Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Padova

Department of Information Engineering

Master's Degree in
Computer Engineering

Machine Learning-based Anomaly Detection for Hydroelectric Power Plants

Supervisor:
Prof. Gian Antonio Susto

Candidate:
Mattia Fanan
2019138

ACADEMIC YEAR 2021-2022
Graduation date: 5 September 2022

Abstract

Purpose - This paper analyzes how anomaly detection can effectively be applied to predictive maintenance in hydroelectric power plants and evaluates the effects SHAP values have in helping the user understand the causes of the reported anomaly.

Methods - We compared the performance of RRCF, KDE, SOD, ECOD, isolation forest, auto-encoder, DeepSVDD, KitNet, and RSHash on data coming from a real-world hydroelectric power plant, through human evaluation (since no label was available and the labeling process was too expensive). Then we checked whether SHAP applied to the best performing model gives informative and correct indications on the cause of the anomaly.

Results - Our results showed that auto-encoders can catch all of the organization's recorded anomalies and propose additional ones, that are later confirmed by the expert of the domain. The application of SHAP is sufficiently able to guide the user toward the features related to the anomaly but is a little slow to be applied in streaming data.

Implications - From a practical perspective, more efficient maintenance leads to less operative costs and higher reliability of the plant, on top of this we used models and algorithms from well-known Python packages so this work is ready to be applied as a production tool. From a social perspective, reducing the costs makes hydroelectric technology more appealing to investors, helping the transition towards renewable energies.

Keywords - Anomaly Detection, Hydroelectric, Interpretability, Machine Learning, Maintenance.

Sommario

Scopo - Questo lavoro analizza come il rilevamento delle anomalie possa essere efficacemente applicato alla manutenzione predittiva delle centrali idroelettriche e valuta gli effetti degli SHAP values nell'aiutare l'utente a comprendere le cause dell'anomalia segnalata.

Metodo - Abbiamo confrontato le prestazioni di RRCF, KDE, SOD, ECO, isolation forest, auto-encoder, DeepSVDD, KitNet e RSHash su un insieme di dati provenienti da una vera centrale idroelettrica, attraverso una valutazione umana (poiché non erano disponibili labels e procurarsele sarebbe stato troppo costoso). Poi abbiamo verificato se SHAP, applicato al modello più performante, fornisce indicazioni informative e corrette sulla causa dell'anomalia.

Risultati - I nostri risultati hanno mostrato che gli auto-encoder sono in grado di cogliere tutte le anomalie registrate dall'organizzazione e di proporne altre, che vengono poi confermate dall'esperto di dominio. L'applicazione di SHAP è sufficientemente in grado di guidare l'utente verso le features relative all'anomalia, ma è un po' lenta per essere applicata ai dati in streaming.

Implicazioni - Da un punto di vista pratico, una manutenzione più efficiente porta a minori costi operativi e a una maggiore affidabilità dell'impianto; inoltre, abbiamo utilizzato modelli e algoritmi tratti da noti pacchetti Python, quindi questo lavoro è pronto per essere applicato come strumento di produzione. Da un punto di vista sociale, la riduzione dei costi rende la tecnologia idroelettrica più interessante per gli investitori, favorendo la transizione verso le energie rinnovabili.

Contents

1 Introduction **1**

2 Maintenance **5**

2.1 Reactive maintenance 6

2.2 Preventive maintenance 7

2.3 Predictive maintenance 8

2.4 Proactive maintenance 9

2.5 Reliability Centered Maintenance 10

3 Machine learning **11**

3.1 Types of learning 12

3.2 Anomaly detection 13

3.3 Model Interpretability 14

3.4 Models 16

3.4.1 IsolationForest 16

3.4.2 Auto-Encoder 17

3.4.3 DeepSVDD 18

3.4.4 KitNet 19

3.4.5 RSHash 20

4 Hydroelectric power plant **21**

4.1 How it works 23

4.2 Operating point 24

4.3 Vibration analysis 25

4.4 Time domain analysis 26

4.5 Frequency domain analysis 27

4.6 Orbit analysis 29

5 Case Study **31**
5.1 State of the art 31
5.1.1 Remaining Useful Life & Risk estimation 32
5.1.2 Highly Multivariate Analysis 33
5.2 Context 34
5.2.1 Data 34
5.2.2 Features 36
5.2.3 Safety 37
5.3 Experiment 38
5.3.1 Choices 38
5.3.2 Anomalies 40
5.3.3 Models' comparison 42

6 Results **43**
6.1 IsolationForest 43
6.2 Auto-Encoder 45
6.3 DeepSVDD 47
6.4 KitNet 49
6.5 RSHash 51
6.6 SHAP 53
6.6.1 Post maintenance misalignment 53
6.6.2 Anomaly of January 30 2022 55
6.6.3 Unconnected sensor 56
6.6.4 New anomaly 58
6.6.5 More recent anomalous period 59

7 Conclusions **61**

A Other experiments **75**
A.1 RRCF 76
A.2 KDE 77
A.3 SOD 78
A.4 ECOD 79

Chapter 1

Introduction

In response to the effects of climate change, in the last years, our society has increased its attention towards the transition from traditional forms of energy production to more sustainable ones, called green energy or renewable energy. This trend is reinforced by governments' plans like the European *green deal* [1]. In this context hydroelectric power plants not only provide the largest share of renewable energy, accounting for more or less 16% of the whole worldwide energy production [2]. But also can be used to mitigate climate change's effects like droughts and floods or to manage water supplies [3].

For the economy of a hydroelectric power plant, the optimization of maintenance is crucial. The size of hydroelectric power plants is generally huge, such that as reported in [4] it can happen that the energy produced accounts for a big share of an entire nation's requirements. So, given that maintenance can reduce the production for several weeks, it is generally the case that they are scheduled in advance to give time for acquiring the pieces and to re-organize the energy production plans. To be able to schedule maintenance in advance minimizing the costs, at least the predictive maintenance policy is adopted, making it necessary to analyze the data collected from the plant's sensors to predict the possible next fault of the system.

The data generated from hydroelectric power plants and in general in the vast majority of industry 4.0 is very high-dimensional, such that automatic systems are used to assert the satisfaction of a handmade set of conditions. But in this way, some anomalous behaviors that don't violate any of the conditions and don't generate any side effects (e.g. enhanced noise in the plant) cannot be discovered, since it is not humanly possible for the personnel to watch the entire stream of data. So, to help the personnel discover all anomalous behaviors of the system, in this thesis work we propose to find an appropriate machine learning model that can both find all the anomalies we were searching for and contextually give the highest quality information to the users. To do this, we compared some of the most promising anomaly detection models that can be found in the literature on the data coming from a real-world hydroelectric power plant.

On top of the comparison, we also tested the applicability of an additional layer that implements the techniques from eXplainable AI (XAI), which means to allow the human users to understand why and how the model came up with its decision. This is especially important when the decision that arises from that result has a big impact in terms of risks, costs, income or safety [5]. Take as an example an algorithm that discovers cancer from the patient's medical exams, you cannot say to the patient "you are 100% safe since the machine in the right corner of the room says so" both for legal reasons and since the action of the doctor is necessary in these situations, but on the other hand the doctor alone can miss some cases. Here is simple to understand that if the doctor can understand the model and use its results as a tool, then the optimal performance can be reached. In this work, we tried to apply a specific algorithm called SHapley Additive exPlanations (SHAP) [6] to help the user find the root causes of the reported anomalies. To the best of the authors' knowledge at the time of writing, this technology has been very recently applied with various objectives in the field of hydroelectric like Flow-duration curves prediction [7] or reservoir water availability forecast [8], or it has been applied with similar objectives but not within the hydroelectric context like [9], [10] or [11]. But never to enhance predictive maintenance in the context of hydroelectric power plants.

For the realization of this work, an essential contribution comes from the collaboration with ANDRITZ HYDRO S.r.l. Unipersonale. Thanks to their help we were able to obtain the data from a real power plant, gather insights about the actual needs of the hydroelectric field, and most importantly the possibility to compare the results of the models with the knowledge only an expert of the domain can have.

The rest of the thesis is organized as follows: Chapter 2 is devoted to a brief introduction to the maintenance's theory along with a clarification about the nomenclature, since it may vary from source to source. Chapter 3 introduces the notions that represent the core of this work, namely machine learning, anomaly detection, interpretability, and the models that here are compared. Then Chapter 4 give a broad overview of how a hydroelectric power plant works, why vibration analysis is important in the context of maintenance, and the common way vibrations are analyzed.

In Chapter 5 we first present a snapshot of the current state of the art to deal with this type of task. After that, we introduce the context of this work, where we explain the particular characteristics the data in this field has, the features we had along with their connection with the problems explained in Chapter 4, to end with a small digression about the requirements of safety the hydroelectric field imposes. Finally, we report the choices we made, the list of the anomalies we used in the model comparison, and how the comparison is designed. In Chapter 6 there are the results from the models with the better performance, the ones introduced in Section 3.4. The others are moved to the appendix to improve readability. Finally, conclusions and future works are discussed in Chapter 7.

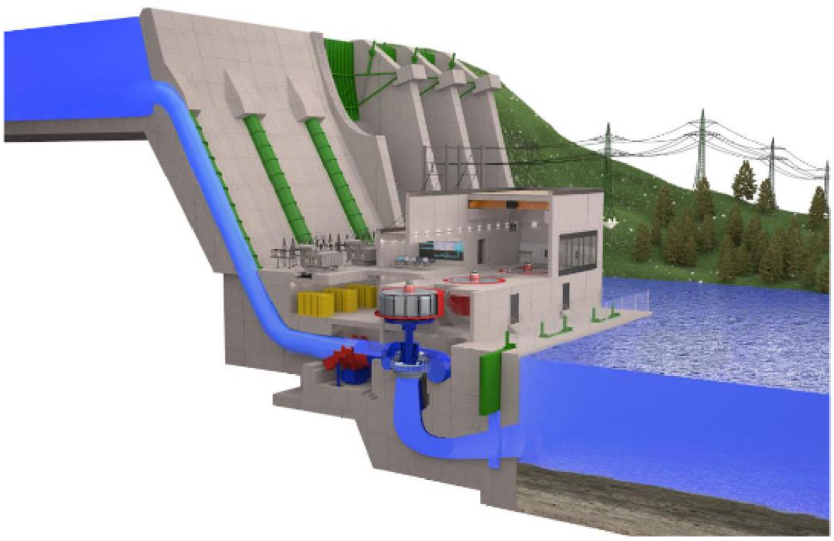


Figure 1.1: Example of a hydroelectric power plant [12]

Chapter 2

Maintenance

Maintenance is an essential concept for modern companies' profitability and competitiveness since it ensures that the standards regarding the system life (management of the asset), safety, availability, efficiency, and quality are respected [13]. An official definition of the term maintenance can be found in the norm ISO:14224 of the year 2004 as "the combination of technical and associated administrative actions intended to retain an item or system in, or restore it to, a state in which it can perform its required function".

From the point of view of a company, in terms of operative costs, the maintenance operations can be very demanding. To make an example we report the statistics from [14], where they found out that the component related to the maintenance can amount from a lower bound of 15% up to 70% of the total production costs. Another important aspect is related to the availability of the production plant since the maintenance operations can last for very long periods depending on the unit to be restored and in some cases, the whole system has to be stopped. So over the years, many strategies have been proposed with the objective of both reducing the impact on the costs and increasing the availability of the system.

In the literature, the maintenance strategies' nomenclature is not always consistent. To have a base to start we can take as reference [15], where the author clarifies what is correct nomenclature currently used in the industry. That is, the maintenance strategies can be categorized into Reactive, Preventive, Predictive, and Proactive. In the following, these four main strategies will be explained more in detail along with Reliability Centered Maintenance, which usually is excluded by the aforementioned group of four since it consists in actuating one of the strategies for each component instead of the same strategy for the whole system.

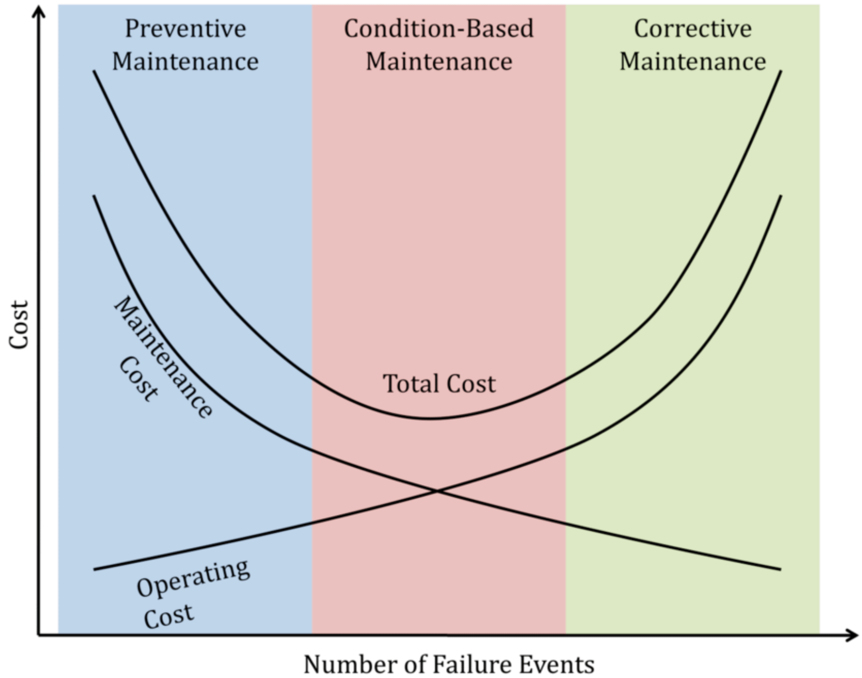


Figure 2.1: Graph of the costs associated with the maintenance strategies [16] (Proactive is often merged with Predictive) .

2.1 Reactive maintenance

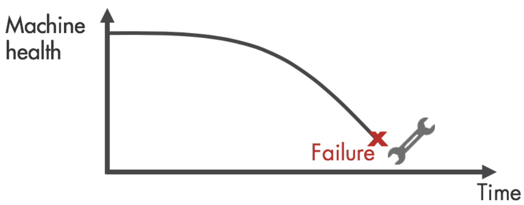
Reactive maintenance is usually also found in the literature under the names of run-to-failure (R2F), failure-based maintenance, or corrective maintenance. Historically it is the first strategy that has been applied in industry [17] and it is the simplest of all the policies. The idea on which it is based is to apply maintenance reacting to the failure, that is when a unit of the system doesn't work properly anymore.

The main advantage it has with respect to the other maintenance strategies is simplicity since we only need a way to tell if a component is not working properly. The drawbacks of this strategy then reside in the fact that the system may be subject to unpredictable stops of production and that contextually with the arising of a fault, also hazards for the safety of the rest of the system or the personnel increase. Consequently, this type of policy can only be applied where the failure of the component is for the most part self-contained and anyway we need to be sure the risks mentioned above are avoided.

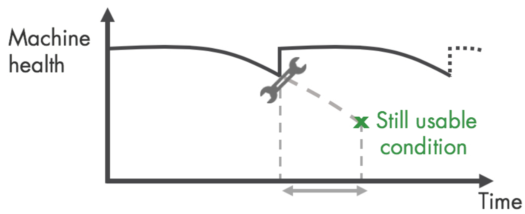
2.2 Preventive maintenance

Also known as scheduled or interval-based maintenance, preventive maintenance (PvM) is a more advanced strategy with respect to the reactive one, it has the objective to avoid any unplanned failure of the system since in many scenarios it is not acceptable to suddenly stop the production. The idea on which this category of maintenance strategies is based is that the system can be analyzed to formalize temporal spans in which the properties of the component are ensured with high probability. An example of an application of this strategy could be the oil change in the car, you are supposed to change it after a certain traveled distance even if it could have been used for some more time.

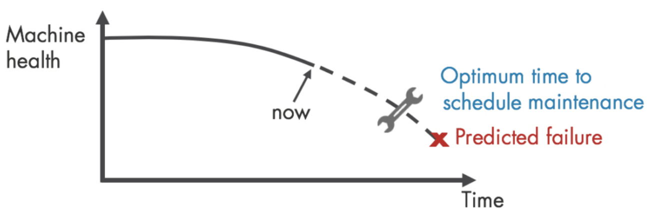
The advantage of this strategy with respect to the reactive one is that the failures are very likely to be avoided. The downsides are the introduction of an in-depth analysis of the system to identify the time constraints and that the life of the components is not exploited in its entirety, leading to more expenses in equipment and personnel.



(a) Reactive maintenance strategy.



(b) Preventive maintenance strategy. Notice the unexploited part of the component life.



(c) Predictive maintenance strategy. Notice how the optimal time for maintenance can be chosen in the period between the time the failure is reported and its onset.

Figure 2.2: Representation of when the maintenance strategies are applied with respect to the failure onset [18].

2.3 Predictive maintenance

Predictive maintenance (PdM) in literature is also known as condition-based maintenance (CBM). The main idea behind this type of maintenance strategy is to measure and analyze the system state, to be able to spot in advance dangerous trends that will definitely lead to the failure of some units in the system. Then a simple way to set alarms on these trends that are usually applied in industry is to set conditions on them, from which comes the name CBM.

An example of an application of this maintenance strategy can be found in [15]. In the context of car maintenance, the brake pads are usually changed when their thickness gets under 2 or 3 millimeters. As we can see the condition is on the state of the pads (their thickness) and the trend measures the erosion accumulated by the pads. It's common knowledge that when the pads are completely eroded then they can't stop the car anymore, so a threshold is placed in such a way to have enough time to appoint the maintenance while being able to still safely use the car.

The advantage of this strategy with respect to the ones we introduced before is that the balance between the exploitation of the unit's lifespan and the avoidance of unexpected failures is optimized to the needs of the user. So after the alarm on a trend measure (key trend indicator KTI) is set, the users have enough time to buy the components and to set the date of the maintenance in the time spot that is more convenient to impact the less is possible the production. The effect is that the maintenance-related costs are comparable with R2F since the maintenance is applied only when needed while the system's reliability, availability, and thus profitability are increased. The downside instead is that the complexity of the structure to produce, store and manage the flow of information that is required to keep an eye on the KTI is a burden that must be taken care of from the company or outsourced.

However, the availability of large quantities of data that comes from the use of predictive maintenance is essential to allow centralized remote control of the system and artificial intelligence models to be exploited in the task of optimization of the maintenance, learning from the data itself.

2.4 Proactive maintenance

As stated in [19] proactive maintenance (PaM) is a natural evolution of predictive maintenance when the trained personnel started to use all the knowledge necessary to deal with the information that came with the use of the aforementioned maintenance strategy to also search for the root causes of the detected failures, and then when possible those causes were removed to extend the lifespan of the system. But in some sources like [20], the two are simply merged under the name of predictive maintenance.

Since it may be cumbersome to understand how proactive maintenance works from the plain definition, we report some examples from [15] and [21] that can help to clarify the explanation.

In the context of car maintenance, it is common knowledge that tires have to be inflated to a certain pressure and their angle with respect to the axis of the car has to be set to a given value to distribute more evenly their degradation in time. So when the user goes to the tire dealer to check and reset the tires' pressure and convergence, he is actually performing proactive maintenance removing the causes of a sub-optimal working condition.

In the context of any system that is dependent on a fluid to carry on its operations, the degree of contamination of the fluid is an essential parameter that leads to failures, since every extraneous particle left in it can reach the working components and increase their natural level of erosion. So if a company has a way to measure the degree of contamination, then the personnel can keep an eye on it. Thus ensuring that when the level becomes dangerous, the correct actions to prevent the system's degradation are promptly performed, like improving filtering or changing the fluid.

The advantage of this strategy is the growing reliability of the whole system added on top of the advantages that predictive maintenance already had. The downside consists of additional costs and knowledge required from the personnel to determine the root causes of faults and sub-optimal working conditions exploiting methods like root cause analysis (RCA), failure modes and effects analysis (FMEA), and fault tree analysis (FTA).

2.5 Reliability Centered Maintenance

Reliability centered maintenance (RCM) has been defined in [22] as "a process used to determine what must be done to ensure that any physical asset continues to do whatever its users want it to do in its present operating context". Essentially it exploits the fact that the system is made of units and it is not mandatory to apply a single maintenance strategy to all of them.

An example of what is meant to be RCM can be found in [15]. In the context of a car maintenance, we already mentioned that the brake pads are under the predictive maintenance strategy, that the tires are under the proactive strategy and the motor's oil is under the preventive one. But we can also say that the light bulbs, that are changed only when they break are under the reactive maintenance strategy. So, taking the car as a whole system we can state that RCM is used as a maintenance strategy.

The advantage of this mixed strategy consists in the fact that it is able to reduce the complexity costs and maintain the reliability and operative costs advantages. The more complex the maintenance strategy used is, the higher the costs to apply it are, so if to a system's unit is applied an advanced strategy and the obtained benefits in terms of production are not enough, a simpler one can be used instead.

Chapter 3

Machine learning

Ever since the beginning of the revolution that brought the computer to be a constant presence in our everyday life, there has always been the idea of making it learn and improve from experience. Such that the man that is recognized as the creator of the computer, Alan Turing wrote in 1950 a seminal paper on the topic [23]. So, over the years researchers drawing on concepts and results from many fields including statistics, artificial intelligence, philosophy, information theory, biology, cognitive science, computational complexity, and control theory [24] were able to create what is now called Machine Learning (ML).

As reported in [25] the existence of algorithms that can exploit experience to improve themselves is a necessity in many scenarios. Having a way to learn from data increases the performance in all the cases where the **knowledge of the problem** is lacking, or too difficult to transfer into a program. An example could be natural language (i.e. language related tasks) where Herdan's law states that "vocabularies' sizes are concave increasing power laws of texts' sizes" [26]. Another scenario is related to the **environment**, in fact when an algorithm is put under tests in a real environment can happen that it doesn't perform well. The cause may be that some properties of the environment can't be measured or that they simply change over time. Also here an example can be found in natural language where new words are invented and rules are changed over time. Finally, there could be the need to **discover hidden structure** in the data.

One of the more common definitions of machine learning algorithm comes from [24] and states "A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".

3.1 Types of learning

Over many years, a wide variety of machine learning algorithms has been discovered to target their specific task, so it is natural to try to put an order with a taxonomy classifying these tasks. At the topmost level of this taxonomy, there are the five main categories that we are going to briefly introduce.

Supervised learning defines all the tasks where the objective is to learn and generalize the map between the input points and the attached labels, whereas to generalize means that the model must select the correct labels even when it has never seen the specific instance. In this case, it is relatively easy to formulate performance metrics to select the best model, e.g. the one that selects the higher number of correct labels. This branch of the taxonomy is usually subdivided into two more: classification where the labels are categorical and from a finite set, and regression where the labels are usually real numbers.

Unsupervised learning is closely related to the density estimation field of statistics. In fact, under this category, the task has at its disposal only the data without labels, and the objective is in general to discover regularities and irregularities in it. To make a few examples of the models we can find under this branch of the taxonomy, we have clustering where the task is to find the similarity between the points given a definition of similarity, then we have the so-called latent variable learning like independent component analysis (ICA), principal component analysis (PCA), etc. And at last, we have anomaly detection that will be explained in Section 3.2.

Self-supervised learning instead uses the data itself as labels, at its core the task here is to mask a part of the input data and then teach the model to reconstruct it. It is very common in fields where there is an abundance of unlabeled data that has some temporal or spatial relation to be learned from the model. Like natural language or computer vision.

Semi-supervised learning is a simple layer of supervised on top of unsupervised or self-supervised, this allows to obtain a regressor or a classifier but with far fewer labels than with pure supervised. It is usually used where the labeling process is expensive, like again in natural language where for example, the task of named entity recognition (NER) is done with a small classifier on top of a self-supervised model (BERT, GPT-3, etc.).

Finally, **Reinforcement learning** models learn by interacting with the environment or a simulation of it. Each action is associated with a reward and the objective is to obtain the highest expected reward with the actions done.

3.2 Anomaly detection

"Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior" is the definition given by [27]. This task has ever gained its fair share of attention from the scientific community, such that important publications on this topic date back to the 19^{th} century [28, 29], and nowadays the developed solutions find applications in a vast range of fields like fraud detection [30], bot detection [31] and industry 4.0 [32].

Anomaly detection predominantly makes use of the unsupervised learning paradigms since it exploits the assumption that the vast majority of the points in the dataset are normal and only a few are to be considered outliers. Most of the times this assumption is satisfied, but papers regarding supervised [33], semi-supervised [34] or even self-supervised [35] are not missing from the literature. The main categories of the algorithms used for anomaly detection are:

- **Statistical methods** were the first types of algorithms that have been applied since the problem was born in the field of statistics. They learn the probability distribution function (PDF) of the normal data and if a point has low PDF then it's an anomaly. Examples can be the evergreen kernel density estimation (KDE) [36, 37], or the more recent ECOD [38].

- **Proximity-based methods** come right after for their simplicity. They exploit the fact that a point distance-wise isolated from the others is probably an outlier if the training set is representative. Examples can be the local outlier factor (LOF) [39], which is almost always taken into account in publications at worst as a reference, or the very recent Rotation-based Outlier Detection (ROD) [40].

- **Machine learning models** can be also used to do anomaly detection, the most common examples are one-class support vector machines (OCSVM), isolation forests, and neural networks that will be explored in detail in this work.

- **Ensembles** of models are born from the necessity of recognizing different types of anomalies. In fact, by an ensemble, we mean using multiple models and then coordinating their results in a single one, where each model can specialize itself on a specific type of anomaly. Examples could be the famous XGBOD [41] or Kit-Net that has been used in this work and will be introduced in Section 3.4.

3.3 Model Interpretability

In some cases it is not enough to trust and rely on the result from a machine learning model: this typically happens when the decision that arises from that result has a big impact on terms of risks, costs, income, or safety. For example, think of a model that evaluates whether to give a loan to someone or not, since in machine learning racial bias is not rare [42], it may happen that underrepresented groups are systematically penalized. To avoid this kind of situation, the need arises to be able to tell how the inputs influence the output (interpretability) and how the parameters justify the result of the model (explainability). While the related literature is still debating the definition of the two terms in the context of XAI, in the following we will use the two terms interchangeably.

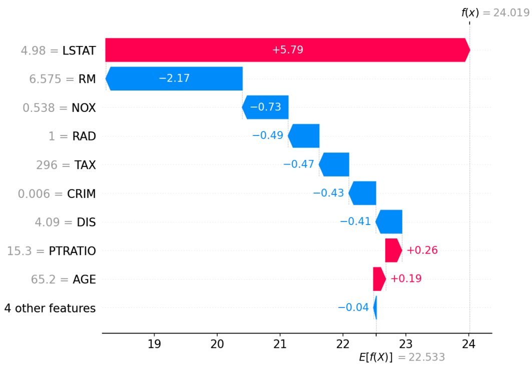
There are also some cases where interpretability is counterproductive as explained in [5]. For example, if the aforementioned model for granting loans can be interpreted by the customers, then strange conditions may arise in which it is convenient to return a credit card since for example the model is known to penalize people with more than two.

In general, interpretability can be used to test the model in the way we report directly from [5] (originally from [43]):

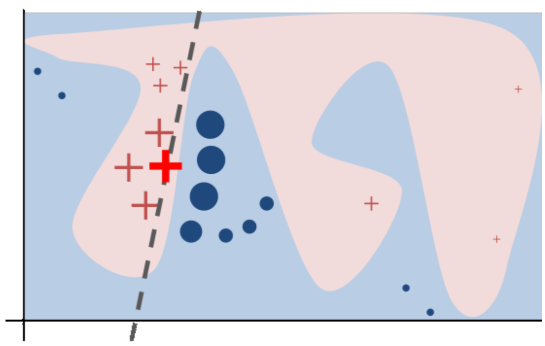
- Fairness: Ensuring that predictions are unbiased and do not implicitly or explicitly discriminate against underrepresented groups. An interpretable model can tell you why it has decided that a certain person should not get a loan, and it becomes easier for a human to judge whether the decision is based on a learned demographic (e.g. racial) bias.
- Privacy: Ensuring that sensitive information in the data is protected.
- Reliability or Robustness: Ensuring that small changes in the input do not lead to large changes in the prediction.
- Causality: Check that only causal relationships are picked up.
- Trust: It is easier for humans to trust a system that explains its decisions compared to a black box.

With linear models, it is easy to understand what pushes the model toward its decision since there is a linear relationship between input and output, but with more complex models like a neural network, the task becomes quite difficult. We can have interpretability algorithms that are specialized for a certain type of model since they exploit its structure or properties, like for example the already motioned ones for isolation forest [44] or [45], or we can have the so-called model-agnostic algorithms that work independently on the model they are analyzing. In this last category, we can find the famous Local Interpretable Model-Agnostic Explanations (LIME) [46], in which essentially the single point is perturbed and a linear model is built (Figure 3.1b) to classify the perturbations weighted by their similarity with the original. The downside of LIME is that it isn't fast enough to be applied in many scenarios and that the explanations are local. Another very recent algorithm is SHapley values Additive exPlanations (SHAP) [6] that is the one used in this work. It uses the same principles as LIME, the perturbations of the instance to explain, but then Shapely values are computed [47] (a concept from game theory). Its implementation is faster, and the explanations are global and additive (Figure 6.31), so we can see how much a given input has contributed to the model output both in negative and positive ways.

The innovations in the field of model interpretability happen quickly since there is a lot of interest in it. Suffice to say [48] reported that the collaboration of models and humans can achieve better results than when they operate alone. To encourage cooperation, interpretability is necessary.



(a) SHAP's additive explanations, their sum is $f(x)$ on the top right corner [49].



(b) LIME's local linear model. The bigger cross is the positive point to explain [50].

Figure 3.1: Examples of representations of the interpretability algorithms.

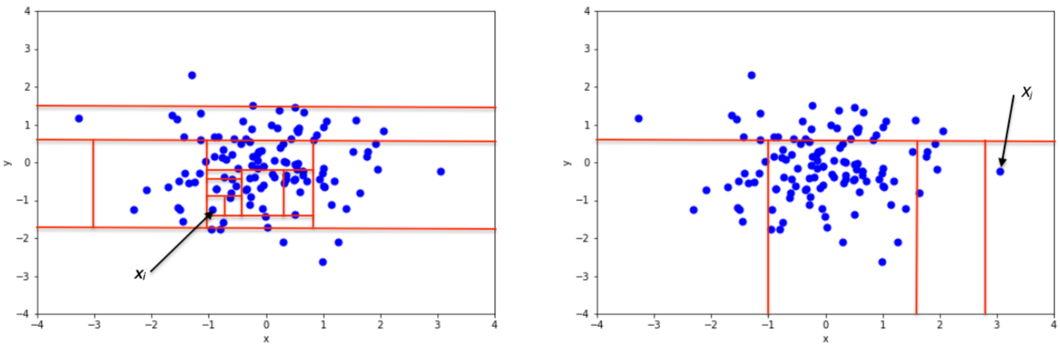
3.4 Models

3.4.1 IsolationForest

Isolation forest was first proposed in [51], despite being a relatively new algorithm it is one of the most used algorithms. Its best qualities are effectiveness, speed, low memory usage, and the numerous specialized algorithms for feature selection and prediction explanation that have been developed over the years ([44] or [45]).

As the name says, the base of this algorithm is an ensemble of decision trees, where each tree takes care of a subset of features and its training set is a sample of the original one (bagging to reduce the model's variance). The process used to build the trees is to add a node, that is a random threshold cut for a random feature until each point in the training set is isolated in a tree's leaf. One can understand that an anomalous point would likely be in a less crowded position and so few cuts are needed to isolate it. In Figure 3.2 we can see the path to a leaf containing the isolated point, since the points have two features the cuts are only horizontal and vertical. Then to give the anomaly score to the input point, the thresholds are executed from the root to a leaf for each of the trees and the bigger the average depth it reaches, the less anomalous the input is.

Now it's easy to see why it is extremely fast and has low memory usage, since all we need to do is to store a series of values that define the cuts, and the execution time is a series of if-then-else which size is kept under control by limits on the number of points per tree, number of trees and their maximum depth. As a side note, we report that for convergence reasons the author defined some thresholds for the aforementioned hyper-parameters, these values are usually used as default by the methods that one can use.



(a) For an anomalous point a few cuts are needed

(b) For a non-anomalous point a lot of cuts are needed

Figure 3.2: Process to isolate an anomalous and a non-anomalous point, the two images are respectively taken from [52] and [53]

3.4.2 Auto-Encoder

Auto-Encoders are a particular type of neural network that is trained to reconstruct its input. The formal definition of the problem has been given in [54] such as to learn two functions, the so-called **encoder** $A : \mathbb{R}^p \to \mathbb{R}^d$ and **decoder** $B : \mathbb{R}^d \to \mathbb{R}^p$ that satisfy 3.1:

$$\underset{A,B}{\operatorname{argmin}} \ \underset{x \in \mathbb{X}}{\mathbb{E}}[\Delta(x, [B \circ A](x))] \tag{3.1}$$

Where Δ is the reconstruction loss function, that usually is the root mean squared errors (RMSE) between input and output.

Auto-Encoders are a relatively mature technology since as many of the ideas at the base of ML, they were published in the '80 [55] and then they exploded in popularity and effectiveness when both the computation capability and availability of data exploded. But noticeable variations were proposed in recent years like the 2013 Variational auto-encoders [56].

The common way they are used is to learn a representation of the input as represented by the code section in Figure 3.3 since auto-encoders can be seen as a generalization of principal component analysis (PCA) [57]. Or in a way that is more suited to anomaly detection, in facts equation 3.1 can be exploited to understand that outliers having less probability to appear in the input may have a greater reconstruction error. At least, if we in some way restrain the power of the NN with dropout and regularization.

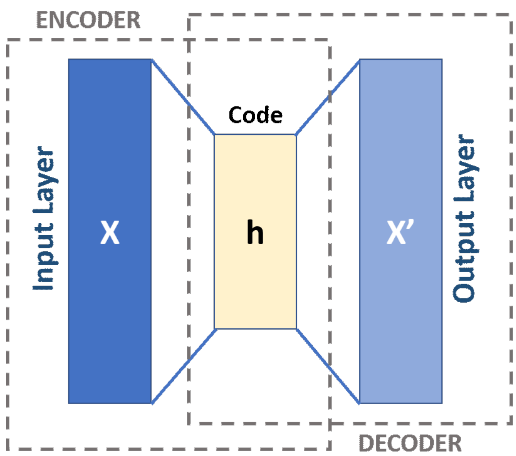


Figure 3.3: Schema of a generic auto-encoder from [58]

3.4.3 DeepSVDD

Deep Support Vector Data Description (Deep SVDD) [59] is a recent algorithm introduced as a NN model that is trained specifically with an anomaly detection related objective. Other models use a side effect of the NN as anomaly score. To make an example auto-encoders are trained to reproduce their input, and statistically, since outliers are rare the reconstruction error is large.

The theory at its foundations there is the old gold Kernel-based One-Class Classification, the same used by One-Class Support Vector Machines. The innovation DeepSVDD brings is the fact that the kernel is approximated by the neural network while minimizing the volume of the hyper-sphere containing all the projected training points as depicted in Figure 3.4.
Given that once the optimal matrix of weights W^* is found the anomaly score for the input x_i is $\|\phi(x_i; W_*) - c\|^2$, then the loss function to train DeepSVDD is:

$$\min_{R,\mathbb{W}} R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max\{0, \|\phi(x_i; \mathbb{W}) - c\|^2 - R^2\} + \lambda^2 \sum_{l=1}^{L} \left\|W^l\right\|_F^2 \tag{3.2}$$

Where:

- The first term minimizes the radius R of the hyper-sphere with center c in the co-domain of the map.
- The second term is a penalty to admit some points outside the hyper-sphere (soft boundary). Has ν as hyper-parameter to set the trade-off between the number of violations and volume of the hyper-sphere and $\mathbb{W} = \{W^l : l = 1..L\}$ as set of the weights of the NN.
- The third term is a regularization term.

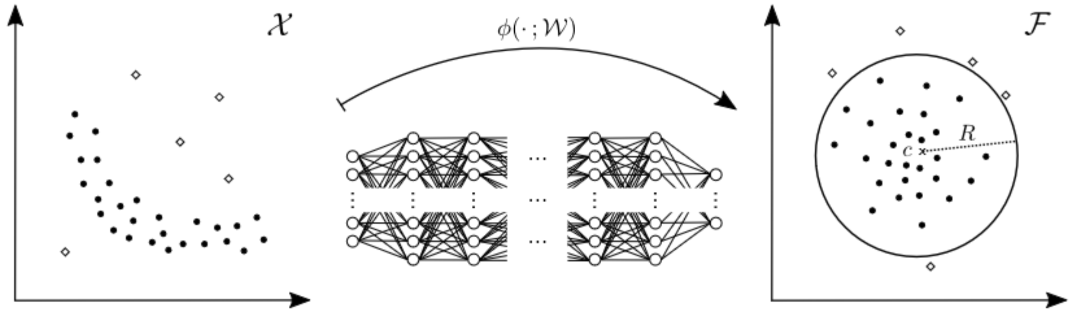


Figure 3.4: Illustration of DeepSVDD's objective directly from the original paper [59]

3.4.4 KitNet

KitNet is a Python implementation of Kitsune, that was first developed as part of a network intrusion detection system [60].

The structure of the anomaly detection part of Kitsune can be seen in Figure 3.5. The first component is an ensemble of small auto-encoders (maximum three layers with at most seven visible neurons) like the ones we explained in Section 3.4.2, their objective is to compute RMSE given the features they receive in input. Then an additional auto-encoder (in Figure 3.5 called output layer) is used as a non-linear voting system to extract the final anomaly score. Before them, there is an optional feature mapper whose objective is to learn an efficient map of the input features for each of the models in the ensemble and then to forward the transformed inputs to them.

This algorithm was built to be fast and to be executed in a streaming fashion. For this reason, it has two working modes: **training** that is used to adjust the weights of the anomaly detection part, in which a single iteration of error back-propagation for each of the input points is executed until a fixed amount of them is analyzed, optionally the same can be done for the feature mapper. Then the model switches to **execute** and stops to update its weights only returning the anomaly score.

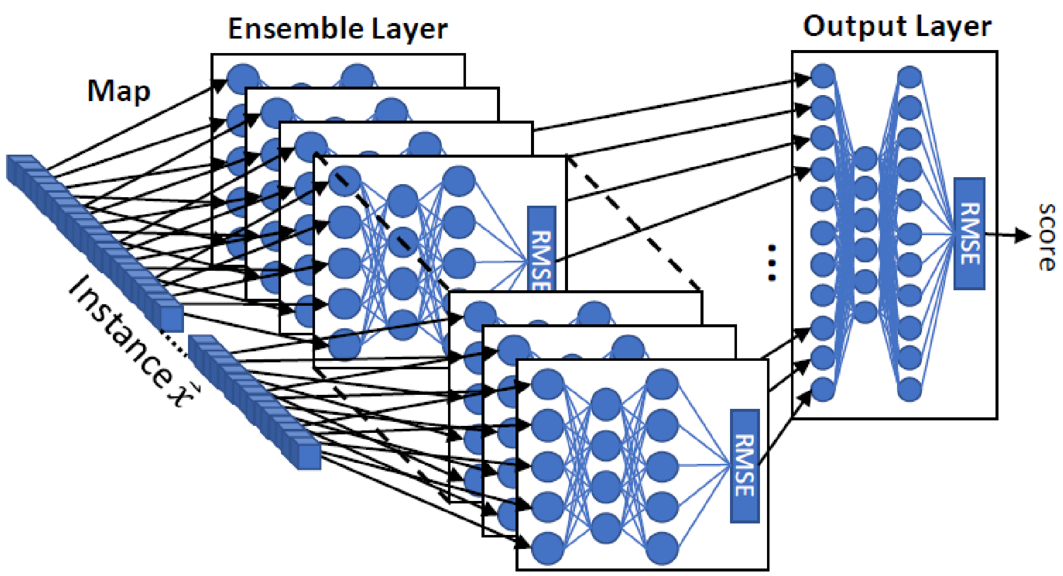


Figure 3.5: Illustration of KitNet's Architecture directly from the original paper [60].

3.4.5 RSHash

Randomized Subspace Hashing algorithm (RSHash) [61] is another algorithm that is suited to work with streaming data like KitNet in Section 3.4.4 and so is designed to be fast and with low memory usage.

This model uses an ensemble of size m, and each of the simpler models is composed of w hash tables. Here we report a simplified explanation of how it works since the implementation details can be found in the original paper.

The first thing that is done is to select a subset of the training, a subset of the features, and some randomized constants that are used in the very next operation for each of the models in the ensemble. Then the points X_i sampled from the training set are min-max normalized into X_i' and transformed into Y_i with:

$$y_{ij} = \begin{cases} -1 & \text{if feature j not selected} \\ \lfloor \frac{x'_{ij}+\alpha_j}{f} \rfloor & \text{otherwise} \end{cases} \tag{3.3}$$

Then w hash tables are updated, in more detail at each hash table H_k is associated with a different hash function h_k built in a particular way explained in the original paper [61] and each point Y_i increments the length of the list in cell $H_k(h_k(Y_i))$ by one.

At the end suppose a point X_t to score and $c_k = len(H_k(h_k(Y_i)))$, take the $log_2(\min\{c_1, .., c_w\})$ if the point were not in the training or $log_2(\min\{c_1, .., c_w\}+1)$ otherwise for overfitting reasons and to treat all the points the same way (if a point is in the training set the minimum returns at least one) and the score is simply the average value from the ensemble's models. It's easy to understand that an anomalous value would obtain a low count in the hash table and so a low score.

Chapter 4

Hydroelectric power plant

A hydroelectric power plant is a very complex system that converts water energy into electric energy. Being artificial intelligence the core subject of this work, the following is to be intended as a fast overview of how hydroelectric power plants work with the objective to both introduce the topic to readers from the artificial intelligence field and to explain the main features used by the models with the fault they are related with.

As reported in [62] the broad variety of natural conditions that allow the exploitation to produce hydroelectric energy makes it possible to have a range of specialized types of power plants:

- **Run of river** type of power plant (Figure 4.3) is built to exploit the water flow of a river. It can receive directly the water if it is built as a dam or a small penstock upon the river, to avoid generating changes to its flow, or can receive water that is redirected from the river into the power plant using a long penstock. Using this power plant we obtain a constant production of energy but it is tightly coupled with the flow of the river, so the generated energy can be affected by seasonal fluctuations.

- **Storage regulation** type of power plant (Figures 1.1 and 4.1) is built to exploit a large reservoir of water (or multiple ones). It accumulates the water in the storage during high water inflow periods and uses it when the flow drops, doing this it remains mostly independent of the seasonality of the flow. We usually find that the reservoir is placed at a higher altitude than the power plant and so they can exploit whichever type the potential energy it is converted to (both pressure and kinetic energy can be exploited), thus generating way more power than the other types of plant.

- **Storage pumping** type of power plant is built to exploit the property of the Francis turbine to also work as a water pump, so it is essentially a storage regulation that can pump the water back into the reservoir when the energy has a low price to exploit it later when the price is high.

- **Tidal** type of power plant (Figure 4.2) is built to exploit the daily tide to lead the water through the turbines four times a day (two high tides and two low tides a day).

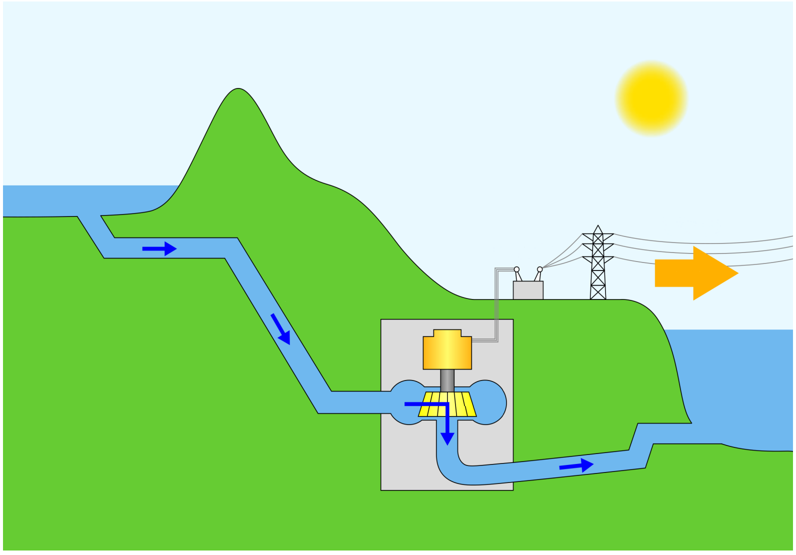


Figure 4.1: Sketch of a storage regulation hydroelectric power plant [63]

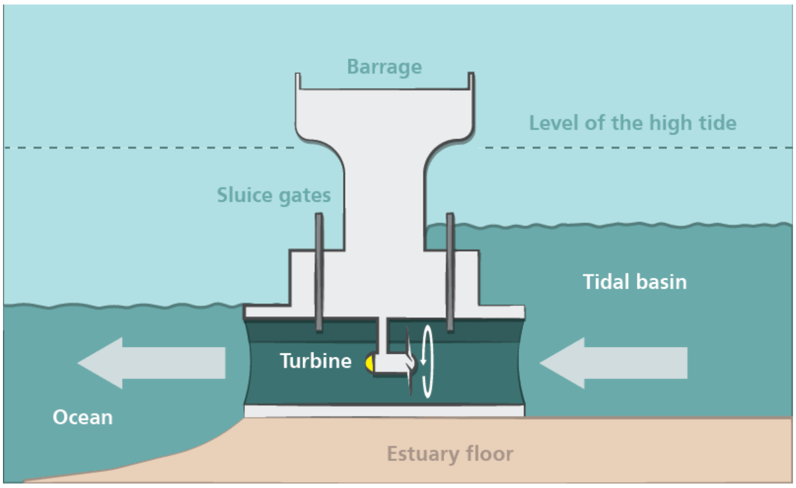


Figure 4.2: Example of a tidal type of hydroelectric plant [64]

Figure 4.3: Example of a run-of-river type of hydroelectric plant [65].

4.1 How it works

The core of how a hydroelectric power plant works is the conversion of the potential energy of the water uphill to electric energy using a turbine connected to an electric generator (yellow elements in the sketch).

The law describing the energy produced by the unit is described in [62] and here it is rewritten in the form that is more commonly used.

$$P(kW) = \frac{\eta \rho g Q(m^3/s) H(m)}{1000} \tag{4.1}$$

Where:

- η is the unit efficiency
- ρ is the density of the liquid used (mass of matter contained by a unit volume)
- g is the gravitational acceleration.
- Q is the flow through the turbine, i.e. the volume of water that flows in it for each unit of time.
- H is the net head seen from the turbine, that is the difference in altitude between the uphill and downhill basins (head) minus the energy losses that occur in the penstock, essentially it's a measure of the potential energy the turbine finds available to exploit.

4.2 Operating point

The behavior of the machine at any moment, i.e."what the machine is doing", is normally called the operating state. More specifically, when the unit is synchronized i.e. actually delivers power to the grid, the combination of parameters in Equation 4.1 is called the operating point. Because of the interdependencies we will describe later, not all these parameters are linearly independent. In the end, to describe the operating point in our case, the following variables are considered: rotation speed (only needed to discriminate some transient operating states such as run up and coast down), power, and wicket gates opening. For simplicity, the net head was considered constant.

A first relation is the one that essentially allows us to not look too closely at the speed. The machine we dealt with in the present work, like the vast majority of the ones used in hydroelectric power plants, is of the synchronous type, meaning the rotation speed and electric frequency of the grid are strictly linked to one another through the number of generator poles, as long as the generator itself is connected to the grid via the main circuit breaker (the case we considered here). Of course, during the startup and shutdown sequences, the unit is disconnected (the main circuit breaker is opened). During startup, the unit acceleration (run-up) is controlled by the turbine speed governor until the rotation speed corresponds to the grid frequency. Then the rotor windings get energized by the exciter. A device called synchronizer operates with the exciter to match the phase of the generator output with the grid so that the main circuit breaker can be closed without excessive electrical unbalances. From that moment on, the unit speed rigidly obeys to the grid itself, and the turbine governor switches to power control mode.

Other relations that occur in Equation 4.1 are that g is constant in the location, ρ depends on the sediments concentration in the water, but can normally be considered constant as well. Then H depends on Q because the energy losses are proportional to the square of the water velocity v, and since we are analyzing a Francis turbine (it works the same also with Kaplan turbines) the flow Q is in turn regulated by the distributor opening, also called wicket gates opening. Finally, the efficiency also depends both on Q and H.

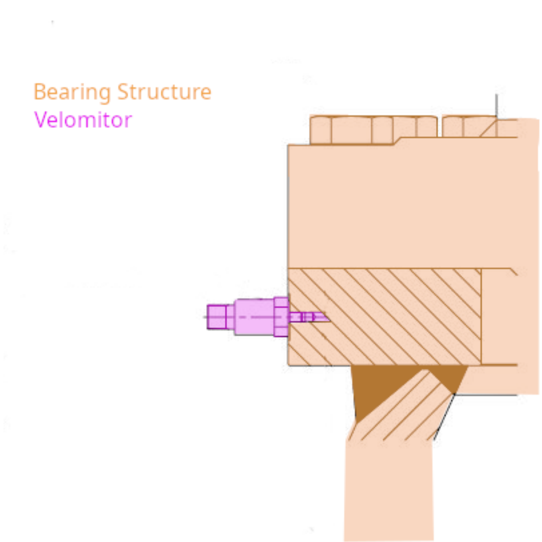
4.3 Vibration analysis

In physics, with the term vibration is meant oscillations of a system about an equilibrium position as written in [66]. Vibration analysis is one of the most important, if not the most important type of analysis that machinery is subject to, especially when rotors are involved. From one side the vibrations themselves are a danger to the system since as [67] reports: machine vibration can accelerate rates of wear (i.e. reduce bearing life) and damage equipment. It can create noise, cause safety problems, lead to degradation in unit working conditions and cause machinery to consume excessive power, thus damaging product quality. In the worst cases, vibration can damage equipment so severely as to knock it out of service and halt plant production. While from the other side as we will see in Section 4.5, in the vibration is contained important information about the state of the machine and the problems that afflict it.

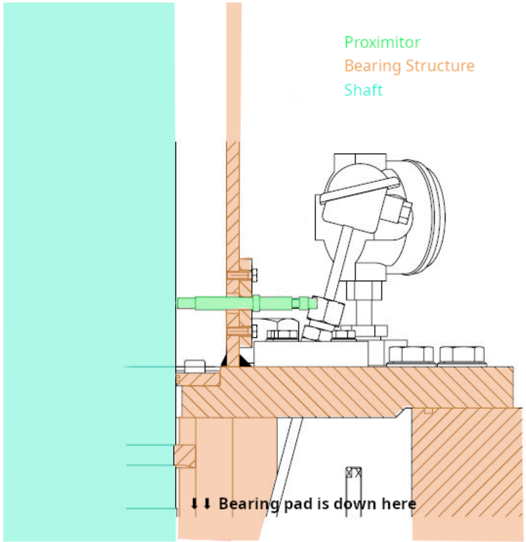
To acquire the signals about the machine vibration, we need to place some sensors. Generally, the type that is used depends on the property of the vibration to be measured as explained in section 4.4. But keeping in mind that the sensors measurements are subject to noise, it's a better idea to use the right type of sensor (accelerometer for acceleration measures) or at most to derive the signal (derive velomitor signal to have an acceleration measure), since integrals add up the signal's noise while contributing with numerical errors.

In Figure 4.4b we can see an example of where a proximitor can be placed to measure the relative displacement that vibration generates between the shaft and the bearing. The sensor is placed integrally with the bearing structure pointing towards the shaft. Ideally, its optimal position should be in the center of the bearing pads, to be able to obtain the greatest precision but to do this a hole in the pad would be necessary and that would destroy the film of oil that keeps the bearing working. In Figure 4.4a we can see an example of where a velomitor can be placed to measure the absolute vibration's velocity of the bearing. Velomitors and accelerometers are less cumbersome to be placed since being absolute measures they don't need two reference points but only one.

Finally, one important thing is that the sensors' frequency can reach the order of kHz. This means that the full signal cannot be continuously recorded for extended periods of time, let alone extracting more informative features like the ones in sections 4.4, 4.5 and 4.6. So a good trade-off that is usually applied is to extract those features only once for each window of signals that are considered informative, like in case of surpassed thresholds, fast variations, etc.



(a) General position of the velomitor that measures the absolute velocity of the bearing structure vibration [12].



(b) General position of the proximitor that measures the relative displacement between the bearing pads and the shaft surface[12].

Figure 4.4

4.4 Time domain analysis

The analysis in the time domain of the signals from the plant is the basic form of analysis done to assert that the measures associated with the plant's RCM continue to satisfy their constraints.

In the context of vibration analysis, we have two types of information we would like to take into account depending on the nature of the signal. For a relative vibration signal, we would like to know the extent of its range, since we use it to assert the rotating parts are not touching the stationary ones. So, one feature that is well suited to this use is **peak to peak**, which is the maximum range the signal has in a window. For an absolute vibration signal, we would like to know how much energy is involved since it is a good estimation of the stresses the system has undergone to. In this case, **RMS** (root mean square) is the most suited feature. Then there is the **crest factor**, which is the zero peak, i.e. the maximum amplitude the signal has in a window, divided by the RMS in the same window. It's used to reveal whether the amplitude of the high-frequency component of the vibration increases compared to the amplitude of overall broadband vibration. One of its qualities is well suited for variable speed machinery because both zero peak and RMS will increase as speed increases [68].

4.5 Frequency domain analysis

The analysis in the frequency domain (spectral analysis) is useful to highlight the various components that participate in creating the overall vibration. Then, the combination of theory and expertise matches these components to the characteristics of the system, each representing a particular phenomenon that can evolve into potential faults.

Usually, in dealing with rotors' vibrations, the frequency components are expressed in terms of the so-called harmonics: the rotation frequency is considered the fundamental frequency of the system or "order 1 harmonic", then the other frequencies are named after it, like in Figure 5.3. This approach is especially helpful when the rotation speed is not fixed, like in run-up or coast-down.

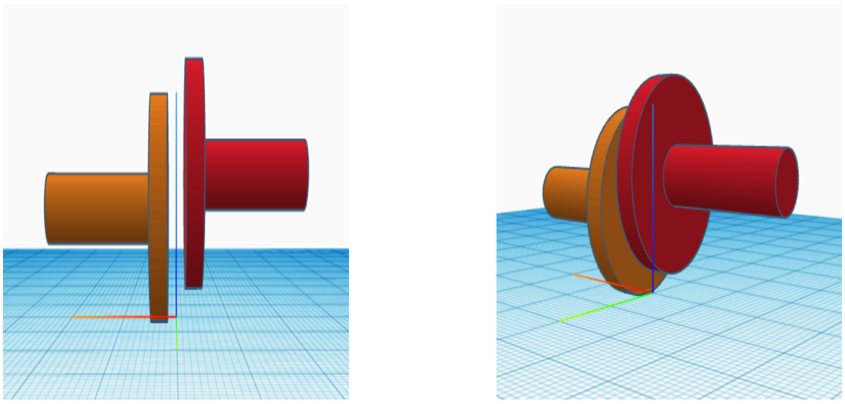
One of the most important problems that can be identified from the power spectrum is the presence of an unbalance in the weight distribution, i.e. a center of mass that is not aligned with the shaft axis. The causes can be both an uneven construction material or as we will see later a radial misalignment between the shaft components. From the literature [69] we know it is linked with a peak on the synchronous component frequency of the spectrum, that is the frequency at which the shaft rotates.

Another problem that can be identified from the power spectrum is the presence of misalignment between the components of the shaft. The shaft of a hydroelectric power plant is an extremely long and heavy element of the system, which usually is built in several pieces that are flanged and bolted together with special rods. But although the personnel has specific methods to connect them in a way to minimize the misalignment, there are always some imperfections in both construction and assembly. So the shaft can be subjected to:

- **Radial misalignment** also called offset or parallel offset misalignment, is an offset between the components axis as can be seen in Figure 4.5. From the literature [70], it generates an increase in the synchronous component of the power spectrum (order 1 harmonics, the principal frequency of the system) depending on the entity of the misalignment.

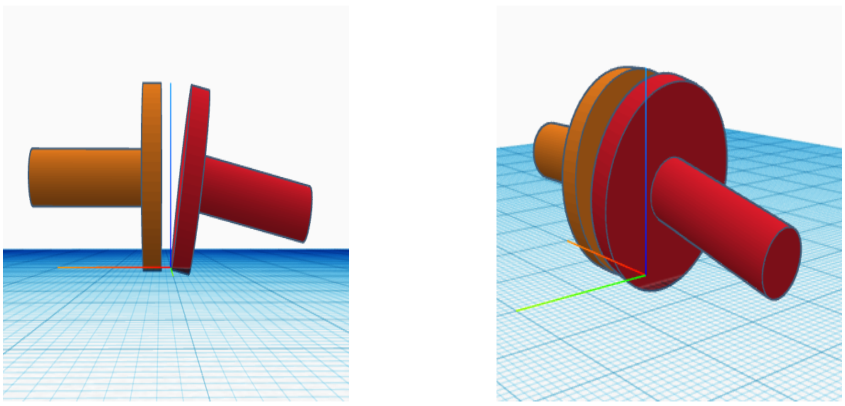
- **Angular misalignment** as can be seen from Figure 4.6, is an angle between the components axis. From the literature [70], it generates an increase in both the synchronous component or the 2 order harmonics (double the fundamental frequency of the system) depending on the entity and configuration of the misalignment.

Finally, the last problem that we report here is a specific effect that affects hydroelectric power plants equipped with Francis turbines. In literature is known as **vortex rope**, and is a cavitation effect generated close to the outlet center of the runner with the shape of a vortex, that as stated in [71] has the consequence of inducing strong pressure pulsation, axial and radial forces and torque fluctuation as well as turbine structure vibration. From the literature [72] we know that the vortex is predominant in the operating condition called **part load**, that depending on the machine design means a discharge Q between 45 and 60% of the Q_{BEP} or best efficiency point discharge (the turbine achieves its highest efficiency with that flow) and that it's principal harmonic of the vortex rope effect can be seen around 20-25% of the principal frequency of the system, then it has also some others relevant tail harmonics around 50% of the principal frequency of the system.



(a) Side view **(b)** Back view

Figure 4.5: Example of parallel misalignment of the shaft components



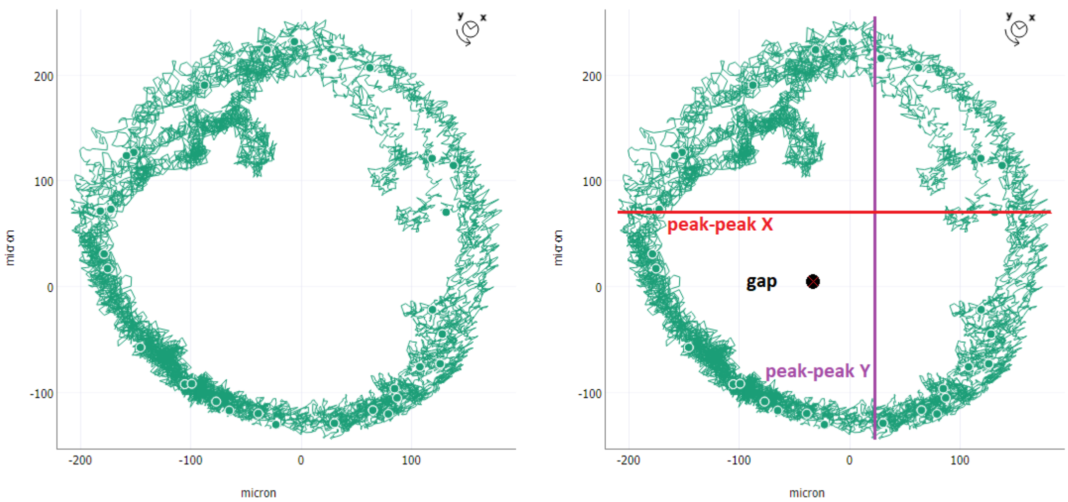
(a) Side view **(b)** Back view

Figure 4.6: Example of angular misalignment of the shaft components

4.6 Orbit analysis

The orbit that the shaft travels inside the bearings is a fundamental entity to be kept under control since the **constructive gap** (i.e. the space inside the bearing) is the limit the orbit can take without dealing damage to the system.

With this objective, two sensors are placed to measure on two axes the relative displacement of the shaft from the bearing pads (the bearings work with an oil film that with the rotation of the shaft itself keeps it at some hundredths of [mm] from the surface of the closest pad in the load direction). Then from these raw signals as we will see in section 5.2.2 we can both extract time domain and visual features with algorithms going through the reconstruction of the orbit as we can see in Figure 4.7a



(a) Orbit reconstructed from shaft X and Y relative displacement from the bearing [12].

(b) Time domain features extracted from the orbit raw signal from the sensors [12].

Figure 4.7

Chapter 5

Case Study

At this point the reader has been introduced to all the theoretical concepts that are related to this work, so now it's time to delineate the context we worked in, what were our objectives, and how we act to achieve such goals.

Independently from the rest, we first want to give a fast overview about the state of the art approaches, i.e. those that currently are used to apply predictive maintenance in industrial setups. The vast multiplicity of methods that can be used confirms that in each field of application the properties of the process and subsequently of the data that comes from it are such that some models perform and some do not. Thus justifying the effort of this work. After that, we will introduce the structure of the data we had at our disposal in both its qualities and defects along with the needs that have to be satisfied. An important thing to keep in mind is that thanks to our collaboration with Andritz Hydro, we were able to work on the data of a real plant. Finally, we will explain the choices we made in terms of technologies, operations on the data, and experiments.

5.1 State of the art

Over the years, many methods have been developed to execute data-driven predictive maintenance. We talk about predictive maintenance since then proactive or RCM can be reached by adding independent technologies like RCA. From the literature, we can see that all the approaches currently considered state of the art can be categorized into remaining useful life (RUL) or risk estimation and highly multivariate analysis.

5.1.1 Remaining Useful Life & Risk estimation

The objective of these types of approaches is to forecast the future of the system and to tell respectively: how much time is expected to last the system until the next break or what is the probability that the system will break in a time horizon from now like can be seen in Figure 5.1.

Approaches of this kind are generally bottom-up [73], in the sense that their scope is contained to simple components of the system or single features. Then an additional system is needed to propagate the results to the overall system [74]. The advantage with respect to the other approach is that it makes it easier to organize the maintenance operations since there is literally a count down to the fault. But on the other side being the task extremely demanding in terms of complexity and given the fact that the data is usually scarce and poor in quality, the models may suffer a loss of precision that propagates and stacks up along the system.

Talking about the real implementation, we have a wide range of possible algorithms that we can use: from the historical dynamical models of the system that use the finite element method (FEM) or experimental modal analysis (EMA) like [75], to the use of self-organizing maps (SOM) and back-propagation networks like in [76]. Among the many other options also using recursive neural network (RNN) or neural Fuzzy (NF) is an option [77].

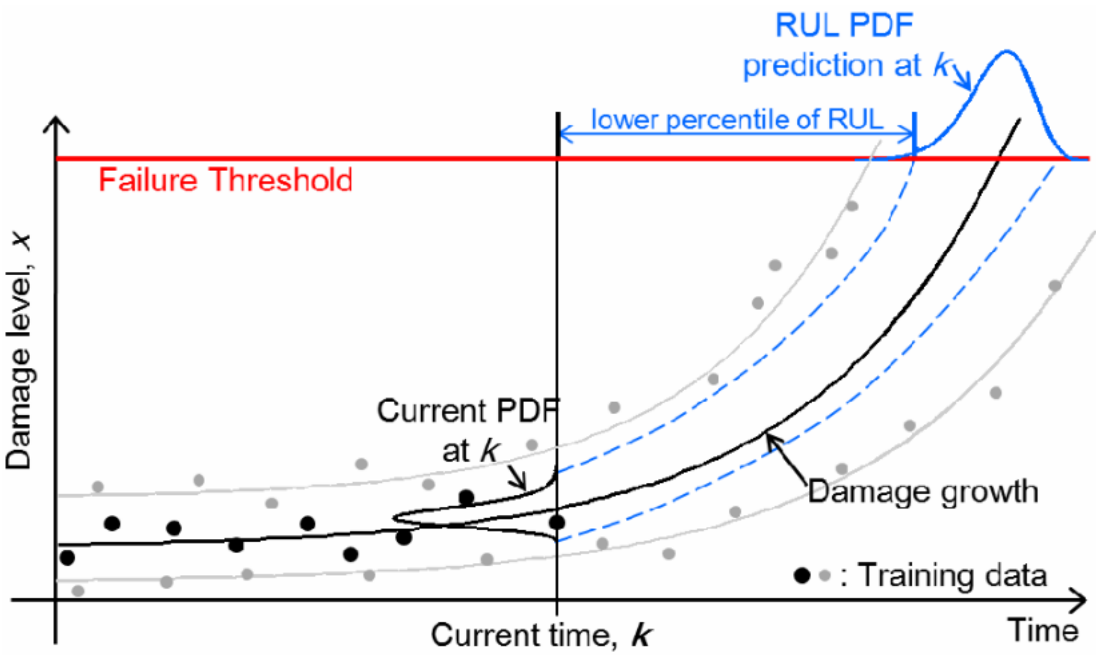


Figure 5.1: Example of the plot of the results from an RUL approach [78].

5.1.2 Highly Multivariate Analysis

Contrarily to Risk estimation, highly multivariate analysis is a top-down approach, in the sense that specialized models are used to detect anomalies using the whole system information like in Figure 5.2. Then a higher level system can be used for the attribution of responsibility to the single features.

The advantage of this approach is that the faults are analyzed by learning the correlation between the features from the data itself, instead of using literature to build the high-level system used in RUL. So it is possible to catch also very complex faults that depend on a high number of variables [79]. But the downside is that the connection with the timing of the maintenance has to be imposed with strategies like [80], even if degradation trends can be easily spotted since they induce a variation of the distribution of the features and on the relations among them.

For the implementation, the most widely used model is isolation forest or one of its variants [81]. But also the aforementioned self-organizing maps [82] or neural network-like models in the form of auto-encoders [83] can be applied.

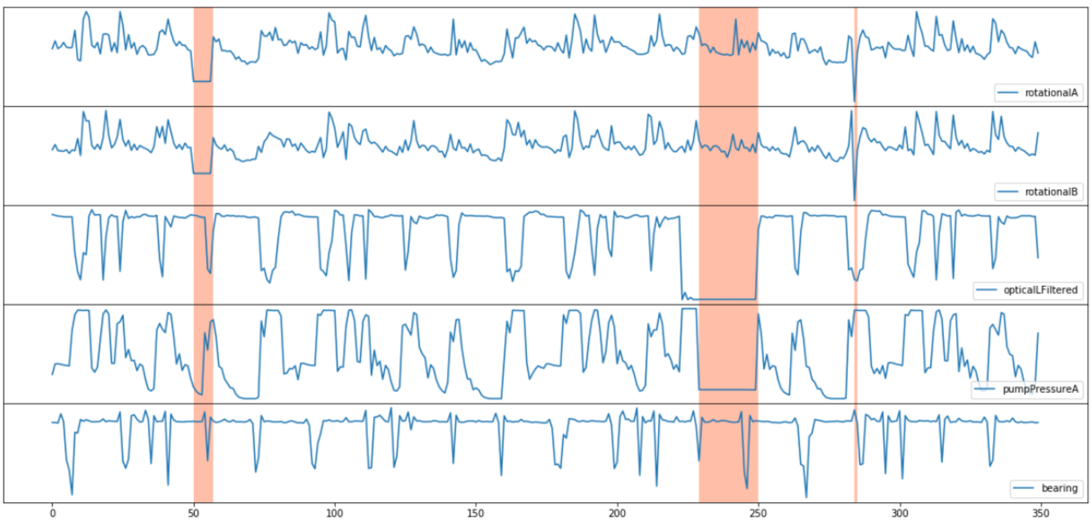


Figure 5.2: Example of the plot of the results from a Multivariate approach [84]. Notice how the anomaly is detected from all the signals jointly.

5.2 Context

5.2.1 Data

As already anticipated, we worked on data from a real hydroelectric power plant. If on one hand, this is to be considered a virtue since the chances an algorithm that works on synthetic data also works in production are far lower, on the other hand, requires extra care in the way the experiment is carried on. So, we will list in the following all the critical points related to the data that ones must keep in mind.

Data scarcity: Unfortunately, even if the flow of data coming from the system is of a considerable size, the generation monitoring system it comes from has been commissioned very recently (January 2022), so we were only able to obtain a modest size dataset on the order of 60000 samples. This aspect is very common in the field of hydroelectric since investments to push for digitization have happened only in the last years. As a reference, in 2018 the project [85] started to digitize Norwegian-Swedish hydroelectric power plants.

Unbalanced Data: In industrial scenarios, it is almost always the case that the system's units have a very long mean time between failures (MTBF), i.e. the system works very seldom it has anomalies. This is particularly true for hydroelectric machinery, because of the strategic importance of the energy production field which drives a design and construction that are extremely robust. For this reason, to have at least some anomalous period to take as a reference when comparing the models, we deliberately took the data from a unit that is known by the company to have gone through some maintenance operations and anomalous conditions during its life.

Unlabeled Data: Like it regularly happens in industry 4.0 and even more often in the context of hydroelectric power plants, the process of data labeling is extremely expensive. For some subtle anomalies, it takes quite some time for multiple experts of the domain to recognize them and individuate their causes. So, given the quantity of data the system produces and the dimensionality of the problem it is not reasonable to have a complete and certain point-wise labeling of the data.

High-dimensional multivariate Data: Since the system is very complex, the resulting stream of data is very high-dimensional with 200 features for each machine. On top of this, multiple aspects co-occur in the generation of the machine's vibrations, so it requires a multivariate analysis to catch all the anomalies that are not related to a unique variable. Obviously being designed for human understanding the features are not linearly independent and so they carry some redundant information. Overall, except for the two features related to the high-frequency content of the power spectrum, that for an error on the data acquisition system were constantly zero, each one has its relations with specific causes of the anomalies like we will see in Section 5.2.2.

Seasonality: Some effects like the vortex rope are dominant or negligible depending on the operating point as defined in section 4.2. For this reason, the companies in the field of hydroelectricity define the so-called operating states (OS), which is a partition of the operating range, in such a way that the system's behavior is expected to be stable inside one of them. But each of these states doesn't appear uniformly in the dataset because hydroelectric power plants are well known to often operate under significant seasonality. When there is a lot of water in the basin the plant is required to operate stably close to its full capacity. From the point of view of the dataset being the power stable we almost always find the unit in the High power operative state. On the other hand, when the water availability becomes lower, the power is modulated between the Units (so to keep a reserve in case of sudden need from the network, frequency control) to accommodate a continuous balance between the incoming flow, the residual capacity of the reservoir, and the grid's needs. As a consequence, the Unit ranges over all the operating states. Finally, some conditions like run-up and coast-down don't represent operating points, rather transient states are needed to reach the operation or recover from it. From Table 5.1 we can see that some OS are not as represented as the others and so they will be recognized as anomalies by the models.

Operative states	Number of samples
High power	31174
Low power	30140
Run up	2290
Coast down	1432
Others	4107

Table 5.1: Table with the number of samples for each operative state.

5.2.2 Features

In the following, we will extend the explanation started in Section 5.2.1 about the data listing all the features that the model can use. At the same time, we draw the connections each one has with the problems seen in Chapter 4.

speed, **wicked_gate_opening** and **power** are the features that describe the operating point explained in Section 4.2.

1x_mag_bearing and **2x_mag_bearing** for both x and y directions, it is the power spectrum of the first and second harmonics and is related to the shaft unbalance and misalignment problems explained in Section 4.5.

1x_ph_bearing for both x and y directions, it is the phase of the spectrum in the fundamental frequency and it's mainly used to infer the shape of the shaft's orbit influenced by pure mechanical factors like imbalance and misalignment.

vortex_bearing for both x and y directions, it is the power spectrum in the frequency the first harmonic of the vortex rope effect explained in Section 4.5 shows its effect.

subsynchronous_bearing is the integral of the frequencies below 1x as shown in Figure 5.3, it is used since other effects influence these frequencies besides vortex rope.

asynchronous_bearing is the integral of the frequencies different from 1x as showed in Figure 5.3. Its primary use is to tell apart bumps from other problems. It is well known that a bump (approximated by a delta function) has Fourier transform with a constant value of one. So, if it is a bump it is very likely that asynchronous_bearing increases, while if there is another problem then the growth in magnitude should be localized in some specific frequency band.

RSI_bearing or rotor-stator interaction as described in [86] is the effect that gives pressure fluctuations resulting from the interaction of the rotating parts (the impeller) and the stationary parts (the distributors, the ones that control the flow) of the machine. The frequency depends on the number of impeller blades multiplied by the number of nodal diameters (respectively 13 and 2 on this machine).

crest_bearing, **pp_bearing** are the crest factor and peak to peak we explained in Section 4.4.

gap_bearing is the average of all positions in the orbit as can be seen in Figure 4.7b.

Orbit#_bearing are feature extracted from the orbit graph in Figure 4.7.

Notice that we used features built from other two pairs of sensors, one for the structure of the machine and one for the absolute displacement of the bearing from a fixed point. They have more or less the same type of features that are used for the shaft bearing relative displacement except for the orbit ones.

5.2.3 Safety

As you might have guessed from the information we scattered in the text up to now, safety is one of the most important concepts when dealing with hydroelectric power plants. In fact, safety is always taken into account with more conservative constraints both at design and operations control times. For this reason maintenance in this field is a very sensitive activity and decisions cannot be delegated to an artificial intelligence model, but instead are meant to be taken by a board of experts.

Being timing the core of maintenance and given the high dimensionality of the data coming from the sensors we explained in Section 5.2.1, it's clear the personnel in charge of maintenance's scheduling avail themselves of some type of autonomous system to detect anomalies, at least a set of thresholds. In this context surely a model with optimal performances can be of great help, but as usually happens, the more power we require from the model the less the reasons for the decisions it recommends are understandable to the personnel.

This last part is what led us to test the application of an additional interpretability component, independent of the model, to both exploit all the power we needed to discover the most complex anomalies and at the same time return to the user an explanation about the result itself.

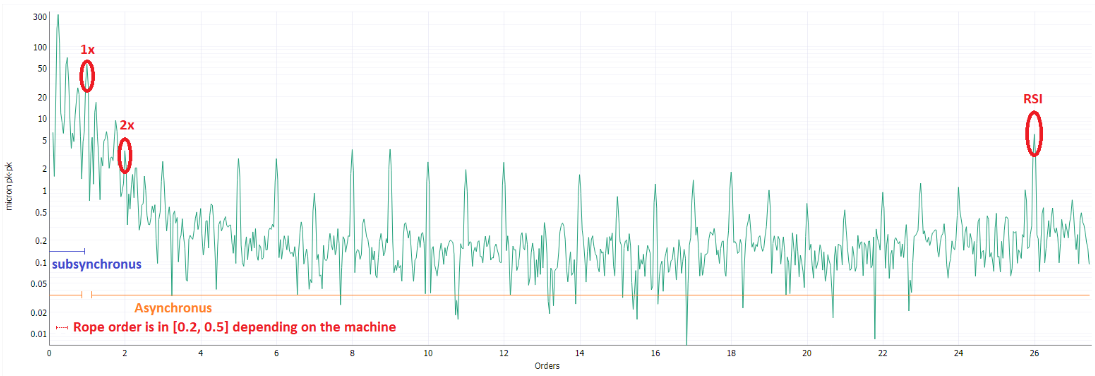


Figure 5.3: Representation of frequencies used to compute the main features [12].

5.3 Experiment

In this chapter, we will describe the choices that have been made to cope with the peculiarities of this specific application, the ones that have been explained in Section 5.2. After that, we will explain how the experiment is designed given the needs that we want the models to satisfy.

5.3.1 Choices

Multivariate anomaly detection: In Section 5.2.1 we anticipated that we have to deal with scarce, unlabeled, and unbalanced data coming from a multivariate process. From this description, it's sensible to model this problem as anomaly detection, since as pointed out in Section 3.2 the vast majority of the models developed to solve this task use the unsupervised learning paradigm. This choice comes in handy when we try to analyze the data as a whole to catch the more complex anomalies since in literature there are plenty of very powerful models for multivariate anomaly detection. Notice that the possibility to use the other learning paradigms of Section 3.1 is excluded since they would require at least a small labeled dataset.

Reducing complexity: The first problem to solve is the fact that some operating states are underrepresented, as said in Section 5.2.1 can make the model see them as outliers and so as anomalies. For this reason, we only analyzed high and Low power data. This is not much of a loss of generality since the others are not so important or represent transitional states like run-up and coast-down. Another choice to take was how many of the total 200 features to analyze since machine learning models are well known to suffer from the curse of dimensionality, i.e. the higher the dimension of the data is the bigger the dataset required to properly train the model is. So, since each machine has multiple components of the same type, e.g. multiple bearings, we decided to only analyze the structure and the turbine bearing, thus reducing the features to 65 (actually 63 since two are discarded being always zero for computational mistake in the plant's condition-based vibration monitoring system) and allowing the models to learn properly with the amount of data we had at our disposal.

Training and test sets: The seasonality we spoke of in Section 5.2.1 instead is not to be considered strictly as a problem, but to have a good representation of all operating states in the training set it is better to take the data from the "low-water" season. Unfortunately, the examined Unit entered the "low-water" season by mid-April, i.e. towards the end of the data, so the training set is placed at the end and that has to be kept in mind when reading the reported anomalies. The test set instead is the whole dataset (from January 2022 to the end of May 2022) since we want to know if the models can find all the anomalies.

Human evaluation: If from one side choosing unsupervised learning saves us from the expensive, complex ad sensible process of labeling the data, from the other side makes it way more difficult to effectively compare the models with one another. For some unsupervised tasks like clustering, there is indeed some performance measure like Silhouettes [87], but here all we could do is visually inspect the results and compare them knowing the anomalies we wanted to find smartly. The core concept is to start from the events that the company has recorded and then use the iterative process in Figure 5.4 to add the newly discovered anomalies. For this, the collaboration with the expert of the domain was essential, since there is the need for extended knowledge and years of experience in the field to be able to find the causes for the new anomalies and thus confirm the result of the model. One important thing is that if an anomaly is not confirmed, it doesn't mean it is an error. There can be non-immediate reasons well hidden in the over 200 features that cause an anomaly. But statistically, anomalies should be rare events, so we considered better those models whose plotted results allowed us to easily distinguish the anomalies between them and from the nominal working conditions.

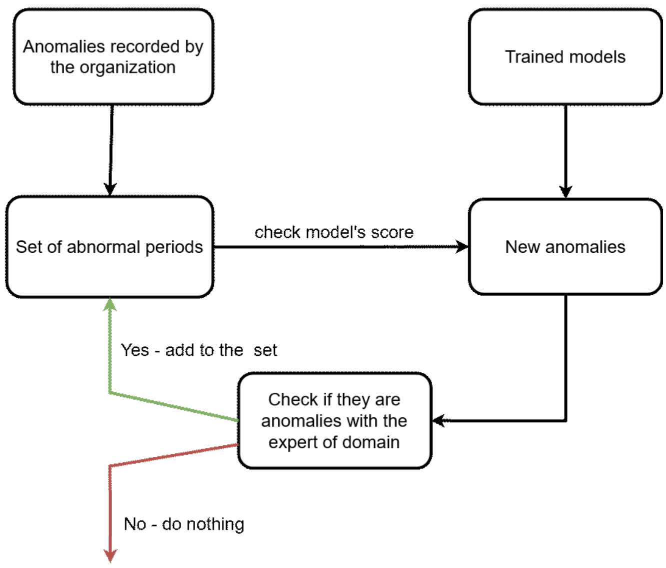
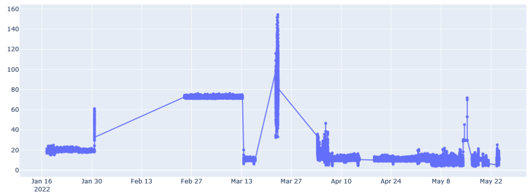
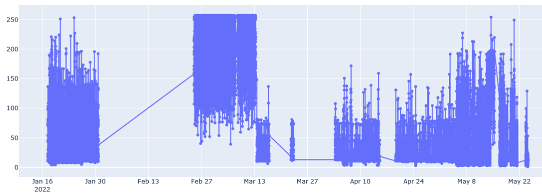


Figure 5.4: Flow chart of the iterative process used to gather the segments of the data that are to be considered as anomalies



(a) Example of the result of a model with well-distinguished anomalies.



(b) Example of the result of a model with hard-to-distinguish anomalies.

5.3.2 Anomalies

Starting from the anomalies that have been recorded by the company from the RCM operations on the machine we are analyzing, we used the iterative process explained in Section 5.3.1 to gather anomalies that were hidden in the complexity of the data, not having generated any fault.

The anomalies we imposed to find in the model's output are:

- On the date January 30 2022 there was a fault that wasn't detectable from the sensors but led to a major maintenance operation, but at time 18:59 and seconds 07, 08, and 11 there can be seen what has been identified as a bump in the system.

- From March 22 2022 to April 6 2022 there is an increase in the machine's structure vibrations along the X axis. As we can see from Figure 5.6 this segment of data is abnormal with respect to its operative condition (remember that vortex rope generates forces and torques). This anomaly was discovered by the models since it didn't surpass any threshold and didn't cause any fault or in loco personnel report.

- From February 24 2022 to March 13 2022 there is an anomaly well known to the company, since the velomitors for the structure vibration were wrongly installed after maintenance. This can be seen as a flat curve in Figure 5.7 (only unfiltered).

- From May 14 2022 to May 16 2022 there was a problem in the connection between the center and the plant, so the signals are held or put to zero.

- After the maintenance it was not possible to reach the optimal alignment between the shaft components, as a result of this the power spectrum in the first and second order harmonics increases as expected given what we have explained in Section 4.5. Actually as reported in Section 5.3.1 we took the training from the most recent part of the dataset, so the model should report the before maintenance data as anomalous.

- At last there is also the fact that in the more recent part of the dataset an increase in the overall vibrations is known to the company and at the time of writing its causes were under investigation. Along with this also connection problems have resulted in some missing features from time to time.

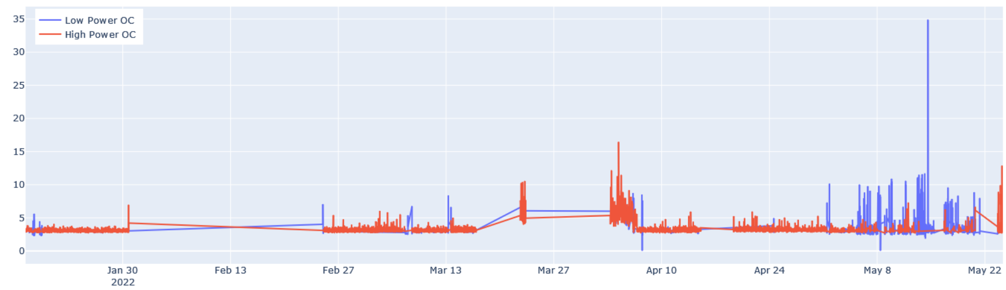


Figure 5.6: Plot of the Crest_structure_X feature from High Power and Low Power operative states.

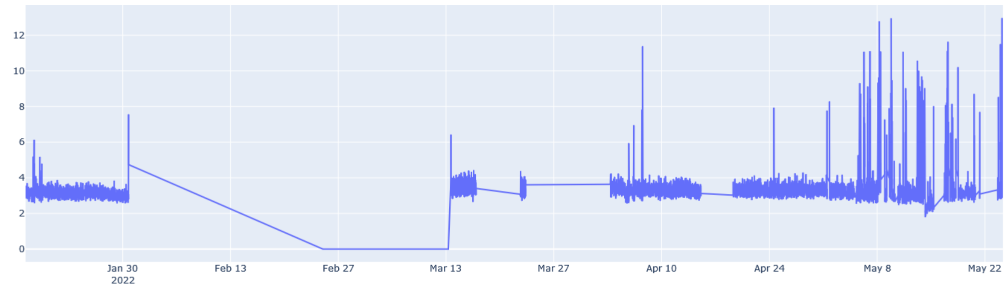


Figure 5.7: Plot of the Crest_structure_Y feature.

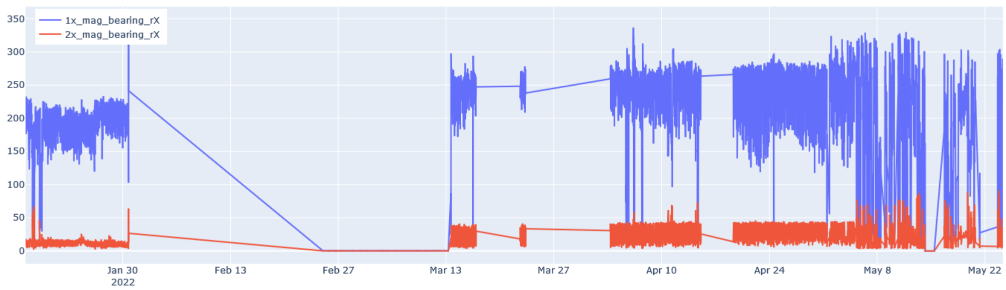


Figure 5.8: Plot of the power spectrum of first and second order to show the increase after the maintenance.

5.3.3 Models' comparison

The experiment we carried on to gather the data for this work, essentially consists of training and testing the set of machine learning models, that were introduced in Section 3.4, with the train-test data explained in 5.3.1. On top of this, we also built a set of datasets to check whether the models are robust to the following conditions (there are two "on-off" conditions so four datasets in total):

- We want the models to be stable in presence of **errors of the sensors**. The system we extracted the data from has the setting to replicate the last seen value in case of empty fields in the data received from the server on the plant site, so these sections have to be shown as an anomaly from the model. But a simple filter can be used to remove this faulty data, so we decided to also try the model performance with the filtered data.

- We want the models to be stable in presence of the **vortex rope effect**. As explained in Section 4.5 it generates torques and forces on the shaft, thus having a big impact on the distribution of the features. So, given that the vortex rope is predominant on part load, that in our work is under the operative condition of low energy, and that its effect becomes negligible after 80% Q_{BEP} that in our work is in high power operative conditions. We make two datasets, one with only high power operative conditions and one with all.

Since the experiment is also a test for the models' capability to deal with the data at different levels, from a small number of features up to several components altogether, when we applied the filter we also kept only the smallest relevant subset of features according to the expert of domain. Namely wicked gates opening, power, peak to peak of the relative displacement of the shaft to the bearing, the crest factor on the machine's structure vibration, Orbit1 and Orbit4.

One very important thing to notice is that the experiment was developed in Python [88] using the PyOD [89], PySAD [90] and SHAP [91] libraries, so the results of this work are ready to be applied in the industry.

Chapter 6

Results

6.1 IsolationForest

As we can see from Figures 6.1, 6.2, 6.3, 6.4 isolation forest is indeed a good model since it found all the anomalies we were searching for (not by chance is one of the most used for anomaly detection). Another important side of its performance resides in its speed since all experiments take less than two seconds to be trained and less than four to give an anomaly score to each point. But the inconvenience is that the anomaly score is rather noisy and that makes it annoying to be read by the personnel.

Overall its performances are very good and we can notice how its performances degrade in the filtered setups. This is an index that probably needs all the features since they carry important information to distinguish the anomalies.

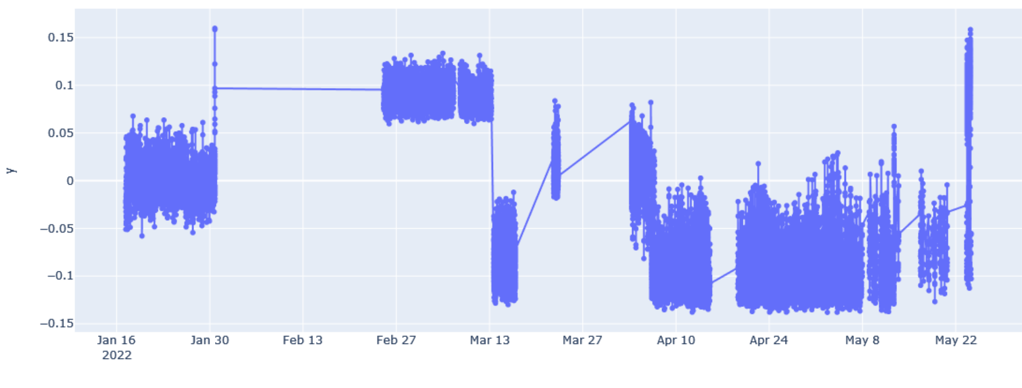


Figure 6.1: Plot of the anomaly score from the isolation forest model with unfiltered data only with High Power OS.

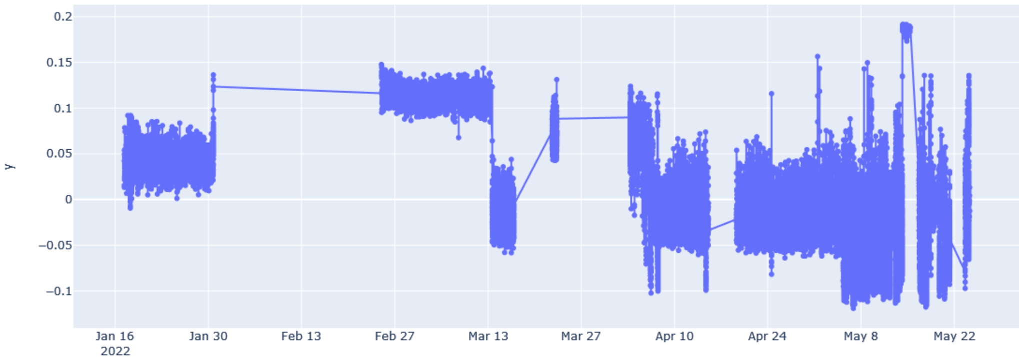


Figure 6.2: Plot of the anomaly score from the isolation forest model with unfiltered data only with all OS.

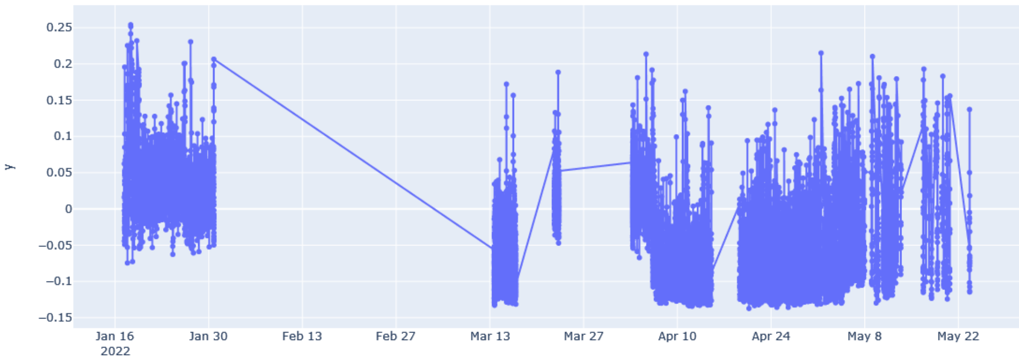


Figure 6.3: Plot of the anomaly score from the isolation forest model with filtered data only with High Power OS.

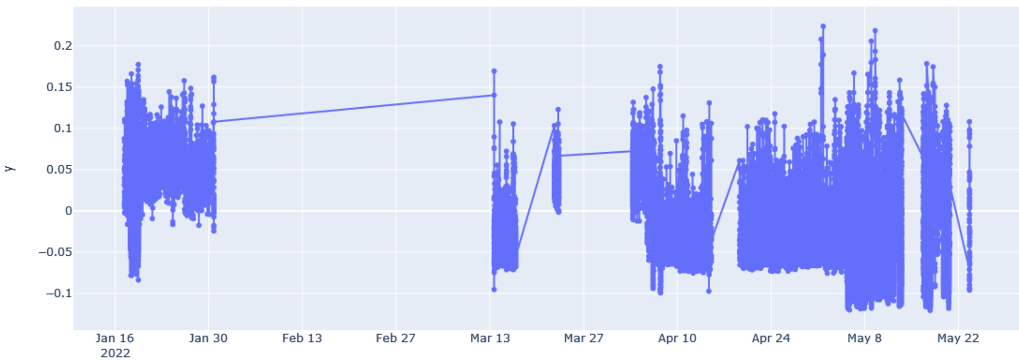


Figure 6.4: Plot of the anomaly score from the isolation forest model with filtered data only with all OS.

6.2 Auto-Encoder

Auto-encoder also found all the anomalies we were searching for, but the anomaly score graph gives a very high score to the anomalies and so anomalies and non-anomalies are more distinguishable making it more suitable for visualization purposes. Another big advantage is that reducing the number of features doesn't impact too heavily on the anomaly score (note that the signal errors of unfiltered data modify the scale, but if re-scaled the two plots are very similar, and that makes it suitable for a system that needs a smaller subset of related features to be analyzed together.

From the point of view of the speed of execution, the training time is up to 305 seconds, way higher than the isolation forest since the model is full of regularization and dropout layers to avoid overfitting. But the execution time is very fast, lower than three seconds that taking into account some randomness is in line with the one of isolation forest.

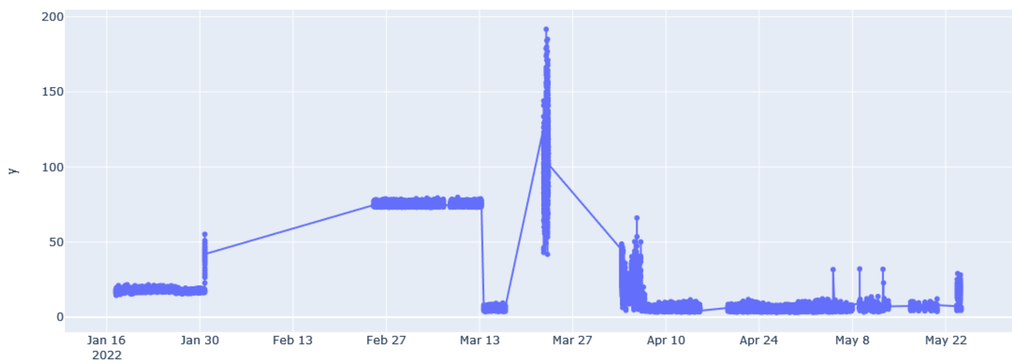


Figure 6.5: Plot of the anomaly score from the Auto-Encoder model with unfiltered data only with High Power OS.

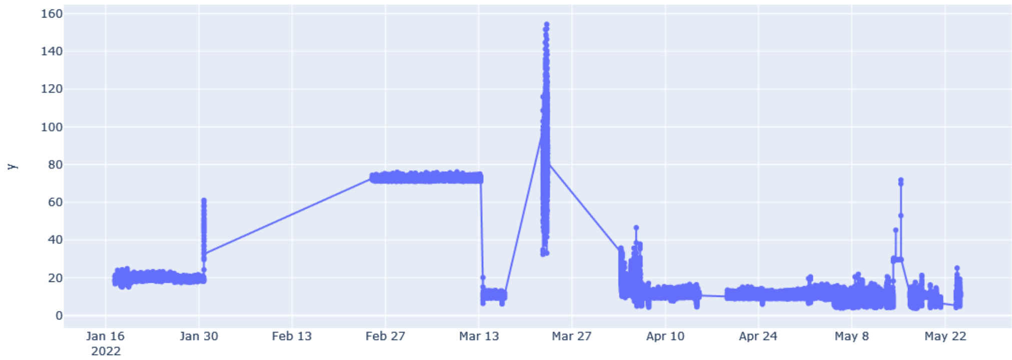


Figure 6.6: Plot of the anomaly score from the Auto-Encoder model with unfiltered data only with all OS.

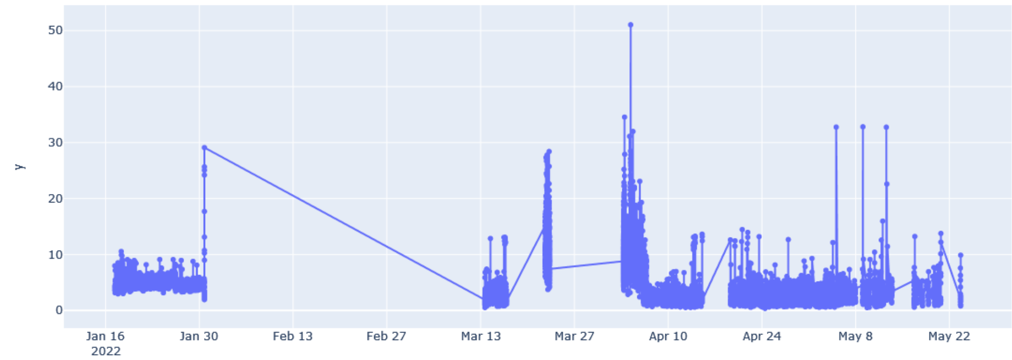


Figure 6.7: Plot of the anomaly score from the Auto-Encoder model with filtered data only with High Power OS.

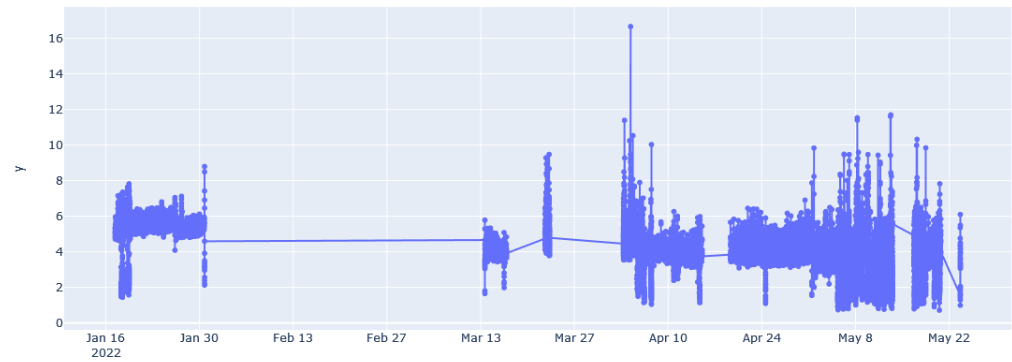


Figure 6.8: Plot of the anomaly score from the Auto-Encoder model with filtered data only with all OS.

6.3 DeepSVDD

DeepSVD is a model built for streams of data but with a different approach to training and testing, it has a function called fit_and_score that point-wise fits and then scores the next instance. If on one hand, it can follow even a non-stationary source of data, on the other the information about constant anomalies like the misalignment after the maintenance is lost. For this model, we were able to winch the training set to be the same as the other models like isolation forest and auto-encoder and then score the entire dataset.

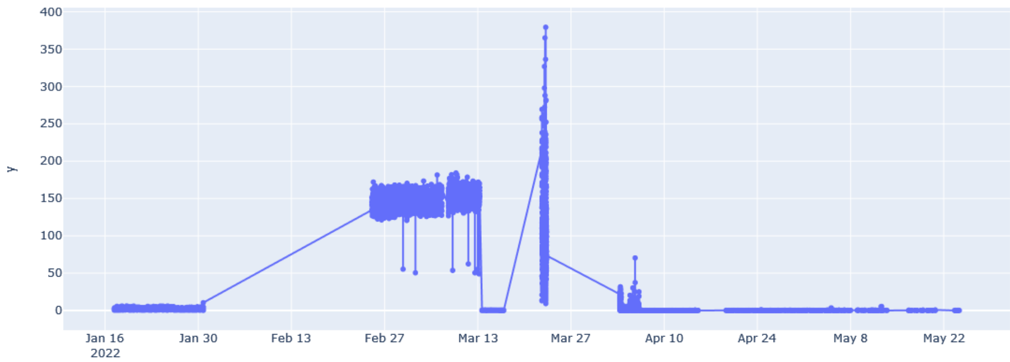


Figure 6.9: Plot of the anomaly score from the DeepSVDD model with unfiltered data only with High Power OS.

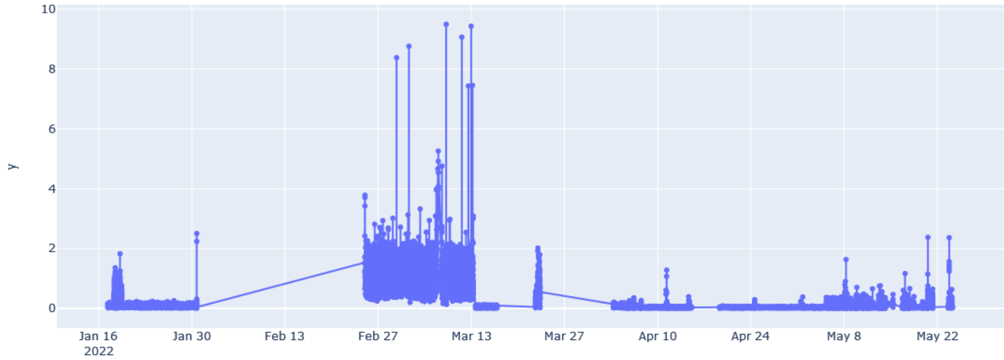


Figure 6.10: Plot of the anomaly score from the DeepSVDD model with unfiltered data only with all OS.

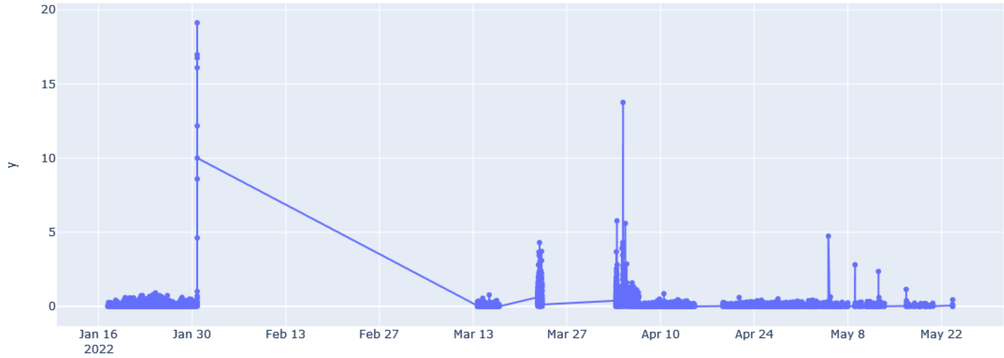


Figure 6.11: Plot of the anomaly score from the DeepSVDD model with filtered data only with High Power OS.

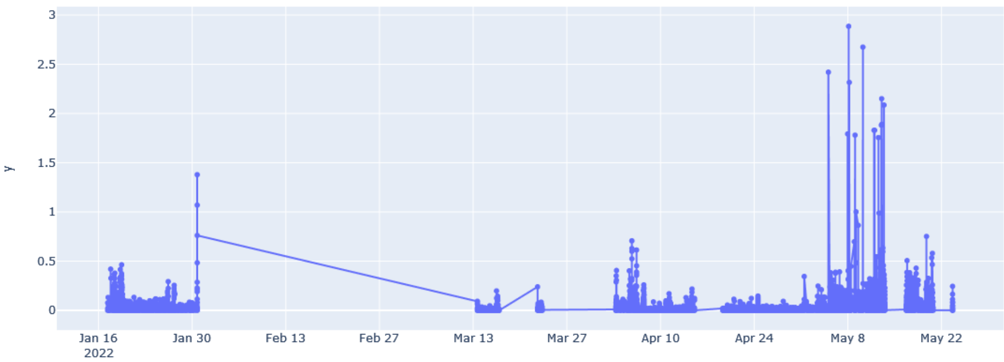


Figure 6.12: Plot of the anomaly score from the DeepSVDD model with filtered data only with all OS.

6.4 KitNet

KitNet is able to recognize all the anomalies and its plot is very similar to the auto-encoder, both in the filtered and unfiltered data so also this model can be used when there is the need to analyze a smaller subset of features. The visual interpretability is lower than the auto-encoder since it has an amplification effect on some anomalies probably given by the fact that it is an ensemble of models.

From the point of view of the speed of execution it is faster than an auto-encoder to be trained with a maximum of 50 seconds (depending on the parameters of the model), but the time it requires to analyze all the data is up to 70 seconds, that is way too high to be able to attach the SHAP module to it.

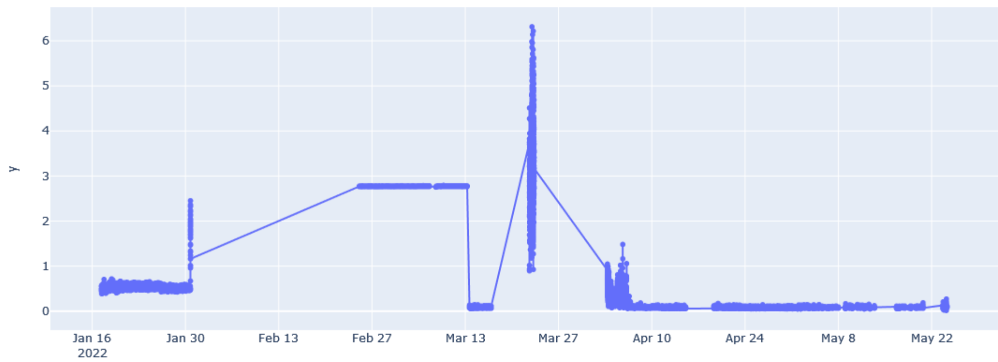


Figure 6.13: Plot of the anomaly score from the KitNet model with unfiltered data only with High Power OS.

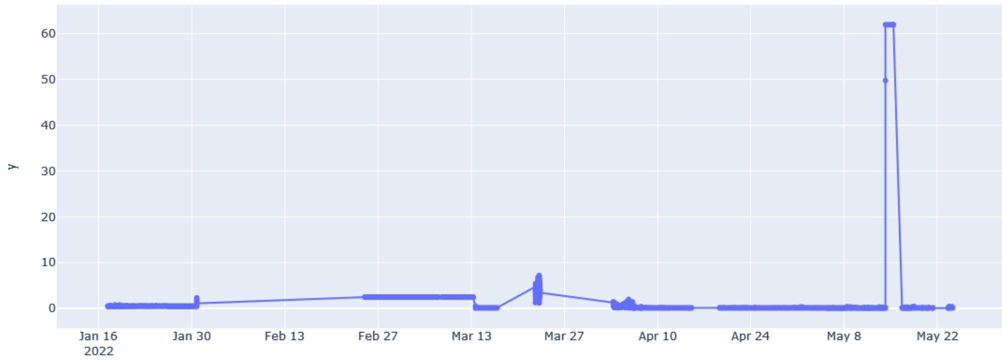


Figure 6.14: Plot of the anomaly score from the KitNet model with unfiltered data only with all OS.

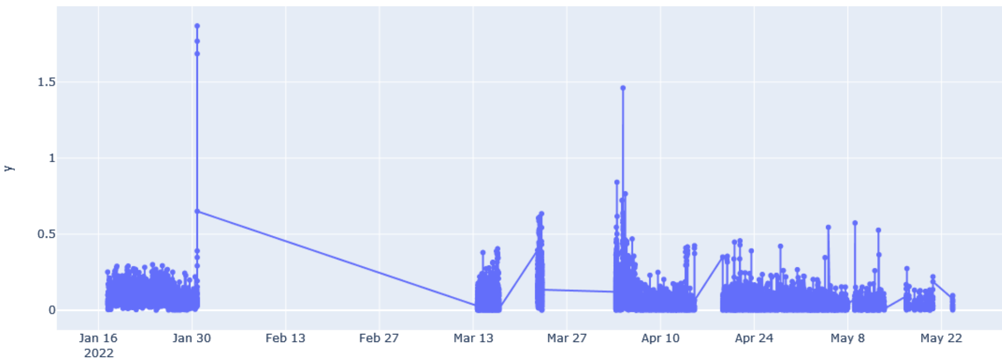


Figure 6.15: Plot of the anomaly score from the KitNet model with filtered data only with High Power OS.

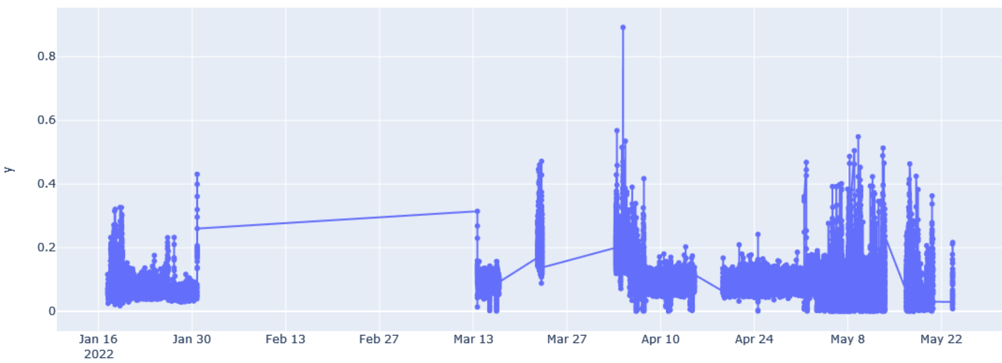


Figure 6.16: Plot of the anomaly score from the KitNet model with filtered data only with all OS.

6.5 RSHash

RSHash is a model built to work with a stream of data like DeepSVD but has a different approach to training and testing, it has a function called fit_and_score that point-wise fits and then scores the next instance and then also scores it. If on one hand, it can follow even a non-stationary source of data, on the other the information about constant anomalies like the misalignment after the maintenance is lost. Also, some of the main anomalies are not recognized.

From the perspective of the execution time is slower than the alternatives taking up to 550 seconds to fit and score the whole dataset.

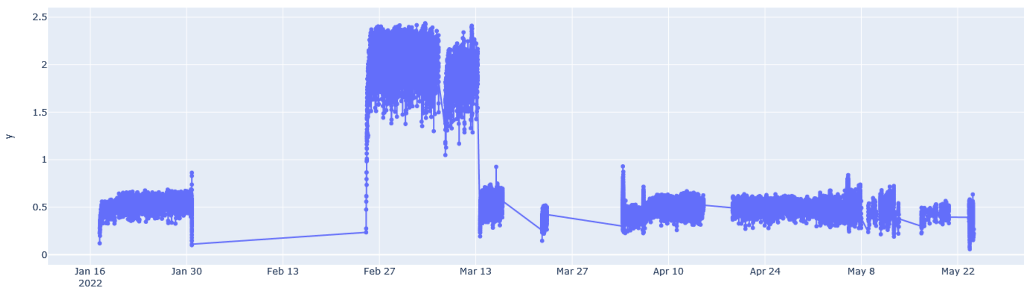


Figure 6.17: Plot of the anomaly score from the RSHash model with unfiltered data only with High Power OS.

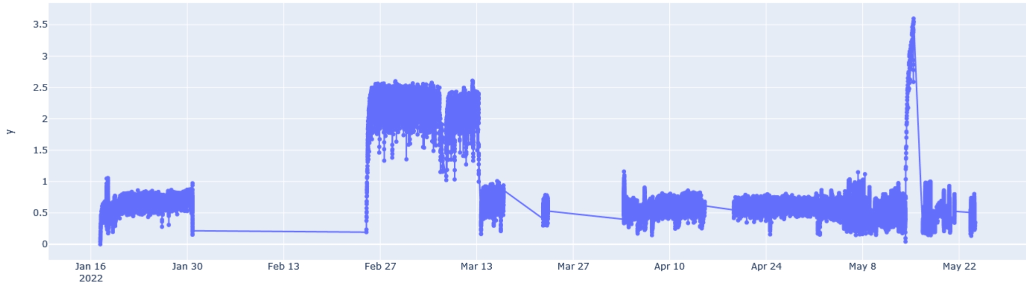


Figure 6.18: Plot of the anomaly score from the RSHash model with unfiltered data only with all OS.

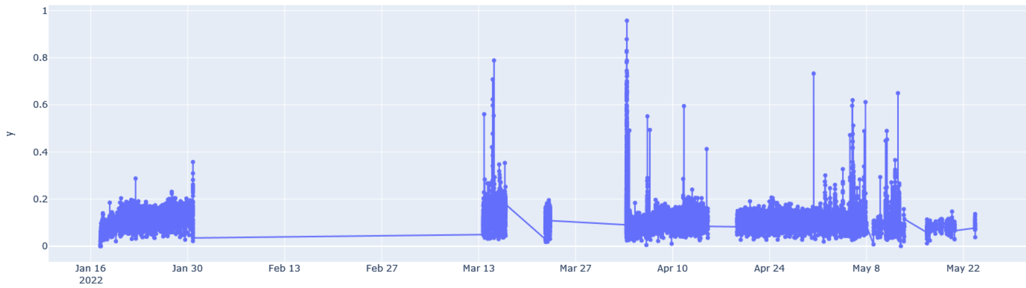


Figure 6.19: Plot of the anomaly score from the RSHash model with filtered data only with High Power OS.

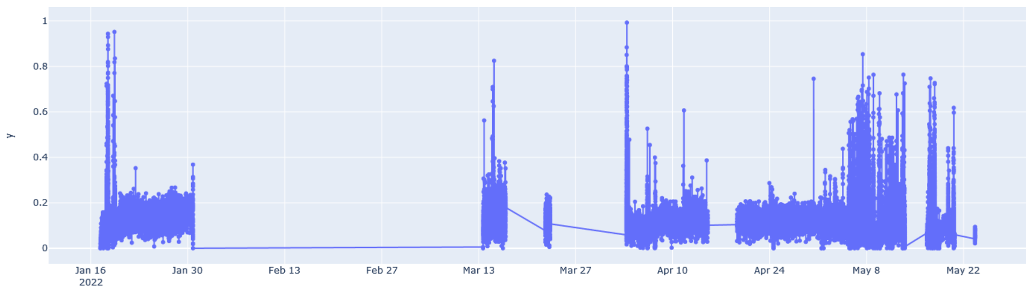


Figure 6.20: Plot of the anomaly score from the RSHash model with filtered data only with all OS.

6.6 SHAP

SHAP values are relatively easy to be interpreted, since being built on Shapely values, they are a fair split of the model output given the contribution of each feature. Then the ones with a value enough higher than zero are features that are to be considered responsible for the fault, the ones enough lower than zero are the ones not related to the anomaly. Only the ones near zero are dubious, but if the anomaly score is high inevitably some features have to stand out.

We found out that despite being extremely powerful, SHAP has shown some undesirable effects: One of them is that interpretability algorithms are designed to highlight a limited number of important features. So, when a lot of them are actively contributing to the fault, it tends to move the weights only towards one of them. Another one is that being a black box model it inherits the lack of insight the model about the properties of the problem, e.g. the magnitude power features are dangerous if high but they are sometimes reported as anomalous when too low since it's a rare event.

6.6.1 Post maintenance misalignment

Searching for the cause of the anomaly score translation towards higher values that can be seen before January 30 (e.g. in Figure 6.6), we encounter for the first time both of the problems discussed earlier.

As we can see from Figure 6.21, both the 1x and 2x components of the relative vibration's power spectrum of the bearing get a substantial increment, as we would expect knowing that we are looking for signs of shaft misalignment. On the contrary, we can see in Figure 6.22 that the synchronous component of the absolute vibration's power spectrum of the bearing decreased.

SHAP, as we can see in Figure 6.23, has highlighted gap_bearing_rX and 1x_magn_bearing_aX as principal causes of the fault. From one side we can give merit to the algorithm that at least has identified the unit's faulty component, the bearing. But on the other side, the relative 1x and 2x magnitude of the relative vibration haven't been highlighted enough (first problem).

The explanation for the correct identification of the faulty component is that the misalignment directly affects the bearing, so it is sensible that the other components' signals were for the most part unaltered by the problem. While an explanation for the misjudgment of the correct feature could be that: from one side gap is very fragile as a feature and tends to give problems even when the anomalies are elsewhere. While from the other the high variability of the relative vibration magnitude in the training set leads SHAP to think that a variation on the absolute one is more relevant, maybe because (reading the signal from right to left since the train is at the end) the absolute magnitude increases going out its range of variation while the other decrease and so there are a discrete number of points with such values.

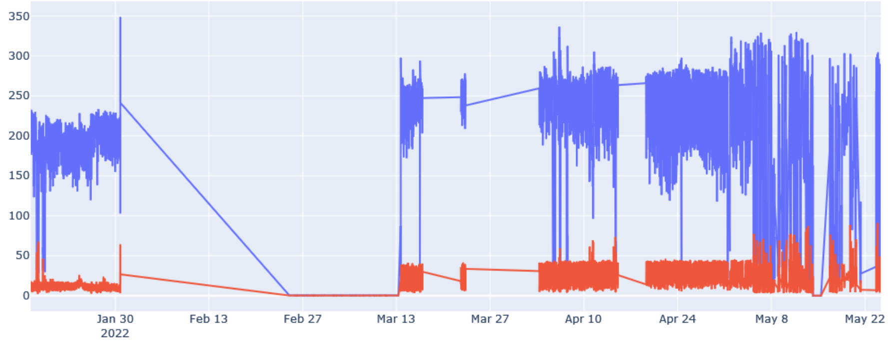


Figure 6.21: Plot of x1 and x2 magnitude for the relative vibration of the bearing, only in the X direction since the Y is almost the same.

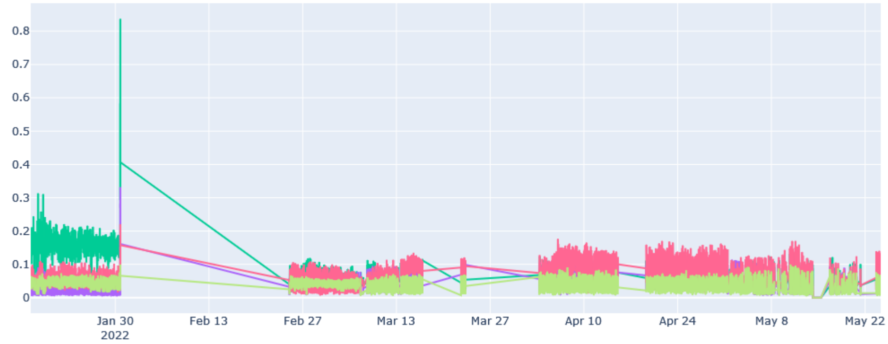


Figure 6.22: Plot of x1 and x2 magnitude for the absolute vibration of the bearing, we can see 1x_magn_bearing_aX that decreased after the maintenance.

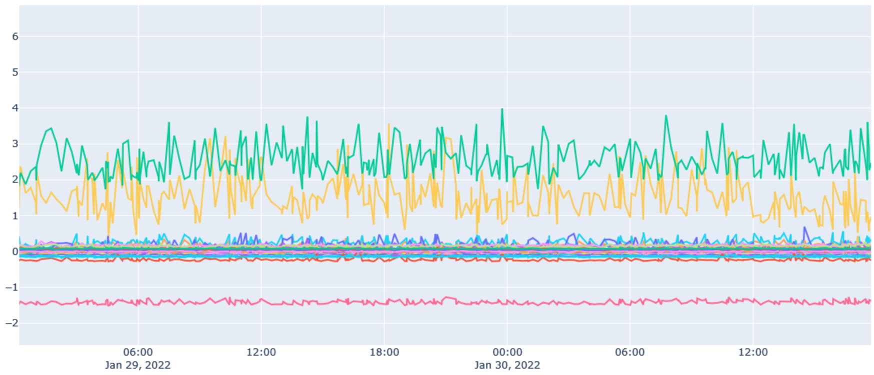


Figure 6.23: SHAP gap_bearing_rX and 1x_magn_bearing_aX values related to the better alignment of shaft components before the maintenance. The one below is gap_bearing_rY.

6.6.2 Anomaly of January 30 2022

Regarding the anomaly of 30 January that can be seen completely in Figure 6.24 and in more detail in Figure 6.25. We can see a first peek with the predominant 1x_magn_bearing_aX and lower async_bearing_rX, async_bearing_rY and 2x_magn_bearing_aX that probably is a bump since the whole spectrum is involved, followed a peak of all the relative and absolute creast_bearing that probably is related to the subsequent increase in the vibrations.

This time instead, SHAP was able to well define the problem (the bump) while catching only part of the localization, since also the structure was involved (we know SHAP can highlight a limited number of features at a time).

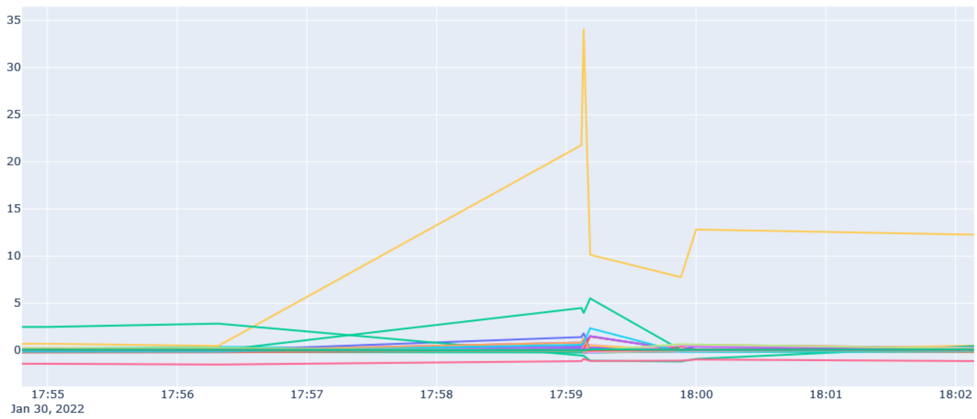


Figure 6.24: Coarse image of the 30 January fault evidencing the enormous importance of the synchronous magnitude of the absolute vibration of the bearing.

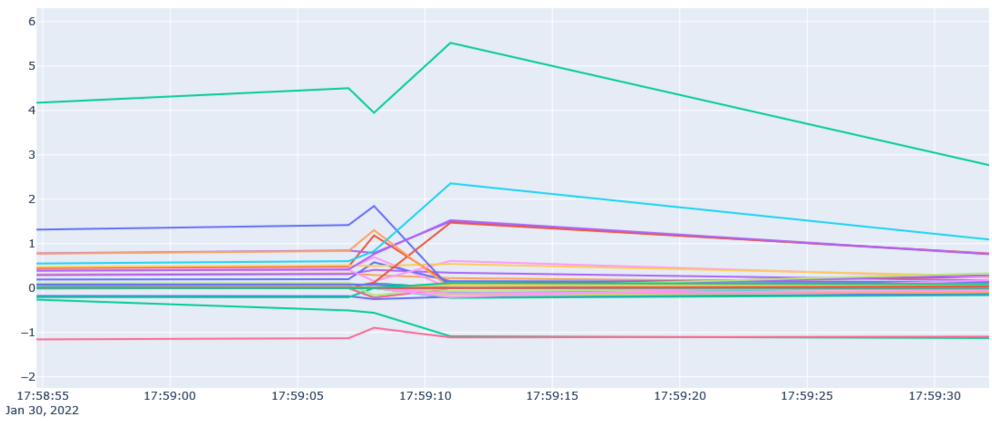


Figure 6.25: Detailed image of the 30 January anomalous double peak's less prominent features explanation.

6.6.3 Unconnected sensor

Here there is not much to say, essentially the fact that a signal was no more available broke a good amount of the other signals.

In Figures 6.26 and 6.27 we can see that gap and crest factor are reported or both relative X and Y displacements, while in Figures 6.28 and 6.29 we can see that the features in that period reach unreasonable values. 2700 microns means that the shaft is out of the bearing since usually the available displacement is about 500 microns summing both sides.

One thing to notice is that also here the gap features absorb all the "blame" for the fault, even if other features contribute to it. In the end, this time was very hard for SHAP to highlight the features that correctly identify the problem.

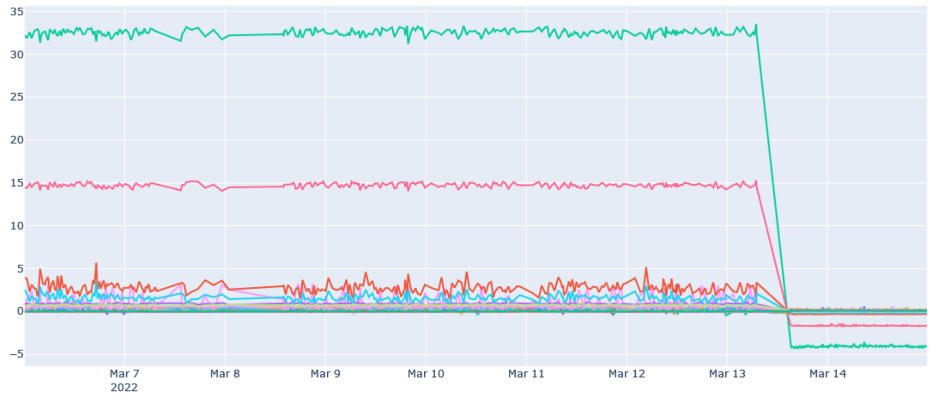


Figure 6.26: SHAP values evidencing that it considers the gap values as important features to identify the cause of the anomaly.

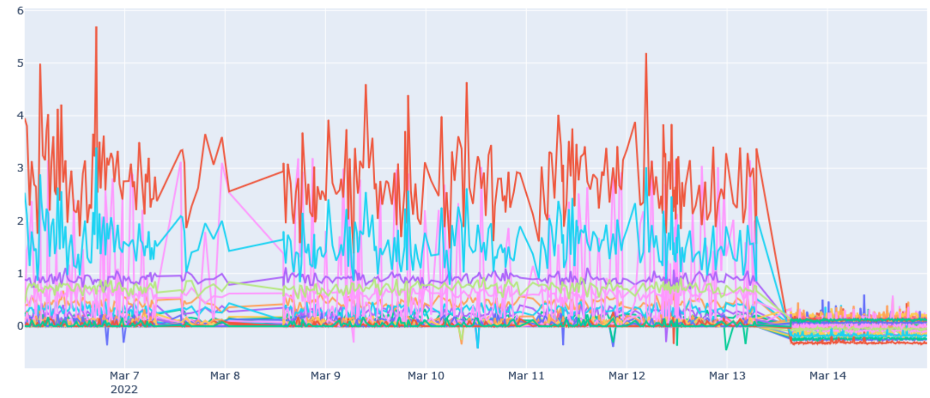


Figure 6.27: SHAP also highlight the the crest factor for both relative X and Y displacement along with the synchronous phase of relative X displacement.

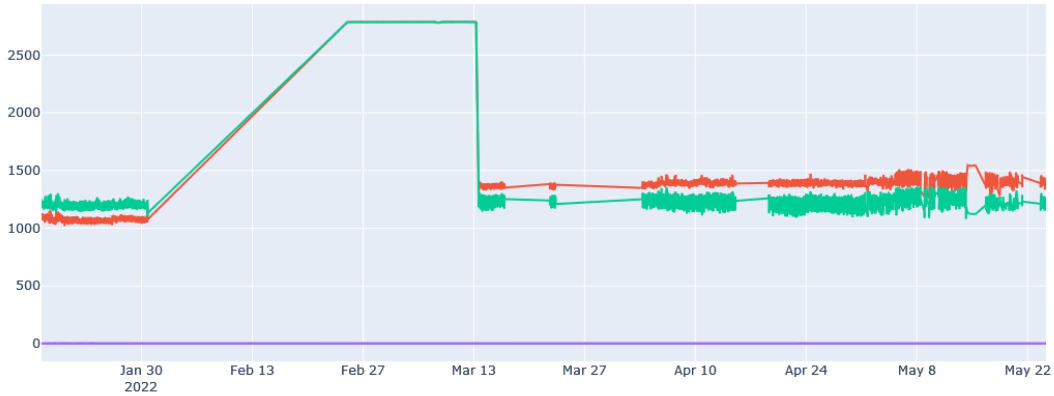


Figure 6.28: Gap for both X and Y directions of the bearing's relative displacement vibration.

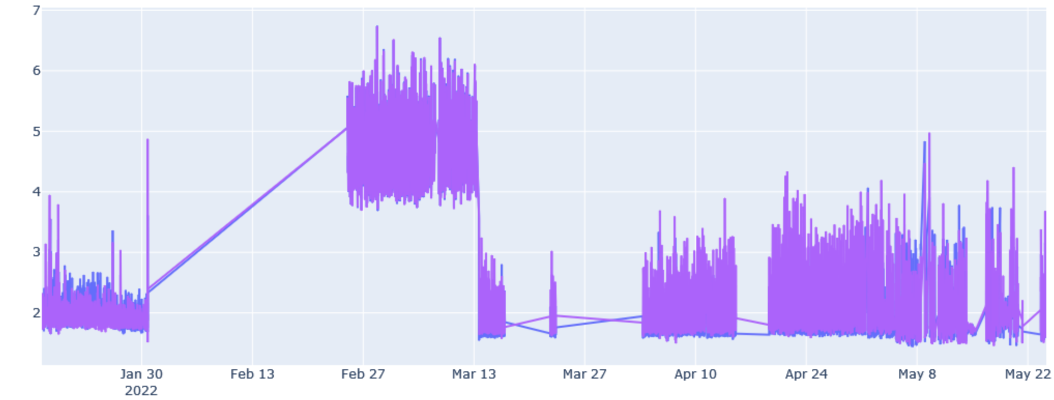


Figure 6.29: Crest factor for both X and Y directions of the bearing's relative displacement vibration.

6.6.4 New anomaly

This anomaly is actually the true stress test for this experiment. It was first discovered by the models since no threshold was surpassed or report of problems was made by the in-loco personnel. As we can see from Figure 5.6 there is a period with a considerable increase in the structure X vibration. With the collaboration of the expert of the domain, we were able to obtain a reasonable explanation.

We already introduced RSI and said that is generated by the interaction between the impeller and the distributor. This effect is known to have greater effects when the distributor is in the upper part of its range. Quick distributor's position changes can easily affect it and make it increase or decrease in amplitude "around" its steady-state pulsation. The investigated machine has a very particular and unfortunate specificity, in the sense that the head-cover and the runner exhibit a coupled eigenmode (where the coupling is given by the water enclosed in the labyrinths and the side-chamber) having both frequency and spatial shape compatible with the RSI pressure field. It means that the machine runs with the head-cover "in resonance" with the RSI pressure field's pulsating forces, thus amplifying whatever happens hydraulically to this effect. It happens that the most sensitive direction is the X one. In the period identified as anomalous, a test had been performed at the site for which some parameters were changed in the turbine governor, which led to a much more reactive wicket-gates response to the grid's frequency perturbations, which also made the RSI pressure field more variable. This cause summed up with the season entering the "low-water" period as explained in Section 5.3.1, which led by its nature much more frequent power set-point variations, hence wicket-gates movements. There is a range of powers (combinations of net head and flow controlled by the wicket-gates) where RSI effects and vortex rope effects sum up to a maximum, and this range was much more frequently crossed during the "low-water" season. The continuous changes in the power set-point also cause pressure waves in the penstock, which are reflected in increased low-frequency content (sub-synchronous). Effects of low-frequency pressure field fluctuations reflect on both relative and absolute vibrations, while the RSI effects are noticeable only in the absolute vibrations and might be more easily masked by the normal operation because the head-cover vibrates a lot at RSI frequency in any case.

This time SHAP has done a great job in both causes recognition and localization as we can see from Figure 6.30.

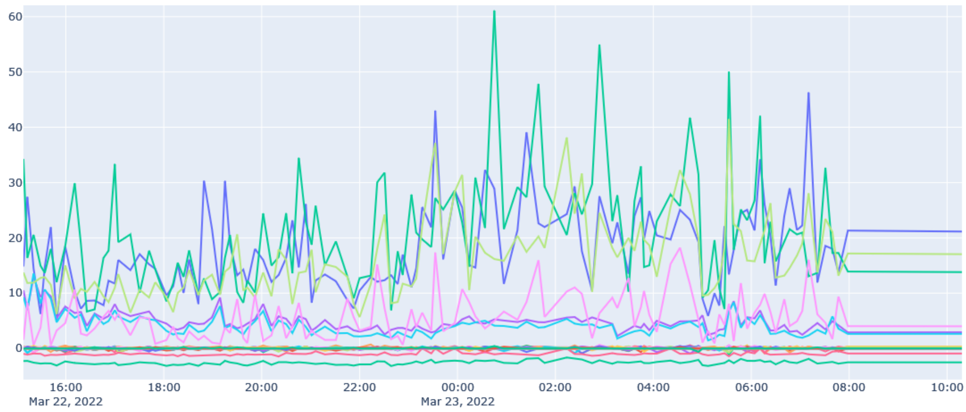


Figure 6.30: SHAP values for subsync_struct_X, x2_magn_struct_X, x1_magn_struct_X, vortex_struct_X, async_struct_X and rms_struct_X for the new anomaly.

6.6.5 More recent anomalous period

In the most recent period, the machine vibration started to increase, while the connection to the plant started to fail from time to time because of electronic problems in a firewall at the site. So in Figure 6.31, one should analyze point by point what is happening. In reality, the combination of the two generates the conditions to let SHAP give higher weights to random features.

When the signal is missing, we saw in Section 6.6.3 that some of the missing features are drawn based on their overall stability in the training set. Plus when the overall vibration gets extremely high, from one side all the features get more anomalous and from the other, the bumps change the distribution of all the features related to the frequency analysis.

In the end, this was reported only to show an instance where SHAP can really do nothing to tell apart the causes of the anomalies.

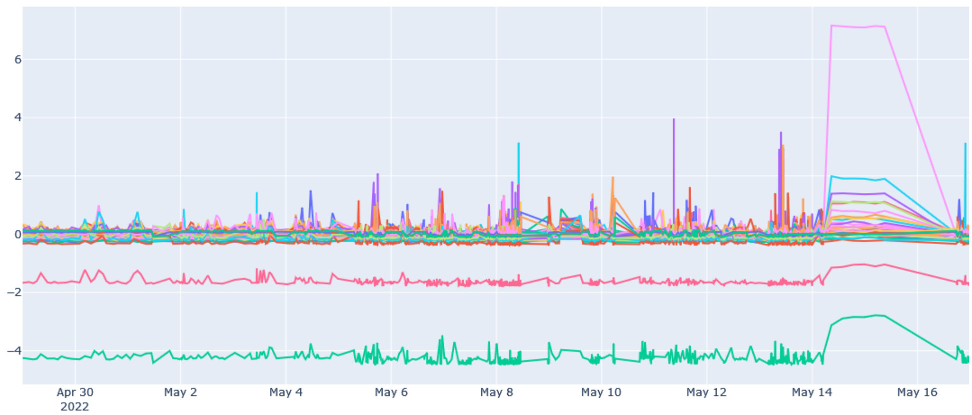


Figure 6.31: SHAP for the anomalous period near the end of the data.

Chapter 7

Conclusions

From the experiment, we can draw two different results. For the anomaly detection part, we found out that there are indeed good algorithms that can capture each anomaly, and among them surely auto-encoder is to be considered the best for this particular field of application. Its best properties are its robustness to anomalies in the training, its speed of execution, the well-proportioned graph it returns (that is nice for visualization purposes), and the fact that is stable with different subsets of features and with different training sets. This last property is the most remarkable since it means that the model can be applied in both a top-down approach that takes into account the whole system, as well as a bottom-up approach that instead oversees the system through small groups of features. For the interpretability part, SHAP indeed has some types of instances in which it performs very well, but sometimes it loses its effectiveness. Going more into detail, it seems that the fact that numerous features co-participate in the same anomaly goes against its principles. Overall, even if it takes its time to compute (for the whole train set it would have taken 170h, so we only take small pieces and heavily sub-sampled segments), it's able for the vast majority of the times to at least identify the faulty system's unit, so using a bottom-up approach could be the key to overcome SHAP weaknesses.

We reported all the results between the main reading and the appendix so if the reader has different needs, some additional ideas can be found in them.

For future works, it may be helpful to have a labeled dataset or a metric to systematically compare the models. Although it will be surely difficult to express usability properties, it will be an advantage in both the automation of the process of discovering the best model and the acceptability of the results.

Bibliography

[1] European Commission. *A European Green Deal*. 2019. URL: https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en (visited on 07/06/2022).

[2] Askari. Mohammad Bagher et al. «Hydroelectric Energy Advantages and Disadvantages». In: 2015.

[3] Philippe Heller, Erik F.R. Bollaert, and Anton J. Schleiss. «Comprehensive system analysis of a multipurpose run-of-river power plant with holistic qualitative assessment». In: *International Journal of River Basin Management* 8.3-4 (2010), pp. 295–304. DOI: 10.1080/15715124.2010.517672. eprint: https://doi.org/10.1080/15715124.2010.517672. URL: https://doi.org/10.1080/15715124.2010.517672.

[4] Evrencan Özcan, Rabia Yumuşak, and Tamer Eren. «Risk Based Maintenance in the Hydroelectric Power Plants». In: *Energies* 12 (Apr. 2019), p. 1502. DOI: 10.3390/en12081502.

[5] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[6] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. DOI: 10.48550/ARXIV.1705.07874. URL: https://arxiv.org/abs/1705.07874.

[7] Shubham Jain et al. «Enhancing prediction and covariate-driven interpretation of stream flow duration curves using Machine Learning algorithms and

surrogate model explanations≫. In: *AGU Fall Meeting Abstracts*. Vol. 2021. Dec. 2021, H22A-06, H22A–06.

[8] Walter Samuel et al. ≪Machine Learning Solutions for Reservoir Modelling≫. In: *AGU Fall Meeting Abstracts*. Vol. 2021. Dec. 2021, H55Q-0931, H55Q–0931.

[9] Liat Antwarg et al. ≪Explaining anomalies detected by autoencoders using Shapley Additive Explanations≫. In: *Expert Systems with Applications* 186 (2021), p. 115736.

[10] Lucas C Brito et al. ≪An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery≫. In: *Mechanical Systems and Signal Processing* 163 (2022), p. 108105.

[11] Heejeong Choi et al. ≪Explainable anomaly detection framework for predictive maintenance in manufacturing systems≫. In: *Applied Soft Computing* 125 (2022), p. 109147. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2022.109147. URL: https://www.sciencedirect.com/science/article/pii/S1568494622004069.

[12] Andritz Hydro. *Property of Andritz Hydro, which granted us permission to use*. https://www.andritz.com/hydro-en.

[13] Balasaheb S Gandhare and Milind Akarte. ≪Maintenance strategy selection≫. In: *Ninth AIMS International Conference on Management*. AIMS International Maharashtra, India. 2012, pp. 1330–1336.

[14] M Bevilacqua and M Braglia. ≪The analytic hierarchy process applied to maintenance strategy selection≫. In: *Reliability Engineering & System Safety* 70.1 (2000), pp. 71–83. ISSN: 0951-8320. DOI: https://doi.org/10.1016/S0951-8320(00)00047-8. URL: https://www.sciencedirect.com/science/article/pii/S0951832000000478.

[15] Evan Zabawski. *Maintenance nomenclature*. Park Ridge, IL : Dec. 2020.

[16] Jamie Coble. ≪Merging Data Sources to Predict Remaining Useful Life–An Automated Method to Identify Prognostic Parameters≫. In: (Aug. 2022).

[17] Geert Waeyenbergh and Liliane Pintelon. «A framework for maintenance concept development». In: *International Journal of Production Economics - INT J PROD ECON* 77 (June 2002), pp. 299–313. DOI: `10.1016/S0925-5273(01)00156-6`.

[18] Gamax Laboratory Solutions. *Maintenance Strategies.* `https://gamaxlabsol.com/matlab/matlab-for-industry/predictive-maintenance/`. Accessed: 2022-08-15.

[19] Cornelius Scheffer and Paresh Girdhar. *Practical machinery vibration analysis and predictive maintenance*. Elsevier, 2004.

[20] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[21] Ernest C Fitch. *Proactive maintenance for mechanical systems*. 5. Elsevier, 2013.

[22] John Moubray. *Reliability-centered maintenance*. Industrial Press Inc., 2001.

[23] Alan M Turing and J Haugeland. «Computing machinery and intelligence». In: *The Turing Test: Verbal Behavior as the Hallmark of Intelligence* (1950), pp. 29–56.

[24] Tom M Mitchell and Tom M Mitchell. *Machine learning*. Vol. 1. 9. McGraw-hill New York, 1997.

[25] Nils J. Nilsson. *INTRODUCTION TO MACHINE LEARNING*. `https://ai.stanford.edu/~nilsson/MLBOOK.pdf`(visited 27-07-2022). 1998.

[26] Leo Egghe. «Untangling Herdan's law and Heaps' law: Mathematical and informetric arguments». In: *Journal of the American Society for Information Science and Technology* 58.5 (2007), pp. 702–709.

[27] Varun Chandola, Arindam Banerjee, and Vipin Kumar. «Anomaly detection: A survey». In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[28] JWL Glaisher. «On the rejection of discordant observations». In: *Monthly Notices of the Royal Astronomical Society* 33 (1873), pp. 391–402.

[29] Francis Ysidro Edgeworth. ≪Xli. on discordant observations≫. In: *The london, edinburgh, and dublin philosophical magazine and journal of science* 23.143 (1887), pp. 364–375.

[30] Gian Antonio Susto et al. ≪A Fraud Detection Decision Support System via Human On-Line Behavior Characterization and Machine Learning≫. In: *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*. 2018, pp. 9–14. DOI: 10.1109/AI4I.2018.8665694.

[31] Yousof Al-Hammadi, Uwe Aickelin, and Julie Greensmith. ≪DCA for bot detection≫. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. 2008, pp. 1807–1816. DOI: 10.1109/CEC.2008.4631034.

[32] Mattia Carletti et al. ≪Explainable Machine Learning in Industry 4.0: Evaluating Feature Importance in Anomaly Detection to Enable Root Cause Analysis≫. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. 2019, pp. 21–26. DOI: 10.1109/SMC.2019.8913901.

[33] Nico Görnitz et al. ≪Toward supervised anomaly detection≫. In: *Journal of Artificial Intelligence Research* 46 (2013), pp. 235–262.

[34] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. ≪Ganomaly: Semi-supervised anomaly detection via adversarial training≫. In: *Asian conference on computer vision*. Springer. 2018, pp. 622–637.

[35] Chun-Liang Li et al. ≪Cutpaste: Self-supervised learning for anomaly detection and localization≫. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9664–9674.

[36] MJ Desforges, PJ Jacob, and JE Cooper. ≪Applications of probability density estimation to the detection of abnormal conditions in engineering≫. In: *Proceedings of the institution of mechanical engineers, part c: Journal of mechanical engineering science* 212.8 (1998), pp. 687–703.

[37] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. «Outlier detection with kernel density functions». In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2007, pp. 61–75.

[38] Zheng Li et al. «Ecod: Unsupervised outlier detection using empirical cumulative distribution functions». In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[39] Markus M Breunig et al. «LOF: identifying density-based local outliers». In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[40] Yahya Almardeny, Noureddine Boujnah, and Frances Cleary. «A novel outlier detection method for multivariate data». In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[41] Yue Zhao and Maciej K Hryniewicki. «XGBOD: improving supervised outlier detection with unsupervised representation learning». In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[42] Sam Corbett-Davies and Sharad Goel. «The measure and mismeasure of fairness: A critical review of fair machine learning». In: *arXiv preprint arXiv:1808.00023* (2018).

[43] Finale Doshi-Velez and Been Kim. «Towards a rigorous science of interpretable machine learning». In: *arXiv preprint arXiv:1702.08608* (2017).

[44] Mattia Carletti, Matteo Terzi, and Gian Antonio Susto. «Interpretable anomaly detection with diffi: Depth-based isolation forest feature importance». In: *arXiv preprint arXiv:2007.11117* (2020).

[45] Qibo Yang, Jaskaran Singh, and Jay Lee. «Isolation-based feature selection for unsupervised outlier detection». In: *Proc. Annu. Conf. Progn. Health Manag. Soc.* Vol. 11. 2019.

[46] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ≪" Why should i trust you?" Explaining the predictions of any classifier≫. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[47] Lloyd S Shapley, Alvin E Roth, et al. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[48] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. ≪Human factors in model interpretability: Industry practices, challenges, and needs≫. In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW1 (2020), pp. 1–26.

[49] slundberg. *boston_waterfall.png*. `https://github.com/slundberg/shap`. Accessed: 2022-08-15.

[50] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. DOI: `10.48550/ARXIV.1602.04938`. URL: `https://arxiv.org/abs/1602.04938`.

[51] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. ≪Isolation forest≫. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[52] Sal Borrelli. *Isolating a Non-Anomalous Point*. `https://commons.wikimedia.org/wiki/File:Isolating_a_Non-Anomalous_Point.png`. Accessed: 2022-08-15.

[53] Sal Borrelli. *Isolating an Anomalous Point*. `https://commons.wikimedia.org/wiki/File:Isolating_an_Anomalous_Point.png`. Accessed: 2022-08-15.

[54] Pierre Baldi. ≪Autoencoders, unsupervised learning, and deep architectures≫. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 37–49.

[55] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation.* Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[56] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes.* 2013. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[57] Elad Plaut. *From Principal Subspaces to Principal Components with Linear Autoencoders.* 2018. DOI: 10.48550/ARXIV.1804.10253. URL: https://arxiv.org/abs/1804.10253.

[58] Michela Massi. *Autoencoder schema.* https://commons.wikimedia.org/wiki/File:Autoencoder_schema.png. Accessed: 2022-08-15.

[59] Lukas Ruff et al. «Deep One-Class Classification». In: *Proceedings of the 35th International Conference on Machine Learning.* Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 4393–4402. URL: https://proceedings.mlr.press/v80/ruff18a.html.

[60] Yisroel Mirsky et al. *Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection.* 2018. DOI: 10.48550/ARXIV.1802.09089. URL: https://arxiv.org/abs/1802.09089.

[61] Saket Sathe and Charu C. Aggarwal. «Subspace Outlier Detection in Linear Time with Randomized Hashing». In: *2016 IEEE 16th International Conference on Data Mining (ICDM).* 2016, pp. 459–468. DOI: 10.1109/ICDM.2016.0057.

[62] John S Gulliver and Roger EA Arndt. *Hydropower engineering handbook.* McGraw-Hill, Inc., 1991.

[63] Σ64. *Pumped Storage Hydroelectric Power Plant.* https://commons.wikimedia.org/wiki/File:Pumped_Storage_Hydroelectric_Power_Plant_01-ja.svg. Accessed: 2022-08-12.

[64] Md Mia, Md. Abidul Islam, and Mehran Islam. «Powering Offshore Structure Using Renewable Energy: A Review Study». In: *SSRN Electronic Journal* 05 (Jan. 2021), pp. 61–80. DOI: 10.2139/ssrn.3807353.

[65] photographer unknown U.S. Army Corps of Engineers. *Chief Joseph Dam*. https://commons.wikimedia.org/wiki/File:Chief_Joseph_Dam.jpg. Accessed: 2022-08-15.

[66] S Graham Kelly. *Advanced vibration analysis*. CRC Press, 2006.

[67] Jeremy Wright. *What Causes Machinery Vibration?* https://www.machinerylubrication.com/Read/25974/signs-tips-machinery-vibration. Accessed: 2022-08-08.

[68] ReliabilityWeb.COM. *The New Generation of Vibration Transmitters*. https://reliabilityweb.com/articles/entry/the_new_generation_of_vibration_transmitters. Accessed: 2022-08-15.

[69] M Attia Hili, T Fakhfakh, and M Haddar. «Failure analysis of a misaligned and unbalanced flexible rotor». In: *Journal of Failure Analysis and Prevention* 6.4 (2006), pp. 73–82.

[70] Molka Attia Hili et al. «Shaft misalignment effect on bearings dynamical behavior». In: *The International Journal of Advanced Manufacturing Technology* 26.5 (2005), pp. 615–622.

[71] Dragica Jošt and Andrej Lipej. «Numerical prediction of non-cavitating and cavitating vortex rope in a Francis turbine draft tube». In: *Strojniški vestnik-Journal of Mechanical Engineering* 57.6 (2011), pp. 445–456.

[72] A Baya et al. «Experimental investigations of the unsteady flow in a Francis turbine draft tube cone». In: *IOP Conference Series: Earth and Environmental Science* 12 (Aug. 2010), p. 012007. DOI: 10.1088/1755-1315/12/1/012007. URL: https://doi.org/10.1088/1755-1315/12/1/012007.

[73] K Vereide et al. «Strategy and application of new technologies for condition monitoring and predictive maintenance of hydro units». In: ().

[74] Farzaneh Ahmadzadeh and Jan Lundberg. «Remaining useful life estimation». In: *International Journal of System Assurance Engineering and Management* 5.4 (2014), pp. 461–474.

[75] Bernhard P. Bettig and Ray P. S. Han. «Predictive Maintenance Using the Rotordynamic Model of a Hydraulic Turbine-Generator Rotor». In: *Journal of Vibration and Acoustics* 120.2 (Apr. 1998), pp. 441–448. ISSN: 1048-9002. DOI: 10.1115/1.2893849. eprint: https://asmedigitalcollection.asme.org/vibrationacoustics/article-pdf/120/2/441/5592537/441_1.pdf. URL: https://doi.org/10.1115/1.2893849.

[76] Runqing Huang et al. «Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods». In: *Mechanical systems and signal processing* 21.1 (2007), pp. 193–207.

[77] Wilson Q Wang, M Farid Golnaraghi, and Fathy Ismail. «Prognosis of machine health condition using neuro-fuzzy systems». In: *Mechanical Systems and Signal Processing* 18.4 (2004), pp. 813–831.

[78] Dawn An, Joo-Ho Choi, and Nam Ho Kim. «Prediction of remaining useful life under different conditions using accelerated life testing data». In: *Journal of Mechanical Science and Technology* 32.6 (2018), pp. 2497–2507.

[79] Thyago P Carvalho et al. «A systematic literature review of machine learning methods applied to predictive maintenance». In: *Computers & Industrial Engineering* 137 (2019), p. 106024.

[80] Gian Antonio Susto et al. «Machine Learning for Predictive Maintenance: A Multiple Classifier Approach». In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), pp. 812–820. DOI: 10.1109/TII.2014.2349359.

[81] Rodrigo Barbosa de Santis and Marcelo Azevedo Costa. «Extended Isolation Forests for Fault Detection in Small Hydroelectric Plants». In: *Sustainability* 12.16 (2020). ISSN: 2071-1050. DOI: 10.3390/su12166421. URL: https://www.mdpi.com/2071-1050/12/16/6421.

[82] Alessandro Betti et al. «Condition monitoring and predictive maintenance methodologies for hydropower plants equipment». In: *Renewable Energy* 171 (2021), pp. 246–253.

[83] Nishchal K Verma et al. «Intelligent condition based monitoring of rotating machines using sparse auto-encoders». In: *2013 IEEE Conference on prognostics and health management (PHM)*. IEEE. 2013, pp. 1–7.

[84] mrbullwinkle et al. *What is Multivariate Anomaly Detector? (Public Preview)*. https://docs.microsoft.com/en-us/azure/cognitive-services/anomaly-detector/overview-multivariate. Accessed: 2022-08-15.

[85] Thomas Michael Welte et al. «MonitorX–Experience from a Norwegian-Swedish research project on industry 4.0 and digitalization applied to fault detection and maintenance of hydropower plants». In: *Hydro 2018-Progress through partnership, Gdansk, Poland, 15-17 October 2018* (2018).

[86] Christophe NICOLET, Nicolas Ruchonnet, and François Avellan. «One-dimensional modeling of rotor stator interaction in Francis pump-turbine». In: *Proceedings of the 23rd IAHR Symposium on Hydraulic Machinery and Systems, Yokohama, Japan*. Vol. 1. CONF. International Association For Hydraulic Research. 2006, pp. 1–15.

[87] Peter J Rousseeuw. «Silhouettes: a graphical aid to the interpretation and validation of cluster analysis». In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[88] *Python*. https://www.python.org/. Accessed: 2022-08-09.

[89] *Python Outlier Detection package*. https://pyod.readthedocs.io/en/latest/index.html. Accessed: 2022-08-09.

[90] *Python Streaming Anomaly Detection package*. https://pysad.readthedocs.io/en/latest/index.html. Accessed: 2022-08-09.

[91] *SHAP - SHapley Additive exPlanations package for Python*. https://shap.readthedocs.io/en/latest/. Accessed: 2022-08-09.

[92] Sudipto Guha et al. ≪Robust random cut forest based anomaly detection on streams≫. In: *International conference on machine learning*. PMLR. 2016, pp. 2712–2721.

[93] Hans-Peter Kriegel et al. ≪Outlier detection in axis-parallel subspaces of high dimensional data≫. In: *Pacific-asia conference on knowledge discovery and data mining*. Springer. 2009, pp. 831–838.

Appendix A

Other experiments

To keep the shape of this work clean we decided to only report the best working experiments in the main document and to move the others here in the appendix. The reasons we considered these models the less performing are:

- Some of the anomalies listed in 5.3.2 are not reported.
- The anomalies are not grouped together but instead, they have a uniform-like distribution, and that is very unlikely to happen given the properties of the system that generates the data as explained in section. 5.2.1.
- The fact that the results show signs of reduced robustness to errors or anomalies in the training data with respect to the models reported in the main part of this work.

A.1 RRCF

Robust Random Cut Forest (RRCF) [92] shows a more or less uniform distribution of anomalies, and that is very unlikely to happen. Additionally, the anomaly related to the structure vibration between March and April is not reported.

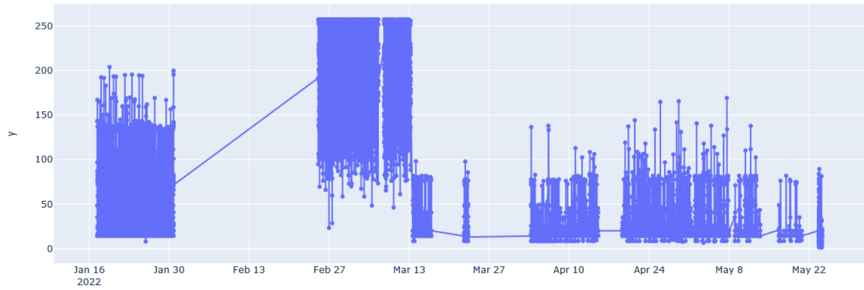


Figure A.1: Plot of the anomaly score from the RRCF model with unfiltered data only with High Power OS.

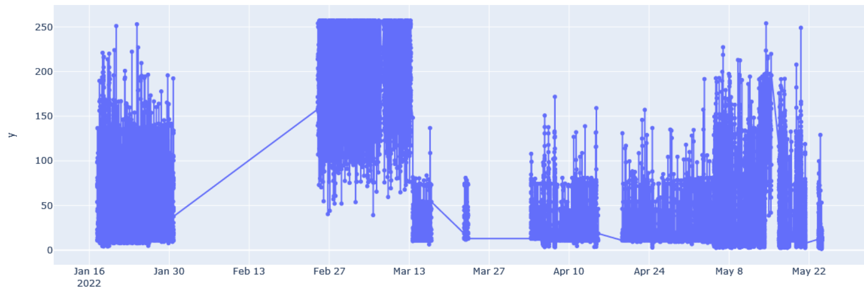


Figure A.2: Plot of the anomaly score from the RRCF model with unfiltered data only with all OS.

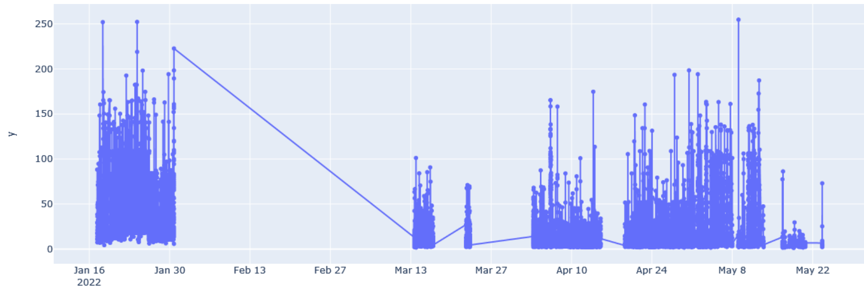


Figure A.3: Plot of the anomaly score from the RRCF model with filtered data only with High Power OS.

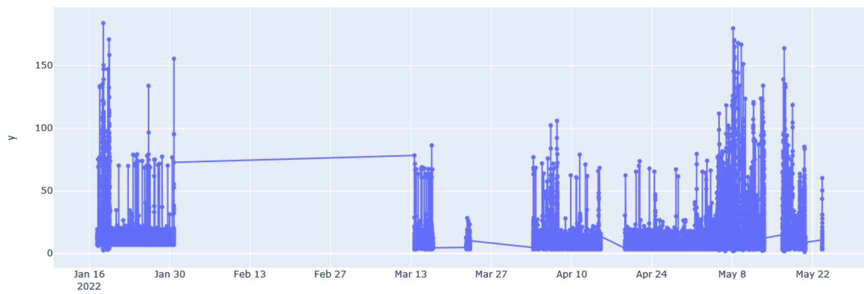


Figure A.4: Plot of the anomaly score from the RRCF model with filtered data only with all OS.

A.2 KDE

The KDE [37] implementation used in this work has the inconvenience that all the points in the training set are assigned to very low anomaly score (flat image towards the end), thus giving up most of the robustness against anomalies in the training set we appreciated in Auto-Encoders. Overall the results are very noisy and excessively responsive to the "unconnected sensors" anomaly.

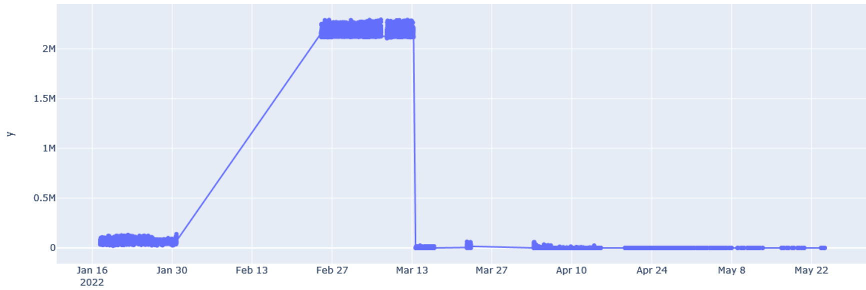


Figure A.5: Plot of the anomaly score from the KDE model with unfiltered data only with High Power OS.

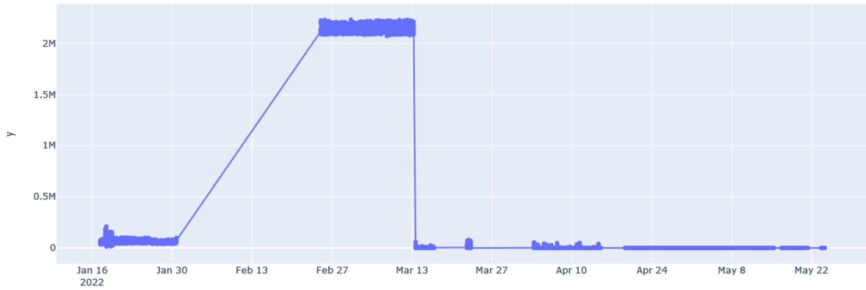


Figure A.6: Plot of the anomaly score from the KDE model with unfiltered data only with all OS.

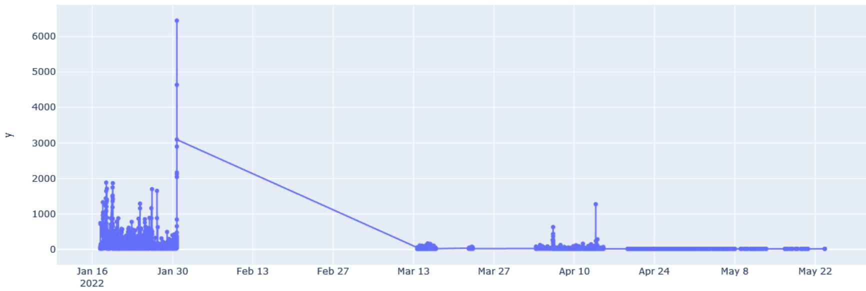


Figure A.7: Plot of the anomaly score from the KDE model with filtered data only with High Power OS.

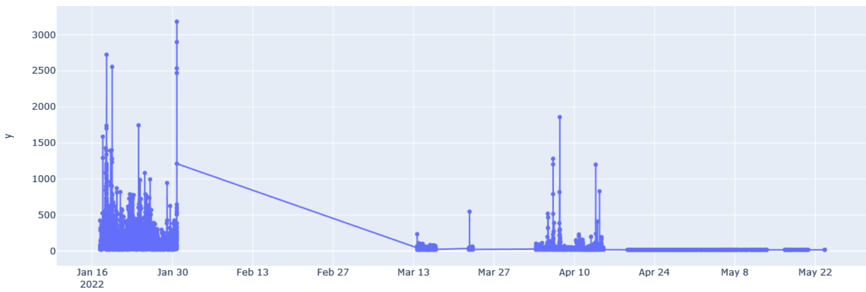


Figure A.8: Plot of the anomaly score from the KDE model with filtered data only with all OS.

A.3 SOD

Subspace Outlier Degree (SOD) [93] produces very noisy results, but the main problem is that the anomaly related to the unconnected sensors between February and March is not reported.

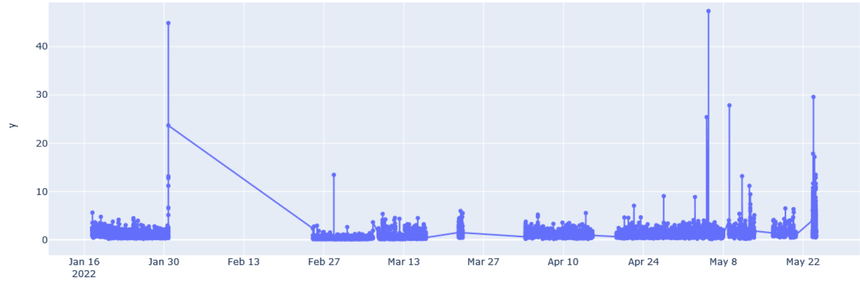


Figure A.9: Plot of the anomaly score from the SOD model with unfiltered data only with High Power OS.

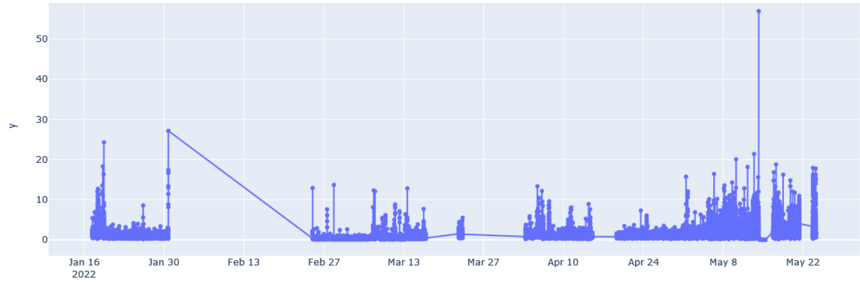


Figure A.10: Plot of the anomaly score from the SOD model with unfiltered data only with all OS.

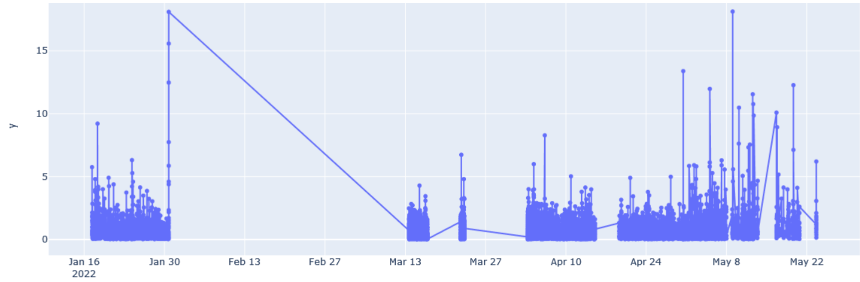


Figure A.11: Plot of the anomaly score from the SOD model with filtered data only with High Power OS.

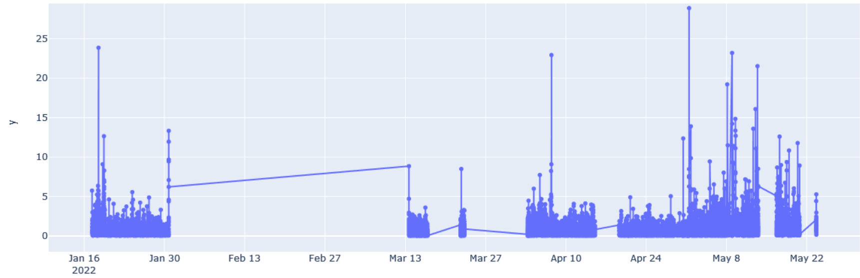


Figure A.12: Plot of the anomaly score from the SOD model with filtered data only with all OS.

A.4 ECOD

Empirical-Cumulative-distribution-based Outlier Detection (ECOD) [38] can indeed find the anomalies more or less effectively at all the stages of the experiment but has been placed here since there is a very small relative difference between the anomalies and the background, and a hypothetical threshold would be less stable than the ones in the main corpus of this work.

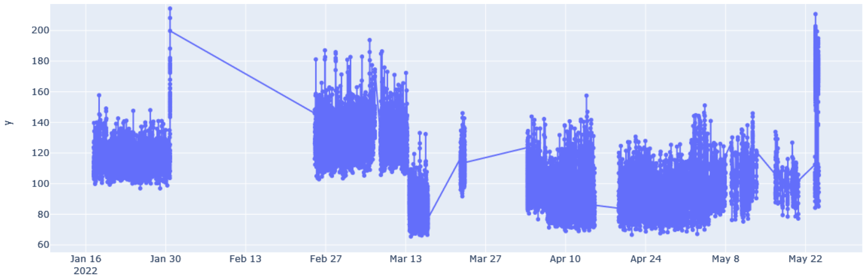


Figure A.13: Plot of the anomaly score from the ECOD model with unfiltered data only with High Power OS.

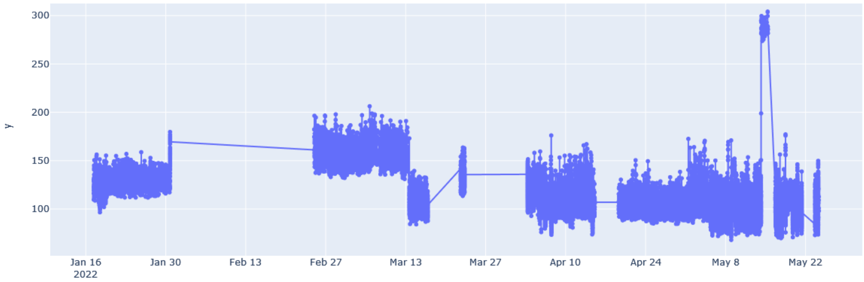


Figure A.14: Plot of the anomaly score from the ECOD model with unfiltered data only with all OS.

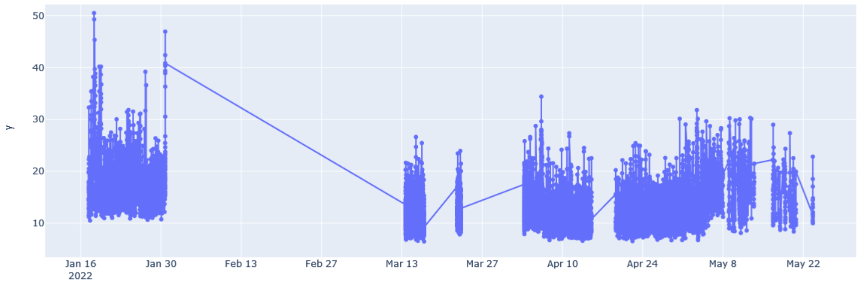


Figure A.15: Plot of the anomaly score from the ECOD model with filtered data only with High Power OS.

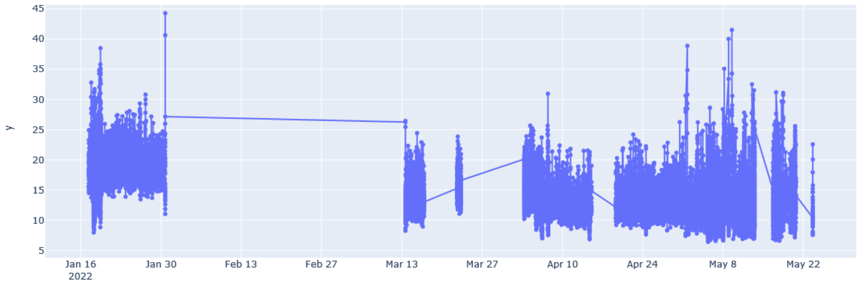


Figure A.16: Plot of the anomaly score from the ECOD model with filtered data only with all OS.