NOVELTY DETECTION ON STREAMING SENSOR DATA FOR IIOT APPLICATIONS

by

Alper Bayram

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2019

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep appreciation for my advisor Prof. Burak Acar. He was always encouraging and supportive throughout my entire research. Even when I felt lost at times, he was always ready to share his time, scientific prominence and wisdom. Without his guidance, it would not be possible to complete this work.

Secondly, I would like to thank Gürkan Aydemir for his assistance, exchange of ideas and being a wonderful colleague throughout the project. His comments, expertise and ideas helped me a lot in these two years. I also extend my sincere gratitude to the people who ignited this project. Deniz Eroğlu and Asst. Prof. Kutluhan Erol always had insightful comments and suggestions with their industrial background.

Without the support of my friends I wouldn't bear the overwhelming periods of life. I would like to thank all of them for always being there whenever I needed their companion.

Lastly, I would like to thank my parents for their endless love and support. Sometimes it doesn't feel right to be loved that unrequitedly. I also thank my sister for having bought *Smash* album. When I listened to it years later, it definitely started series of reactions that shape not only my music taste, but my persona as well.

ABSTRACT

NOVELTY DETECTION ON STREAMING SENSOR DATA FOR IIOT APPLICATIONS

Assessment of present and future condition of industrial machinery is one of the core ideas that constitute Industry 4.0 paradigm. Predictive maintenance depends on integrated sensors and machine learning algorithms to achieve this assessment based on the internal parameters of machinery. This type of maintenance could save plant costs and improve efficiency while reducing fatal defects in machinery. It automates the maintenance process and reduces the number of periodic checks.

Bearings are used in rotating machinery extensively. However, bearing faults are common and could cause time and financial loss if they occur unexpectedly. Machine learning could be used in predictive maintenance framework to predict the health status of a bearing. Bearing fault analysis research has been traditionally conducted on its vibration signature. Due to nature of data, each bearing should be modelled separately or machine learning algorithms should be robust against environment or different machinery settings.

In the present work unsupervised novelty detection framework on streaming vibration data is proposed. The framework is built in an unsupervised manner since each bearing is considered individually and building models for each of them is impractical. Since faulty samples are not available initially, novelty detection methods are applied on bearing degradation data. The results show that detection of bearing faults and other state changes can be made using novelty detection methods. Detection could be achieved earlier than conventional methods for some cases.

ÖZET

ENDÜSTRİYEL NESNELERİN İNTERNETİ UYGULAMALARI İÇİN AKAN SENSÖR VERİSİ ÜZERİNDE AYRILIK SEZİMİ

Endüstriyel makinelerin anlık ve gelecekteki durumlarını değerlendirmek Endüstri 4.0 paradigmasının ana yapı taşlarından birisidir. Öngörücü bakım, makinelerin kendi parametrelerini göz önüne alır ve tümleşik sensörler ile makine öğrenmesi algoritmalarının yardımıyla da bu değerlendirmeyi gerçekleştirir. Bu şekilde yapılan bakım makinelerdeki geri dönülemez hataların önüne geçerek, fabrika giderlerini azaltır ve verimliliği arttırır. Bakım süreci otomatikleştirilerek rutin kontrollerin sayısı azaltılır.

Rulmanlar dönen makinelerde yaygın olarak kullanılmaktadır. Fakat rulman hataları sıklıkla görülmekte, zaman kayıplarına ve maddi kayıplara yol açmaktadır. Rulmanların hasar durumlarını tahmin edebilmek için öngörücü bakım çerçevesinde makine öğrenmesi teknikleri kullanılabilir. Rulman hata analizi araştırmaları geleneksel olarak titreşim ölçümleri üzerinde yapılmıştır. Verinin doğası gereği her bir rulman ayrı modellenmeli ya da makine öğrenmesi algoritmaları farklı ortamlarda ve makine ayarlarında çalışabilir olmalıdır.

Bu çalışmada akan titreşim verisi üzerinde gözetimsiz ayrılık sezimi yapısı önerilmiştir. Yapı, her bir rulman ayrı olarak düşünüldüğü ve hepsini modellemek kullanışsız olacağı için, gözetimsiz çalışabilecek şekilde kurulmuştur. Hatalı örnekler ilk aşamada mevcut olmayacağı için, rulman gelişim verisi üzerinde ayrılık sezimi yöntemleri uygulanmıştır. Sonuçlara göre rulman hatalarının ve diğer durum geçişlerinin tespiti ayrılık sezimi yöntemleriyle yapılabilir. Bazı durumlarda halihazırdaki yöntemlere göre daha erken geçiş tespiti yapılabilir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A_k	k^{th} component of Fast Fourier Transform of A
B_d	Rolling element diameter
C	KL divergence between two joint distributions
$\mathcal{C}_t$	Cell state of LSTM cell at time step t
$\tilde{\mathcal{C}}_t$	Candidate hidden state of LSTM cell at time step t
c	Number of recurrent units in LSTM cell
cA_i	i^{th} approximation coefficient of Discrete Wavelet Transform
cD_i	i^{th} detail coefficient of Discrete Wavelet Transform
D_i	Mahalanobis distance of observation x_i
E_a	Percentage energy of approximation coefficients
E_d^j	Percentage energy of detail coefficients at j_{th} level
E_{total}	Total energy of Discrete Wavelet Transform coefficients
e_i	Reconstruction error for i^{th} observation sample
$H(\tau)$	Hazard function
h_t	Hidden state of LSTM cell at time step t
f_{shaft}	Bearing shaft frequency
f_t	Forget gate of LSTM cell at time step t
i_t	Input gate of LSTM cell at time step t
$i*$	Index of the GMM component with smallest squared Mahalanobis distance between data and its centre
L	Number of hidden layers in LSTM-Autoencoder
M_i	Number of samples in i^{th} data window
N_r	Number of rotating elements in a bearing
$\mathcal{O}$	Big O notation that describes the complexity of an algorithm
o_t	Output gate of LSTM cell at time step t
P	Joint probability distribution in high-dimensional space
P_d	Bearing pitch diameter
$Perp$	Perplexity
$P(r_t\|r_{t-1})$	Conditional prior on the novelty point

$P(r_t\|x_{1:t})$	Joint distribution associated with the samples x_t^r
$P(x_{t+1}\|x_{1:t})$	Posterior distribution associated with the samples x_t^r
$P(x_t\|\chi_t^r)$	Predictive probability associated with the samples x_t^r
p	Degrees of freedom
p_i	Probability of i^{th} sample
$p_i(x)$	Probability distribution of mixture of Gaussians
$p_{j\|i}$	Probability of x_j chosen by x_i as its neighbor sample
Q	Joint probability distribution in low-dimensional space
$q_{j\|i}$	Low dimensional counterpart of $p_{j\|i}$
r_t	Run-length since last novelty
S_x	Scatter matrix of dataset χ
s_i	First order difference of reconstruction error
T	Number of iterations
T_j	j^{th} timestamp of a given data stream
$U^{(i,f,o,g)}$	Parameter matrix of LSTM gates between input and current hidden layer
X_j^d	d-dimensional vector arriving at timestamp T_j
x_i	Observation sample in sample data x
$\tilde{x}_i$	Prediction for i^{th} observation sample
$\vec{x}_i$	i^{th} column vector of a given dataset
x_t^r	Sample set associated with r_t
W_i	i^{th} data window
$W^{(i,f,o,g)}$	Parameter matrix of LSTM gates between previous and current hidden layers
w_i	Eigenvector corresponding to the i^{th} largest eigenvalue
y_i	Low dimensional counterpart of x_i
$\alpha(t)$	Momentum
η	Learning rate
μ_x	Mean of dataset x
$\tilde{\mu}$	Estimated mean vector
$\tilde{\mu}_n$	Rolling mean of $(x_n)_n$

$\tilde{\mu}_{n,m}$	Batch mean of (x_n) between n and m
$\tilde{\Sigma}$	Estimated covariance matrix
$\tilde{\Sigma}_n$	Rolling covariance matrix
$\tilde{\Sigma}_{n+m}$	Updated covariance matrix
σ^2	Variance of sample data x
σ_i	Gaussian kernel function
ϕ	Rolling element contact angle
χ_t^r	Hyperparameters that generate the distribution of x_t^r
ψ	Dropout rate for LSTM model

LIST OF ACRONYMS/ABBREVIATIONS

BPFI	Ball pass frequency inner race
BPFO	Ball pass frequency outer race
FTF	Fundamental train frequency
BPF	Ball pass frequency
CPS	Cyber-Physical Systems
CWRU	Case Western Reserve University
DM	Data Mining
EXPoSE	Expected Similarity Estimation
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
IIoT	Industrial Internet of Things
IMS	Center for Intelligent Maintenance Systems
IOS	Internet of Services
IoT	Internet of Things
KL	Kullback-Leibler
KPI	Key Performance Indicator
LSTM	Long-Short Term Memory
PCA	Principal Component Analysis
PHM	Prognostics and Health Management
RMS	Root-mean-square
SNE	Stochastic Neighbor Embedding
SPLL	Semi-Parametric Log-Likelihood
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
SY	Changxing Sumyoung Technology Co., Ltd.
t-SNE	t-Stochastic Neighbor Embedding
XJTU	Xi'an Jiaotong University

1. INTRODUCTION

The main objective of the present work is to perform streaming novelty detection on real-world predictive maintenance problems of rotating machinery. Predictive maintenance in rotating machinery context is based on monitoring mechanical condition of the equipment using heat, vibration or acoustic characteristics. Current and near future conditions can be assessed and maintenance can be applied at any time needed.

Due to the integrated nature of today's advanced manufacturing industry, any unanticipated failure from the simplest case to most complex one may lead to increase in costs and inefficient use of human and natural resources. Conventionally manufacturing processes are based on methods like corrective and preventive maintenance, however these are neither sufficient in the sense of eradicating the faults nor sustainable. Manufacturers need to reduce the downtime and cost of replacement of part of a machinery in order to be more competitive. As the manufacturing has evolved, diagnosis and prognosis methods have evolved as well. Previously, maintenance was executed only when it is definitely necessary. However, due to its nature, industrial processes are defined as chaotic and dynamic, which makes continuous monitoring and prognosis imperative.

Better assessment can be made by monitoring the system parameters continuously to find out when and in which part of the machinery that failure may occur. If such information is available, then the optimal action can be taken at the right time. Service life of components can be maximized if their conditions can be inspected correctly. Maintenance can be planned accordingly if potential failures are predicted. Condition-based maintenance, or usually called as predictive maintenance, is an improvement over preventive maintenance and it relies heavily on integrated sensors and analysis algorithms to schedule maintenance. Basic idea behind predictive maintenance is saving costs and improving efficiency.

The main obstacles to apply predictive maintenance framework in industry environment can be concluded into a couple of points:

- Necessary data that shows the working condition of machinery should be accessible in real-time. Degradation occurs in real-time and this reflects in captured data.
- Industrial big data approaches, like cloud-based data storage and decision-making from massive heterogeneous data sources, should be well-integrated with condition monitoring.
- Ideal data-driven models to monitor impending faults. Current literature largely contains supervised fault diagnosis and prognosis methods. However, each machinery has a unique vibration signature due to its nature and since machinery starts working at healthy status, faulty data is not available initially. Also working conditions may vary during the process which will reflect in captured data.

Since only healthy state data is available to build a model for fault diagnosis and prognosis, there should be an unsupervised method which marks a change point, namely a point where an earlier and a later state differ from one another. Change detection can be referred to as anomaly detection or novelty detection depending on the application. In the present work they are used synonymously.

The capability of recognizing unknown states is the one side of the coin, on the other side these methods should be applicable in real-time. Streaming data processing is a major concern of Big Data paradigm. Large volumes of temporal data arrives with high speed and decision support systems can be built if machine learning algorithms have the capability of decision making in real-time.

The structure of the remainder of the thesis can be summarized as follows: Chapter 2 gives an introduction to the predictive maintenance in Industry 4.0 concept. The advantages of predictive maintenance over its predecessors are briefly explained. Then rolling element bearings and how their fault analysis have been made conventionally are explained in detail. Then streaming novelty detection problem, its main challenges, the

state-of-the-art algorithms are given. In Chapter 3 the approach for streaming novelty detection on vibration signal is explained. Detection methods and dimension reduction techniques are given in detail. Bearing degradation datasets are introduced in Chapter 4. Experiments are conducted based on these datasets and methods given in Chapter 3. The results of the experiments are discussed on various aspects in Chapter 5. Main conclusions and contributions are summarized in Chapter 6.

2. BACKGROUND

2.1. Impact of Industry 4.0

Since the very beginning of industrial production it keeps on evolving and the changes are sometimes so dramatic we call them industrial revolutions. The fourth industrial revolution is a modernization era in the manufacturing industry that combines the recent advances in internet technologies, cyber-physical systems monitoring, mobile devices with high computational capabilities and artificial intelligence methods. The main principle in Industry 4.0 can be summarized as collecting and sharing data real-time and take advantage of these resources. This implies decentralized control systems rather than current centralized ones [1].

2.1.1. Maintenance Strategies

Maintenance is a necessity for every rotating machinery to preserve its optimal working conditions. Its extensity depends on the nature and value of the machinery. The following maintenance schemes have been used in industrial plants historically:

- In *corrective maintenance* there are not any measures until a breakdown happens. This scheme could be very inconvenient, since breakdown could happen unexpectedly. If the machinery was stopped earlier, repair might have been less costly [2].
- *Preventive maintenance* combines type of measures taken during the past two decades in many plants. Each critical machinery is stopped for a complete inspection periodically [3]. All the inspected faults and defects are repaired. However, there is always a risk of breakdown if it occurs between periodic inspections. Therefore, preventive maintenance is still not the most cost-efficient type of maintenance.
- Corrective and preventive maintenance policies are insecure, since there might be defects that are overlooked and can lead to breakdowns. In order to overcome

this drawback *predictive maintenance* has emerged as a new trend. The main objective of predictive maintenance improving security while reducing production costs that might occur due to repairs. It is based on continuous monitoring of health indicators of the machinery to detect any defect. This monitoring gives an insight about how to schedule maintenance in the near future. The main advantages of predictive maintenance over its predecessors can be summarized as the following:

(i) Industrial process is only shut down before imminent failure.

(ii) Costs are reduced by avoiding irrevocable damage occurs.

(iii) Machines are more available and reliable.

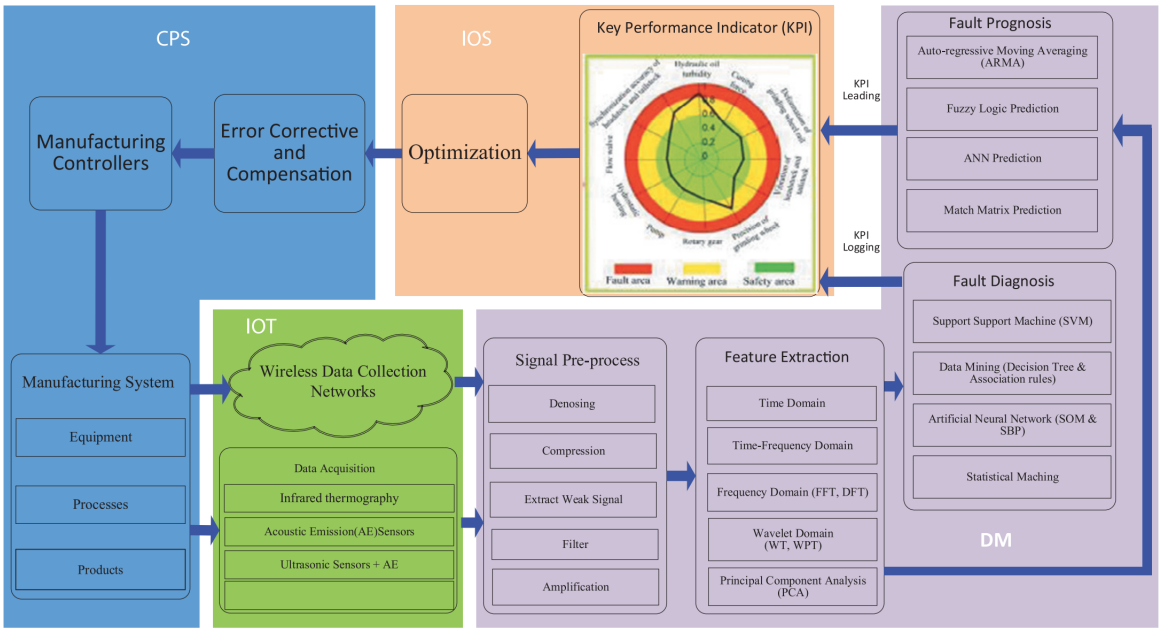


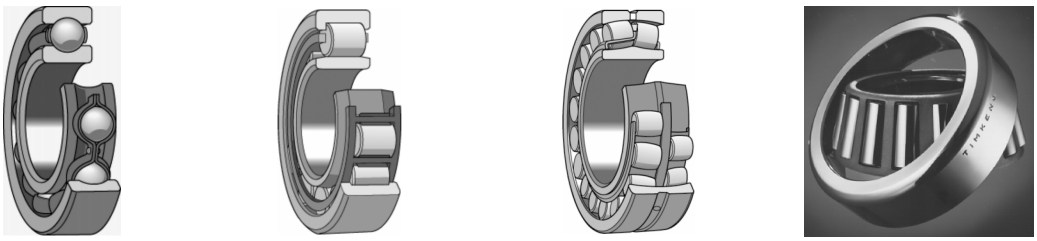
Figure 2.1: An example framework of predictive maintenance in Industry 4.0 [4].

2.2. Rolling Element Bearings and Vibration Analysis

2.2.1. Rolling Element Bearings

Rolling element bearings are designed to transfer the main load through the elements that are in rolling contact rather than in sliding contact. Elements between inner and outer races minimizes the rotational friction and they minimize rubbing.

There are different types of rollers such as balls, spherical, cylindrical and tapered. In today's industrial applications, ball bearings are widely used since they are the cheapest ones. They are shown in 2.2a.



(a) Ball (b) Cylindrical (c) Spherical (d) Tapered

Figure 2.2: Different types of rolling elements in rolling elements bearing [5].

The roller bearings are those shown in 2.2b, whose components are usually cylinders with length slightly greater than diameter. The main difference with ball bearings is that roller bearings with cylindrical elements have typically an higher load capacity, but they are affected by a lower capacity and higher friction under loads perpendicular to the primary supported direction.

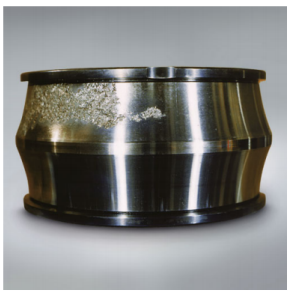
Spherical rollers have the outer ring provided with an internal spherical shape and the particular rollers shape. In the Center for Intelligent Maintenance Systems(IMS) and Xi'an Jiaotong University - Changxing Sumyoung Technology (XJTU-SY) data, which are particularly used in the present work, there are vibration samples collected from this type of bearings. One example is shown in 2.2c. Due to their geometry, they are difficult to produce, therefore they are expensive.

Tapered rollers in 2.2d differs from other types with their ability of supporting both radial and axial load. Also they can carry higher loads than thanks to greater contact area.

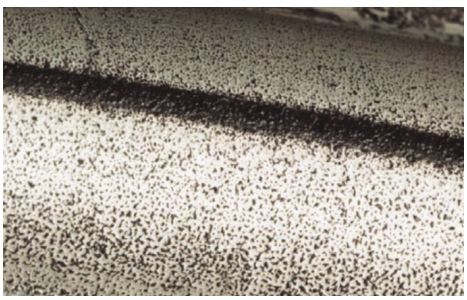
2.2.2. Possible Defects on Ball Bearings and Their Vibration Pattern

Assuming rolling element bearings have no defects and they are perfectly aligned, they still generate vibration. Any defect on the elements or bearing causes a change

in this vibration pattern. There are different types of defects that might occur on a bearing and in Figure 2.3 you can see couple of examples.



(a) Inner race flaking



(b) Outer race pitting

(c) Brinelling

(d) Corrosion

Figure 2.3: Examples of different types of faults on rolling element bearings [6].

If there is a defect on one of the bearing races or rolling element, impulses will be generated periodically with a certain frequency. Bearing vibrates at its resonance frequency when there is no defect, however vibration starts to include other frequencies rate at which the rolling elements go over the defect when there is one. These frequencies are of interest to detect bearing faults and Taylor (2003) discussed characteristic fault frequencies for bearings in his work [7]. These frequencies are widely used in industry and bearing diagnostics research. If we know about the bearing geometry, we can easily calculate them (in Hz) using following Equations (2.1a) to (2.1d).

$$\textit{Ball pass frequency inner race(BPFI)} = \frac{N_r}{2} \times f_{shaft} \times \left(1 + \frac{B_d}{P_d} \times \cos\phi\right) \tag{2.1a}$$

$$\textit{Ball pass frequency outer race(BPFO)} = \frac{N_r}{2} \times f_{shaft} \times \left(1 - \frac{B_d}{P_d} \times \cos\phi\right) \tag{2.1b}$$

$$\textit{Fundamental train frequency(FTF)} = \frac{f_{shaft}}{2} \times \left(1 - \frac{B_d}{P_d} \times \cos\phi\right) \tag{2.1c}$$

$$\textit{Ball pass frequency(BPF)} = \frac{P_d}{2B_d} \times f_{shaft} \times \left[1 - \left(\frac{B_d}{P_d} \times \cos\phi\right)^2\right] \tag{2.1d}$$

where N_r is the number of rotating elements, f_{shaft} is the shaft frequency which is basically rotation per second, P_d is the pitch diameter, B_d is the rolling element diameter and ϕ is the contact angle.

2.2.3. Vibration Monitoring using Signal Processing Techniques

- *Time-domain techniques:* Signal processing techniques for condition monitoring by vibration analysis have been used in prognostics research for many years [8,9]. Using these techniques we can extract features from the raw signal which can be indicators for an impending failure.

  In time domain root-mean-square(RMS), peak and kurtosis are good indicators for incipient fault. However, they go back to normal levels as the bearing defect becomes severe since vibration signal becomes random and fault signature is buried again [8].
- *Frequency-domain techniques:* The most common method to extract frequency features in a signal is Fast Fourier Transform(FFT). There are usually peaks at the defect frequencies, if they are not masked by other vibration components.

$$A_k = \sum_{m=0}^{N-1} a_m e^{-j2\pi\frac{mk}{N}} \qquad k = 0, \ldots, n-1 \tag{2.14}$$

The first frequency domain feature is the peak value of FFT:

$$Peak_{FFT} = \max(A_k) \tag{2.15}$$

Table 2.1: Time domain features.

RMS	$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$	(2.2)
Variance	$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$	(2.3)
Peak Value	$PVT = \max(\lvert x_i \rvert)$	(2.4)
Crest Factor	$\frac{PVT}{RMS} = \frac{\max(\lvert x_i \rvert)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}}$	(2.5)
Kurtosis	$\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{n \times \sigma^4} - 3$	(2.6)
Clearance Factor	$\frac{PVT}{\left(\frac{1}{n}\sum_{i=1}^{n}\sqrt{x_i}\right)^2}$	(2.7)
Impulse Factor	$\frac{PVT}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i \rvert}$	(2.8)
Shape Factor	$\frac{RMS}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i \rvert}$	(2.9)
Line Integral	$\sum_{i=0}^{n}\lvert x_{i+1} - x_i \rvert$	(2.10)
Peak To Peak Value	$\max(x_i) - \min(x_i)$	(2.11)
Shannon Entropy	$S = \sum_{i=0}^{n} p_i \log p_i, \quad p_i = \frac{x_i^2}{\sum_{i=0}^{n} x_i^2}$	(2.12)
Skewness	$\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^3}$	(2.13)

The second is energy of FFT:

$$E_{FFT} = \sum_{k=0}^{n-1} A_k \tag{2.16}$$

- *Time-Frequency techniques:* Vibration signals acquired from rotating machinery are usually non-stationary and frequency domain analysis may not be sufficient in that case. The solution could be time-frequency analysis, and wavelet transform is used to extract wavelet features. Percentage of energy corresponding to approximate and each detail coefficients are calculated. Total energy can be found as:

$$E_{total} = \sum_{j=1}^{l} \sum_{i=1}^{nj} (cD_i^j)^2 + \sum_{i=1}^{nl} (cA_i^l)^2 \tag{2.17}$$

So the percentage energy of approximation coefficient is:

$$E_a = \frac{\sum_{i=1}^{nl} (cA_i^l)^2}{E_{total}} \tag{2.18}$$

For detail coefficients percentage energies are the following:

$$E_d^j = \frac{\sum_{i=1}^{nj} (cD_i^j)^2}{E_{total}} \tag{2.19}$$

where l is the level of decomposition, and j is the number of detail coefficient.

2.3. Streaming Novelty Detection

Streaming data processing helps the decision-making process in real-time. A data stream is an infinite sequence of elements S,

$$S = \{(X_1, T_1), \ldots, (X_j, T_j), \ldots\} \tag{2.20}$$

where X_j is a d-dimensional vector arriving at time stamp T_j. There are two different types of time stamps. Explicit time-stamp is generated when data is arrived, and implicit time-stamp is given by a stream processing system.

Streaming data arrives continuously and it is usually noisy and corrupted. These characteristics pose a couple of challenges when processing streaming data:

- Incoming data volume may overcome the processing capacity. Therefore processing techniques must handle data rate in order to prevent loss of information.
- Value of data usually decreases over time, so recent data might be sufficient for many applications.

Therefore data stream could be unbounded in size and storing entire stream may not be efficient for prognosis purposes. Online learning algorithms which are able to continuously update the classifier(novelty detector) might be a solution. There needs to be a predictor which makes decisions as data arrives based on information obtained from previous stream and learn at the same time.

Conventional machine learning algorithms are first trained and then predict to classify new samples. The information from new samples are not used to improve the predictor. This is called offline learning [10]. These methods usually assume training samples are from a sample subspace which is constantly available and its samples are independent and identically distributed. In many real-world situations where streaming novelty detection is used these assumptions are not valid, such as internet-traffic, financial transactions, surveillance systems and fault detection in rotating machinery [11]. There needs to be an online learning where algorithm processes each instance on arrival and keeps learning continuously without the need for storage.

There are couple of works on online streaming novelty detection in the recent years. One-class SVMs was introduced in [12], however computational complexity restricts their usage in high dimensional datasets. Self adaptive and dynamic k-means is another unsupervised online anomaly detection method [13]. Expected Similarity

Estimation (EXPoSE) is kernel-based and able to efficiently compute the similarity between new data points and the distribution of regular data [10]. However these techniques either assumes a predefined distribution of data or they are computationally too complex to run in any environment. In addition, data storage is required to train most of these methods and when processing streaming data the most efficient way is to treat each arriving sample in a one pass manner.

3. METHODS

3.1. Novelty Detection Based Machine Monitoring Framework

For many years machine monitoring literature tackled with the problem in the following order:

- Model machine analytically and extract fault indicative features from the model.
- Compile this indicators along with healthy working levels.
- Check machine periodically to detect any defect.

Since there are too many components, machines and assets in a plant, deriving accurate models for each of them are nearly impossible. Even if reliable analytical models are derived, monitoring each of them manually is not really cost-efficient. Therefore, we can assume that the fully description of the cases that might occur in vibrational data won't be available.

As predictive maintenance becomes mainstream computer aided maintenance is found as an invaluable solution. Current rolling element fault diagnosis literature is heavily based on multiclass classification which is basically constructing a model to distinguish between each fault state and normal state. It has been widely used in the industry, especially with the emerging supervised learning algorithms in the past couple of decades. However, it has some drawbacks, for instance each machine starts working in a healthy state and to train these algorithms faults have to be induced artificially. There are multiple types of faults for each component and to be able to cover all possible defects too many experiments have to be run.

Novelty detection methods could overcome this issue since there is only need for healthy state vibration data. Once the machine starts working healthy vibration data could be used to build a model. This model will be an unsupervised streaming novelty detection algorithm since faulty vibration data will not be available in the first place.

In the present work this model uses time and frequency features of raw vibration signal as an input. Once a novelty is detected, a human expert can analyze the vibration data for further actions.

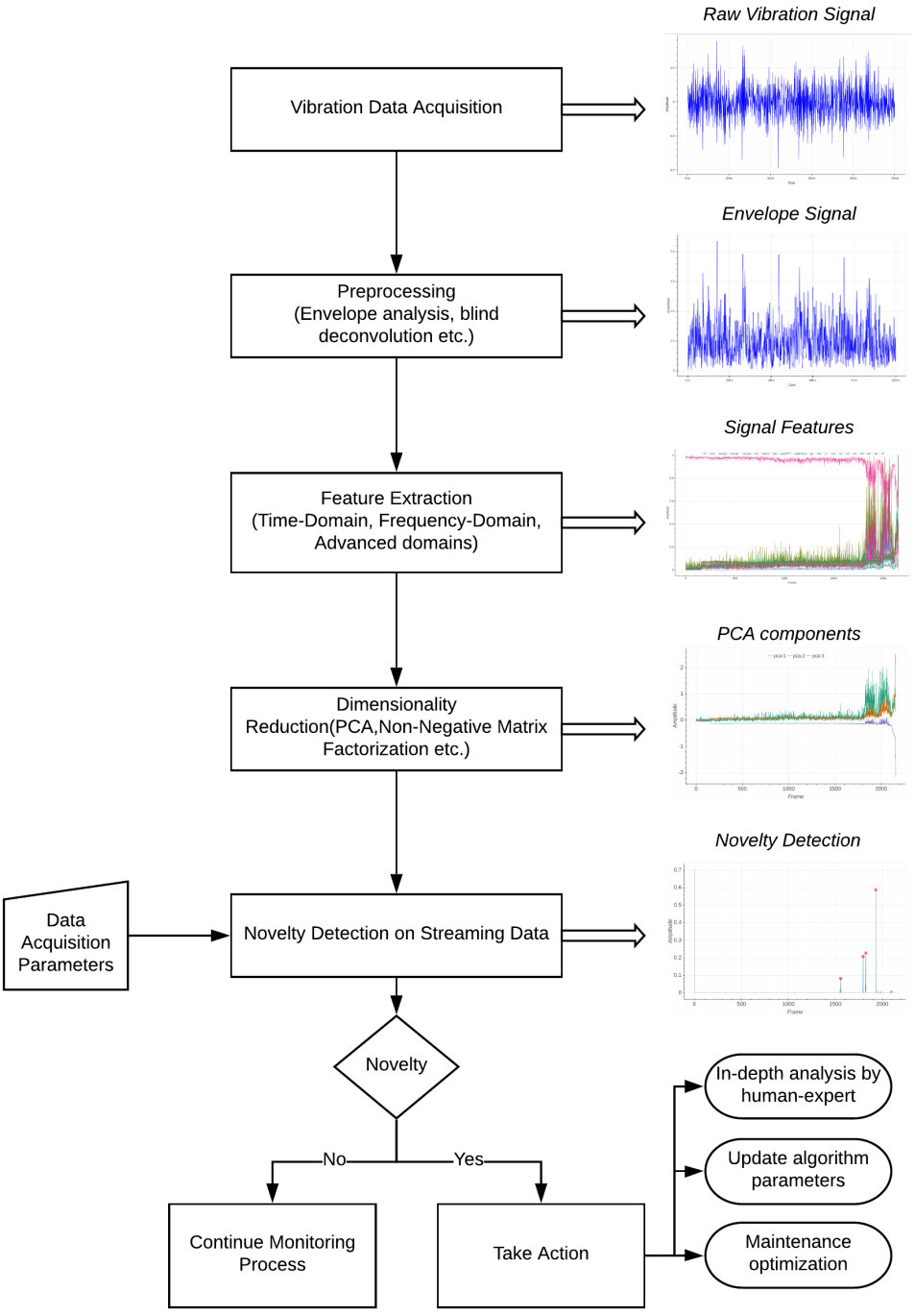


Figure 3.1: Machine monitoring framework.

3.2. Dimension Reduction in Feature Space

3.2.1. Principal Component Analysis(PCA)

Principal component analysis can be used to analyze the structure of a data set or allow the representation of the data in a lower dimensional dataset [14].

Let $\{\vec{x_i}\}$ be a set of N column vectors of dimension D. Define the scatter matrix S_x of the data set as

$$S_x = \sum_{i=1}^{N} (\vec{x}_i - \mu_x)(\vec{x}_i - \mu_x)^T \tag{3.1}$$

where μ_x is the mean of the dataset

$$\mu_x = \frac{1}{N} \sum_{i=1}^{N} \vec{x}_i \tag{3.2}$$

The d largest principle components are the eigenvectors $\vec{w}_i$ corresponding to the d largest eigenvalues. d can be chosen arbitrarily with $d < D$. The eigenvectors of S can usually be found by using singular value decomposition.

The dominant eigenvectors describe the main directions of variation of the data. For example, if a dataset had 2 large eigenvalues, then the data variation is described largely by linear combinations of the 2 corresponding eigenvectors.

The d eigenvectors can also be used to project the data into a d dimensional space.

$$W = [\mu_1, \mu_2, \ldots, \mu_d] \tag{3.3}$$

The projection of vector $\vec{x}$ is $\vec{y} = W^T\vec{x}$. The corresponding scatter matrix S_y of the vectors $\{\vec{y_i}\}$ is:

$$S_y = W^T S_x W \tag{3.4}$$

3.2.2. t-Stochastic Neighbor Embedding(t-SNE)

Stochastic neighbor embedding (SNE) is a nonlinear manifold learning algorithm which maintains the consistency of the neighborhood probability distribution between high-dimensional and low-dimensional space. The traditional Euclidean distance-based similarity is transferred to conditional probability-based similarity. Gaussian distribution is adopted to simulate the similarity relationship between observation samples. Given that observation sample x_i and bandwidth of the Gaussian kernel function σ_i, $p_{j|i}$ is the probability of x_j chosen by x_i as its neighbor sample,

$$p_{j|i} = \frac{e^{\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}}} \tag{3.5}$$

For closer samples, $p_{j|i}$ is relatively high, and it will be almost infinitesimal for largely separated samples. In low-dimensional space, SNE adopts the Gaussian distribution to measure the similarity between low-distinction samples in a similar fashion:

$$q_{j|i} = \frac{e^{-\|y_i - y_j\|^2}}{\sum_{k \neq i} e^{-\|y_i - y_k\|^2}} \tag{3.6}$$

where y_i and y_j are low-dimensional counterparts of high dimensional points x_i and x_j and conditional probability $q_{j|i}$.

However there are two major drawbacks of SNE which were responded by Maaten and Hinton (2008) [15]. They introduced t-SNE to overcome the complexity of objective function of SNE and so-called "crowding" problem, if data are separated from one

another in high-dimension they must be gathered in the process of mapping in low-dimension.

First problem is solved by applying the characteristic of symmetry to get simpler objective function and optimize the gradient form, which is referred to as symmetric SNE. Instead of minimizing the sum of Kullback-Leibler(KL) divergences between the conditional probabilities $p_{j|i}$ and $q_{j|i}$ KL divergence between a joint probability distribution, P, in the high-dimensional space and a joint probability distribution, Q, in the low-dimensional space is minimized:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.7}$$

The second problem is solved using Student t-distribution with one degree freedom to measure the similarity of low-dimensional space. $(1 + \|y_i - y_j\|_2^2)^{-1}$ is the reciprocal of points separated from one another in low-dimensional space to $\|y_i - y_j\|$. So the joint distribution q_{ij} is:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|_2^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|_2^2)^{-1}} \tag{3.8}$$

In addition, the t-distribution function offers the same performance as the Gaussian function because the t-distribution function is the mixture of infinite Gaussians. So the gradient of t-SNE is:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - qij)(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{3.9}$$

In summary, non-similarity is associated with points far from each other and similarity is associated with points close to each other in t-SNE. t-distribution function reduces the complexity of objective function.

input : Data: $\chi = \{x_1, x_2, \ldots, x_n\}$
Perplexity: $Perp$
Number of iterations: T
Learning rate: η
Momentum: $\alpha(t)$
output: Low-dimensional data representation: $Y^{(T)} = \{y_1, y_2, \cdots y_n\}$
begin
compute $p_{j|i}$ with perplexity $Perp$ (using Equation 3.5)
set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
sample initial solution $Y^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $N(0, 10^{-4}I)$
for $t \leftarrow 0$ **to** T **do**
compute low-dimensional affinities q_{ij} (using Equation 3.8)
compute gradient $\frac{\delta C}{\delta Y}$ (using Equation 3.9)
set $Y^{(t)} = Y^{(t-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(t)(Y^{(t-1)} - Y^{(t-2)})$
end
end

Figure 3.2: t-SNE algorithm in its simplest form [15].

3.3. Streaming Novelty Detection Methods

3.3.1. Mahalanobis Online Streaming Novelty Detection

The classical definition of Mahalanobis distance is for every $d-$dimensional observation x_i of the sample $\{x_1, \ldots, x_n\}$:

$$D_i = \sqrt{(x_i - \tilde{\mu}_i)\tilde{\Sigma}^{-1}(x_i - \tilde{\mu}_i)^T} \tag{3.10}$$

where $\tilde{\mu}$ and $\tilde{\Sigma}$ are estimated mean and covariance matrix of the sample set respectively. For streaming novelty detection, this measure is implemented online, which means in the beginning there isn't any knowledge about the distribution of the samples and it is learned as the samples arrive. The algorithm updates the mean and covariance matrix of the sample set. The following formulas were used to update the mean and covariance

matrix:

$$\tilde{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{3.11}$$

where $\tilde{\mu}_n$ is the rolling mean of $(x_n)_n$

$$\tilde{\mu}_{n,m} = \frac{1}{m-n}\sum_{k=n+1}^{m} x_k \tag{3.12}$$

where $\tilde{\mu}_{n,m}$ is the batch mean of (x_n) between n and m. So the updated mean is,

$$\tilde{\mu}_{n+m} = \tilde{\mu}_n + \frac{m}{n+m}(\tilde{\mu}_{n,n+m} - \tilde{\mu}_n) \tag{3.13}$$

In order to update covariance matrix, first rolling covariance matrix should be known:

$$\tilde{\Sigma}_n = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \tilde{\mu}_n)(x_k - \tilde{\mu}_n)^T \tag{3.14}$$

Then updated covariance matrix can be calculated as:

$$\tilde{\Sigma}_{n+m} = \frac{n-1}{n+m-1}\tilde{\Sigma}_n + \frac{1}{n+m-1}\sum_{i=0}^{m-1}\frac{n+i}{n+i+1}(x_{n+i+1} - \tilde{\mu}_{n+i})(x_{n+i+1} - \tilde{\mu}_{n+i})^T \tag{3.15}$$

A novelty score, simply Mahalanobis distance is calculated for each observation.

$$D_i = \sqrt{(x_i - \tilde{\mu}_{i-i})\tilde{\Sigma}_{i-1}^{-1}(x_i - \tilde{\mu}_{i-1})^T} \tag{3.16}$$

3.3.2. Bayesian Online Streaming Novelty Detection

In their paper, Adams and Mackay (2007) describes a change-point detection algorithm that uses Bayesian reasoning and operates online [16]. The assumption is, if a change occur in the data stream, distribution changes and new sequence of

independent and identically distributed samples arrive.

When a new sample arrives there are two possibilities. First one is it comes from current distribution, which means we increase the run-length by one and use it to improve the estimated parameters of current distribution using Bayes theorem. The other one is it comes from different distribution, thus it is a novelty. We initialize the distribution and reset the run-length.

Current run r_t is the time length since last novelty and x_t^r is the sample set associated with current run. The hyperparameters that generates the distribution of this sample set is χ_t^r. The conditional prior on the novelty point $P(r_t|r_{t-1})$ is calculated as the following:

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}+1) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}+1) & \text{if } r_t = r_{t-1}+1 \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

where $H(\tau)$ is the Hazard function [17]:

$$H(\tau) = \frac{P_{gap}(g=\tau)}{\sum_{t=\tau}^{\infty} P_{gap}(g=t)} \tag{3.18}$$

In the original paper authors used a special case for $P_{gap}(g)$, where it is a discrete exponential distribution with timescale λ. Hazard function becomes constant at $\frac{1}{\lambda}$ in that case.

The objective is calculating predictive distribution by integrating over the posterior distribution on the current run length r_t:

$$P(x_{t+1}|x_{1:t}) = \sum_{r_t} P(x_{t+1}|x_t^{(r)}, r_t) P(r_t|x_{1:t}) \tag{3.19}$$

input : Data: $X = \{x_1, x_2, \ldots, x_n\}$

Distribution hyperparameters: χ_{prior}

output: Prediction probabilities: $P(x_{t+1}|x_{1:t})$

begin

initialize $P(r_0 = 0)$ and $\chi_1^0 = \chi_{prior}$

for $t \leftarrow 0$ **to** n **do**

observe new sample x_t

evaluate predictive probability:

$\pi_t^{(r)} = P(x_t|\chi_t^r)$

calculate growth probability:

$P(r_t = r_{t-1} + 1, x_{1:t}) = \pi_t^{(r)} P(r_{t-1}, x_{1:t-1})(1 - H(r_{t-1}))$

calculate novelty probability:

$P(r_t = 0, x_{1:t}) = \sum_{r_{t-1}} \pi_t^{(r)} P(r_{t-1}, x_{1:t-1}) H(r_{t-1})$

calculate normalization constant:

$P(x_{1:t}) = \sum_{r_t} P(r_t, x_{1:t})$

find posterior distribution:

$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{P(x_{1:t})}$

calculate new hyperparameters $\chi_{t+1}^{(r+1)}$ for run-length $r + 1$

perform prediction:

$P(x_{t+1}|x_{1:t}) = \sum_{r_t} P(x_{t+1}|x_t^{(r)}, r_t) P(r_t|x_{1:t})$

end

end

Figure 3.3: Bayesian online change-point detection algorithm [16].

According to Bayesian theorem posterior distribution $P(r_t|x_{1:t})$ is:

$$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{P(x_{1:t})} \tag{3.20}$$

To find it recursively, joint distribution can be written as the following:

$$\begin{aligned} P(r_t, x_{1:t}) &= \sum_{r_{t-1}} P(r_t, r_{t-1}, x_{1:t}) \\ &= \sum_{r_{t-1}} P(r_t, x_t | r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(x_t | r_t, r_{t-1}, x_{1:t-1}) P(r_t | r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(r_t | r_{t-1}) P(x_t | x_t^{(r)}) P(r_{t-1}, x_{1:t-1}) \end{aligned} \tag{3.21}$$

Equation 3.21 is used as a novelty score for Bayesian online novelty detection method. For different run-length values novelty scores for last r_t points can be calculated. In the present work r_t is chosen as 10 last data points.

3.3.3. Semi-Parametric Log-Likelihood(SPLL)

Assume that the change detection criteria are calculated from pre-specified windows of data W_1 and W_2. As a special case of log-likelihood change detection semi-parametric log-likelihood criterion(SPLL) was introduced by Kuncheva (2013) [18]. Suppose that the data before change-point is from mixture of Gaussians $p_1(x)$ with c components each with same covariance matrix. The parameters of distribution are estimated using data from first window W_1. Using data from second window W_2 change detection criteria is derived as following:

$$SPLL = \max(SPLL(W_1, W_2), SPLL(W_2, W_1)) \tag{3.22}$$

where,

$$SPLL(W_1, W_2) = \frac{1}{M_2} \sum_{x \in W_2} (x - \mu_{i*})^T \Sigma^{-1} (x - \mu_{i*}) \tag{3.23}$$

where M_2 is the number of objects in W_2, Σ is the covariance matrix estimated using first window W_1 and $i*$ the index of the component with the smallest squared

Mahalanobis distance between x and its centre,

$$i* = \arg\min_{i=1}^{c} (x - \mu_i)^T \Sigma^{-1} (x - \mu_i) \tag{3.24}$$

If $p_1(x)$ is really mixture of Gaussians and W_2 comes from $p_1(x)$ the squared Mahalanobis distances have a chi-square distribution with p degrees of freedom. Expected value is also p and standard deviation is $\sqrt{2p}$. If W_2 is from another distribution, mean of distances will deviate from p. Then windows will be swapped and $SPLL$ will be calculated again. The result of 3.22 will be used as an novelty score. It should be noted that window size is fixed to 10 in the present work. For the implementation of SPLL algorithm code repository created by Lucy Kuncheva was used [19].

input : Data: $\chi = \{x_1, x_2, \ldots, x_n\}$
Window size: n
output: SPLL novelty score: SPLL
begin
  for $t \leftarrow 0$ **to** $len(\chi) - n$ **do**
    assign W_1 and W_2 windows each has length n
    compute clusters in each window
    find weighted average covariance matrix Σ using the clusters of W_1
    find $SPLL_1$, swap windows W_1 and W_2 and calculate $SPLL_2$
    SPLL = max($SPLL_1$,$SPLL_2$).
  end
end

Figure 3.4: SPLL algorithm.

3.3.4. LSTM-Autoencoders

Long-short term memory(LSTM) neural networks [20] are recurrent models that have been used for many years for handwriting recognition, speech recognition and sentiment analysis. LSTM based encoder decoder structure was proposed for machine translation task which is a sequence-to-sequence learning task [21]. Since then similar

structures were used for image captioning [22] and natural language generation and reconstruction [23].

In Figure 3.5 you can see an example LSTM cell. Here h is the hidden state, i, f, o are input, forget and output gates respectively. All these gates are calculated with same equations, except their parameter matrices W and U are different. W is the recurrent connection between previous and current hidden layers and U is the parameter that connects the input to the current hidden layer. i, f, o are called gates because when they are multiplied with another vector element-wise, we simply allow a part of this vector to let through. In forget gate how much of the previous state we want to let through is defined. Input gate lets through a part of newly computed state and the output gate defines the amount of internal state we want to expose to the higher layers at the next time steps. The gates have the same dimensions, which is also described as the size of hidden state.

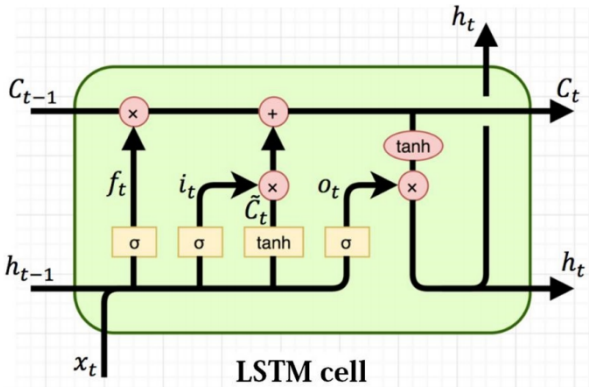


Figure 3.5: An example LSTM cell [24].

$$
\begin{aligned}
i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\
f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\
o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\
\tilde{C}_t &= \tanh(x_t U^g + h_{t-1} W^g) \\
C_t &= \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \\
h_t &= \tanh(C_t) * o_t
\end{aligned}
\tag{3.25}
$$

Current input and previous hidden state are used to calculate candidate hidden state $\tilde{\mathcal{C}}$. $\mathcal{C}$ can be defined as internal memory of the LSTM cell and it is a combination of forget gate, multiplied by its previous self, and input gate, multiplied by newly computed hidden state. So if forget gate or input gate consists of all zeros we could completely forget old memory or newly computed state respectively. Finally, new hidden state can be computed by multiplying output gate by the internal memory.

An autoencoder is an artificial neural network with symmetrical structure, that is used to encode input data in an unsupervised manner. The aim is to learn a lower dimensional representation of input data in the hidden layer. The component between input and hidden layer called encoder, whereas the component between hidden layer and output is called decoder. Decoder is the reconstructing side where the aim is to generate original input from encoded part.

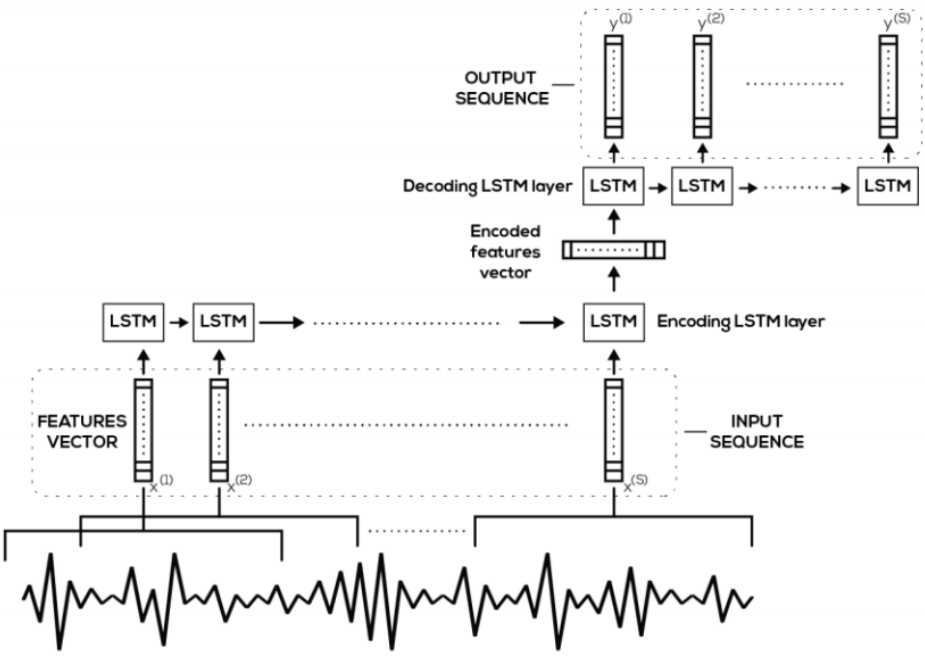


Figure 3.6: LSTM-Autoencoder [25].

Figure 3.6 is a basic LSTM autoencoder architecture with single LSTM layer on both encoder and decoder side. The window length is an input parameter to the model. When encoder reaches the last state, the resulting cell state will be given to decoder's initial state.

According to specifications of input data, hyperparameters like hidden layer size and input window length are learned for initialization. For training phase other hyperparameters like batch size and number of epochs should be specified. Since the process needs to happen online, normal data is captured first from stream. Once novelty detection task is learned, reconstruction error for point i is calculated:

$$e_i = |x_i - \tilde{x_i}| \tag{3.26}$$

As novelty score first order difference of reconstruction error is used:

$$s_i = e_i - e_{i-1} \tag{3.27}$$

The novelty score of every instance in a window is compared with a predefined threshold, and values over that threshold are predicted as anomalies.

input : Data: $\chi = \{x_1, x_2, \ldots, x_n\}$
Threshold: τ
Encoder dimension: *enc_dim*
Decoder dimension: *dec_dim*
Dropout: ψ
Learning rate: η
output: Novelty score : $s = \{s_1, s_2, \cdots s_n\}$
begin
  initialize seq2seq model with given parameters
  use healthy samples from χ to train the model
  for $t \leftarrow 0$ **to** $len(\chi) - n$ **do**
    calculate s_t for each sample χ_t using trained model
    if $e_t \geq \tau$ **then**
      label χ_t as an novelty
  end
end

Figure 3.7: LSTM-Autoencoder algorithm.

3.4. Technology

This chapter introduces the programming language and platform used in the thesis and justifies the advantages of choosing them. This chapter also includes the packages, libraries and open-source software that are used in this thesis.

3.4.1. Python

Python was released in 1991 and it is a very popular open source programming language since then. Python is an interpreted language, has a design philosophy which emphasizes code readability, and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java [26]. Python 3.7 was used for this project.

There are numerical calculation, data manipulation and machine learning packages written for Python for almost any conceivable function packages. The ones that are used in thesis are as the following:

(i) *NumPy:* This library is used to work with multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. [27].

(ii) *SciPy:* It was built on NumPy array framework, and includes advanced scientific functions like integration and optimization functions [28].

(iii) *Pandas:* This library allows importing CSV files and stores them as DataFrame objects which are data structures that make easier the manipulation of data. The other Python libraries' functionalities could be applied to DataFrame object as well [29].

(iv) *tqdm:* Smart progress meter for loops [30].

(v) *PyWavelets:* An open source wavelet transform software for Python. It was used to extract wavelet features from the raw vibration samples [31].

(vi) *keras:* Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano [32]. TensorFlow backend

was used for this project.

(vii) *Scikit-learn:* A machine learning library that features various classification, regression and clustering algorithms [33].

(viii) *Bokeh:* An interactive visualization library [34].

(ix) *Matplotlib:* A Python plotting library [35].

(x) *Jupyter Notebook:* A web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text [36].

4. EXPERIMENTS

4.1. Bearing Datasets

In today's prognostics and health management community, vibration-based diagnosis and prognostics arouses an ever-growing interest. The well accepted one by (prognostics and health management)PHM community is the Case Western Reserve University Data [37]. It has been used in numerous works since published. There are also fairly new datasets published by Paderborn University [38], University of Ottawa [39] and Politecnico di Torino [40]. All these datasets include healthy and damaged samples taken from varying operating conditions. Several fault types were either introduced artificially or they occurred at the end of accelerated life tests. They are precious for PHM community and have potentials for diagnostics purposes, however prognostics research needs endurance tests with natural degradation to come up with more viable solutions. The dataset provided by Center for Intelligent Maintenance Systems(IMS) at University of Cincinnati [41] has been widely used by prognostics community since it was published. Ali et al. (2015) used it for accurate remaining useful life estimation [42] and Mortada et al. (2011) used it for early damage detection [43]. Both of these problems require continuous and naturally degrading signal, that started from a healthy state. IMS bearing dataset was collected in a way that it reflects real defect propagation processes. XJTU-SY bearing dataset were also collected for a similar purpose and they are provided by the Institute of Design Science and Basic Component at Xi'an Jiaotong University (XJTU), Shaanxi, P.R. China and Changxing Sumyoung Technology Co., Ltd. (SY), Zhejiang, P.R. China [44]. There are run-to-failure data of 15 rolling element bearings acquired by degradation experiments. In the present work, IMS and XJTU-SY datasets are used extensively to come up with accurate streaming novelty detection methods for rolling element vibration signals.

4.1.1. IMS Bearing Dataset

- *Data test rig:* The aforementioned datasets used by PHM community usually have recordings from simulated or seeded damages. There are many ways to simulate inner/outer race or ball defects, such as drilling the surface or machining with electrical discharge. However natural defects in early stages may not be detected, if methods and algorithms are only developed using simulated data. IMS bearing data distinguishes itself since faults occur naturally in the end of run-to failure tests.

  In a specially designed test rig in Figure 4.1, bearing run-to failure tests were performed under normal load conditions until the accumulation of debris on a magnetic plug exceeded a certain level. When the accumulation reaches this level that indicates an impending failure and an electrical switch closes to end the experiment.

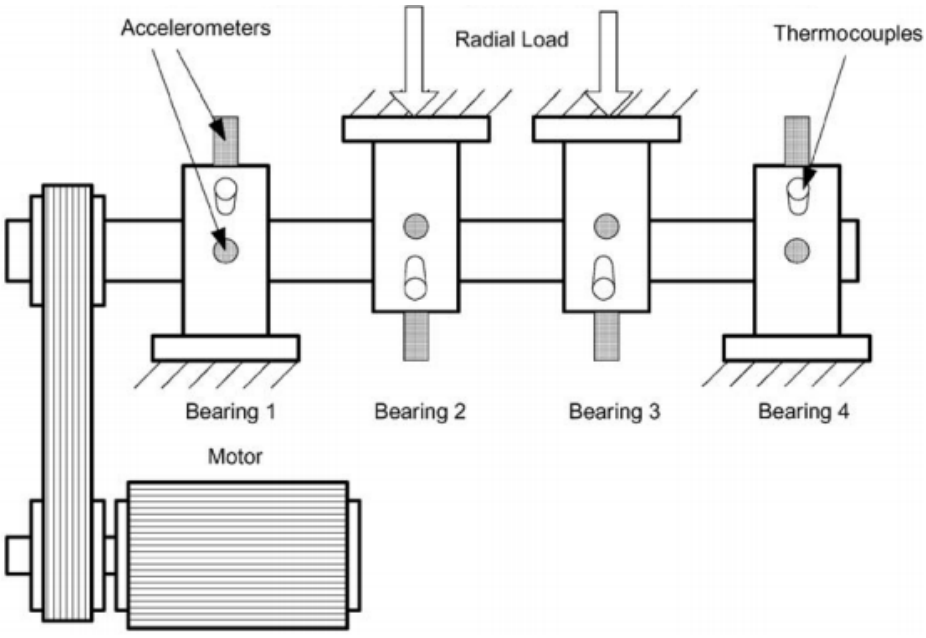


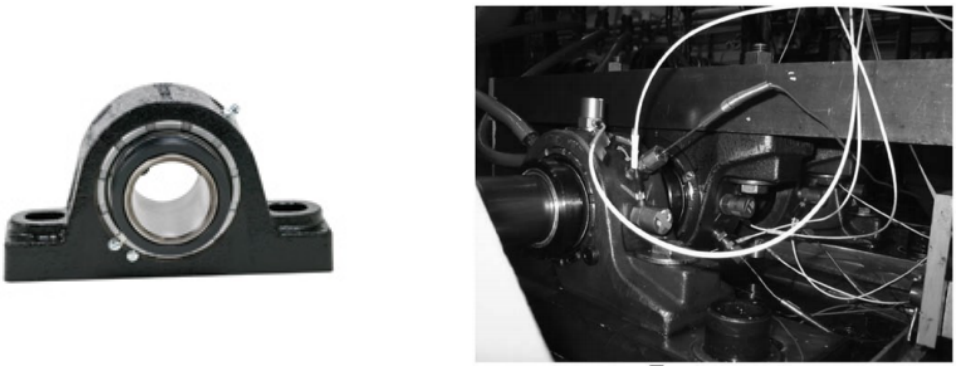
Figure 4.1: IMS bearing test rig [41].

On each bearing housing a PCB 353B33 High Sensitivity Quartz ICP® accelerometer was installed. An AC motor, coupled by a rub belt, kept the rotation speed constant at 2000 rpm. 6000 lbs. of radial load applied onto the shaft and bearing by a spring mechanism. At the outer race of each bearing thermocouples were attached to record bearing temperature. Four double row Rexnord ZA2115 bearings were mounted onto

the shaft as shown in Figure 2.1. Bearing characteristics are given in Table 4.1.

Table 4.1: Characteristics of Rexnord ZA2115.

Rolling element per row	16
Pitch diameter	2.815 inch
Roller diameter	0.331 inch
Contact angle	15.17°



(a) Rexnord ZA2115 Roller Bearing [45]. (b) Example of Rexnord ZA2115 on test rig [41].

Figure 4.2: The bearings were force lubricated via oil circulation system that regulates the flow and the temperature of the lubricant.

- *Data collection:* National Instruments DAQCard-6062E data acquisition card and LabVIEW were used to collect vibration data. Datasets composed of files of one second length where each file has 20,480 data points. In the original paper it is mentioned that sampling frequency is 20kHz, but analysis by Gousseau et al. (2016) suggests it is in fact 20.48kHz [46]. Data acquisition was made in every ten minutes, except for the first forty-three files of the first dataset which were recorded in every five minutes. For the first dataset, time intervals were marked where acquisition was switched off. There are also discontinuities in the third dataset, but they are not as frequent as the first dataset. Also for each bearing, two accelerometers were used for the acquisition of the first dataset.
- *Characteristic fault frequencies:* Using Equations (2.1a) to (2.1d) and bearing characteristics given in Table 4.1 the frequencies are calculated in Table 4.3.

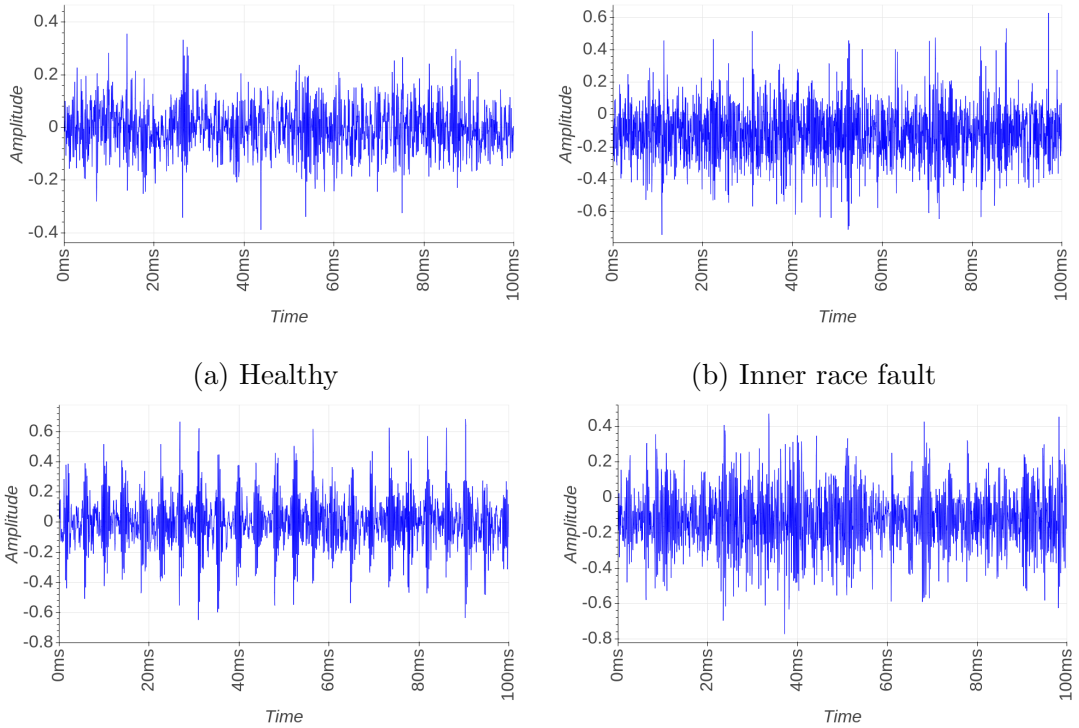
Table 4.2: IMS dataset descriptions.

Dataset	Number of files	Number of channels	Endurance duration	Signal duration	End of test damages
1	2156	8	34d 11h 33m 32s	35m 56s	Bearing 3(Inner race) Bearing 4(Rolling)
2	984	4	6d 19h 50m 0s	16m 24s	Bearing 1(Outer race)
3	4448	4	31d 9h 34m 11s	74m 8s	Bearing 3(Outer race)

Table 4.3: Characteristic fault frequencies for ZA2115.

$BPFI = 293.96\ Hz$	$BPFO = 234.04\ Hz$	$FTF = 14.18\ Hz$	$BPF = 138.52\ Hz$

- *Dataset analysis:* In Figure 4.3 a simple time domain response of healthy vibration signal and each designated fault type are depicted. Healthy sample is one of the earliest ones captured, whereas faulty ones are all taken at the end of endurance tests. Time-frequency analysis is based on the Short Time Fourier Transform(STFT) which is an effective tool for tracking frequencies along the experiments. STFT of each sample, without overlap is used to create spectrograms in Figure 4.4. Spectrograms are all created on a logarithmic scale. Except outer race fault in Bearing 1 of Dataset 2, bearing defects cannot easily be seen in frequency spectrum until the end of experiments. In [46] different preprocessing methods were suggested for IMS bearing dataset, but none of them provides a solution for early prediction of bearing defects. As Gousseau et al. (2016) mentioned in their work, the third dataset hasn't been used by PHM community and contains false-negatives, thus it is discarded from the experiments in this thesis.



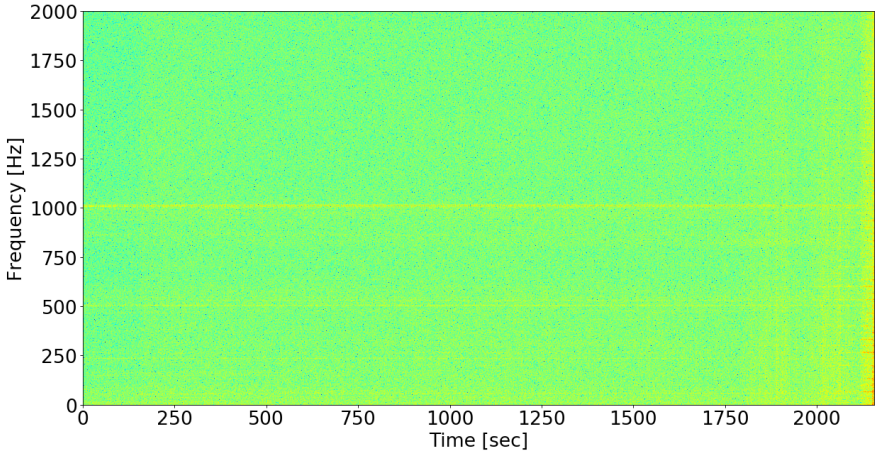
(a) Healthy

(b) Inner race fault

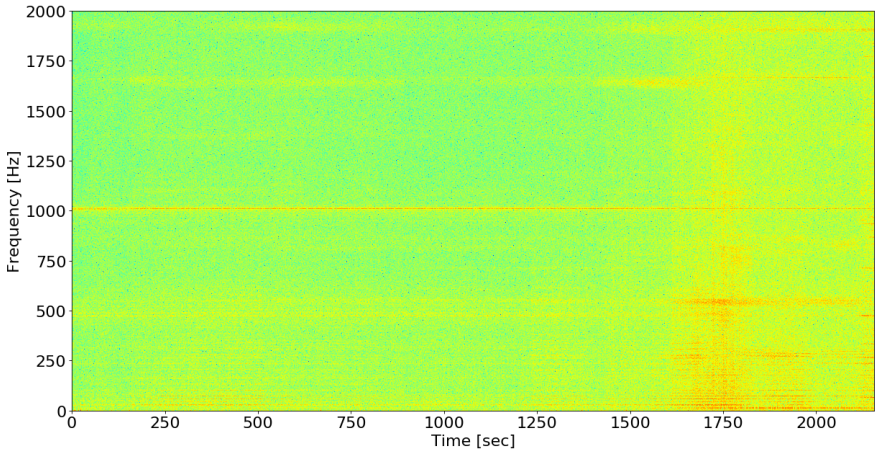
(c) Outer race fault

(d) Rolling element fault

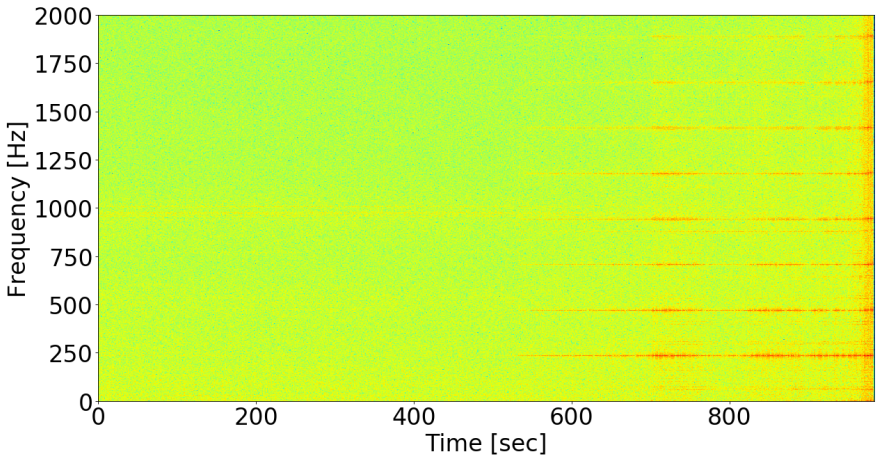
Figure 4.3: Example raw vibration signals from IMS Dataset. Figure 4.3a is one of the earliest samples of Bearing 1 in Dataset 2. In the end of the experiment there is outer race fault in the same bearing. Figure 4.3c is a sample where we can assume there is a severe defect on the outer race of the same bearing. Figure 4.3b and Figure 4.3d are both taken from the late samples of Dataset 1.



(a) Bearing 3 of Dataset 1 shows signs of inner race fault after 2000 seconds (Experiment day 32) of test.



(b) Although there are signs of defect in Bearing 4 of Dataset 1, BPF can clearly be seen at the end of the experiment.



(c) Bearing 1 of Dataset 2 has outer race fault in the end.

Figure 4.4: Spectrograms of each pre-defined defect in IMS Dataset.

4.1.2. XJTU-SY Bearing Dataset

- *Data test rig:* In Figure 4.5, you can see the components of bearing test bed. Unlike IMS bearing dataset, which includes single operating condition experiments, this dataset is designed to conduct the accelerated degradation tests of rolling element bearings under different operating conditions (i.e., different radial force and rotating speed). The type of bearing in this dataset is LDK-UER204 and its characteristics are given in Table 4.4.

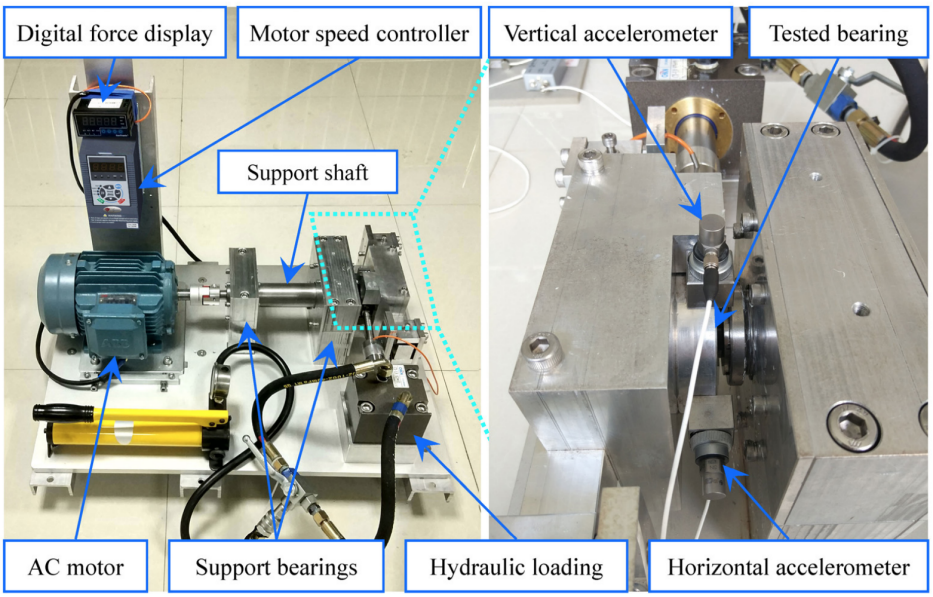


Figure 4.5: XJTU-SY bearing testbed [44].

Table 4.4: Characteristics of LDK-UER204.

Rolling element per row	8
Pitch diameter	34.55 mm
Roller diameter	7.92 mm
Contact angle	0°

- *Data collection:*Two accelerometers of type PCB 352C33 were used to collect vibration signals. One is mounted on horizontal axis and the other is mounted vertical axis. Sampling frequency is 25.6kHz and duration is 1.28s for each sample. Each sample is taken in every one minute. In the present work only the samples captured by horizontally mounted accelerometer are used.

Table 4.5: XJTU-SY dataset descriptions.

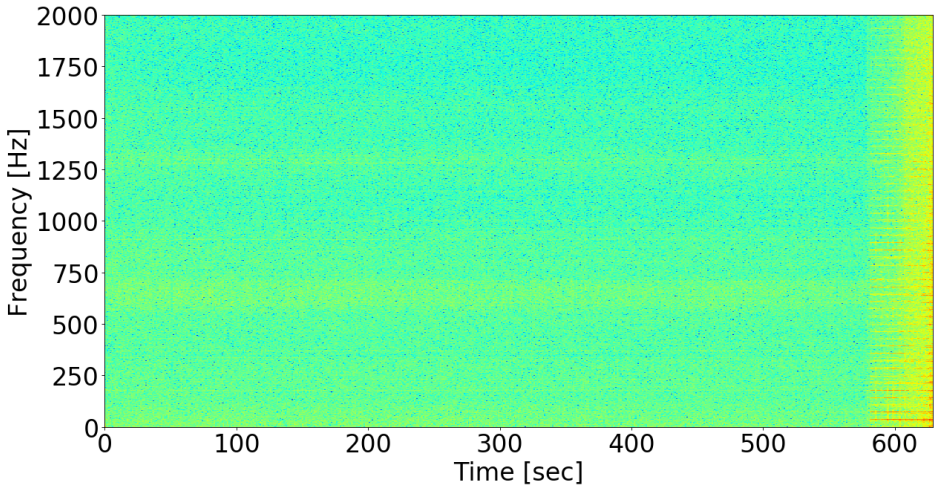
Condition	Bearing	Number of samples	Total Duration	Fault element
35Hz/12kN	Bearing 1_1	123	2h 3m	Outer
	Bearing 1_2	161	2h 41m	Outer
	Bearing 1_3	158	2h 38m	Outer
	Bearing 1_4	122	2h 2m	Cage
	Bearing 1_5	52	52m	Inner/Outer
37.5Hz/11kN	Bearing 2_1	491	8h 11m	Inner
	Bearing 2_2	161	2h 41m	Outer
	Bearing 2_3	533	8h 53m	Cage
	Bearing 2_4	42	42m	Outer
	Bearing 2_5	339	5h 39m	Outer
40Hz/10kN	Bearing 3_1	2538	42h 18m	Outer
	Bearing 3_2	2496	41h 36m	Inner/Ball Cage/Outer
	Bearing 3_3	371	6h 11m	Inner
	Bearing 3_4	1515	25h 15m	Inner
	Bearing 3_5	114	1h 54m	Outer

- *Characteristic fault frequencies:* Using Equations (2.1a) to (2.1d) and bearing characteristics given in Table 4.4 the frequencies are calculated in Table 4.6.

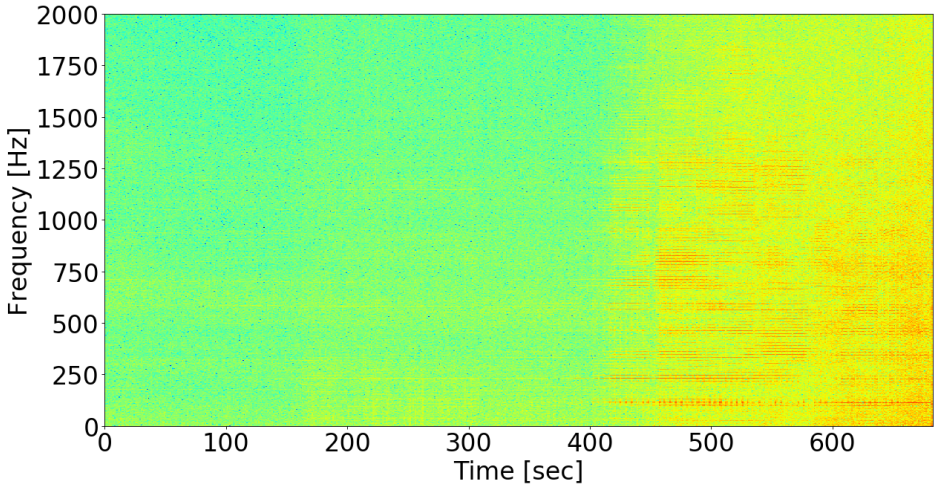
Table 4.6: Characteristic fault frequencies for LDK-UER204.

$BPFI = 196.68\ Hz$	$BPFO = 123.32\ Hz$	$FTF = 15.42\ Hz$	$BPF = 82.66\ Hz$

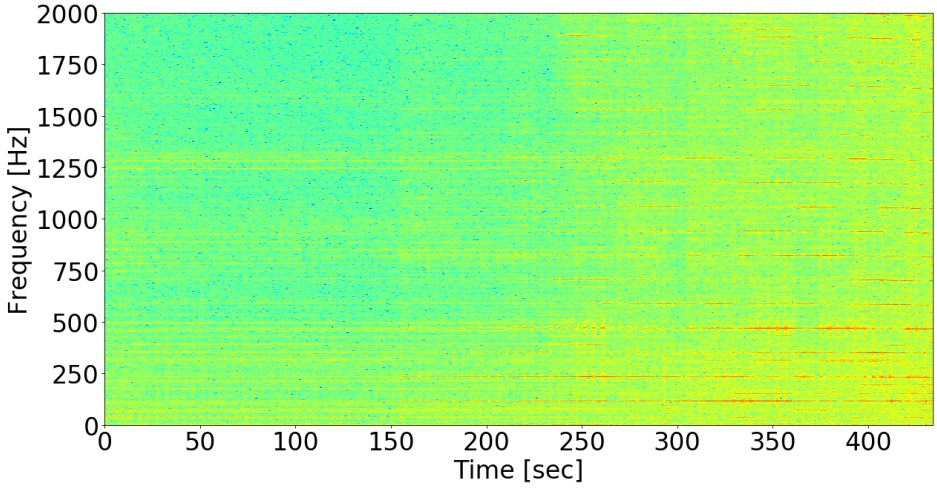
- *Dataset analysis:* Although there are more samples in XJTU-SY dataset than IMS dataset, most of them have short durations and may not contain enough information for novelty detection research. Therefore, only couple of those are used in the experiments and you can see their spectrograms in Figure 4.6. Bearing 3_2 is particularly challenging since multiple fault types are found in the end of the degradation tests.



(a) Bearing 2_1 - Inner race fault.

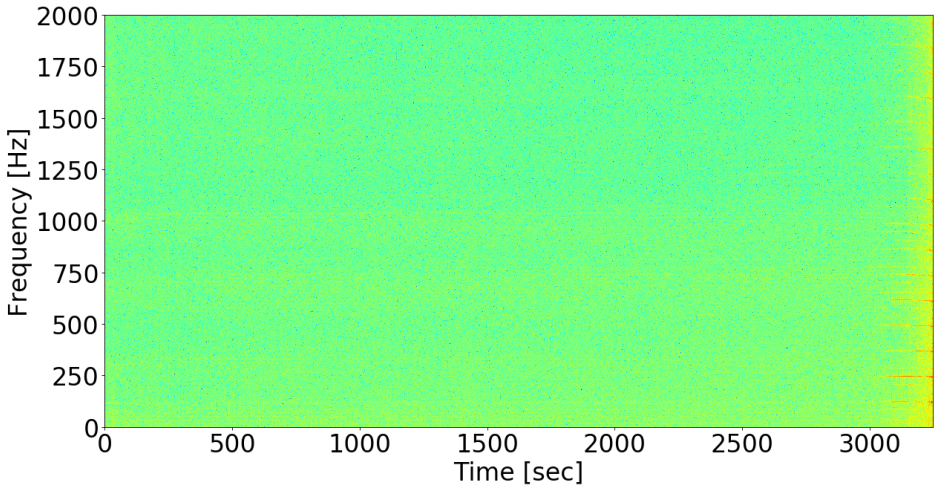


(b) Bearing 2_3 - Cage fault.

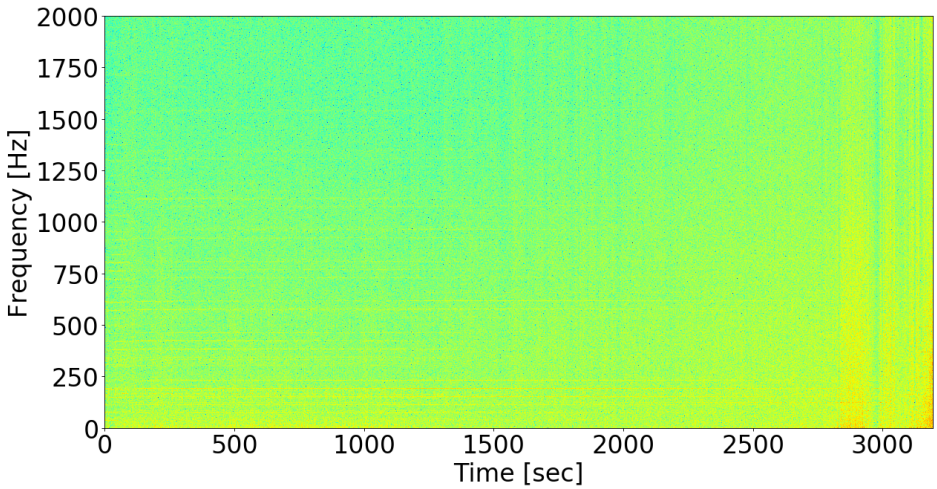


(c) Bearing 2_5 - Outer race fault.

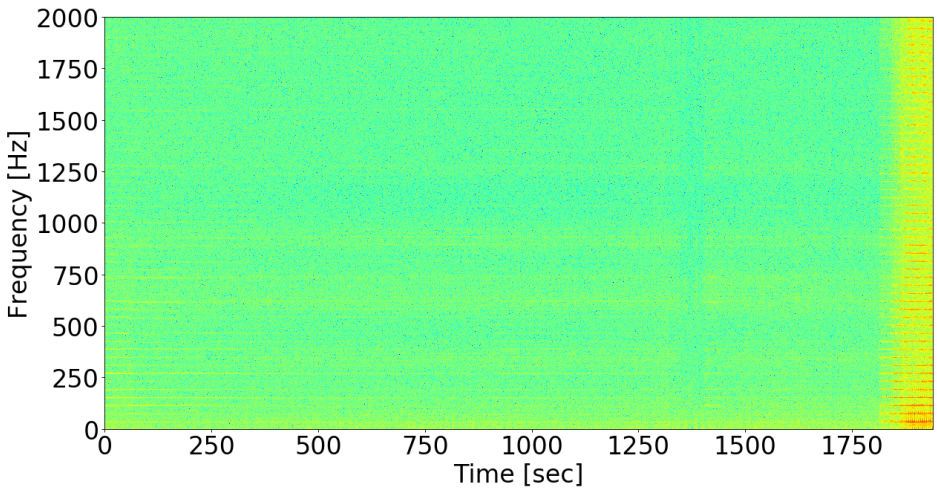
Figure 4.6: Spectrograms of pre-defined defects in XJTU-SY Dataset.



(d) Bearing 3_1 - Outer race fault.



(e) Bearing 3_2 - Every part of this bearing shows signs of defect.



(f) Bearing 3_4 - Inner race defect.

Figure 4.6: Spectrograms of pre-defined defects in XJTU-SY Dataset(cont.).

4.2. Experimental Design and Results

In Section 2.2.3, well-known feature extraction techniques are given. They are calculated for each sample and only initial samples, which we assume as healthy, used for PCA. Usually first two components can explain healthy data, but first three components used to transform the whole data since it is necessary for some bearing samples. For each bearing vibration signal extracted features and their transforms are depicted in Appendix A.

After data is transformed novelty detection methods are applied and novelty scores are calculated. For each method the major novelty points are labeled as state changes. These major novelty points can simply be identified as the ones with highest novelty score in a certain window. t-SNE plots are depicted for these states. These plots are also useful for evaluation of each novelty detection method.

One of the reliable methods for fault diagnosis is supervised machine learning methods. Although bearing datasets are not labeled, they start from healthy state and they end up with a bearing fault. So the assumption in the experiments, first quarters of the signals are healthy and the last 40 frames are faulty. An RBF kernel based SVM [47] is trained for each single record independently using the aforementioned healthy and faulty segments of that bearing recording as training data. Then each recording is temporarily labeled using that recordings classifier output probabilities, probability of being faulty is calculated for each sample. The first sample with 0.8 probability is designated as the start of bearing defect. In the end three different segments for each signal is specified: healthy, intermediate and faulty.

Both IMS and XJTU-SY datasets have vibration samples that are captured during bearing degradation tests. Although we are certain about the health state of a bearing in the end of an experiment, these datasets do not include any categorical information about this state during the experiments. Since there aren't any labels the comparison can only be made with other methods in the literature or more conventionally, by looking at the frequency spectrum and characteristic frequencies. The

spectrograms in Figure 4.4 for IMS dataset and in Figure 4.6 for XJTU-SY dataset are used to compare the streaming novelty detection methods for early detection of bearing faults.

4.2.1. IMS Bearing Dataset

- *Dataset 1 - Bearing 3:* According to dataset description in [41] this bearing has inner race fault in the end of degradation experiments. This was confirmed by Gousseau et al. (2016) [46] and inner race defect could be detected as early as 29.2 days into the experiment in the same work. In Figure A.1 there is also a noticeable change 19 hours into the experiment, which could be a result of load change or some external effect. This one was not addressed in [46].

  In each method different novelty scores are calculated and highest peaks in certain horizontal distance from other neighbouring peaks are designated as novelty points. These points are reported in each figure.

  SPLL method in Figure 4.9b is able to find the early change at Day 1, but it failed in the detection of actual inner race defect. However, in Figure 4.9c Bayesian method is able to detect the bearing fault at 29 day 3 hour into the experiment. This is slightly better than what is reported in [46].

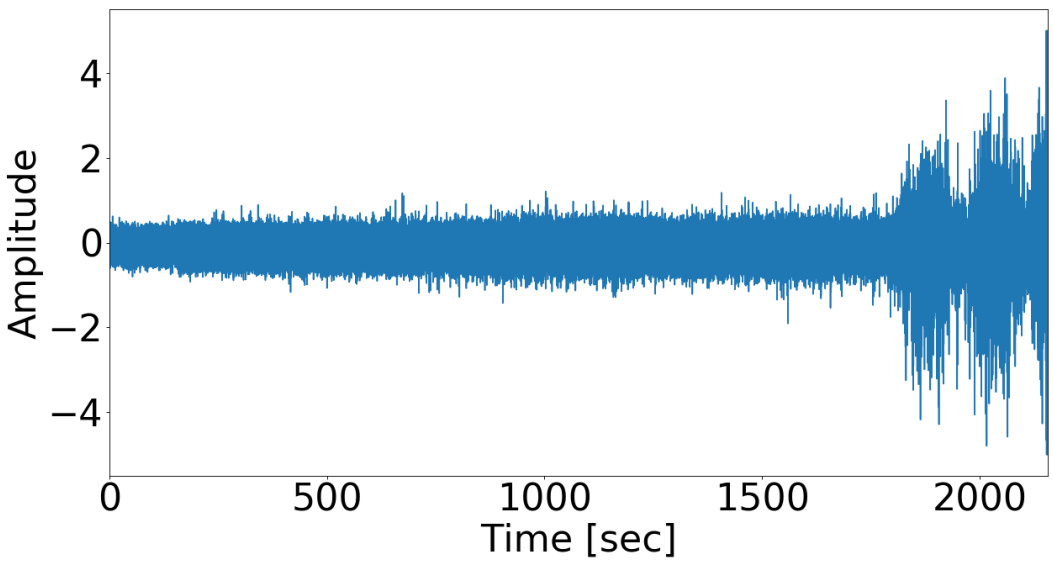


Figure 4.7: Raw vibration data of Dataset 1 - Bearing 3.

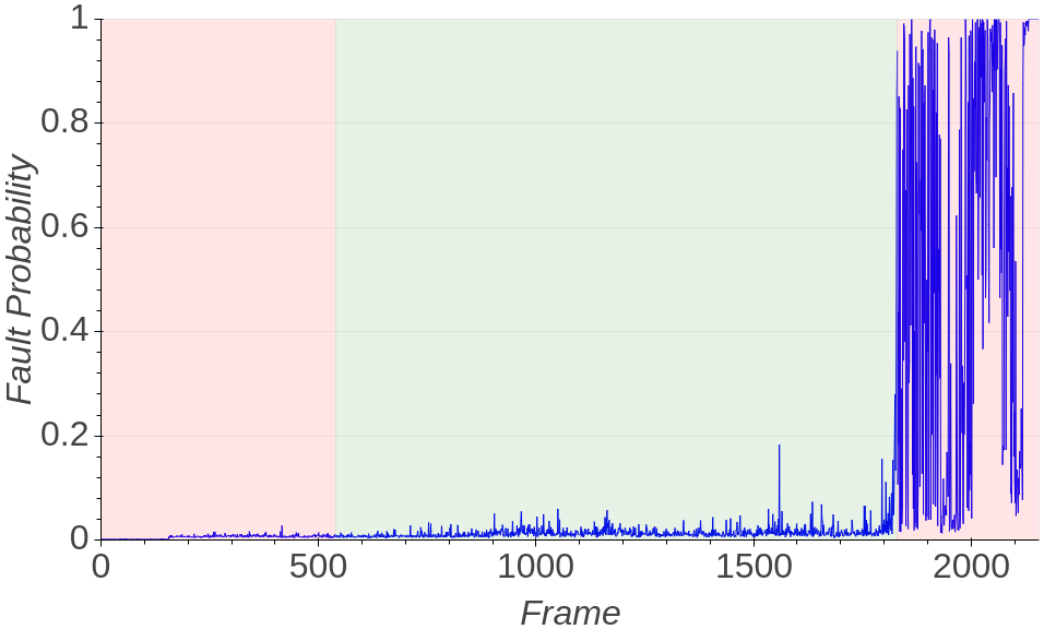
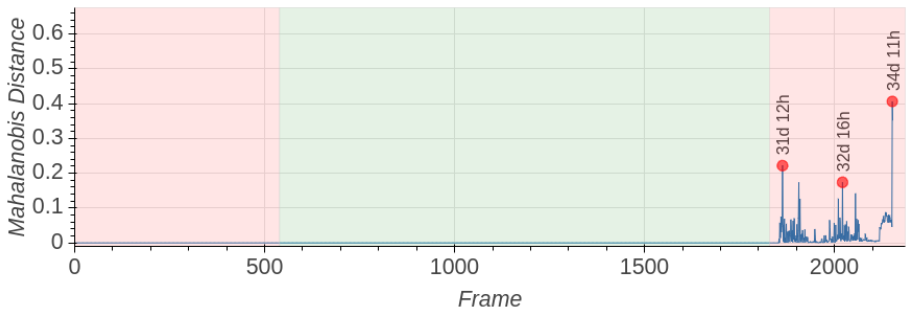
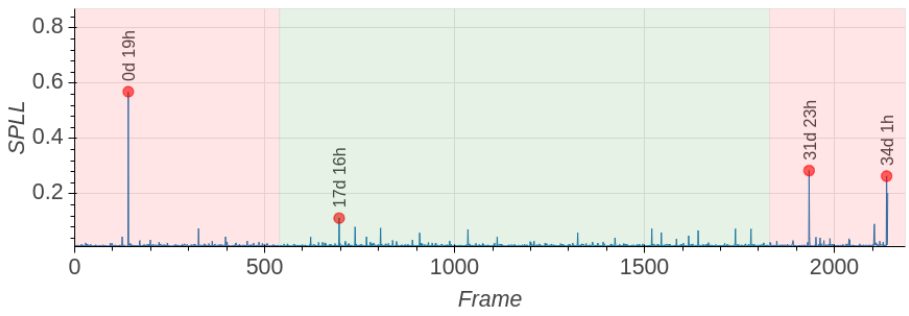


Figure 4.8: *Segments for Dataset 1 - Bearing 3*: Frame 1829 is the first sample with probability higher than 0.8. First 539 frames had been labeled as healthy and they were used to train SVM classifier.

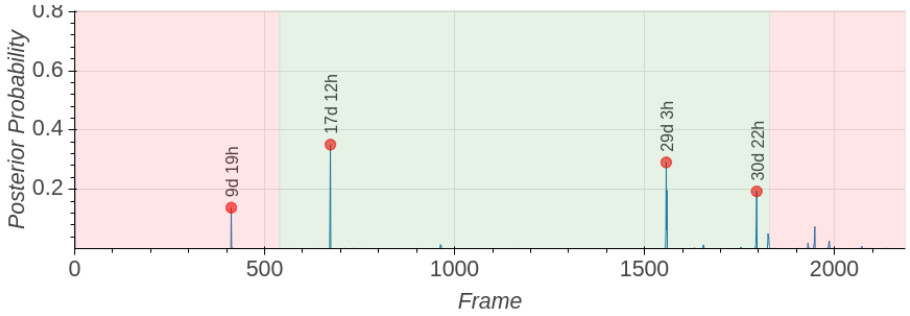


(a) Mahalanobis

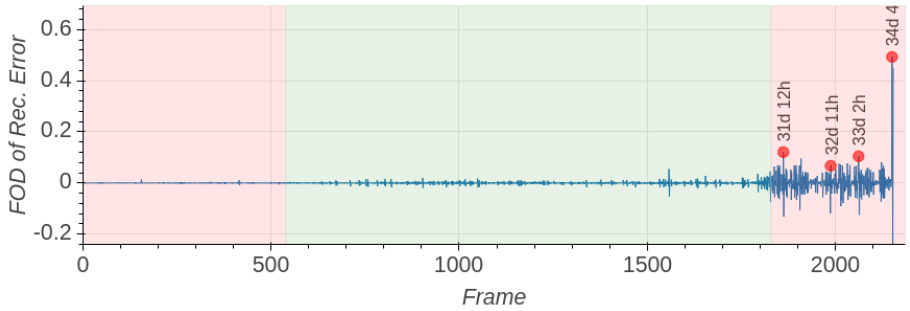


(b) SPLL

Figure 4.9: *Novelty detection scores for Dataset 1 - Bearing 3:* Bayesian method outperformed others in terms of detection of inner race defect. SPLL method is the only one that is able to detect the change in Day 1.



(c) Bayesian

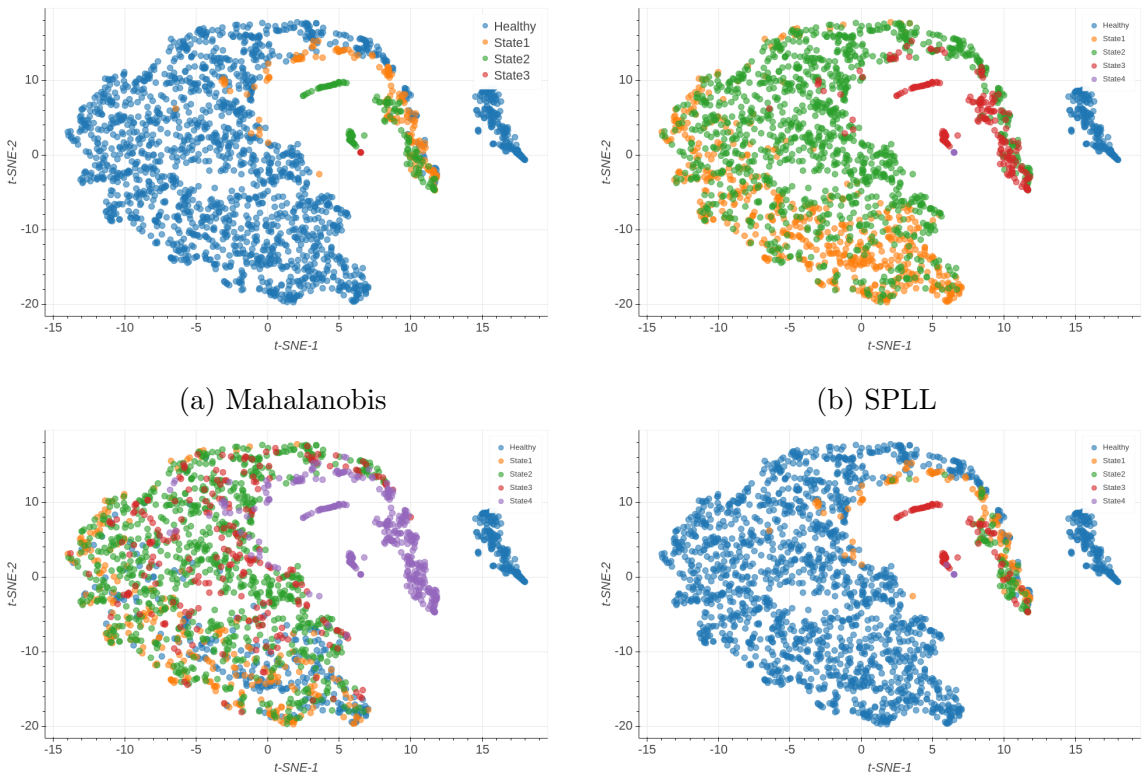


(d) LSTM-Autoencoder

Figure 4.9: *Novelty detection scores for Dataset 1 - Bearing 3:* Bayesian method outperformed others in terms of detection of inner race defect. SPLL method is the only one that is able to detect the change in Day 1(cont.).

Table 4.7: Number of novelties and frame numbers detected by each novelty detection method in different segments.

	Healthy	Intermediate	Faulty
Mahalanobis	-	-	3 *(1863,2021,2152)*
SPLL	1 *(141)*	1 *(619)*	2 *(1933,2137)*
Bayesian	1 *(412)*	3 *(673, 1557,1824)*	-
LSTM-Autoencoder	-	-	4 *(1862,1988,2062,2151)*



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.10: *t-SNE visualizations for Dataset 1 - Bearing 3:* In Figure 4.10b change-point at Day 1 can be confirmed by clear distinction between healthy and other samples. Mahalanobis and LSTM-Autoencoder are also found anomalies late at the experiment, but SPLL outperformed them in the earlier change-point at Day 1. *State 3(red)* samples in Figure 4.10c do not form any cluster although they were assumed to have early signs of inner race defect.

- *Dataset 1 - Bearing 4:* This time defect becomes clear around Day 28 and it can be detected with every method in the present work. However, in [46] the earliest detection can be made around Day 18, which is also achieved with SPLL, Bayesian and LSTM-Autoencoder models as well. There is again a novelty point around 19 hour mark, which might be explained with external impact. There is a break at around that point where data acquisition is stopped. This point is also observable in Bearing 3, which is essentially in the same test bed. Therefore it is likely that bearing defect does not have a role in this change-point.

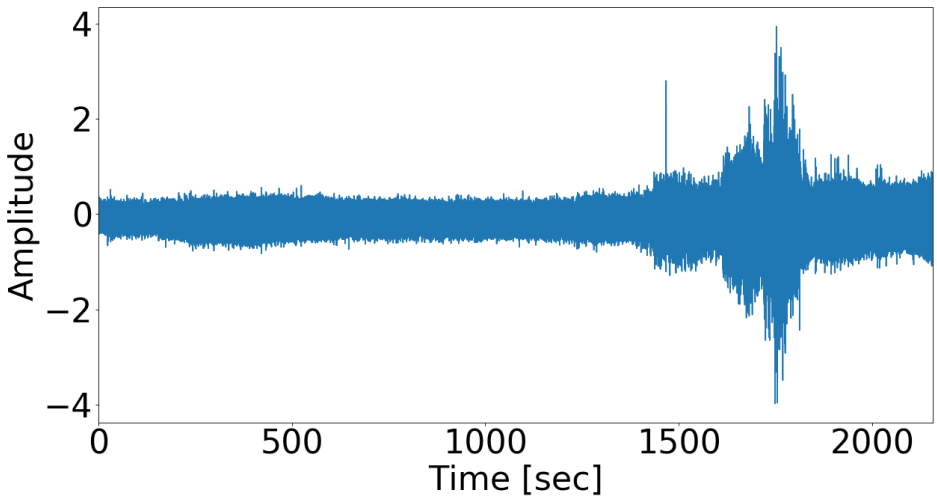


Figure 4.11: Raw vibration signal of Dataset 1 - Bearing 4.

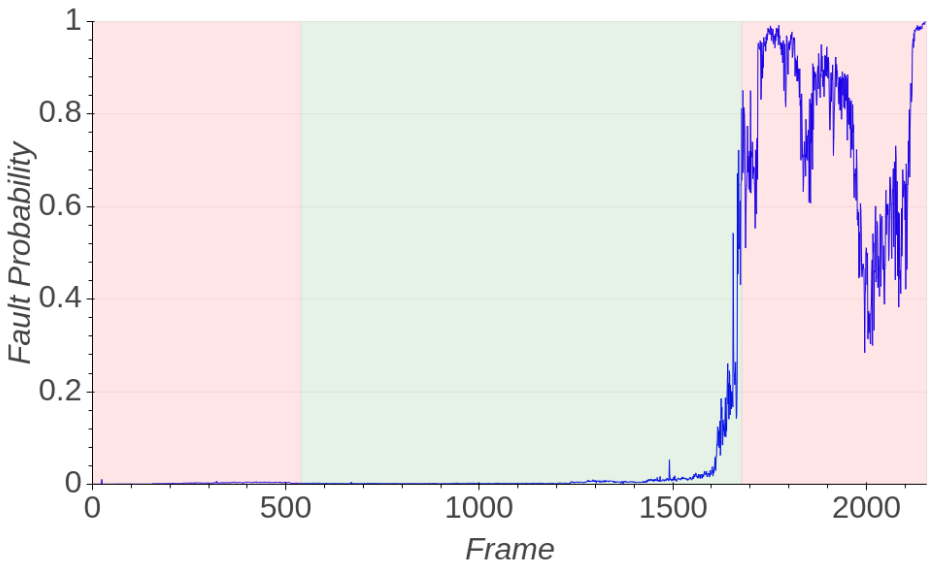
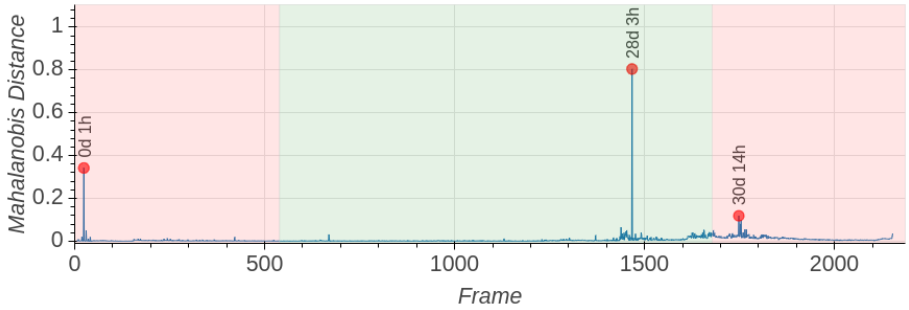
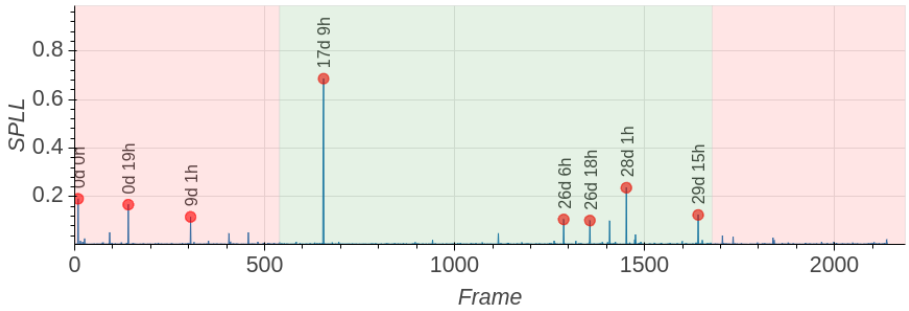


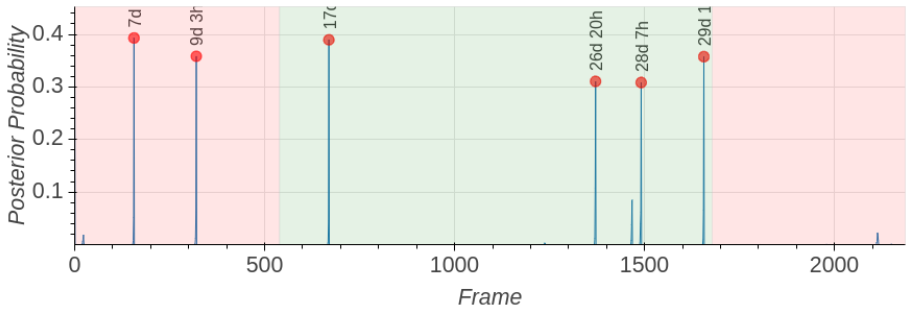
Figure 4.12: *Segments for Dataset 1 - Bearing 4*: Frame 1678 is the first sample with probability higher than 0.8. First 539 frames had been labeled as healthy and used to train SVM classifier.



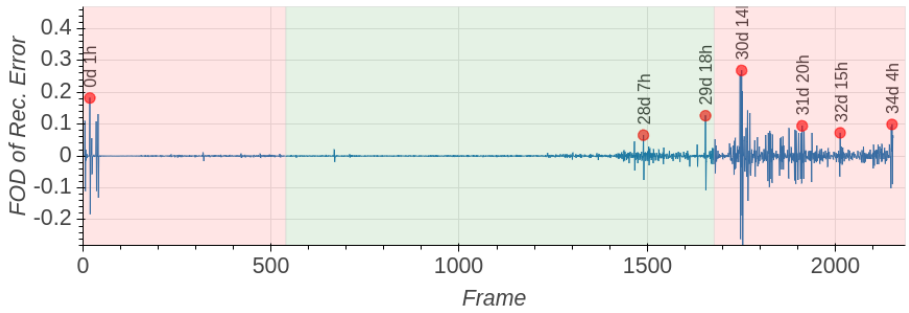
(a) Mahalanobis



(b) SPLL



(c) Bayesian

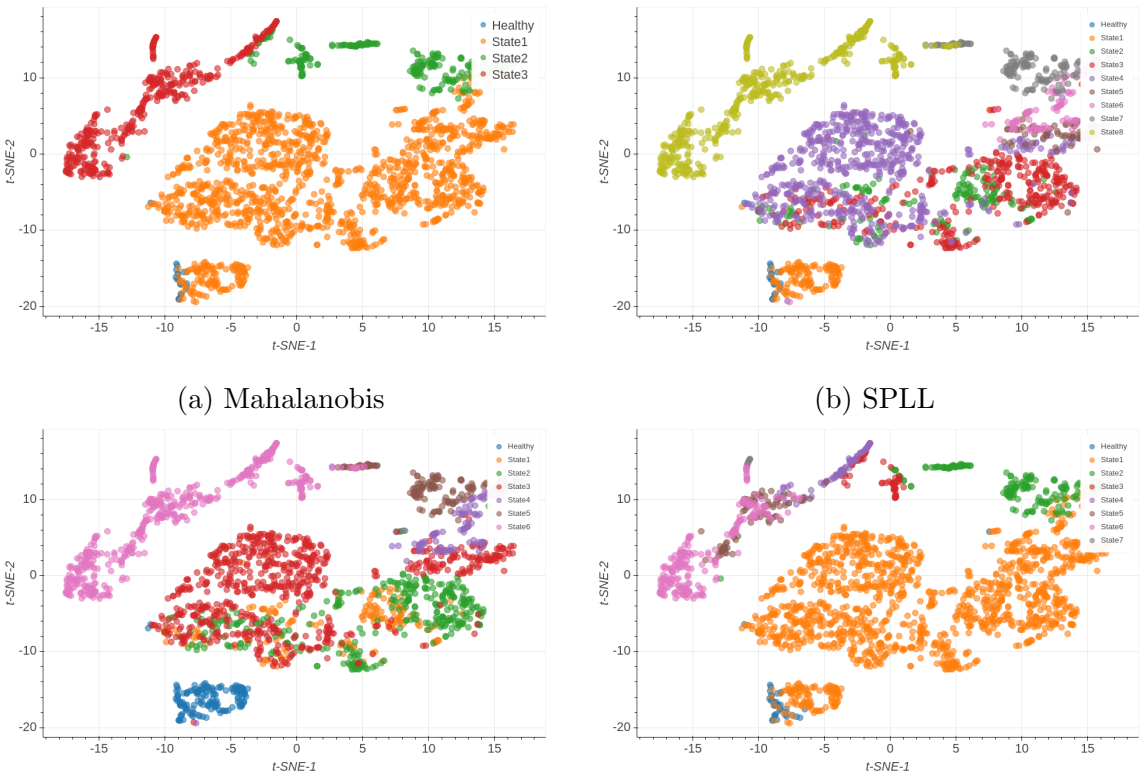


(d) LSTM-Autoencoder

Figure 4.13: *Novelty detection scores for Dataset 1 - Bearing 4:* Earlier distortions in the features which can be seen in Figure A.2a. These distortions are detected as novelties. SPLL and Bayesian methods outperformed other two methods by detecting the defect at Day 18, which is also confirmed in [46].

Table 4.8: Number of novelties and frame numbers detected by each novelty detection method in different segments.

	Healthy	Intermediate	Faulty
Mahalanobis	1 *(24)*	1 *(1467)*	1 *(1748)*
SPLL	3 *(9,141,305)*	5 *(655,1287,1356, 1452,1641)*	-
Bayesian	2 *(156,320)*	4 *(669,1371, 1491,1656)*	-
LSTM-Autoencoder	1 *(40)*	2 *(1466, 1650)*	4 *(1680,1751, 1800,2151)*



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.14: *t-SNE visualizations of each method for Dataset 1 - Bearing 4:* The earlier novelties detected by Mahalanobis, SPLL and LSTM-Autoencoder are clearly false-positives. Bayesian meth- od finds the healthy samples correctly. In Figure 4.14b the transition between *State 3(red)* and *State 4(purple)* seems to be correct. Likewise, in Figure 4.14c the transition between *State 2(green)* and *State 3(red)* is a similar case, which confirms the novelty point at Day 18.

- *Dataset 2 - Bearing 1:* Bearing defect is easily recognizable from the spectrogram 4.4b in that case, however the starting point can be determined more accurately. All methods are able to detect novelty at around frame 700(4 days 18 hours into the experiment), but Gousseau et al. (2016) reported 3.5 day as the earliest detection point [46].

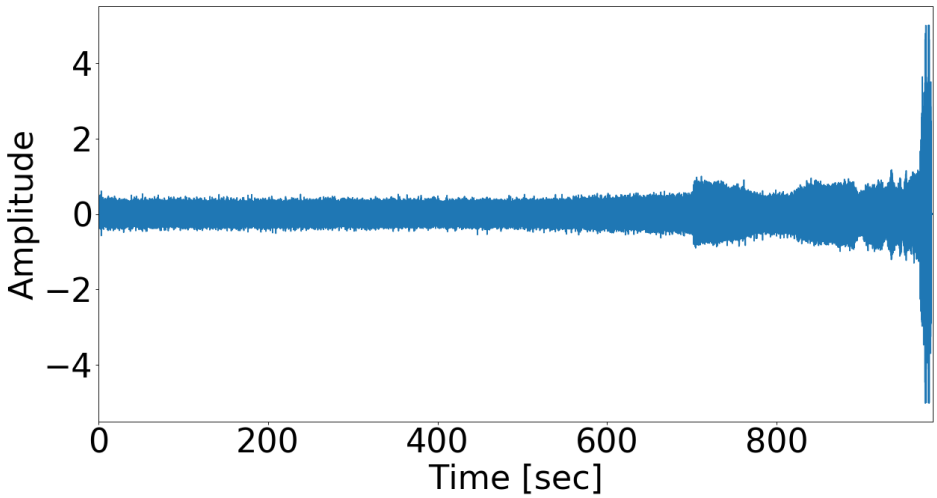


Figure 4.15: Raw vibration signal of Dataset 2 - Bearing 1.

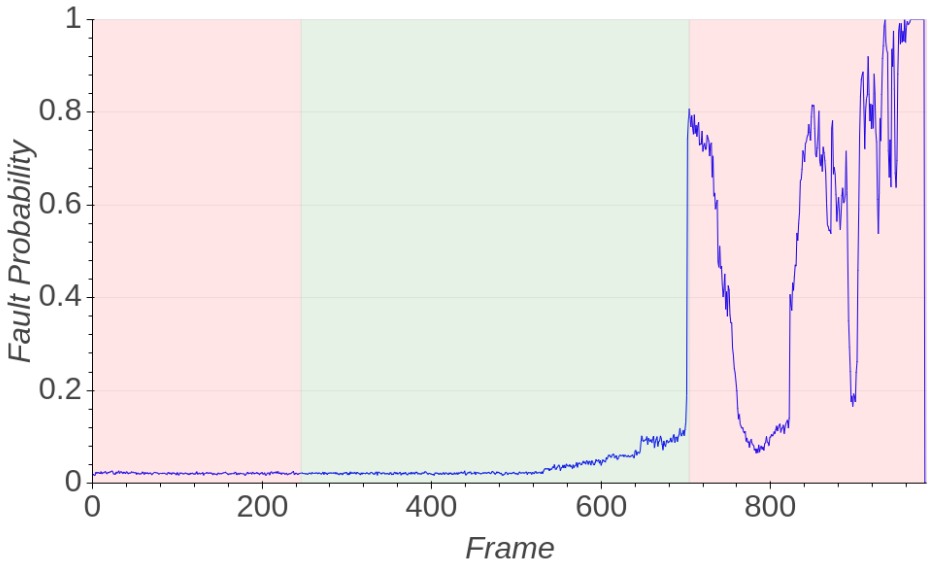
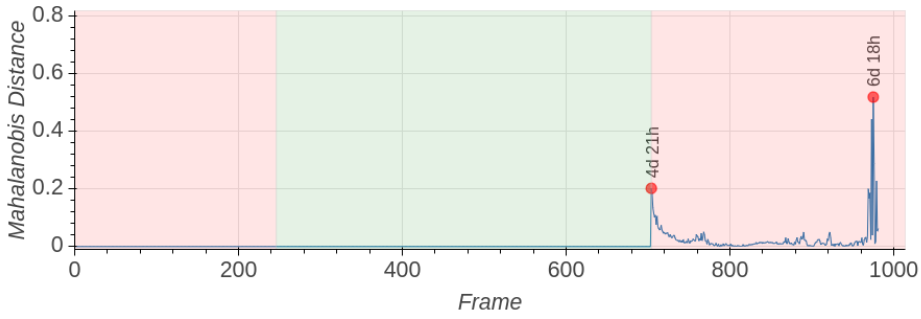
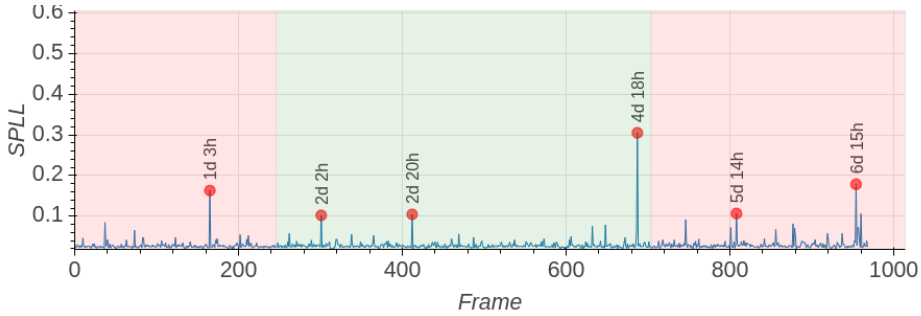


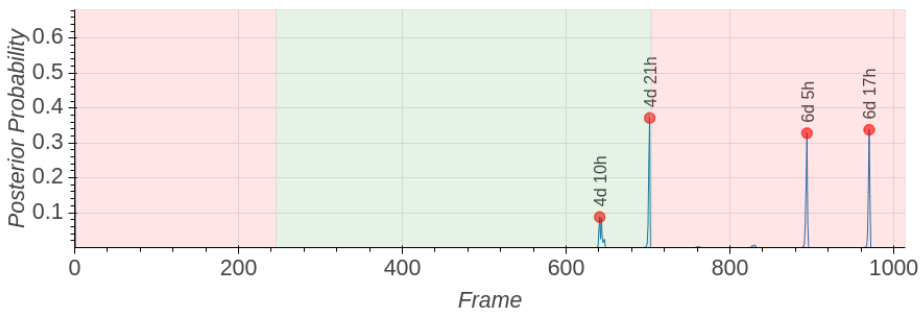
Figure 4.16: *Segments for Dataset 2 - Bearing 1*: At Frame 704 anticipated threshold was reached. Frames before 246 were labeled as healthy samples in order to train the classifier.



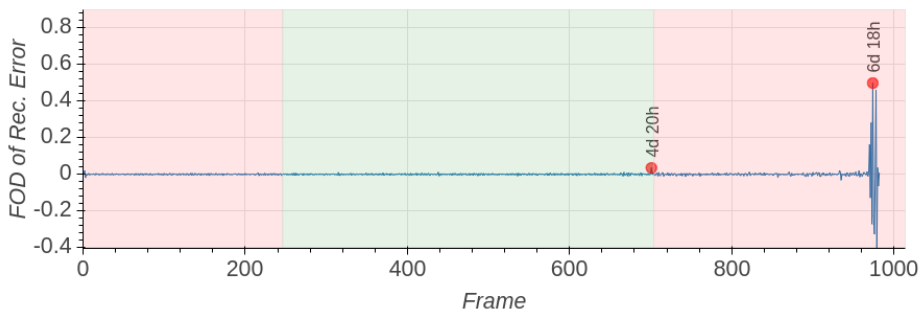
(a) Mahalanobis



(b) SPLL



(c) Bayesian

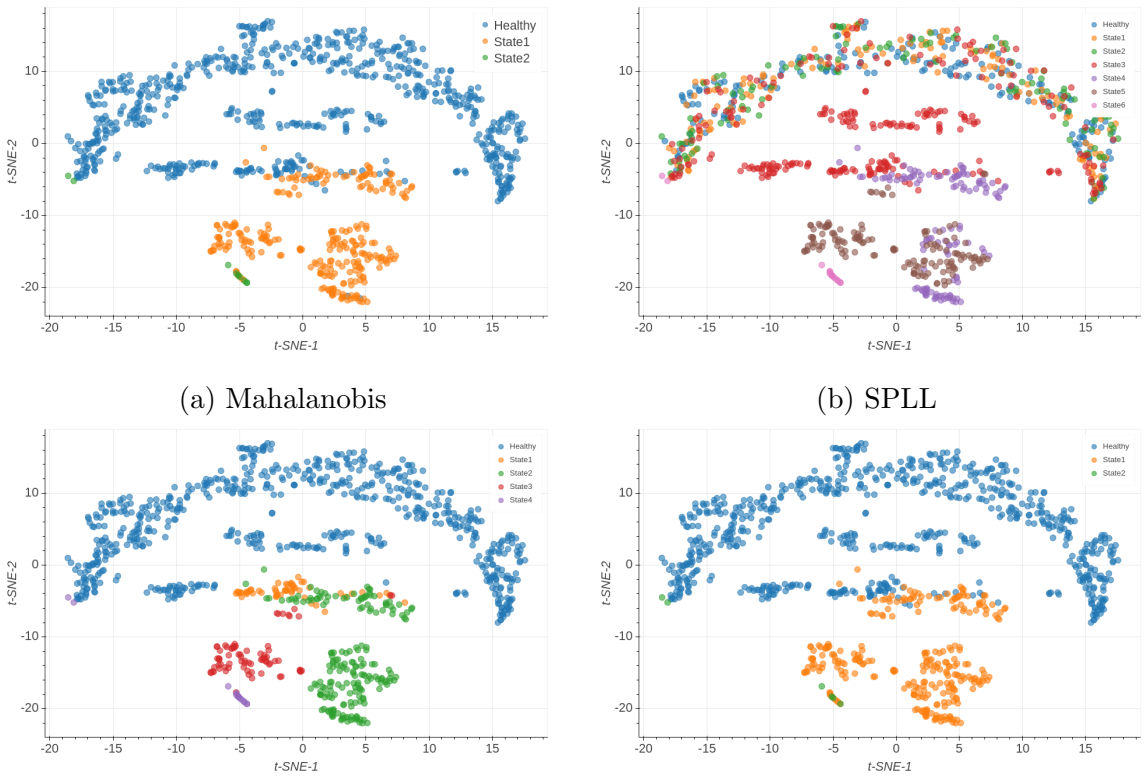


(d) LSTM-Autoencoder

Figure 4.17: *Novelty detection scores for Dataset 2 - Bearing 1:* Outer race novelty can easily be detected in this case, however further preprocessing is required for earlier diagnosis. Although SPLL method found some novelties at Day 2 and Day 3, from Figure 4.18b, it is clear that these points are false-positives.

Table 4.9: Number of novelties and frame numbers detected by each novelty detection method in different segments.

	Healthy	Intermediate	Faulty
Mahalanobis	-	-	2 *(704,975)*
SPLL	1 *(165)*	3 *(301,412,687)*	2 *(808,954)*
Bayesian	-	2 *(647,702)*	2 *(895,970)*
LSTM-Autoencoder	-	1 *(638)*	1 *(974)*



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.18: *t-SNE visualizations of each method for Dataset 2 - Bearing 1:* Mahalanobis and LSTM-Autoencoder methods ended up with almost identical results, but there is a clear distinction between *State 2(green)* and *State 3(red)* in Figure 4.18c. This implies another novelty at Day 7. SPLL method has clear false-positives earlier although it is able to detect outer race defects later on.

4.2.2. XJTU-SY Dataset

- *Bearing2_1*: All methods resulted in novelty detection at 9 minutes into the experiment, where stream clearly changes its state.

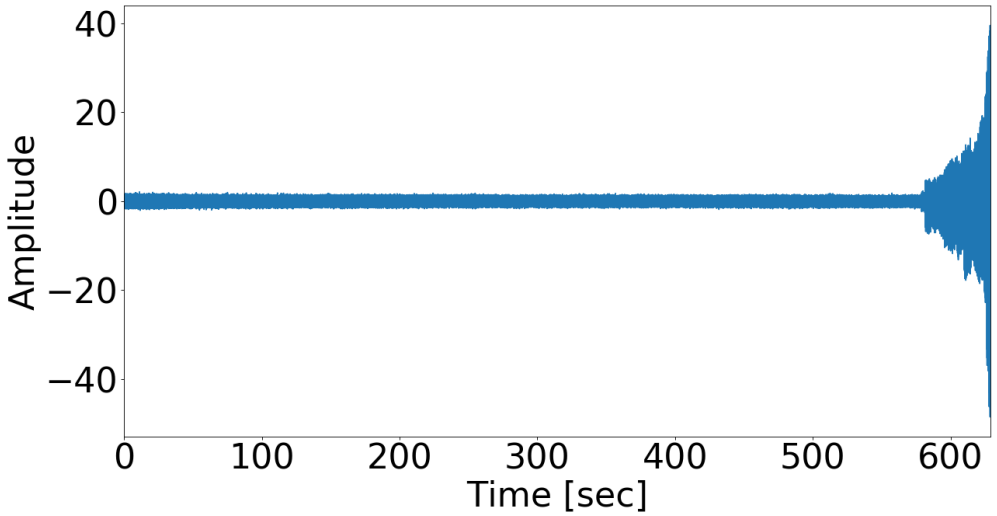


Figure 4.19: Raw vibration signal of Bearing2_1.

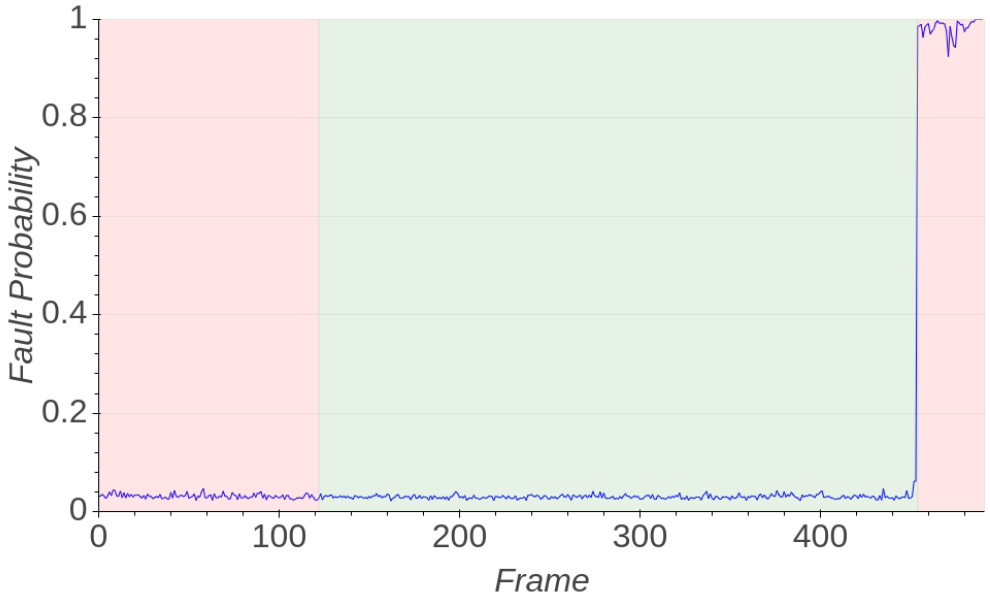
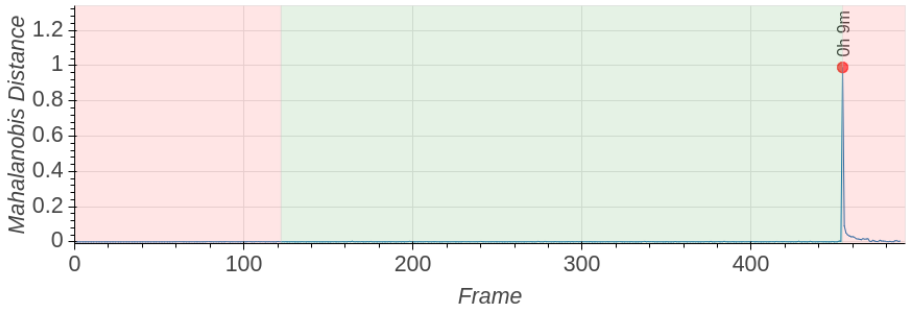
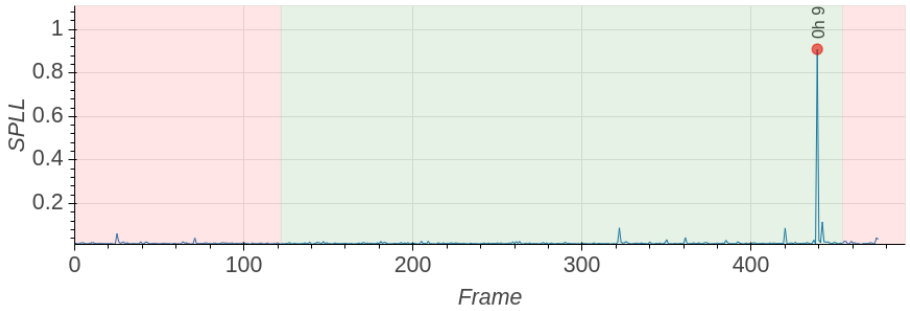


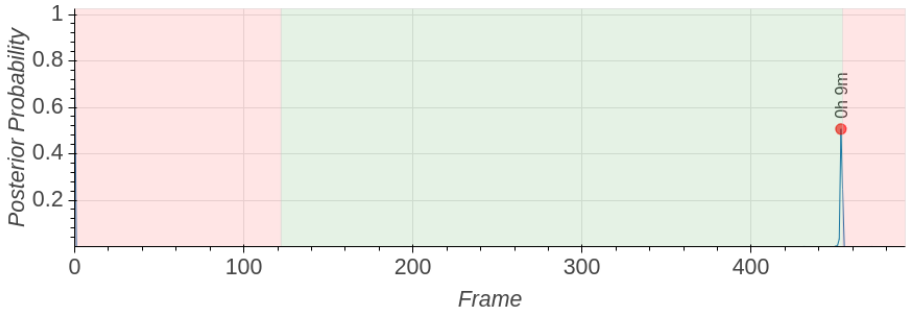
Figure 4.20: *Segments for Dataset 2 - Bearing 1*: This bearing has the obviuos fault at Frame 454, which was confirmed by the classifier result.



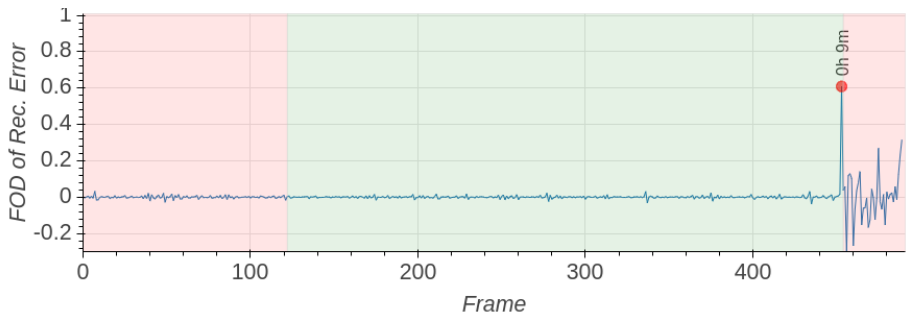
(a) Mahalanobis



(b) SPLL



(c) Bayesian



(d) LSTM-Autoencoder

Figure 4.21: *Novelty detection scores for Bearing2_1*: Results are exactly the same. Obvious novelty has been found at 9 minute mark by each method.

Table 4.10: Each method ended up with the same novelty for Dataset 2 - Bearing 1 of XJTU-SY Dataset.

	Healthy	Intermediate	Faulty
Mahalanobis	-	1 *(453)*	-
SPLL	-	1 *(453)*	-
Bayesian	-	1 *(453)*	-
LSTM-Autoencoder	-	1 *(453)*	-

(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.22: *t-SNE visualizations of each method for Bearing2_1*: After fault occurs, samples form a different cluster immediately.

- *Bearing2_3:* Cage fault is normally harder to detect than other defects. The results suggests two novelty zones. The initial one is between 2-3 minutes and the latter one is between 6-7 minutes into the experiment. LSTM-Autoencoder cannot detect any of these and only has a false-positive.

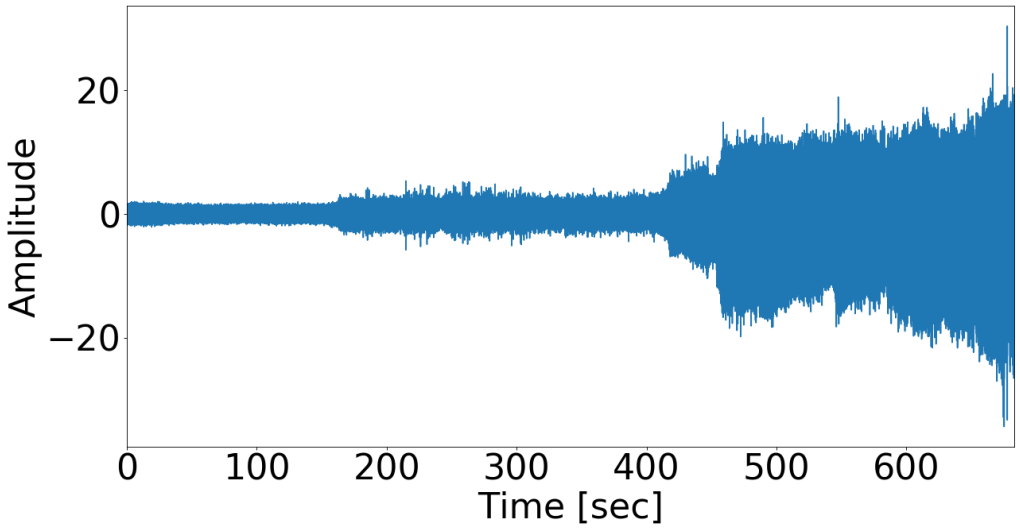


Figure 4.23: Raw vibration signal of Bearing2_3.

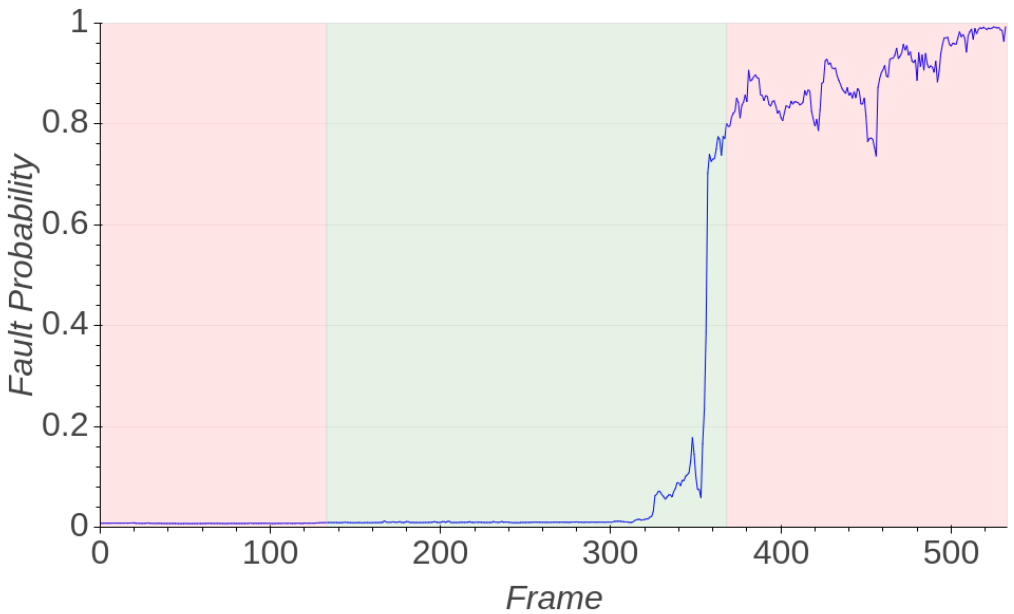
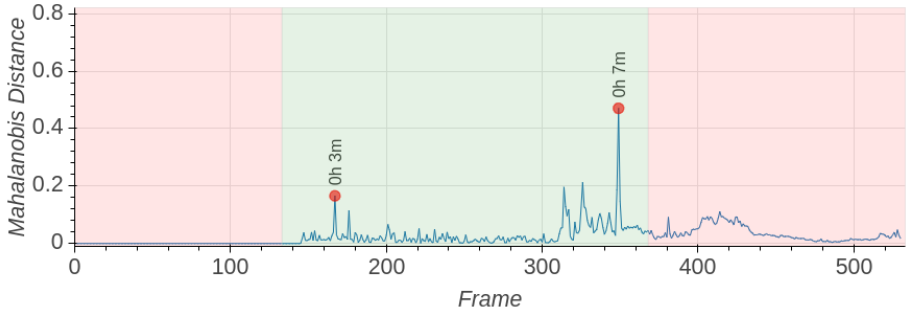
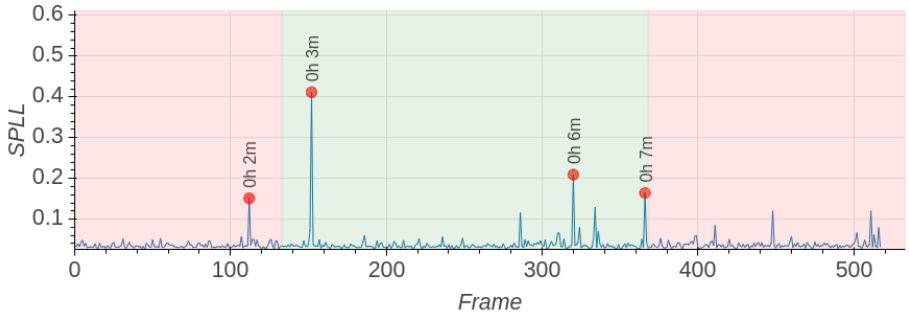


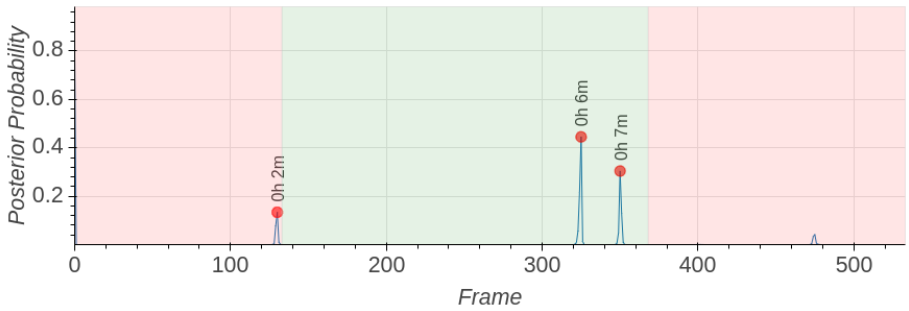
Figure 4.24: *Segments for Dataset 2 - Bearing 3*: Until Frame 133 the samples are assumed to be healthy, the fault starts at Frame 368 according to the classifier.



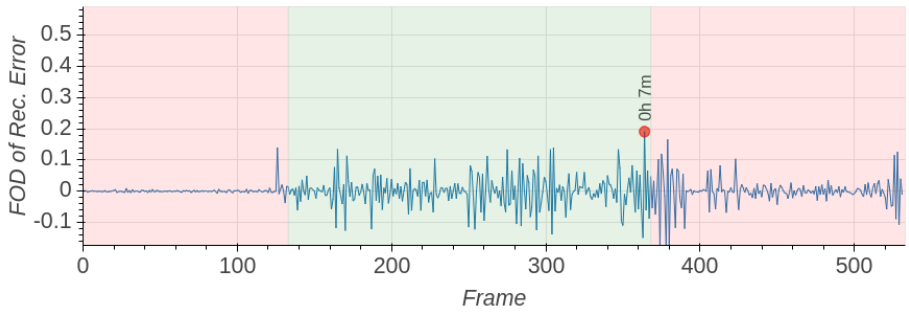
(a) Mahalanobis



(b) SPLL



(c) Bayesian

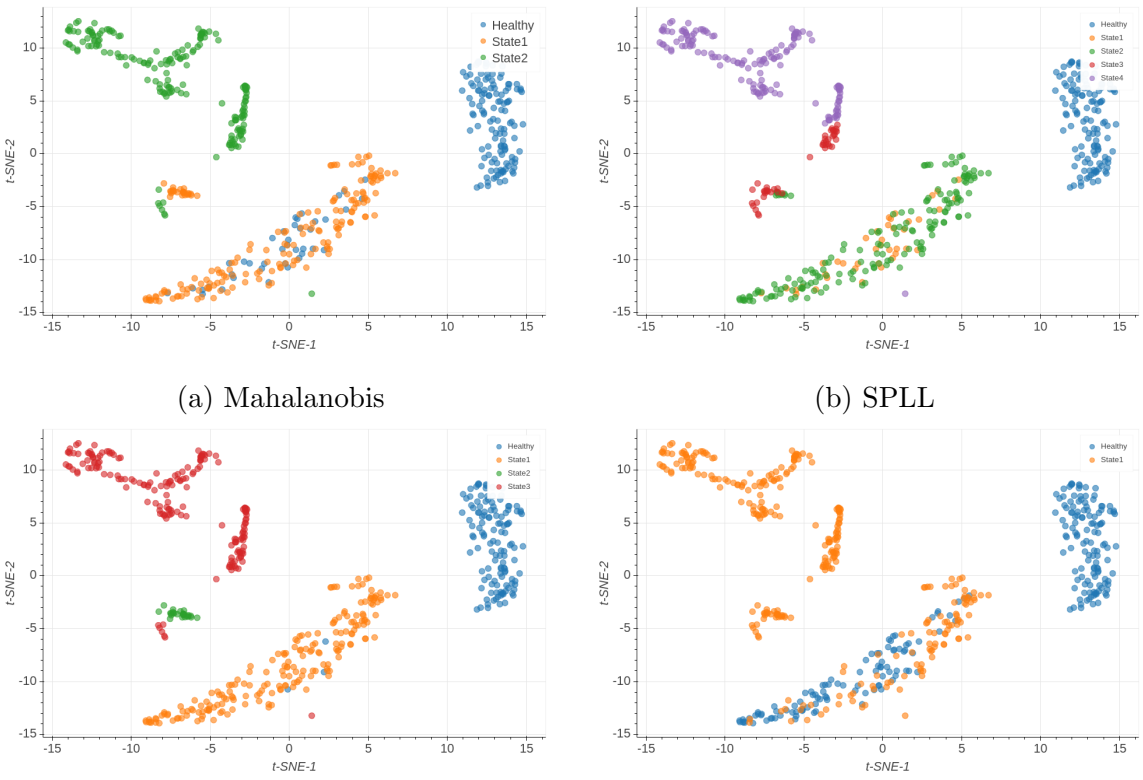


(d) LSTM-Autoencoder

Figure 4.25: *Novelty detection scores for Bearing2_3:* Novelty at 6 minute mark can easily be seen in Figure 4.6b as well. Earlier novelty may not be distinguished from the spectrogram, but it is detected using novelty detection methods.

Table 4.11: There aren't any novelties in the faulty stage. There might be no degradation after fault occurs. It might happened instantly and faulty state might not have become worse as the test continues. However there are multiple novelties found in the intermediate stage.

	Healthy	Intermediate	Faulty
Mahalanobis	-	2 *(167,349)*	-
SPLL	1 *(112)*	3 *(152,320,366)*	-
Bayesian	1 *(130)*	2 *(320,325)*	-
LSTM-Autoencoder	-	1 *(364)*	-



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.26: *t-SNE visualizations of each method for Bearing2_3*: Although first three methods give close results, Bayesian method is more accurate than SPLL and Mahalanobis when clusters are considered in t-SNE plots. In 4.26c, there are less incorrectly marked samples in each cluster.

- *Bearing2_5:* In Figure 4.27 and A.6a, it could be clearly seen that severity level increases in time. Except first novelty found by Mahalanobis algorithm, it seems that novelty points are detected correctly.

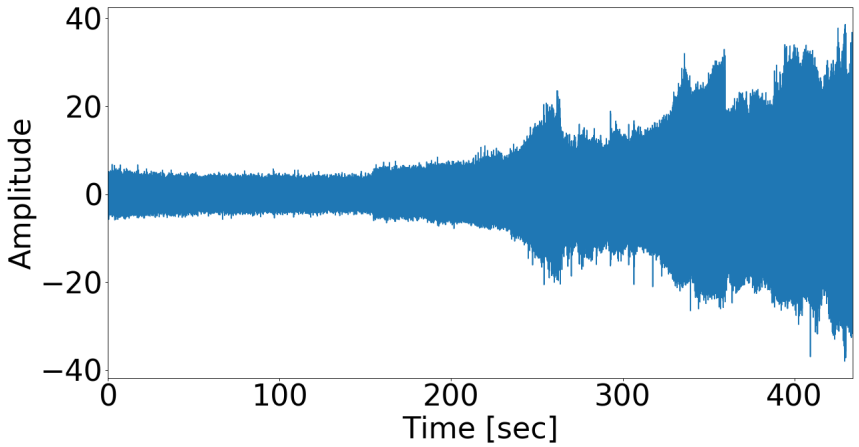


Figure 4.27: Raw vibration signal of Bearing2_5.

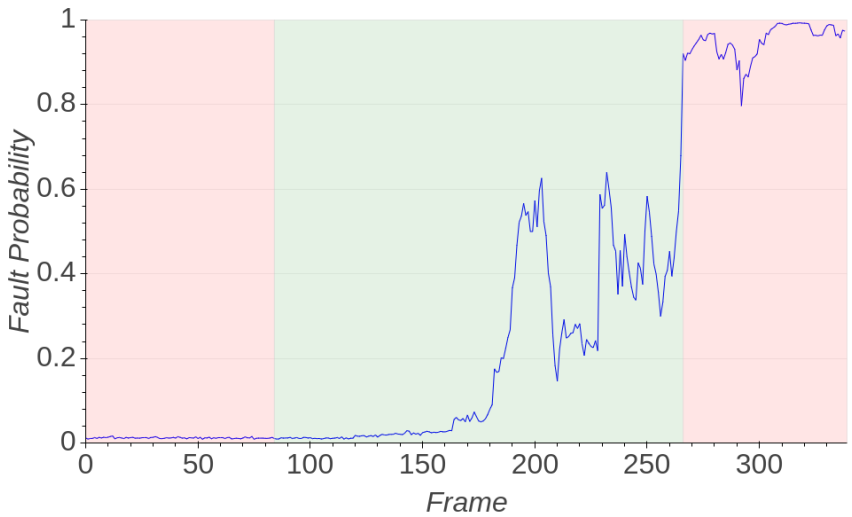
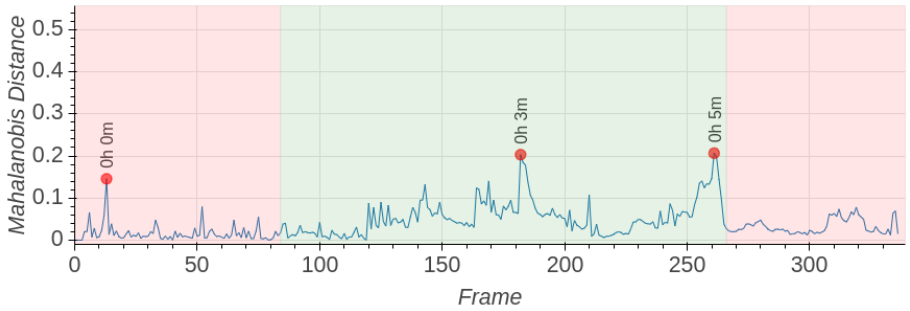
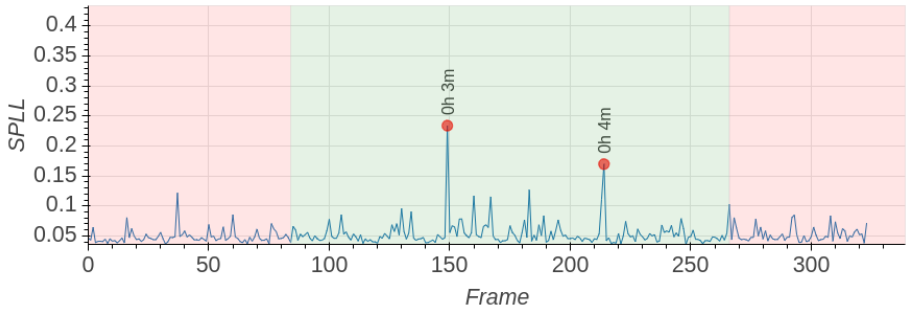


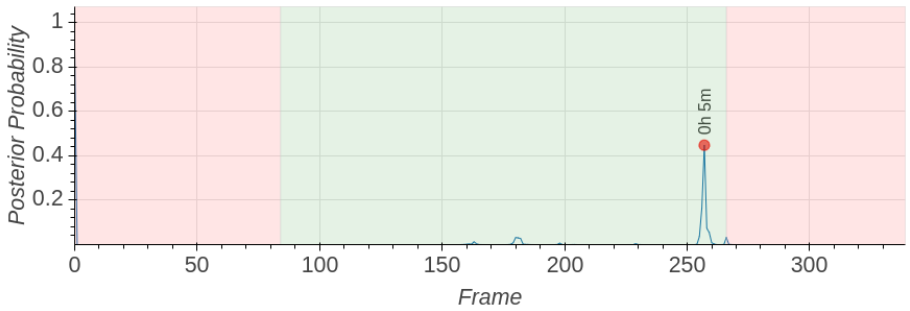
Figure 4.28: *Segments for Dataset 2 - Bearing 5*: In the intermediate stage the probability of being a faulty sample increases gradually unlike the previous bearings. This mean there is a chance for early detection. Samples until Frame 84 were assumed to be healthy and classifier detected Frame 266 as the first frame with at least 0.8 probability of being unhealthy.



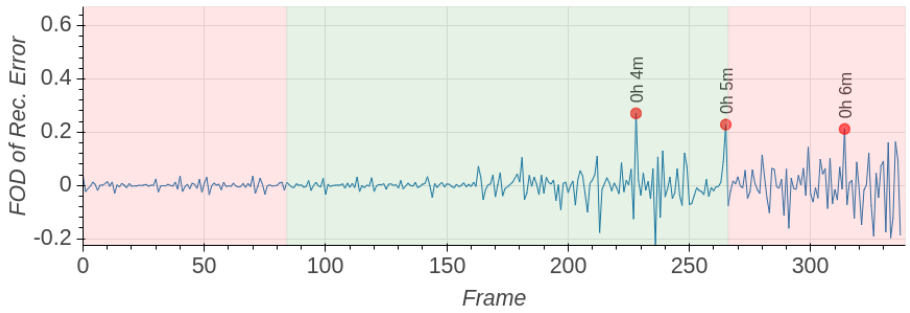
(a) Mahalanobis



(b) SPLL



(c) Bayesian

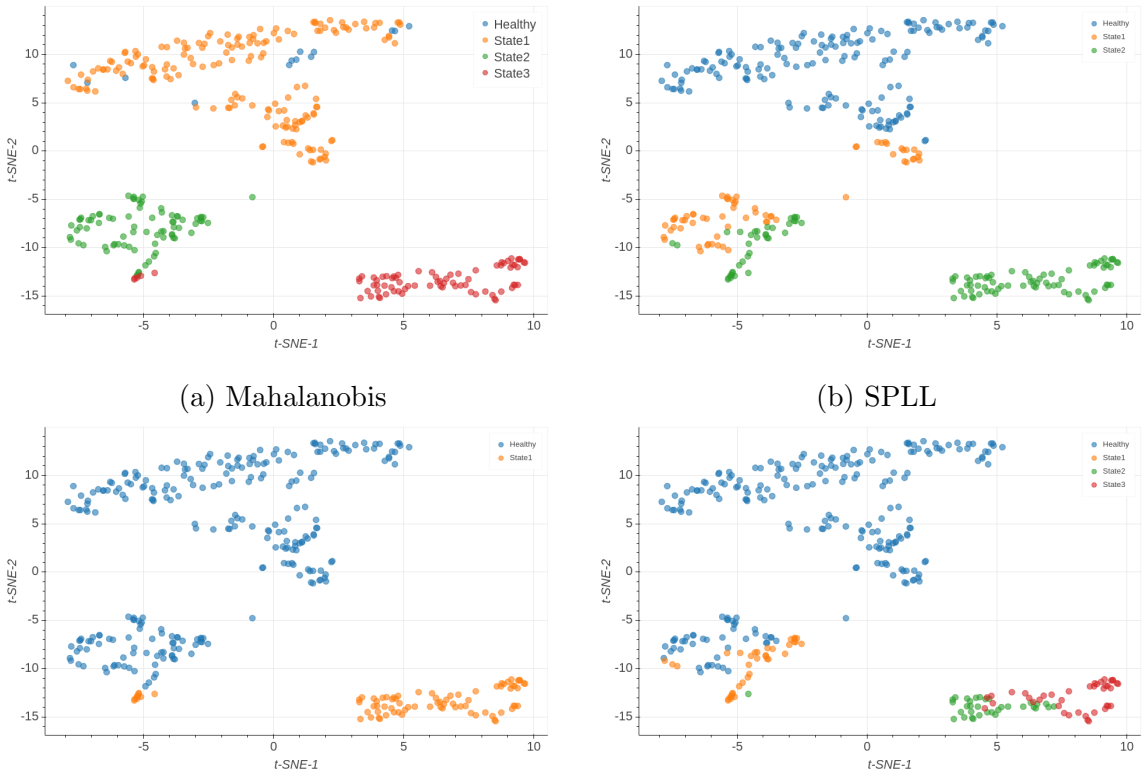


(d) LSTM-Autoencoder

Figure 4.29: *Novelty detection scores for Bearing2_5:* Mahalanobis and SPLL methods are able to detect the first novelty around 3 minute mark. The later ones were also being detected by other methods. LSTM-Autoencoder also explains the severity change after initial novelty.

Table 4.12: As the raw vibration data and classification probabilities suggest there are changes during the intermediate stage. Most of the novelties were detected in that region.

	Healthy	Intermediate	Faulty
Mahalanobis	1 *(13)*	2 *(169,261)*	-
SPLL	-	2 *(149,214)*	-
Bayesian	-	1 *(257)*	-
LSTM-Autoencoder	-	2 *(228,265)*	1 *(299)*



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.30: *t-SNE visualizations of each method for Bearing2_5:* Other than first novelty, Mahalanobis method is able to detect other two transition points between the states. Second novelty was detected a little earlier than it should be by SPLL method. LSTM-Autoencoder correctly distinguishes samples captured later in the experiment. If you look at Figure 4.30d, between *State 2(green)* and *State 3(red)* there is a clear distinction in that cluster.

- *Bearing3_1:* This bearing has outer race defect at the end and this novelty was confirmed by each method as can be seen in Figure 4.33. Between 10 and 21 minutes into the experiment there are other novelties discovered by SPLL, Bayesian and LSTM-Autoencoder methods, which cannot be observed in Figure 4.31. Spectrogram of this bearing in Figure 4.6d suggests the same, however there is a change in variance in that interval if we look at Figure A.7 closely.

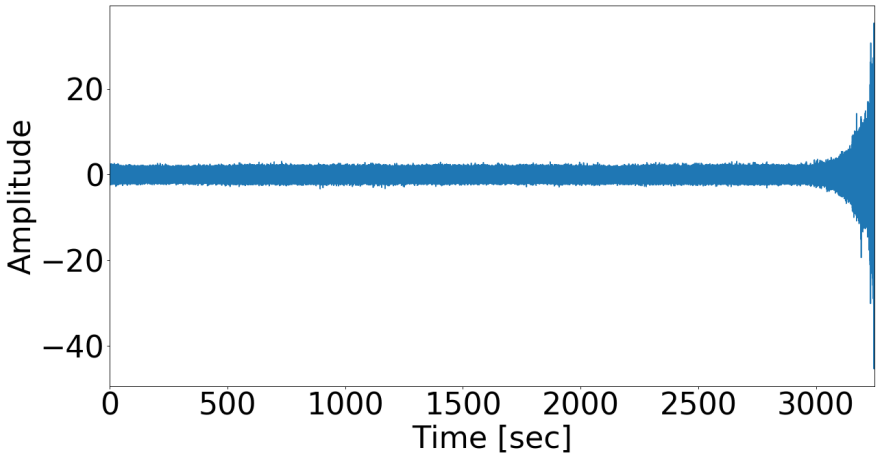


Figure 4.31: Raw vibration signal of Bearing3_1.

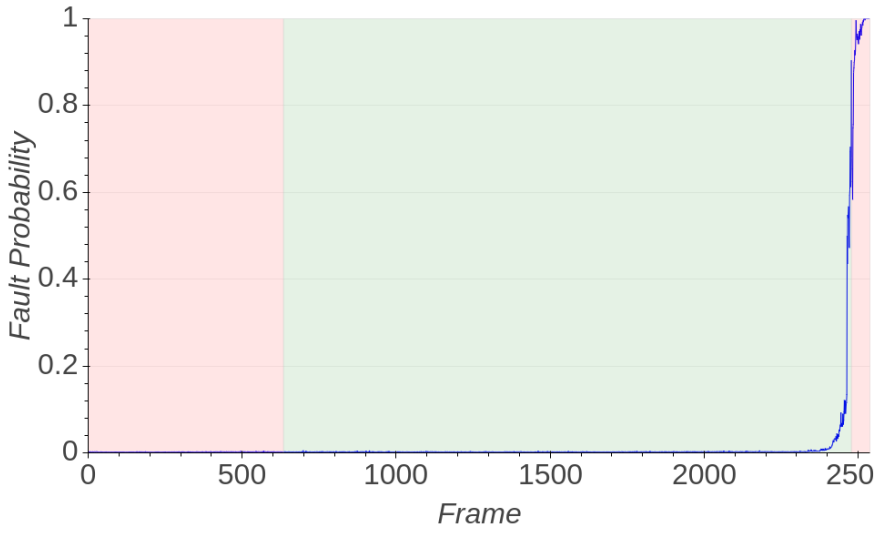
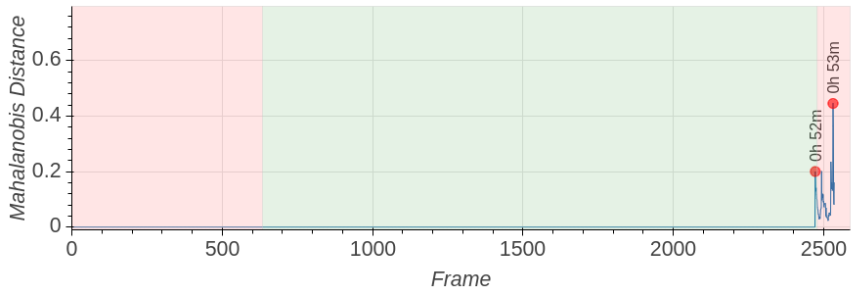
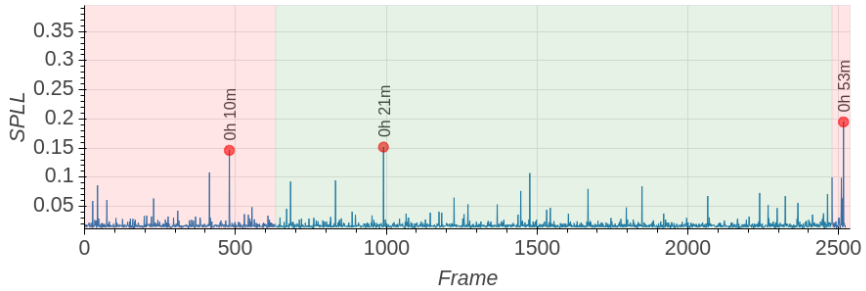


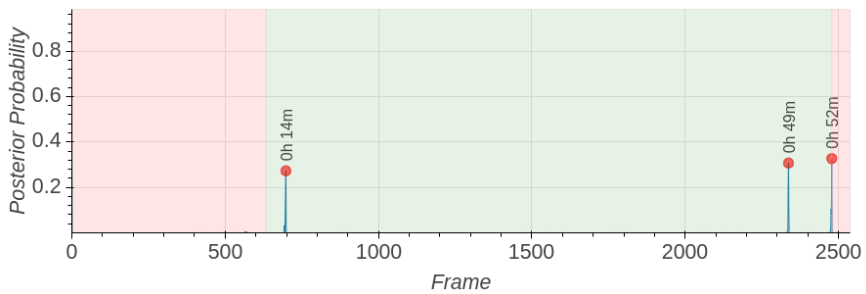
Figure 4.32: *Segments for Dataset 3 - Bearing 1*: Healthy stage seems to continue for a while in this bearing. The assumption is again the first quarter of data is healthy(until frame 634). The classifier results show that the probability reaches the 0.8 threshold at Frame 2478.



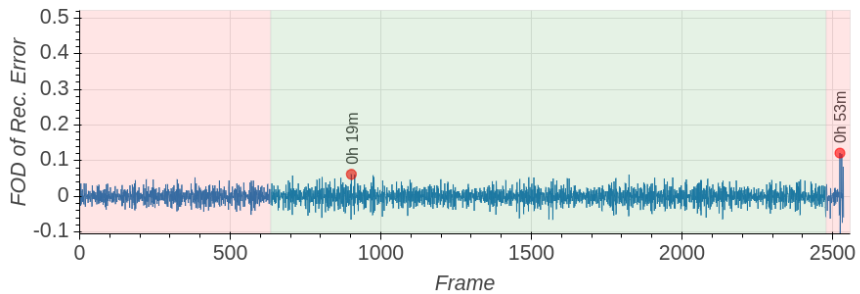
(a) Mahalanobis



(b) SPLL



(c) Bayesian

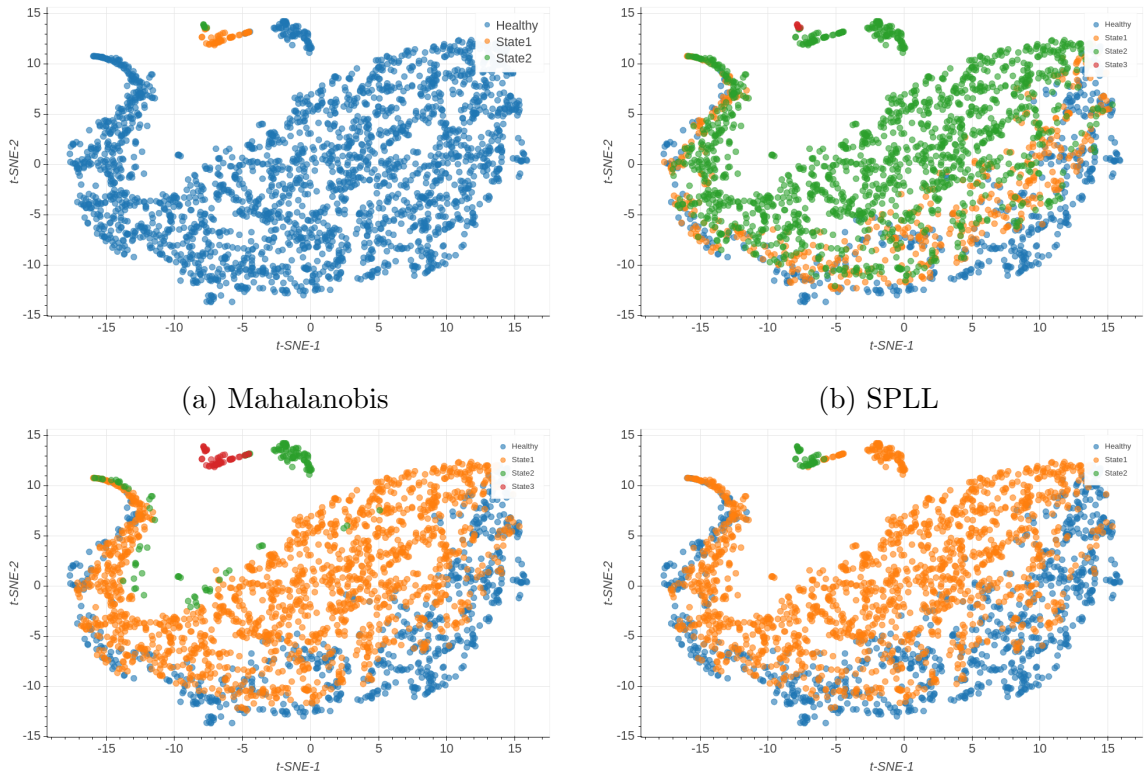


(d) LSTM-Autoencoder

Figure 4.33: *Novelty detection scores for Bearing3_1*: Although there are some early novelties found in 10-21 minutes interval, the actual defect starts at Minute 49 according to Bayesian method. The other methods detect this novelty a little later than Bayesian method.

Table 4.13: The second noveelty detected by Bayesian method could be interpreted as an early detection.

	Healthy	Intermediate	Faulty
Mahalanobis	-	1 *(2472)*	1 *(2531)*
SPLL	1 *(480)*	1 *(931)*	1 *(2516)*
Bayesian	-	2 *(698,2337)*	1 *(2478)*
LSTM-Autoencoder	-	1 *(902)*	1 *(2524)*



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.34: *t-SNE visualizations of each method for Bearing3_1:* Early novelties claimed by SPLL, Bayesian and LSTM-Autoencoder divides larger cluster into sub-clusters. However, there are two small clusters formed unambiguously after defect starts to show itself. Only Bayesian method is able to distinguish the severity level of this defect as seen in Figure 4.34d.

- *Bearing3_2:* Novelty detection in this bearing is particularly difficult, since almost every type of defect occurred during the degradation test. However, there is a clear novelty at 47 minute mark, which could be confirmed in Figure 4.35 and Figure 4.6e.

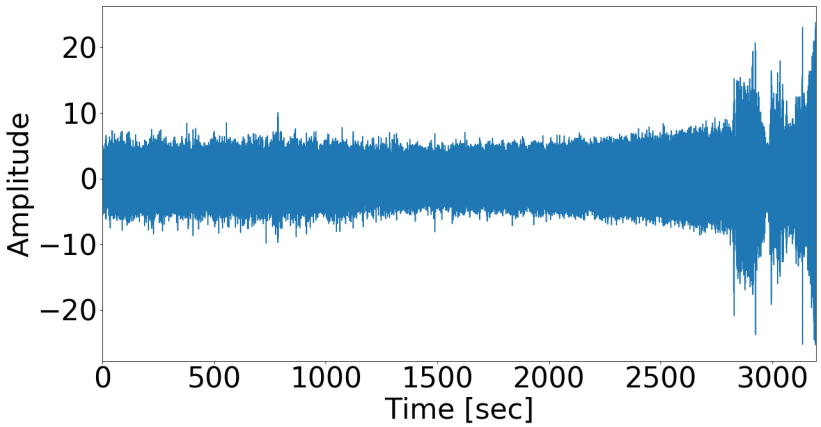


Figure 4.35: Raw vibration signal of Bearing3_2.

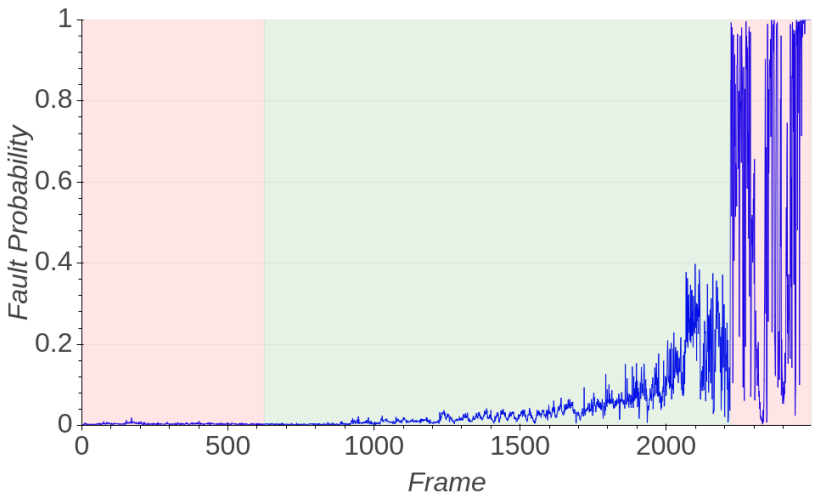
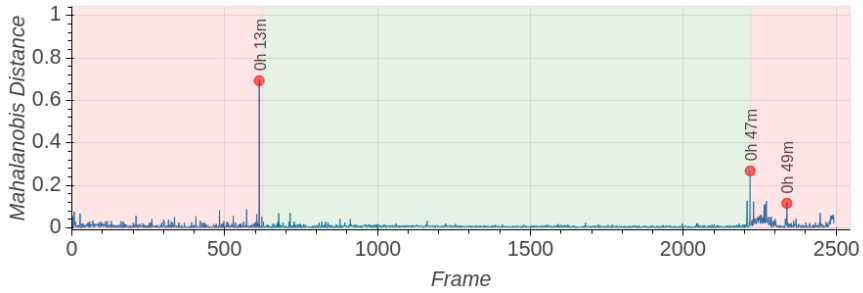
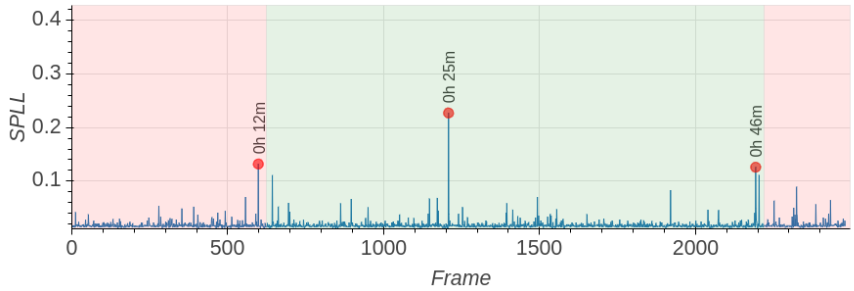


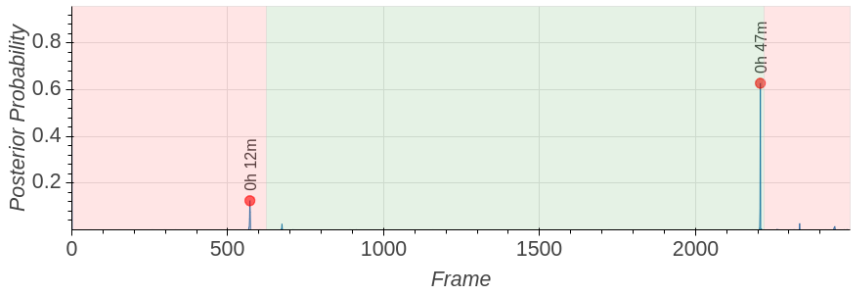
Figure 4.36: *Segments for Dataset 3 - Bearing 2*: The transition between intermediate and faulty segments isn't as smooth as the other bearings. This is because the dataset used to train the classifier. Faulty data sample contains to many different features statistically which doesn't help to create a robust classifier. In this case healthy stage is assumed to end at 624 and the fault was found to be start at Frame 2220 by the classifier.



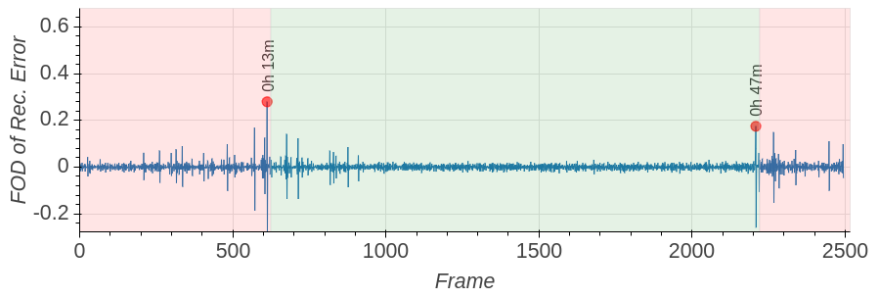
(a) Mahalanobis



(b) SPLL



(c) Bayesian

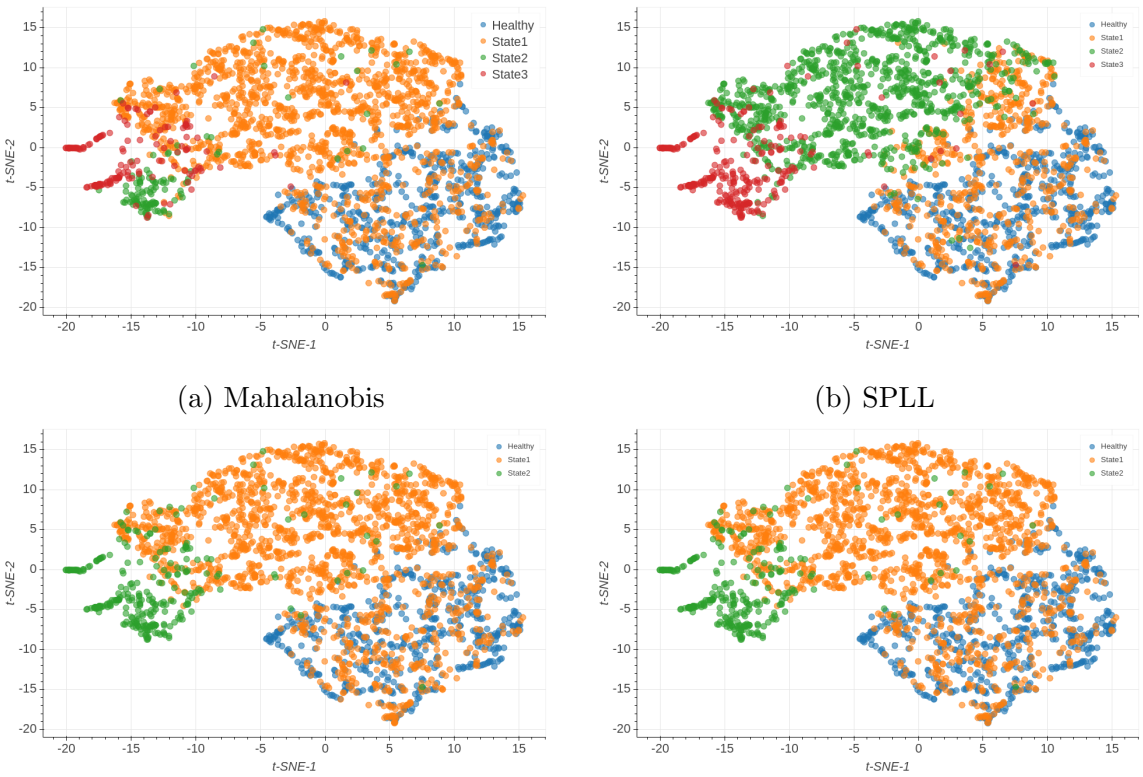


(d) LSTM-Autoencoder

Figure 4.37: *Novelty detection scores for Bearing3_2*: Each method detects a novelty around 12-13 minute mark. However it is neither obvious in t-SNE Figure 4.38 nor can be seen in Spectrogram Figure 4.6e. SPLL method finds a novelty at 25 minute mark, however this cannot be confirmed in neither of other conventional methods, so it is considered as false-positive.

Table 4.14: The novelties detected in healthy segment by each method are close to each other. Also around Frame 2208 there could be an early detection since its neighborhood was marked as a novelty by each method.

	Healthy	Intermediate	Faulty
Mahalanobis	1 *(613)*	1 *(2219)*	1 *(2339)*
SPLL	1 *(598)*	1 *(1208,2193)*	-
Bayesian	1 *(571)*	1 *(2208)*	-
LSTM-Autoencoder	1 *(612)*	1 *(2208)*	-



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.38: *t-SNE visualizations of each method for Bearing3_2*: Clusters are not easily recognizable in this case. Bayesian and LSTM-Autoencoder have detected most obvious novelties, SPLL detects one more novelty which is probably false-positive from the Figure 4.38b. Mahalanobis method also claims one more novelty at 49 minute mark, simply an advanced stage of bearing defect.

- *Bearing3_4:* This bearing has inner race defect at the end of the experiment, which was confirmed by all the methods. At Minute 30, vibration signal shows novelty signs. SPLL and LSTM-Autoencoder methods detected other novelties between 21-26 minute marks however they cannot be confirmed in Figure 4.42.

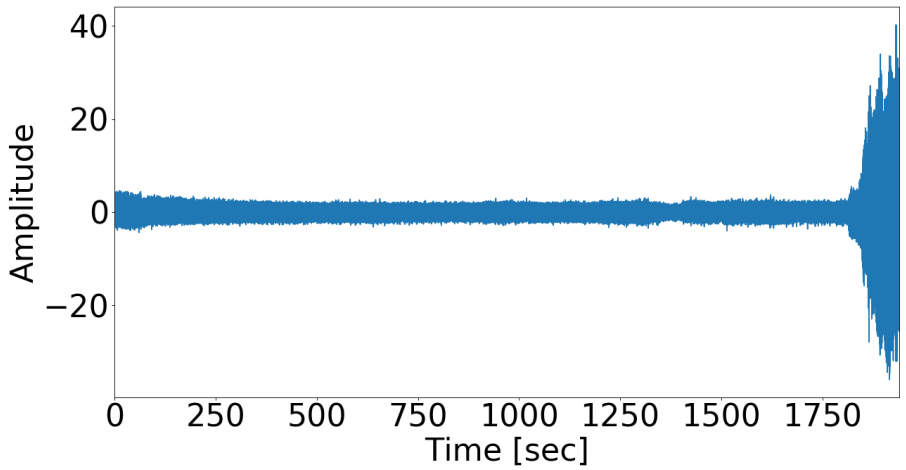


Figure 4.39: Raw vibration signal of Bearing3_4.

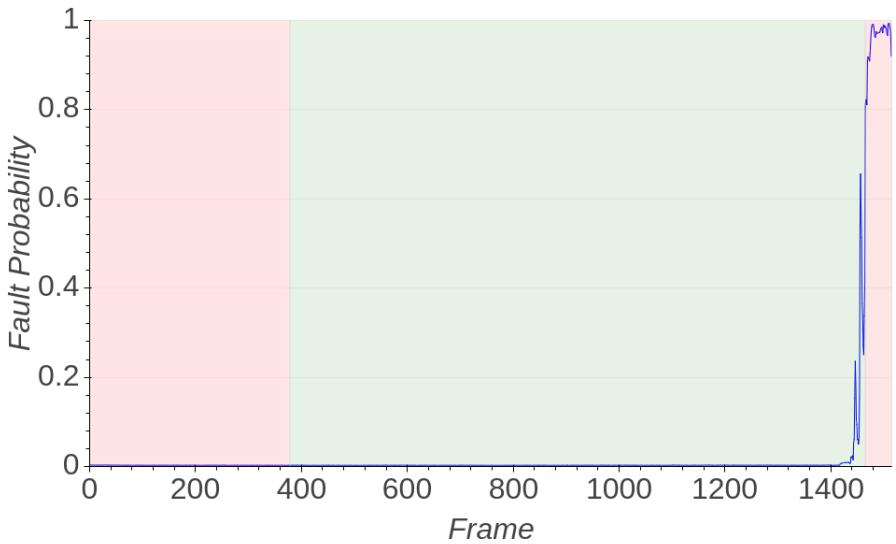
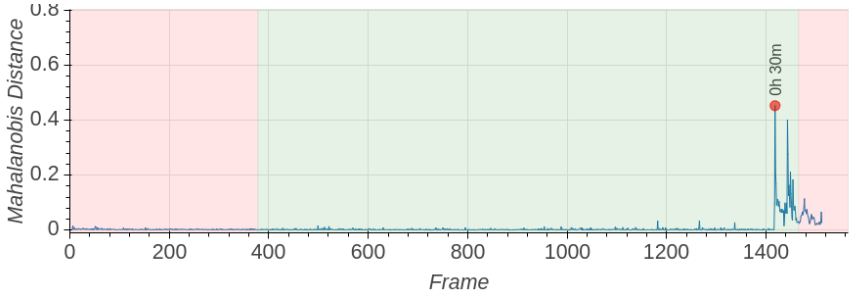
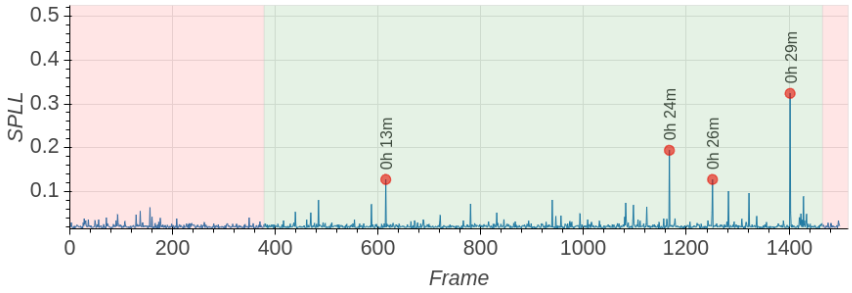


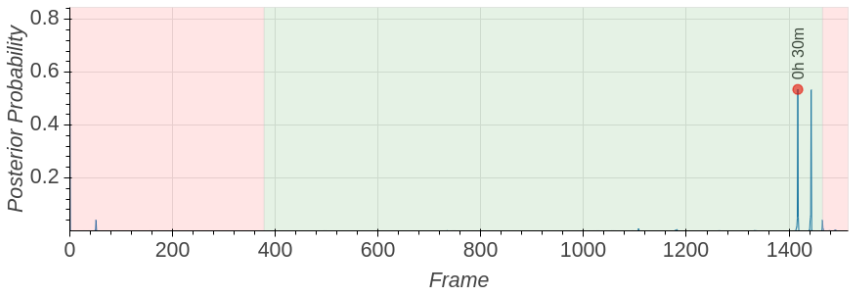
Figure 4.40: *Segments for Dataset 3 - Bearing 4*: Bearing defect immediately shows itself around Frame 1450. The transition between intermediate and faulty segment is clear. Healthy stage ends at Frame 328 and the fault was found to be start at Frame 1465 by the classifier.



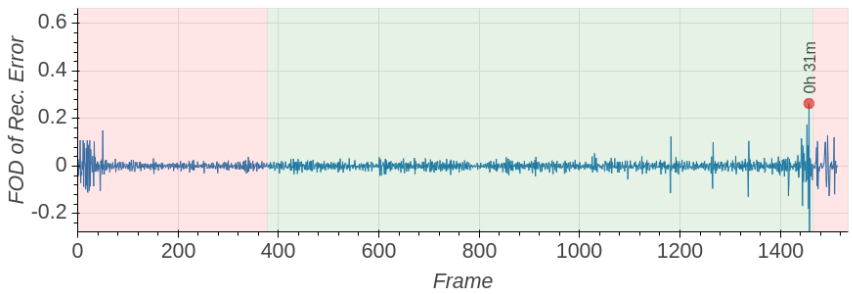
(a) Mahalanobis



(b) SPLL



(c) Bayesian

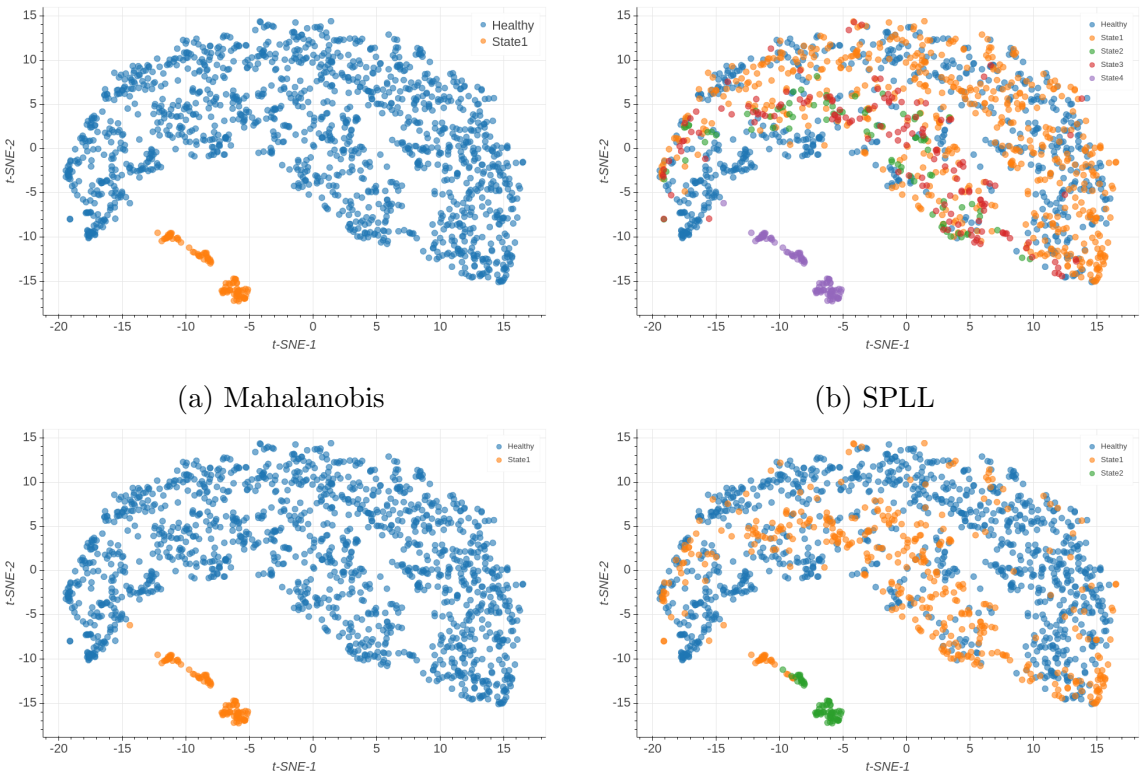


(d) LSTM-Autoencoder

Figure 4.41: *Novelty detection scores for Bearing3_4*: Although the actual defect start point seems like 30 minute mark, SPLL method detects novelties at 24 and 26 minute marks. These may be regarded as early detection of inner race fault, however there is no clear distinction between the clusters at that point as suggested in Figure 4.42. Similarly LSTM-Autoencoder detects a novelty at 21 minute mark and this again could be considered as early detection of the novelty.

Table 4.15: Only Intermediate stage contains novelties according to applied novelty detection methods.

	Healthy	Intermediate	Faulty
Mahalanobis	-	1 *(1418)*	-
SPLL	-	4 *(615,1161,1257,1402)*	-
Bayesian	-	1 *(1417)*	-
LSTM-Autoencoder	-	1 *(1442)*	-



(a) Mahalanobis

(b) SPLL

(c) Bayesian

(d) LSTM-Autoencoder

Figure 4.42: *t-SNE visualizations of each method for Bearing3_4*: The novelties detected other than 30 minute mark are probably false-positives as can be seen in Figure 4.42b and 4.42d. After that novelty, samples form a new cluster as can be seen as *State 1(yellow)* in Figure 4.42a and Figure 4.42c.

5. DISCUSSION

As discussed in Chapter 3 deriving different models for each machine in a plant, and monitoring them continuously is not cost-efficient. There might be defects which weren't defined in the current literature and a model that detects a transition from one state to another might be a better solution. Since faulty data is not available assuming machine started to work in a healthy state, an unsupervised novelty detection method is required to detect changes in a vibration data stream.

In Chapter 4 the results for novelty detection on streaming vibration data using four different methods are presented. There are two similar bearing degradation datasets in the current literature: IMS and XJTU-SY datasets. There are previous works on early fault detection for IMS datasets [46]. Almost all results obtained by Gousseau et al. (2016) are confirmed in the present work. For Dataset 2 - Bearing 1, the detection was achieved at 4 days and 11 hours into the experiment, which was claimed to be 3.5 days in their work. When SVM classifier is applied to the frames in this dataset, the probability of being a fault does not increase suddenly at 3.5 days.

The obvious defects were detected by each method for each vibration sample in IMS dataset. However, there are some earlier novelties detected which could stem from a change in motor setting. For example SPLL method detects a clear novelty 19 hours into the experiment(frame 156), which can be seen in Figure A.1. This same novelty is again detected in Bearing 4(Figure 4.13b), and this strengthens the hypothesis that the nature of the novelty is a change in motor settings. Samples for these two bearings were collected simultaneously.

Since there is no label in both datasets, and only information is the fault type of the bearing in the experiments, we wanted to have a reliable comparison and built an SVM model using the samples from the earliest and the latest of the vibration samples for the same bearing. Then we predicted a probability of being healthy(or unhealthy) for each sample. Assuming the real defect occurred at the probability of at least 0.8, we

segmented our data into three. Then we wanted to see how do the detected novelties fit into the segments' actual probability distribution. The outliers are usually close to segment transition points(especially between intermediate and faulty segment), which means these novelties actually an early indication of a bearing failure.

SPLL method has more false-positives compared to other methods, but this can be improved using different sized windows. Especially for Dataset 2 - Bearing 1 case, other methods show the bearing has no state change up to 4.5 day mark, but SPLL detects early novelties. These are clear false-positives.

XJTU-SY dataset is fairly new and it has become available publicly even more recently. Although there are not any works to compare the results for now, spectrograms in Figure 4.6 could help. Like the experiments for other dataset, there is almost none false-negatives in these experiments. LSTM-Autoencoder in Figure 4.25d fails to detect actual defect and has a false-positive anomaly at 4 minute mark.

Cases like Bearing3_2 in XJTU-SY dataset are the primary motivation of this study. This bearing includes each type of fault, which makes harder to understand the fault from frequency spectrum. Novelties are detected at 12(13 in some methods) and 47 minute marks by each method which implies they are true-positives.

In general, Mahalanobis and Bayesian based novelty detection methods resulted in most reliable scores. Although Bayesian method requires run-length parameter, score does not change drastically for close run-lengths. Mahalanobis method does not require any parameter.

Time complexity of SPLL algorithm is $\mathcal{O}(n^2)$ whereas Bayesian and Mahalanobis methods have linear time complexity. LSTM-Autoencoder has a time complexity of $\mathcal{O}(nLc^2)$ where L is the number hidden layers in the architecture and c is the number of recurrent units in each LSTM cell, which is encoder or decoder size. Considering the successful detections by Bayesian and Mahalanobis algorithms they can be considered as best choices for novelty detection in streaming vibration data.

The framework proposed in Figure 3.1 also suggests further steps once a novelty is detected by these algorithms. Once it is detected this novelty can be labeled as one of the pre-defined faults in the literature. If the type of bearing fault is known then it can be used to update the parameters of novelty detection algorithm. The labelling can be made by a human expert, but since continuous monitoring is not required anymore work load is reduced drastically.

If we consider machine monitoring framework in IIoT context, the process can be moved to cloud as well. Assuming novelty detection process works on-site, when novelty is detected the compression ratio can be decreased to send more information to the cloud. On the other hand, healthy vibration data contains less information and it compression ratio could be set high until a novelty comes. If detection is implemented on cloud, then the algorithm parameters can be updated more easily. Also there will be more resource available to learn from and train the algorithms, since data is being sent from other plants and machines simultaneously.

6. CONCLUSIONS AND FUTURE WORK

This work is devoted prognostics and health management of bearings based on vibration signature. The core idea behind this work is detecting a state transition(novelty) in bearing vibration signal as early as possible. Our assumption was, there is not available faulty data in order to train supervised machine learning methods. Each machine should be considered individually, and building supervised models for each of them would be costly. All these motivations led us to a machine monitoring framework in Figure 3.1, where detection is realized in an unsupervised manner. The methods were chosen for online novelty detection in streaming data, where models learn from data in one instance at a time. Once novelty is detected a human expert can always go back and analyze the data in detail. Using this framework these conclusions can be extracted from the experiments:

- Novelty detection methods are able to detect actual defects. They can also detect setting changes that could happen earlier in the experiments as well.
- In some cases, novelties could be detected earlier than they become obvious in frequency spectrum.
- The severity level of each defect can be detected using these algorithms. t-SNE plots show apparent clusters which were formed due to different level of degradation.
- Although XJTU-SY dataset is similar to IMS, it is fairly new and needs to be analyzed in detail.
- Among the novelty detection methods Bayesian ended up with most reliable results. Although it may not be able to detect degradation in its initial phase, setting changes and fault levels can be distinguished using Bayesian method.
- This work relies on feature extraction and dimensionality reduction. Features are generated from each sample which contains 1 or 1.28 seconds of data for IMS and XJTU-SY datasets respectively (20480 and 32768 datapoints). Novelty detection on raw vibration data is more difficult but also more rewarding challenge.

Hopefully new databases like XJTU-SY and IMS will be available publicly, which would increase the productivity of PHM domain immensely. All PHM research, like this thesis, should be applied in a real industry environment. Therefore, not only academia, industrial partners should also welcome these research for more productivity and sustainability.

REFERENCES

1. Shrouf, F., J. Ordieres and G. Miragliotta, "Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm", *2014 IEEE international conference on industrial engineering and engineering management*, pp. 697–701, IEEE, 2014.

2. Wang, Y., C. Deng, J. Wu, Y. Wang and Y. Xiong, "A corrective maintenance scheme for engineering equipment", *Engineering Failure Analysis*, Vol. 36, pp. 269–283, 2014.

3. de Faria Jr, H., J. G. S. Costa and J. L. M. Olivas, "A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis", *Renewable and sustainable energy reviews*, Vol. 46, pp. 201–209, 2015.

4. Wang, K., "Intelligent predictive maintenance (IPdM) system–Industry 4.0 scenario", *WIT Transactions on Engineering Sciences*, Vol. 113, pp. 259–268, 2016.

5. Harris, T. A. and M. N. Kotzalas, *Essential concepts of bearing technology*, CRC press, 2006.

6. Americas, N. B., *Bearing Doctor - Bearing Maintenance Guide*, 2018, `http://www.nskamericas-literature.com/en/bearing-doctor/`, accessed at June 2019.

7. Taylor, J., *The Vibration Analysis Handbook: A Practical Guide for Solving Rotating Machinery Problems*, VCI, 2003, `https://books.google.com.tr/books?id=V4GxAAAACAAJ`, accessed at June 2019.

8. Howard, I., *A Review of Rolling Element Bearing Vibration Detection, Diagnosis and Prognosis*, Defence Science and Technology Organization Canberra (Australia), 1994.

9. Kimotho, J. K. and W. Sextro, "An approach for feature extraction and selection from non-trending data for machinery prognosis", *Proceedings of the second european conference of the prognostics and health management society*, pp. 1–8, 2014.

10. Schneider, M., W. Ertel and F. Ramos, "Expected similarity estimation for large-scale batch and streaming anomaly detection", *Machine Learning*, Vol. 105, No. 3, pp. 305–333, 2016.

11. Martínez-Rego, D., O. Fontenla-Romero and A. Alonso-Betanzos, "Power wind mill fault detection via one-class ν-SVM vibration signal analysis", *The 2011 International Joint Conference on Neural Networks*, pp. 511–518, IEEE, 2011.

12. Camci, F. and R. B. Chinnam, "General support vector representation machine for one-class classification of non-stationary classes", *Pattern Recognition*, Vol. 41, No. 10, pp. 3021–3034, 2008.

13. Lee, S., G. Kim and S. Kim, "Self-adaptive and dynamic clustering for online anomaly detection", *Expert Systems with Applications*, Vol. 38, No. 12, pp. 14891–14898, 2011.

14. Pearson, K., "LIII. On lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Vol. 2, No. 11, pp. 559–572, 1901.

15. Maaten, L. v. d. and G. Hinton, "Visualizing data using t-SNE", *Journal of machine learning research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

16. Adams, R. P. and D. J. MacKay, "Bayesian online changepoint detection", *arXiv preprint arXiv:0710.3742*, 2007.

17. Forbes, C., M. Evans, N. Hastings and B. Peacock, *Statistical distributions*, John Wiley & Sons, 2011.

18. Kuncheva, L. I., "Change detection in streaming multivariate data using likelihood detectors", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 5, pp. 1175–1180, 2013.

19. Kuncheva, L. I., *Change-detection*, 2016, `https://github.com/LucyKuncheva/Change-detection.git`, accessed at June 2019.

20. Hochreiter, S. and J. Schmidhuber, "Long short-term memory", *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

21. Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation", *arXiv preprint arXiv:1406.1078*, 2014.

22. Bengio, S., O. Vinyals, N. Jaitly and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks", *Advances in Neural Information Processing Systems*, pp. 1171–1179, 2015.

23. Li, J., M.-T. Luong and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents", *arXiv preprint arXiv:1506.01057*, 2015.

24. Varsamopoulos, S., K. Bertels and C. G. Almudever, "Designing neural network based decoders for surface codes", *arXiv preprint arXiv:1811.12456*, 2018.

25. Rossi, A., F. Montefoschi, A. Rizzo, M. Diligenti and C. Festucci, "Auto-Associative Recurrent Neural Networks and Long Term Dependencies in Novelty Detection for Audio Surveillance Applications", *IOP Conference Series: Materials Science and Engineering*, Vol. 261, p. 012009, Oct 2017, `https://doi.org/10.1088%2F1757-899x%2F261%2F1%2F012009`.

26. Summerfield, M., *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*, Pearson Education, 2007.

27. Oliphant, T. E., *A guide to NumPy*, Vol. 1, Trelgol Publishing USA, 2006, `http://www.numpy.org/`, accessed at June 2019.

28. Jones, E., T. Oliphant, P. Peterson *et al.*, *SciPy: Open source scientific tools for Python*, 2001–, `http://www.scipy.org/`, accessed at June 2019.

29. McKinney, W., "Data Structures for Statistical Computing in Python", S. van der Walt and J. Millman (Editors), *Proceedings of the 9th Python in Science Conference*, pp. 51 – 56, 2010, `http://pandas.pydata.org`, accessed at June 2019.

30. da Costa-Luis, C., S. L., H. Mary, K. Altendorf, noamraph, M. Korobov, I. Ivanov, M. Bargull, G. CHEN, mjstevens777, M. D. Pagel, James, C. Newey, Todd, S. Malmgren, Socialery, M. Nordlund, M. Zugnoni, J. McCracken, Hugo, F. Dill, D. Panteleit, Alexander, A. Rothberg, D. Fu, D. Bau, A. Persaud, A. Portnoy, A. Kottke and A. Umer, *tqdm/tqdm: tqdm v4.31.1 stable*, 2006, `https://tqdm.github.io/`, accessed at June 2019.

31. Lee, G. R., R. Gommers, F. Wasilewski, K. Wohlfahrt and A. O'Leary, *PyWavelets/pywt: PyWavelets v1.0.3*, 2006, `https://pywavelets.readthedocs.io/en/latest/`, accessed at June 2019.

32. Chollet, F. *et al.*, *Keras*, 2015, `https://keras.io`, accessed at June 2019.

33. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.

34. Bokeh Development Team, *Bokeh: Python library for interactive visualization*, 2014, `http://www.bokeh.pydata.org`, accessed at June 2019.

35. Hunter, J. D., "Matplotlib: A 2D graphics environment", *Computing in Science &*

Engineering, Vol. 9, No. 3, pp. 90–95, 2007.

36. Kluyver, T., B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla and C. Willing, "Jupyter Notebooks – a publishing format for reproducible computational workflows", F. Loizides and B. Schmidt (Editors), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87 – 90, IOS Press, 2016, `https://jupyter.org/`, accessed at June 2019.

37. Smith, W. A. and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study", *Mechanical Systems and Signal Processing*, Vol. 64, pp. 100–131, 2015.

38. Lessmeier, C., J. K. Kimotho, D. Zimmer and W. Sextro, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: a benchmark data set for data-driven classification", *Proceedings of the European conference of the prognostics and health management society*, pp. 05–08, 2016.

39. Huang, H. and N. Baddour, "Bearing vibration data collected under time-varying rotational speed conditions", *Data in brief*, Vol. 21, pp. 1745–1749, 2018.

40. Daga, A. P., A. Fasana, S. Marchesiello and L. Garibaldi, "The Politecnico di Torino rolling bearing test rig: Description and analysis of open access data", *Mechanical Systems and Signal Processing*, Vol. 120, pp. 252–273, 2019.

41. Qiu, H., J. Lee, J. Lin and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics", *Journal of sound and vibration*, Vol. 289, No. 4-5, pp. 1066–1090, 2006.

42. Ali, J. B., B. Chebel-Morello, L. Saidi, S. Malinowski and F. Fnaiech, "Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network", *Mechanical Systems and Signal Processing*, Vol. 56, pp. 150–172,

2015.

43. Mortada, M.-A., S. Yacout and A. Lakis, "Diagnosis of rotor bearings using logical analysis of data", *Journal of Quality in Maintenance Engineering*, Vol. 17, No. 4, pp. 371–397, 2011.

44. Wang, B., Y. Lei, N. Li and N. Li, "A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings", *IEEE Transactions on Reliability*, 2018.

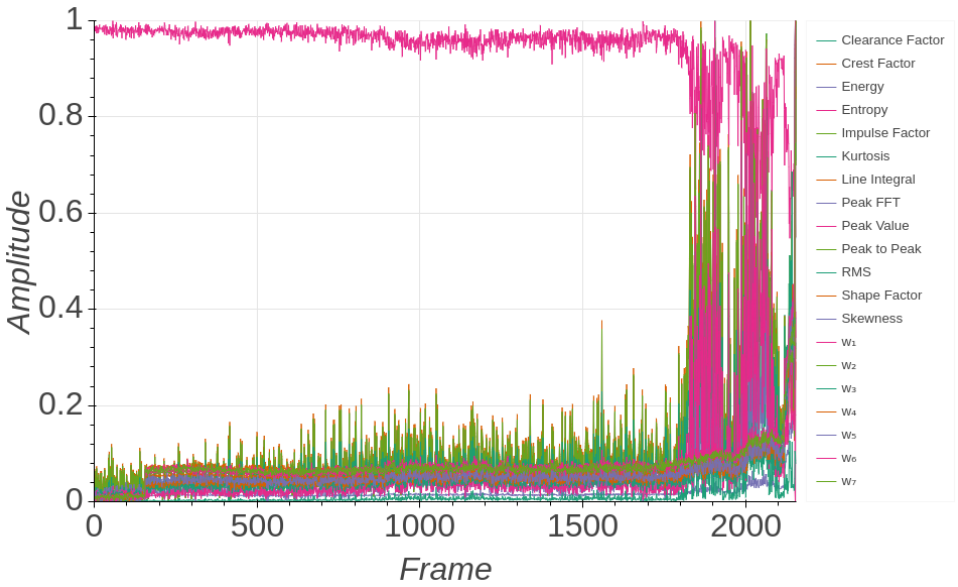
45. Rexnord, *Rex ZA2115 2000 Series Solid-housed Pillow Block Spherical Roller Bearing*, 2019, `https://www.rexnord.com/Products/Bearings/Rex-Spherical-Roller-Bearings/Solid-housed-Pillow-Blocks/ZA2000/ZA2115`, accessed at June 2019.

46. Gousseau, W., J. Antoni, F. Girardin and J. Griffaton, "Analysis of the Rolling Element Bearing data set of the Center for Intelligent Maintenance Systems of the University of Cincinnati", *CM2016*, 2016.

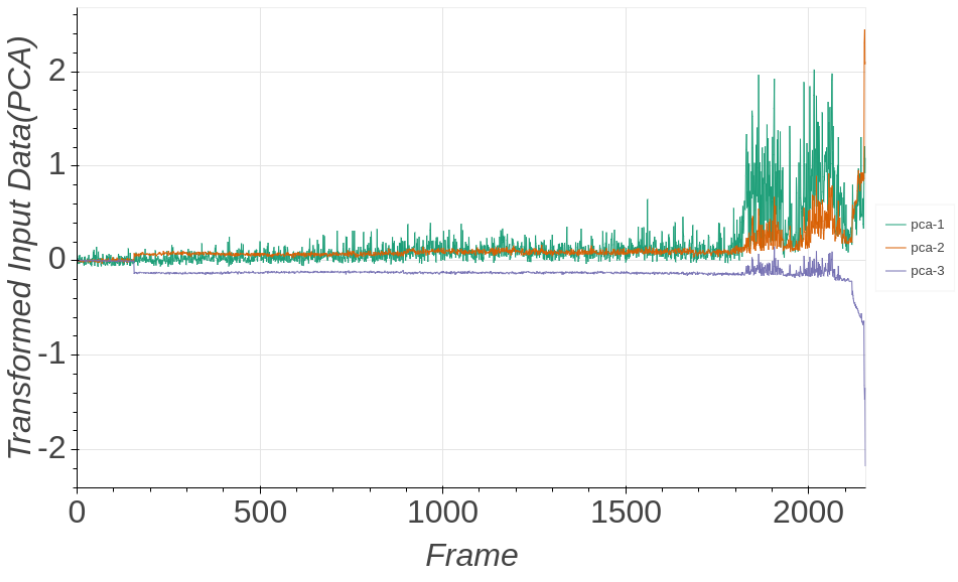
47. Widodo, A. and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis", *Mechanical systems and signal processing*, Vol. 21, No. 6, pp. 2560–2574, 2007.

APPENDIX A: FEATURES AND PCAS

This chapter includes extracted features and their transformations after PCA.

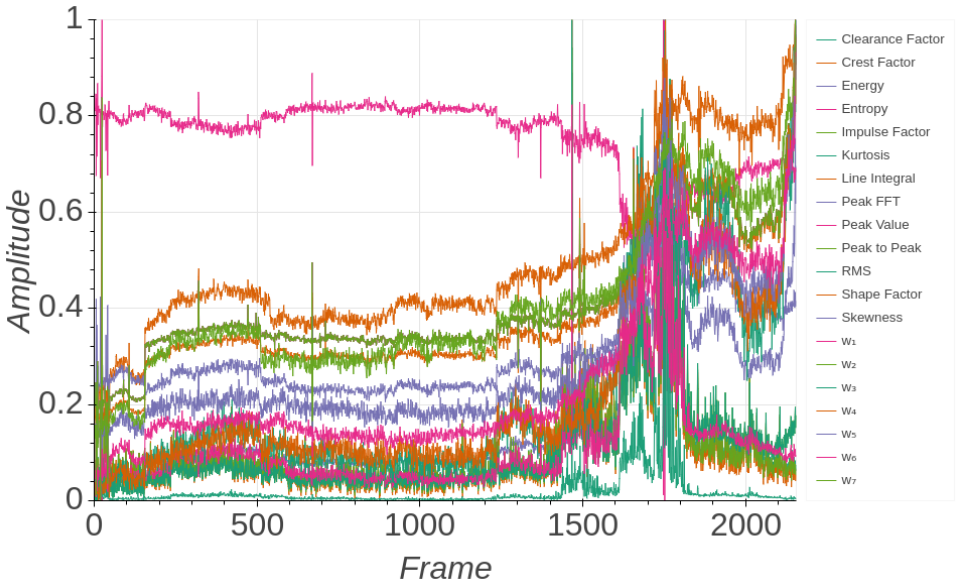


(a) Extracted features from raw vibration signal.

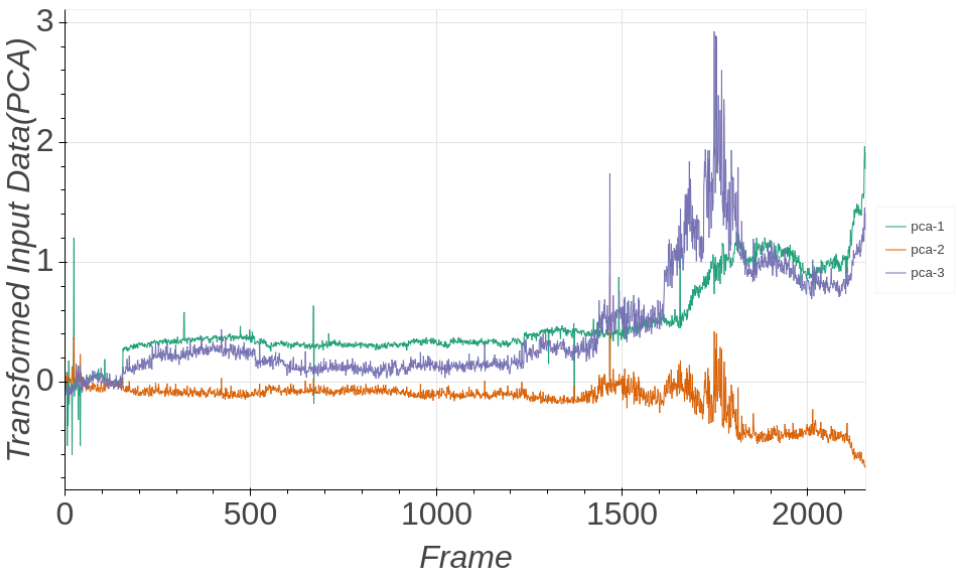


(b) Transformed features using first 3 PCs.

Figure A.1: IMS Dataset 1 - Bearing 3.

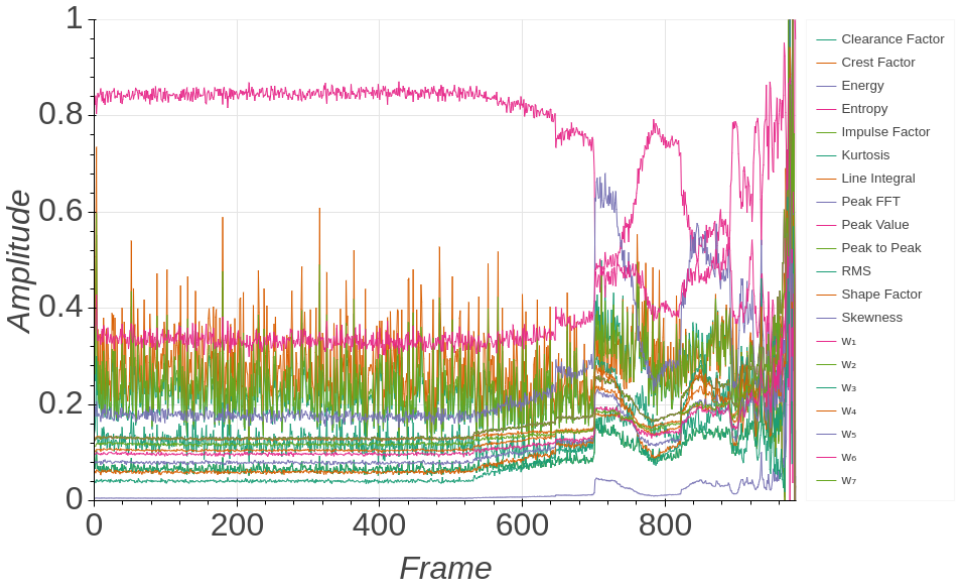


(a) Extracted features from raw vibration signal.

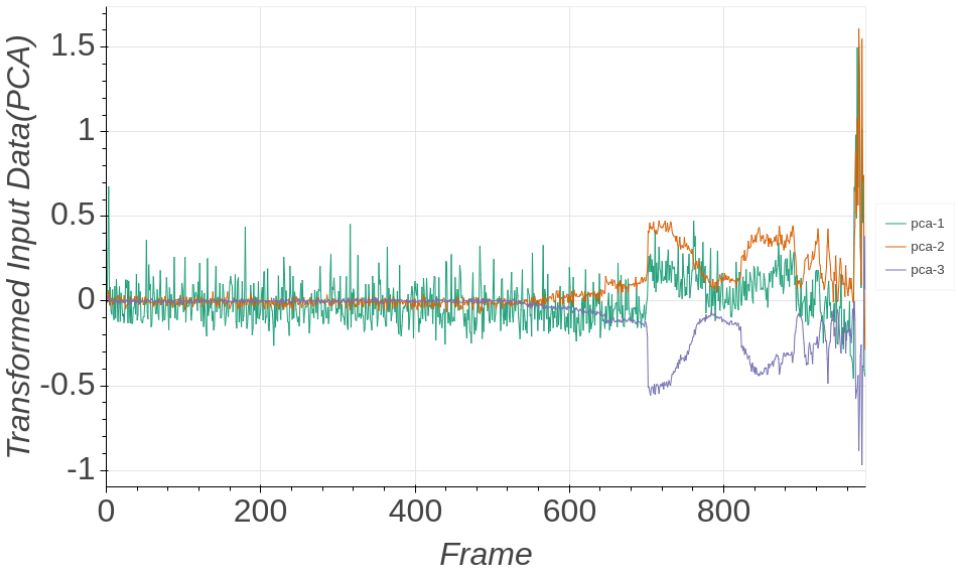


(b) Transformed features using first 3 PCs.

Figure A.2: IMS Dataset 1 - Bearing 4.

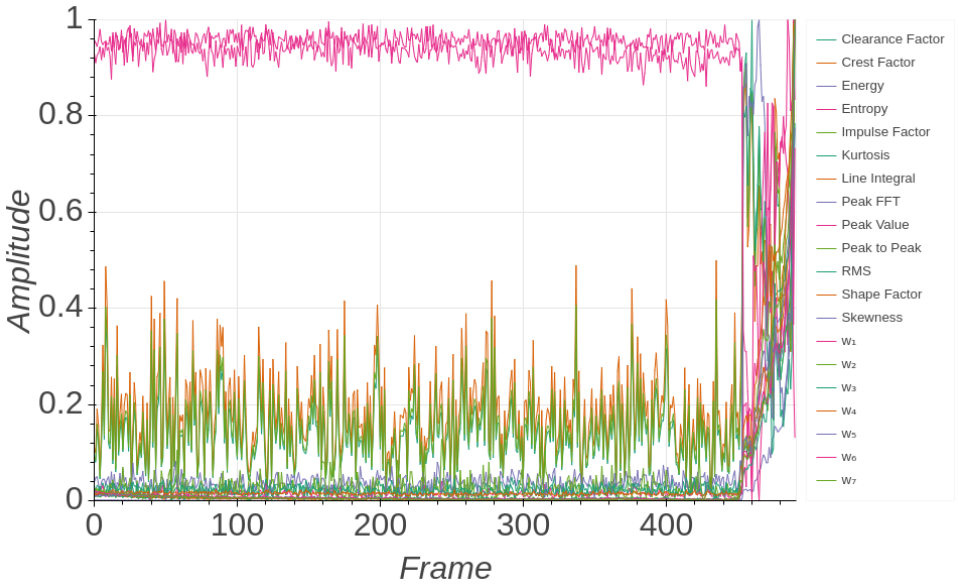


(a) Extracted features from raw vibration signal.

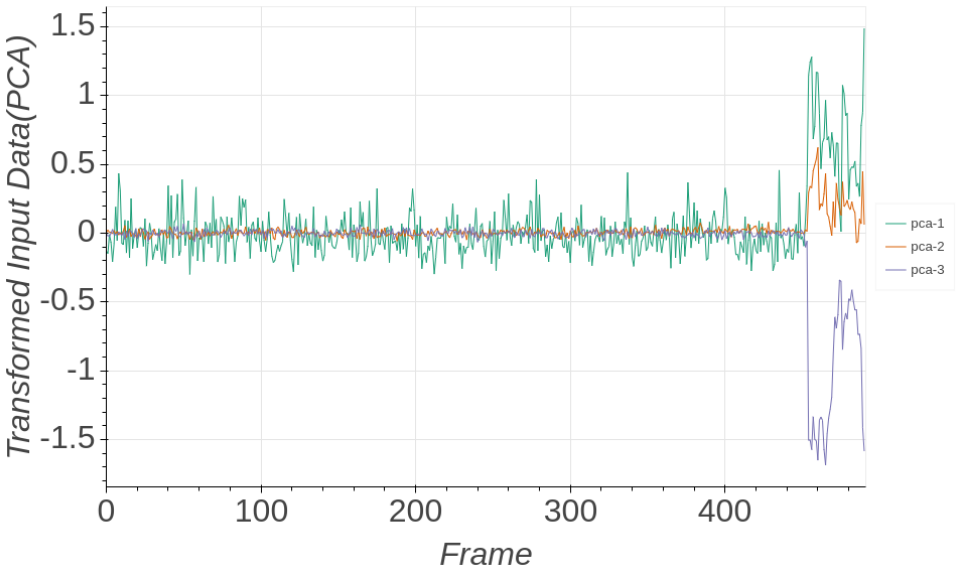


(b) Transformed features using first 3 PCs.

Figure A.3: IMS Dataset 2 - Bearing 1.

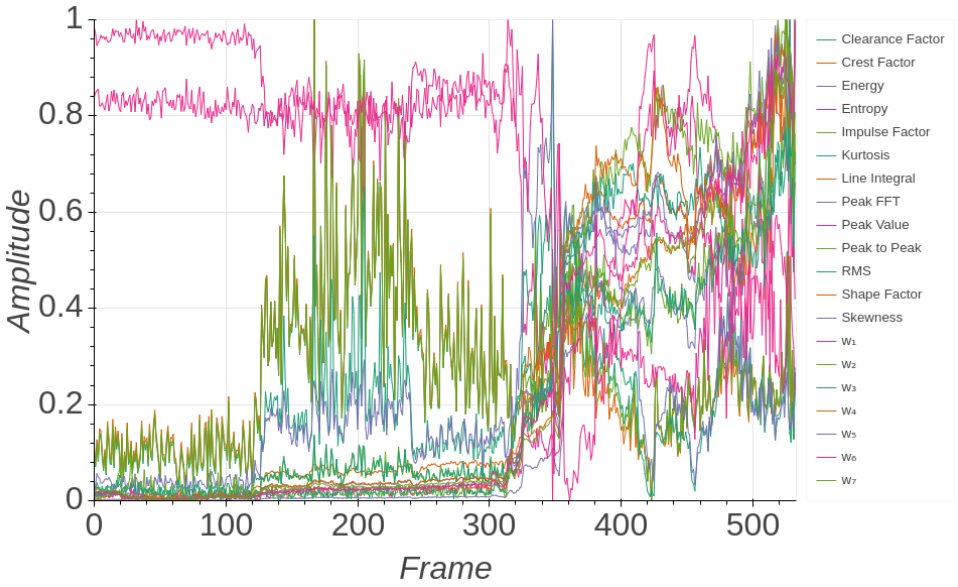


(a) Extracted features from raw vibration signal.

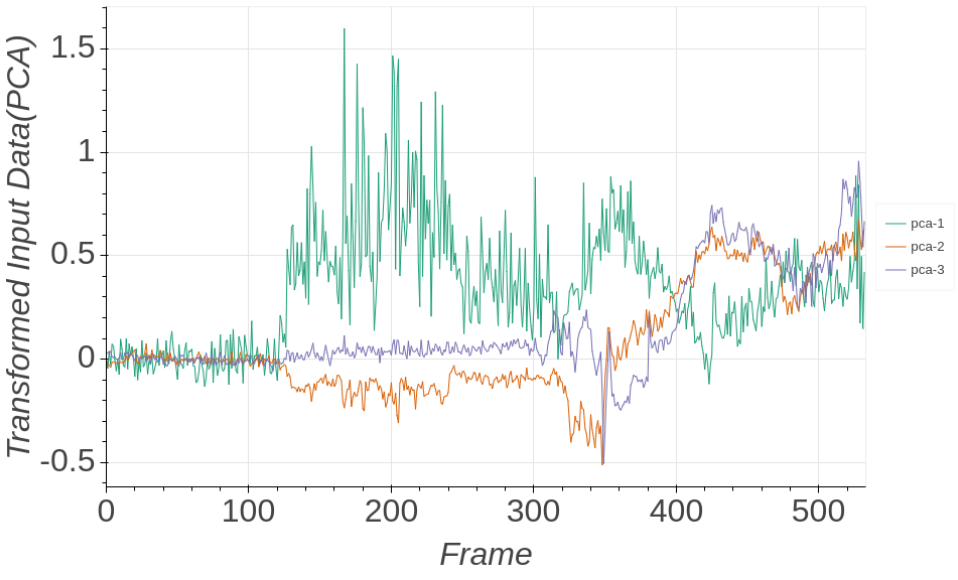


(b) Transformed features using first 3 PCs.

Figure A.4: XJTU-SY Dataset 2 - Bearing 2_1.

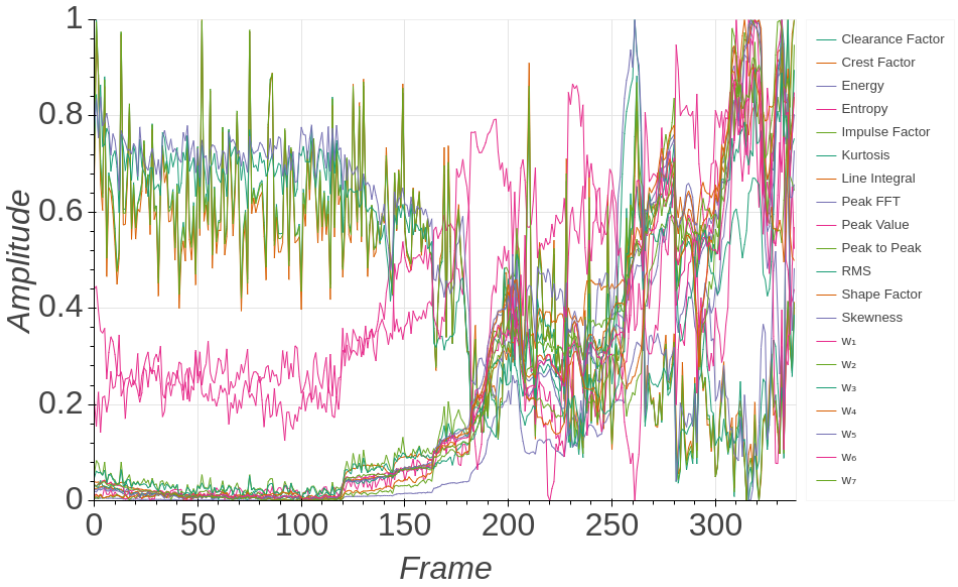


(a) Extracted features from raw vibration signal.

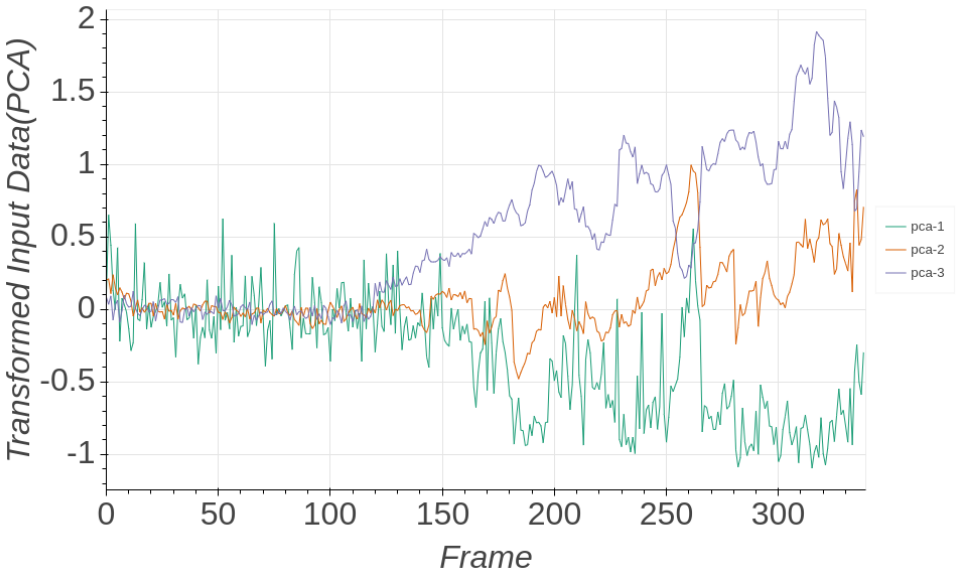


(b) Transformed features using first 3 PCs

Figure A.5: XJTU-SY Dataset 2 - Bearing 2_3.

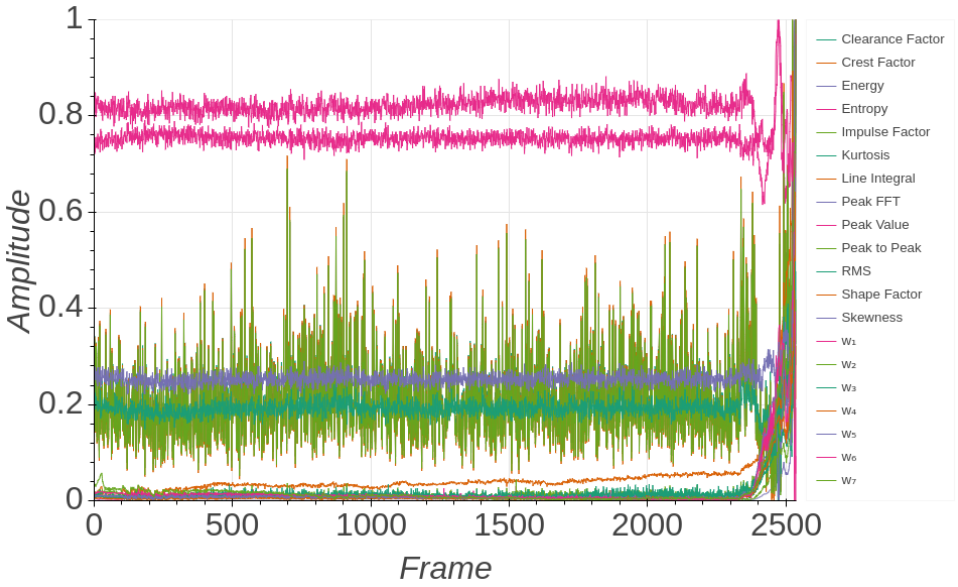


(a) Extracted features from raw vibration signal.

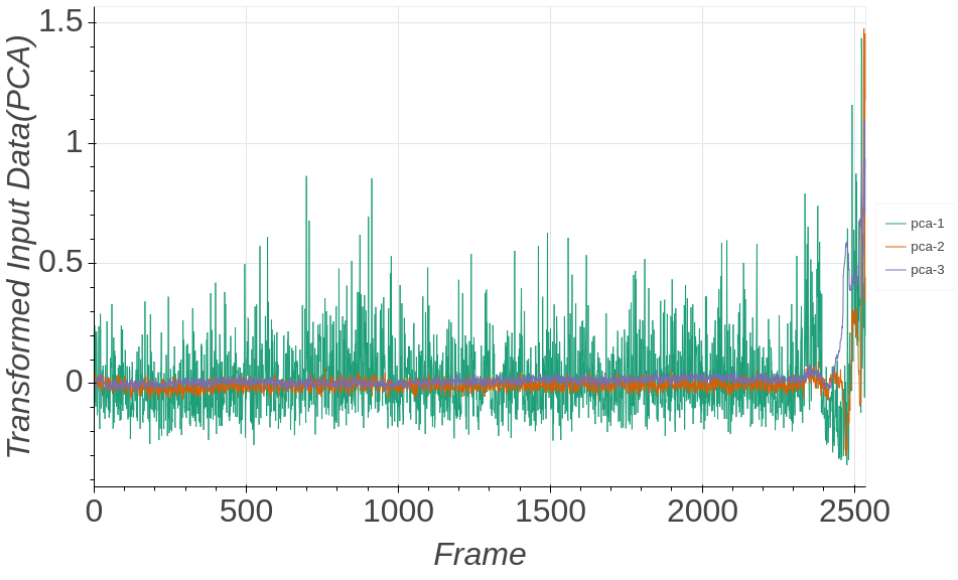


(b) Transformed features using first 3 PCs.

Figure A.6: XJTU-SY Dataset 2 - Bearing 2_5.

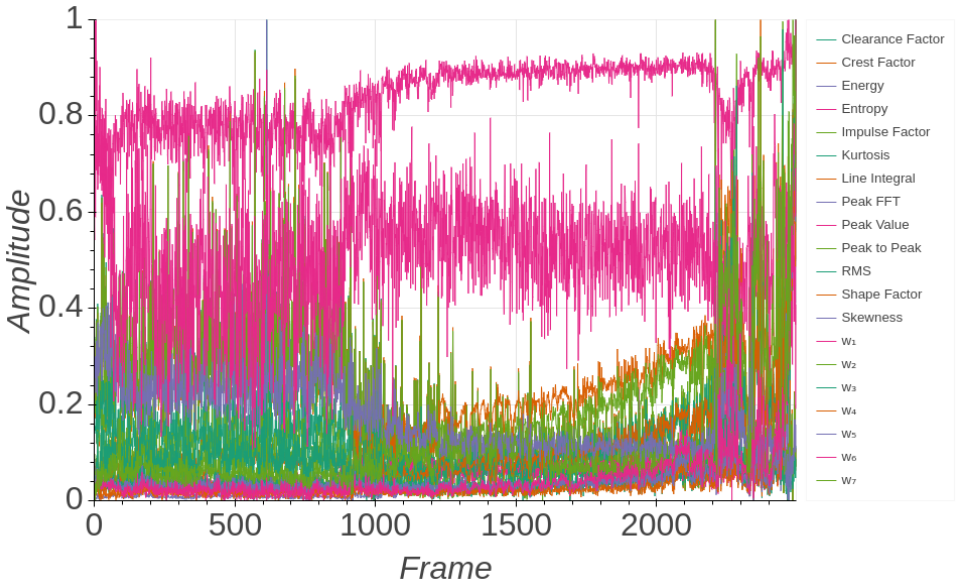


(a) Extracted features from raw vibration signal.

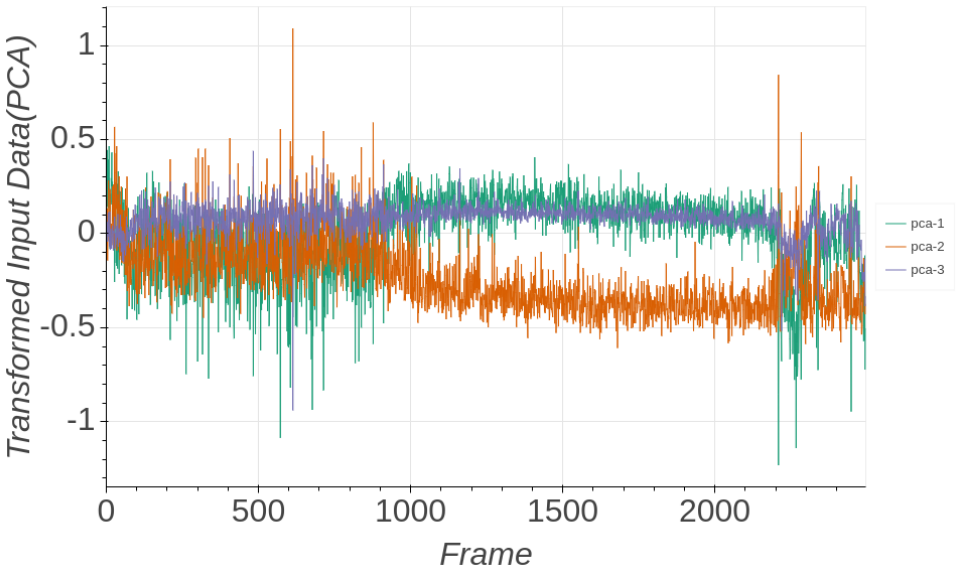


(b) Transformed features using first 3 PCs.

Figure A.7: XJTU-SY Dataset 3 - Bearing 3_1.

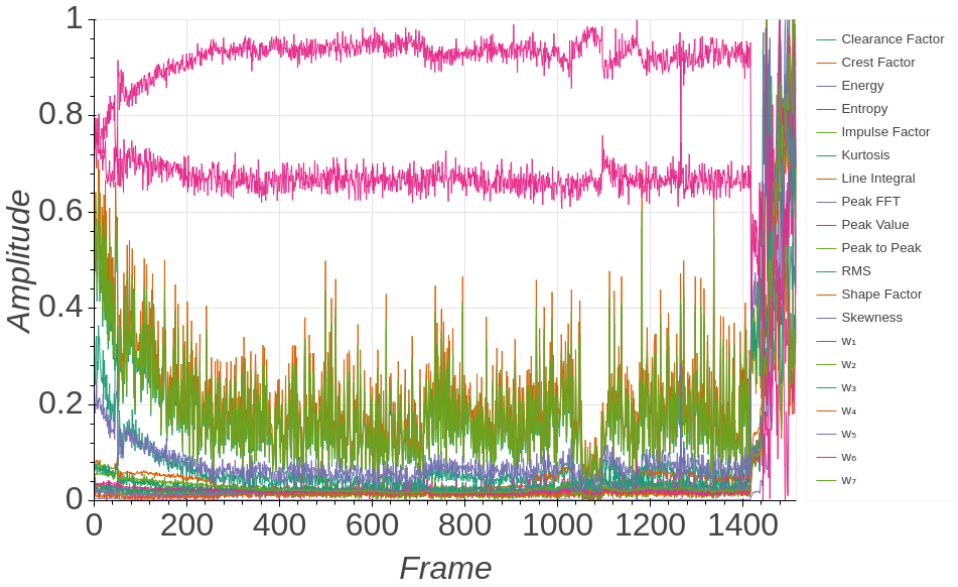


(a) Extracted features from raw vibration signal.

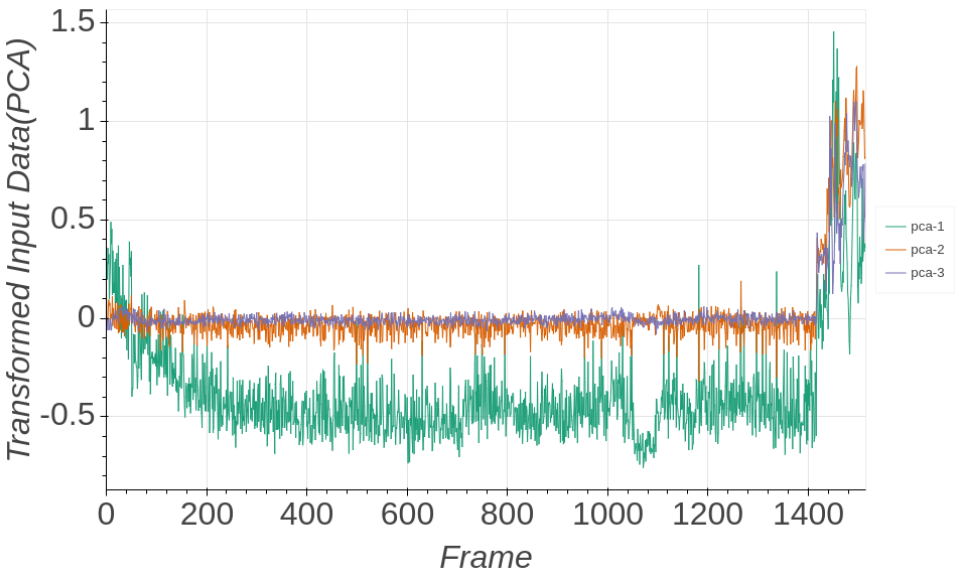


(b) Transformed features using first 3 PCs.

Figure A.8: XJTU-SY Dataset 3 - Bearing 3_2.



(a) Extracted features from raw vibration signal.



(b) Transformed features using first 3 PCs.

Figure A.9: XJTU-SY Dataset 3 - Bearing 3_4.