Faculty of Science and Engineering
Department of Advanced Computing Sciences

Anomaly Detection for Predictive Maintenance of Industrial Robots: Final Report

MSc DSDM & MSc AI

Research Project - Group 18

Authors
Cristina Afanasii (i6230178), Gabriel Graells Solé (i6376459), Enders Turkers (i6238665), Sonja Tang (i6345924), Ludwig Schulte (i6216104), Mario Real Enrique (i6376763)

Supervisors
Prof. Dr. Anna Wilbik, Prof. Dr. Matus Mihalak
Dr. Marcin Pietrasik

Coordinator
Dr. Linda Rieswijk

Contents

1 Introduction **1**
1.1 Scope of the project . . . 1
1.2 Research Questions . . . 2

2 Background **2**
2.1 Process background . . . 2
2.2 Data . . . 3
2.3 Literature Review . . . 3

3 Methodology **5**
3.1 Real-World Anomalies . . . 5
3.1.1 Rail Alignment . . . 5
3.1.2 Rail Degradation . . . 5
3.1.3 Worn Bearings . . . 5
3.1.4 Gearbox Problems . . . 6
3.2 Synthetic Anomaly Simulation . . . 6
3.2.1 Point Anomaly . . . 6
3.2.2 Gaussian Noise Anomaly . . . 6
3.2.3 Shift Anomaly . . . 6
3.2.4 Sinusoidal Anomalies . . . 6
3.3 Data Treatment and Signal Processing . . . 6
3.3.1 Homogenization . . . 7
3.3.2 Interpolation . . . 7
3.3.3 Spectrum Decomposition . . . 7
3.4 Machine Learning . . . 8
3.4.1 Local Outlier Factor . . . 8
3.4.2 Isolation Forest . . . 9
3.5 Statistical Methods . . . 9
3.5.1 Baseline Statistical Model (Z-scores) . . . 9
3.5.2 Hidden Markov Models . . . 10
3.6 Evaluation . . . 12

4 Experiments And Results **13**
4.1 Synthetic Anomaly . . . 13
4.2 Signal Processing . . . 16
4.2.1 Spectrum Decomposition . . . 16
4.3 Machine Learning . . . 17
4.3.1 Local Outlier Factor . . . 17
4.3.2 Isolation Forest . . . 18
4.4 Statistical Methods . . . 18
4.4.1 Baseline Statistical Model . . . 18
4.4.2 Hidden Markov Model . . . 19
4.5 Evaluation . . . 21
4.5.1 Local Outlier Factor . . . 21
4.5.2 Baseline Statistical Model . . . 21
4.5.3 HMM . . . 22

5 Discussion **24**
5.1 Challenges and Limitations . . . 26

6 Conclusion & Future Work **26**
6.1 Future Work . . . 26

Bibliography **III**

A Anomalies - How they manifest in natural datasets **IV**

B Example of Decomposed Signal **VIII**

C Local Outlier Factor Experiment Results **VIII**

D Evaluation Hidden Markov Models **XII**

D.1 Posterior Probabilities Scatter Plots XII

D.2 Posterior Probabilities Histogram Plots XIII

1 Introduction

In the modern industrial landscape, digitalisation is playing an ever increasingly important role in manufacturing processes. This is exemplified in various facets of the entire process, but one approach that is generating more attention in recent time is the usage of artificial intelligence to enhance and streamline industrial processes. Over time, multi-purpose industrial robots have emerged as key elements to heighten productivity in factories.

In the domain of manufacturing, these industrial robots play a pivotal role in automating various tasks such as assembly, welding, material handling, etc. These robots operate in complex and dynamic environments, often performing repetitive and high-speed movements over extended periods. Unfortunately, because these robots are susceptible to wear and tear, component degradation, and unforeseen malfunctions, they can experience costly downtime, production delays, and safety hazards.

Predictive Maintenance is an approach that aims to avoid such costs and potential hazards. Predictive Maintenance helps avoid failures by monitoring the condition of the equipment and performing maintenance when the condition worsens instead of a specified maintenance schedule [1]. In order to perform effective Predictive Maintenance, problems must first be detected, and then reported upon. An effective way to do this is via Anomaly Detection. Anomaly detection typically refers to the data analysis technique in which data points/observations that deviate from the norm are identified [2]. These deviations are considered anomalies and they can represent failures in real life.

This project is performed for and in collaboration with VDL, an international manufacturing company from the Netherlands. VDL is known for producing vehicles of a wide range of use. Currently, VDL is taking a step further towards innovating the industry of battery production for electric cars using a new battery assembly line. The industrial process of producing batteries is complex and thus facilitated by several robotic systems such as industrial robot arms (e.g., IRB 66640 model 6-axis robot arm), a special track to move the robot itself, Automated Guided Vehicles (AGVs) and so on.

Thus Anomaly Detection can aid Predictive Maintenance by identifying anomalies as indicators or real life failure risks. In the context of VDL's battery line, robot maintenance can be performed when deviations from the robot's normal performance are detected using advanced analysis, thus prompting predictive maintenance via anomaly detection. This way costly downtime and production delays can be avoided.

1.1 Scope of the project

VDL aims to develop an anomaly detection system for the battery line robots, particularly focusing on the robots that move along a track powered by motors. The pivotal goal of this project is to identify and predict anomalies in the functioning of these robots, such as malfunctions or failures in the operating equipment, by analyzing the torque, speed, and position measurements produced by the robot.

Since anomalies in the track measurement (TL) are indicated by irregular spikes and/or shift in torque, the detection system must accurately identify such abnormalities and distinguish them from normal variability. This is crucial because deviations are representative of real world malfunctions or wear and tear damage that represent downtime risk factors.

The project comes with several challenges:

- Dataset - due to the novelty and the prototype nature of the battery line, data from the robots of interest is not available. In turn, data from robots with a similar functionality from a different production line was accessed and provided for the goals of this project.
- Anomalies - even though anomalies are indicative of breakdown risks, those occur quite rarely in natural datasets, thus anomalies should be simulated and injected in natural datasets for evaluation.
- Evaluation - Since natural datasets rarely contain obvious anomalies, evaluating the effectiveness of the solution poses a challenge. A special evaluation strategy has to be devised.
- Transferability - the data provided comes form robots that perform different tasks in different production lines with potentially different production environments. Thus, the identified solution has to be transferable to the battery line.

The project includes several aspects intended to address each of the challenges:

- **Data Collection and Enhancement:** Preprocess the data to clean and normalize it, ensuring it is suitable for analysis and model training. Simulate anomalies for robust model training and evaluation.
- **Model Development:** Train AI models that are appropriate for time-series anomaly detection such as Isolation Forests, Local Outlier Factor, and Hidden Markov Model. Devise a baseline statistical model for comparison.
- **Model Evaluation Strategy:** Employ synthetic anomaly simulation for evaluating the models' ability to identify different types of anomalies, using a Support Vector Machine.
- **Collaboration and Deployment:** Gain an in-depth understanding of the battery line operation, including the role of robots, AGVs, and the specifics of station operations. Plan for the integration of the anomaly detection model into VDL's existing monitoring systems, including considerations for real-time analysis and alerts.

1.2 Research Questions

Given the scope of our project, the main goal is to answer the following problem statement: Which methods give best results for anomaly detection in industrial robots for the battery line project at VDL?

Based on our problem statement, we have identified the following research questions:

RQ 1 How do anomalies manifest in natural datasets (i.e. data collected from similar robots within the company)

RQ 2 How can anomalies be simulated (physically or synthetically) in order to train an anomaly detection model?

RQ 3 What predictive approaches can be used to identify anomalies in industrial robots?

RQ 4 Which anomaly detection approach is best suited for a real-world application of predictive maintenance at the VDL's battery line?

RQ 5 How can such models be evaluated in the dearth of real-world anomaly data?

2 Background

2.1 Process background

Battery pack production consists of several steps performed in a specific order. First, the main battery module undergoes a quality check. Next, the battery is placed inside a casing and forwarded with an AGV to a human operator who installs the circuitry and cooling system of the battery. When the electrical connections are made, several tests are being performed to ensure adequate functioning of the battery. The AGVs transport the module to the next step of the process which is dedicated to screwing the lid on top of the casing – the step of the process that is under the main scope of our project. Lastly, a final leak test is performed to ensure water tightness.

The process of screwing the lid on top of the battery casing is carried out by two arm robots sliding across a track. Due to the variety of battery types that can be assembled, the tooling of the robot has to be changed accordingly. Such tooling is supplied to the robot arm also by AGVs. A video system detects the objects held by the robot and assesses their quality. Further, the robot arm picks the lid of the battery and presents it to the other robot arm which applies sealing material. The lid is then placed on top of the module while the assisting video systems ensure the alignment of the screwing holes is adequate. Further, the screws that are appropriate for the type of battery being assembled are sucked into a tube by the robot and then used to tighten the lid to the casing with the battery module inside.

The production process carried out by the robot arms is of particular interest to us due to their complex and varied tasks. Breakages can happen to the track along which the robot arms slide, causing unwanted downtime. Thus, the company is intent on avoiding such downtime by means of predictive maintenance on the aforementioned robots.

Table 1: Distribution of data provided by VDL. It portrays the amount of time series, robots and time series per robot for each model of robot arm.

Robot arm	n. series	n. series (w. dupl.)	n. robots	series/robot
IRB 6640	237	300	52	4 - 6
IRB 6700	14	14	5	1 - 5
IRB 7600	61	603	cca. 3	5 - 23

2.2 Data

VDL has granted access to a dataset. The data consists of several time series taken from multiple *ABB IRBT 6004* model tracks within the factory. Although these tracks are not part of the battery line, they use the same model as those in the battery line. The tracks accommodate three different models of robot arms with varying specifications. The data that was originally available had a large amount of duplicates, which were not immediately obvious to identify. The final distribution of data before and after removing duplicates is shown in Table 1.

The dataset comprises two types of measurements, denoted as "Ta" and "TL". The "Ta" group corresponds to a predetermined sequence of movements lasting 3.48 seconds that is consistent for each track but may vary between different devices. Following this, the "TL" series begins, where the robot arm is moved describing the programmed set of moves. Similarly, the sequence of movements is consistent for each machine but varies in shape and duration across different tracks.

The time-series data includes the following variables:

- Time.
- Position.
- Speed.
- Torque.
- Torque feedforward.

Position, speed and torque are measured in the motor, whereas torque feedforward is the predicted necessary torque to achieve the desired movement. This metric is obtained through a model of the system dynamics. By applying this estimated torque in advance, the controller can anticipate the motor's needs and preemptively counteract disturbances or deviations from the desired trajectory. This method works in conjunction with the feedback from sensors to maintain precise control over the motor. However, the "TL" measurements don't include the torque feedforward as a variable.

Throughout our analysis and experiments, we will exclusively use the "TL" measurements. In these sequences, the robot arm travels the entire length of the track, which is crucial since some anomalies may be caused by specific sections of the track rails. A sample of this data is portrayed in Figure 1.

2.3 Literature Review

Anomaly detection methodologies can be broadly categorized into three main groups: signal processing analysis, statistical analysis, and machine learning or deep learning techniques. It's worth noting that these categories are often intertwined to form comprehensive anomaly detection systems. Most existing methodologies follow a similar procedure: first, a non-anomalous dataset undergoes characterization through analysis techniques; next, a model is optimized using this curated data; finally, a decision mechanism is defined to determine whether a given input is anomalous.

Anomaly detection finds application in various fields, ranging from cybersecurity and healthcare to network management. Therefore, our literature review extended beyond anomaly detection in industrial robots to encompass general anomaly detection scientific publications.

The authors of [3] propose a Principal Component Analysis (PCA)-based detection system that measures anomalies using the reconstruction error of data points. They also introduce novel techniques to address PCA limitations. Other papers, such as [4], leverage PCA in combination with Kalman filtering, enhancing PCA through Karuhen-Loeve transform (KLT).

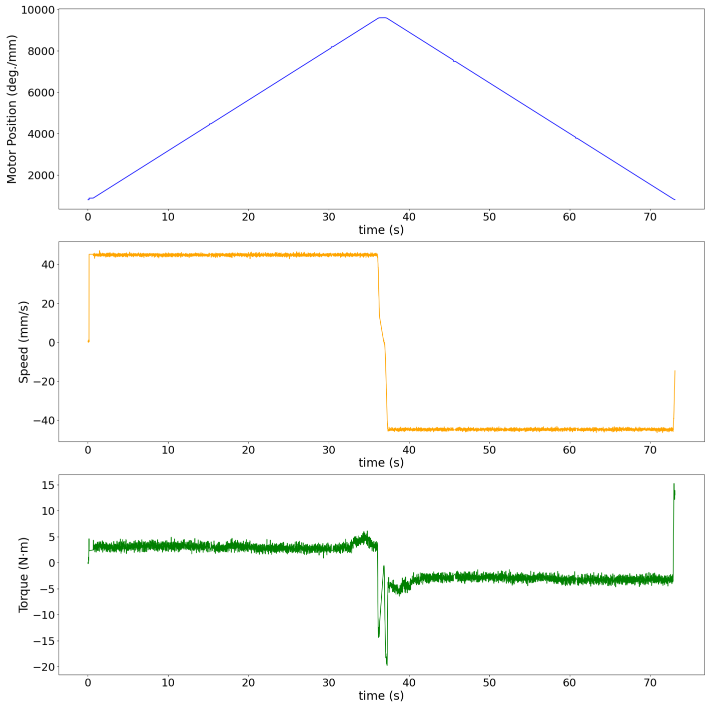


Figure 1: Data corresponding to a "TL" measurement.

Numerous studies advocate for Hidden Markov Models (HMMs) as a statistical modeling approach for anomaly detection [5] [6] [7]. In this approach, a dataset representing normal behavior is used to train an HMM. During inference, a series of observations is presented to the trained model, which then outputs the likelihood of observing the given sequence. An established threshold delineates anomalies from normal observations based on this likelihood estimation. The primary challenge with HMMs lies in defining a threshold that effectively balances the false-positive and true-positive rates.

Isolation forests feature prominently in various studies [8] [9] [10] [11]. These models train a set of trees to detect and isolate outliers (anomalies) within observations.

Furthermore, novel machine learning (ML) and deep learning (DL) techniques, particularly sequential models like Long Short-Term Memory (LSTM) and Transformers, are increasingly employed for anomaly detection [12] [13] [14]. These models excel at capturing long-term dependencies and patterns in sequential data, which are crucial for analyzing complex behavioral data in anomaly detection tasks. The papers [12–14] describe LSTM or Transformer models equipped with decision functions that determine the normality of a sequence.

Another approach for anomaly detection in predictive maintenance is Transfer Learning. In [15] the authors propose Transfer Learning as a solution to the problem of training data being different from the data collected in a real industrial scenario due to varying tasks performed by a single robot. Because of this problem, many approaches cause false alarms. The authors use manifold alignment to find common subspaces between the source and target domains. Features are extracted from the robot's speed, position, and torque using short-time Fourier Transform (STFT). Subsequently, these features are transformed into principal components using Principal Component Analysis (PCA), and then the distance between the train and test data is computed to determine if there are any anomalies.

Borgi et al. [16] use a Linear Regression approach based on the electrical signal of the robot's arm and the position accuracy of the robot. An anomaly is considered to occur when the actual position deviates from the prediction. This is established using correlation analysis.

A popular approach for anomaly detection is clustering, specifically k-means. An example of k-means being used to detect anomalies in predictive maintenance for industrial robots is presented in [11]. Additionally, [17] illustrates k-means clustering as a way to extract additional features. Subsequently, these features were used to train a Neural Network classifier.

In their extensive research, Riazi et al. [11] test a suite of different predictive methods for anomaly detection intended for an industrial robot arm that moves along a belt. Among the tested methods

Table 2: Research Methods and Papers

Category	Method	Papers
Signal Processing Analysis	PCA with Kalman Filtering	[3] [4]
	KLT with Kalman Filtering	[3] [4]
	Transfer Learning with STFT and PCA	[15]
Statistical Methods	PCA with Kalman Filtering	[3] [4]
	KLT with Kalman Filtering	[3] [4]
	Hidden Markov Model	[5] [6] [7]
	Transfer Learning with STFT and PCA	[15]
	Linear Regression	[16]
ML & DL	Isolation Forest	[8] [9] [10] [11]
	LSTM with SVM	[13] [14]
	Transformers	[12]
	Transfer Learning with STFT and PCA	[15]
	K-means with Neural Network	[11]

were clustering (k-means), classification (k-Nearest Neighbour, Support Vector Machine), outlier detection (Histogram Based Outlier Detection, Local Outlier Factor, Angle-Based Outlier Detection), and subspace methods (Isolation Forest, PCA). All listed methods perform well on datasets that had 90% to 95% normal (non-anomalous) data.

3 Methodology

3.1 Real-World Anomalies

A robot and its track can experience various types of anomalies, which point to various types of failures within its mechanical parts. Knowledge of these can be used to simulate synthetic anomalies that are closer to real-world phenomena.

3.1.1 Rail Alignment

A robot's track is made up of various different sections, which are assembled together when the robot is mounted. If not done carefully, or over time, these rail segments can become misaligned, meaning that the robot needs to use more torque than usual to overcome these small hurdles. This type of anomaly can lead to increased wear on the track, so that it needs to be replaced more frequently, if not detected. This anomaly can be detected within the data as small bumps in the signal. It also usually occurs in a symmetrical manner, since the robot has to go past the misaligned section twice (back and forth).

3.1.2 Rail Degradation

Over time, sections of the rails can degrade, with scrapes and and bumps in the metal. This can result in the robot requiring additional torque to overcome these hurdles, which shows in the data as spikes of torque values.

3.1.3 Worn Bearings

Another part that is affected by wear and tear over time are the bearings in the tracks wheels. When this happens, there will be more friction in the wheels, and thus the robot will need to use more torque on average to cross the track. This manifests in the data as exaggerated torque values, compared to other robots. It can also get worse over time, so the torque values will get more extreme the longer this problem is left untreated.

3.1.4 Gearbox Problems

Over time, the gearbox can wear down as well, which affects the turning capabilities of the robot, meaning that it will take longer to turn to go backwards. This can be observed in data as an increase in distance between the middle torque value peaks. It is however not very relevant for this analysis, since the gearbox has separate ways of being tested, independently from the torque values

3.2 Synthetic Anomaly Simulation

To generate synthetic anomalies, we consulted robot maintenance experts which provided us with examples of different types of anomalies. Under Appendix A some examples are provided of how anomalies manifest in natural datasets and how these can be spotted in the torque signal of the robots. Using this expertise, a simulation strategy for synthetic anomalies was devised.

Synthetic anomalies are generated based on how real-world anomalies are represented in the `torqueactual` column of the robot execution dataset.

3.2.1 Point Anomaly

Point anomalies represent a sudden spike in the torque time-series. They can be caused by small obstructions in the track that force the robot to increase its torque to overcome them. Synthetic anomalies are created by modifying a random point in the time-series, adding a random value. This random value is generated by sampling from a Gaussian distribution with $\mu = 0.0$ and a parameterized σ.

3.2.2 Gaussian Noise Anomaly

Gaussian noise anomalies represent a more disturbed, noisy signal with higher frequency and more abrupt amplitudes. Synthetic anomalies are created by modifying each point in the time-series by adding random Gaussian noise with $\mu = 0.0$ and a parameterized σ. These anomalies can be utilised to simulate worn down bell bearings, which would look similar in data.

3.2.3 Shift Anomaly

Shift anomalies increase or decrease the amplitude of a range of points in the time-series. The method for generating shift anomalies takes three parameters. A start point, the index of the initial point of the shift, a length that is the number of point to be shift from start and the strength. For the range of points affected by the shift the standard deviation of those point is computed and a random value is sampled from a Gaussian Distribution with $\mu = 0.0$ and σ equal to the standard deviation of the values ranges multiplied by the strength. The sampled value is then added to all values in the range. The goal of this logic is to add shift translations that follow the nature of the original values in the shift range. This can effectively simulate wear of a particular part of the track.

3.2.4 Sinusoidal Anomalies

Sinusoidal anomalies modify a range of values in the time-series to follow a sinusoidal oscillator. This introduces the sinusoidal frequency to the signal without altering the existing frequencies for the affected values in the range. It requires three parameters: the length (number of values affected by the anomaly), the amplitude (of the sinusoid), and whether the anomaly is mirrored. Since the sequence of torque values is somewhat symmetrical, the anomaly can be present on both sides of the time-series from the symmetry point. The starting point of the anomaly is selected randomly. The idea behind this anomaly type is to simulate misaligned rails in the robot's track, as those manifest in a similar way, and are usually symmetric since the robot goes back and forth.

3.3 Data Treatment and Signal Processing

To prepare the data for the models, we carried out several processes.

3.3.1 Homogenization

Ensuring the sequences were similar was crucial for model training. Our data depicted similar movements of the track, but some differed in direction. Therefore, we changed the reference system for the signals so that all of them had the same shape and started the movement in position 0 as shown in Figure 2. During this process, we discarded a few sequences whose shapes were completely different from the others.

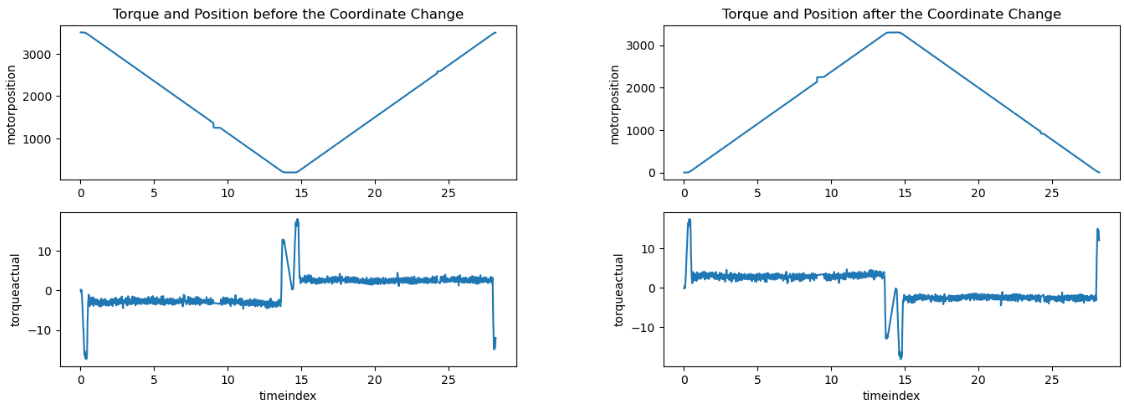


Figure 2: Example of coordinate transformation for standardizing the dataset.

During data processing, we discovered that the dataset contained duplicates differing only in sequence ID, while the values remained the same. Initially, the robot with the most data had 304 distinct sequences, but after removing duplicates, we were left with only 23. This situation necessitated resampling the sequences before training the HMM models.

3.3.2 Interpolation

Some of the approaches required the use of data in the frequency domain. To perform spectrum decomposition effectively, a constant sampling rate is recommended. Although our signals had the same average sampling rate, it did not remain constant throughout the sequence. Therefore, we applied linear interpolation to the signals, standardizing the sampling rate to the average value of 100 samples per second. An example of linear interpolation is shown in Figure 3.

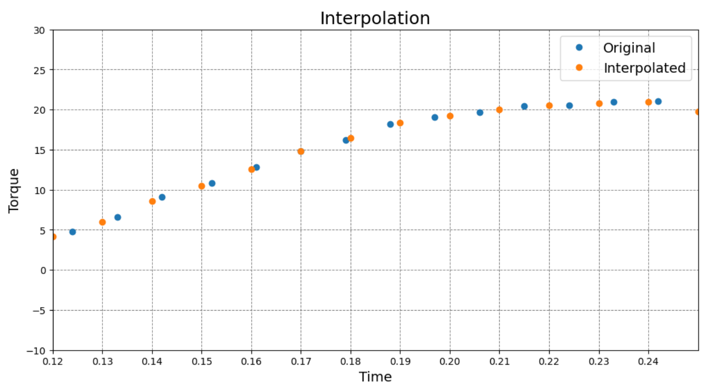


Figure 3: Example of an interpolation transformation

3.3.3 Spectrum Decomposition

Starting from the interpolated sequences, it is possible to transform the time series into the frequency domain using the Fast Fourier Transform (FFT):

$$X_\omega = \sum_{n=0}^{N-1} x_{t_n} e^{\frac{-i2\pi\omega t_n}{N}} \qquad t = t_0, ..., t_{N-1} \tag{1}$$

This algorithm computes the Fast Fourier Transform (FFT) of a sequence $\{x_t, t_j\}$ where $n \in \{0, n-1\}$, and transforms it into a sequence X_ω, ω_j. In our specific case, the initial values consist of the torque and the time index, while the resulting values include frequencies and the amplitude of these frequencies that constitute the torque signal. Figure 4 portrays an example of this transformation. The sampling rate of the interpolated sequences is 100 samples per second, so the maximum attainable frequency (Nyquist frequency) is 50 Hz.

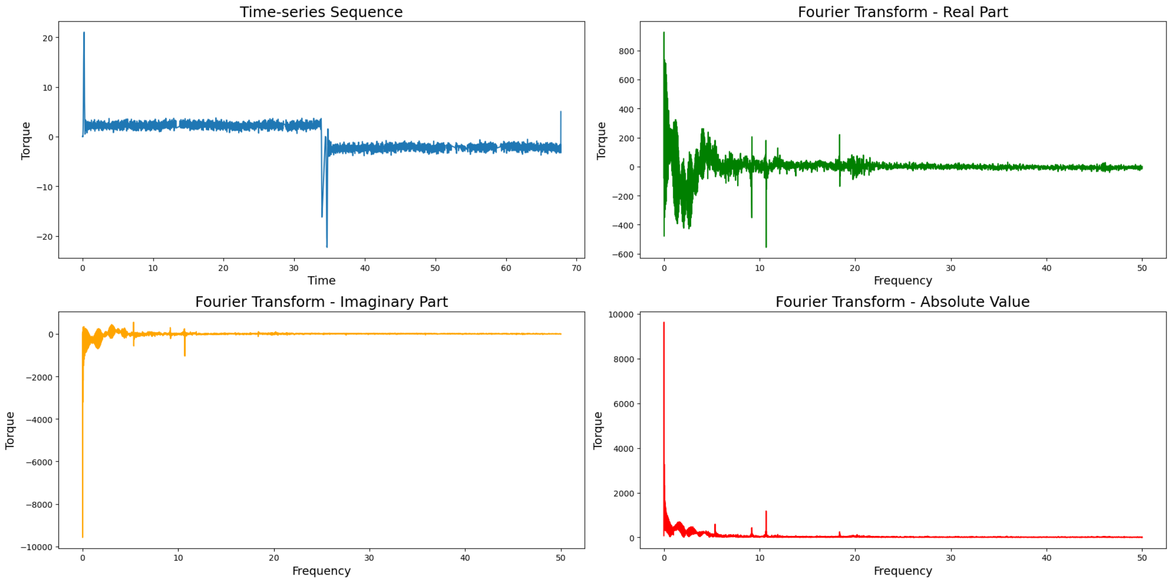


Figure 4: Fourier transform of torque signal.

We may also define the inverse transformation (Inverse FFT), that allows us to recover the original signal starting from the frequencies and amplitudes X_ω, ω_j:

$$x_t = \frac{1}{N} \sum_{k=0}^{K-1} X_{\omega_k} e^{\frac{i2\pi\omega_k t}{K}} \qquad \omega = \omega_0, ..., \omega_{K-1} \tag{2}$$

By combining these two algorithms, we can decompose the signal into components corresponding to distinct spectra. This transformation aims to extract the most relevant frequencies from our signal. The selected spectra will then be used to train an HMM for each one. Finally, the results from the different models will be integrated using an SVM.

3.4 Machine Learning

3.4.1 Local Outlier Factor

Local Outlier Factor (LOF) is a density-based clustering method that identifies points in a sequence which significantly deviate from the density of its neighbours. This method finds outliers by calculating an outlier score for each point. This is achieved by calculating the inverse average distance of the point to its k-nearest neighbours. Then the average of the ratios of the point and its neighbours is calculated, which represents the outlier score. A score that is significantly higher than 1 indicates that the point is an outlier. The motivation behind trying this method is due to the absence for a need of labeled data, which is the case with our dataset. Additionally LOF is a versatile method that can be used for a multitude of tasks, its main advantage though is its ability to find subtle local outliers as opposed to global differences thanks to the density based scoring. For a more in-depth explanation of LOF see [18].

The python implementation of LOF allows for two overarching types of tasks: outlier detection and novelty detection [19]. While outlier detection refers to finding outliers in an unsupervised dataset, novelty detection assumes the training data does not contain outliers and rather checks new unseen data for unexpected datapoints, called "novelties" [19]. This way we can test whether LOF is capable of detecting injected anomalies as novelties. We will use the four standard model performance metrics such as accuracy, precision, recall, and f1 score to determine how LOF treats synthetic anomalies when used as a novelty detection tool.

It is also of interest to potentially discover natural anomalies in the dataset, however validating such results poses a significant challenge.

3.4.2 Isolation Forest

Isolation Forest (iForest) is an unsupervised machine learning algorithm used for anomaly detection. Unlike traditional clustering methods, Isolation Forest works by isolating observations through random partitioning. This technique is particularly effective for detecting anomalies because anomalies are few and different, making them easier to isolate.

The motivation behind using Isolation Forest for anomaly detection comes from its unique approach to identifying anomalies. Traditional methods often rely on distance or density measures, which can be computationally expensive and less effective in high-dimensional spaces. Isolation Forest, on the other hand, operates in linear time, making it scalable to large datasets. Moreover, it is capable of handling high-dimensional data and capturing complex data distributions.

Isolation Forest constructs multiple decision trees (isolation trees) by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. Firstly, each data point is recursively partitioned until it is isolated from the rest of the data. Secondly, the number of splits required to isolate a data point is recorded. Anomalies, being few and different, tend to have shorter paths because they are easier to isolate. Thirdly, the anomaly score is calculated based on the average path length across all trees. A lower average path length indicates a higher likelihood of being an anomaly.

In this analysis, we implemented Isolation Forest to detect anomalies in a synthetic dataset with various types of anomalies. A random sequence was generated at the beginning, and synthetic anomalies were introduced to create point anomalies. For feature scaling, the data was normalized using `StandardScaler` to ensure that the model performs well irrespective of the data scale. After that, an Isolation Forest model was trained on the normalized data, and the model predicted anomalies in the data, which were then visualized.

3.5 Statistical Methods

3.5.1 Baseline Statistical Model (Z-scores)

The goal of this model is to introduce a baseline model such that we have a model to compare with our more advanced implementations. How this model works is that it produces a z-score for each time-index of the given any time-series. In this context, the time-series means all set of time-series of specific robot-id, series name, and parameter name. So, given these specific parameters we gather each existing time-series and perform a set of operations on each time-index via transpose. A simple representation of this can be seen in Figure 5.

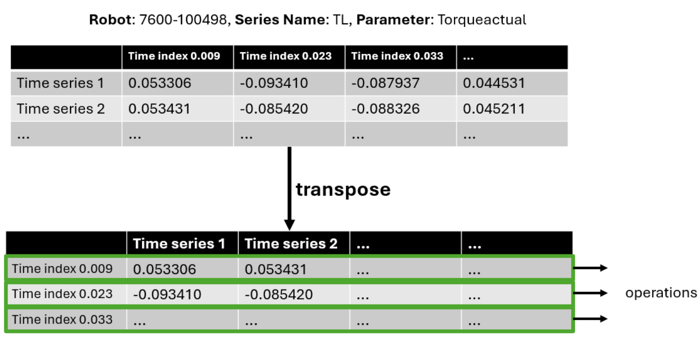


Figure 5: An example representation of the process followed given the parameters.

In order to perform outlier detection, it is required to make sure that each time-index data satisfies the normality assumption. This is because if the normality assumption for the data being tested is not valid, then a determined outlier may in fact be due to the non-normality of the data rather than the presence of an outlier [20]. Looking into our plots of several time-indexes we can see that all of them follows a close-to-normal distribution 6. Hence, because most of them does not follow a perfect normal distribution, this case is most observable close to start and end time indexes, it means that the performance of the baseline models will be slightly affected. The other requirement is to identify if we are trying to identify a single outlier or multiple outliers. Since the model will work by taking a given (anomaly) time-series and going through each time index data, checking whether it is an outlier or not, means that we are only testing for a single outlier in each time index. To conclude, with our shown normality distributions we are allowed to proceed with the generation of z-scores. The z-scores represents how many standard deviations the given data is from the mean and uses the formula given below [20].

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

In addition, for the possible case of limited number of sample size we also create modified z-scores. As the generated normal z-scores can be misleading (particularly for small sample sizes) due to the fact that the maximum Z-score is at most

$$\frac{(n-1)}{\sqrt{n}} \tag{4}$$

[20]. In order to handle this the equation below is used to create the modified z-scores.

$$M_i = \frac{0.6745 * (x_i - \mu)}{MAD} \tag{5}$$

Moreover, because we have also concluded that our goal is to identify single outliers we can additionally use the Grubbs Statistical Hypothesis Test. Which is a test used for detecting a single outlier in a data set that follows an approximately normal distribution [20].

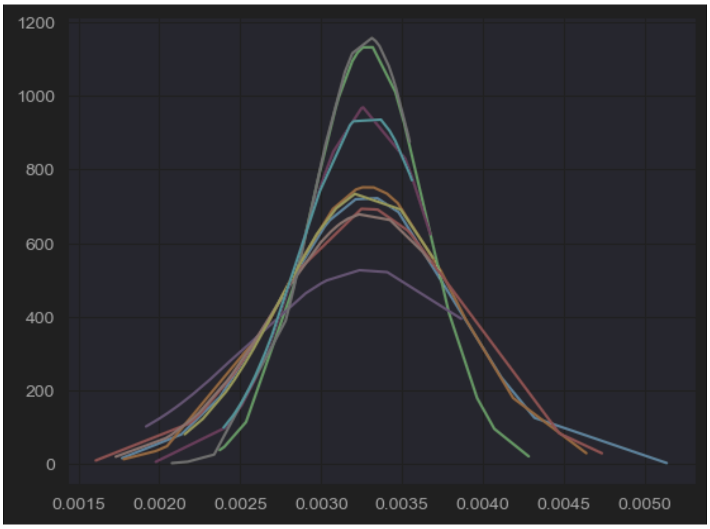


Figure 6: A representation of several close-to-normal distributions of different time indexes.

3.5.2 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models that represent systems with observable events influenced by hidden states. These models are particularly useful for analyzing sequences where the underlying causes of the observed events are not directly visible. HMMs are highly relevant in the context of anomaly detection for industrial robots, where identifying deviations from normal operational patterns is crucial.

An HMM consists of:

- A set of N hidden states $Q = q_1, q_2, \ldots, q_N$.
- A transition probability matrix A, where a_{ij} represents the probability of transitioning from state i to state j.

- An observation likelihood matrix B, where $b_j(o_t)$ gives the probability of observing o_t from state j.
- An initial state distribution π, where π_i is the probability of starting in state i.

In anomaly detection for industrial robots, HMMs can model the sequence of operations and detect deviations that indicate potential failures. The Markov assumption, where the probability of a state depends only on the previous state, and the output independence assumption, where the probability of an observation depends only on the state that produced it, make HMMs well-suited for this task. By training an HMM on normal operational data, the model can learn the typical state transitions and observations. Once trained, the HMM can evaluate new sequences of observations to identify anomalies, which appear as unlikely state transitions or unusual observations.

To enhance the model's accuracy and robustness, Mixture Gaussian HMMs has been used. A Mixture Gaussian HMM allows each hidden state to be associated with a mixture of Gaussian distributions rather than a single Gaussian distribution. This approach can capture more complex data patterns and variations, making it particularly effective for modeling the diverse operational states of industrial robots.

For example, in a Mixture Gaussian HMM, the torque values of a robot arm during normal operations can be modeled using a mixture of Gaussian distributions for each state. This allows the model to account for variations within each operational state more accurately. When the model later detects a sequence of torque values that significantly deviates from these learned patterns, it can flag this as a potential anomaly, prompting further inspection or maintenance.

The Fordward-Backward Algorithm

Training Hidden Markov Models (HMMs) involves learning the transition probabilities (A matrix) and the emission probabilities (B matrix) from a sequence of observations. The standard approach for this is the Forward-Backward Algorithm, also known as the Baum-Welch Algorithm [21], which is a specific case of the Expectation-Maximization (EM) algorithm [22].

HMM Training: Forward-Backward Algorithm

- **Initialization:** Start with an initial estimate of the transition and emission probabilities.
- **E-Step (Expectation Step):** Compute the forward probabilities (α) and backward probabilities (β).
  - The forward probability, $\alpha_t(i)$, is the probability of being in state i at time t given the observed sequence up to time t.
  - The backward probability, $\beta_t(i)$, is the probability of the remaining observations from time $t+1$ to the end, given state i at time t.
- **M-Step (Maximization Step):** Use the forward and backward probabilities to compute the expected counts of transitions and emissions.
  - Re-estimate the A and B matrices based on these expected counts.

The Forward Algorithm

The goal of the forward algorithm is to compute the likelihood of a particular observation sequence given an HMM with particular A and B matrices. Formally, $P(O|\lambda)$ where $\lambda = (A, B)$.

HMM Likelihood Estimation: Forward Algorithm

- **Initialization:** Start with an initial transition, emission and prior probabilities.
- **Recursion:** Estimates the probability of being at state i at time t given the observation sequence up to t.

$$\alpha_t(i) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq N, 1 < t \leq T \tag{6}$$

- **Return:**

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{7}$$

3.6 Evaluation

The evaluation serves as a comparative analysis tool to assess the performance of various models in detecting anomalies within robotic operational data. Due to the absence of anomalies in the provided dataset, synthetic anomalies are introduced using parameterized synthetic anomaly generators. These parameters should also influence the evaluation process, as less intrusive anomalies are inherently more difficult to detect and thus require more sensitive and advanced anomaly detection methods.

The proposed method begins with synthetic anomaly generation using a semi-randomized process that leverages specific parameters to introduce varying levels of anomalies. Following this, a predictive step is undertaken where models generate estimates on both non-anomalous and synthetically anomalous data. These models are evaluated using a confusion matrix [23]. In Table 3, *True* denotes anomalous observation sequences, whereas *False* denotes non-anomalous observation sequences.

Table 3: Confusion Matrix for Binary Classification

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FN)	True Positive (TP)

Given the critical nature of anomaly detection in industrial robots, the evaluation should prioritize models with higher accuracy in the False Negative Rate (FNR) (8). Missing a True Positive (i.e., failing to detect an anomaly) can have catastrophic economic consequences, as it may lead to production halts due to robot failure. Consequently, the proposed model should be biased towards minimizing the FNR.

$$FNR = \frac{FP}{FP + TN} \tag{8}$$

It is important to note that this strong bias towards FNR is based on the assumption that subsequent steps to verify the presence of anomalies incur low time costs. For instance, visual inspection of robot tracks or the utilization of auxiliary visualization tools to examine operational sensory data should be employed to confirm the presence of anomalies.

The False Negative Rate (FNR) performance is evaluated relative to the level of disturbance in the synthetic anomaly data. The evaluation favors models that are capable of detecting anomalies generated with parameters that create small distortions, leading to more sensitive models. This decision is motivated by a preventive approach. We believe that it is better to start with a model that may initially have high False Positive Rate, as it can be fine-tuned to reduce its sensitivity over time.

4 Experiments And Results

4.1 Synthetic Anomaly

Anomaly simulation consists of two main modules. The first module contains all the methods necessary to generate each type of anomaly as defined in subsection 3.2. This module is encapsulated within the delivered code in a class named `NoiseMachine`. The second module implements methods, one for each type of anomaly, to generate anomalies iteratively and pseudo-randomly. The goal of this module is to facilitate the exploration of the effects that different parameter values have on synthetic anomalies, as well as to assist in creating synthetic anomaly time-series for model evaluation purposes. This module is encapsulated in the code within a class named `NoiseFactory`. All methods are implemented as generators [24], preventing memory usage errors and allowing for online inference.

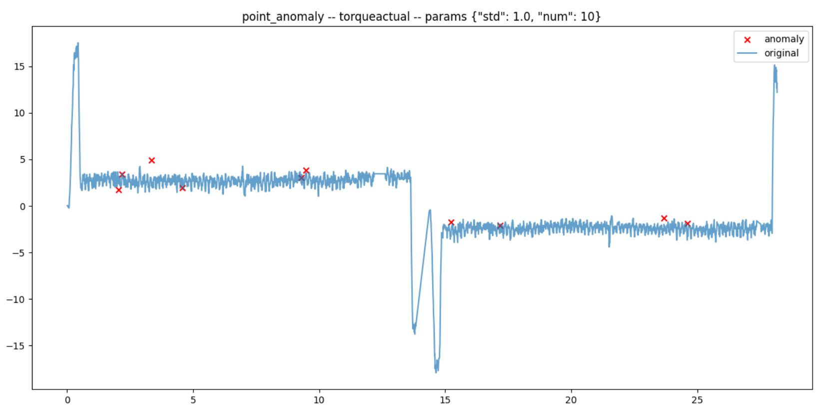
Synthetic Anomaly – `NoiseMachine`
All synthetic anomaly methods are designed following four principles:

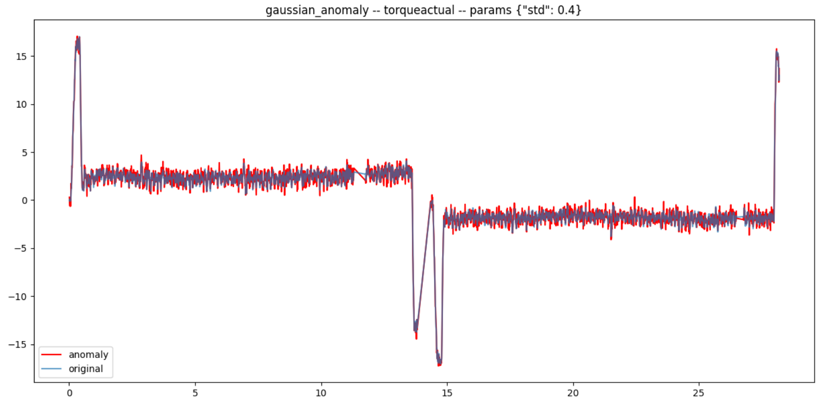
1. **Parameterized**: All methods should include parameters to module the anomaly intensity.

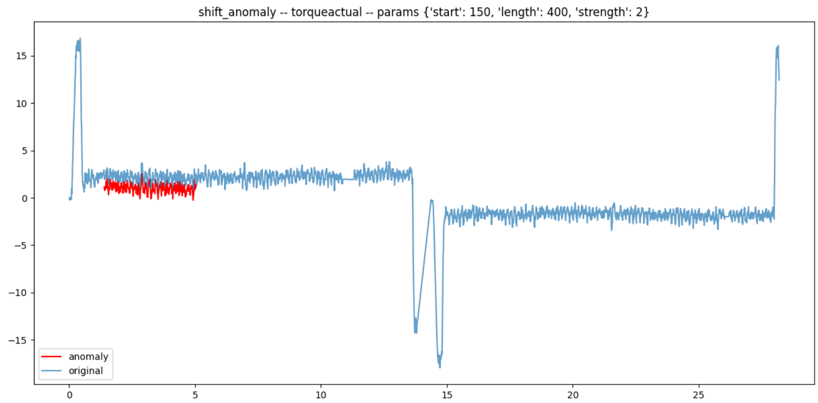
2. **Parameters Independence**: If a method accepts more than one parameter the effect of each parameter is independent to the others. Hence no interaction between parameters.

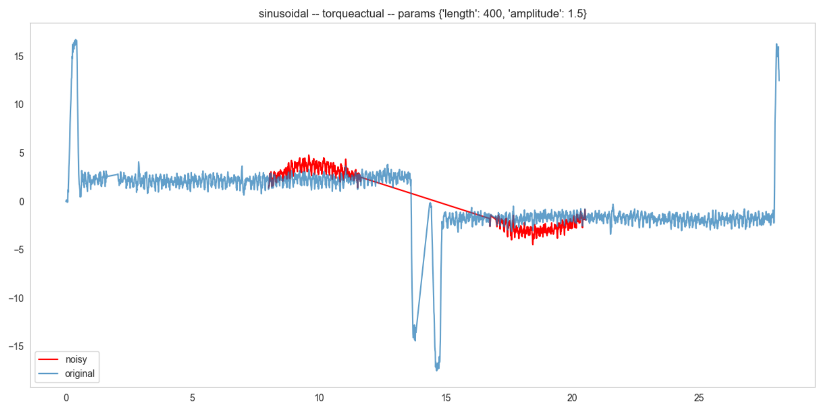
3. **Positive Effect**: As parameters tend to positive infinity the distortions on the generated anomalies should increase. All parameters should take values between 0 to $+\inf$.

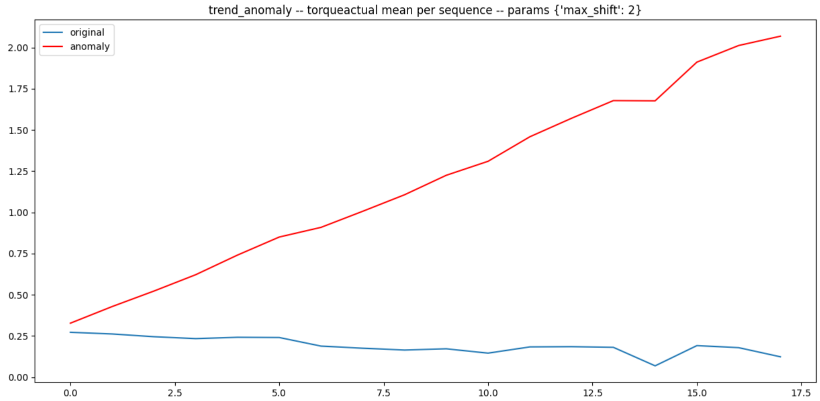
4. **Not Deterministic**: All methods should include some randomization factor.

<table>
<tr><th>Point Anomaly</th><th>Parameters</th></tr>
<tr><td>
</td><td>Standard Deviation
Number of Points</td></tr>
</table>

Gaussian Noise	Parameters
	Standard Deviation

Shift Anomaly	Parameters
	Start Length Strength

Sinusoidal Anomaly	Parameters
	Length Amplitude

Trend Anomaly	Parameters
	Maximum Shift

Synthetic Anomaly – `NoiseFactory`

In this section, the parameter values chosen to generate a set of anomalies are presented. With the current experiment setting, the `NoiseFactory` generates a total of 185 synthetic anomaly sequences from one non-anomalous sequence.

The values chosen for most parameters follow a logarithmic scale, instead of a more naive approach of selecting equidistant values. The motivation behind this design choice is the Positive Effect principle 3. We want to explore lower values in the parameter space to create small-level anomalies. This experimental choice is also conditioned by our evaluation criterion, which biases towards sensitive models. Hence, we aim to generate small-level anomalies.

Below is the list of selected parameters for each anomaly type:

- **Point Anomaly**:
  - Standard Deviation: 5 values ranging from 1 to 10, with log-scale distances.
  - Number of Points: 5 points from 5 to 10, equidistant.
- **Gaussian Noise**:
  - Standard Deviation: 5 values ranging from 0.01 to 0.1, with log-scale distances.
- **Shift Anomaly**:
  - Start Point: 5 values random sampled from Uniform Distribution between 200 to number of points in sequence.
  - Point Length of Shift: 5 values ranging from 1500 to 15000, with log-scale distances.
  - Strengths: 5 values ranging from 1 to 10, with log-scale distances
- **Trend Anomaly**:
  - Maximum Shift: 5 values ranging from 0.1 to 10, with log-scale distances.
- **Sinusoidal Anomaly**:
  - Point Length: 5 values ranging from 150 to half of the total number of points in the sequence, with log-scale distances.
  - Amplitudes: 5 values ranging from 1 to 5, with log-scale distances.

4.2 Signal Processing

4.2.1 Spectrum Decomposition

In order to explore the effect that the different types of anomalies have on the frequencies of the signal, we selected a random sequence of the interpolated dataset and added each type of the anomalies one at a time. Then, we computed the relative difference between the Fourier transforms of the anomalous and non-anomalous sequences following:

$$relative\ diff\ (a, b) = \frac{|a - b|}{|a| + |b|} \tag{9}$$

This way, we derived Figure 7, that portrays the effect that the different types of anomalies have in the frequency domain. It is clear from the graph that point anomalies and gaussian anomalies the higher the frequency the more they affect the signal. Conversely, the shift, trend and sinusoidal anomalies have the biggest impact in the low frequencies. This effect is notably more pronounced for the sinusoidal anomaly, for which the relative difference is zero for the middle and high frequencies.

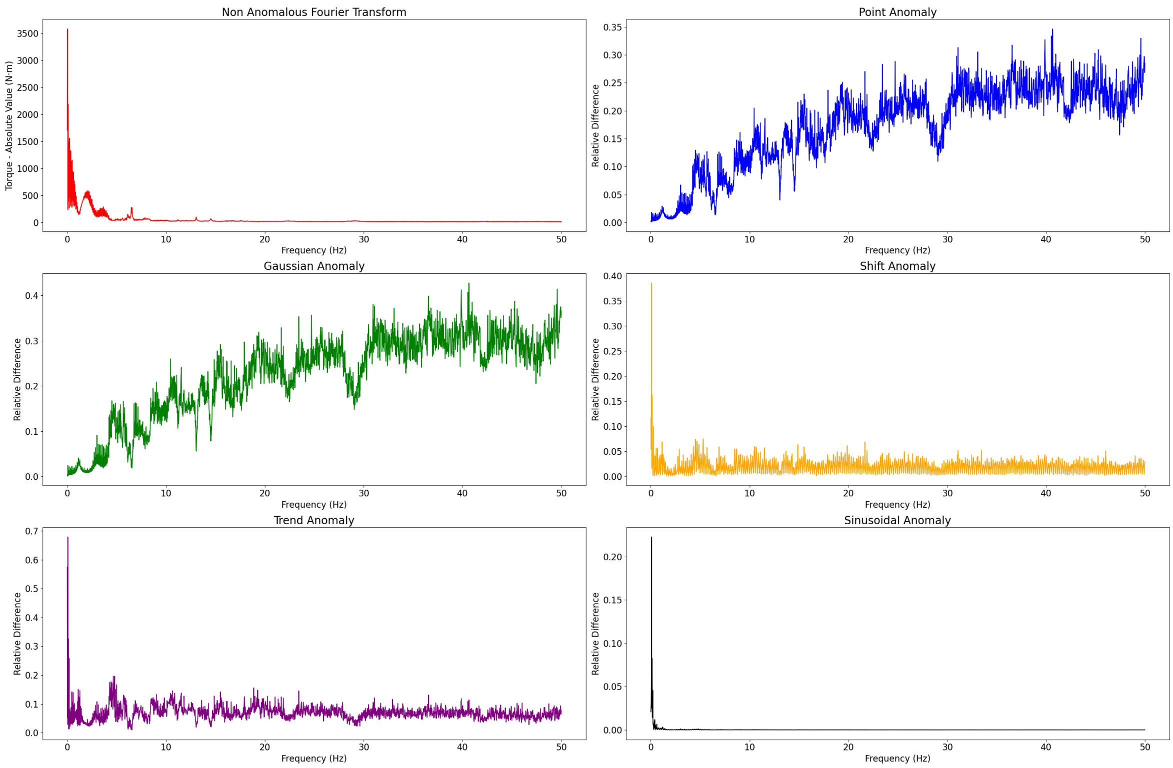


Figure 7: Average normalized difference between the Fourier transforms of the original signal and 5000 derived anomalous signals, categorized by different types of anomalies.

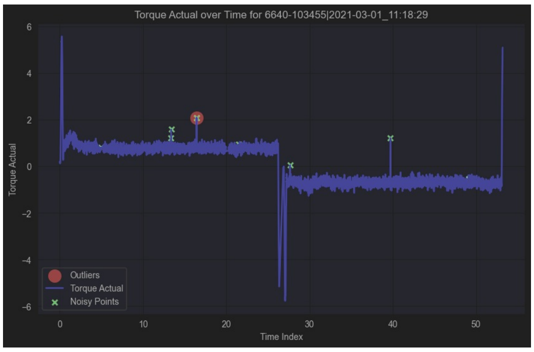
In order to prepare the signals for training the HMM, we used the FFT and the inverse FFT to decompose each signal in 25 different spectra as portrayed in Figure 23 under Appendix B. The y-axis in the graph is particularly relevant as it shows the contribution of each spectrum to the original signal.

The spectrum decomposition approach allows us to filter the most relevant spectra and use each to train an HMM. We selected the first six spectra as the most relevant for the HMM experiments because they significantly contributed to the signal and yielded the best results.

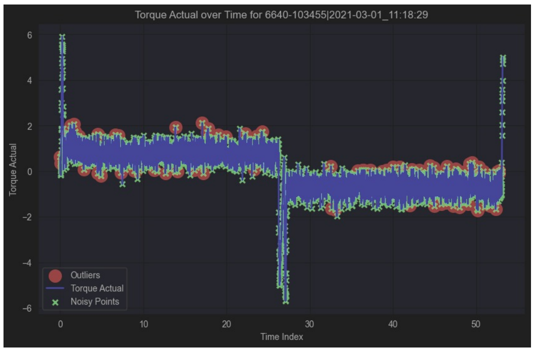
4.3 Machine Learning

4.3.1 Local Outlier Factor

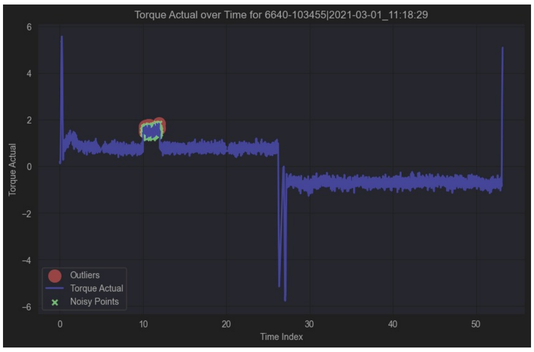
Local Outlier Factor was implemented using the `sklearn` library module `LocalOutlierFactor` in python. The model was trained on interpolated, normalized data coming from all the robots, specifically for the task of novelty detection. A random sequence was then selected for anomaly simulation. Figure 8 shows an example of how LOF detects anomalies (novelties).



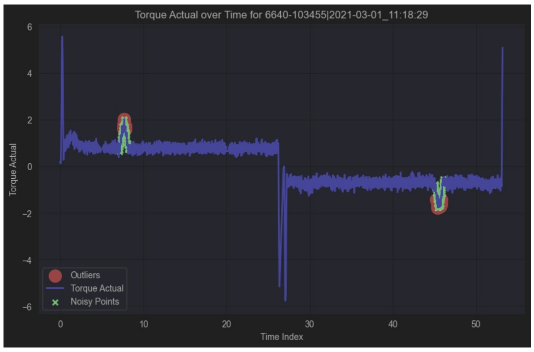
(a) Point Anomaly, n=10



(b) Gaussian Noise Anomaly, std=0.3



(c) Shift Anomaly, start=1000, length=200, strength=7.0



(d) Sinusoidal anomaly, amplitude = 1, length = 100

Figure 8: LOF novelty detection of artificial anomalies on a random sequence

From Figure 8 we can draw the following observations:

- **Point Anomaly:** as Figure 8a shows, LOF did not identify injected points as anomalous. In most cases, the algorithm is able to successfully identify a few points as anomalous, however the rest remain undetected.
- **Gaussian Noise:** as Figure 8b shows, while LOF did not identify all points as anomalous, there are significantly more outliers detected which is consistent with the type of anomaly generated.
- **Shift Anomaly** as Figure 8c shows, in this specific example, LOF successfully identifies the shift as an anomaly, however the algorithm's ability to do so is dependent on the strength of the shift, as less obvious shifts are generally not detected well.
- **Sinusoidal:** as Figure 8d shows, LOF successfully identifies the fluctuations as anomalous in this specific example, however, not all sequences show the same success, while most times the algorithm identifies at least a small part of the fluctuation as anomalous, big portions may remain undetected.

Additionally, as a sanity check, the algorithm was also tested on the sequence where the added noise was 0 (no noise). As expected, LOF did not identify novelties.

Lastly, LOF can also be used to find natural anomalies, by accessing the outlier scores of the training data. There is such an example of a potential natural anomaly identified in Figure 9. There is a small fluctuation in torque at the beginning of the sequence which could mean that there is a part of the track that is rarely used, thus it takes more torque to traverse it. This and the other anomaly around the middle of the sequence could be indicative of a gearbox issue. Nevertheless, due to a lack of a labeled dataset, the validity of this result remains unknown.

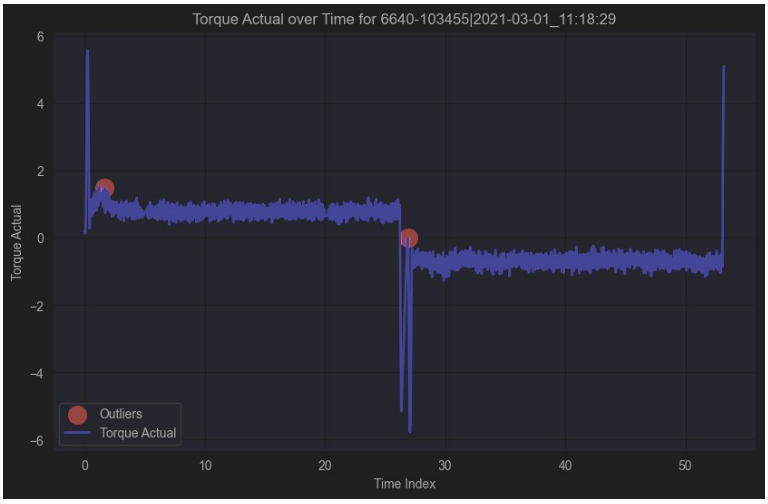


Figure 9: Potential Natural Anomaly found by LOF

4.3.2 Isolation Forest

Point Anomaly: The Isolation Forest model was able to detect some of the point anomalies but missed others. Point anomalies are sudden and distinct deviations in the data, and while Isolation Forest can detect some of these, it might not be consistent, especially if the anomalies are not significantly different from the rest of the data points. Because Isolation Forest relies on the data distribution and anomalies being far from the majority of the data points. If point anomalies are not significantly different in magnitude, the model might fail to isolate them as outliers.

Gaussian Noise Anomaly: The Isolation Forest method struggles significantly with Gaussian noise anomalies. The noise makes the data look more random, and the model struggles to differentiate between normal fluctuations and anomalies. Because Gaussian noise increases the overall variance in the data, and the model finds it challenging to create partitions that clearly separate anomalies from normal data when the noise is distributed across all data points.

Shift Anomaly: Shift anomalies, where a segment of data is suddenly shifted, are somewhat detected by the Isolation Forest. The model can detect the abrupt change, but the performance might vary depending on the shift magnitude and duration. Isolation Forest may not consistently detect shifts, especially if the shift is gradual or the segment is not sufficiently long. Therefore, the effectiveness of detection relies on the shift being substantial enough to stand out in the data distribution.

Trend Anomaly: Trend anomalies are poorly detected by the Isolation Forest. The model struggles to identify a gradually increasing or decreasing trend as an anomaly, as it primarily focuses on the distribution of individual points rather than trends over time. Isolation Forest is not designed to detect gradual changes or trends over time. The model's decision trees focus on isolating individual data points rather than understanding the sequence and progression of the data.

The Isolation Forest method has mixed results with different types of synthetic anomalies. It is more suited for detecting point anomalies and might struggle with noise, shifts, and trends. This is due to its reliance on partitioning the data space based on individual data points rather than understanding temporal patterns or continuous shifts.

4.4 Statistical Methods

4.4.1 Baseline Statistical Model

There are three versions of the implemented baseline model which are shortly explained below:

- Baseline Z-Score: Here we use the z-score formula applied on each time index of the given anomaly time series. Later, we apply the threshold rule such that if z-score > 3.5 we assume that the given data (on this specific time index) is an outlier, else not outlier.

- Baseline Modified Z-Score: Similar to previous method. Only difference being is that the threshold rule is instead applied on the modified z-scores.

- Baseline Grubbs' Test: Here we use the Grubbs test with parameter alpha $= 0.05$ to decide whether if the given time index data is an outlier or not.

For each of these variants of the baseline model, we will be looking into their accuracy, recall, and precision; on point anomaly, shift anomaly, and sinusoidal anomaly. The reason we skip gauss anomaly, and trend anomaly is because both of these anomalies are applied on the whole time series. Making it more suitable to use for observing probability (that the model returns) changes. Since our aim here is to perform binary classification these are not suitable here.

The parameters of this experiment are "robot-name = 7600-100498, series-name = TL, param-name = torqueactual". Although this experiment can be done on any parameters, these parameters were deemed the most suitable for this project. For the experiment of Baseline Models, the `NoiseFactory` class was used with parameter num_sequences = 1. The reason being is that even though the run-time of Baseline Z-score model is fast, the other baseline models were slow taking too much time. Thus, because of the run-time limitations the experiment was only done on a single anomaly time-series with all the different anomaly parameters that is already included in class `NoiseFactory`. With the final results averaged, all the results can be seen in Tables 4, 5, and 6.

Table 4: Results of Baseline Z-Score.

Noise Type	**Accuracy**	**Precision**	**Recall**
Point Anomaly	0.983010	0.010769	0.205651
Shift Anomaly	0.909305	0.181345	0.069136
Sinusoidal Anomaly	0.887987	0.775078	0.521065

Table 5: Results of Baseline Modified Z-Score.

Noise Type	**Accuracy**	**Precision**	**Recall**
Point Anomaly	0.903750	0.001502	0.158063
Shift Anomaly	0.843063	0.113779	0.154641
Sinusoidal Anomaly	0.787681	0.399440	0.332964

Table 6: Results of Baseline Grubbs' Test.

Noise Type	**Accuracy**	**Precision**	**Recall**
Point Anomaly	0.000898	0.000898	1.000000
Shift Anomaly	0.074739	0.074739	1.000000
Sinusoidal Anomaly	0.206544	0.206544	1.000000

4.4.2 Hidden Markov Model

This subsection begins by outlining the common aspects of the different experiments. The model used is a variation of the vanilla HMM algorithm; specifically, we use a Mixture of Gaussian HMM (MG-HMM) [25]. The MG-HMM models the emission probabilities with a mixture of multiple Gaussian distributions rather than a single Gaussian distribution. This approach enhances the model's ability to represent complex and multi-modal distributions, making it better suited for complex data. Given the intricate nature of the problem, this seems like the most sensible choice.

The number of hidden states is fixed at 50, determined through empirical testing. We observed that a smaller number of hidden states negatively affected model convergence, while a larger number of hidden states increased training time without significant performance returns.

All models are trained with the time-series `torqueactual` present in the provided datasets. Torque sensory data encodes a great amount of information regarding the robot state, making it suitable for evaluating operational risks. It is important to mention that the most successful models are trained with enhanced torque signals through pre-processing prior to training.

A Naive Approach
In our initial attempt, we directly utilized the raw `torqueactual` signals as input for the Hidden Markov Model (HMM), training it with the Forward-Backward algorithm. However, this approach did not yield satisfactory results, as the model struggled to converge. To address this, we increased the number of hidden states, hoping to capture more complexity. Unfortunately, this adjustment resulted in excessively long training times without achieving convergence.

Recognizing that the model was overwhelmed by the data's complexity, we opted to simplify the input data. As outlined in earlier sections, we implemented a spectrum decomposition preprocessing step to achieve this reduction in complexity.

HMM & Spectrum Decomposition
To simplify the data complexity, we decomposed the original torque signal into its constituent frequencies. We then trained a HMM for each spectral decomposition of the torque signal, achieving convergence.

We evaluated the results by calculating the posterior probabilities of the torque sequences. We observed that across all spectra, the log-probabilities exhibited a bimodal distribution, Fig 10. This pattern can be attributed to the type of robot model, where each model represents a mode in the distribution. This effect is caused by one of the robot models having twice the number of data points, Fig 11.

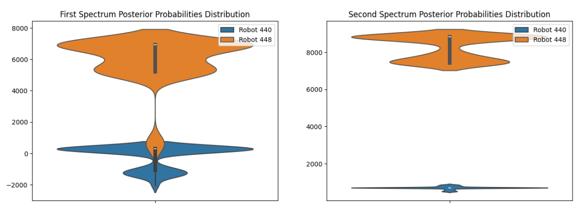


Figure 10: Posterior log-probability of by robot type for the first two spectrum

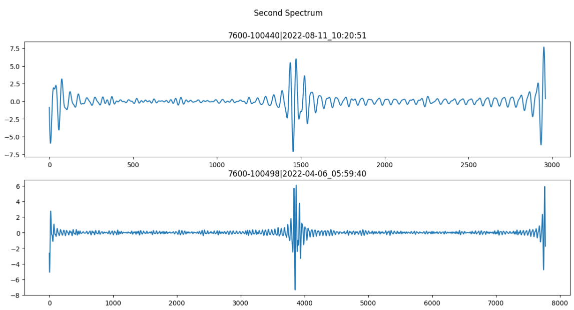


Figure 11: Torque second spectrum decomposition by robot

We concluded that training models separately at both the spectrum and robot levels would be more effective, primarily to enhance model sensitivity.

Final Result
In parallel with our efforts to train the Hidden Markov Model (HMM), we discovered that the provided data contained numerous repeated sequences. Originally, we had about 350 unique sequences per robot type, but this number was reduced to just 18, posing a significant challenge. The complexity of the torque signal, compounded by the limited sample size in our dataset, made training any model difficult. To address this issue, we employed random sampling of sequences from the original dataset and trained the model with a re-sampled dataset. This approach allowed us to increase the number of usable sequences and reduce the risk of over-fitting.

As discussed in section 4.2.1 not all spectra encode the same amount of information or contribute equally to signal reconstruction. Therefore, we opted to train the model specifically for the first six spectra, which encode the most information. All six spectra-models achieved convergence during the training phase. Figure 12 illustrates the posterior log-probabilities for each spectrum using non-anomalous sequences.

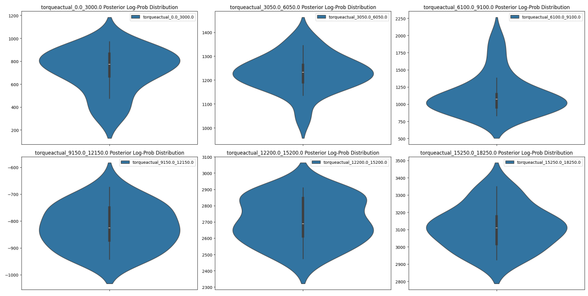


Figure 12: Posterior log-probability non-anomalous data.

All spectra generally show positive log-probabilities, suggesting that the models have successfully captured the relationship between observations and hidden states. However, there is an exception: the fourth frequency range consistently displays negative values. We hypothesized that this particular range primarily captures noise from the torque signal, which complicates the learning process. Despite numerous experiments, this remains the most plausible explanation.

4.5 Evaluation

4.5.1 Local Outlier Factor

The overall performance of LOF for novelty detection of synthetic anomalies was determined through experiment. For each type of anomaly, a range of parameters were tested for a couple of outlier score thresholds, each time picking 50 random sequences. For the evaluation, the length of the sequence along with the number of anomalies, number of detected outliers, and true positives were measured and then averaged per each type of model tested. The results can be seen in Table 7 under Appendix C. Additionally, the results are visualized in Figures 24-27. Note that trend anomalies were not tested due to the fact that LOF is a local algorithm and it is not expected to handle changes in sequences over time.

We can observe in Figure 24 that LOF performs poorly for Gaussian anomalies, however, it must be considered that Gaussian noise is applied to the entire data, and the model is tuned to only account for a small percentage of anomalies. Considering that, LOF still detects a significant amount of anomalies regardless of the standard deviation parameter which should be indicative of need for maintenance in a manufacturing setting.

We see that the accuracy is very high for the rest of the anomaly types which is a good indicator that the identified novelties are anomalies regardless of how strong or weak their influence is on the signal. Judging by the recall metrics, the model does not find all the anomalous points, however in a practical sense finding even a small proportion of the true positives could be enough to prompt a maintenance check. The model is able to achieve good precision for sinusoidal and shift anomalies given that the strength of the shift is higher than 1. This is a very promising result – in a manufacturing environment, a minimal amount of false positives would be preferred. Further hyperparameter tuning could improve the model and make it more suited for the needs of the business.

4.5.2 Baseline Statistical Model

From our final Tables 4, 5, and 6: We will first look into Table 4.

In this table we realize that because in our generated anomaly time series there are, in general, few anomalies added and our Baseline Z-Score model is prone to assign the data as non-outlier, it gives a high accuracy. However, because the Baseline Z-Score model is prone to making non-outlier classification it is better to investigate into Precision and Recall. It is interesting to see that Recall is higher in point anomaly, whereas Precision is higher in other types of anomalies. It is important to remember here that by looking into Z-Scores we only look into how far the given anomaly time-index data is from the original data. Meaning, when we get frequency related anomaly data points, which still resides inside/close to original data it will perform bad as the Z-Score will be low. Which is what we generally realise here, as long as the amplitude/strength parameters of the anomaly generators are low, the Z-Score related algorithms will have trouble identifying them. With this reasoning, it makes sense that the Baseline Z-Score model had the best performance in sinusoidal anomalies. As the greatest distance from the original data can be introduced in the sinusoidal anomaly because of the sinusoidal shape.

The similar reasoning can be applied for the second Table 5 too. However, interestingly, when we would have expected a higher score compared to the Baseline Z-Score table, it in fact has lower scores. This might be due to many reasons where the biggest one could be because of the nature of the data. Since the modified z-score heavily relies on median identification, when the given time-series data contains too many repeated values, because of the formula, a division by 0 occurs (i.e., MAD becomes 0 as for calculation of MAD the list needs to be subtracted from the median value and then need to take the median of the remaining list again; if the median value repeats too much the median of the remaining list becomes 0). To fix this a removal of parts of the data had to be made, making it losing significant information.

Finally, for last Table 6, we can say that the Grubbs' Test has either a very high inclination for classifying data as an outlier, or it is not suitable to use in this case.

To conclude, it is important to identify the types of anomalies we may come across with. If the anomalies are amplitude-based it is a good idea to use the Baseline Z-Score model, if it is frequency-based then a more advanced model is suitable such as HMM. We also realize that the nature of the data is crucial if we want to do more advanced Z-Score related operations such as modified Z-Scores.

4.5.3 HMM

The evaluation of the HMM model is structured as detailed in subsection 3.6. We compute the log-probabilities for the non-anomalous sequences across each spectrum. Subsequently, using the `NoiseFactory` module, we generate anomalous signals for each type of synthetic anomaly. In a manner similar to the non-anomalous data, we then calculate the inferred posterior log-probabilities for each anomalous sequence across the spectra. From this preliminary results a set of hypothesis/questions emerged.

- Do all spectrum-models consistently capture the presence of anomalies, as indicated by uniformly lower log-probabilities?
- How do parameters affect the inference results?
- Is there a clear separation between the log-probabilities of non-anomalous and anomalous sequences?

Figure 10 demonstrates the impact of point anomalies on the log-probabilities of non-anomalous signals. In the plot the x-axis represents the log-probability of an undistorted signal and the y-axis represents the log-probability when the signal is distorted with anomalies. A dotted-dashed line with a slope of 1 is drawn for reference. Points falling below this line indicate a decrease in log-probability due to the presence of anomalies, suggesting that the anomalies were significant. Points on the line suggest that the anomalies had no discernible impact on the sequence's perceived likelihood. Conversely, points above the line indicate an increase in log-probability, suggesting that the anomalies enhanced the perceived normality of the signal. This latter scenario typically occurs with lower levels of distortion, where the introduced distortion still aligns with the typical distribution of normal signals.

The color gradient from blue to red on the plot denotes the level of distortion introduced by the anomalies, with blue representing lower and red indicating higher distortion levels. Additional plots categorizing different types of anomalies are provided in Appendix A.

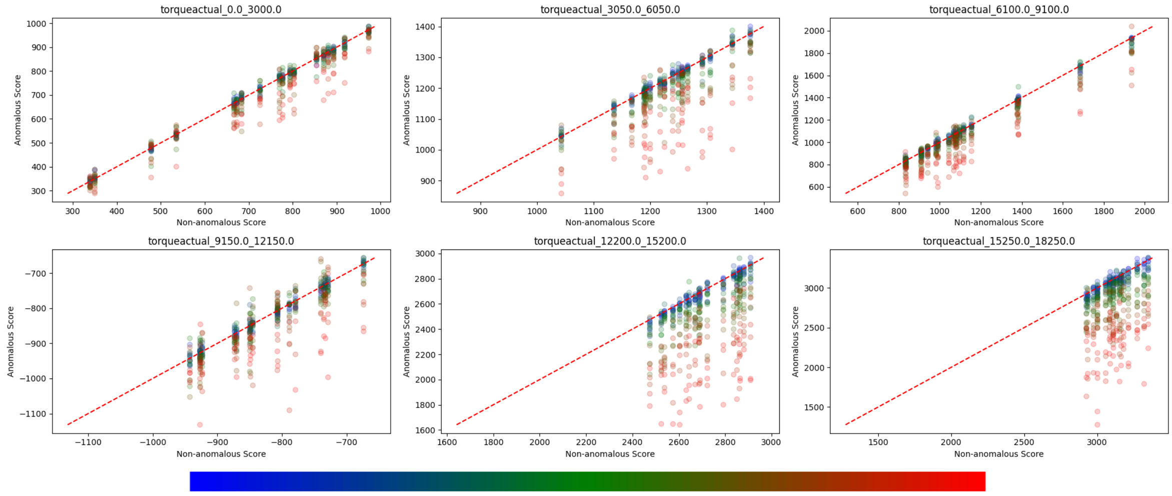


Figure 13: Posterior log-probability change for Point Anomalies

Note: as discussed in section 4.4.2 the model trained for the spectrum defined between the frequencies 9150 to 12150 should is not considered in our results analyses.

As discussed in Section 4.2.1, certain spectra are more effective at detecting anomalies. This observation is supported by the log-probability scatter plots. In Figure 13 as the frequencies within the spectra increase, the accuracy of the model also improves. Contrary to the properties observed in the original signal, higher frequencies contain more information about the presence of anomalies. An extreme case of this effect occurs with a sinusoidal anomaly. When values in a sequence are forced to conform to a sinusoidal shape, the resulting distortion is reflected in the spectrum containing the frequency of the sinusoid. The remaining spectra largely remain unaffected by the anomaly. This can be clearly seen in the inferred probabilities display in Figure 14.

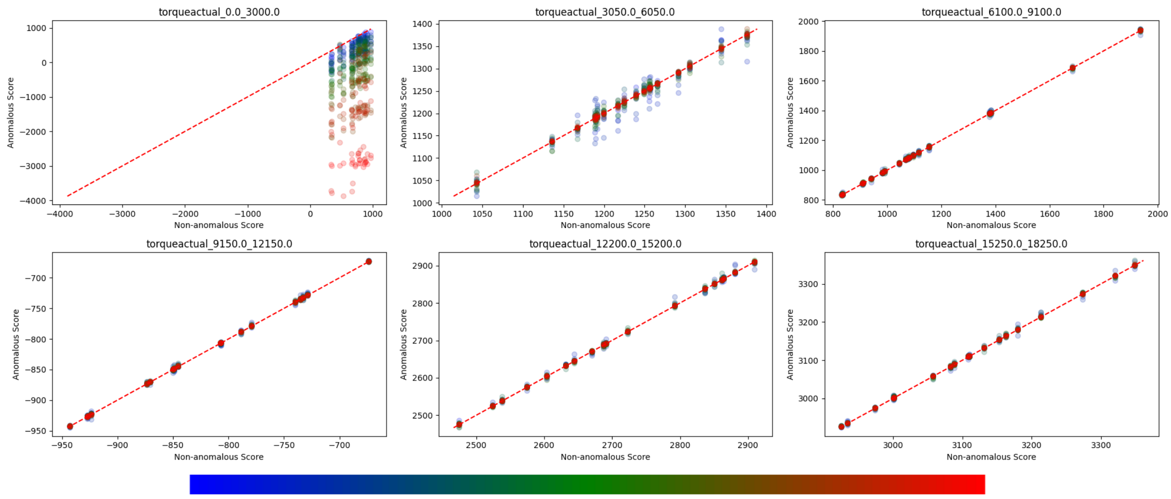


Figure 14: Posterior log-probability change for Sinusoidal Anomalies

Therefore, we can conclude that not all model-spectra are equally capable of capturing the presence of anomalies. However, all models demonstrated an ability to identify anomalies with high precision.

HMM output lower posterior probabilities for the anomalous sequences as the parameters generating them increase, indicating sensitivity of to the anomalies distortion levels.

The trained HMM did not achieve the desired outcome of distinctly separating the inferences across spectra and anomaly types, as shown in Figure 15. There are two primary reasons for these unsatisfactory results. First, the generated anomalies were confined to realistic levels that mimic real-world conditions. Systems designed for predictive maintenance should be capable of detecting even subtle anomalies in operational signal data. Consequently, we chose to use realistic parameters instead of selectively choosing parameters to achieve more favorable outcomes. The second reason pertains to the fact that certain anomalies are only present in specific spectra. As a result, the models trained on these spectra yield posterior probabilities that fall within the non-anomalous distribution.

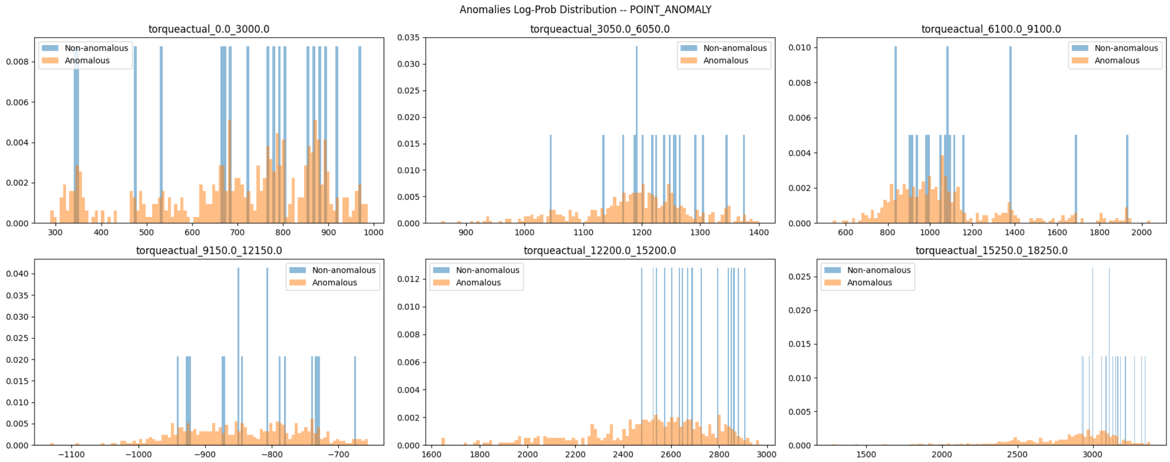


Figure 15: Posterior log-probabilities Histograms for Point Anomalies

To synthesize the outputs from the Hidden Markov Models (HMMs), we trained a Support Vector Machine (SVM) [26]. The SVM's input features consist of a six-value vector representing the log-probabilities generated by each model-spectrum. To enhance separability and introduce non-linearities, these input features are transformed using a Radial Basis Function (RBF) kernel [27]. The SVM training utilized 18 non-anomalous signal probabilities in conjunction with 18 randomly sampled anomalous signal probabilities from the entire pool of generated anomalies. Despite the limited quantity of non-anomalous data, the accuracy results obtained were notably impressive.

In the testing phase, we evaluated the model's performance using a new set of signals: 18 randomly selected anomalous signals and 50 randomly selected non-anomalous signals. The model successfully identified all the non-anomalous signals, but failed to recognize 18 out of the 50 anomalous signals, as detailed in Figure 16.

We conducted a Monte Carlo performance test focusing exclusively on randomly sampled anomalous signals using the SVM. In this experiment, the True Positive Rate (TPR) achieved was 81%, and the False Negative Rate (FNR) was 18%. Upon visually inspecting the false negatives, we determined that they corresponded to anomalies with lower-level distortions, as shown in Figure 17.

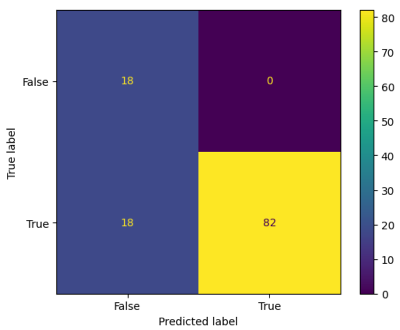


Figure 16: Confusion Matrix

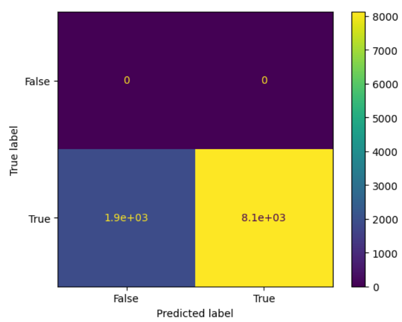


Figure 17: Confusion Matrix Monte Carlo

5 Discussion

This project focused on identifying and developing an effective anomaly detection system for predictive maintenance of industrial robots used at VDL's new battery line. The explored techniques combined data preprocessing, machine learning, and statistical methods for finding anomalies in the torque signals of the robots. Several research questions were posed to facilitate our research, and we will further discuss each of these questions in particular.

RQ 1 **How do anomalies manifest in natural datasets (i.e. data collected from similar robots within the company)?** It was established that anomalies in the data coming from the robots can manifest as spikes, shifts, or fluctuations in the torque signal. Torque anomalies map to real life mechanical issues of the track that can lead to breakages, thus by investigating anomalous patterns in the torque, we can achieve a clearer image of the health of the track.

RQ 2 **How can anomalies be simulated (physically or synthetically) in order to train an anomaly detection model?** Thanks to the mapping of real-world mechanical issues in the track to the torque signal of the robot, it is possible to easily replicate such anomalies synthetically without interfering with the physical system. An anomaly simulation mechanism was devised to recreate different types of torque anomalies and insert them into a sequence in a controlled manner. This mechanism can be applied in further research in the event of absence of labeled anomalies. Nevertheless, it is to be noted that the parameters that control the anomaly have to be chosen accordingly to reflect real anomalous scenarios.

RQ 3 **What predictive approaches can be used to identify anomalies in industrial robots?** Upon investigating three classes of predictive approaches such as signal processing, statistical methods, and machine learning it was established that all three can lead to meaningful anomaly detection in the robot data. In this project, we explored signal decomposition used together with HMM models, z-score-based statistical models, and machine learning models such as Isolation Forest and Local Outlier Factor.

The results of the HMM models trained with decomposed signals are promising. The model is sensitive toward injected anomalies and outputs low posterior probabilities, which means that it successfully identifies anomalous sequences. In the case of the z-score baseline models, a strong comparable performance to HMM is achieved for anomalies that are based on amplitude (shifts, sinusoidal). While Isolation Forest was good for isolating single torque spikes as anomalies, it struggled to single out any other types of anomalies. Lastly, Local Outlier Factor managed to achieve interesting results that could prompt further investigation. The model achieved high accuracy scores and good precision for anomalous patterns while being less prone to false positives.

RQ 4 **Which anomaly detection approach is best suited for a real-world application of predictive maintenance at VDL's battery line?** In a production environment there are certain aspects that are usually prioritized. An anomaly detection system that is used to prompt maintenance before breakages occur should strike a balance between identifying the anomalies that could potentially indicate breakage risks, but at the same time ignore anomalies that do not impact the health of the physical system. An anomaly detection system that is sensitive (HMM) would deal well with the former, while a system that is not as sensitive (LOF) will do better with the latter aspect.

What is more suited for VDL's battery line depends on the business's resources and goals, and the tradeoffs it is willing to make. On one hand, we have that HMM is a sensitive, high-performing model that requires more resources. On the other hand, LOF is a fast and simple approach that might sometimes miss anomalies. Nevertheless, with proper tuning of either of the two models, LOF or HMM, an optimal balance can be achieved.

As for the z-score-based statistical method, it would be less preferred in comparison due to its shortcomings related to the detection of frequency-based anomalies that are expected to happen in real production scenarios.

RQ 5 **How can such models be evaluated in the dearth of real-world anomaly data?** With limited real-world anomaly data, the only way to evaluate the performance of our approaches was by simulating anomalies and testing the ability of our models to identify each type of artificially injected anomaly. However, without any concrete real examples, it is difficult to assess how closely the artificial anomalies replicate the real ones.

5.1 Challenges and Limitations

Throughout this project, we had to take on an array of different challenges:

- **Dataset:** With a lack of data from the battery line, the project had to be carried out on data from similar robots. That data was mixed and contained a staggering amount of duplicates which lowers the amount of accessible data we had – a threat to the validity of our models.
- **Anomalies:** Our dataset did not contain any known natural anomalies which could potentially skew the results. Similarly, it could cause misleading results obtained by the use of artificial anomalies.
- **Evaluation:** As mentioned before, because the anomalies were generated based on a general understanding of how the physical system works, there could be factors or hidden processes that are unaccounted for, thus leading to biased results.
- **Transferability:** While our models seem to do well on the provided data which comes from various sources, it is still possible that the battery line could have detrimentally different environmental factors that could affect the patterns of the anomalies. Nevertheless, the experimental process of this project has demonstrated that it is possible to apply the same solution to a different environment given that there is available data and some knowledge about anomalies that could potentially occur.

6 Conclusion & Future Work

This project has successfully developed anomaly detection approaches for predictive maintenance in industrial robots at VDL, providing an avenue for further development once the new battery line becomes operational. Models like HMM with spectral decomposition, Isolation Forest, LOF, and z-score statistical model were trained and evaluated using a robust anomaly simulation and injection method. The results show potential for effective predictive maintenance on a production setup similar to the new battery line.

6.1 Future Work

This project represents a stepping stone towards a more robust final solution, thus, based on our results we propose the following future steps:

- Data Acquisition - it is necessary to aquire data form the target source, the batery line. This way the solution can be better adjusted to the specific needs of the battery line.
- Physically simulated anomalies- to better understand the impact of mechanical issues on the torque of the robots, it would be beneficial to physically recreate the mechanical issues and quantify how they manifest in the torque.
- Model tuning for HMM and LOF - some hyper-arameter tuning and further experimentation could ensure the developed solutions are applicable to the new battery line.

References

[1] Anwar Meddaoui, Mustapha Hain, and Adil Hachmoud. The benefits of predictive maintenance in manufacturing excellence: a case study to establish reliable methods for predicting failures. *The International Journal of Advanced Manufacturing Technology*, 128(7-8):3685–3690, 2023.

[2] Redhwan Al-amri, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A Al-Sharafi, and Ammar Ahmed Alkahtani. A review of machine learning and deep learning techniques for anomaly detection in iot data. *Applied Sciences*, 11(12):5320, 2021.

[3] D. Brauckhoff, K. Salamatian, and M. May. Applying pca for traffic anomaly detection: Problems and solutions. pages 2866–2870, 2009.

[4] Joseph Ndong and Kavé Salamatian. Signal processing-based anomaly detection techniques: a comparative analysis. pages 32–39, 2011.

[5] Jinbo Li, Witold Pedrycz, and Iqbal Jamal. Multivariate time series anomaly detection: A framework of hidden markov models. *Applied Soft Computing*, 60:229–240, 2017.

[6] Daehyung Park, Zackory Erickson, Tapomayukh Bhattacharjee, and Charles C Kemp. Multimodal execution monitoring for anomaly detection during robot manipulation. pages 407–414, 2016.

[7] Nico Goernitz, Mikio Braun, and Marius Kloft. Hidden markov anomaly detection. 37:1833–1842, 07–09 Jul 2015.

[8] Julien Lesouple, Cédric Baudoin, Marc Spigai, and Jean-Yves Tourneret. Generalized isolation forest for anomaly detection. *Pattern Recognition Letters*, 149:109–119, 2021.

[9] Dong Xu, Yanjun Wang, Yulong Meng, and Ziying Zhang. An improved data anomaly detection method based on isolation forest. 2:287–291, 2017.

[10] Wahid Salman Al Farizi, Indriana Hidayah, and Muhammad Nur Rizal. Isolation forest based anomaly detection: A systematic literature review. pages 118–122, 2021.

[11] Mohammad Riazi, Osmar Zaiane, Tomoharu Takeuchi, Anthony Maltais, Johannes Günther, and Micheal Lipsett. Detecting the onset of machine failure using anomaly detection methods. In *Big Data Analytics and Knowledge Discovery: 21st International Conference, DaWaK 2019, Linz, Austria, August 26–29, 2019, Proceedings 21*, pages 3–12. Springer, 2019.

[12] Xiangyu Cai, Ruliang Xiao, Zhixia Zeng, Ping Gong, and Youcong Ni. Itran: A novel transformer-based approach for industrial anomaly detection and localization. *Engineering Applications of Artificial Intelligence*, 125:106677, 2023.

[13] Tolga Ergen and Suleyman Serdar Kozat. Unsupervised anomaly detection with lstm neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8):3127–3141, 2020.

[14] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. Network anomaly detection using lstm based autoencoder. page 37–45, 2020.

[15] Arash Golibagh Mahyari et al. Robust predictive maintenance for robotics via unsupervised transfer learning. In *The International FLAIRS Conference Proceedings*, volume 34, 2021.

[16] Tawfik Borgi, Adel Hidri, Benjamin Neef, and Mohamed Saber Naceur. Data analytics for predictive maintenance of industrial robots. In *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, pages 412–417. IEEE, 2017.

[17] Ji-Hyun Yoo, Young-Kook Park, and Seung-Soo Han. Predictive maintenance system for wafer transport robot using k-means algorithm and neural network model. *Electronics*, 11(9):1324, 2022.

[18] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[19] F. Pedregosa, 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. *Outlier detection with Local Outlier Factor (LOF)*, 2023. Accessed: 2024-06-24.

[20] NIST/SEMATECH. e-handbook of statistical methods - detection of outliers.

[21] Wikipedia contributors. Baum–welch algorithm — Wikipedia, the free encyclopedia, 2024. [Online; accessed 23-June-2024].

[22] Wikipedia contributors. Expectation–maximization algorithm — Wikipedia, the free encyclopedia, 2024. [Online; accessed 23-June-2024].

[23] Wikipedia contributors. Confusion matrix — Wikipedia, the free encyclopedia, 2024. [Online; accessed 23-June-2024].

[24] Wikipedia contributors. Generator (computer programming) — Wikipedia, the free encyclopedia, 2024. [Online; accessed 23-June-2024].

[25] Guorong Xuan, Wei Zhang, and Peiqi Chai. Em algorithms of gaussian mixture model and hidden markov model. 1:145–148 vol.1, 2001.

[26] Wikipedia contributors. Support vector machine — Wikipedia, the free encyclopedia, 2024. [Online; accessed 24-June-2024].

[27] Wikipedia contributors. Radial basis function kernel — Wikipedia, the free encyclopedia, 2024. [Online; accessed 24-June-2024].

A Anomalies - How they manifest in natural datasets

One of the main failure points of the robotic system assisting the sealing and screwing of the battery module lid, is the track. The track consists of railings that support the robot and it's movement in two directions - forward and back. When something goes wrong with the track, it can be usually detected in the torque that the robot uses to perform the movement. If the track is not in an ideal state, the robot will usually have to compensate by using more torque.

Depending on the type of track issue, we distinguish a specific pattern of anomalous torque usage. For the scope of this project we consider 4 types of such anomalies:

- **Rail alignment** - A rail alignment issue is characterized by rails that do not connect perfectly due to wrong installation, thus causing the robot to use more torque to traverse them. This can usually be observed as a sudden spike in the torque of the robot in the place where the misalignment happened. Since the torque sequence is recorded for the whole movement forth and back, the robot will have to traverse the same area of the track 2 times, causing us to observe the spikes symmetrically to the middle of the sequence as it can be seen in Figure 18. In such cases the spikes remain consistent until maintenance is performed.
- **Rail degradation** - This is an issue that appears due to usage of the rails. With time, the rails become damaged causing the robot to have fluctuating levels of torque for the position of the rail which is damage, resembling a slight sinusoidal signal as it can be seen in Figure 19.
- **Tolerance** - A tolerance problem appears when the bearings in the rails become bad over time. This causes more torque to be used on average and a general inability of the gearbox to balance the speed which manifests as bigger spike fluctuations. An example of such anomaly can be seen in Figure 21.
- **Gearbox** - when there is an issue with the gearbox, it becomes hard for the robot to turn which causes it to use more torque at the turning point which is the middle of the sequence. We can observe big spikes near the middle big spike (see 22.

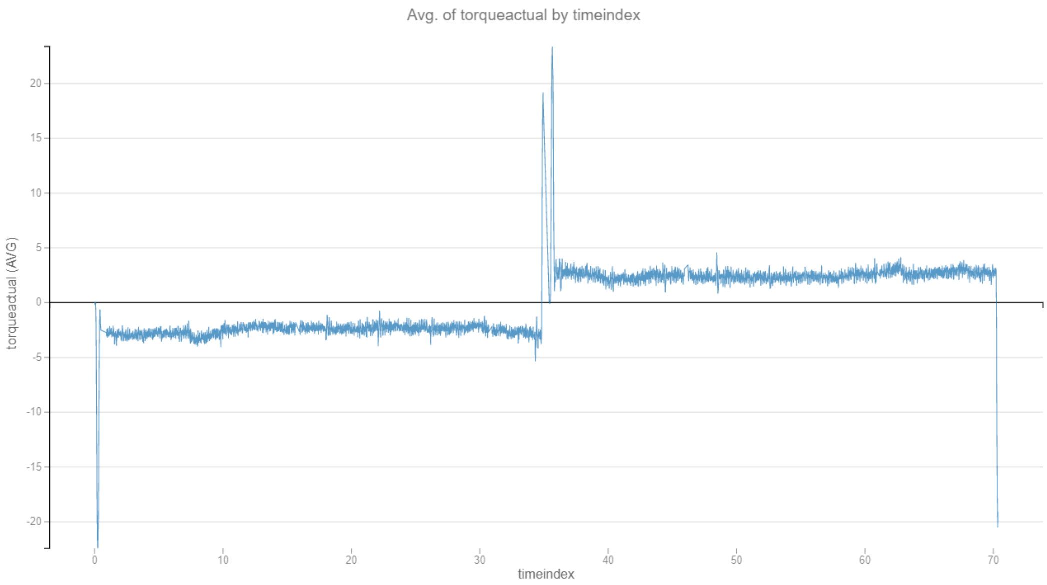


Figure 18: Rail Misalignment represented by evenly spaced spikes in the torque

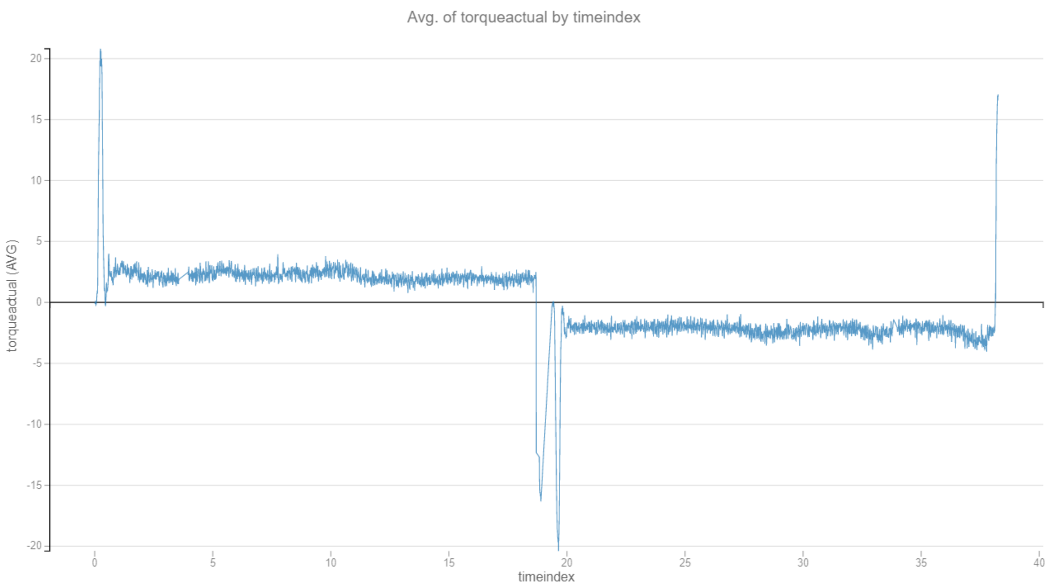
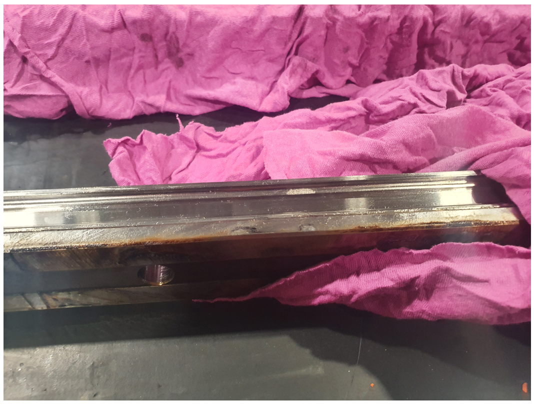
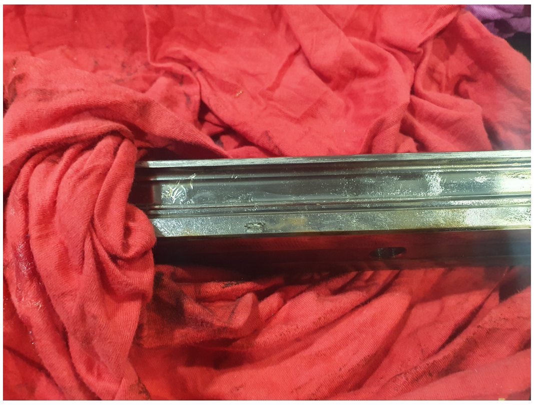


Figure 19: Rail degradation seen as a fluctuation in torque

(a)



(b)



(c)

(d)

Figure 20: a-d: Degradation of the rails

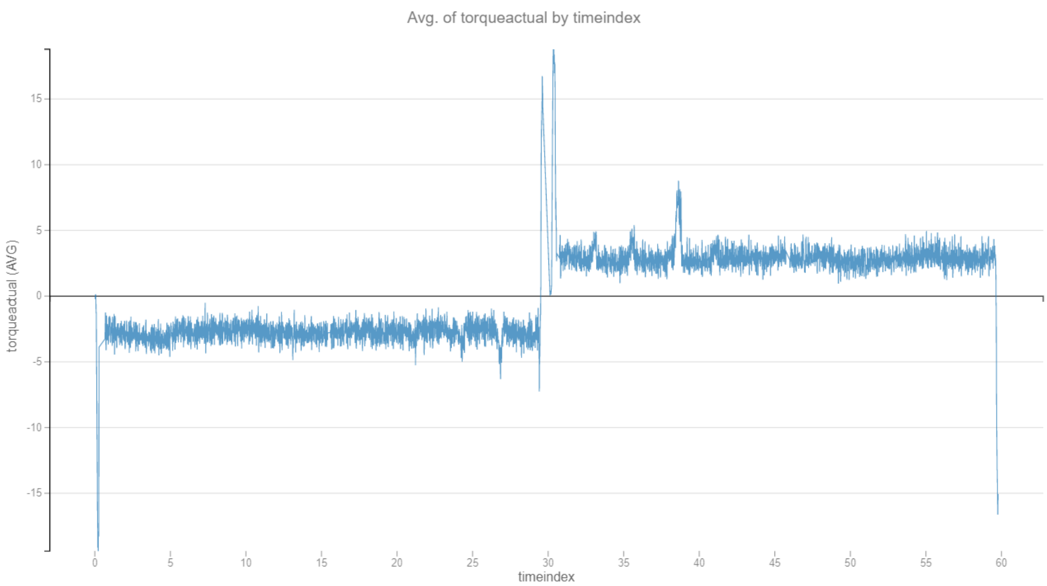


Figure 21: Rail bearings anomaly

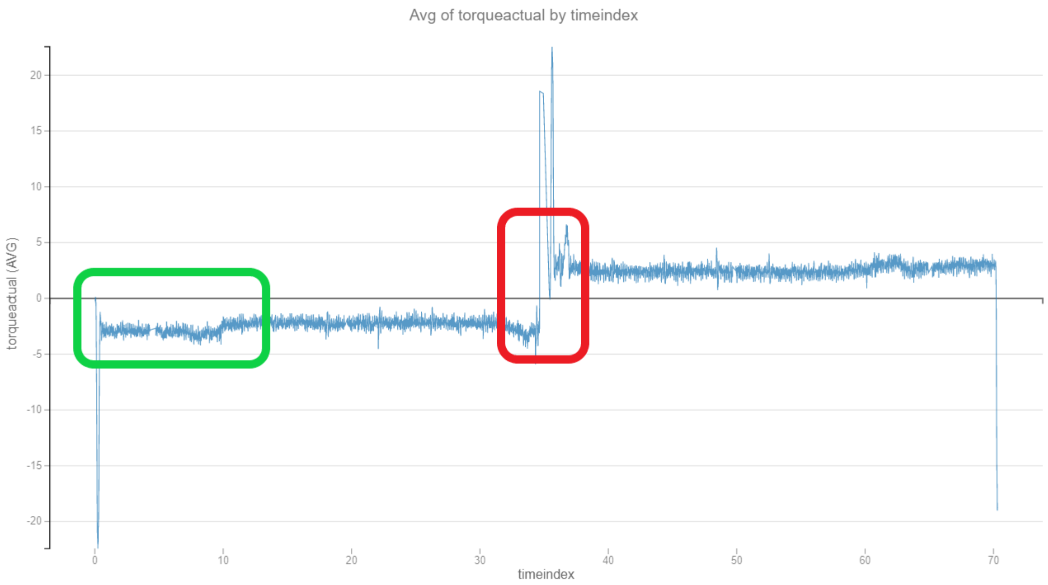


Figure 22: Gearbox anomaly

B Example of Decomposed Signal

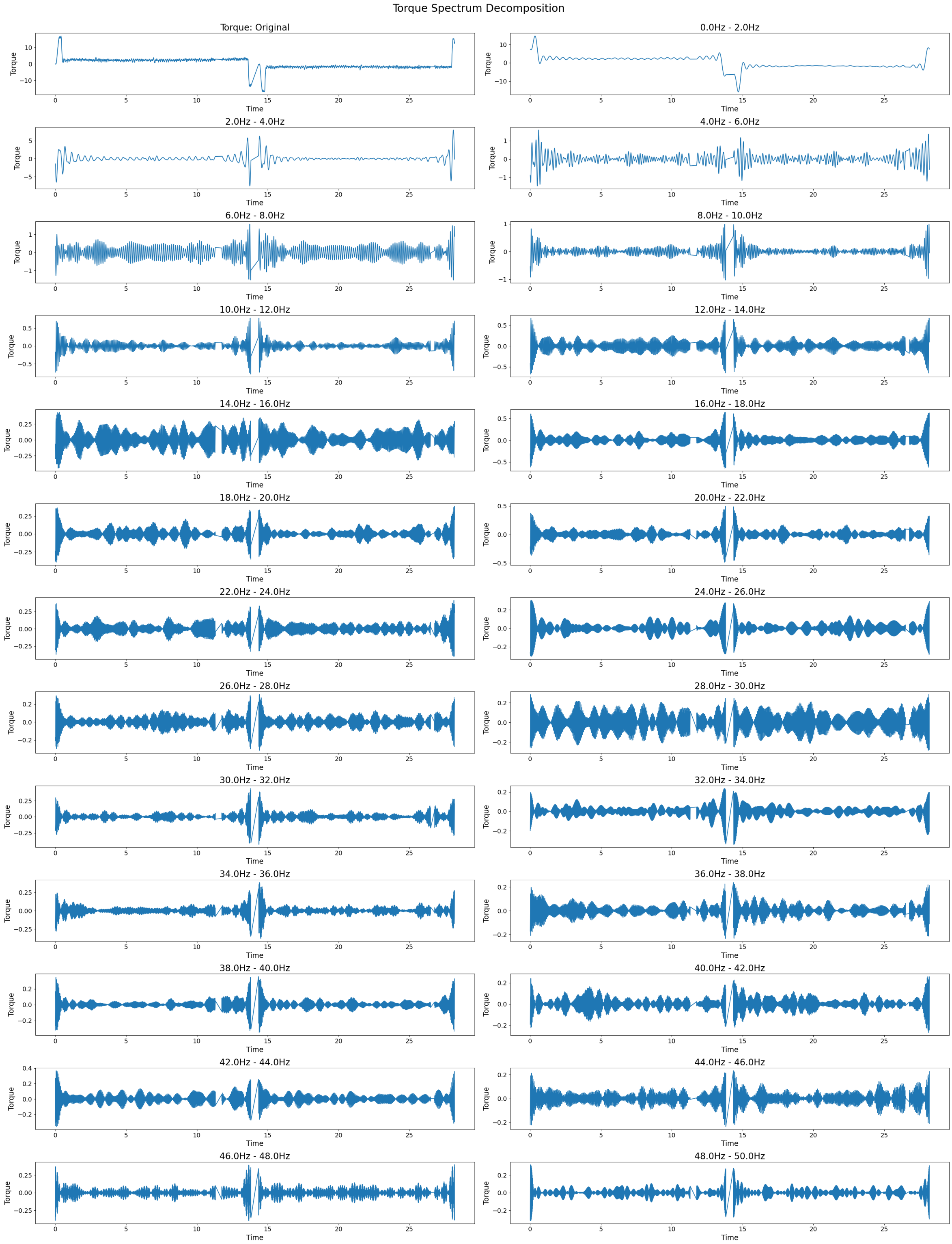


Figure 23: Torque signal decomposed in different spectra.

C Local Outlier Factor Experiment Results

Table 7: Local Outlier Factor - novelty detection on artificial anomalies experiment results

anomaly	**threshold**	**params**	**accuracy**	**precision**	**recall**	**f1**
gaussian	0.5	0.1	0.5	0.56	0.01	0.01
gaussian	0.5	0.2	0.5	0.56	0.03	0.06
gaussian	0.5	0.3	0.51	0.55	0.07	0.12
gaussian	0.5	0.4	0.52	0.56	0.14	0.22
gaussian	0.75	0.1	0.5	0.6	0.0	0.01
gaussian	0.75	0.2	0.5	0.51	0.02	0.03
gaussian	0.75	0.3	0.5	0.53	0.05	0.09
gaussian	0.75	0.4	0.51	0.54	0.1	0.16
gaussian	1.0	0.1	0.5	0.51	0.0	0.0
gaussian	1.0	0.2	0.5	0.53	0.01	0.02
gaussian	1.0	0.3	0.5	0.53	0.04	0.07
gaussian	1.0	0.4	0.51	0.55	0.07	0.13
gaussian	1.5	0.1	0.5	0.53	0.0	0.0
gaussian	1.5	0.2	0.5	0.57	0.01	0.01
gaussian	1.5	0.3	0.5	0.56	0.02	0.03
gaussian	1.5	0.4	0.51	0.58	0.04	0.07
point	0.5	(10, 1)	1.0	0.14	0.34	0.19
point	0.5	(10, 2)	1.0	0.16	0.54	0.25
point	0.5	(5, 1)	1.0	0.07	0.33	0.11
point	0.5	(5, 2)	1.0	0.1	0.49	0.16
point	0.75	(10, 1)	1.0	0.22	0.33	0.26
point	0.75	(10, 2)	1.0	0.29	0.45	0.36
point	0.75	(5, 1)	1.0	0.16	0.3	0.2
point	0.75	(5, 2)	1.0	0.21	0.47	0.29
point	1.0	(10, 1)	1.0	0.28	0.25	0.27
point	1.0	(10, 2)	1.0	0.34	0.37	0.35
point	1.0	(5, 1)	1.0	0.22	0.26	0.24
point	1.0	(5, 2)	1.0	0.26	0.35	0.3
point	1.5	(10, 1)	1.0	0.37	0.19	0.25
point	1.5	(10, 2)	1.0	0.45	0.31	0.37
point	1.5	(5, 1)	1.0	0.26	0.12	0.17
point	1.5	(5, 2)	1.0	0.31	0.27	0.29
shift	0.5	1	0.97	0.34	0.04	0.07
shift	0.5	10	0.99	0.64	0.46	0.53
shift	0.5	4	0.98	0.43	0.16	0.23
shift	0.5	7	0.98	0.76	0.42	0.54
shift	0.75	1	0.98	0.12	0.01	0.01
shift	0.75	10	0.99	0.88	0.59	0.7
shift	0.75	4	0.98	0.71	0.28	0.4
shift	0.75	7	0.99	0.81	0.38	0.52
shift	1.0	1	0.98	0.06	0.0	0.0
shift	1.0	10	0.99	0.86	0.43	0.57
shift	1.0	4	0.99	0.62	0.2	0.3
shift	1.0	7	0.99	0.95	0.47	0.63
shift	1.5	1	0.98	0.89	0.08	0.15
shift	1.5	10	0.99	0.96	0.34	0.5
shift	1.5	4	0.98	0.92	0.11	0.2
shift	1.5	7	0.99	0.79	0.3	0.44
sinusoidal	0.5	1	0.98	0.48	0.37	0.42
sinusoidal	0.5	2	0.98	0.57	0.7	0.63
sinusoidal	0.5	3	0.98	0.55	0.7	0.61
sinusoidal	0.5	4	0.98	0.56	0.7	0.62

Table 7: Local Outlier Factor - novelty detection on artificial anomalies experiment results

anomaly	threshold	params	accuracy	precision	recall	f1
sinusoidal	0.75	1	0.98	0.54	0.33	0.41
sinusoidal	0.75	2	0.98	0.57	0.62	0.6
sinusoidal	0.75	3	0.98	0.54	0.65	0.59
sinusoidal	0.75	4	0.98	0.58	0.69	0.63
sinusoidal	1.0	1	0.98	0.62	0.32	0.42
sinusoidal	1.0	2	0.99	0.59	0.62	0.61
sinusoidal	1.0	3	0.98	0.56	0.55	0.56
sinusoidal	1.0	4	0.98	0.59	0.63	0.61
sinusoidal	1.5	1	0.98	0.53	0.18	0.27
sinusoidal	1.5	2	0.98	0.61	0.52	0.56
sinusoidal	1.5	3	0.98	0.62	0.57	0.59
sinusoidal	1.5	4	0.98	0.66	0.63	0.65

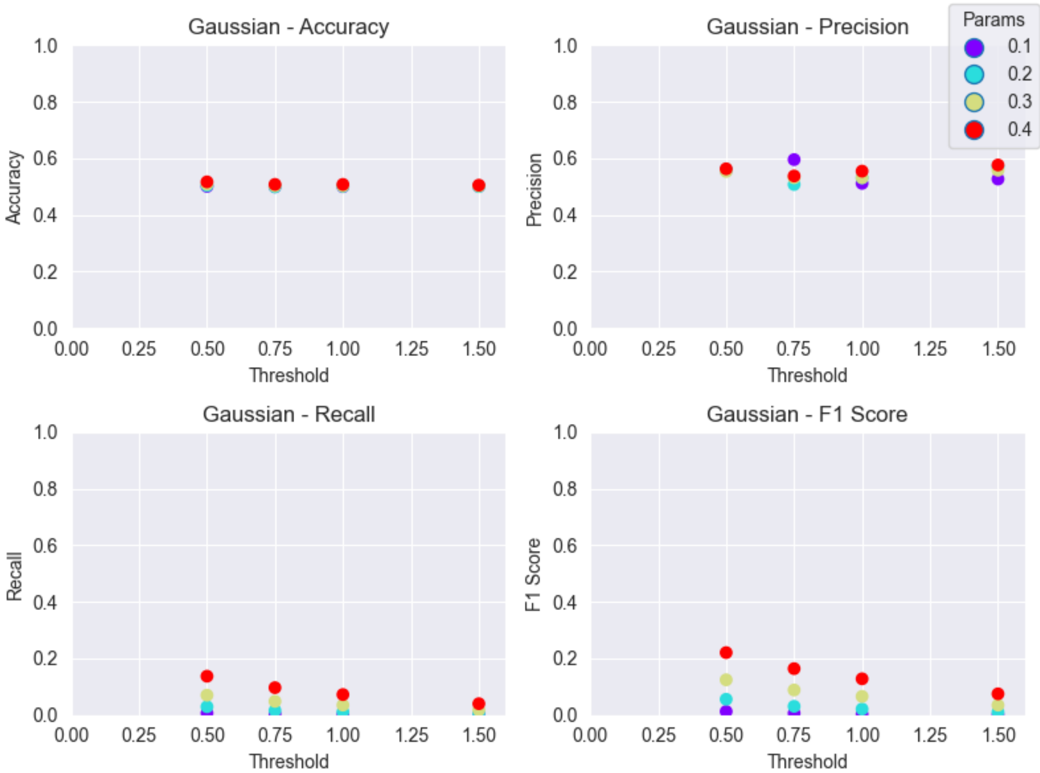


Figure 24: Results of LOF Experiments - Gaussian Anomaly

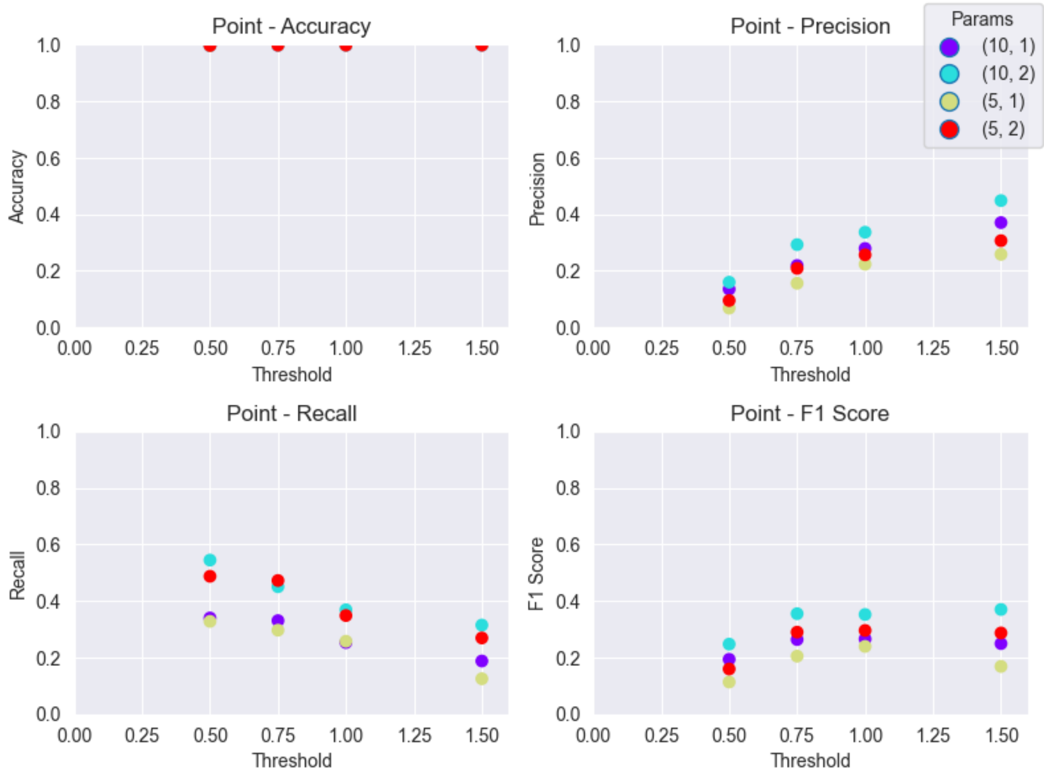


Figure 25: Results of LOF Experiments - Point Anomaly

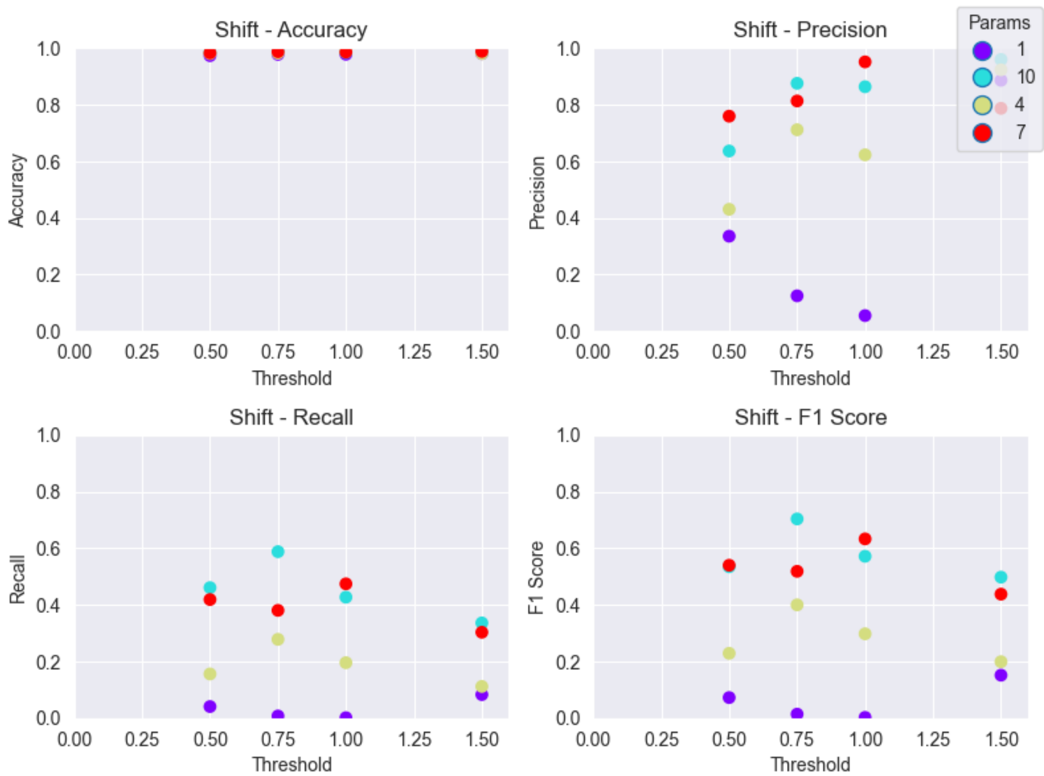


Figure 26: Results of LOF Experiments - Shift Anomaly

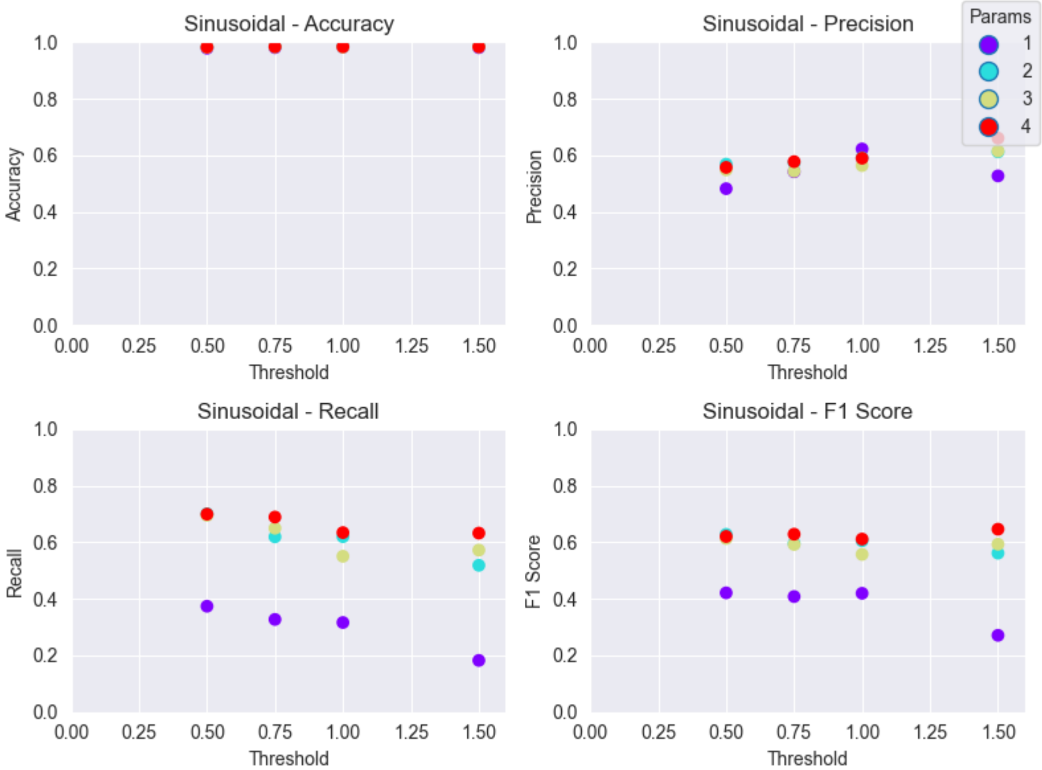


Figure 27: Results of LOF Experiments - Sinusoidal Anomaly

D Evaluation Hidden Markov Models

D.1 Posterior Probabilities Scatter Plots

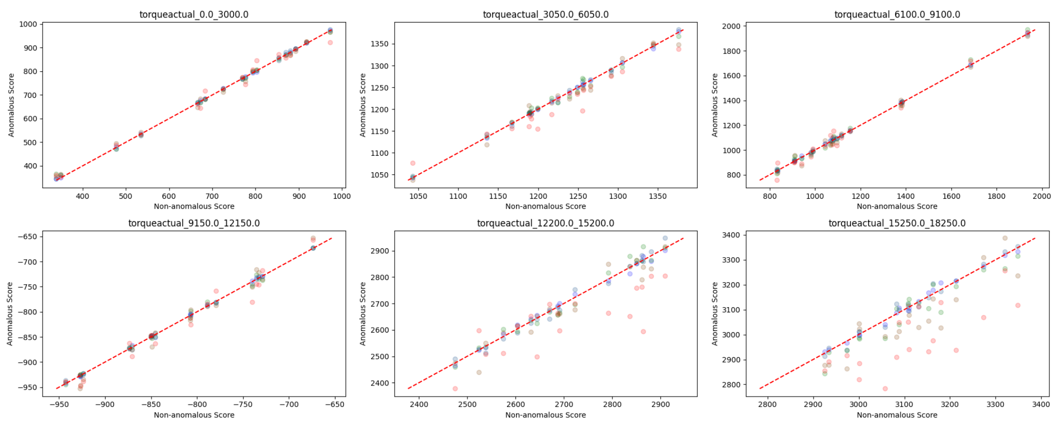


Figure 28: Posterior log-probability change for Gaussian Noise Anomalies

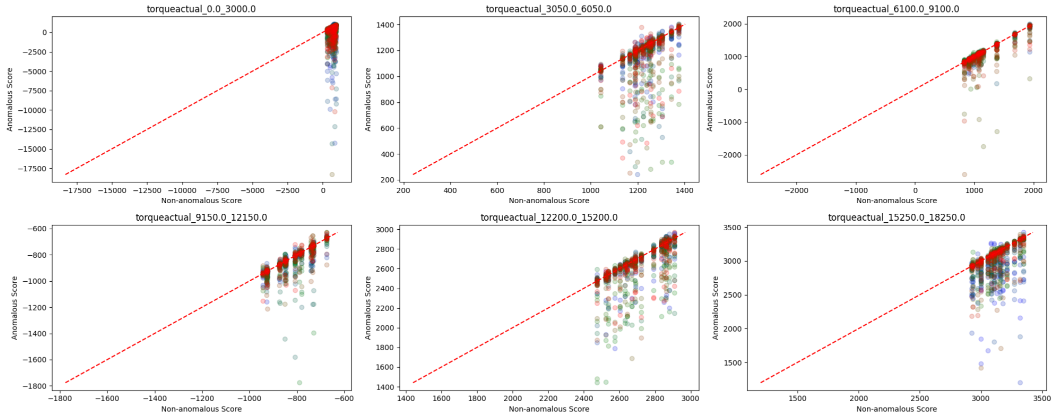


Figure 29: Posterior log-probability change for Shift Anomalies

D.2 Posterior Probabilities Histogram Plots

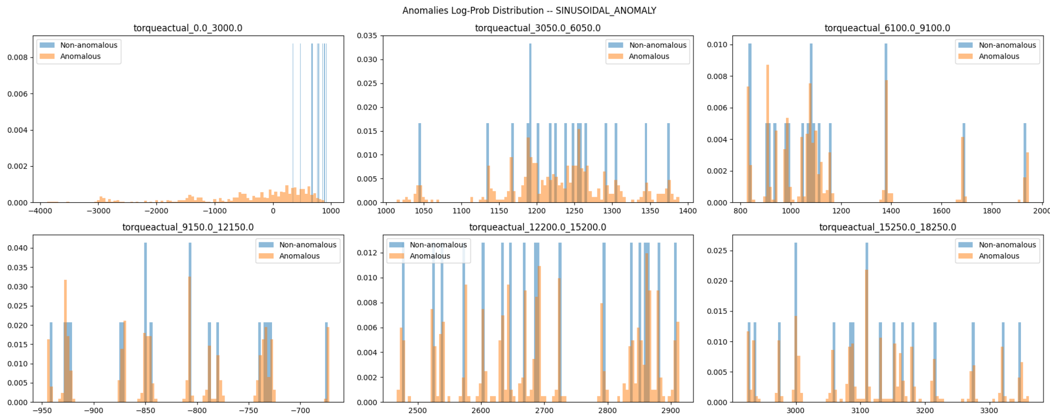


Figure 30: Posterior log-probability histogram for Sinusoidal Anomalies

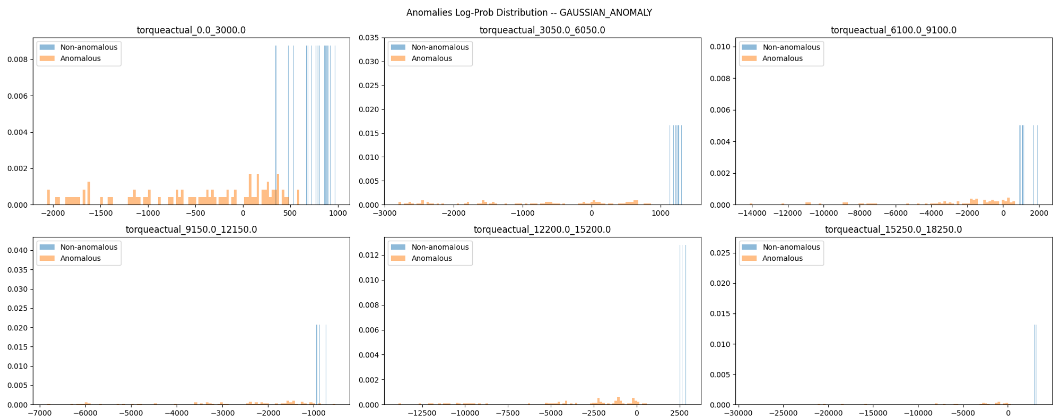


Figure 31: Posterior log-probability histogram for Gaussian Anomalies

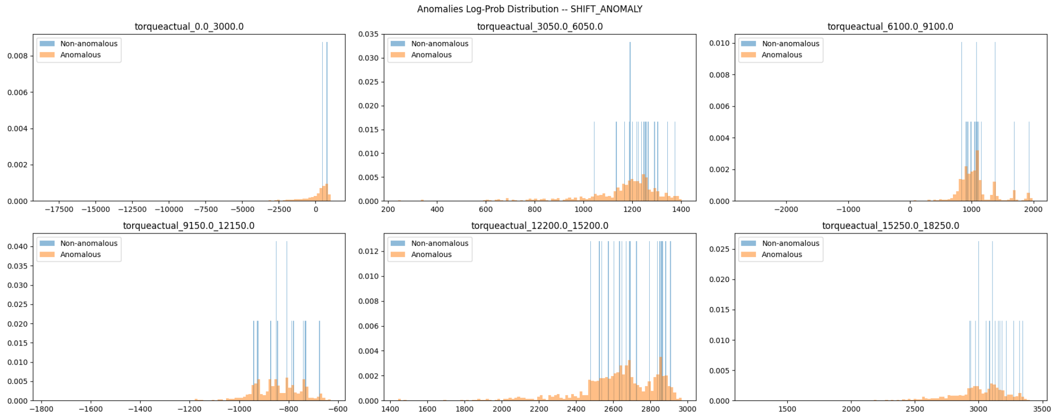


Figure 32: Posterior log-probability histogram for Shift Anomalies