

# CONDITION MONITORING AND FAULT DETECTION OF A LASER OSCILLATOR FEEDBACK SYSTEM

ARNE GRÜNHAGEN,<sup>1,2,3</sup> ANNIKA EICHLER,<sup>2,3</sup>  
MARINA TROPMANN-FRICK,<sup>1</sup> GÖRSCHWIN FEY<sup>2</sup>

Hamburg University of Applied Sciences, HAW, Hamburg, Germany  
arne.gruenhagen@desy.de, marina.tropmann-frick@haw-hamburg.de

Hamburg University of Technology, TUHH, Hamburg, Germany  
arne.gruenhagen@desy.de, annika.eichler@desy.de, goerschwin.fey@tuhh.de  
Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany  
arne.gruenhagen@desy.de, annika.eichler@desy.de

The successful operation of industrial plants like the European X-Ray Free Electron Laser relies on the correct functioning of many dynamic systems that operate in a closed loop with controllers. In this paper, we present how data-based machine learning methods can monitor and detect disturbances of such dynamic systems based on the controller output signal. We implement four feature extraction methods based on statistics from the time domain, statistics from the frequency domain, characteristics of spectral peaks, and the autoencoder latent space representation of the frequency domain. These extracted features require no system understanding and can easily be transferred to other dynamic systems. We systematically compare the performance of 19 state-of-the-art fault detection methods to decide which combination of feature extraction and fault detection is most appropriate to model the condition of an actively controlled phase-locked laser oscillator. Our experimental evaluation shows that especially clustering algorithms are very well suited for detecting disturbed system conditions.

**Keywords:**  
fault detection,  
feature extraction,  
autoencoder,  
clustering,  
outlier detection

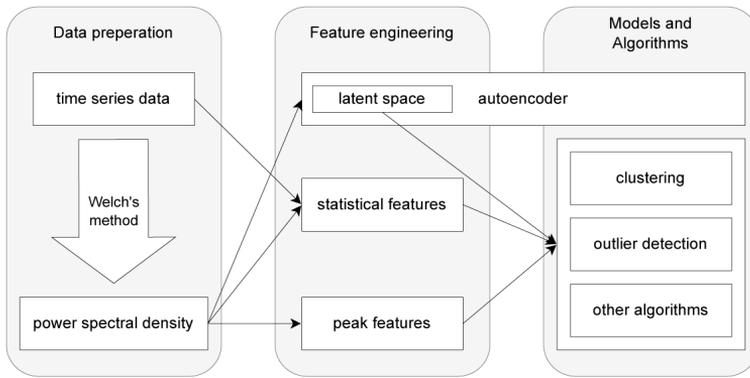
## 1 Introduction and Motivation

The European X-ray Free-Electron Laser (EuXFEL) [1] is a large-scale linear particle accelerator located in Hamburg, Germany. A 1.3 GHz Radio Frequency (RF) Master Oscillator (MO) is used to synchronize various components of the accelerator by distributing the RF signal as a timing reference source. Since this electrical distribution via coaxial cables is heavily influenced by the environment (e.g. humidity, temperature, electromagnetic fields), an optical synchronization system is installed that is less vulnerable to these environmental condition changes [2]. This optical synchronization system provides ultra-stable reference timing information to the accelerator components and the experimental setups with an integrated timing jitter in the range of a few femtoseconds. The main component of this optical synchronization system is a mode-locked pulsed laser oscillator that is phase-locked to the MO delivering an ultra-stable optical reference used to locally resynchronize RF sources, to lock optical laser systems, and to diagnose the arrival time of the electron beam along various locations for fast beam based feedbacks. Not only does the laser not produce a completely noise-free signal, but the emitted signal is also influenced by environmental disturbances (i.e., electrical, acoustical, mechanical, and optical) resulting in amplitude and phase fluctuations. To synchronize the laser oscillator to the MO, the relative phase error between a harmonic of the laser pulse repetition rate and the MO reference is determined and fed to a Proportional-Integral (PI) controller in a feedback loop. This controller acts on the laser oscillator cavity length to lock the laser oscillator repetition rate to the 1.3 GHz MO frequency with a loop bandwidth in the order of 1 kHz to 10 kHz [3]. Since the controller compensates for disturbances, the controller output signal is an ideal data source to detect potential disturbances that increase the integrated timing jitter and therefore decrease the synchronization performance.

The aim of this work is to detect changes in the controller output signal which may indicate environmental disturbances, disturbances in the MO reference or disturbances in the internal detection chain. This goal is achieved by realizing the fault detection pipeline depicted in Figure 1. In the data preparation step, we extract the power spectral density (PSD) from the controller output signal using Welch's method [4] such that the fault detection can be based on both, data from the time domain and data from the frequency domain. In the feature engineering step, we

implemented three different methods to extract meaningful features to fit several fault detection models. The phases will be explained in detail in Sections 3 and 4.

In the following, we summarize related work in Section 2. Then we describe the data preparation and feature engineering steps in Section 3. Section 4 gives a brief overview on the methods selected for fault detection and Section 5 gives a detailed overview on the experimental validation of the proposed fault detection pipeline. We conclude this work highlighting specific findings and giving plans on future work in Section 6.



**Figure 1: Fault detection pipeline**

Source own.

## 2 Related Work

Despite extensive literature about fault detection and anomaly detection in the area of manufacturing systems [5,6,7,8,9] only a few publications address fault detection of dynamic systems in closed-loop control. Especially, literature on data-based fault detection is very rare.

The authors of [10] use linear transfer functions to represent the actively controlled system under review and its controller. These models build the core of their fault diagnosis since they evaluate the discrepancy between the physical system output and the model output and the discrepancy between the physical controller output and the model output. These discrepancy measures are used as an anomaly score. Also, the authors of [11] use mathematical models to describe a physical system and

compare the model behavior with the behavior of the controlled system. Based on the difference between the system output and model output, faulty system conditions are identified. In [12], the authors address control-loop data from a real system. They implement different fault detection mechanisms for different fault types, namely an oscillation detection based on an autocorrelation function, the detection of sluggish-tuned loops using the so-called idle index, quantization detection, and a saturation detection method. Again their approach requires a deep system understanding.

Feature extraction for different industrial sectors is addressed by many publications. The authors of [13,14,15] each extract different basic statistics from the time domain, like the mean, the maximum, the minimum, the root mean square, or the entropy. In [16,17], the authors analyze frequency-domain vibration signals and decide on the system's health condition based on the values of domain-relevant frequency components. In [18] the authors calculate both, statistics from the time domain and statistics from the frequency domain as features for standard fault detection methods.

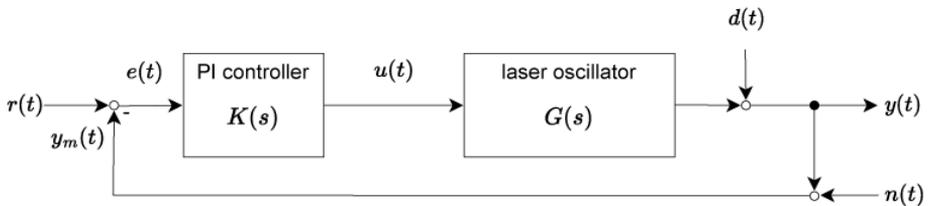
The authors of [19] developed a bi-directional long short-term memory neural network that works directly on time series data as a fault diagnosis mechanism. They compare the results of the bi-directional long short-term memory neural network with standard models that are fitted on time and frequency-related statistics. Also, the authors of [20] use neural networks in the form of a relational autoencoder to extract high-level features. We conclude that most of the related work addressing dynamic systems uses control theory models and therefore requires a deep understanding of the control theory behind the system. Existing publications using data-based fault detection methods do not address controller data.

### 3 Data Engineering

In this section, we describe what kind of data is used and how the data is processed for building meaningful models that can describe the condition of laser oscillators.

Figure 2 shows a simplified version of the laser oscillator control loop. The input  $e(t)$  to the PI controller is the difference between the reference signal  $r(t)$ , which in the case of the laser oscillator is the phase of the reference signal provided by the

electrical timing information coming from the MO, and the phase of the signal generated by the laser oscillator  $y(t)$ , affected by environmental disturbances  $d(t)$ . The output  $u(t)$  of the PI controller feeding into the laser oscillator is a voltage that affects the cavity length of the laser oscillator and thereby adjusting the phase of the laser signal. This outgoing signal, also called feedback signal, contains information about disturbances that the PI controller is processing and is therefore a valuable source of information for fault detection.



**Figure 2: Overview of the laser oscillator control scheme**

Source: own.

$K(s)$  is the controller at state  $s$

$G(s)$  is the laser oscillator at state  $s$

$r(t)$  is the reference signal which the system output should follow

$y(t)$  is the laser oscillator output

$n(t)$  is the noise added by the measurement

$y_m(t)$  is the laser oscillator output with the added measurement noise

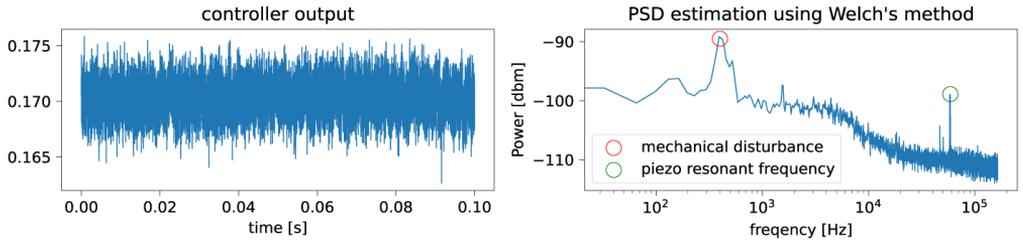
$e(t)$  is the difference between  $y_m(t)$  and  $r(t)$  and the input to the controller

$u(t)$  is the controller output

$d(t)$  is the disturbance acting on the laser oscillator output

### 3.1 Time and Frequency Domain

The controller's output signal contains values in the range from 0 to 1 and is measured with a sampling rate of 0.32 MHz. To check what kind of disturbances affect the system, the operators of the optical synchronization system mainly study the PSD estimation.



**Figure 3: Example for time series signal and PSD during normal operation**

Source: own.

Figure 3 shows examples of the feedback signal in time domain and the respective PSD in the frequency domain during healthy operation. The time series signal is an oscillating signal containing the changes to the cavity length of the laser oscillator. Due to the oscillating nature of the feedback signal in the time domain, single data points cannot reflect the entire state of the system and therefore it is mandatory to look at a series of data points. For our calculations, each series contains 30000 datapoints, which is equivalent to 0.1 s.

We calculate the PSDs using Welch's method [4]. Welch's method divides the time series data into overlapping segments, computes a modified periodogram for each segment, and averages the periodograms to the resulting PSD. Our PSD calculation uses Hanning windows containing 10000 data points with an overlap of 5000 datapoints. As a result, each PSD consists of 5000 datapoints. The shape of the PSD and its peaks at certain frequencies are characteristic of the current state of the system. For example, the increased power at 400 Hz comes from a mechanical disturbance of the laser oscillator and the peak at 60000 Hz originates from the piezo resonant frequency (see Figure 3).

In either case, considering the time-domain signals or the PSD in frequency domain, we work with a series of data points, also called frames. Depending on the frame size, the fault detection algorithms may have to work with high dimensional data, which can lead to poor fault detection performance. For this reason, we use several feature engineering techniques to reduce the dimensionality of the input data. In the following, we describe three feature engineering techniques applied to the data.

### 3.2 Statistical Feature Extraction

We use the *tsfresh* Python package [21] to calculate a bunch of statistics from the data frames. Table 1 gives an overview of the extracted statistics, a short description, and, if applicable, the corresponding parameter choices. This statistical feature extraction is applied to the time frames and to the PSDs. In both cases, the resulting dataset contains 34 values for the time-domain frame or PSD, respectively.

Table 1: Summary of extracted statistics from data series x

Statistic	Description	Parameter values
maximum	maximum of x	-
minimum	minimum of x	-
mean	mean $\mu$ of x	-
standard deviation	standard deviation of x	-
variation coefficient	$\frac{\text{standard deviation}}{\text{mean}}$	-
variance	variance $\sigma$ of x	-
skewness	skewness of x, determined with the adjusted Fisher-Pearson standardized moment coefficient G1	-
kurtosis	kurtosis of x, determined with the adjusted Fisher-Pearson standardized moment coefficient G2	-
root mean square	root mean square of x	-
quantile	The quantile is the value that is greater than the q-th proportion of the values in x	q $\in$ [0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9]
autocorrelation for lags	$R(l) = \frac{1}{n-l} \sum_{i=0}^{n-l} (x_i - \mu)(x_{i+l} - \mu)$ , where n denotes the length of x	l $\in$ [0,1,2,3,4,5,6,7,8,9]
linear trend attributes	different attributes of the linear regression from x	pvalue, rvalue, intercept, slope, stderr
absolute energy	sum over squared values of x	-

### 3.3 Peak Feature Extraction

The peak feature extraction addresses spectral data from the frequency domain. Peaks within a power spectrum are special characteristics since they show how much the PI controller was correcting and at which frequency. This behavior provides insights about possible disturbances. We define a potential peak as every point in a series of data points which has a value higher than both of its neighboring data points. We filter out the irrelevant peaks, i.e. peaks due to noise, by specifying that a peak should have a minimum prominence of 45 dbm and a minimum value of -105

dbm. The prominence of a peak is defined as the vertical distance to the highest valley. The minimum height and the minimum prominence are specific to the controller output signal and learned from experimental evaluations. Based on this peak detection we implemented three feature extraction algorithms that identify peak related characteristics.

### 3.3.1 Number of Peaks per Area

We divided the frequency range in smaller regions, each covering 5000 Hz, and count the number of peaks. With a maximum frequency of 160000 Hz we have a total number of 32 features.

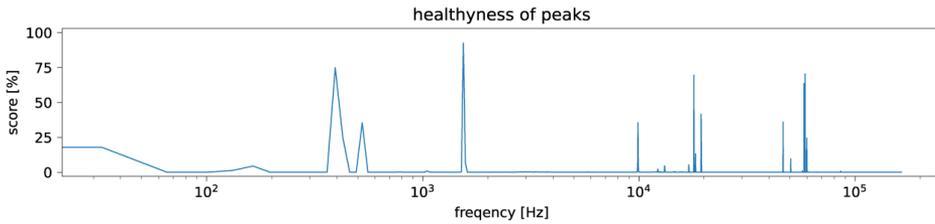
### 3.3.2 Characteristics of the Most Prominent Peaks

We identified the five most prominent peaks, from which we extract the prominence, the height, the width, and the frequency. While the prominence, the height, and the width are numerical values, the frequency is a categorical value because a higher frequency does not imply a worse or better system condition. Therefore, we again divided the frequency range into regions of 5000 Hz, and for each region we count the number of prominent peaks.

### 3.3.3 Peak Healthiness

This feature extraction method gives each extracted peaks following our set of <sup>observed PSDs</sup> constraints a score between 0 and 1 that determines whether the peak belongs to a healthy or unhealthy operation. For that we acquired controller output data during healthy operation and extracted all peaks following our criteria from the PSDS. Based on these peaks we assigned each frequency  $f$  a healthiness score  $healthiness(f) = \frac{\# \text{ peaks at } f}{\# \text{ peaks at } f}$ . The resulting distribution of healthy peaks is depicted in Figure 4.

In the feature extraction step, we identify the ten most prominent peaks and for each peak we take the healthiness score from the previously determined distribution as a feature.



**Figure 4: Probability distribution of frequencies having a healthy peak**

Source: own.

### 3.4 Autoencoder Latent Space

We use a feedforward AutoEncoder (AE) [22] trained on PSDs from healthy and disturbed operations. Using the AE’s encoder, we transform a complete PSD into the AE latent space vector, which is used as a feature vector for fault detection methods. The basic structure of the AE is shown in Table 2. The AE consists only of fully connected layers and each layer, except for the output layer, is followed by the leakyRELU activation function.

**Table 2: Overview Autoencoder**

Layer	input	1. encoding	2. encoding	latent space	1. decoding	2. decoding	output
Dimension	5000	500	100	10	100	500	5000

## 4 Selected Models

In this section, we describe what kind of algorithms we use to model the behavior of the laser oscillator based on the controller’s output signal. The purpose of these algorithms is to automatically decide whether the laser oscillator is currently disturbed or not. We divided the fault detection algorithms into the classes: clustering algorithms, outlier detection algorithms, and other algorithms that are neither based on clustering nor outlier detection.

## 4.1 Clustering Algorithms

Clustering algorithms aim to group data samples into classes with similar elements. Clustering requires the concept of a metric, which may differ from algorithm to algorithm [23]. For the purpose of fault detection, we assume that similar data samples belong to the same class. We use the following clustering algorithms:

- Clustering based local outlier factor (CBLOF) [24]
- K-means clustering [25]
- Balanced iterative reducing and clustering using hierarchies (BIRCH) [26]
- Gaussian mixture model (GMM) [27]

## 4.2 Outlier Detection Algorithms

Outlier detection algorithms aim to identify rare items or events that differ significantly from the rest of the dataset [28]. Assuming that faulty data samples can be classified as outliers compared to healthy data samples, we use the following outlier detection algorithms:

- Local outlier factor (LOF) [29]
- Angle-based outlier detection (ABOD) [30]
- Connectivity-based outlier detection (COF) [31]
- Isolation-based outlier detection (IOF) [32]
- K-nearest neighbor detection (KNN) [33]
- Copula-based outlier detector (COPOD) [34]
- Empirical cumulative distribution outlier detection (ECOD) [35]
- Linear model deviation-based outlier detection (LMDD) [36]
- One-class support vector machine (OCSVM) [37]
- Stochastic outlier selection (SOS) [38]

## 4.3 Other algorithms

In addition to the clustering algorithms and outlier detection algorithms, we use the following algorithms to detect a disturbed system:

- Kernel density estimation (KDE) [39]

- Kernel principal component analysis (KPCA) [40]
- Minimum covariance determinant (MCD) [41]
- Principal component analysis (PCA) [42]
- Sampling [43]

In addition to the algorithms fitted on the feature dataset, we trained an AE with the structure shown in Figure 2 on PSDs belonging to healthy system operation. Therefore, the AE only learns to reconstruct PSDs belonging to a healthy operation. The AE fault detector uses a threshold on the reconstruction loss, which is realized using the mean squared error (MSE) between the input PSD and the reconstructed PSD at the output layer. The fault detection is based on the assumption that PSDs belonging to healthy system operation have a low MSE, while PSDs belonging to poor system conditions have a high MSE.

## **5 Experimental Evaluation**

The experiments were performed using the Python libraries `tsfresh` [21], `PyOD` [44], and `Scikit-learn` [45]. The runtimes were measured on a Windows 11 operating system running Python 3.9 with a processor Intel(R) Core(TM) i7-1185G7 @ 3.00 GHz and 16 GB of RAM.

### **5.1 Dataset Summary**

To evaluate the feature extraction technique and fault detection algorithms we generated disturbances at different frequencies by playing tones of single frequencies. The tones were played through a surface speaker mounted directly on the optical table next to the laser oscillator one after the other at the same power. For evaluating the combination of feature extraction we recorded fitting data and validation data under the same conditions as summarized in Table 3. From both, the time frames and the PSDs, we extracted the features as described in Section 3 and normalized the extracted features using Z-normalization [46]. The number of features per data frame depends on the feature extraction method and is shown in Table 4. The peak characteristic feature extraction leads to the highest dimension and the AE latent space feature extractor to the lowest.

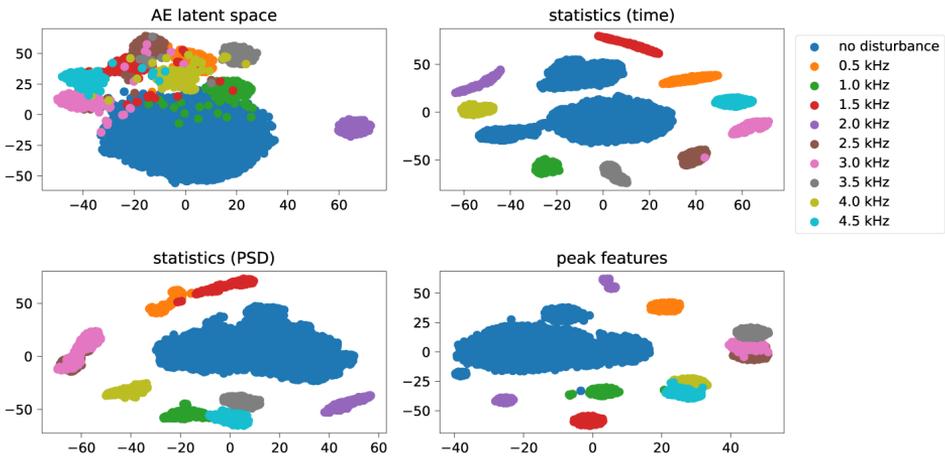
Figure 5 shows the four fitting data sets of the four different feature extraction methods (AE latent space, statistics from the time domain, statistics from PSDs, and peak features).

**Table 3: Summary of fitting dataset and validation dataset**

Condition	Fitting data		Validation data	
	# Frames	Portion	# Frames	Portion
no disturbance	4208	60.49 %	231	7.97 %
0.5 kHz disturbance	305	4.38 %	296	8.97 %
1.0 kHz disturbance	305	4.38 %	296	8.97 %
1.5 kHz disturbance	306	4.4 %	296	10.22 %
2.0 kHz disturbance	305	4.38 %	296	10.22 %
2.5 kHz disturbance	305	4.38 %	296	10.22 %
3.0 kHz disturbance	305	4.38 %	231	10.25 %
3.5 kHz disturbance	306	4.4 %	296	10.22 %
4.0 kHz disturbance	306	4.4 %	296	10.22 %
4.5 kHz disturbance	306	4.4 %	296	10.22 %

**Table 4: Numer of features per dataframe**

Feature Extraction Method	statistics (time)	statistics (PSD)	Peak characteristics (PSD)	AE latent space
Number of extracted features	34	34	94	10



**Figure 5: Feature vizualization by t-SNE**

Source: own.

To represent the multidimensional feature datasets in a two-dimensional space, we used t-distributed Stochastic Neighbor Embedding (t-SNE) [47]. The visualization of the data is intended to provide a basis for evaluating the algorithms in Section 5.4.

It can be seen that the data points recorded under the same disturbances or no disturbance are clustered for all feature extraction methods. However, the clusters on the AE latent space dataset are significantly closer, sometimes even with an overlap, than the clusters on the other feature datasets. Since we are analyzing fault detection methods in this work, it is noteworthy that there is only an overlap between disturbed data points and undisturbed data points on the AE latent space dataset. In particular, the data from the 1.5 kHz disturbance have a strong overlap with the undisturbed data points. Using statistics from time series, statistics from PSDs, or peak features there are only overlapping clusters between data points of different disturbance types. Furthermore, it is noticeable that the undisturbed datapoints based on time statistics form two separate clusters rather than one cluster.

## 5.2 Algorithms Parameters

Most of the algorithms selected contain controllable parameters that influence different aspects of the algorithm. A summary of all the parameters used is given in Tables 5, 6, 7.

**Table 5: Clustering algorithms’ parameters**

Algorithm	Parameter	Values
BIRCH	threshold	0.2, 0.4, ..., 3.8, 4.0
	branching factor	20, 40, 60, 80, 100
CBLOF	# clusters	2, 3, 4, 5, 6, 7, 8, 9, 10, 15
	alpha	0.5, 0.6, 0.7, 0.9
	beta	1.5, 2, 5, 7, 10, 15
GMM	# components	2, 3, 4, 5, 6, 7, 8, 9, 10, 15
K-means	# clusters	2, 3, 4, 5, 6, 7, 8, 9, 10, 15

**Table 6: Outlier detection algorithms’ parameters**

Algorithm	Parameter	Values
ABOD	# nearest neighbors	5, 10, ..., 95, 100
COF	# nearest neighbors	5, 10, ..., 95, 100
COPOD	-	-
ECOD	-	-
IOF	-	-
KNN	# nearest neighbors	5, 10, ..., 95, 100

Algorithm	Parameter	Values
LMDD	-	-
LOF	# nearest neighbors	5, 10, ..., 95, 100
OCSVM	-	-
SOS	perplexity	5, 10, ..., 95, 100

**Table 7: Other algorithms' parameters**

Algorithm	Parameter	Values
KDE	bandwith	0.4, 0.6, ..., 2.0, 2.2
KPCA	# components	1, 2, ..., 10
MCD	-	-
PCA	# components	1, 2, ..., 10
Sampling	-	-

### 5.3 Performance Criteria

We repeated the experiment under similar environmental conditions to verify the quality of the algorithms (see Table 3). The validation data consists of 2896 data frames, each covering 0.1 s. Therefore, a live fault detection requires a maximum inference duration of 289.6 s on the whole validation dataset. To evaluate the system state at each point in time, it is necessary that the fault detection algorithms also operate at such a high speed. We evaluate this criterion by measuring the time it takes each algorithm to classify the validation data samples and determine the inference speed by dividing the measured duration by the number of frames. Additionally, we measured the time each algorithm needs to be fitted.

To evaluate the feature extraction methods and algorithms qualitatively we are using the area under the receiver operating characteristic (AUROC) [48] as a performance metric. The AUROC score is defined as the area underneath the ROC curve and ranges between 0 and 1, where a score of 1 implies a perfect predictor, an AUROC score of 0 implies that the predictor gives always wrong predictions, and an AUROC score of 0.5 indicates that the predictor makes random guesses. We calculate the AUROC scores of both, the fitting dataset and the validation dataset.

The AUROC metric does not provide information on which of the disturbances are classified correctly and which of them are misclassified. Therefore, we also calculate the classification accuracy  $\text{TruePredictions}(\text{condition})$  for each condition, either a disturbed frequency or undisturbed respectively.

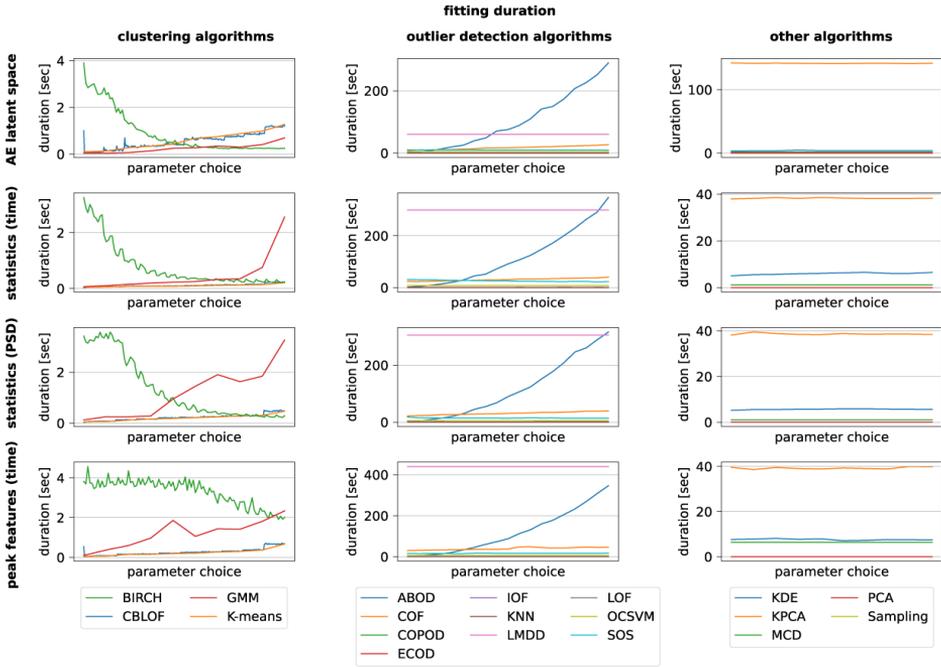
We repeated the process of fitting and evaluation ten times with different random seeds and determined the mean value for each metric. In summary, we determined the mean values of the following metrics for each combination of feature extraction method and fault detection algorithm:

- Fitting duration (fitting dataset)
- Inference duration (validation dataset)
- AUROC score (fitting dataset)
- AUROC score (validation dataset)
- Condition specific accuracies (validation dataset)

## **5.4 Results**

In this section, we describe the results of the algorithms applied to the experimental data. Combining the feature extraction methods and the different parameter choices, we built 3084 models on the different feature datasets (AE latent space, statistics from time series signals, statistics from PSDs, peak characteristics).

The fitting durations of all algorithms related to the feature extraction method and the choice of parameters are depicted in Figure 6. The fitting durations only include the fitting of the algorithms and not the transformation of the recorded data into the features. All clustering algorithms require very little time to be fitted for all feature extraction methods and all parameter choices. Among the outlier detection algorithms, the LMDD algorithm and ABOD have by far the longest fitting durations. Among the other algorithms, KPCA needs the longest time to be fitted for all feature extraction methods. It is noticeable that KPCA using the AE latent space features takes more than 100 seconds longer to be fitted than the other feature extraction methods.



**Figure 6: Fitting durations**  
Source: own.

Figure 7 shows the duration needed by the algorithms to classify the validation data samples. The inference duration consists of both the feature extraction part and the algorithmic classification part. The feature extraction methods are based on very efficient signal processing algorithms, such as fast Fourier transforms or basic statistical calculations. Therefore, feature extraction has a small impact on the overall inference duration. The maximum allowed inference duration is 289.6 s. This criterion is fulfilled by all algorithms for all feature extraction methods and all parameter settings. All clustering algorithms perform particularly well, followed by the other algorithms and the outlier detection algorithms. For all feature extraction methods, ABOD has the worst inference duration when many nearest neighbors are used. The LMDD algorithm has the second highest inference duration.

In the following, we describe the ability of the algorithms to classify disturbed data samples as disturbed and non disturbed data samples as normal. For both the fitting dataset and the validation dataset, we manually assigned a label to each data sample,

either undisturbed or disturbed. The AUROC score is based on the manually assigned reference and the labels assigned by the fault detection algorithms.

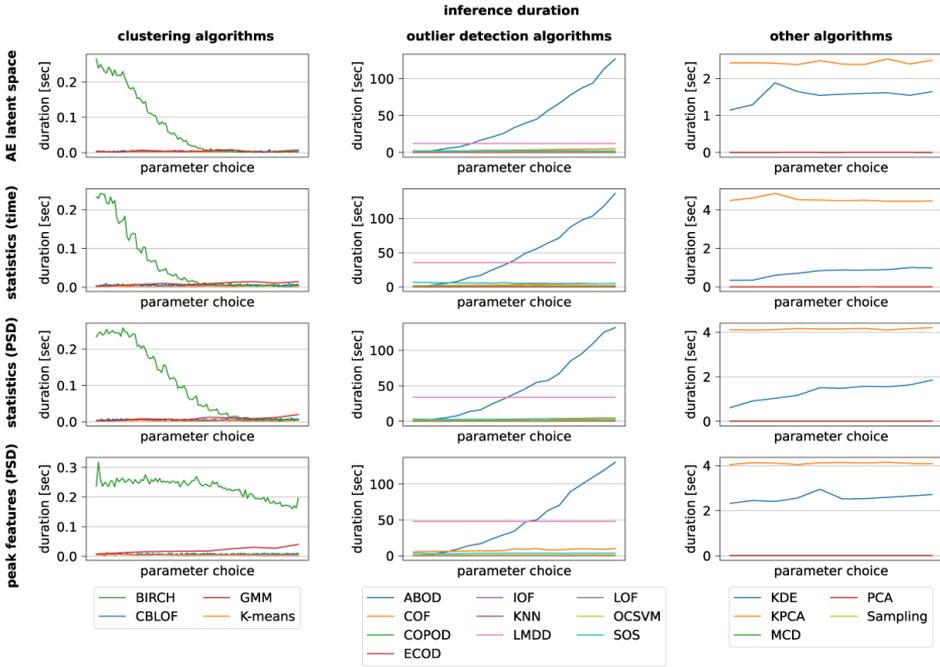


Figure 7: Inference durations

Source: own.

### 5.4.1 Clustering Algorithms

The AUROC results and the condition specific accuracies of the clustering algorithms with respect to the feature extraction methods are depicted in Figure 8.

In general, features from the PSDs (statistics, and peak features) form a good basis for clustering algorithms to reliably identify disturbed laser oscillator feedback systems, since all clustering algorithms except the GMM achieve very good AUROC scores and high accuracies for all conditions. The GMM algorithm does not achieve satisfactory results for any combination of parameter setting and feature extraction method. It is noticeable that the condition specific accuracies obtained by GMM show that the GMM algorithm classifies all data samples as disturbed. From the

results of the CBLOF algorithm, it can be seen that the fault detection quality is strongly dependent on the choice of input parameter. At an alpha of 0.5, the best AUROC scores of 1.0 are obtained on the fitting and validation dataset regardless of the choice of cluster number and beta for all feature extraction methods. Birch, and the K-Means algorithms achieve perfect results on the validation dataset for features from PSDs and the correct parameter choice.

The very good results of the clustering algorithms can be described with the help of the structure of the examined data. As the t-SNE embeddings of the data set already indicate (see Figure 5), the data measured under similar conditions are positioned in cluster-like structures, especially using the PSD statistics and the peak features.

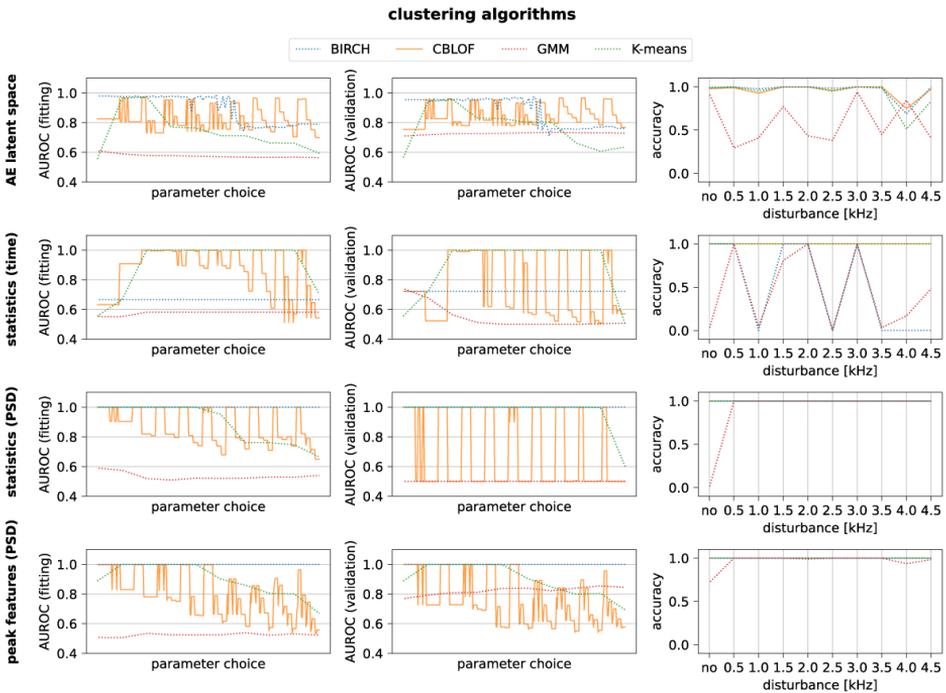


Figure 8: Results of clustering algorithms

Source: own.

### 5.4.2 Outlier Detection Algorithms

The AUROC scores and the condition-dependent accuracies of the outlier detection algorithms are shown in Figure 9.

In general, it can be seen that no combination of outlier detection algorithm and feature extraction method achieves a perfect AUROC score of 1.0. It is also noticeable that the choice of parameters for the outlier detection algorithms has no great influence on the result, because the maximum AUROC scores hardly differ from the minimum AUROC scores per algorithm. In contrast to the clustering algorithms, the outlier detection algorithms achieve very poor AUROC scores on the feature datasets that use PSDs as a basis. Among all outlier detection algorithms, KNN achieves the highest AUROC score of 0.9148 using the AE latent space as features. The corresponding condition specific conditions show that the data recorded under no excitation are correctly classified with an accuracy of 0.8788. The accuracies that KNN detects an excited system from the controller data are all higher than 0.9.

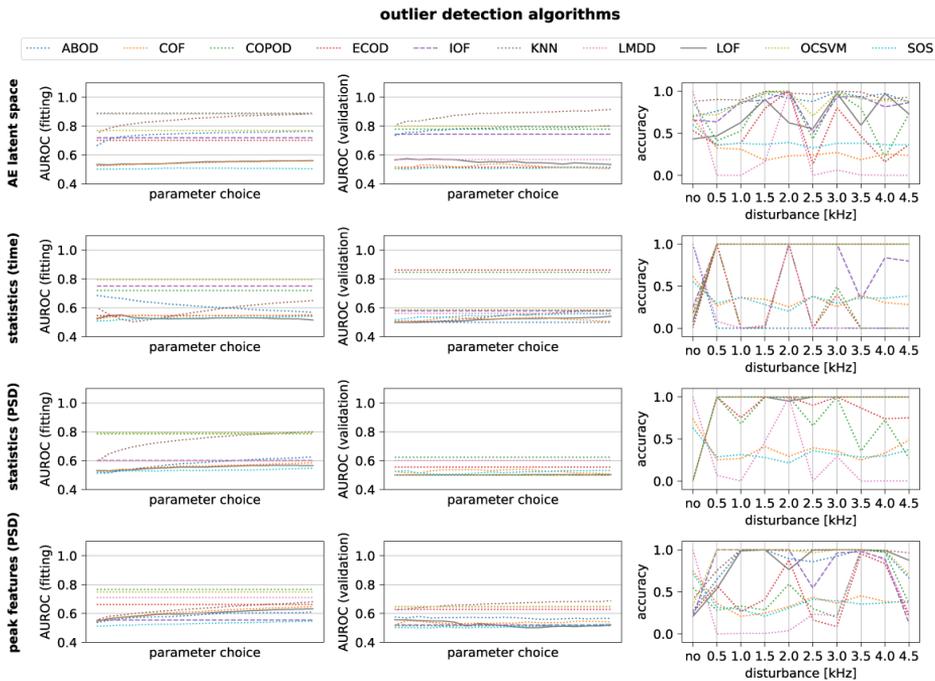


Figure 9: Results of outlier detection algorithms

Source: own.

### 5.4.3 Other Algorithms

The AUROC scores and the condition-specific accuracies of the other algorithms depending on the feature extraction methods used are shown in Figure 10.

It is noticeable that all algorithms that were fitted with PSD statistics are predictors that classify all validation data as disturbed. This implies that the algorithms cannot generalize the error detection learned on the PSD statistics fitting dataset because not all data samples from the fitting data set are classified as disturbed. Furthermore, it can be seen that similar to the outlier detection algorithms, none of the other algorithms achieve a perfect AUROC score of 1.0 on the validation dataset.

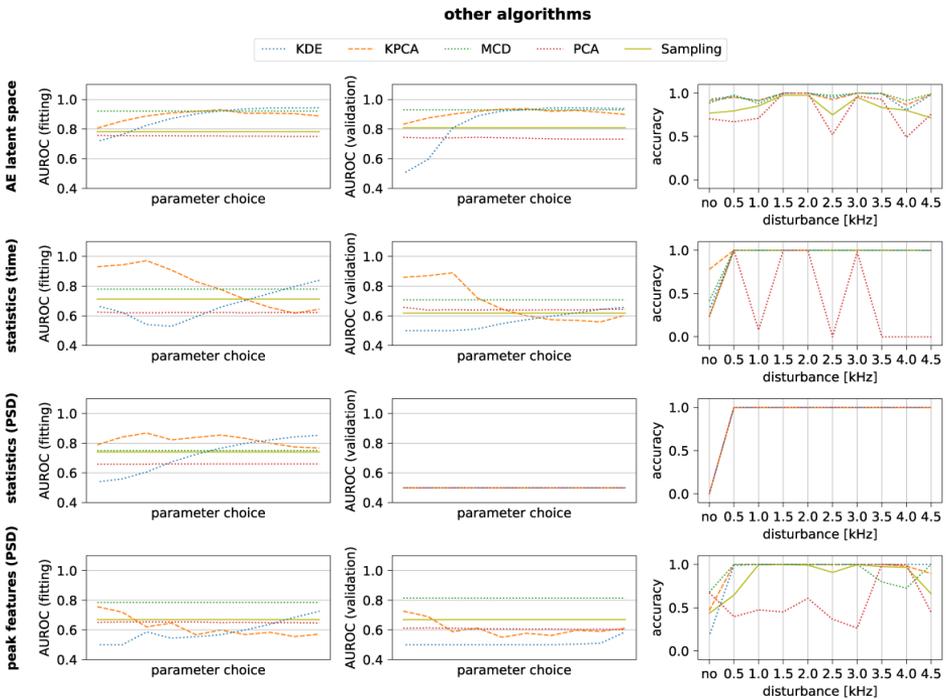


Figure 10: Results of other algorithms

Source: own.

For all feature extraction methods KPCA achieves the highest AUROC scores, with the highest value of 0.94 being achieved with the AE latent space as the feature. The number of principal components leading to the highest AUROC scores for the

respective feature extraction methods differ. Therefore, there exists no correct choice of principal components such that KPCA can describe the error detection behavior for all feature datasets. The second highest AUROC score on the validation dataset is also achieved on the AE latent space by the MCD algorithm.

In addition, the results of the AE trained on non-disturbed PSDs are summarized in Table 8. The AE fault detector achieves perfect AUROC scores of 1.0 on both the fitting and validation datasets.

**Table 8: Autoencoder fault detector results**

Training duration	Inference duration	AUROC (fitting)	AUROC (validation)
220.8 s	0.133498 s	1.0	1.0

#### 5.4.4 Summary

Table 9 gives an overview of the algorithms and their parameter configuration that achieve an AUROC score higher than 0.95 on the validation dataset. If an algorithm achieves such an AUROC score with multiple parameter combinations, we selected the parameter combination that gives the best AUROC score and the lowest inference duration.

The AE works directly on the PSDs. Therefore, no prior feature extraction is required. Among the algorithms that require prior feature extraction, only the clustering algorithms K-means clustering and CBLOF achieve very good AUROC scores on all validation datasets. Additionally, BIRCH achieves very good AUROC scores on the validation dataset using either the AE latent space, PSD statistics, or peak characteristics. Furthermore, it stands out that no algorithm which is fitted with the AE latent space achieves a perfect AUROC score. The best algorithms that do not belong to the clustering algorithms are KPCA having an AUROC score of 0.9368 and KDE with an AUROC score of 0.9436, both using the AE latent space as feature.

As described in Section 5.1, the feature datasets each form clusters according to the type of disturbance, which explains why clustering algorithms in particular work so well. The overlap of disturbed and undisturbed data on the AE latent space dataset

(see Figure 5) also constitutes for the fact that none of the selected algorithms achieves a perfect AUROC score when using the AE latent space as feature input.

**Table 9: Best fault detection results on the validation dataset**

Feature	Algorithm	Parameter	AURC	Inference duration in s
-	AE	-	1.0	0.1335
AE latent space	BIRCH	threshold: 2.4	0.9726	0.007
		branching factor: 80		
	K-means	# clusters: 4	0.9559	0.0014
	CBLOF	# clusters: 10	0.9667	0.0018
alpha: 0.5				
beta: 5				
statistics (time)	CBLOF	# clusters: 9	1.0	0.002
		alpha: 0.6		
		beta: 1.5		
	K-means	# clusters: 4	1.0	0.002
statistics (PSD)	BIRCH	threshold: 3.8	1.0	0.0019
		branching factor: 20		
	CBLOF	# clusters: 2	1.0	0.0018
		alpha: 0.5		
		beta: 1.5		
K-means	# clusters: 5	1.0	0.0019	
peak characteristics	BIRCH	threshold: 4.0	1.0	0.1606
		branching factor: 40		
	CBLOF	# clusters: 2	1.0	0.0018
		alpha: 0.7		
		beta: 1.5		
K-means	# clusters: 6	1.0	0.0036	

## 6 Conclusion

In this paper, we investigated the ability of data-based fault detection algorithms in combination with four feature extraction methods to model the condition of an actively controlled phase-locked laser oscillator and determined the best methods and parameters for detecting disturbances that affect the healthy operation of the synchronization system. The fault detection methods were validated experimentally by disturbing the system acoustically. We evaluated the classification performance for each combination of feature extraction, fault detection method, and algorithmic-specific parameters using the fitting duration, inference duration, and AUROC scores as quality measures.

From the classification results, we can conclude that very good prediction results can be obtained without deep system expertise. Comparing the prediction results of the different types of algorithms, we notice that clustering algorithms achieve the best results regardless of the feature extraction methods. Moreover, there is no combination of an algorithm not belonging to the clustering algorithms and a feature extraction method that achieves a perfect AUROC score on the validation dataset. Additionally, there is no combination of a fault detection algorithm and the AE latent space as a feature extractor that achieves a perfect AUROC score on the validation dataset. With an AUROC score of 1.0 and a inference duration of 0.0018 s when applied to the validation dataset, the combination of CBLOF and peak characteristics or the combination of CBLOF and statistics from PSDS achieve the best results. However, we would like to draw particular attention to the performance of the AE fault detector, as it does not require prior feature extraction and can thus be applied directly to any dynamic system controlled in a closed loop. In addition, the inference time for the validation dataset is below the maximum acceptable threshold for real-time fault detection.

The experimental evaluation used in this work is based on the excitation of different frequencies at the same level by a surface loudspeaker. For future work, we plan to investigate what minimum interference intensity must be present for a fault detection algorithm to be effective and to extend the fault detection mechanism by specifying the exact type of fault, rather than just a binary classification of healthy or disturbed. We also want to extend the fault detection mechanism to a predictive maintenance module that can predict when the next faulty operating point will occur.

### **Acknowledgement**

We acknowledge the support by DASHH (Data Science in Hamburg - HELMHOLTZ Graduate School for the Structure of Matter) with the Grant-No. HIDSS-0002.

## References

- [1] Sobolev E, Zolotarev S, Giewekemeyer K, Bielecki J, Okamoto K, Reddy HKN, et al. Megahertz single-particle imaging at the European XFEL. *Communications Physics*. 2020;3(1).
- [2] Schulz S, Czwalinna M, Felber M, Fenner M, Gerth C, Kozak T, et al, editors. *Few-Femtosecond Facility-Wide Synchronization of the European XFEL*: JACoW Publishing, Geneva, Switzerland; 2019.
- [3] Heuer M. Identification and control of the laser-based synchronization system for the European X-ray Free Electron Laser [doctoralThesis]. Technische Universita't Hamburg-Harburg; 2018. Available from: <http://tubdok.tub.tuhh.de/handle/11420/1706>.
- [4] Welch P. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*. 1967;15(2):70-3.
- [5] Zheng H, Wang R, Yin J, Li Y, Lu H, Xu M. A new intelligent fault identification method based on transfer locality preserving projection for actual diagnosis scenario of rotating machinery. *Mechanical Systems and Signal Processing*. 2020;135:106344. Available from: <https://www.sciencedirect.com/science/article/pii/S0888327019305655>.
- [6] Siahpour S, Li X, Lee J. Deep learning-based cross-sensor domain adaptation for fault diagnosis of electro-mechanical actuators. *International Journal of Dynamics and Control*. 2020;8(4):1054-62.
- [7] Bagheri M, Zollanvari A, Nezhivenko S. Transformer Fault Condition Prognosis Using Vibration Signals Over Cloud Environment. *IEEE Access*. 2018;6:9862-74.
- [8] Gamal M, Donkol A, Shaban A, Costantino F, Di G, Patriarca R. Anomalies detection in smart manufacturing using machine learning and deep learning algorithms. In: *Proceedings of the International Conference on Industrial Engineering and Operations Management, Rome, Italy; 2021*. p. 1611-22.
- [9] Lopez F, Saez M, Shao Y, Balta EC, Moyne J, Mao ZM, et al. Categorization of Anomalies in Smart Manufacturing Systems to Support the Selection of Detection Mechanisms. *IEEE Robotics and Automation Letters*. 2017;2(4):1885-92.
- [10] Quevedo J, Puig V, Escobet T. Model Fault Detection of Feedback Systems: How and Why to Use the Output of the PID Controller? *IFAC Proceedings Volumes*. 2000;33(4):319-24. *IFAC Workshop on Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, 5-7 April 2000*. Available from: <https://www.sciencedirect.com/science/article/pii/S1474667017382630>.
- [11] Puig V, Quevedo J. Fault-tolerant PID controllers using a passive robust fault diagnosis approach. *Control Engineering Practice*. 2001;9(11):1221-34. *PID Control*. Available from: <https://www.sciencedirect.com/science/article/pii/S0967066101000685>.
- [12] Bauer M, Auret L, Bacci di Capaci R, Horch A, Thornhill NF. Industrial PID Control Loop Data Repository and Comparison of Fault Detection Methods. *Industrial & Engineering Chemistry Research*. 2019;58(26):11430-9. Available from: <https://doi.org/10.1021/acs.iecr.8b06354>.
- [13] Wang Q, Liu J, Wei B, Chen W, Xu S. Investigating the Construction, Training, and Verification Methods of k-Means Clustering Fault Recognition Model for Rotating Machinery. *IEEE Access*. 2020;8:196515-28.
- [14] Duong BP, Kim JM. Non-Mutually Exclusive Deep Neural Network Classifier for Combined Modes of Bearing Fault Diagnosis. *Sensors*. 2018;18(4). Available from: <https://www.mdpi.com/1424-8220/18/4/1129>.
- [15] Kim D, Heo TY. Anomaly Detection with Feature Extraction Based on Machine Learning Using Hydraulic System IoT Sensor Data. *Sensors*. 2022;22(7). Available from: <https://www.mdpi.com/1424-8220/22/7/2479>.
- [16] Li H, Yun Xiao D. Fault diagnosis based on power spectral density basis transform. *Journal of Vibration and Control*. 2015;21(12):2416-33. Available from: <https://doi.org/10.1177/1077546313487242>.

- [17] Wang Z, Yang J, Li H, Zhen D, Xu Y, Gu F. Fault Identification of Broken Rotor Bars in Induction Motors Using an Improved Cyclic Modulation Spectral Analysis. *Energies*. 2019;12(17). Available from: <https://www.mdpi.com/1996-1073/12/17/3279>.
- [18] Sundaram S, Zeid A. Smart Prognostics and Health Management (SPHM) in Smart Manufacturing: An Interoperable Framework. *Sensors*. 2021;21(18). Available from: <https://www.mdpi.com/1424-8220/21/18/5994>.
- [19] Zhang N, Chen E, Wu Y, Guo B, Jiang Z, Wu F. A novel hybrid model integrating residual structure and bi-directional long short-term memory network for tool wear monitoring. *The International Journal of Advanced Manufacturing Technology*. 2022;120(9):6707-22.
- [20] Meng Q, Catchpole DR, Skillicorn DB, Kennedy PJ. Relational Autoencoder for Feature Extraction. *CoRR*. 2018;abs/1802.03145. Available from: <http://arxiv.org/abs/1802.03145>.
- [21] Christ M, Braun N, Neuffer J, Kempa-Liehr AW. Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*. 2018;307:72-7. Available from: <https://www.sciencedirect.com/science/article/pii/S0925231218304843>.
- [22] Tschannen M, Bachem O, Lucic M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:181205069*. 2018.
- [23] Xu R, Wunsch D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*. 2005;16(3):645-78.
- [24] He Z, Xu X, Deng S. Discovering cluster-based local outliers. *Pattern Recognition Letters*. 2003;24(9):1641-50. Available from: <https://www.sciencedirect.com/science/article/pii/S0167865503000035>.
- [25] Hartigan JA, Wong MA. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society Series C (Applied Statistics)*. 1979;28(1):100-8. Available from: <http://www.jstor.org/stable/2346830>.
- [26] Zhang T, Ramakrishnan R, Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases. *SIGMOD Rec*. 1996 jun;25(2):103-14.
- [27] Reynolds DA. Gaussian mixture models. *Encyclopedia of biometrics*. 2009;741(659-663).
- [28] Singh K, Upadhyaya S. Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*. 2012;9(1):307.
- [29] Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: Identifying Density-Based Local Outliers. *SIGMOD Rec*. 2000 may;29(2):93-104. Available from: <https://doi.org/10.1145/335191.335388>.
- [30] Kriegel HP, Schubert M, Zimek A. Angle-Based Outlier Detection in High-Dimensional Data. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '08. New York, NY, USA: Association for Computing Machinery; 2008. p. 444-52.
- [31] Wang Y, Li K, Gan S. A Kernel Connectivity-based Outlier Factor Algorithm for Rare Data Detection in a Baking Process. *IFAC-PapersOnLine*. 2018;51(18):297-302. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018. Available from: <https://www.sciencedirect.com/science/article/pii/S2405896318319980>.
- [32] Liu FT, Ting KM, Zhou ZH. Isolation-Based Anomaly Detection. *ACM Trans Knowl Discov Data*. 2012 mar;6(1).
- [33] Angiulli F, Pizzuti C. Fast outlier detection in high dimensional spaces. In: *European conference on principles of data mining and knowledge discovery*. Springer; 2002. p. 15-27.
- [34] Li Z, Zhao Y, Botta N, Ionescu C, Hu X. COPOD: Copula-Based Outlier Detection. In: *2020 IEEE International Conference on Data Mining (ICDM)*; 2020. p. 1118-23.
- [35] Li Z, Zhao Y, Hu X, Botta N, Ionescu C, Chen G. ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering*. 2022:1-1.
- [36] Arning A, Agrawal R, Raghavan P. A Linear Method for Deviation Detection in Large Databases. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. AAAI Press; 1996. p. 164-9.

- [37] Schoelkopf B, Williamson R, Smola A, Shawe-Taylor J, Platt J. Support Vector Method for Novelty Detection. In: Proceedings of the 12th International Conference on Neural Information Processing Systems. NIPS'99. Cambridge, MA, USA: MIT Press; 1999. p. 582-8.
- [38] Janssens J, Huszar F, Postma E, van den Herik H. Stochastic outlier selection. Tilburg centre for Creative Computing, techreport. 2012;1:2012.
- [39] Parzen E. On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics. 1962;33(3):1065-1076. Available from: <https://doi.org/10.1214/aoms/1177704472>.
- [40] Hoffmann H. Kernel PCA for novelty detection. Pattern Recognition. 2007;40(3):863-74. Available from: <https://www.sciencedirect.com/science/article/pii/S0031320306003414>.
- [41] Rousseeuw PJ, Driessens KV. A Fast Algorithm for the Minimum Covariance Determinant Estimator. Technometrics. 1999;41(3):212-23. Available from: <https://www.tandfonline.com/doi/abs/10.1080/00401706.1999.10485670>.
- [42] Shyu ML, Chen SC, Sarinnapakorn K, Chang L. A novel anomaly detection scheme based on principal component classifier. Miami Univ Coral Gables FL Dept of Electrical and Computer Engineering; 2003.
- [43] Sugiyama M, Borgwardt K. Rapid distance-based outlier detection via sampling. Advances in neural information processing systems. 2013;26.
- [44] Zhao Y, Nasrullah Z, Li Z. PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of Machine Learning Research. 2019;20(96):1-7. Available from: <http://jmlr.org/papers/v20/19-011.html>.
- [45] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825-30.
- [46] Goldin DQ, Kanellakis PC. On similarity queries for time-series data: Constraint specification and implementation. In: Montanari U, Rossi F, editors. Principles and Practice of Constraint Programming – CP '95. Berlin, Heidelberg: Springer Berlin Heidelberg; 1995. p. 137-53.
- [47] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(11).
- [48] Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006;27(8):861-74. ROC Analysis in Pattern Recognition. Available from: <https://www.sciencedirect.com/science/article/pii/S016786550500303X>.