Anomaly detection Updates - 01

TEAM MEMBERS:

NITHIN M KANNAL
PADMAPRIYA R
AMOGGHA C H

MENTOR: DR A.AJINA

Overview of Goals

1. DATA COLLECTION AND PREPROCESSING
2. PATTERN ANALYSIS OF NORMAL BEHAVIOR
3. INTRODUCTION TO BASELINE MODELS
4. TRAINING ON DATA PREPROCESSING TECHNIQUES AND INITIAL DATA ANALYSIS
5. INTRODUCTION TO STATISTICAL METHODS FOR PATTERN ANALYSIS
6. FAMILIARIZATION WITH BASIC ANOMALY DETECTION ALGORITHMS

1

Goal #1

Data Collection and Preprocessing

WHERE WE COLLECTED THE DATA

- https://data.mendeley.com/datasets/ztmf3m7h5x/6

	bearingA_x	bearingA_y	bearingB_x	bearingB_y
0	-0.138363	0.028935	-0.019773	-0.002564
1	-0.101087	0.012587	-0.005409	0.015794
2	-0.105067	-0.003972	0.027830	0.026102
3	-0.181645	0.080939	-0.012655	0.050237
4	-0.153244	0.031137	-0.042393	0.046297

DATA DESCRIPTION

- This dataset presents time-series dataset including vibration, data for rotating machines under varying load conditions. The conditions of the rotating machine include normal, bearing inner race faults, bearing outer race faults, shaft misalignment, and rotor unbalance with three different load conditions.
- we have taken the vibration data zip file for the preprocessing.
- the model has been trained with the normal data and tested on bearing inner race faults data
- the dataset here is almost cleaned.

PREPROCESSING

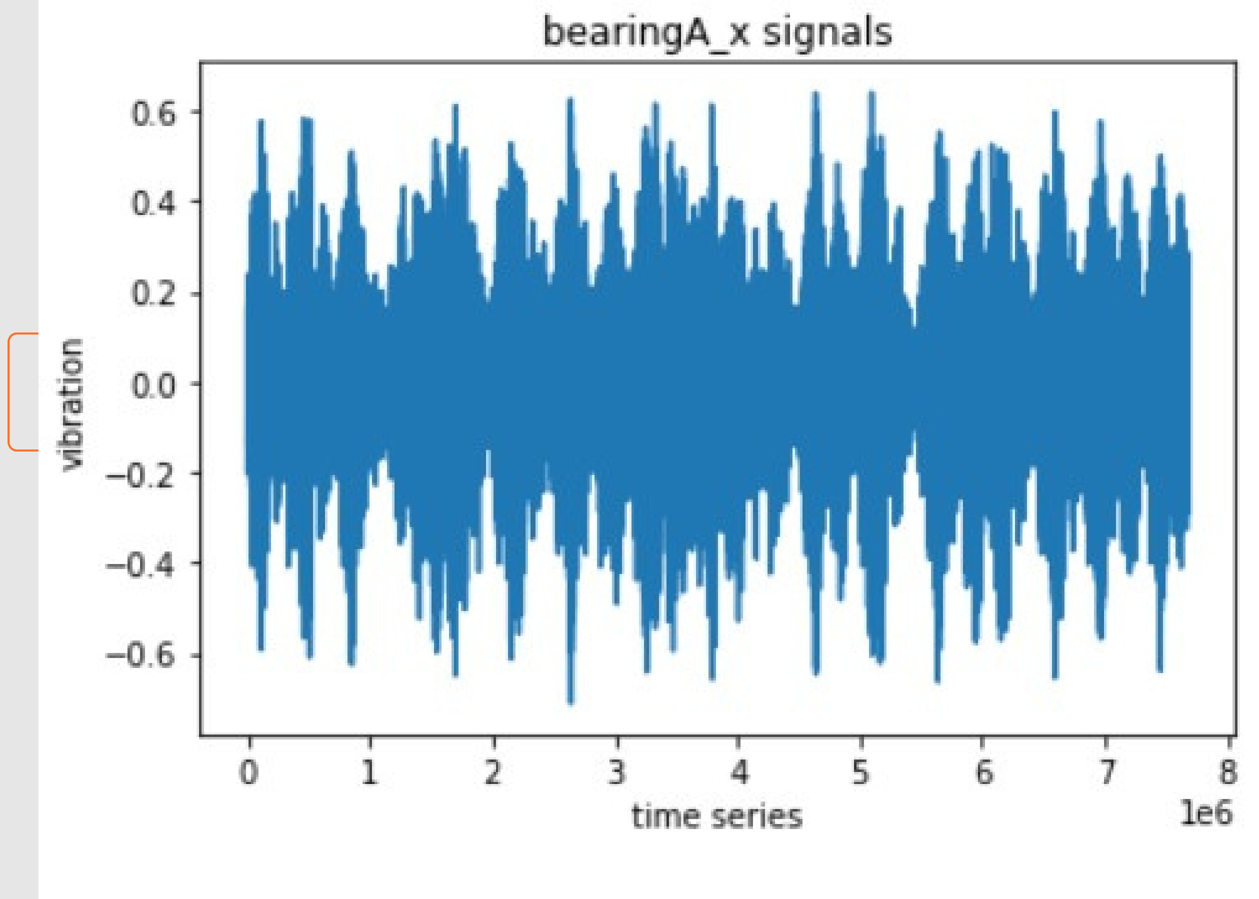
- Handling Missing Values:
  - we found no instances of missing values. This indicated that the dataset was complete, with all necessary information available for analysis. As a result, no imputation or removal of missing values was required.
- Detrending:
  - Upon examination, it was determined that the dataset had already undergone detrending, resulting in a mean of 0. This ensured that any inherent trends within the data were effectively mitigated, laying a solid foundation for subsequent analyses.
- Standardization:
  - To facilitate uniformity and enhance model performance, standard scalarization was applied to scale the features within the dataset. By transforming the features to a common scale ranging from 0 to 1, we ensured consistency in feature scales, which is essential for model convergence and overall predictive accuracy.

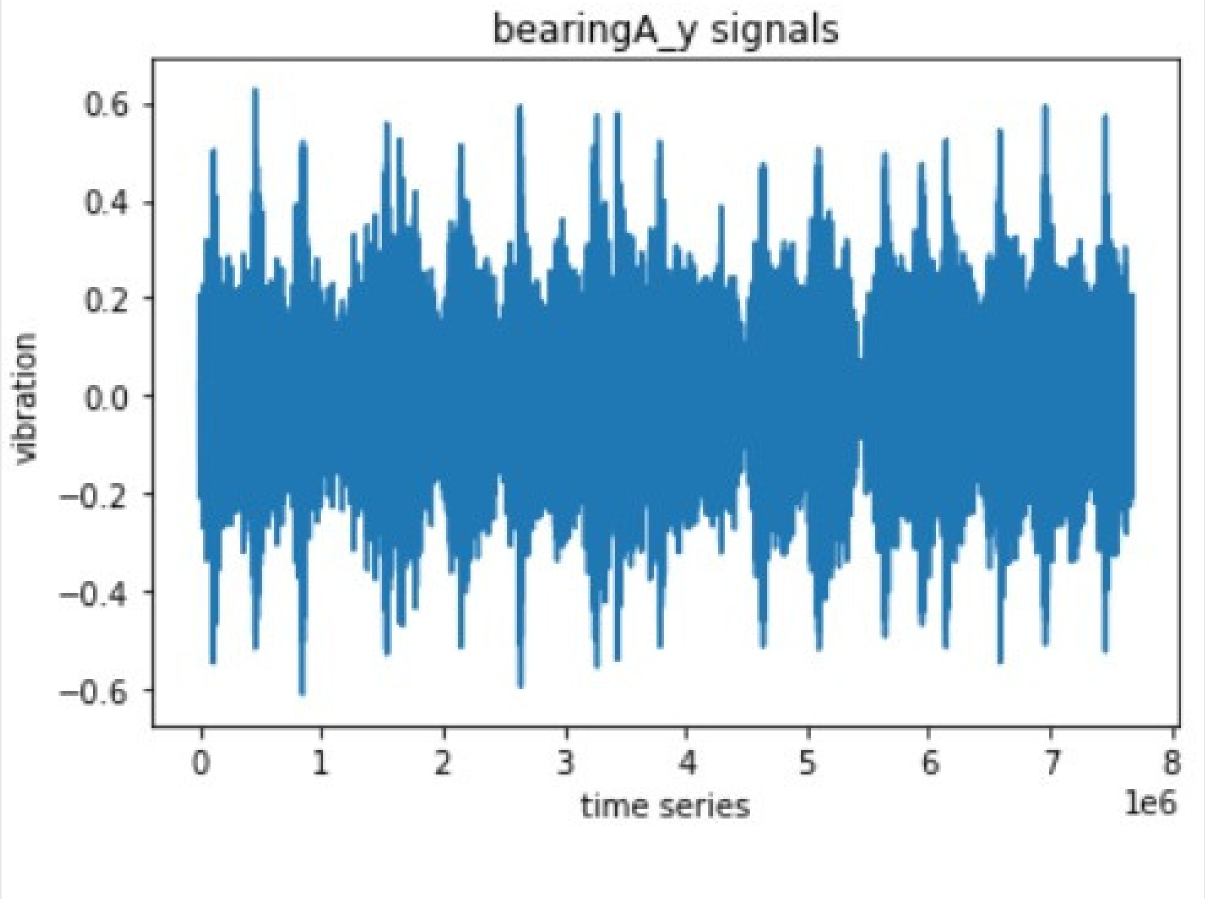
2

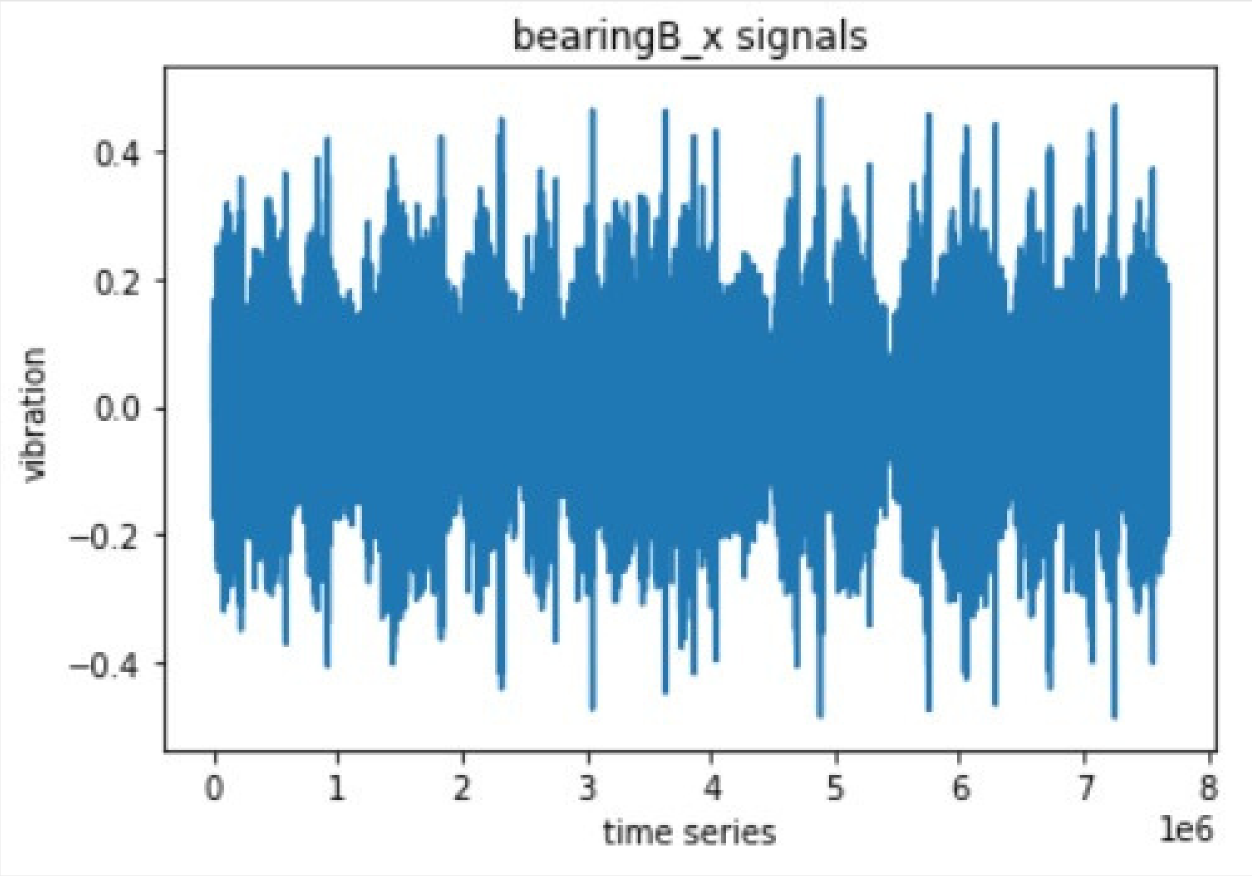
Goal #2

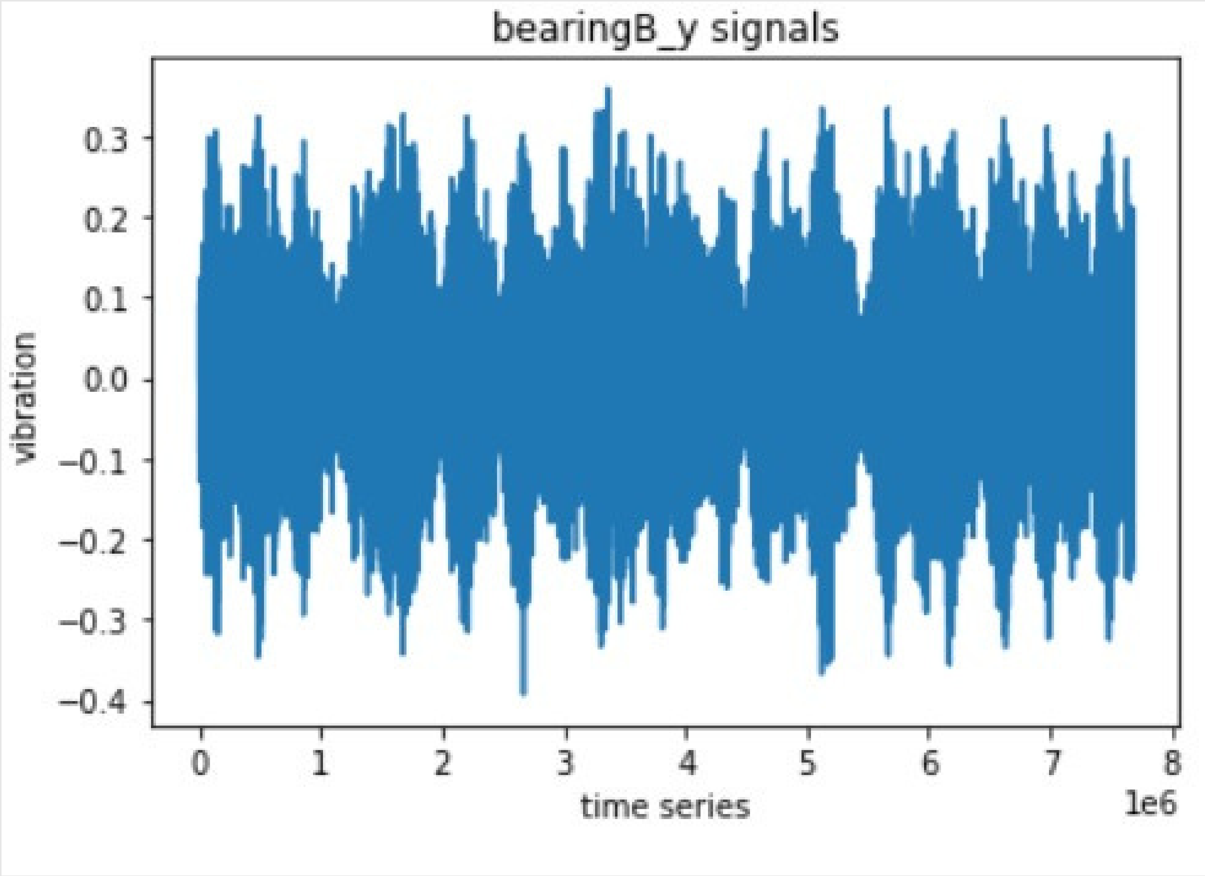
Pattern Analysis of Normal Behavior

VISUAL REPRESENTATION OF THE NORMAL VIBRATIONAL SIGNALS



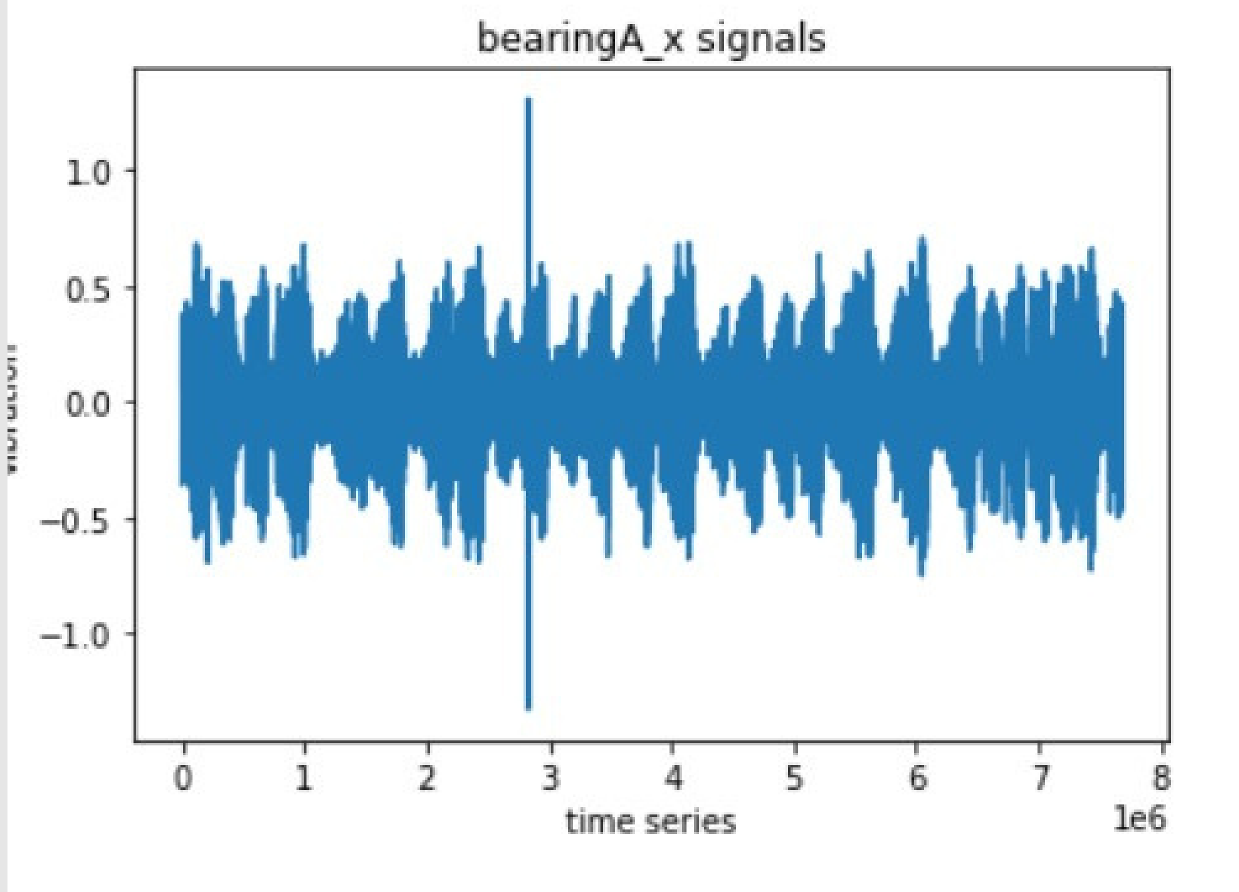


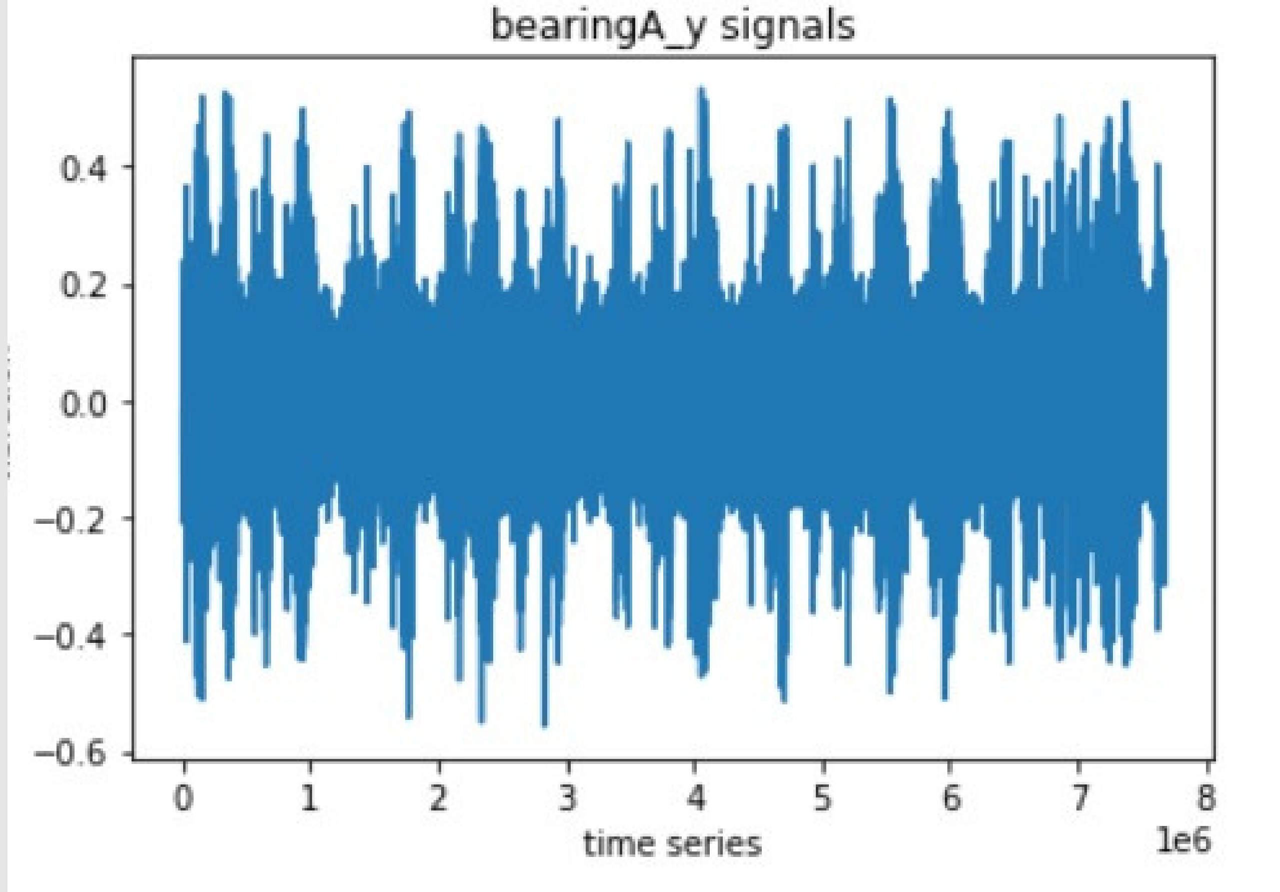


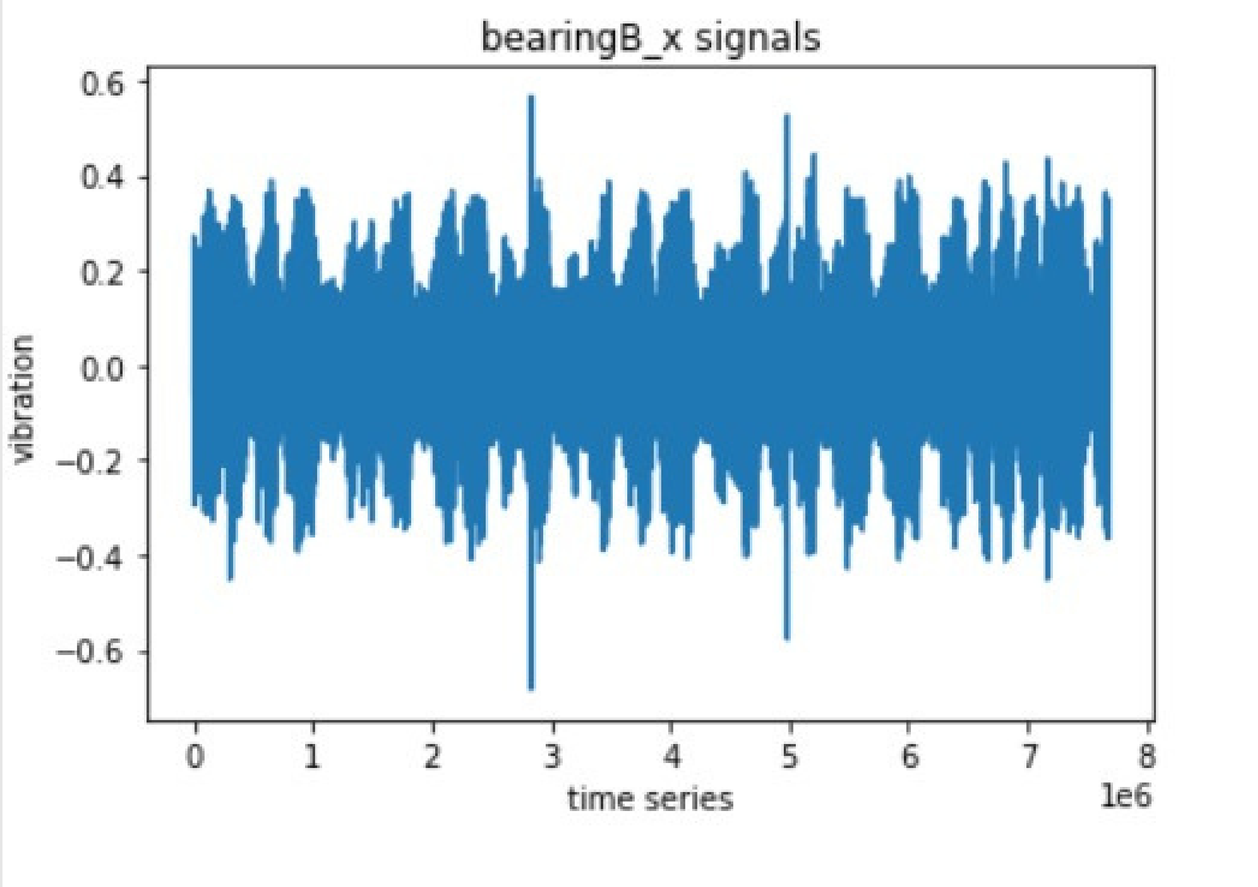


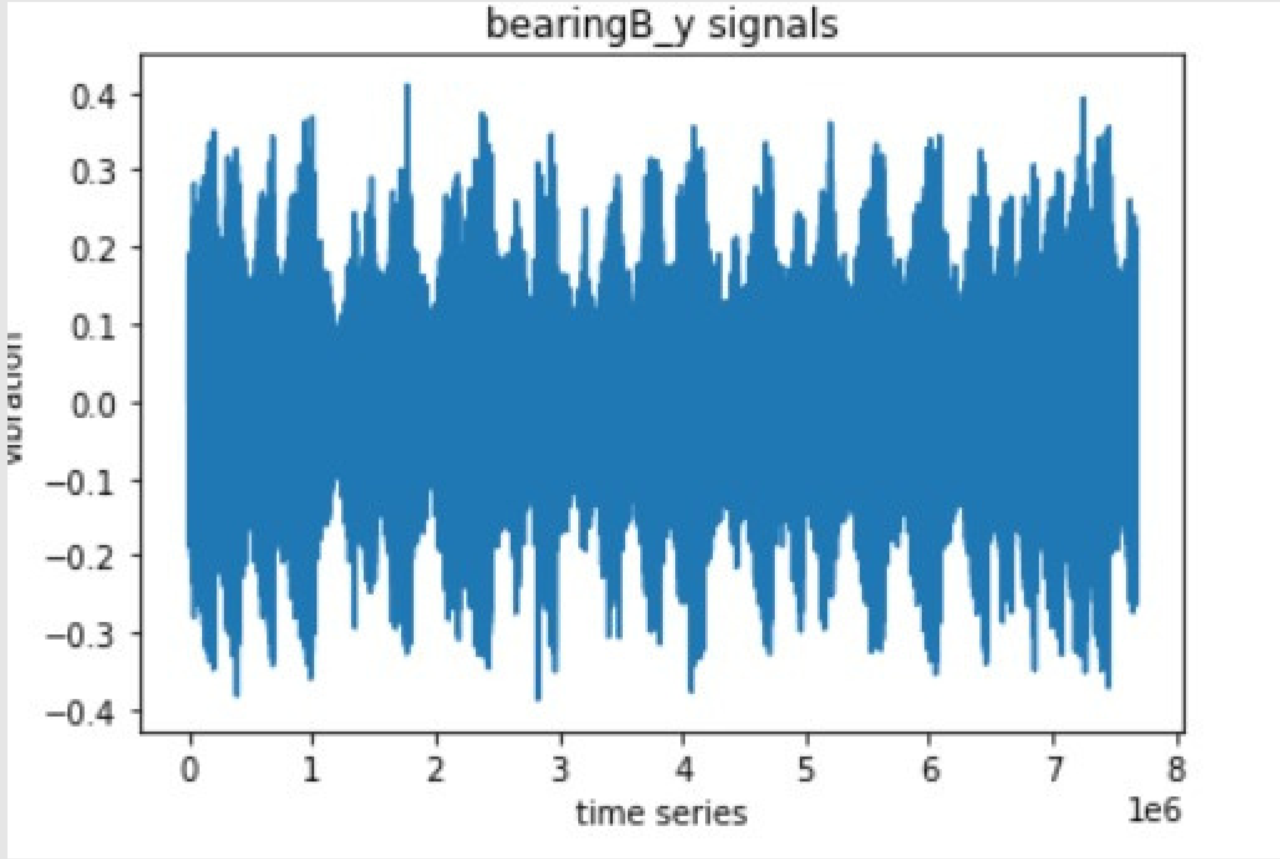
Pattern Analysis of Abnormal Behavior

VISUAL REPRESENTATION OF THE ABNORMAL VIBRATIONAL SIGNALS









3

Goal #3

Introduction to Baseline Models

HOW AUTO ENCODERS WORK

- An autoencoder is a type of neural network designed to learn efficient representations of input data, often used for dimensionality reduction or feature learning. In the context of anomaly detection, autoencoders are trained to reconstruct normal data accurately while producing high errors for anomalous instances.

MODEL ARCHITECTURE

- The autoencoder model consists of an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation (latent space), while the decoder reconstructs the original input from this representation.
- In the provided code, the encoder and decoder are implemented using Long Short-Term Memory (LSTM) layers, which are well-suited for sequential data like time series.
- The model is trained to minimize the Mean Absolute Error (MAE) between the input and the reconstructed output.

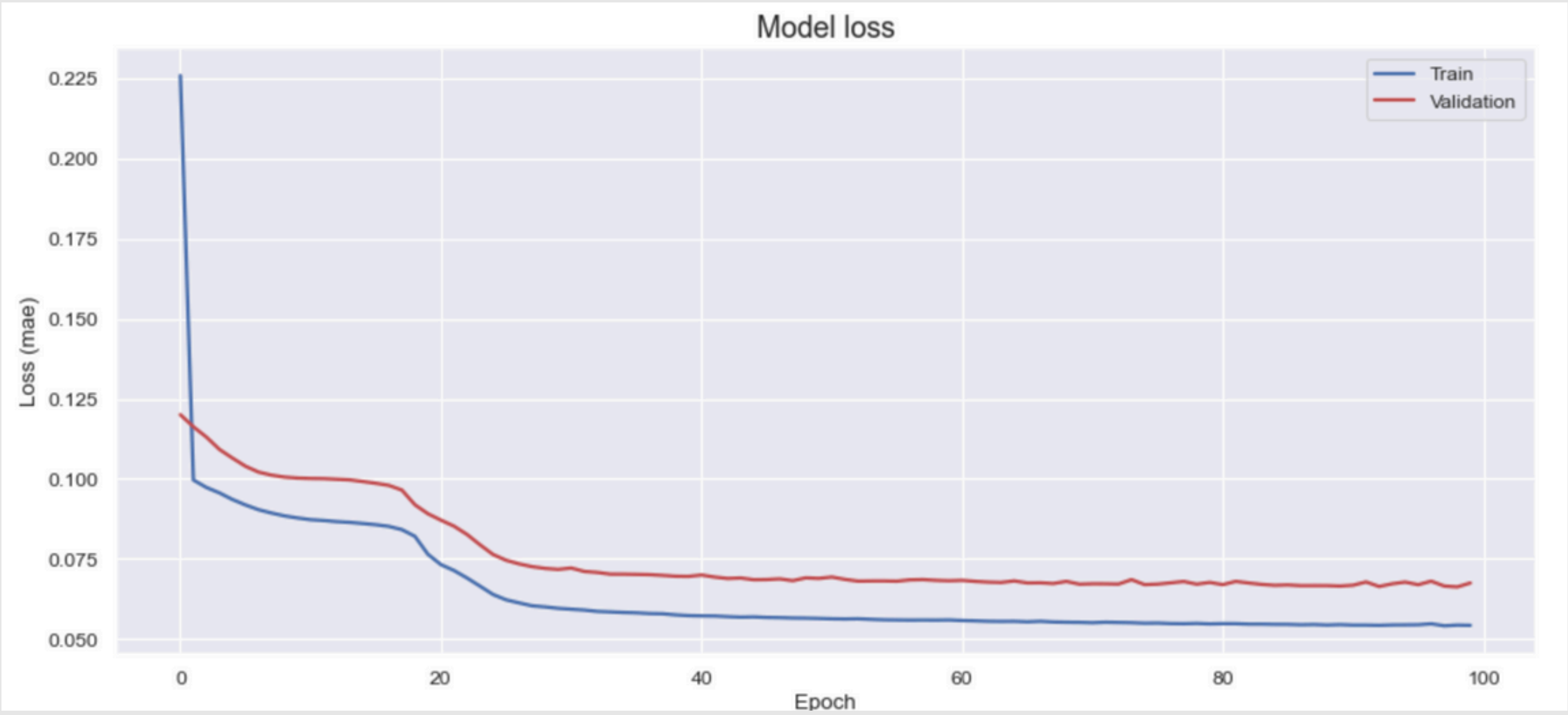
TRAINING PROCESS

- During training, normal data samples are fed into the autoencoder, and the model learns to reconstruct them accurately with low error.
- An anomaly detection threshold is established based on the reconstruction errors of the training data. Typically, this threshold is set as a certain number of standard deviations above the mean reconstruction error.
- Instances with reconstruction errors exceeding the threshold are flagged as anomalies.

In summary, while autoencoders offer promising capabilities for anomaly detection, their adoption in defense ship sensor systems requires careful consideration of computational constraints, real-time processing requirements, and model generalization challenges.

Results and Discussion of Autoencoders model

MODEL LOSS




```
Calculated Threshold: 0.14274695546493468
```

[22]:
```
X_pred = model.predict(X_test)
X_pred = X_pred.reshape(X_pred.shape[0], X_pred.shape[2])
X_pred = pd.DataFrame(X_pred, columns=test.columns)
X_pred.index = test.index

scored = pd.DataFrame(index=test.index)
Xtest = X_test.reshape(X_test.shape[0], X_test.shape[2])
scored['Loss_mae'] = np.mean(np.abs(X_pred-Xtest), axis = 1)
scored['Threshold'] = threshold
scored['Anomaly'] = scored['Loss_mae'] > scored['Threshold']
scored.head()
```

```
47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step
```

[22]:

	Loss_mae	Threshold	Anomaly
3001	0.032407	0.142747	False
3002	0.070658	0.142747	False
3003	0.021788	0.142747	False
3004	0.044782	0.142747	False
3005	0.054804	0.142747	False

TEST OUTPUT ON NORMAL DATASET

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
Row 1:
Predicted Loss (MAE): [2.76528462]
Threshold: 0.14274695546493468
Anomaly: [ True]

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
Row 2:
Predicted Loss (MAE): [3.31512994]
Threshold: 0.14274695546493468
Anomaly: [ True]

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
Row 3:
Predicted Loss (MAE): [5.06467219]
Threshold: 0.14274695546493468
Anomaly: [ True]
```

TEST OUTPUT ON ANOMALY DATASET

4

Goal #4

Training on data preprocessing techniques and initial data analysis

We focus on training machine learning models to identify potential abnormalities and failures associated with misalignments in machinery. This approach enhances the reliability and operational efficiency of industrial equipment. The visuals attached here shows the behavior of signals where the mean taken to be 0. The threshold for the dataset has to found using the methods like autoencoder and other statistical which can be taken to any of the dataset. The feature shown after after the visualization and stat calculation are

- Spectral Analysis: Breaking down the vibration signals into their frequency components to identify dominant frequencies and their amplitudes. Changes or anomalies in the spectral distribution can highlight issues like misalignments or unbalanced components.
- Interquartile Range (IQR): This helps identify the typical range of data and is useful for setting thresholds for anomaly detection.
- Outliers: Points outside the typical range can indicate anomalies. Persistent outliers or a shift in the number of outliers can be a sign of developing faults.

Preprocessing techniques : Standard scalarization that takes all the values from the defined range which is -1 to 1 here. The general model handles all other cleaning like missing values and detrending .

Model Summary

```
 Layer (type)                Output Shape
=================================================
 dense (Dense)               (None, 32)

 dense_1 (Dense)             (None, 16)

 dense_2 (Dense)             (None, 8)

 dense_3 (Dense)             (None, 16)

 dense_4 (Dense)             (None, 32)

 dense_5 (Dense)             (None, 4)

=================================================
Total params: 1,644
Trainable params: 1,644
```

5

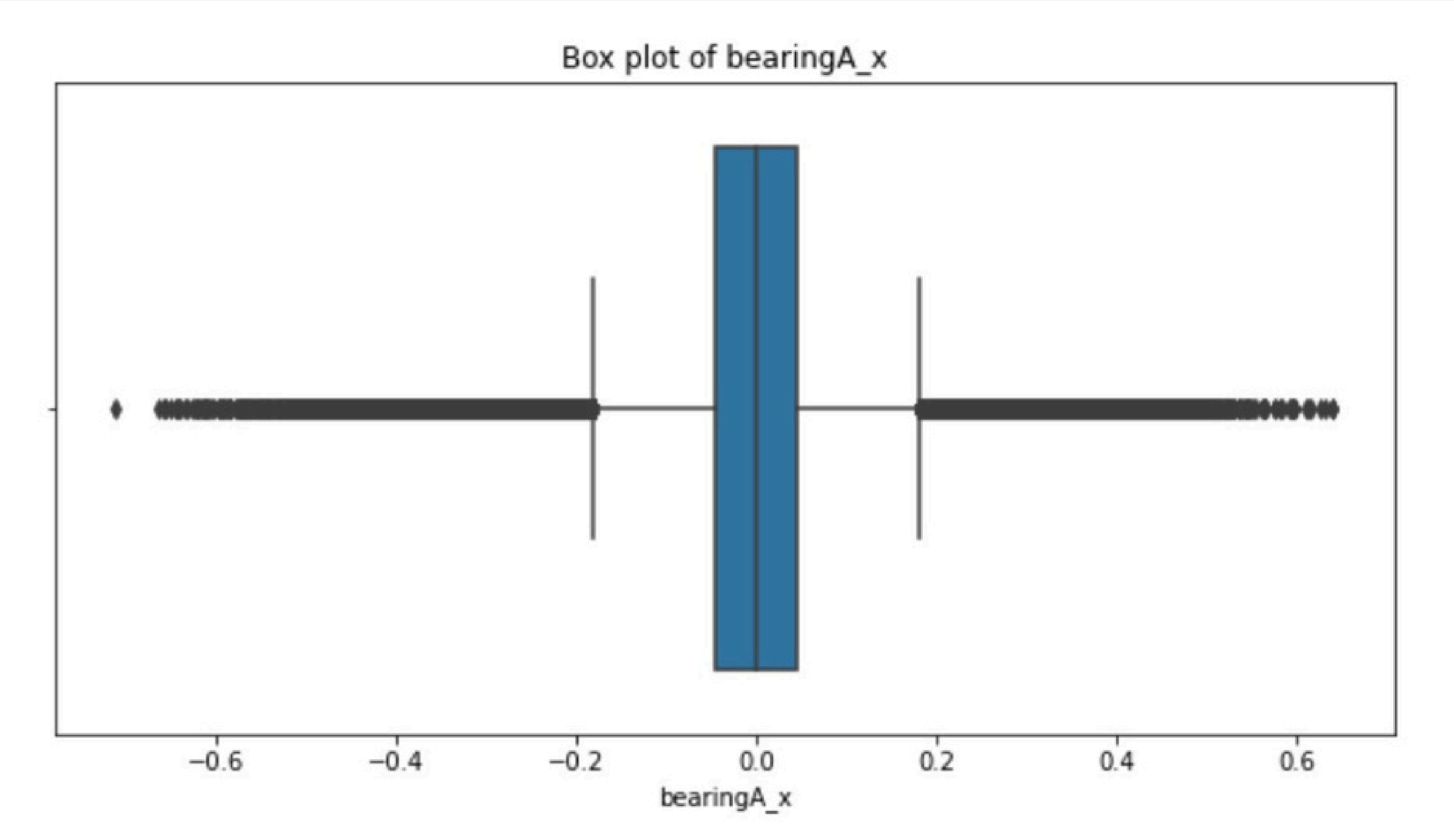
Goal #5

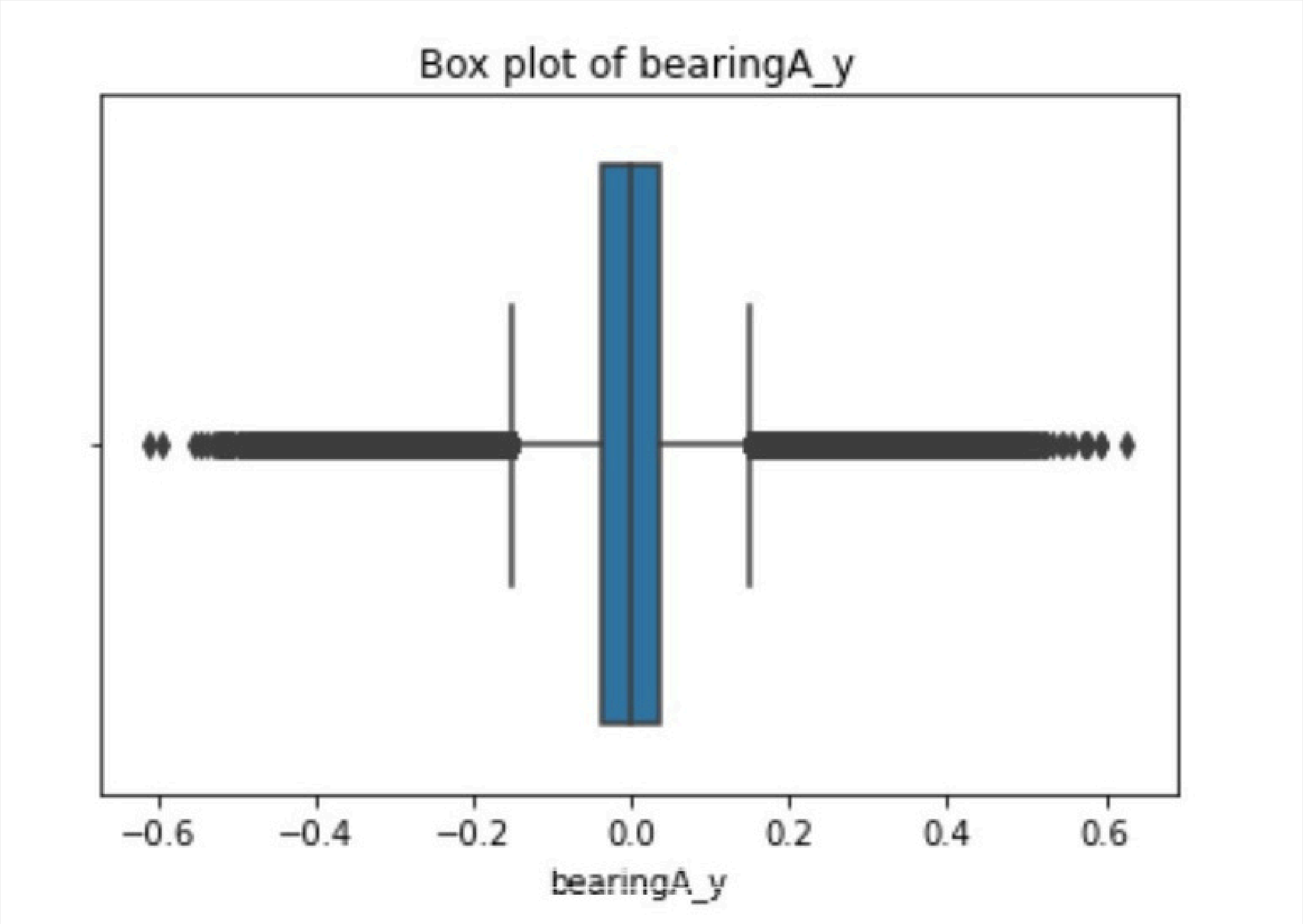
Introduction to statistical methods for pattern analysis

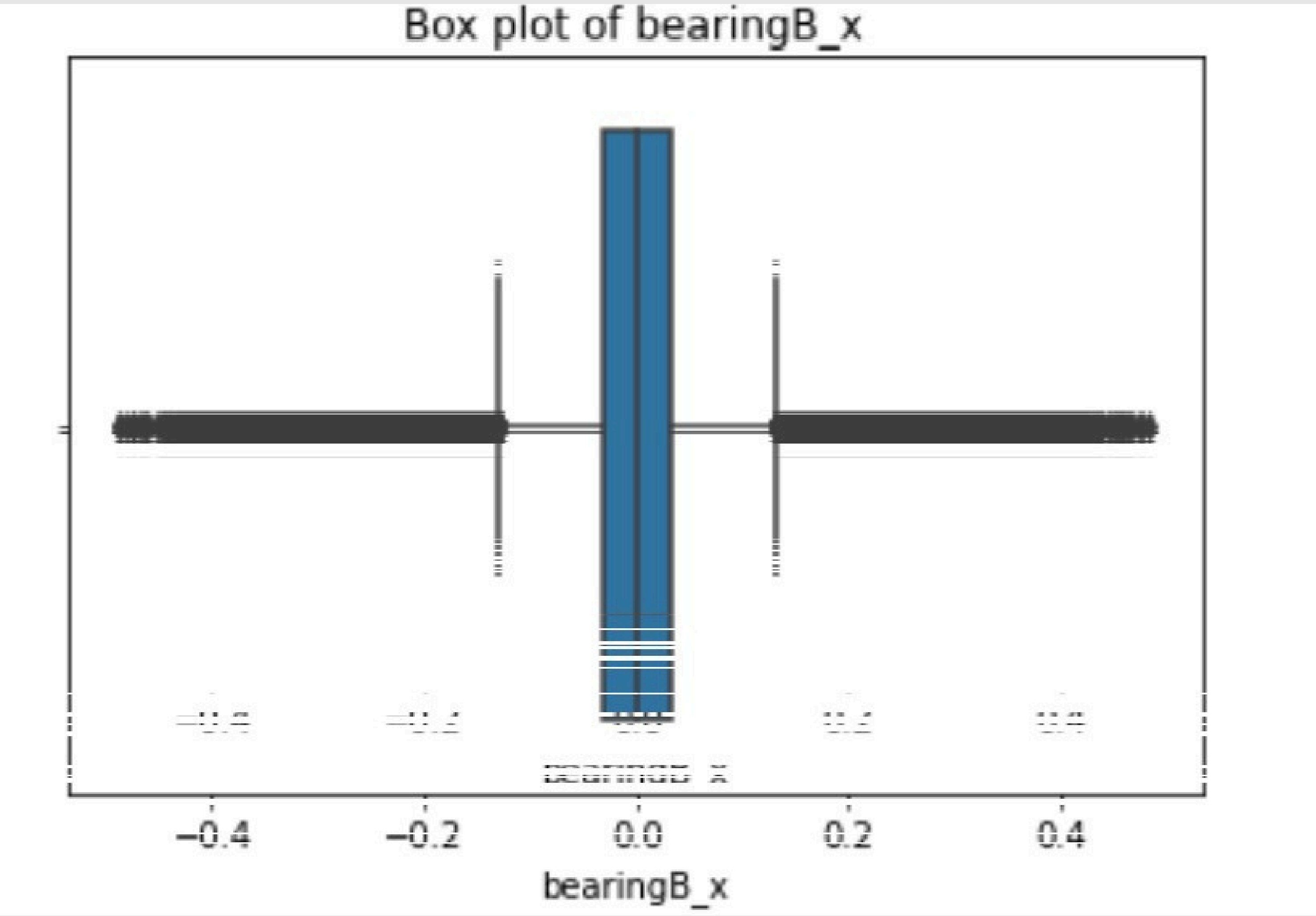
Statistical Insights into Dataset Characteristics

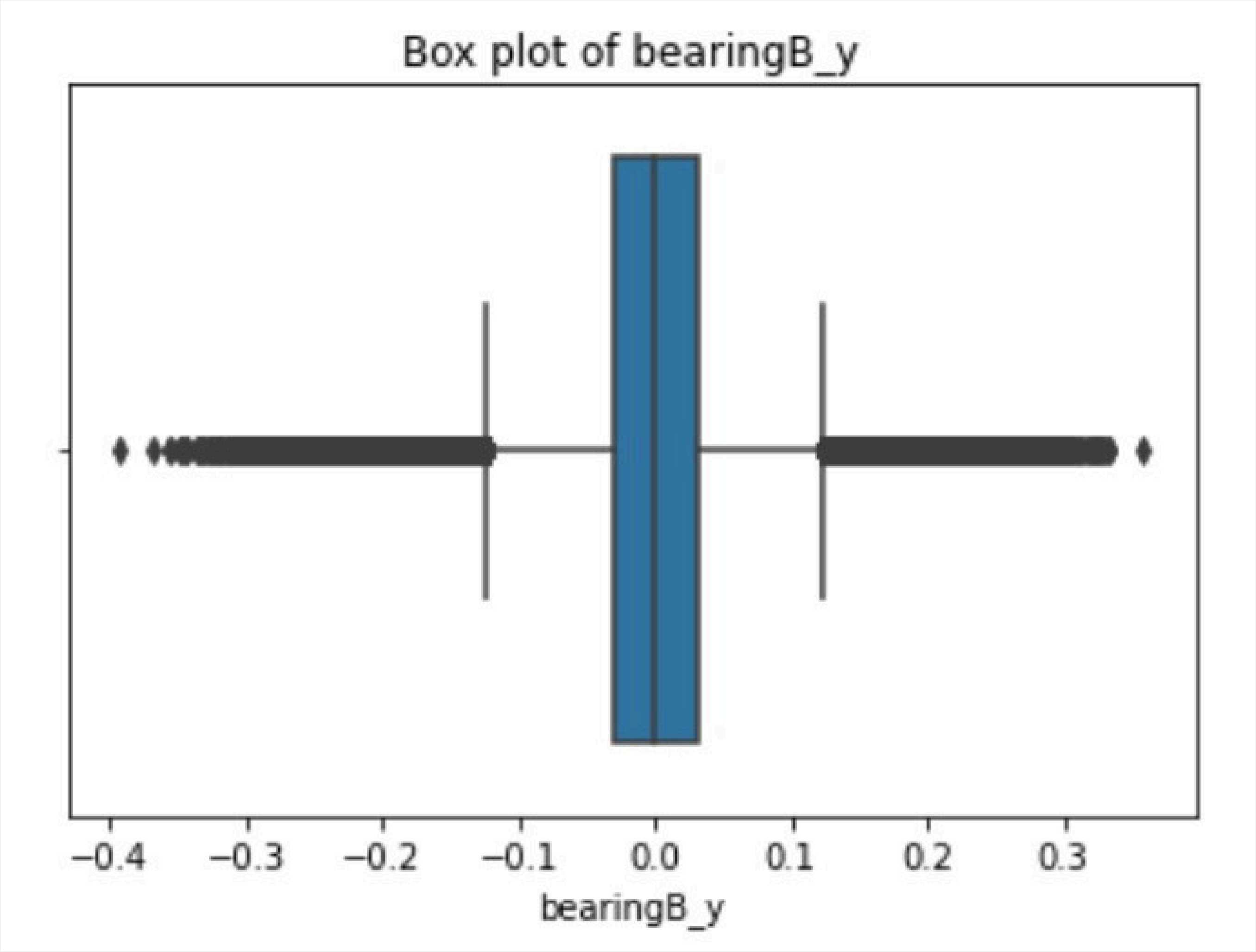
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7680000 entries, 0 to 7679999
Data columns (total 4 columns):
 #   Column      Dtype
---  ------      -----
 0   bearingA_x  float64
 1   bearingA_y  float64
 2   bearingB_x  float64
 3   bearingB_y  float64
dtypes: float64(4)
memory usage: 234.4 MB
None
   bearingA_x  bearingA_y  bearingB_x  bearingB_y
0   -0.138363    0.028935   -0.019773   -0.002564
1   -0.101087    0.012587   -0.005409    0.015794
2   -0.105067   -0.003972    0.027830    0.026102
3   -0.181645    0.080939   -0.012655    0.050237
4   -0.153244    0.031137   -0.042393    0.046297
         bearingA_x    bearingA_y    bearingB_x    bearingB_y
count  7.680000e+06  7.680000e+06  7.680000e+06  7.680000e+06
mean  -7.079860e-06 -8.365586e-06  8.810208e-07 -1.785907e-07
std    8.704768e-02  6.525712e-02  5.650941e-02  5.175903e-02
min   -7.118620e-01 -6.112134e-01 -4.879654e-01 -3.934964e-01
25%   -4.540956e-02 -3.751190e-02 -3.257719e-02 -3.099364e-02
50%   -6.664322e-05 -1.236540e-04 -1.831933e-05 -9.237764e-06
75%    4.529565e-02  3.749749e-02  3.263552e-02  3.112439e-02
max    6.399783e-01  6.272295e-01  4.837299e-01  3.587767e-01
```

Box plot visualization for normal data

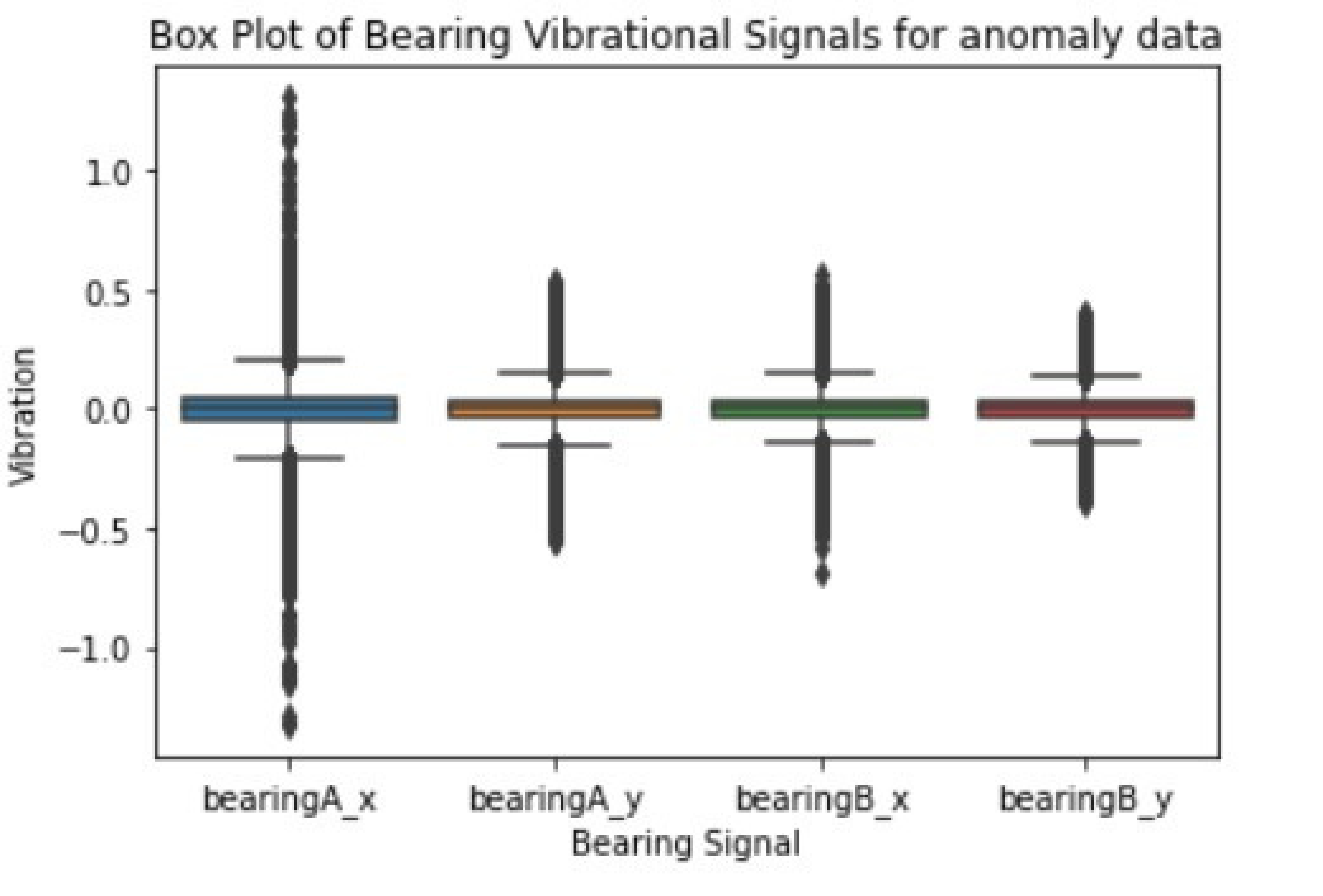








Box plot visualization for anomaly data



6

Goal #6

Familiarization with basic anomaly detection algorithms

Listed down the following anomaly detection algorithms:

- LSTM (Long Short-Term Memory) Networks
- PCA (Principal Component Analysis)
- Generative Adversarial Networks (GANs)
- Autoencoders
- Variational Autoencoders (VAEs)
- SVM (Support Vector Machines) with LOF (Local Outlier Factor)
- One-Class SVM
- ADTK (Anomaly Detection Toolkit)
- Restricted Boltzmann Machines (RBMs)
- GANs with Binary Cross-Entropy Loss
- LSTM with Autoencoder using Mean Absolute Error (MAE) Loss

Autoencoder Implementation:

Specifically delved into the implementation and experimentation with autoencoder-based anomaly detection method. Autoencoders were chosen for their ability to reconstruct normal data patterns and highlight deviations, making them a popular choice for anomaly detection tasks.

Thank you!