Machine learning-based rotating machinery fault classification based on vibration time series data

Sylwester Szewczyk

AGH University of Krakow, Faculty of Mechanical Engineering and Robotics, Department of Robotics and Mechatronics

ABB Business Services sp. z o.o, Process Automation Marine&Ports

1. Introduction

Fault detection in rotating machinery with the help of vibration sensors offers the possibility to detect damage to machines at an early stage and to prevent production downtimes by taking appropriate measures. The analysis of vibration data using methods of machine learning promises a significant reduction in the associated analysis effort and a further improvement in diagnostic accuracy. Here is the first draft of the dataset, which can be used to classify anomalies and faults that often occur in rotating machinery. At the end, there are a few examples of the classification methods with code, as well as descriptions of other proposed methods.

2. Dataset description

Dataset is composed of 4044 multivariate time-series acquired by accelerometers. It is the combination of datasets such as COFAULDA and MAFAULDA. In the future there will be additional data added, coming from the laboratory tests performed at ABB lab in Kraków.

Current 4044 measurements comprises three different simulated states: normal function, imbalance fault and horizontal misalignment. This section describes the database.

Below there are examples of time series for normal state of the machine:

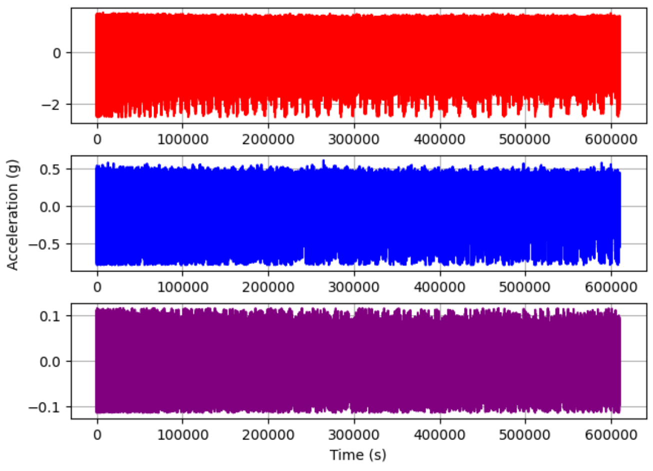

Figure 1. Vibration time series for normal machine state

3. Measurement sequence

Each measurement has the following parameters:

- sampling rate: 50 kHz

- duration: 5 s

- number of samples: 250 000

Below is the summary of measurements divided by machine state:

Machine state	Number of measurements
Normal	1734
Imbalance	1128
Horizontal misalignment	1182

Summary of the variants of faults:

- imbalance

Weight [g]	Number of measurements
20	294
25	282
30	282
35	270

- horizontal misalignment

Misalignment [mm]	Number of measurements
0.5	300
1	294
1.5	294
2	294

4. Dataset structure

Dataset stored in the Parquet file contains features that come from the measurement itself, as well as specifically calculated features derived from the vibration time series that might be used for teaching the classification model. Light versions of the dataset lack time series due to performance issues - although they are available.

Below is a description of the features extracted from the vibration signal time series. These features were computed in both time and frequency domains and are used as input for machine learning models to classify the type of mechanical fault present in the system:

- **fault_type** - A categorical label indicating the operational condition or type of mechanical fault present in the machine. This includes both healthy (normal) operation and various fault classes such as imbalance and misalignment. This column serves as the target variable in the machine learning classification task.
- **variant** - A subcategory or severity level of the given fault_type. It represents specific experimental configurations used to simulate different fault intensities — for example, degrees of shaft misalignment (e.g., 0.5 mm, 1.0 mm) or levels of unbalance (e.g., 6g, 10g, 15g). This feature provides context for fault simulation but is not used directly in the classification model.
- **speed_rpm** - The rotational speed of the machine shaft, measured in revolutions per minute (RPM). This feature represents the operating condition of the machine and can influence vibration characteristics.
- **sensor_name** - Indicates the orientation of the vibration sensor with respect to the machine. Possible values include x, y, and z, representing the horizontal, vertical, and axial directions, respectively. This allows analysis of fault signatures in different spatial axes, which can be critical for identifying direction-specific anomalies.
- **rms** - Root Mean Square (RMS) value of the vibration signal. It reflects the overall energy or power of the signal and is commonly used as an indicator of machine health. Higher RMS values may suggest abnormal behavior or faults.
- **std** - Standard deviation of the vibration signal. It measures the variability or spread of the signal amplitude around its mean. A high standard deviation may indicate unstable or erratic vibrations.
- **peak_to_peak** - The difference between the maximum and minimum values of the vibration signal. It captures the range of amplitude swings and is sensitive to transient shocks or impacts.
- **kurtosis** - A statistical measure of the "tailedness" or peakedness of the vibration signal distribution. High kurtosis values may suggest the presence of impulsive events, such as bearing defects or other sudden mechanical impacts.

- **skewness** - A measure of the asymmetry of the signal distribution. Positive or negative skewness may indicate uneven loading or directional bias in the vibration pattern.
- **fft_amp1** - Amplitude of the first dominant frequency component from the Fast Fourier Transform (FFT) of the vibration signal. Useful for identifying fundamental rotating frequencies associated with imbalance.
- **fft_amp2** - Amplitude of the second dominant frequency component from the Fast Fourier Transform (FFT) of the vibration signal. Useful for identifying fundamental rotating frequencies associated with misalignment.

5. Classification of the machine faults

A. Approach 1: Random Forest and XGboost on extracted timeseries features

To demonstrate the potential for high-end analysis using the described dataset, classification was performed based on a set of calculated features. This set includes the ‘speed_RPM’ values along with various vibration indicators computed for each measurement in the dataset.

Random Forest and XGBoost models were trained on these features. When trained on the entire dataset, both models showed a noticeable drop in prediction accuracy — approximately 82.4% for Random Forest and 83.3% for XGBoost.

These results serve as a baseline and illustrate the capabilities of the vibration dataset. The models can be further refined and extended in future work.

Random Forest:

```
df = pd.read_parquet("FaultDatabase.parquet")

features = ['speed_rpm', 'rms', 'std', 'peak_to_peak', 'kurtosis', 'skewness',
'fft_amp1', 'fft_amp2']
target = 'fault_type'

for col in features + [target]:
    if col not in df.columns:
        raise ValueError(f"No column: {col}")

X = df[features]
y = df[target]

X_train_val, X_test, y_train_val, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)

X_train, X_val, y_train, y_val = train_test_split(
```

```python
    X_train_val, y_train_val, test_size=0.25, random_state=42,
stratify=y_train_val)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']
}

rf = RandomForestClassifier(random_state=42)
search = RandomizedSearchCV(rf, param_grid, n_iter=30, cv=3,
                            scoring='accuracy', random_state=42, n_jobs=-1,
verbose=1)
search.fit(X_train_scaled, y_train)
best_rf = search.best_estimator_

for split, X_s, y_s in [('Walidacja', X_val_scaled, y_val), ('Test',
X_test_scaled, y_test)]:
    y_pred = best_rf.predict(X_s)

y_test_pred = best_rf.predict(X_test_scaled)
```

XGBoost:

```python
df = pd.read_parquet("FaultDatabase.parquet")

features = ['speed_rpm', 'rms', 'std', 'peak_to_peak', 'kurtosis', 'skewness',
'fft_amp1', 'fft_amp2']
target_col = 'fault_type'

X = df[features]
y_raw = df[target_col]

le = LabelEncoder()
y = le.fit_transform(y_raw)

for i, label in enumerate(le.classes_):
    print(f"{i}: {label}")

X_train_val, X_test, y_train_val, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42)
```

```
X_train, X_val, y_train, y_val = train_test_split(
    X_train_val, y_train_val, test_size=0.25, stratify=y_train_val,
random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

xgb = XGBClassifier(
    objective='multi:softmax',
    use_label_encoder=False,
    eval_metric='mlogloss',
    random_state=42
)

param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'max_depth': [3, 5, 7, 10],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0]
}

search = RandomizedSearchCV(
    xgb,
    param_distributions=param_grid,
    n_iter=25,
    cv=3,
    scoring='accuracy',
    n_jobs=-1,
    verbose=1,
    random_state=42,
    error_score='raise'
)

search.fit(X_train_scaled, y_train)
best_model = search.best_estimator_

y_test_pred = best_model.predict(X_test_scaled)
y_test_labels = le.inverse_transform(y_test_pred)
```

B. Approach 3: Convolutional Neural Network on raw vibration time series

Convolutional Neural Networks (CNNs) are able to recognize patterns in data and to perform classification tasks based on these recognized patterns. An unbalance classification with CNNs, which receive the windowed data directly as input, is therefore promising. The advantage here is that no further data preprocessing is necessary and the effort involved in creating the algorithm is therefore comparatively low.

6. Summary and outlook

This work presents a vibration dataset consisting of time series data collected from rotating machinery under various fault conditions. The dataset includes both raw signals and calculated features, enabling its use in a wide range of machine learning applications related to fault detection and diagnostics.

To illustrate its potential, basic classification models (Random Forest and XGBoost) were trained on a selected feature set. While these models already achieve reasonable accuracy, they primarily serve as an entry point for further exploration. The accompanying code can be extended and adapted for testing more advanced classification or prediction techniques.

The dataset, together with the provided baseline models, offers a valuable foundation for developing and benchmarking data-driven approaches in condition monitoring of rotating machinery.