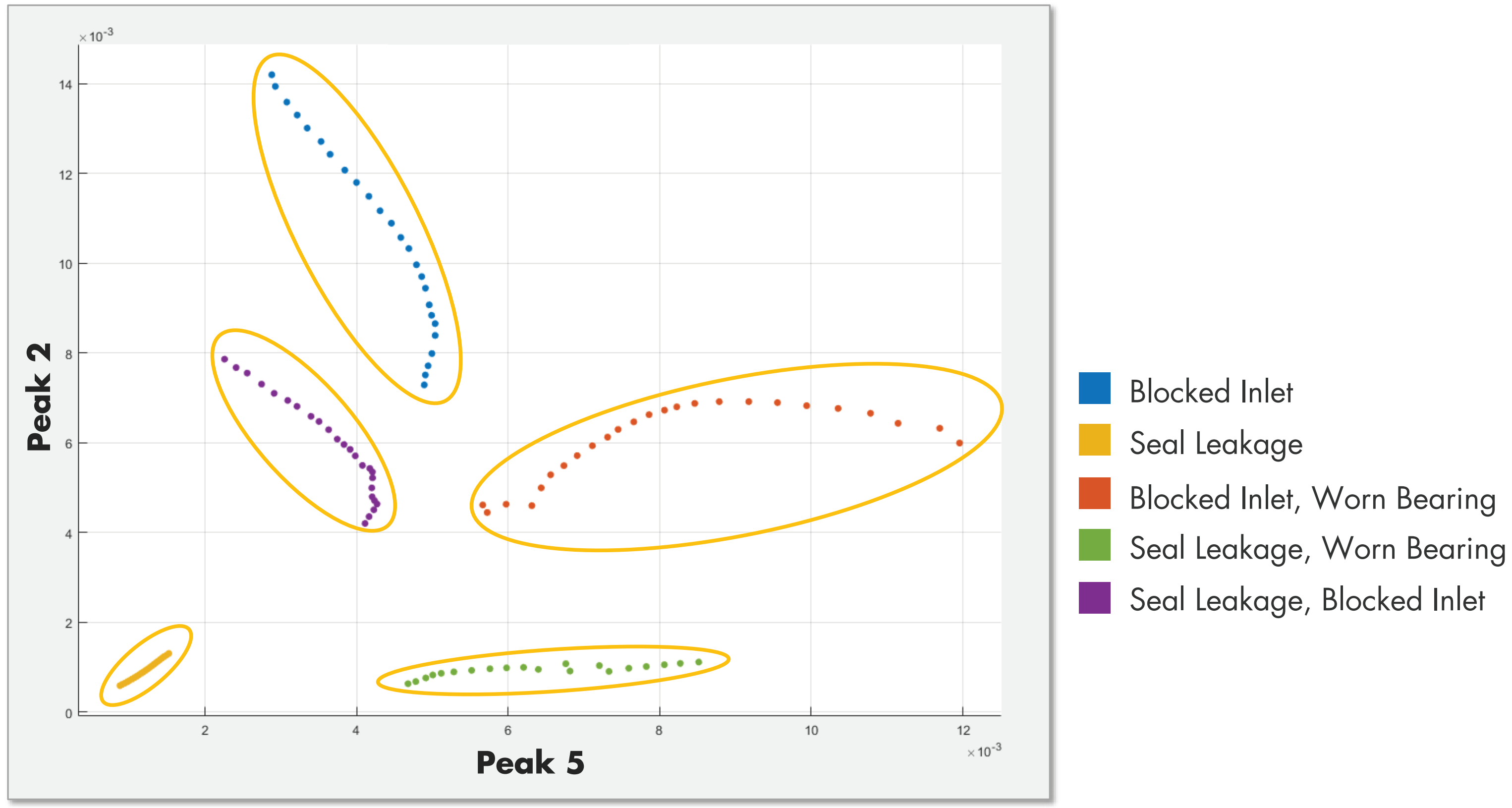
- **Time-domain features:** Mean, variance, skewness, kurtosis
- **Frequency-domain features:** Peaks and peak frequencies



Using Time-Domain Features for Identifying Condition Indicators - Continued

After selecting the frequency-domain features, try to perform an analysis like you did with the time-domain features. The below plot shows the second and fifth peaks with respect to each other.

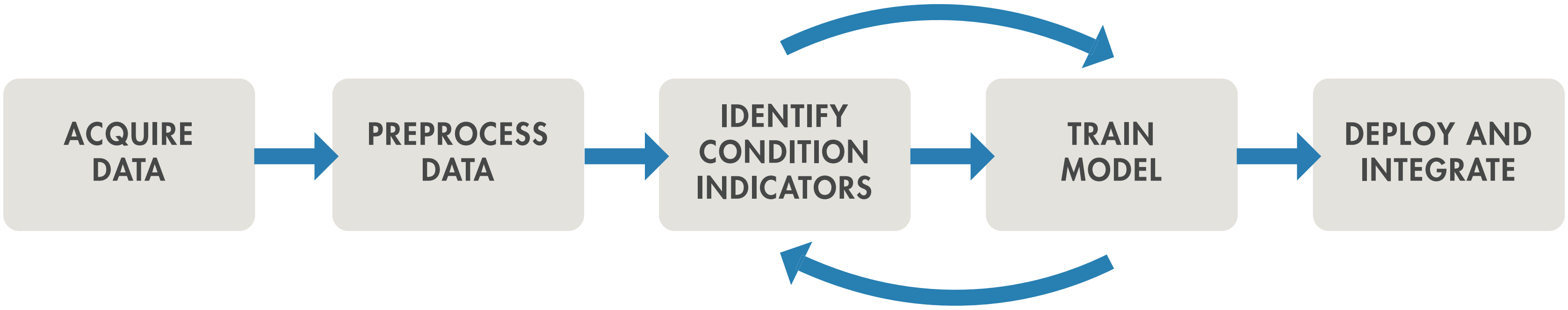
These features effectively separate different groups, which are highlighted with yellow circles. This means that the selected features are distinctive and good candidates for training a machine learning model.



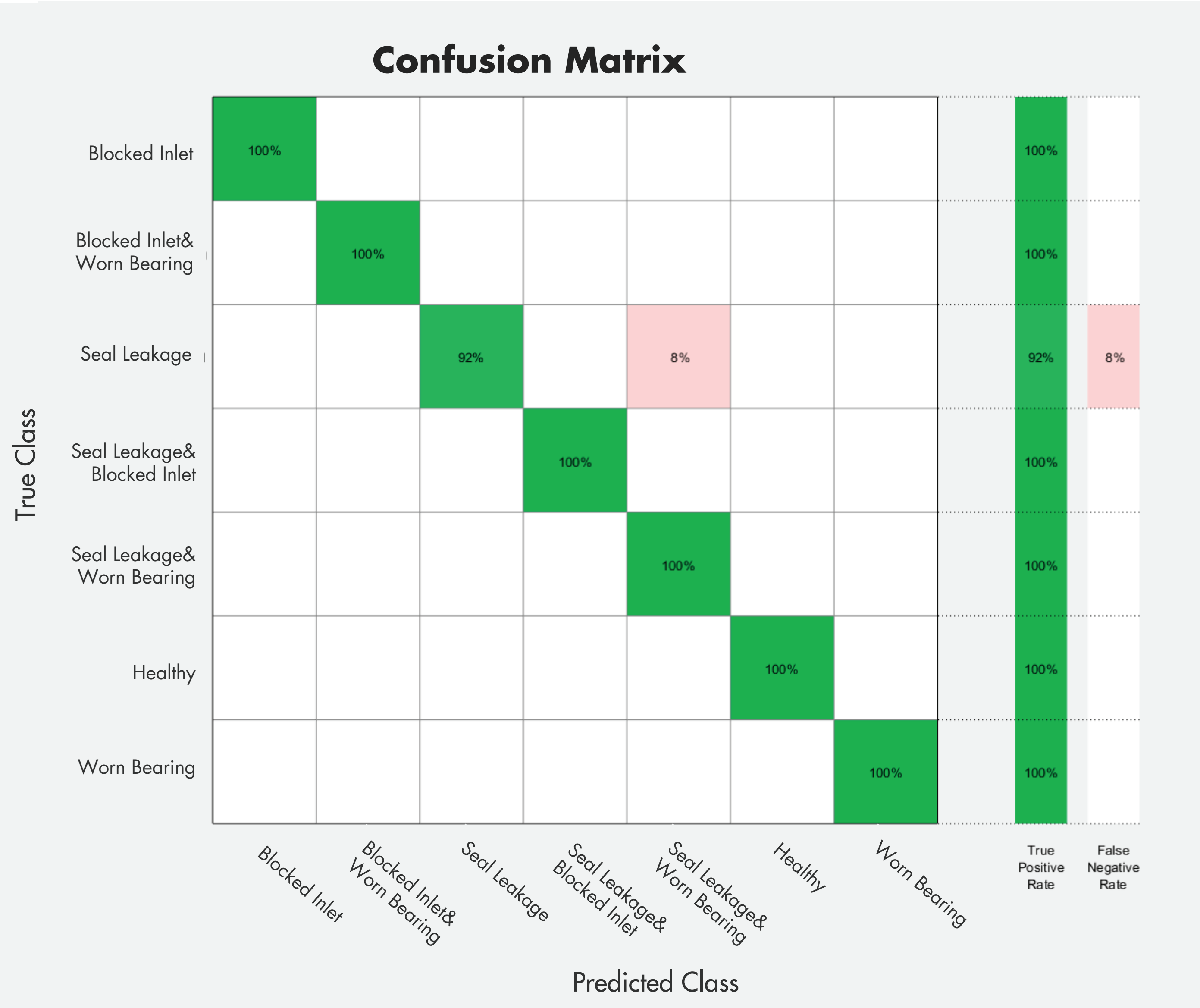
Note that when you're investigating these features, not only are you looking for different clusters, but you also want them to be further away from each other. This makes it easier for the trained models to identify new data points.

The previous page shows plots of the peaks for each of the faults, with the faults in color and the healthy condition in black. Notice that the "worn bearing" plot contains only Peak 3 for the worn bearing and healthy conditions. This is why these conditions don't show up on a plot representing Peaks 2 and 5. This is another reason why we need multiple features to effectively separate the different groups.

Training the Model



After extracting the condition indicators, you can train a *machine learning model* with the extracted features and check the accuracy of the trained model with a confusion matrix. The confusion matrix at right shows the results of one the best-performing classifiers that has been trained with the extracted features. The *Classification Learner app* in MATLAB helps you find the best classifier for your dataset quickly.



The plot shows the true positive rates in green and false negative rates in red.

If you're satisfied with the accuracy of your machine learning model, you can continue with deploying your predictive maintenance algorithm and integrating it into your system. Otherwise, you should revisit the feature extraction step of the predictive maintenance workflow and try training machine learning models with different sets of features, as highlighted with the arrows in the workflow chart.

You may be wondering, how many features are enough to train a machine learning model? Unfortunately, there's no magic number. Just remember that machine learning models can benefit from a high-dimensional set of features that are distinctive and can effectively differentiate fault types.

Learn More

Watch

Predictive Maintenance Tech Talks - Video Series

Predictive Maintenance in MATLAB and Simulink (35:54) - Video

Feature Extraction Using Diagnostic Feature Designer App (4:45) - Video

Read

Overcoming Four Common Obstacles to Predictive Maintenance - White Paper

Explore

Predictive Maintenance with MATLAB - Code Examples

Predictive Maintenance Toolbox - Overview

» Try Predictive Maintenance Toolbox