# Applications of anomaly detection for predictive maintenance at the JET tokamak

Andries Rosseau

Faculty of Sciences
Department of Physics and Astronomy
Ghent University
2018–2019

Promotor: Prof. dr. Geert Verdoolaege
Co-promotor: Prof. dr. ir. Sofie Van Hoecke

A thesis submitted in partial fulfillment for the degree of Master of Science in Physics and Astronomy at Ghent University.

Ghent University

# Applications of anomaly detection for predictive maintenance at the JET tokamak

Andries Rosseau
Faculty of Sciences
Department of Physics and Astronomy

Academic year: 2018 – 2019

Promotor: Prof. dr. Geert Verdoolaege
Faculty of Engineering and Architecture
Department of Applied Physics

Co-promotor: Prof. dr. ir. Sofie Van Hoecke
Faculty of Engineering and Architecture
Department of Electronics and Information Systems

June 15, 2019

# Permission to Use

Roeselare, June 15, 2019.

# Acknowledgements

I would like to take the opportunity and thank my promotors, Professor Geert Verdoolaege and Professor Sofie Van Hoecke, for the advice and guidance offered throughout the course of this thesis. Their constructive suggestions and encouragements were of key value during the creation of this work, and I would especially like to thank them for the very enjoyable period while working together on this thesis.

I am also greatly indebted to Azarakhsh Jalalvand and Tamir Mazaev for the insightful discussions on the data and results, and also for their critical questions that forced me to think deeper about a problem and look for new ways to approach it.

My sincere thanks go out to the researchers at JET, especially to Roger Buckingham, Jake Stephens and Frank Schoofs. Their expert insights and guidance on the research topics were indispensable for this work, and their time and efforts invested are very much appreciated.

I would also like to thank my fellow student, Robin Somers, for the pleasant discussions on nuclear fusion and machine learning, and for taking the time to proof-read my work while he himself also had a thesis to write.

Finally, I would like to thank Lieze Coussement for her unwavering support and encouragements, and my parents, for providing me with all the opportunities to get to where I am now.

# Abstract

The worldwide effort on fusion research aims to realize a means of producing clean and safe energy for future generations. At the JET tokamak, extensive research is being performed to help accomplish this goal, but as with all complex machinery, component failures occur. In this work, two failure cases at JET are addressed with the goal of predictive maintenance by means of anomaly detection and other machine learning techniques.

The first case concerns turbomolecular pump failures at the JET vacuum system. A solution for detecting unhealthy pump behaviour is proposed using semi-supervised anomaly detection based on time series data from sensor signals. Deviations from normal behaviour are flagged when incoming sensor data are considered too dissimilar to a pool of healthy training data. A first model using principal component analysis and multivariate Gaussian modeling is devised that uses the Mahalanobis distance to the center of the healthy distribution as an anomaly score. A threshold is applied to the anomaly scores, and samples with scores above this threshold are flagged. A similar approach is taken for a second model, based on auto-encoder neural networks. Instead of the Mahalanobis distance, the reconstruction error from the auto-encoder network is used, and a sliding time window approach is used to include time correlations. The network is again trained only on a pool of healthy data, so reconstruction errors will be larger for data deviating from this behaviour. An appropriate threshold is set, and error scores for time windows above this threshold are flagged. Both models show an increase in anomaly scores leading up to a strong anomalous peak representing the failure. The auto-encoder network, however, flags less false positives and shows a clearer distinction and transition between healthy and anomalous data. A discussion of the results and suggestions for implementation in fusion operations are provided, along with possible extensions of the model.

The second use case deals with the S1 current switch. As a switch ages, failures and slow operations occur more frequently. Based on the analysis of two voltage signals through time, a logistic regression model is trained to classify between good, slow and failure operations. The results of the classifier show promise, with F1-scores above 0.9 for all classes. Still, the model is trained and tested only on a small and unbalanced dataset. A semi-supervised clustering approach is therefore proposed to build a more robust classifier by combining the small labeled dataset with the rest of the unlabeled samples. This approach requires little human effort, while still making use of all available switch operation samples. Finally, a rudimentary strategy for predictive maintenance is proposed using the devised classifier and a degradation scoring system.

The results of both cases show potential for the use of machine learning in fusion operations and serve as an invitation to further investigate the merits of a data-driven approach for problem solving in device maintenance and fusion research in general.

# Table of contents

# Chapter 1

# Goal and motivation

The title of the thesis, ‘Applications of anomaly detection for predictive maintenance at the JET tokamak’, hints at the interaction of two major subjects: the first one is *data science*, through the topic of anomaly detection and predictive maintenance, and the second one is *nuclear fusion*, through the use of data from JET (Joint European Torus), the largest operational tokamak fusion device in the world. Both topics are scientifically engaging and fascinating to work on, but next to an academic challenge, data science and nuclear fusion might also have a significant impact on our society. The following few pages will go into some depth about the motivation for choosing these topics as the backbone of this thesis, and why they concern not only scientists.

## 1.1 The intersection of two worlds

### 1.1.1 The fossil world

At the end of the eighteenth century, the industrial revolution reshaped our society. The burning of coal to power steam turbines proved that parts of manual labor could be automated, and it ushered in a new age of growth. We moved away from living on farmlands and found a new home into ever expanding cities. Cars replaced horses, ships no longer needed wind to move forward, and when the Wright brothers pulled off their impressive feats mid-air, the world looked in awe. The sky was not even the limit, shown by the Apollo missions, when astronauts ventured beyond our blue planet. The world had mastered its usage of fossil fuels. Even now, oil, coal and gas remain the building blocks of our economy, and we have very meticulously been building our society around them for the last two centuries.

For a while, mankind was convinced that the apparent endless provision of fossil fuels would pave the way to achieve any goal we aspired to as a species. But with the continued large-scale usage of oil, coal and gas also came a drawback: the burning of fossil fuels is directly connected to global warming and the destruction of valuable ecosystems. When these ancient fuels are burned, carbon-rich byproducts enter our atmosphere and create a strong greenhouse effect, which slowly heats our globe to temperatures that – if we do not act on this information – will lead to the widespread occurrence of destructive phenomena like droughts, wildfires, storms and floods. Eventually this global warming will cause the mass extinction of many animal and plant species. Not only will this mean a catastrophic loss for life on this planet, it will also cause severe economic damage, and especially affect the weakest nations on earth. Migration caused by climate change will be on a scale that has never been experienced before, as will be the challenge trying to mitigate it. The rising of global sea levels will effectively reduce land mass, and weather patterns will be permanently altered. Clean drinking water will become increasingly scarce, and on top of that, a warmer, wetter world also means more chances for diseases to spread widely.

While this narrative might sound dramatic, the data does really point to these scenarios happening in the future [1]. That is, if mankind does not look for ways to avoid or mitigate the dangers associated with uncontrolled global warming. To help solve this universal problem, there are many paths one can pursue. One of them is to turn to technological advancements, for example in the energy supply sector. There is a high agreement amongst climate scientists that the energy supply sector is the greatest contributor to global greenhouse gas emissions [2], and thus advancements in this sector would have a significant impact. Nuclear fusion is a very promising candidate for a future renewable energy source, since fusion produces almost no $CO_2$, especially when compared to fossil fuels. Fusion has the added benefit of producing significantly less long-lived radioactive waste, compared to nuclear fission. The development of fusion is, however, still in the research phase, with scientists and engineers working hard on solving the remaining challenges that inhibit the production of practical fusion reactors. The International Thermonuclear Experimental Reactor – commonly known as ITER – is an ambitious international project currently under construction in the Provence region of France. There, the next generation of fusion experiments will be performed. The ITER project is not the first of its kind; it will draw heavily from the knowledge gained at other fusion projects, including another tokamak device, the Joint European Torus (JET). It is at JET that an impressive part of the work in fusion research has been – and still is – done. The past few decades of experiments have led to the accumulation of an extensive amount of data, and it is these data that may contain unexplored pathways to aid researchers in their future work.

### 1.1.2 The data world

With catchy terms like 'deep learning' or 'big data', every business nowadays wants to jump on the data-driven bandwagon. But many techniques for analyzing data have been around for ages. A lot of us are probably familiar with classical statistics, but even more advanced machine learning techniques like artificial neural networks were already rather well-known in the 1970s. Why is it then that data science is becoming so immensely popular only now? One likely reason is the exponential increase in computing power that made the efficient execution of many algorithms possible. But another, probably even more important reason, is the massive increase in the amount of data we produce and therefore is available for analysis. In 2018, search engines like Google processed around 5 billion searches per day [3]. In *one minute*, we send more than 150 million e-mails, watch more than 4 million YouTube videos, and generate more than

4 million likes on Facebook. People took more than 1.2 trillion pictures in 2017 alone, mainly with their smartphone cameras [4, 5, 6, 7]. These are just some of the dazzling numbers of data we produce, and these numbers are growing exponentially. It is not surprising then that companies like Google or Facebook invest heavily in data-driven solutions, and enhance their products with artificial intelligence that eagerly learns from the provided data. Although the often-cited analogy: 'data is the new oil', is not a very accurate one, it does grasp the underlying notion that data is becoming incredibly valuable in our economy. Organizations that are skilled at processing data seem to get an edge on competitors.

Besides using artificial intelligence to improve customer services and boost productivity, an increase in intelligent agents might also put jobs at risk. Carl Frey and Michael Osborne from the University of Oxford studied 702 occupational groupings and found that "47 percent of U.S. workers have a high probability of seeing their jobs automated over the next 20 years." [8] The advent of artificial intelligence has brought its merits, but also its share of new challenges. All parties involved must definitely handle the implications of a highly automated world with care.

## 1.2 Research goal

### 1.2.1 General outline

So where does this thesis fit in this global story? One of the main advantages of data science is that it is widely applicable in many fields. There are many examples showing that the combination of data science and existing expert knowledge form a great combination. This is exactly what we try to achieve with this dissertation; the data from years of fusion experiments at JET might deliver additional value by processing it with appropriate algorithms. In this work, we attempt to introduce some of the approaches from data science to fusion operations. An introduction to nuclear fusion and machine learning is presented, followed by an exploration of two use cases at JET: the turbomolecular pumps and the S1 current switches.

As a result of the analysis, an automated approach was devised to aid researchers in managing these recurrent problems. The focus of the adopted methods was on anomaly detection in the context of predictive maintenance to predict equipment failures and avoid a possible setback for fusion operations.

### 1.2.2 Turbomolecular pumps

The turbomolecular pumps at JET are used for creating a high vacuum for plasma operations, pumping away gas during or after operations. One particular model of pumps that has been modified for use in fusion research experiences frequent failures. To avoid such failures in the future, a model is built to give early indications of deviation from normal behaviour and to signal to operators that the pump might be working under suboptimal conditions. This is done to try and prevent a complete failure from happening. The model is kept as general as possible to provide the possibility of extending it to similar situations in future experiments.

### 1.2.3 S1 current switches

The S1 circuit breakers (switches) on JET interrupt the current flowing from the poloidal fly-wheel generator converter to the central solenoid that is driving the plasma current. When a switch approaches the end of its lifetime after many operations, its operations become more unreliable. We present a roadmap for predictive maintenance for the switches: from the automated classification of reliable and unreliable behaviour, to semi-supervised labeling of the data, and finally to the first steps of predictive maintenance.

By working through these cases, we hope that this thesis can show the potential of data science as a powerful framework supporting fusion operations, and spark further discussion in applying this framework to solve new exciting problems in research.

# Chapter 2

# An introduction to nuclear fusion

This chapter will introduce the reader to the concept of nuclear fusion. The focus here will be on the tokamak magnetic confinement approach and a deuterium-tritium fuel, and is by no means a complete account. For a more comprehensive introduction to nuclear fusion, *Plasma Physics and Fusion Energy* [9] by Jeffrey Freidberg, is recommended. Still, this section provides the main theoretical concepts to understand how fusion reactions work and how to build a working reactor. Some knowledge of Newtonian physics is assumed. A few quantum physics topics are touched upon, without going into too much detail as to not confuse the perhaps unfamiliar reader.

## 2.1 The fusion reaction

An operational fusion reactor harvests the energy of atoms. More precisely the nuclear binding energy between the protons and neutrons that make up these atoms.[1] Protons and neutrons are collectively also called nucleons. The simplest and most common atom in our universe is hydrogen, and it consists of one proton[2]. If the proton has an additional neutron attached to it, we call it deuterium[3]. Deuterium is also known as ‘heavy hydrogen’. If the deuterium manages to add yet another neutron, the result is the radioactive isotope called tritium. Given the right conditions, a deuterium (D) and tritium (T) atom in close proximity can fuse into a single neutron and a helium atom consisting of two protons and two neutrons, releasing a large amount of energy in the process:

$$\mathrm{D} + \mathrm{T} \;\longrightarrow\; {}^{4}\mathrm{He} + \mathrm{n} + 17.6\ \mathrm{MeV}. \tag{2.1}$$

It is exactly this release of energy that is of great interest to fusion research. There are other fusion reactions that release a similar amount of energy, but the deuterium-tritium reaction is the most probable candidate for an operational fusion reactor, due to optimal reaction conditions that are easier to obtain in a tokamak fusion reactor than for the other reactions. The

[1] Atoms also consist of electrons, but only the atomic nucleus is considered for now.

[2] By hydrogen, we mean the hydrogen-1 isotope, also called protium.

[3] Actually, deuterium without its electrons is called deuteron, but we will stick with deuterium for simplicity.

word ‘easier’ is perhaps a bit misleading, since optimal conditions are still hard to obtain, and sustaining these conditions long enough remains one of the main challenges in fusion research. What exactly is meant by ‘optimal conditions’ will be explained in section 2.2.5 on ignition.

### 2.1.1 Fusion or fission?

By merging two light elements, fusion exploits the nuclear energy between nucleons. But so does nuclear fission, by *splitting* heavy atoms like uranium. Then why does the fusion of deuterium and tritium also yield energy, instead of consuming it? It seems to be opposite to the process used in our well-established nuclear fission reactors. To find an answer to this apparent contradiction, we have to delve a bit deeper into nuclear physics. The opposing energy mechanisms are a direct consequence of the nature of the forces that bind the nucleons of atoms together. To see how these nuclear forces have seemingly different effects for different elements, the *binding energy curve* from Figure 2.1 is provided for different nucleon numbers. The nucleon number is the sum of the number of protons and neutrons, and since the masses of both proton and neutron are so similar, it is also known as the mass number $A$.

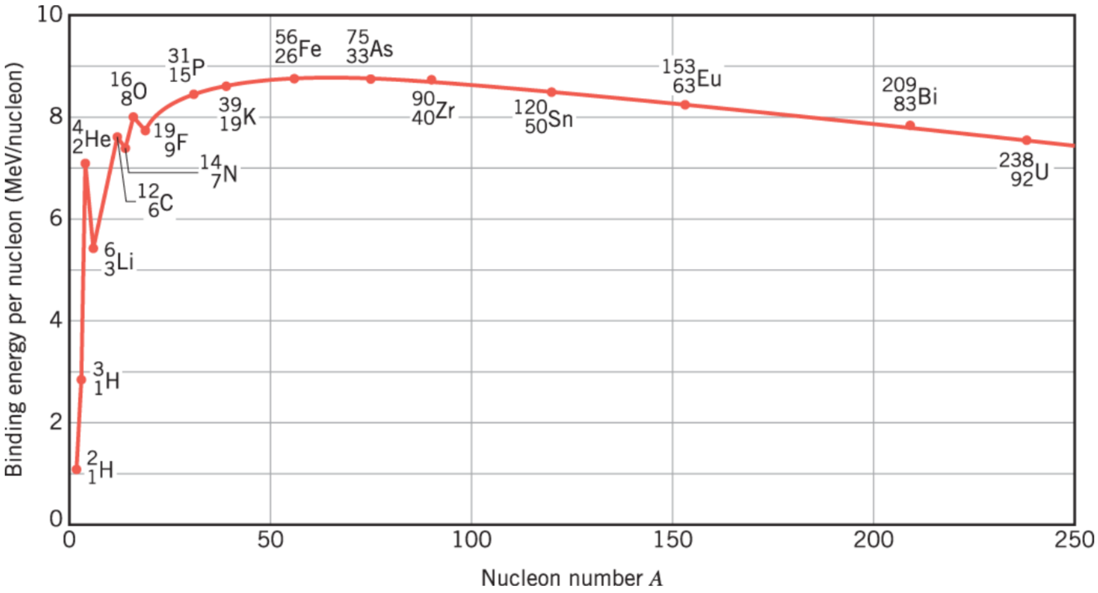

Figure 2.1: Binding energy per nucleon for different nucleon numbers $A$. The group of elements around $^{56}$Fe are the most stable. $^{4}$He corresponds to an interesting peak for lighter elements, and there is a decreasing tail for heavier elements.

The binding energy curve shows the binding energy per nucleon, not just for every chemical element of the periodic table, but also for isotopes of these elements. An isotope of a chemical element has the same amount of protons (which defines the name of a chemical element), but might have any amount of neutrons. *Stable* isotopes, however, are configurations of an element in which the amount of neutrons create a stable state for the atom. For example: $^{56}$Fe is a stable configuration of iron, with 26 protons and 30 neutrons, but $^{30}$Fe, with 26 protons and only 4 neutrons, or $^{98}$Fe, with 72 neutrons, are definitely not stable states; the ratio between protons and neutrons is too imbalanced. The more protons an element contains, the more neutrons it needs to stabilize it. This is because of the balance between the repulsive electrostatic force that tries to separate the equally charged protons, and the attractive nuclear force between all nucleons that compensates this repulsion. This is further discussed in 2.1.2.

A higher binding energy per nucleon means that the nucleons of an element are more strongly bound together, which means they are less prone to separation, e.g., by external impacts. It is apparent from Figure 2.1 that the most stable elements are the ones around the $^{56}$Fe isotope of iron, since they need about 8.8 MeV of energy per nucleon to be separated. At the higher end of nucleon numbers, we find elements like $^{235}$U and $^{239}$Pu. These isotopes are used in nuclear fission reactions to produce energy. When these atoms are split into lighter ones, e.g., by bombarding them with neutrons, the resulting products are elements that are closer to the iron group and have higher binding energies per nucleon, which make them more stable. The difference in binding energies for all the nucleons involved is released as kinetic energy (and gamma rays). It is this energy that is captured in a nuclear fission plant, and is then used for commercial electricity production. An example of a very typical reaction is:

$$\mathrm{n} \;+\; {}^{235}\mathrm{U} \;\longrightarrow\; {}^{140}\mathrm{Xe} \;+\; {}^{94}\mathrm{Sr} \;+ 2\mathrm{n} + \sim 200\ \mathrm{MeV}. \tag{2.2}$$

The amount of energy produced in a fission reaction is about 200 MeV. These energies are about ten times larger than our deuterium-tritium reaction energies, but keep in mind though that deuterium and tritium are about a hundred times lighter than uranium. So per unit mass of reacting input fuel, fusion creates more energy. When the energy outputs are compared, fusion processes produce about five times more energy from a gram of deuterium-tritium than fission reactions from a gram of uranium. Both fission and fusion reactions still produce about a million times more energy per gram of input fuel compared to fossil fuel reactions, since energy production through the burning of fossil fuels is caused by chemical reactions, not nuclear ones. Fusion and fission reactions produce very little $CO_2$ compared to the burning of fossil fuels, and fusion has the added benefit of producing significantly less long-lived radioactive waste compared to fission.

If we look at the other side of our binding energy curve in Figure 2.1, an interesting peak occurs at $^{4}$He, where the binding energy per nucleon is around 7.1 MeV. This is a large increase in binding energy from deuterium ($^{2}$H) and tritium ($^{3}$H), which have respective binding energies of about 1.1 and 2.8 MeV per nucleon. It is this interesting property that is used for the production of energy in fusion reactors. The merging of a deuterium and tritium atom into helium and a neutron is – just like the fission of uranium – energetically favourable, and the difference between the binding energies per nucleon is even larger than for fission transformations.

### 2.1.2 The binding potential for nucleons

Up until now, we have rather vaguely been talking about 'nuclear interactions', but to understand the process of fusion, a closer look at nuclear forces is required. There are actually two 'kinds' of nuclear forces: the weak nuclear force and the strong nuclear force. The strong force holds the protons and neutrons in atoms together[4]. It does so rather firmly when nucleons are in close proximity to each other, as is the case in atoms. The strong force is the strongest fundamental force known to us today, though it has to be noted that its full strength is most apparent *inside* nucleons, instead of *between* them. Nevertheless, the strength of this force is the reason why nuclear reactions involve energy differences that are so much larger than chemical reactions caused purely by the electromagnetic force.

The strong force does not have the ability to change the constituents that make up a nucleon, called quarks. In more simple terms: it cannot turn a neutron into a proton or vice versa. That is the domain of the weak force. The weak force is responsible for fundamental processes like beta decay, where a neutron in the nucleus of an atom decays to a proton, while emitting an electron and an ultralight particle called a neutrino[5].

For deuterium-tritium fusion, the relevant force out of the two nuclear forces is the strong force. Together with the electromagnetic force and some quantum mechanical effects, it determines the shape of the binding energy curve in Figure 2.1. Every atom internally has an interplay of the attractive short-range nuclear force between nucleons and the repulsive electrostatic Coulomb force between protons. The nuclear force is called short-ranged, because it is only felt between two nucleons when they are in very close proximity to each other (about the order of the nucleon radius). A schematic plot of the nuclear potential is shown in Figure 2.2 as a blue line.

For larger distances between the nucleon centers, there is no nuclear potential, but if the gap is closed, the potential eventually becomes attractive. The minimum of this potential is the sweet spot in which nucleons in our atoms operate and stay bound to each other. Although for the sake of generality no units are shown in Figure 2.2, the minimum of the nuclear potential between two nucleons typically occurs around a distance of 0.8 fm. If one tries to decrease the distance even further, a quantum mechanical effect, called Pauli exclusion, creates a very strong repulsion that inhibits two nucleons from occupying the same space.

The electrostatic Coulomb force, however, is long-ranged. This means that its influence is felt between two protons even if they are far away. The strength of the Coulomb force increases quadratically with decreasing distance, which means that protons in close proximity feel a stronger repulsion force. The Coulomb potential therefore goes as the inverse of the distance. The Coulomb potential is shown as a red line in Figure 2.2. Even though this potential favours a separation between protons, the total repulsion from the Coulomb potential is compensated by the total effect of the stronger nuclear potential for all nucleons in stable atoms, and the

[4] It is also more fundamentally responsible for interactions between quarks, the small constituents that make up our protons and neutrons, so the nuclear force between nucleons is actually a residual force.

[5] Actually, it is an antineutrino, but the difference is negligible here.

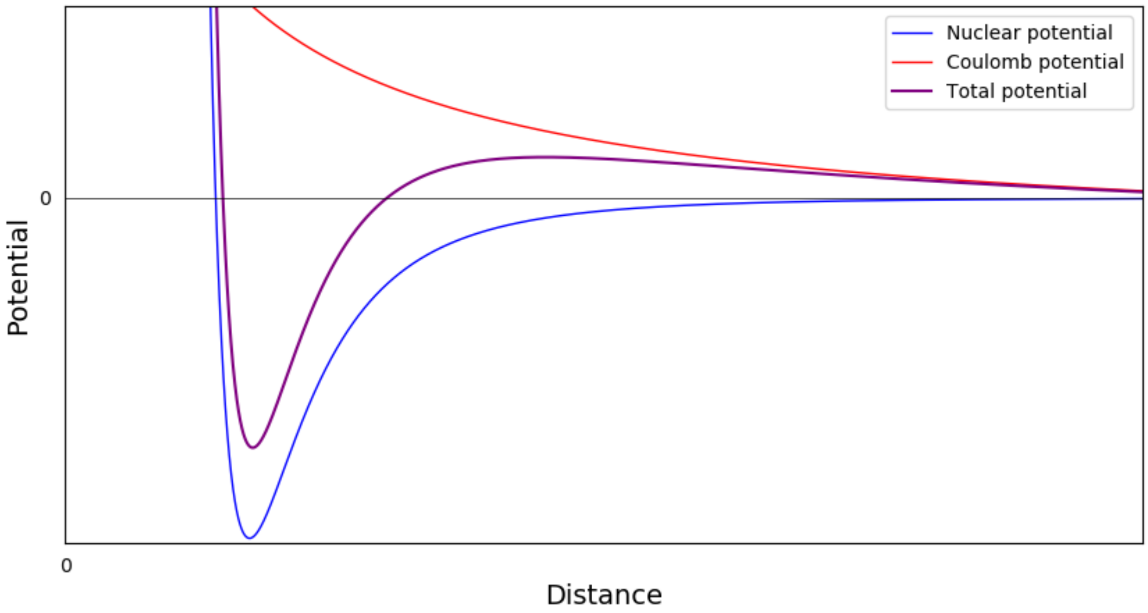

Figure 2.2: A schematic representation of the potentials present between nucleons.

nucleons are thus effectively bound. The sum of the Coulomb and nuclear potential is given as a purple line in Figure 2.2.

The interaction of the nuclear and Coulomb force is an intuitive start to understand how nucleons in a nucleus stay bound to each other and why some configurations are more stable than others. There is, however, more explaining to do when one asks questions like: ‘can we bind two protons together, given that we bring them close enough?’, ‘is a nucleus solely made out of neutrons more stable, since there are no Coulomb forces?’. These are some flavourful cases that are very interesting (e.g., the first question could lead us to the process of fusion in the Sun, and the second might trigger a discussion on neutron stars), but are out of scope for this introduction. The interested reader is referred to the literature on nuclear (astro-)physics.

### 2.1.3 Exploiting the binding energy curve

The previous discussion can now aid in understanding why our binding energy curve from Figure 2.1 has its typical inverted U-shape. The sum of the potentials of all nucleons is different for each element. For small elements, the total attractive potential by the strong nuclear force increases with increasing nucleon number. The repulsive forces are still easily compensated by the nuclear forces from all nucleons and the result is an even more strongly bound state. However, since the strong force is so short-ranged (only neighbours attract each other), the impact of its total binding potential declines compared to the increasing Coulomb potential for larger elements (all protons repulse all protons). There is an optimal balance around iron, but after that, the binding effect of the nuclear force saturates, and the repulsive force between protons starts to weaken the bonds.

How can we use this information to fuse deuterium and a tritium into helium? Since helium-

4 has a greater binding energy per nucleon, it would make sense that bringing deuterium and tritium together would initiate the energetically favourable reaction into helium. But in order to bring these atoms in very close proximity to each other, the strong repulsive force between the two positively charged nuclei needs to be overcome first. If the nuclei have sufficiently high kinetic energies, this repulsive force can be overcome and the attractive nuclear forces can kick in and deliver the final energy profit. Giving particles this amount of energy can be done by heating them up. This is why temperatures used in experimental fusion devices are typically above 100 million Kelvin, or more commonly expressed in the field of fusion as about 10 to 15 keV[6].

Now that we discussed why fusion reactions produce energy, we need to figure out how to balance the energy creation and energy loss processes of a reactor. This is discussed in the next section.

## 2.2 Power balance

We know what our fusion reaction looks like, and why it produces energy instead of consuming it. The next step is to know how to capture this surplus of energy and turn it into consumable electricity. When a fusion reaction between a deuterium and tritium core occurs, the 17.6 MeV of released energy in the end products is manifested as additional kinetic energy for the neutron and helium atom. This kinetic energy is apportioned inversely with mass by the laws of conservation of energy and momentum. Since the helium nucleus consists of four nucleons and the neutron is a single nucleon, the neutron gets four times as much energy as the helium core. Our 17.6 MeV is divided by five: four parts are given to the neutron (14.4 MeV) and one to the helium core (3.5 MeV). The very energetic neutrons from a fusion reaction leave the fuel mixture and are slowed down in a lithium envelope, transferring their energy to produce steam, which is then used to drive a turbine to produce electricity, just like a conventional power plant.

To create these energetic neutrons, the Coulomb barrier has to be overcome so that deuterium and tritium are in close proximity and have a chance to fuse. Overcoming this barrier costs energy, supplied as heat to the fusion fuel mixture. To create a functional fusion reactor, these energy costs cannot outweigh the energy gains. In other words: the output power $\mathrm{P}_{out}$ should be larger than the input power $\mathrm{P}_{in}$. There are some important processes that have an influence on the power balance. We will discuss the major ones.

### 2.2.1 Fusion power density

The first process is an obvious one: the energy production from fusion reactions. Each reaction produces $\mathrm{E}_f = 17.6$ MeV. If the reaction rate $\mathrm{R}_{12}$, which is the number of fusion collisions per unit time and per unit volume, is known, it can be multiplied with the reaction energy $\mathrm{E}_f$ to

[6] Very often, in high energy physics, the unnecessary Boltzmann constant $k_B$ is omitted in favour of directly using energies as temperatures with the conversion formula $\mathrm{E} = \mathrm{k}_B$ T. The unit of energy can be chosen freely, but mostly a multiple of the electronvolt (eV) is used. One eV equals 11 604 K.

obtain the fusion power density. The complete derivation of the reaction rate will not be given here, but some heuristic arguments are provided to make it plausible.

If there are more deuterium and tritium particles in a given volume, chances of fusion reactions happening will increase, so $\mathrm{R}_{12} \sim n_1 n_2$, with $n_1$ and $n_2$ the particle densities of deuterium and tritium. If the fusion reaction has a high cross-section $\sigma$, the reaction rate will increase, so $\mathrm{R}_{12} \sim \sigma$. If the relative velocity between the two particles is large, then it means there is more kinetic energy ($E = mv^2/2$) available and it is more likely that the Coulomb barrier will be surpassed, which explains $\mathrm{R}_{12} \sim v$. The fusion power density [W/m$^3$] is finally given by:

$$\mathrm{P}_f = \mathrm{E}_f \, n_1 n_2 \, \langle \sigma v \rangle. \tag{2.3}$$

We assume that particle velocities in our fusion reactor are in thermodynamic equilibrium over timescales longer than the nuclear collision time, and therefore follow a Maxwell distribution[7]. This means that the velocities of different particles can vary, even if the temperature of the mixture is uniform. We are interested in the power density for a statistically relevant amount of particles, so the right hand side of (2.3) is an average over all possible *velocities*. Only the cross-section times the relative velocity, $\sigma v$, depends on the relative velocity, so this quantity is averaged.

Equation (2.3) is an important one. The fusion power density should be maximized as much as possible. There is little that can be changed about $\mathrm{E}_f$ (it is already a very high energy release per reaction), but there exists an optimal partition for the particle densities and an optimal temperature that corresponds to the largest $\langle \sigma v \rangle$. Starting with the former: given an amount of deuterium-tritium gas, what is this optimal proportion between the two? If we define $n$ as the sum of the individual densities $n_1$ and $n_2$, we can replace $n_2$ with $n_2 = n - n_1$. If the derivative of the right hand side of (2.3) is taken with respect to $n_1$ and afterwards set equal to zero, the value of $n_1$ that maximizes the power density can be found. This simple calculation finds that $n_1 = n/2$. This means that a 50-50 mixture of deuterium and tritium is optimal. One can find the temperature that produces the largest $\langle \sigma v \rangle$ by doing a numerical simulation, assuming a Maxwell distribution, for every temperature. The results are shown in Figure 2.3. For the deuterium-tritium reaction, the maximal $\langle \sigma v \rangle$ lies around temperatures of 70 keV. However, this is not the temperature around which to operate our fusion reactor. There are other power balance considerations that have to be taken into account. The next paragraph will discuss an important energy loss process: Bremsstrahlung radiation, while 2.2.3 will discuss the general power balance. The optimal temperature turns out to be about 15 keV, given the ignition conditions discussed in 2.2.5.

Though the energy release per reaction cannot be altered, the 1/5th energy partition of the helium core stays inside the reactor and provides additional heating to our deuterium-tritium mix, which lowers the external heating costs once the fusion reactions have started (cf. section 2.2.3). The neutrons leave the fuel mixture to produce net energy. In a fusion reactor, a part

[7] Since Coulomb collisions are much more frequent than nuclear collisions in magnetic confinement reactors, this is a valid assumption.

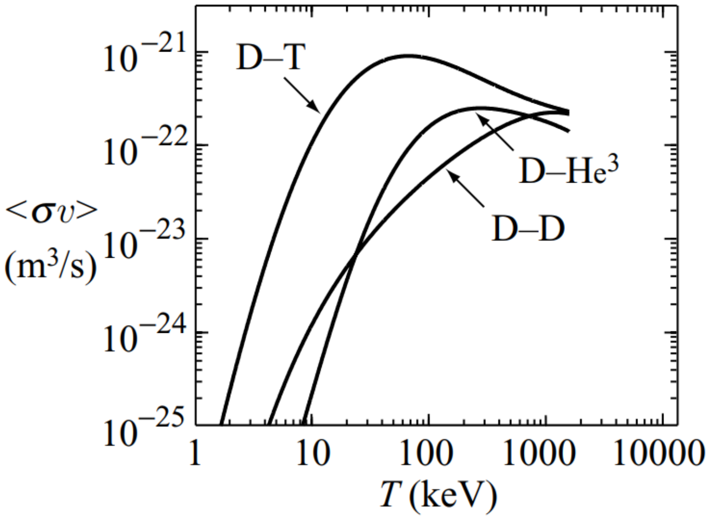

Figure 2.3: Velocity averaged $\sigma v$ for several fusion reactions as a function of temperature [9].

of this net energy could in theory be used for the remaining external heating power. In reality, this will most likely not happen, but it helps to see how the power output can be positive.

### 2.2.2 Radiation losses

When a force accelerates an electric charge, the charge sends out electromagnetic radiation. Some kinds of radiation, like cyclotron radiation, can be reabsorbed by fusion particles. In our reactor, positively charged deuterium, tritium and helium cores accelerate electrons, which then emit X-rays which are not reabsorbed. This process is called *Bremsstrahlung*, and is unavoidable. It means an unfortunate loss of power for our reactor: to keep the reactor environment at the desired temperature, compensation with extra heating power has to be provided. The Bremsstrahlung power loss, $\mathrm{P}_{Br}$, scales quadratically with the charge number and scales with the square root of the temperature:

$$\mathrm{P}_{Br} \sim Z^2 T^{1/2}. \tag{2.4}$$

A higher charge number means higher radiation losses, so it is desirable to avoid impurities in our fusion mix. One example is sputtered material from the wall of the fusion reactor. Another are helium cores, with $Z = 2$; they can be removed after transferring their surplus of energy to the deuterium and tritium cores to avoid part of the radiation losses.

### 2.2.3 The power balance equation

Finally, the power balance equation can be constructed. The rate of change of the total kinetic energy $W$ of the fusion mixture over time is given as:

$$\frac{dW}{dt} = \mathrm{P}_\alpha + \mathrm{P}_H - \mathrm{P}_L. \tag{2.5}$$

These quantities are not expressed as densities, but as energy rates over the total reactor volume. $P_L$ stands for the rate of energy loss, with contributions from processes like Bremsstrahlung radiation and thermal conduction. $P_\alpha$ stands for the total heating power of the produced helium cores, often called alpha particles. Although only a small percentage of the tritium-deuterium fuel undergoes fusion at a given time, the summed 3.5 MeV energies from the produced alpha particles are an important contribution to the heating component, since the operating temperature for fusion is 'only' about 15 keV. Finally, $P_H$ stands for the extra external heating power, when the alpha particle heating alone is not enough.

When the fusion reactor starts at room temperature, the deuterium and tritium atoms in the reactor will need heating to get to their ideal fusion operating temperatures. This means that

$$\frac{dW}{dt} > 0. \tag{2.6}$$

Since there is almost no fusion at these lower temperatures, the external heating power, $P_H$, will be large to compensate the losses and build up the desired temperature (or equivalently, the total kinetic energy). So $P_\alpha \approx 0$, and $P_H > P_L$. After a while, the desired temperature is achieved, which results in

$$\frac{dW}{dt} \approx 0, \tag{2.7}$$

and consequently

$$P_\alpha + P_H = P_L. \tag{2.8}$$

$P_\alpha$ is now contributing significantly, and together with $P_H$ compensates the loss term $P_L$.

### 2.2.4 Break-even operation

One could wonder what happened to the second part of the fusion power: the neutron energies. The neutrons leave the fuel mixture to produce electricity, and are not directly part of the fusion system anymore[8]. Going back to the introduction of this section, we name the *useful* energy output of the neutrons $P_{out}$, and the external heating power $P_{in}$. If $P_{out}$ is larger than $P_{in}$, we have – in theory – built a successful fusion reactor. One could call the moment that this $P_{out}$ becomes equal to $P_{in}$ 'break-even'. In fusion circles, though, 'break-even' is mostly preserved for a reactor where the *total* fusion power of the entire reactor (helium and neutron energies) outweighs the external heating input power. We will use this definition from now on. Note that $P_{in}$ is not a fixed value that is eventually reached by $P_{out}$: during the heating of the fuel, more and more fusion reactions occur, so $P_\alpha$ is gradually taking over a part of the heating and the need for external heating power decreases.

Fusion operations at JET have reached about two thirds of the break-even point. There is another, more difficult milestone, for fusion reactors. If this milestone is achieved, or near-achieved, a commercial fusion reactor could in principle be built. It is the milestone of ignition, discussed in the following paragraph.

[8] This is a simplified view, since the neutrons are actually also part of the tritium breeding process.

### 2.2.5 Ignition condition

If a fusion reactor could hold enough particles together at a sufficiently high temperature for a sufficiently long time, eventually the alpha particle heating would be strong enough as to completely replace the external heating power source. We could then turn off the external heating, and the fusion reactor would heat itself through the high energy alpha particles. We would only need to provide new deuterium-tritium fuel to replace the fused ones and keep the particle density high enough. This self-heating process in thermodynamic equilibrium is expressed through the power balance equation (2.5), with $dW/dt \approx 0$ and $\mathrm{P}_H \rightarrow 0$:

$$\mathrm{P}_\alpha = \mathrm{P}_L. \tag{2.9}$$

By ‘a sufficiently long time’, we mean that the characteristic time measure that indicates how fast the kinetic energy leaves the reactor when all heating is turned off, is sufficiently large. This time measure is called the *energy confinement time* $\tau_E$, and it is formally the characteristic scale of the exponential energy decay $e^{-t/\tau_E}$ in the reactor, as seen in Figure 2.4. The energy confinement time can be measured experimentally.

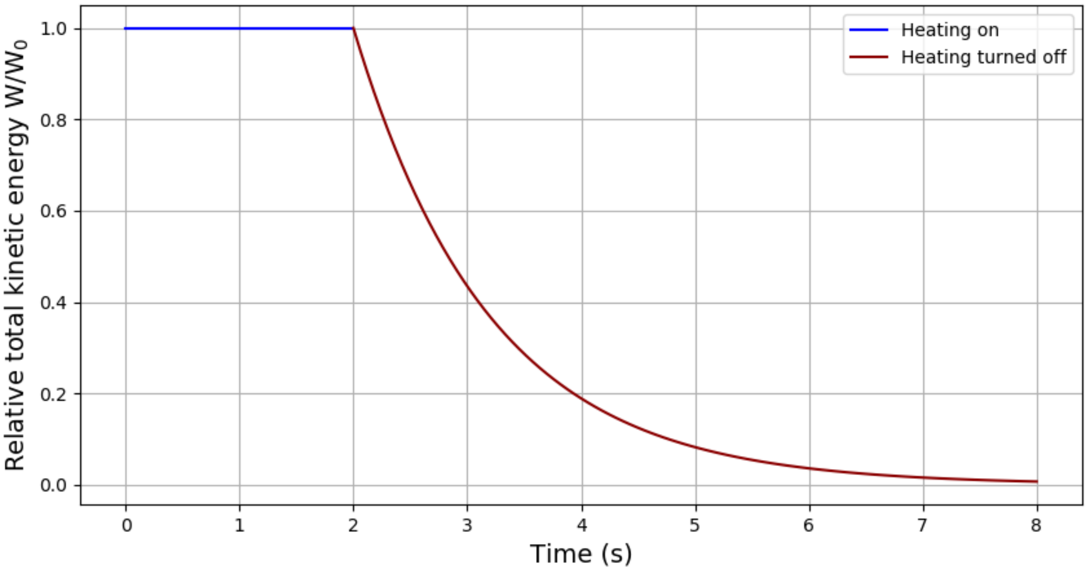

Figure 2.4: Course of the relative kinetic energy through time in a fusion reactor with and without heating. When the heating is turned off, the energy declines exponentially with a characteristic energy confinement time $\tau_E$. Here $\tau_E = 1.2s$, which is a realistic value for JET.

What is the optimal particle density, temperature and confinement time to reach ignition? The ignition condition of (2.9) provides the necessary ingredients. $\mathrm{P}_\alpha$ is known from the fusion power density equation (2.3) integrated over the volume of the reactor. The complete fusion reaction energy $\mathrm{E}_f$ is simply replaced by the alpha particle energy $\mathrm{E}_\alpha$. The power loss $\mathrm{P}_L$ is the rate of energy loss of the reactor. Even when the heating is on and the total energy $W$ stays at the same base level $W_0$, energy is being lost with a rate of $dW_L/dt$ as if there was no heating:

$$\begin{aligned} \mathrm{P}_L &= -\frac{d}{dt} W_L \\ &= -\frac{d}{dt} W \quad \text{(no heating)} \\ &= -\frac{d}{dt}\left(W_0\, e^{-t/\tau_E}\right) \\ &= \frac{W}{\tau_E}. \end{aligned} \tag{2.10}$$

The minus sign in (2.10) is due to the convention taken here that powers are expressed as their absolute values. We define $n$ as the sum of the two densities: $n \equiv n_1 + n_2$. If the fusion power output is maximized, the deuterium and tritium densities are the same, so $n = 2n_1 = 2n_2$. When thermodynamic equilibrium is assumed, the average energy of a particle is $3k_BT/2$, so the total energy of the reactor volume is:

$$\begin{aligned} W &= \frac{3}{2}\, k_B T \cdot N \\ &= \frac{3}{2}\, k_B T \cdot 2n\, V \\ &= 3\, n\, T\, V. \end{aligned} \tag{2.11}$$

In the last line, the Boltzmann constant was omitted in favour of the convention that temperatures are expressed in energy units. The relaxed assumption was made that the particle densities and temperatures are uniform throughout the reactor. In reality, this is not necessarily the case, but if the density and temperature profiles are positive and smooth, one can simply take the average of the density times temperature to make (2.11) general. Another important remark is that the factor 2 that appears before $n$ comes from also taking the electrons of the deuterium and tritium atoms into account. The reason for this will be discussed in the next section.

In light of the previous considerations, the ignition condition (2.9) becomes:

$$\mathrm{E}_\alpha\, \frac{1}{4} n^2 \langle \sigma v \rangle\, V = \frac{3\, n\, T\, V}{\tau_E}. \tag{2.12}$$

Divide by the volume and density $n$, rearrange the terms and multiply both sides by $2T$:

$$2\, nT\, \tau_E = \frac{24\, T^2}{\langle \sigma v \rangle\, E_\alpha}. \tag{2.13}$$

The ideal gas law, $pV = NT$, for the deuterium-tritium fuel is $p = 2nT$, so the ignition condition in the pressure, temperature and density space finally becomes:

$$p\, \tau_E = \frac{24\, T^2}{\langle \sigma v \rangle\, E_\alpha}. \tag{2.14}$$

This ignition condition draws a line in the $p\, \tau_E$ vs. $T$ space, shown in Figure 2.5. For the deuterium-tritium reaction, the minimum of $p\,\tau_E$ is 8.3 atm s, and corresponds to a temperature of about 15 keV.

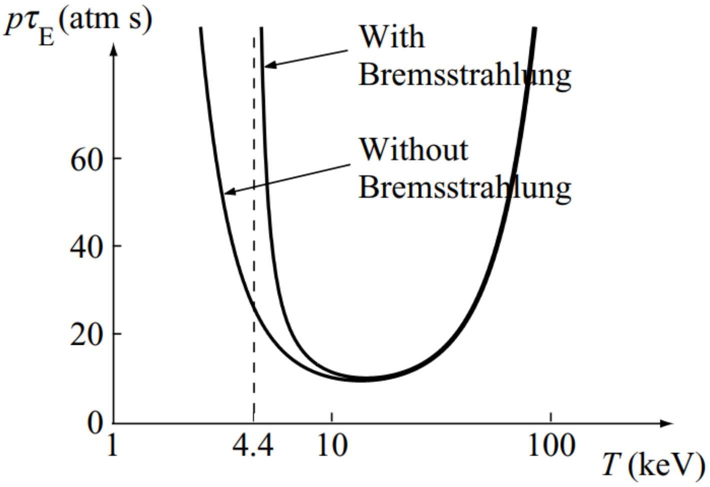

Figure 2.5: Critical $p\,\tau_E$ for ignition as a function of temperature [9].

### 2.2.6 Triple product

The minimum value of $p\,\tau_E$ is theoretically the easiest configuration to achieve in a magnetic confinement reactor. Since $p = 2nT$, it tells us something about the minimum of the product between $n$, $T$ and $\tau_E$. This $n\,T\tau_E$ product is called the *triple product*. For the deuterium-tritium reaction at the minimum of 15 keV, it is:

$$n\,T\tau_E = 3 \cdot 10^{21} \text{ keV s/m}^3. \tag{2.15}$$

If ignition conditions are to be achieved, a fusion reactor needs at least $3 \cdot 10^{21}$ keV s/m$^3$ as the value of the triple product. Other temperatures than 15 keV can be used to achieve ignition, as Figure 2.5 shows, but these increase the triple product value, so are harder to achieve.

## 2.3 Plasma physics for nuclear fusion

The previous sections dealt with why fusion reactions produce energy and what general power balance considerations there are to build a viable fusion reactor. Section 2.2 concluded that the triple product, $n\,T\tau_E$, should be at least $3 \cdot 10^{21}$ keV s/m$^3$. The easiest way to achieve this would be at a temperature of about 15 keV. Until now, deuterium-tritium fusion was almost exclusively discussed with the assumption of pure deuterium and tritium cores, and that they somehow get heated to their desired temperatures. But in reality, the fuel mixture gets inserted in the reactor as a neutral gas, which means that there is also an electron attached to each deuterium or tritium atom. A very important consequence of working at fusion temperatures of about 15 keV, however, is that the deuterium and tritium atoms are completely ionized due to the thermal heating energy. This means that our reactor does not contain a neutral gas anymore, but a *plasma*. The total number density of the particles in the reactor is now two times the density of the summed deuterium and tritium ion densities, so $n_{tot} = 2n = 2(n_1 + n_2)$. The ionization means that the electromagnetic force can be used to contain the charged ions and electrons in the reactor. This is the main principle behind magnetic confinement fusion reactors. This section will describe some of the characteristics of a plasma and how electromagnetic properties can be leveraged to keep the plasma inside of the reactor, and thus promote nuclear fusion reactions. This will be done through the lens of building a magnetic confinement tokamak reactor.

### 2.3.1 What is a plasma?

When a neutral gas gets energized through heating or a strong electromagnetic field, a partially or fully ionized gaseous substance with a significant fraction of quasi-free electrons can form. These electrons make the ionized gas electrically conductive. This state of matter is called a plasma. It is quasi-neutral, since some – or all – neutral atoms split into equal parts of positive ions and negative electrons, and the possibly remaining atoms were neutral to begin with. It features a collective behaviour, imposed by the long-range electromagnetic interactions in the plasma. Generally, a plasma moves as a whole, with typical length and time dimensions depending on several important plasma parameters. It is often called 'the fourth state of matter'.

There are many naturally occurring plasmas. In fact, it is the most abundant form of ordinary matter in the observable universe. Lightning is an example of a partially ionized plasma, and the interior of the sun is an example of a fully ionized plasma. The sun is a particularly interesting example, since its energy production also comes from nuclear fusion, though it involves a different fusion process called the proton-proton chain.

If an electric field is introduced in a plasma, electrons quickly rearrange themselves and the electric field is neutralized. As a consequence, no significant large-scale electric field can exist in the (unmagnetized) plasma. The ability to shield out an external electric field is a defining characteristic of a plasma, and it is called *Debye shielding.* To quantify the criteria that specify an ionized gas as a plasma, the electrical quasi-neutrality, the Debye length, the plasma frequency and the Debye sphere are discussed. This introduction will give an intuitive explanation of these concepts, but the interested reader is referred to *Introduction to Plasma Physics* by Francis Chen.

Quasi-neutrality

The charge density of a plasma is given by

$$\rho(\mathbf{r}, t) = \sum_i n_i q_i + n_e q_e, \tag{2.16}$$

with the subscript $i$ standing for (positive) ions and $e$ for electrons. The sum over possible different ion species will be omitted from now on, since only deuterium-tritium ions interest us for practical purposes. They both have the same charge number, $Z_i = 1$, so their densities can be summed and written as $n_i = n_1 + n_2$. When the charge density is averaged over a sufficiently large space and/or time, it turns out that the plasma is *quasi-neutral*, meaning that

$$\langle\rho\rangle = \langle n_i\rangle q_i + \langle n_e\rangle q_e \approx 0. \tag{2.17}$$

Consequently, $\langle n_i\rangle \approx \langle n_e\rangle \equiv n$. This $n$ is the ion and electron density that was used in the previous sections. Small local and temporal deviations from this electrical neutrality occur often throughout the plasma, but the electrostatic fields between charges react by restoring the neutrality. These disturbances and restoring processes have the plasma particles fluctuate around the equilibrium state, and these fluctuations show plasma-characteristic length and time scales, called the Debye length and plasma frequency.

### Debye length

The Debye length is intuitively the characteristic length-scale up to where plasma particles show deviations from charge neutrality. Above it, quasi-neutrality holds. This is not a hard cut-off: it is based on the *average* kinetic energy available to particles in the plasma. Since some particles have more energy than others, they can also deviate further from equilibrium. Globally though, the average maximum deviation length will be the Debye length. Another way to look at it, is if a positive test charge $q_t$ is placed in an infinitely large quasi-neutral plasma, electrons will rush to it to negate the charge. The bare potential of the test charge is

$$V_t = \frac{q_t}{4\pi\epsilon_0 r}, \tag{2.18}$$

where $r$ is the distance from the charge. But when the electrons in the plasma are gathered around the charge to negate it, the potential – now called the Debye potential – goes as[9]:

$$V_D = \frac{q_t}{4\pi\epsilon_0 r} e^{-r/\lambda_D}. \tag{2.19}$$

Here, $\lambda_D$ is the Debye length, and it is given by

$$\lambda_D^2 = \sum_s \frac{\epsilon_0 T_s}{\langle n_s\rangle q_s^2}, \tag{2.20}$$

with $s$ the different species in the plasma: the ions and electrons. If $r$ is smaller than $\lambda_D$, the potential practically follows the bare Coulomb potential from (2.18). If $r$ gets larger than $\lambda_D$, it decays exponentially. The Debye length is the transition length scale for the two regimes. In the case of a deuterium-tritium fusion plasma, the ion and electron contributions to the Debye length are equal (same temperature, average density and squared charge), so the total Debye

[9] No derivation provided here.

length is expressed as $\sqrt{2}$ times the electron Debye length:

$$\lambda_D^2 = 2\frac{\epsilon_0 T}{\langle n_e \rangle e^2} \tag{2.21}$$

The Debye length depends on the temperature of the plasma[10] and the particle density. If the temperature increases, particles have more thermal energy and can deviate more easily from their average positions. If the density increases, the deviation-suppressing background of plasma particles strengthens its grip: there are more opposite charges pulling the particle back.

If the Debye length is much smaller than the macroscopic length $L$ of the ionized gas container,

$$\lambda_D << L, \tag{2.22}$$

it can be called a plasma. Otherwise, quasi-neutrality is not guaranteed. A tokamak fusion device has a Debye length of about $10^{-4}$ m, and the radius of a tokamak is on the order of meters, so this criterion is definitely fulfilled.

Debye sphere

For the Debye length to be a statistically relevant concept, there needs to be a sufficient amount of particles in the sphere spanned by the Debye length. The Debye sphere is simply $4\pi\lambda_D^3/3$, and the amount of particles inside the Debye sphere is

$$N_D = n \times \frac{4\pi\lambda_D^3}{3}. \tag{2.23}$$

The amount of particles in the Debye sphere needs to be much larger than one ($N_D >> 1$). Note that from the definition of the Debye length, the amount of particles actually scales as $1/\sqrt{n}$, so if the density increases, the amount of particles in the Debye sphere decreases. This can be countered by a higher temperature. For typical densities ($n \approx 10^{20}$ m$^{-3}$) and temperatures ($T \approx 15$ keV) in a tokamak, this condition is fulfilled.

Plasma frequency

Just as the Debye length is the characteristic length scale of charge fluctuations around neutrality, the plasma frequency indicates the characteristic time scale. If some electrons deviate from equilibrium due to their thermal energy, and are all slightly displaced in one direction with respect to the remaining ions, the charge separation will create a temporary electric field that tries to restore the quasi-neutrality. Electrons will be attracted towards the original positions, but as they are accelerated towards the original position, they gain inertia and overshoot the equilibrium position. They will then fluctuate like a harmonic oscillator around the equilibrium

[10] *The* temperature of a plasma supposes thermodynamic equilibrium between all particles, which is a good approximation for fusion plasmas.

position. Electrons, not ions, fluctuate most, since they are so much lighter than ions and react much faster. The frequency with which an electron will fluctuate around the equilibrium position is called the plasma frequency $\omega_{pe}$, and is given by:

$$\omega_{pe}^2 = \frac{\langle n_e \rangle q_e^2}{m_e \epsilon_0}. \tag{2.24}$$

The plasma frequency expression for ions is just the same, and a total plasma frequency can be acquired by summing the squared contributions and then taking the final square root, but the mass of the ions in the denominator makes this contribution negligible so only the electron plasma frequency is used. The total plasma frequency becomes

$$\omega_p^2 \approx \omega_{pe}^2 = \frac{\langle n_e \rangle e^2}{m_e \epsilon_0}. \tag{2.25}$$

It is important to note that the plasma frequency does not depend on the temperature, only on the density. It indicates a fundamental time scale in a plasma. For an ionized gas to be qualified as a plasma, the plasma frequency should be much larger than macroscopic frequencies (e.g., the inverse confinement time or stability frequencies). This is the case in tokamaks with densities of about $10^{20}$ m$^{-3}$. A typical plasma frequency in a tokamak is about $10^{12}$ s$^{-1}$.

### 2.3.2 Motion of plasma particles in a tokamak magnetic field

At fusion temperatures, the deuterium-tritium reactor gas gets completely ionized. Since all the previous criteria are fulfilled, it is a plasma with quasi-neutrality. It is also a hot plasma, with an approximate thermodynamic equilibrium between ions and electrons at the appropriate scales. Although Coulomb collisions are, relatively speaking, rare in a hot plasma (collective, long-range Coulomb behaviour dominates), they still occur frequently enough to achieve thermodynamic equilibrium at time scales much shorter than periods between fusion reactions. One way to keep this hot plasma inside of a reactor is to confine it in an engineered magnetic field so that the outwards directed negative temperature and pressure gradient of the plasma does not diffuse the particles into the wall of the reactor. This diffusion might not only decrease the temperature and inhibit fusion reactions from occurring, but also damage the wall of our expensive reactor. In this section, the motion of a particle is discussed in the engineered magnetic field from a very prominent nuclear reactor configuration, the tokamak. The JET fusion device has a tokamak configuration, and so will ITER. The magnetic field is constructed one component at a time, until a solid confinement field has been built. The mathematical derivations are kept light, and only the main field components will be discussed, so as to focus on the conceptual idea behind confinement.

A tokamak is shaped like a torus[11]. The plasma particles are confined inside the torus by applying several magnetic fields. In Figure 2.6, a schematic drawing of a tokamak is presented.

[11] Many tokamak designs deviate in some way from the perfect torus form, but all close in on themselves in a circular shape in the toroidal direction

To understand the magnetic field lines in a tokamak, a short introduction to some relevant electromagnetic phenomena is provided.

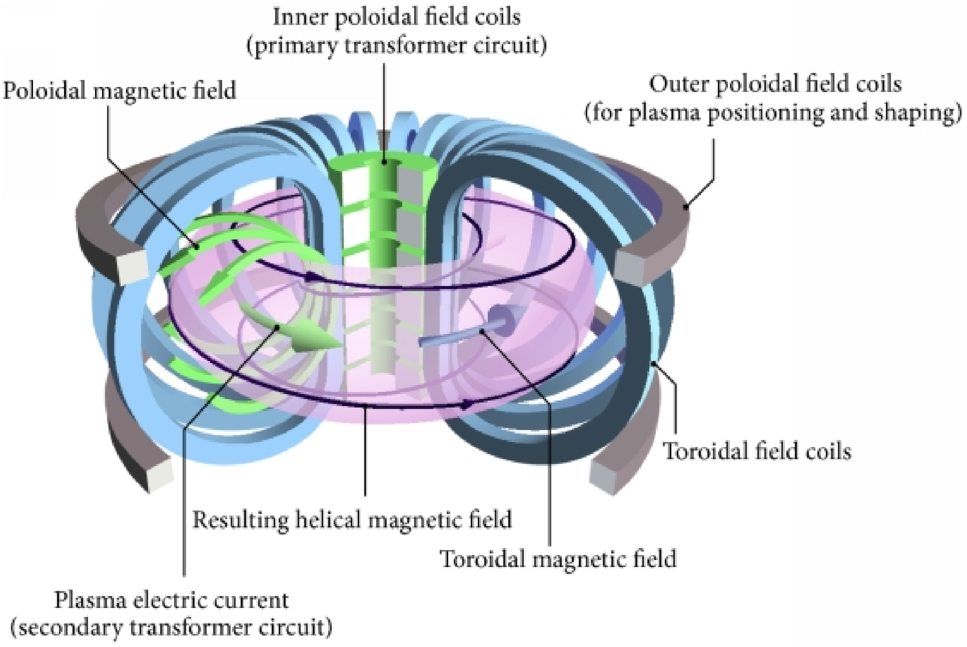
Inner poloidal field coils
(primary transformer circuit)
Poloidal magnetic field
Outer poloidal field coils
(for plasma positioning and shaping)
Toroidal field coils
Resulting helical magnetic field
Toroidal magnetic field
Plasma electric current
(secondary transformer circuit)

Figure 2.6: Schematic view of a tokamak and the main field coils with corresponding magnetic fields and resulting helical field [11].

Newton's equation of motion dictates that a particle in an electric and magnetic field obeys the following equation:

$$m\,\frac{d}{dt}\mathbf{v} = q\,\mathbf{E}(\mathbf{r},t) + q\,(\mathbf{v}\times\mathbf{B}(\mathbf{r},t)) + \mathbf{F}_{ext}(\mathbf{r},t). \tag{2.26}$$

For a homogeneous[12] and stationary[13] electric field, this reduces to

$$m\,\frac{d}{dt}\mathbf{v} = q\mathbf{E}. \tag{2.27}$$

From equation (2.27), it can be found that the field accelerates the particles in the plasma. The equation governing the position $\mathbf{r}$ is:

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\frac{q}{m}\mathbf{E}\,t^2. \tag{2.28}$$

If the field is homogeneous and stationary, the acceleration is constant.

[12] Homogeneous means that the field has the same strength and direction everywhere. This is also often called a uniform field.

[13] Stationary means that the field does not change through time.

Applying a homogeneous and stationary magnetic field $\mathbf{B}(\mathbf{r},t) = B_z\mathbf{e}_z$ in the z-direction[14] reduces equation (2.26) to

$$m\,\frac{d}{dt}\mathbf{v} = q(\mathbf{v}\times\mathbf{B}). \tag{2.29}$$

Through dot-multiplication with $\mathbf{v}$, it becomes clear that the kinetic energy remains constant throughout the motion, so a charged particle does not gain kinetic energy from applying a stationary magnetic field:

$$m\,\frac{d}{dt}\mathbf{v}\cdot\mathbf{v} = q\,(\mathbf{v}\times\mathbf{B})\cdot\mathbf{v} = 0 \implies \frac{d}{dt}(mv^2) = 0. \tag{2.30}$$

The velocity is split into components parallel ($\mathbf{v}_{\|} = \mathbf{v}_z$) and perpendicular ($\mathbf{v}_{\perp}$) to the magnetic field. The particle's motion is not altered in the parallel direction, since the cross-product $\mathbf{v}_z \times B_z\mathbf{e}_z = 0$. In the direction perpendicular to the magnetic field line, however, a cyclotron motion occurs, leading to the following circle motion around the cyclotron center:

$$m\frac{v_{\perp}^2}{r} = |q|v_{\perp}B_z, \tag{2.31}$$

with the cyclotron radius (also called gyroradius or Larmor radius) $r_c$ given by:

$$r_c = \frac{mv_{\perp}}{|q|B_z}. \tag{2.32}$$

This cyclotron motion, combined with the undisturbed motion parallel to the magnetic field, results in a helical motion of particles around the uniform magnetic field lines, shown in Figure 2.7.

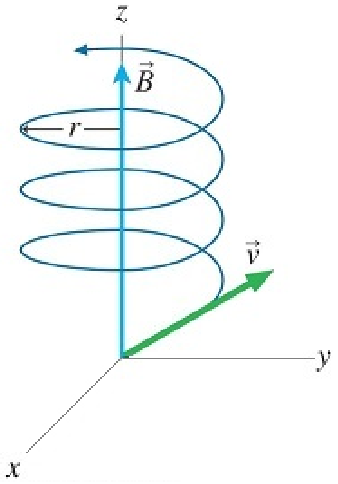

Figure 2.7: Movement of a charged particle in a homogeneous and stationary magnetic field.

The helical motion of charged particles in a uniform field is a very useful property for confining particles. If a sufficiently strong uniform magnetic field is applied to a plasma, the motion of the

[14] Any direction would have been fine, but the z-direction was chosen without loss of generality.

plasma particles is confined to a motion parallel to the magnetic field lines. The perpendicular motion is heavily suppressed, since the particles rotate around their guiding center in (very) small circles. If one could build an infinitely long tube with a strong uniform field inside, parallel to the main axis of the tube, (almost all) plasma particles could in theory move through the tube without colliding with the wall. In reality, though, one cannot build an infinite reactor, so a practical solution is to bend the tube in a circle and have it close in on itself, forming a torus. The long magnetic field lines from the tube are now replaced by toroidal field lines. This principle is one of the main mechanisms of confinement in a tokamak. A schematic view of the helical motion of the particles is given in Figure 2.8.

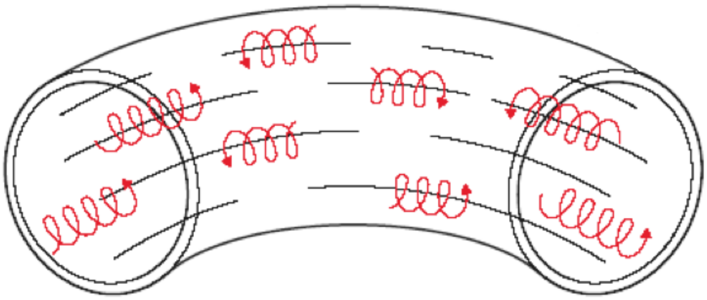

Figure 2.8: The motion of charged particles in a torus with idealized uniform toroidal magnetic field lines. The magnitude of the cyclotron radius is not to scale.

The disadvantage of bending the tube into a torus is that the magnetic field is no longer uniform. The non-uniformity of the toroidal field results in drift velocities that have the plasma particles drift into the wall once more. Without going into the particular details of the why and how of these drift velocities[15], one solution is to implement a poloidal magnetic field on top of the toroidal one, resulting in a combined helical-shaped magnetic field. This is illustrated in Figure 2.6. The helical field negates the outwards directed drift velocity. Still, this is not a complete solution for confining the plasma. There is still turbulence in the plasma, and magnetic instabilities can cause massive energy and particle losses to the plasma. Trying to mitigate these instabilities is still an ongoing research area. One simple reason why ITER is so large is that the volume of the plasma scales as $\sim R^3$, but the surface that confines it scales as $\sim R^2$, so with increasing $R$ the relative area through which particles and energies can escape through all possible mechanisms becomes smaller compared to the volume, which improves confinement.

The toroidal field is applied by running a current through the toroidal field coils. The toroidal field coils are placed symmetrically in the toroidal direction, but the shapes of the coils follow the *poloidal* direction around the torus. By running a high current through each of them, magnetic field lines form inside of the coil, as illustrated left in Figure 2.9. The combined magnetic fields of the coils produce the total toroidal field, much like a bent solenoid wrapped around the torus.

The poloidal field is induced directly by the current running in the plasma, and indirectly by

[15] The interested reader is referred to the literature on Hall drift, gradient drift and centrifugal drift velocities.

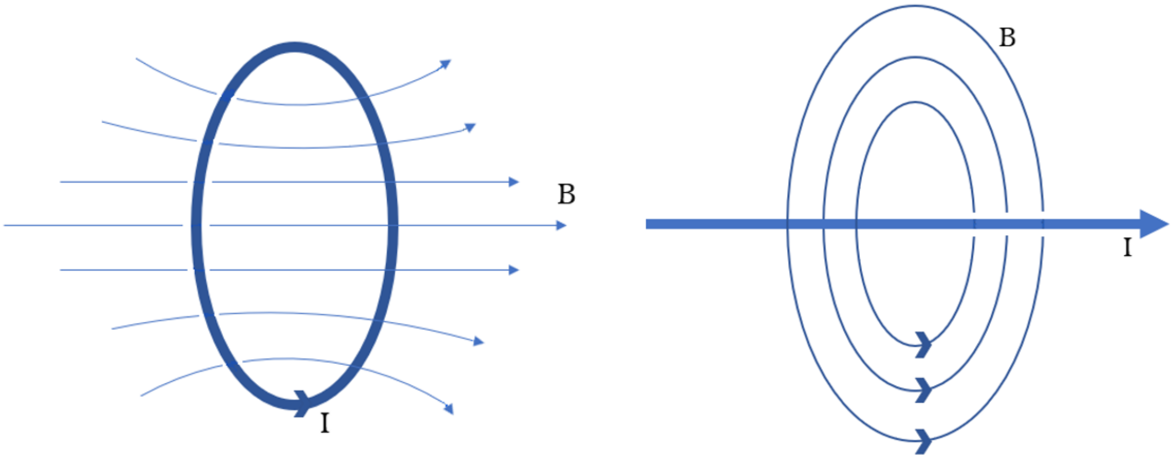

Figure 2.9: Left: A current running through a field coil induces a toroidal magnetic field. Right: A current running in the plasma induces a poloidal magnetic field.

the inner poloidal field coils that create the plasma current. The direct induction of the poloidal field through the plasma current is illustrated on the right side of Figure 2.9.

ITER will have toroidal magnetic fields up to 5.3 T, and the toroidal magnetic field strength of JET is about 3.5 T. The 5.3 T from ITER is considered a strong magnetic field: it is induced by running a current through superconducting coils. To give a scale of the size of the magnets involved, and of tokamaks in general, a scale model of JET and ITER is shown in Figure 2.10.

### 2.3.3 Plasma heating

To get a plasma heated to temperatures of about 15 keV, several heating mechanisms can be utilized on top of each other. The initial one, which gets the plasma to already impressive temperatures of 1 to 3 keV, is called *ohmic heating*. Ohmic heating exploits the resistivity of the plasma when there is a current running through it. This current is induced by the primary transformer circuit, which basically consists of a large central solenoid in the middle of the tokamak and the plasma itself as the secondary winding. When a very high current is run through the central solenoid, a strong magnetic field is created. The plasma will react to a *variation* in this magnetic field by producing its own current to try and negate the solenoid magnetic field. So changes in the central solenoid current increase the plasma current. In contrast to metallic conductors, the resistivity of a plasma actually decreases when the current increases, so there is a decreasing ‘return on investment’ to the heating from the plasma current. Still, a higher possible maximum current in the central solenoid can produce a higher plasma current and result in a larger ohmic heating, so at first glance one could think to ramp up the current to produce the desired temperatures. After about 3 keV, though, magnetic instabilities cause too much power loss, and increasing the plasma current is thus not viable anymore.

To bridge the gap between the 3 keV and the 15 keV temperatures where alpha self-heating takes over, one needs auxiliary heating processes. Two key auxiliary heating mechanisms are *neutral beam injection* and *radio frequency heating*. Neutral beam injections inject particles with

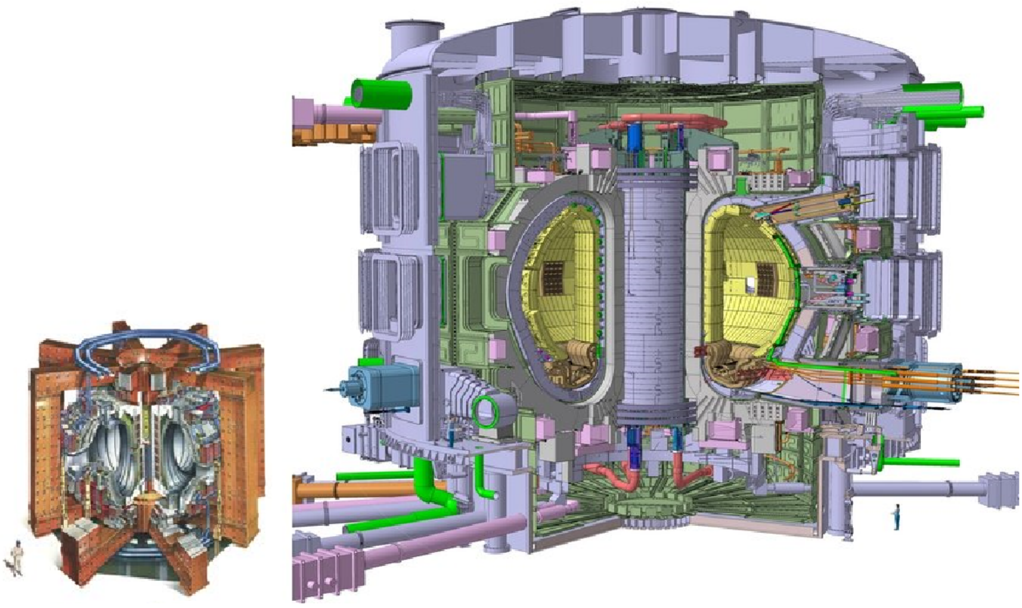

Figure 2.10: Left: the Joint European Torus (JET) tokamak. Right: the International Thermonuclear Experimental Reactor (ITER). JET's plasma volume is about 80 m$^3$ and has an energy output of about 16 MW, on the order of the break-even point. ITER's plasma volume is about 800 m$^3$ and will produce some 500 MW of power, ten times its input power. Illustrations and persons are to scale.

very high energies into the plasma. These highly energetic particles then distribute their energies throughout the plasma and heat it. An example of how to get such high energy particles, is to accelerate a positive ion in a particle accelerator, giving it a high kinetic energy, and before shooting it into the plasma, having it pass through a cold neutral gas so that it strips away an electron and enters the plasma as a neutral component. A typical energy of such a particle is about 150 keV.

Radio frequency heating works by sending an energetic wave into the plasma from an antenna in the wall of the reactor, and have the plasma absorb the energy when the wave is near a resonance frequency. This technique can target zones in the plasma by adjusting the energy of the waves to ultimately have the desired resonance frequency when it arrives in the right plasma zone. Electrons or ions can be targeted in the plasma, each with their own resonance frequencies.

### 2.3.4 What lies ahead?

ITER will try and show the world that a commercial fusion reactor is possible. To do that, they will have to get closer to ignition than has ever been done before[16]. The biggest challenge lies in holding a stable plasma long enough without losing the energy to the environment. Magnetic instabilities and turbulence, together with engineering and cost constraints, make this a very

[16] Ignition is not required for a practical fusion reactor. It can even be a desirable property of a reactor that it cools down on its own if the remaining heating power is turned off.

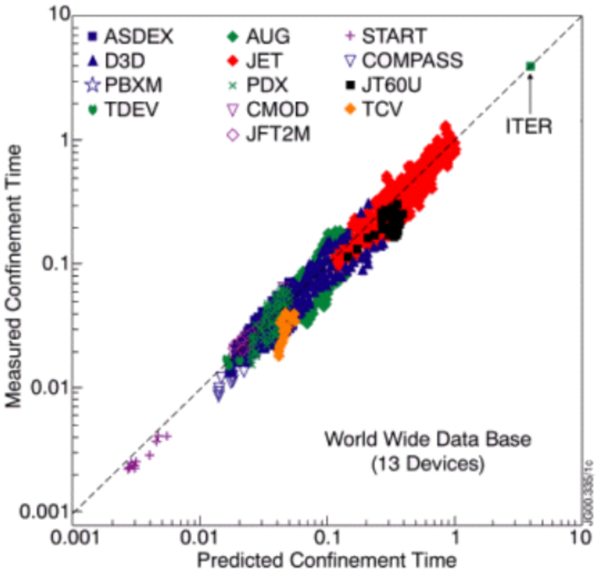

Figure 2.11: Based on data from previous fusion experiments and the parameters that will be used to build ITER, a very promising energy confinement time of about 3.7 s is predicted [12].

hard challenge to tackle, but researchers believe an energy confinement time of about 3.7 s will be achieved and about 500 MW[17] of power will be produced with only about 50 MW of input [12]. Confinement times and power ratios like this have no precedent, and fusion researchers base their predictions on extrapolations from similar fusion devices, like JET, to gain confidence in their predictions. A semi-empirical formula was devised that can predict confinement times for fusion devices. Based on the parameters ITER will have, this formula is used to predict the confinement time. The extrapolation of the energy confinement time, based on the data of many previous reactors, is shown in Figure 2.11.

ITER was already being conceptualized in the early eighties, when the JET reactor was not even finished. Just like then, fusion researchers are now thinking about what the next step will look like when ITER is in operation. This next step is generally called DEMO, and denotes the phase of fusion research that provides prototypes for commercial fusion reactors. Although the timeline and technical specifications vary, the objective is the same for all parties involved: building the nuclear fusion reactor that will demonstrate industrial-scale fusion. ITER foresees first plasma around 2025, and DEMO is foreseen to go into operation by 2050 [13]. These inspiring displays of engineering and international cooperation will hopefully lead the way to global, clean and safe fusion energy for all.

[17] A nuclear fission reactor produces about 1000 MW of power.

# Chapter 3

# An introduction to machine learning

This chapter aims at introducing the reader to the field of machine learning, with a focus on algorithms that revolve around the cases treated in the thesis. Machine learning can be considered an important subfield of data science, which is the research area that contains everything regarding the methods, processes and algorithms to extract insights and solve problems with data. There are no well-defined boundaries that cover what comprises machine learning and what not. For example, there are some who consider the field of statistics to lie outside the realm of machine learning. Here a more unifying view is adopted, where many of the underlying principles governing statistics apply to machine learning, and vice versa.

Machine learning can also be seen as part of the field of artificial intelligence. It is certainly true that machine learning – and especially deep learning – has dominated artificial intelligence research for quite some time now, but artificial intelligence is comprised of more than just the machine learning aspect. Optimal search algorithms, Bayesian inference models and game-playing are just some of the other topics approached by this lively field. Again, the boundaries between different techniques are vague, and cross-overs between subfields are frequent. An introduction to the field of artificial intelligence is given in *Artificial Intelligence: A Modern Approach*, by Peter Norvig and Stuart Russell [14].

## 3.1 Supervised learning

In machine learning, one often wants to model the function that governs the relationship between an input and an output, based on many example inputs and corresponding outputs, also called labels. Examples of labeled data are pictures with corresponding descriptions, a sentence from a foreign language with its corresponding translation, or hospital records from patients with corresponding health status. The modeling of the relationship between the input (e.g., pictures showing a dog or a cat) and the label (e.g., ‘dog’) by training on many examples, is called supervised learning. Models that use supervised learning can adjust their internal knowledge state by getting corrections from the label when they make a wrong prediction.

Most machine learning models start off knowing nothing about the real world. When a model has to predict if there is a cat or a dog in a picture, its internal state might produce the random result of 'dog: 0.4' and 'cat: 0.6'. If the prediction is wrong (the real label was 'dog'), the internal state will shift, so that next time it might predict a similar sample correctly. This process continues for all available training examples, often multiple times. Let's call $f(x)$ the true function that maps input samples $x$ to their true labels $r$, and let's call $g(x|\theta)$ the model function that tries to *approximate* the real underlying truth of $f(x)$. $\theta$ represents the internal state parameters of the function that have to be adjusted to resemble $f(x)$ as best as possible. The difference between the prediction $y = g(x|\theta)$ and the true label $r = f(x)$[1] is what makes the internal state of the algorithm change. If the difference is large, a significant internal shift happens. If there is only a small difference, or no difference at all (e.g. if the picture is a cat and was predicted as 'cat: 1.0'), the internal state will remain about the same. The measure of the difference between a prediction and the true label is determined by the *loss function*, $L(r, y)$, which is tailored to the problem at hand. The purpose of learning is to minimize this loss function for all training examples in the hope that when the model is presented with new, unseen data – where this time, the labels are unknown – it will still produce acceptable results. When a model performs well on data that it has not encountered before, we say that it is able to *generalize* what it has learned from the training samples. For this, it is assumed that the data from the training set adequately represents the data that will be encountered in the 'real world', otherwise the model will not know what to do with new samples and perform poorly. Minimizing the value of the total loss function for all training samples corresponds to finding a set of optimal internal parameters $\theta*$, expressed as:

$$\theta* = \operatorname*{argmin}_{\theta} \sum_i L(r_i, y_i), \tag{3.1}$$

with the sum over all training samples $x_i$ with corresponding labels $r_i$. Generally, the more training data, the better the optimization of the model parameters will be. Of course, much depends on the choice of the model, and every model has its limitations; there is unfortunately no one-size-fits-all solution, in machine learning known as the 'No Free Lunch Theorem' [15].

Supervised algorithms can broadly be separated into two categories: classification algorithms and regression algorithms. They mainly tackle different kinds of problems, but they both work by training on examples with specified labels to minimize a loss function.

### 3.1.1 Classification

The example of recognizing dogs and cats that has been used up to now is an instance of a *classification problem*. Classification algorithms take as input several variables (also called parameters or features), and based on these variables determine what class a sample belongs to. Classes are discrete categories, like 'dog' and 'cat' in our binary classification example.

One simple but important classification algorithm will be discussed here, called *logistic regression*[2]. It will serve as a way to introduce some important concepts in machine learning.

[1] Actually, $r = f(x) + n$, where $n$ is the real-life noise on the true function. It is often assumed to be Gaussian.

[2] Make no mistake, logistic regression is a classification algorithm. Admittedly, it is a confusing terminology.

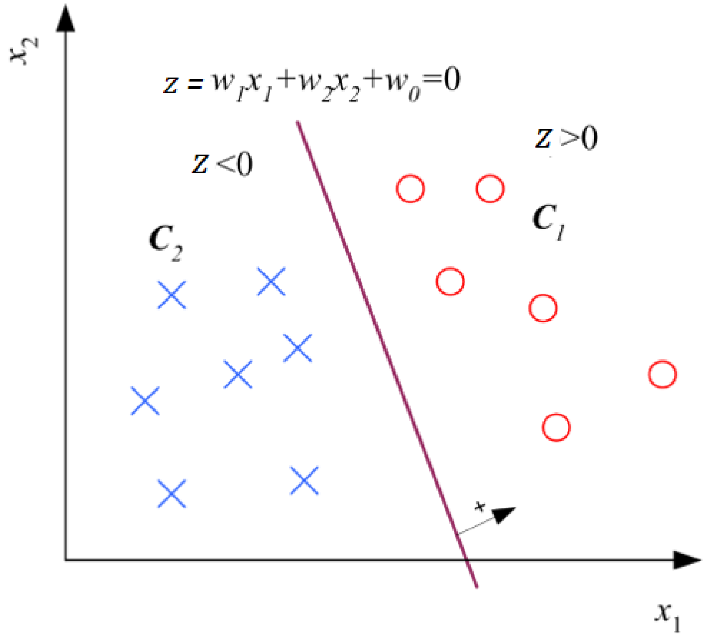

Figure 3.1: Training samples of two classes, $C_1$ and $C_2$, are to be separated by a straight line.

A simple binary classification problem will be considered. Samples from two classes, $C_1$ and $C_2$, must be classified correctly based on two available features, $x_1$ and $x_2$. To distinguish one from the other, an optimal boundary line[3] between the two classes is sought. This is illustrated in Figure 3.1.

When a boundary line is found that separates the labeled training samples, new samples belonging to either $C_1$ or $C_2$ can then be classified according to their position relative to this line. The line – or hyperplane in more dimensions – is also called the *decision surface*, and can be described by the analytical expression of a flat surface in feature space. Here, this is just the expression of a straight line in the two-dimensional $x_1x_2$ space:

$$z(x_1, x_2|\mathbf{w}) = w_1x_1 + w_2x_2 + w_0 = 0, \tag{3.2}$$

where the elements of the vector $\mathbf{w} \equiv (w_0, w_1, w_2)$ represent the inner knowledge state $\theta$ of the model. We have to adjust the weights $w$ by minimizing a loss function until the model correctly separates all training samples, like Figure 3.1 shows. Perfectly separating all samples is only possible if a straight line can be drawn between the samples. If this is not possible, the resulting separation line will be the best possible fit to the data. We call this strategy *linear classification*, since the decision surface contains only linear terms[4]. To gradually minimize the loss function, the separation boundary can be placed anywhere as a start. This initial boundary predicts a

[3] In two dimensions, this separation boundary is just a simple straight line, but for more dimensions, the boundary is more generally called a hyperplane or hypersurface.

[4] More complex decision surfaces can be considered by introducing higher order terms, or any other non-linear function, but the space in which to minimize the loss function gets larger, which brings the 'curse of dimensionality'. Support vector machines, another branch of machine learning techniques, have an ingenious way of dealing with this, called the kernel trick.

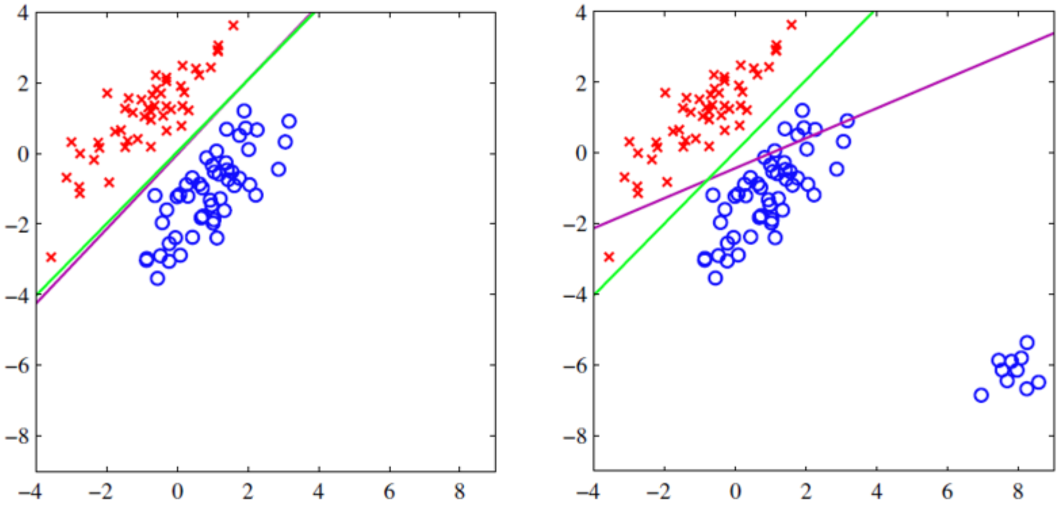

Figure 3.2: Two examples of separation boundaries obtained by a squared distance loss function. The green line represents the true separation between classes, the purple line is the one obtained by gradient descent with the squared distance loss. When outliers are introduced (right figure), they skew the boundary and create a model that will produce poor results on new samples.

first estimation of the labels of the training samples, and divides them into two sectors, $z > 0$ and $z < 0$. Since the first initialization of the decision boundary will probably get many samples wrong, an update in the right direction is required. An intuitive idea would be to use the sum of the squared distances to the decision surface for every sample as the loss function to be minimized:

$$L_{tot} = \sum_i z_i^2. \tag{3.3}$$

The negative gradient of this loss function can be used to adjust the weights in the direction of the minimum of the loss function:

$$\mathbf{w} \longrightarrow \mathbf{w} - \eta \, \nabla_{\mathbf{w}} L_{tot}, \tag{3.4}$$

where $\eta$ is called the *learning rate*, which determines how fast the descend to the minimum of the loss function should be. A too high learning rate might mean we overshoot the minimum at every update; a too small learning rate might mean it takes forever to approach the minimum, or for certain loss functions it might get stuck in a suboptimal minimum. This updating of the weights (and thus of the decision surface) is done until the loss function has been minimized. This strategy is called *gradient descent*. When the algorithm has converged to a solution, it is supposed to be ready to classify new and unseen samples. However, we have to be careful: our squared distance loss function was actually not a reliable option for our classification problem. It is not robust to outliers, and it punishes points that are classified ‘too well’, as shown in Figure 3.2, since their squared distances are also adding to the loss function and thus misdirect the decision surface.

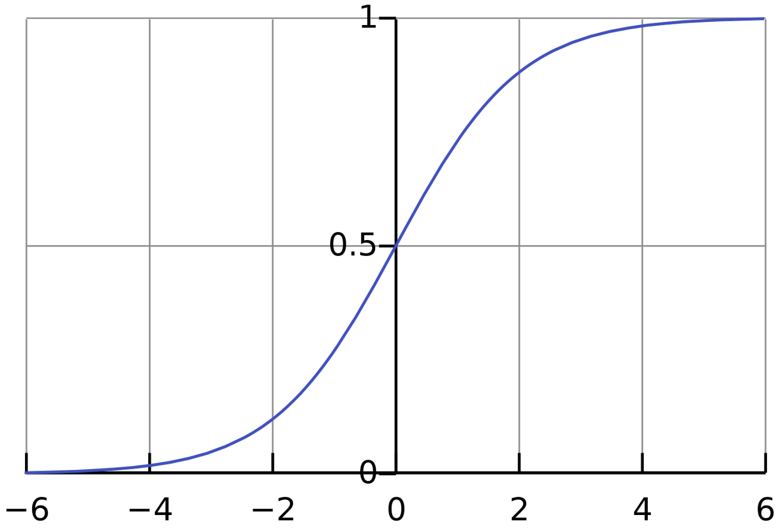

Figure 3.3: The logistic function $g = \dfrac{1}{1+e^{-z}}$, also called a sigmoid.

To find a more robust loss function, an update to the simple decision boundary classification algorithm can be implemented. Until now, samples were classified as $C_1$ or $C_2$ based on their location compared to the decision surface. A measure of how 'rightly' or 'wrongly' samples were classified, was until now based on the squared[5] distance to that decision surface. The loss function based on this intuitive measure turned out to be unreliable, so another measure to express the certainty of the classification prediction has to be introduced. It has to be a differentiable measure, so the loss function can be minimized during each update, and it has to be robust. One simple trick is to superimpose a logistic function $g = (1 + e^{-z})^{-1}$, shown in Figure 3.3, on top of the distance $z$ to the decision surface, so $g \equiv g(z(x_1, x_2|\mathbf{w}))$. This effectively reduces the infinite range of possible distances to the interval between 0 and 1, and introduces a variable $p = g(z)$ that softens the boundary between the two classes. $p$ represents an approximation of the probability that a sample belongs to the $C_1$ class on the $z > 0$ side of the decision surface, and $1 - p$ represents the approximate probability of belonging to the other $C_2$ class. To convert this soft decision boundary back into a discrete classification, we can assign samples with $p > 0.5$ into $C_1$, and samples with $p < 0.5$ into $C_2$.

The introduction of the logistic function and the interpretation of $g(z)$ as approximate probabilities[6] $p$ is of little use if we just convert the probabilities back into discrete classifications. Luckily, the logistic function values are very useful for defining a robust loss function. This loss

[5] One could use the absolute distance as a measure, and only sum the losses for wrongly classified samples so that the summed loss function would still be differentiable and the problem of punishing overly good predictions would be avoided. This is called the perceptron loss function, but has the problem that the boundary obtained is not optimal, and the solution does not converge if the classes are not linearly separable.

[6] This interpretation of probabilities has a statistical grounding that is out of scope here. The interested reader is referred to the literature on maximum likelihood estimation.

function is called the *cross-entropy loss*, or the *negative log-loss*. It is given by:

$$L_{cross-entropy} = -\sum_{c} r_c \; log \; p_c, \tag{3.5}$$

where the sum over all classes $c$ (here $C_1$ and $C_2$) is taken. The total loss for all training samples can then be obtained by summing this expression for every sample. The true labels, $r_c$, are discrete labels, so they equal 0 or 1 (e.g., '$r_1$: 0' and '$r_2$: 1'). The predicted probabilities, $p_c$, represent a real number between 0 and 1. Only one $r_c$ equals 1, the other equals 0, so the cross-entropy loss for a sample could in principle be reduced to

$$L_{cross-entropy} = -\, r_c \; log \; p_c \qquad (\mathrm{y_c} = 1), \tag{3.6}$$

but this function is not differentiable, which is a requirement for a practical loss function. In Figure 3.4, the loss function is shown for the true label $r_c$ equal to 1.

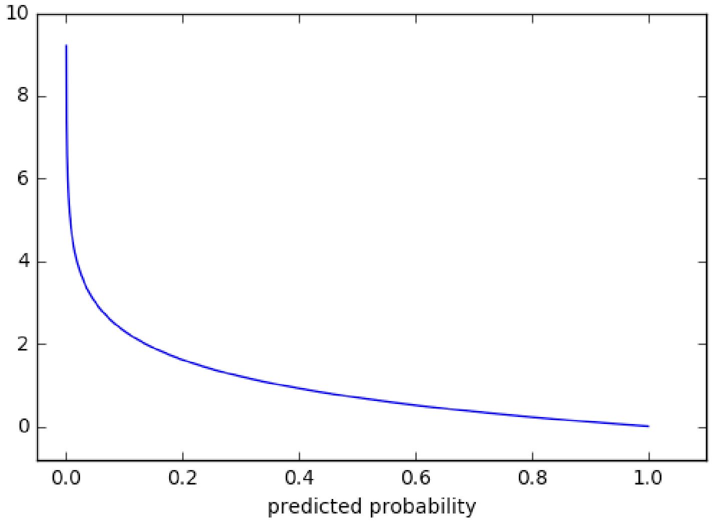

Figure 3.4: Cross-entropy error for the true label. If the predicted probability is, e.g., 0.2, the error is larger than for 'correct' probability values above 0.5.

The cross-entropy loss is one of the most widely used loss functions for classification in general.

Up until now, the discussion has been based on a binary classification problem. If the extension to more classes is made, some adjustments are in order. One possible strategy for multi-class classification is to adopt a one-vs-all strategy, where a binary problem is solved for every separate class vs. the rest of the classes, and afterwards for every sample the highest class probability score is used for the final prediction of its label. Another possibility is to solve binary classification problems between every combination of two classes, called the one-vs-one scheme. Here, a sample gets assigned to the class in which it has been classified the most for all separate binary classifications.

Logistic regression is a widely used linear classification technique, and by going through the steps of optimizing the decision boundary and using the logistic function to obtain a suitable loss

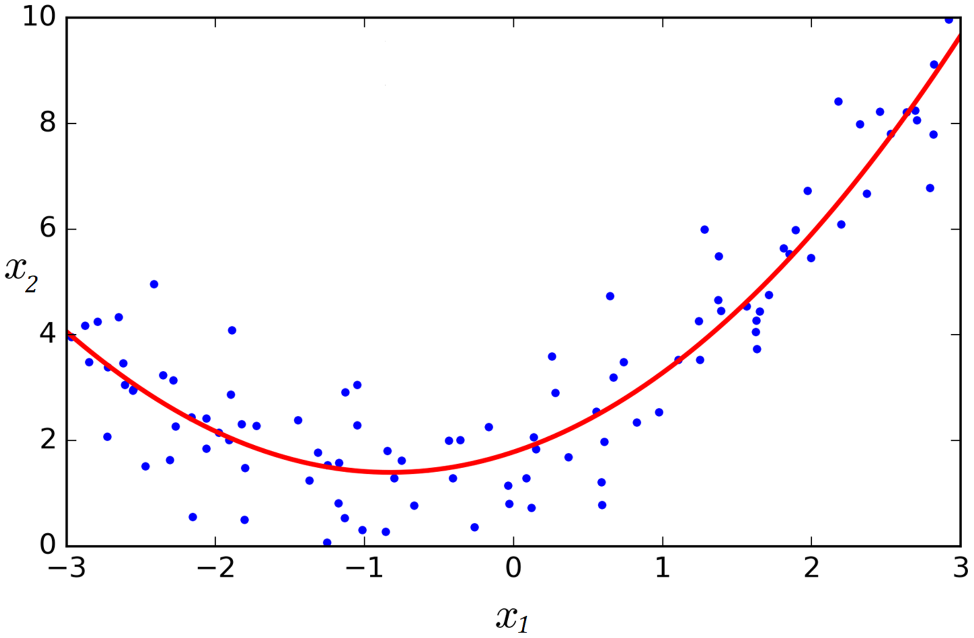

Figure 3.5: A regression line (red) is drawn, which should approximate the underlying function governing the data well.

function, many fundamental concepts in machine learning were touched upon. Besides logistic regression, many other classification algorithms exist. A notable example are support vector machines, which in a way can be viewed as an upgrade to the logistic regression model that maximizes the margin between the decision surface and the closest samples. Another important branch of algorithms are artificial neural networks, which are discussed in 3.2.2.

### 3.1.2 Regression

Another major branch in supervised learning is regression. In classification, it was all about predicting a class label from the feature values. In regression, it is about approximating the underlying function governing the relation between the features of the samples. An example is shown in Figure 3.5, where again there are two features, $x_1$ and $x_2$, but now the curve governing the relation between the features is to be predicted, shown as the red line. An approximation can be made with arbitrary complexity. A simple linear function

$$z(x_1, x_2|w_0, w_1, 1) = w_2x_2 + w_1x_1 + w_0 = 0 \implies x_2 = \frac{w_1}{w_2}x_1 + \frac{w_0}{w_2}, \tag{3.7}$$

is one of the possibilities, where the weights $\mathbf{w} = (w_0, w_1, w_2)$ need to be optimized[7] to find the best approximation possible with this complexity. The same supervised optimization approach is used to find these optimal weights as for the optimization of classification models. We are not restricted to linear approximations: any order of complexity is possible, as shown in Figure 3.6, but it is important to use a complexity that is neither too restrictive and therefore *underfits* the

[7] There is always one weight that can be eliminated. In equation (3.7), $w_2$ can be set to 1. The actual value does not matter, since it can be absorbed in the learning rate anyhow.

data (like both top examples in Figure 3.6), nor a complexity that is too sensitive to the specific training samples and *overfits* to the data (like the bottom right example).

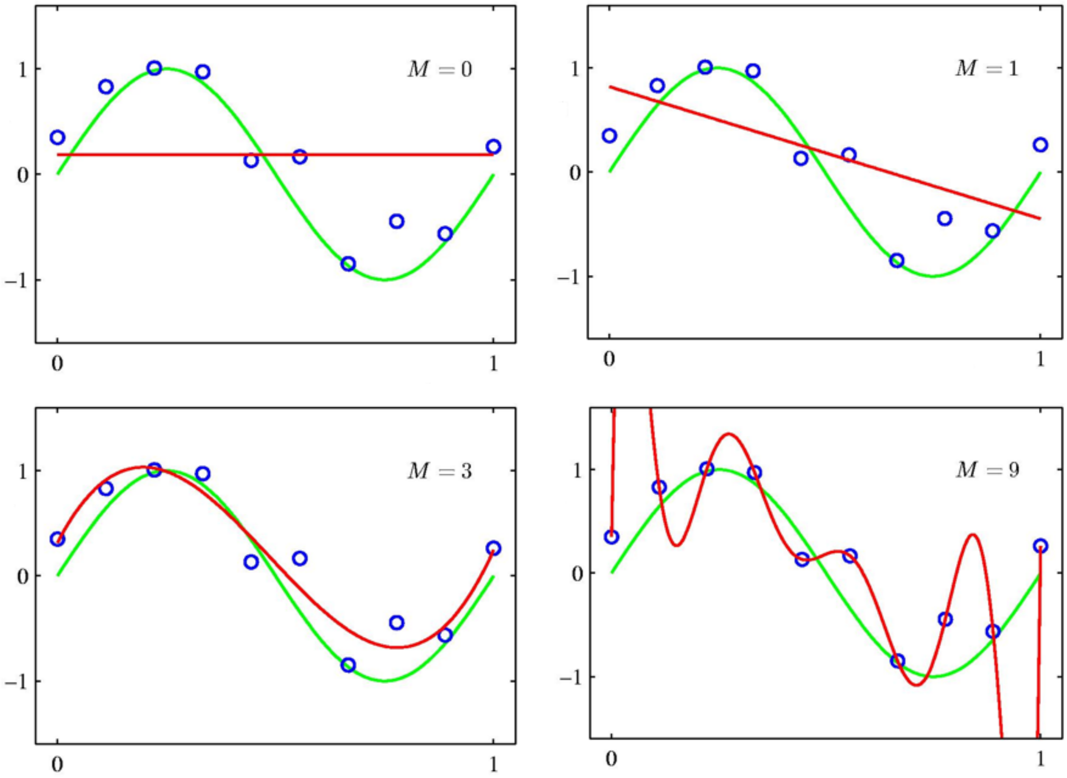

Figure 3.6: Different orders of complexity $M$ are used to fit a curve to the data. The green curve represents the true underlying function governing the data. The top examples show an underfit to the data, the bottom right example shows an overfit to the data.

Fitting a curve to data requires an optimization strategy. Again, a loss function has to be minimized in order to obtain an estimation of the true relation between the variables. Suppose we have a bunch of samples with features $x_1$ and $x_2$, and given the $x_1$ of a new, unseen test sample, we want to predict what its $x_2$ value will be. To do that, we can fit a curve $y = g(x_1|\mathbf{w}) = w_0 + w_1 x_1 + w'_1 x_1^2 + ...$ to the training samples by making sure that the sum of the squared distances between the estimated curve values, $y$, and the real values, $x_2$, is minimized. The total squared distance loss is given by summing over all samples:

$$L(r, y) = \sum_i (r_i - y_i)^2, \tag{3.8}$$

where in our example, $r = x_2$ is the actual value of the sample. This squared distance is defined in the direction of the to be predicted feature, and not orthogonal to the curve, as in equation (3.3). This is illustrated in Figure 3.7. The squared distance loss is now an appropriate loss function, since samples that have a great distance from the curve should be punished by the loss function. Finally, the weights of the estimate function are updated using gradient descent from equation (3.4).

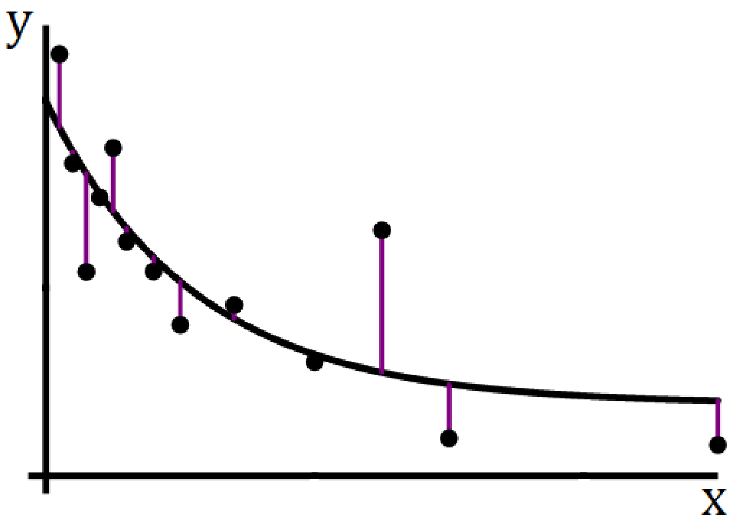

Figure 3.7: Curve fit by minimizing the squared distance loss function to estimate the function for predicting $y$, given $x$.

To obtain the right complexity for function estimates, several techniques for the *regularization* of the estimate exist. Regularization makes sure that an estimator does not overfit to the training samples by learning the training data by heart, with all its noisy irregularities that do not tell anything about the true underlying function governing all samples in real life. If not, the overly complex estimate function will not generalize well to unseen data. Regularization is a very significant part of building a good machine learning model, but the nuances of fine-tuning a model are out of scope for this introduction. The interested reader is guided to the literature on ridge regression and other regularization techniques, to explore the tricky concept of the *bias-variance trade-off*.

## 3.2 Unsupervised learning

Supervised learning algorithms have the luxury of working with labeled examples to learn a mapping function from inputs to outputs. In reality, there are often no labels available for many datasets. Still, information regarding the patterns in the data can be obtained with *unsupervised learning*. For example, the clustering of data into groups of similar samples can be done with techniques like the *k-means clustering* algorithm [16] or *Gaussian Mixture Models* [17]. Another possibility of unsupervised learning is the reduction of the dimensionality in the data to improve efficiency and effectiveness of learning algorithms, or to understand or visualize the data better. Unsupervised learning comprises a wide array of machine learning techniques, but this section will focus on two major algorithms: *principal component analysis* [18] and *auto-encoder neural networks* [19], since they form an important part of the anomaly detection models devised in chapter 4.

### 3.2.1 Principal component analysis

If input data is high-dimensional, this often brings along many challenges in machine learning. These challenges are collectively known as the 'curse of dimensionality'. They concern the difficult search for optimal solutions in very large feature spaces, or distorted distances between data samples due to bloated dimensionalities. The common theme of these problems is that when the dimensionality increases, the volume of the feature space increases so fast that the available data becomes sparse. To compensate, the amount of extra data needed to obtain statistical significant results grows exponentially. If structured regions in the data need to be discovered, the sparsity between samples obstructs an efficient organization. For these situations, models could benefit from dimensionality reduction techniques. These techniques should keep most of the predictive information in the data, while significantly reducing the amount of dimensions. There are many supervised and unsupervised dimensionality reduction techniques available. Here, one of the most popular dimensionality reduction algorithms will be discussed: principal component analysis, or PCA.

Principal component analysis does not necessarily reduce the dimensionality of the feature space. At its core, it is a technique that calculates new uncorrelated feature axes that are ranked according to the percentage of the variance they explain in the data. An example of the transformation from original feature vectors into uncorrelated ones is illustrated in Figure 3.8. It is only when the tail of the new feature axes – those that explain the least amount of variance – is discarded, that dimensionality reduction has been performed. The underlying assumption in using PCA is that the remaining *principal components* will contain most of the information that was contained in the data in the original feature space. This assumption is however not necessarily true; there could have been important predictive information in the tail that was eventually discarded. Nevertheless, principal component analysis is very often used for basic dimensionality reduction, since it easily provides good results.

To understand the calculation of the principal components, some linear algebra is required. The first step in finding the principal components is calculating the covariance matrix $V$ of the original features, based on the data available. Nowadays, many computational libraries exist for doing this. The covariance matrix expresses the covariance between two separate variables on the off-diagonal elements, and the variance per variable on the diagonal. Next, the eigenvalues and eigenvectors of the symmetric covariance matrix $V$ are calculated. Eigenvectors and eigenvalues are defined by the eigenvalue equation:

$$V \cdot \mathbf{v}_i = \lambda_i \ \mathbf{v}_i, \tag{3.9}$$

where the eigenvectors $\mathbf{v}_i$ have the property that they are only rescaled by the eigenvalues $\lambda_i$ when the covariance matrix $V$ is applied to them. The dimensionality of the covariance matrix is the amount of features squared, so there are as many eigenvectors as there are original features. Many computational libraries exist for solving eigenvalue equations. An interesting property of symmetric matrices is that the eigenvectors corresponding to different eigenvalues are orthogonal

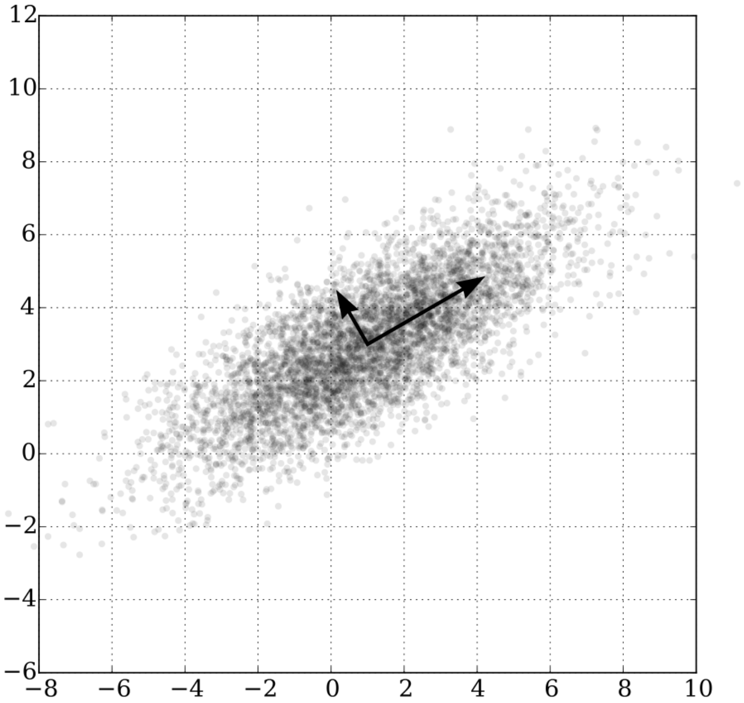

Figure 3.8: Two correlated feature axes are transformed to two new orthogonal feature axes, where the first one is in the direction with maximum variance, and the second is orthogonal to it. If the second dimension is now discarded, the dimensionality is reduced from 2 to 1, while keeping as much variance in the data as possible.

to each other (and hence linearly independent). These eigenvectors can be used as an orthogonal basis to span the same feature space as the original feature vectors. On top of that, the new orthogonal feature vectors have a covariance matrix with only eigenvalues on the diagonal of the matrix, so there is no covariance between any two feature vectors. Up until now, there has been no loss of information, only a change of basis. The eigenvalues on the diagonal still represent all the variance in the data. If we now want to reduce the dimensionality of the feature space, we can omit the features that explain the least amount of variance: the eigenvectors with the smallest eigenvalues/variance.

PCA is a powerful technique to construct new uncorrelated features and reduce dimensionality, while still keeping most of the variance in the data. It is again stressed that this is an unsupervised technique: if it is used to prepare data for a supervised task, careful consideration on how many dimensions to remove is advised. Also, for PCA to have the desired effect, an appropriate rescaling of the features is advised before PCA is applied, since large relative scale differences between features will distort the variances picked up by the PCA algorithm.

### 3.2.2 Auto-encoder neural networks

Auto-encoder neural networks are part of a prominent branch of machine learning algorithms, called *artificial neural networks*. These neural networks have a structure that is inspired by the way biological neural networks, like the human brain, process information. Artificial neural networks have received a lot of attention in machine learning research this past decade, thanks to the extraordinary results they achieve in fields like computer vision and natural language processing. Many more advanced artificial neural networks, e.g., convolutional and recurrent neural networks, are smart adaptations to the baseline neural network architecture of the *multilayer perceptron*, or MLP. The multilayer perceptron was historically used in the context of supervised learning, but by now several applications for unsupervised learning exist.

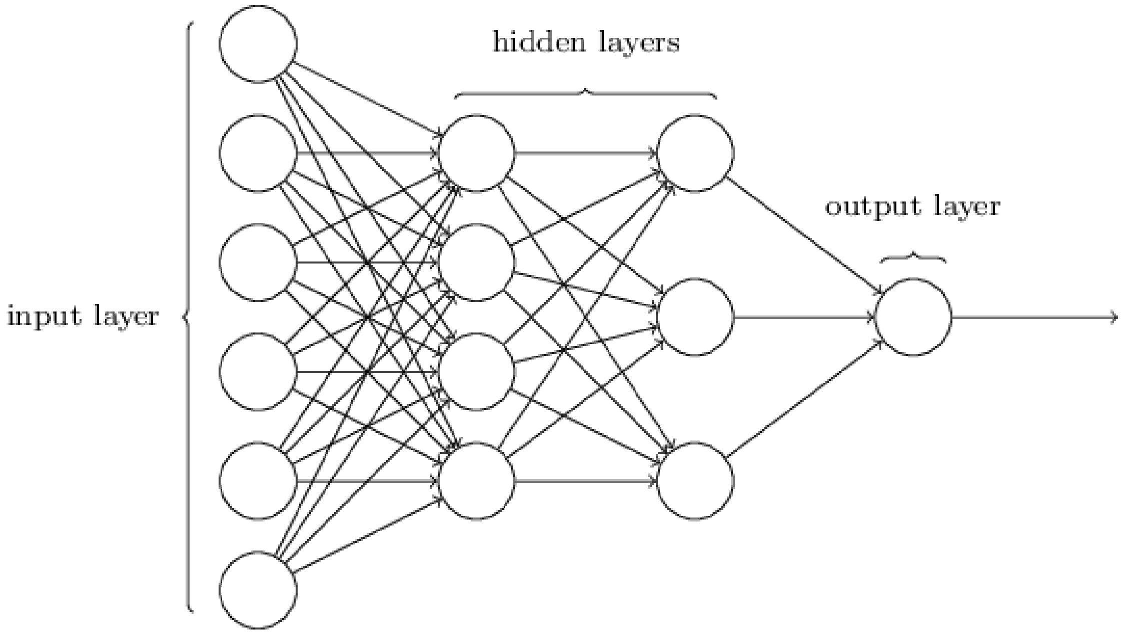

Figure 3.9: A classic multilayer perceptron architecture. The 6-dimensional input layer is followed by two hidden layers that eventually lead to the output layer.

An MLP consists of *neurons*, which are the building blocks of all neural networks. The neurons are ordered in layers: an input layer, one or more hidden layers, and an output layer. The neurons between layers are connected, as illustrated in Figure 3.9, and information from the input layer can propagate through the network until it reaches the output layer. A neuron in the first hidden layer receives a signal from every neuron in the input layer. These signals are weighted, and an activation function is put on top of the (weighted) sum of all signals to create a new signal that will be forwarded to every neuron in the following layer. This process continues for every neuron in every layer until the output layer is reached. This way, the fully connected network maps an input to an output. If we take a closer look at the processing that happens in one neuron, we find that it is very similar to the logistic regression algorithm that we are by now familiar with. The weighted sum $z$ over all the input signals $i$ can be written as:

$$z = \sum_i w_i x_i + w_0, \tag{3.10}$$

which corresponds to the familiar sum of the input features, where $w_0$ represents the intercept value for the neuron. Just like with logistic regression, the weights need to be optimized to obtain the best approximation to the function that we are trying to model, only now every neuron has its own set of weights, and neurons from subsequent layers are connected. The output signal of each neuron is acquired by applying an appropriate activation function on top of the weighted sum $z$, as shown in Figure 3.10. This activation function can be the familiar sigmoid function, or another activation function suited for the problem.

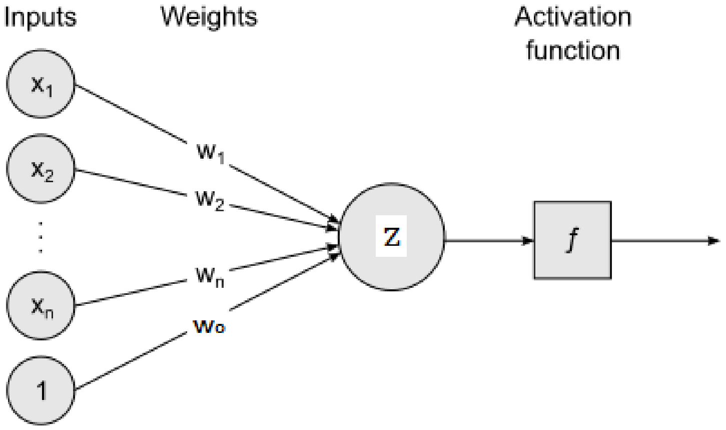

Figure 3.10: Information processing flow of a single neuron. The incoming signals $x_i$ from the previous layer are multiplied with the weights $w_i$ and then summed. An activation function $f$ is then applied to this weighted sum $z$, and the result is passed to all neurons in the subsequent layer as an input signal.

The optimization of the weights of all neurons happens by minimizing a loss function with gradient descent. This shows that the core framework of machine learning introduced in section 3.1 is universally applicable. We consider a classification problem, where the cross-entropy loss is often a good choice. For classification, the output layer of the MLP will consist of an amount of neurons equal to the amount of classes[8], and for every output neuron, the incoming signals from the last hidden layer will be weighted and summed and a specific output activation function will be applied. A commonly used one is the *softmax* activation function $s$:

$$s(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \tag{3.11}$$

where $z_i$ represents the weighted sum of the current output neuron, and the denominator sums over the $z_j$ of all output neurons, including $z_i$. The softmax activation function can be considered

[8] For binary classification though, one neuron suffices.

the multi-dimensional extension of the sigmoid function, and produces normalized probabilities for every output node. These probabilities represent the predictions of the neural network, and the class corresponding to the node with the highest probability represents the 'hard' prediction of the model. The soft probabilities are used to calculate the cross-entropy loss. To minimize the loss, a strategy called *backpropagation* is used. It is based on the familiar procedure of gradient descent, but now the weight derivative of the loss function of the output layer is dependent on the activations of the neurons from the previous layer, which are themselves dependent on the neurons from the layer before, and so forth. By using the chain rule for derivatives, the information gathered from the loss function can be backpropagated all the way to the input layer. As a result, the weights for all neurons can be fine-tuned, and the neural network has an effective optimization strategy to approximate the function governing the relation between the inputs and outputs.

One of the greatest strengths of neural networks is the ability to effortlessly capture non-linear behaviour in the function mapping the input to the output, thanks to the non-linear activations functions in each layer. Also, neural networks make feature engineering redundant, since the first layers of the neural network are automatically trained to create adequate representations of the input. In a way, artificial neural networks incrementally learn more complex features, and form a feature hierarchy. The drawback of many (deep) artificial neural network models is that they require a lot of data to train on to model the complex underlying behaviour, while the often huge parameter space of the networks is prone to overfitting.

Now that the inner workings of the multilayer perceptron are known, the auto-encoder variant can be considered. An auto-encoder neural network has the same amount of neurons in its input layer as in its output layer. After the input layer, the first half of the hidden layers contain gradually decreasing amounts of neurons, so that the input data is compressed into a smaller subspace. The second half of the hidden layers then have increasing numbers of nodes to try and reconstruct the data from the compressed subspace. The first half is called the *encoder*, the second is called the *decoder*. The goal of an auto-encoder neural network is to have an output layer that mimics the input as closely as possible. An example of an auto-encoder neural network architecture is shown in Figure 3.11.

Auto-encoders are a particular kind of unsupervised learning, since the input data itself is used to calculate the loss with the output values. They are sometimes called 'self-supervised' algorithms. An often used differentiable loss function is the *mean squared error* (MSE), given by:

$$L_{MSE}(x_i, \hat{x}_i) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2, \tag{3.12}$$

with $x_i$ the original feature values, $\hat{x}_i$ the reconstructed feature values, and $n$ the dimensionality of the input.

Auto-encoders have many interesting applications in machine learning. One possibility is to use them as a dimensionality reduction model. The bottle-neck layer of the auto-encoder has

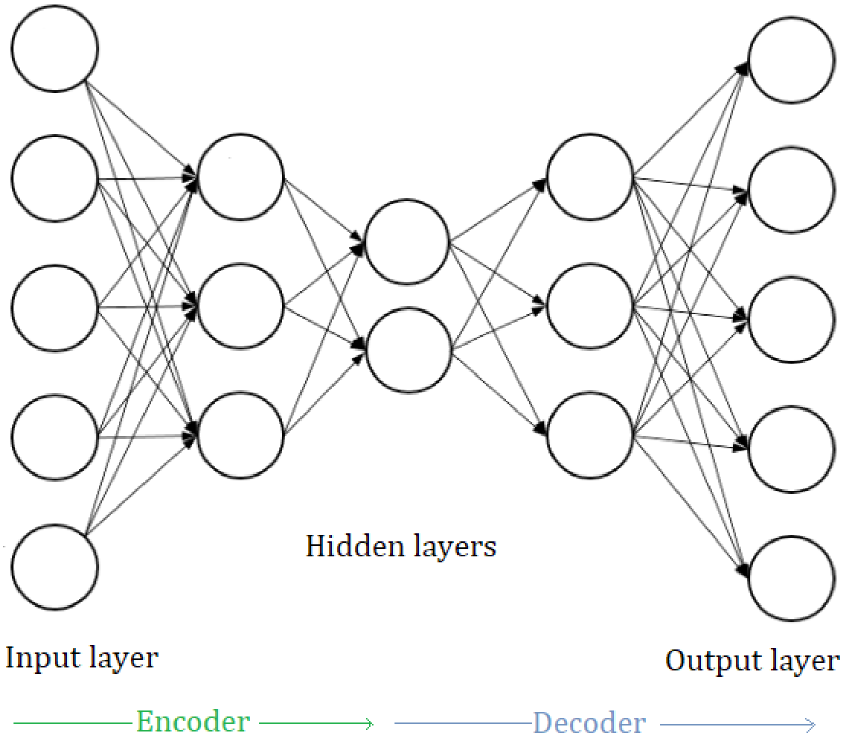

Figure 3.11: Example of an auto-encoder neural network architecture with three hidden layers. The input and output layers have the same amount of neurons.

to capture as much of the input information as possible, in order for the decoder to be able to correctly reconstruct the input. This makes the unactivated output values of the bottle-neck layer very suitable lower-dimensional features. The entire dataset can be used to train an auto-encoder, and afterwards every sample can be run through the encoder part only, so that suitable low-dimensional features emerge. This approach can be very effective, but the drawback is that a separate auto-encoder has to be trained for most datasets, and resulting encoded feature values will not make sense if the input to the network does not resemble the training data.

On the decoder side, auto-encoder networks can be used for *generating* variations on input data. These algorithms are called *variational auto-encoders*, and they can, e.g., generate variations of human faces, or have styles of music blend into each other [20]. Variational auto-encoders differ from regular auto-encoders in that their bottle-neck layer consists of two encoded vectors of equal size, instead of one. The first vector contains mean values $\mu$, and the second vector contains corresponding standard deviations $\sigma$. The reason for this, is that regular auto-encoders produce encoded representations which are discrete points in the reduced feature space. For a generative model, random variations on the encoded feature vector of an input sample are required. With only discrete points, this is not feasible. Therefore, distributions of similar groups in the encoded space are learned instead of discrete points, so that randomness can be introduced for each encoded representation of a sample, or outputs can even be completely randomly generated from the encoded distributions.

## 3.3 Predictive maintenance

Predictive maintenance is a collection of techniques designed to predict equipment failure, with the goal of acting on this information to prevent the failure from occurring. Monitoring a system to detect upcoming failures allows maintenance to be planned on designated moments before an unacceptable high failure chance is reached, while at the same time avoiding costs associated with performing maintenance too frequently out of caution. The need for predictive maintenance is increasing in a time where systems become increasingly complex and automated. A properly structured maintenance strategy can reduce costs, decrease maintenance man-hours, improve efficiency and increase the total production output overall. The results from predictive maintenance might also help to understand the operation of the equipment better.

### 3.3.1 Strategies

Traditionally, maintenance is performed over fixed time intervals, called 'preventive maintenance', or worse, by waiting until a failure occurs and then react to it, called 'reactive maintenance'. The reactive maintenance strategy has no costs up-front, but often leads to high costs down the road when equipment eventually fails at the most inconvenient times. Preventive maintenance has the advantage of planned maintenance at convenient times, and may avoid failures before they happen, but it also leads to an increase in maintenance costs as parts are replaced even when not required yet. Another risk with frequent preventive maintenance is that there is a higher risk for human errors when parts are replaced or inspected often. Installing a defective part or incorrectly reassembling the system will introduce a liability into the system, and a failure will probably occur before the next scheduled maintenance, resulting in the same inconveniences of the reactive approach. Predictive maintenance avoids the problems associated with both by warning ahead of time if the system approaches a failure, so that a convenient time for maintenance can be planned without causing major disruptions to operations. Of course, predictive maintenance is no crystal ball: predictions of future trends of the system's condition will always have a degree of uncertainty, and no system can be completely spared of unexpected events. Much depends also on the data available to create predictive algorithms; some data allow more tailored predictive maintenance approaches than others.

An evaluation of the system's condition can be done in an offline or online fashion. The online continuous monitoring is most often applied, and uses real-time sensor measurements like vibrations, temperatures, electrical signals etc. to assess the condition of the system. When the condition monitoring indicates a loss in performance or predicts a failure within a certain time span, operators are warned and maintenance can be scheduled, hereby reducing costs substantially and increasing the system's reliability.

When sensor measurements or other relevant data regarding the system are gathered, monitoring the health of the system requires appropriate analysis of the incoming data stream. The predictive algorithms used for this analysis will recognize patterns and generate insights that are then supplied to the operators in the form of alarms or other meta-information. These insights can be investigated to assess a corrective action if needed. Early predictive maintenance algorithms were mostly rule-based, with specific thresholds on sensor data obtained by careful analysis of experts of the system. A simple example: when the temperature of a system surpasses a limit, *together* with a high motor velocity value, an alarm is triggered. A high motor velocity or temperature on its own would not have triggered any alarms. These if-then statements are meticulously built around one specific system, and lack generalizing properties to other systems. Rule-based systems are labour intensive, and need to be implemented for each system by experts. In contrast, more general approaches can be found in machine learning. Multiple machine learning algorithms exist to address predictive maintenance challenges, and these algorithms are extendable to almost any device given that the proper data is available. Predictive maintenance machine learning algorithms search for a representation of the health status of the equipment based on the available data, and use this representation to predict valuable information like the estimated remaining lifetime of a system, or the probability of failure at that time. More on the prediction of the remaining lifetime is discussed in chapter 4, and [21, 22] provide more details on predictive maintenance in general.

### 3.3.2 Anomaly detection

Many of the machine learning models used for predictive maintenance are based on *anomaly detection* algorithms, including the models presented in this thesis. This section will introduce some of the basic concepts in anomaly detection. For more details, and an overview of anomaly detection applications, see [23].

Anomaly detection deals with the problem of finding instances in a dataset that do not represent normal behaviour. These instances are called anomalies or outliers. Anomaly detection is used in a variety of contexts, of which one is predictive maintenance. Detecting anomalies in the data can provide important information that triggers an action to respond to the anomalies. Some examples are anomalous MRI images indicating the presence of a malignant tumor, or anomalies in credit card transaction data signalling credit card or identity theft. Detecting anomalies in data has been researched in statistics as early as the 19th century, but over time, a wide variety of anomaly detection techniques have been developed. Many of these techniques are domain-specific, while others serve a more general purpose.

Figure 3.12 shows a simple example of anomalies in a two-dimensional dataset. There is a normal region, indicated by the green boundary, but data far away from this region is flagged as anomalous. Data is considered anomalous if it differs enough from normal data, but ‘enough’ is defined by the problem at hand. If new data is introduced with properties that were previously unobserved, they might at first be flagged as anomalous. However, if enough similar samples occur and a trend is observed, the data might be added to the pool of normal behaviour. This is for example done in anomaly detection with auto-regressive approaches like the ARIMA model [24].

A simple approach to anomaly detection would be to define regions representing normal behaviour, and indicate any data instance outside of this region as anomalous. This is not an

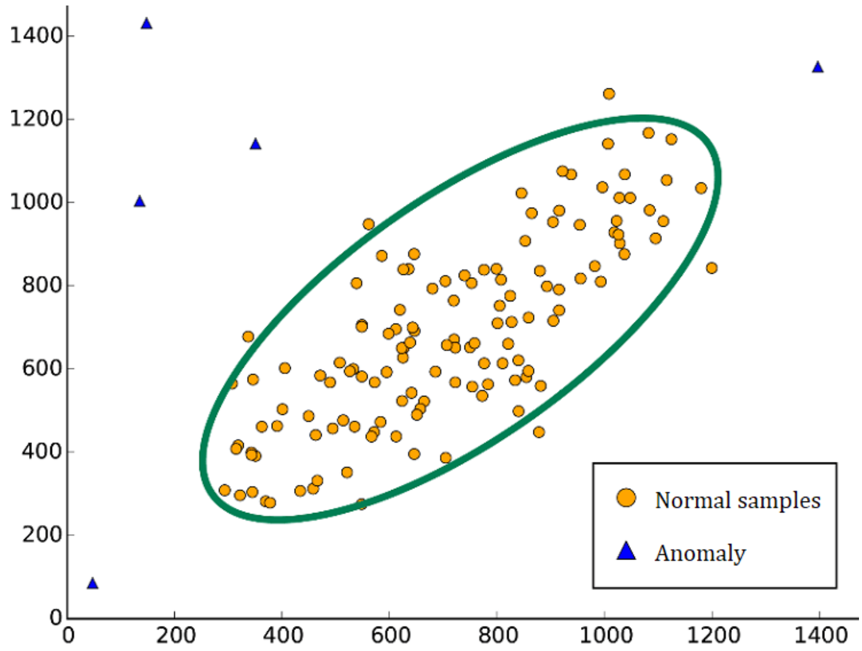

Figure 3.12: A simple example of anomalies in a two-dimensional data set.

easy task, however, and several challenges need to be solved before this is possible. Some of these challenges are:

- The boundary between normal and anomalous behaviour is often unclear. Samples close to the boundary are difficult to classify, and a simple, yet complete mathematical description of the normal behaviour region can be hard to obtain.
- In many cases, normal behaviour changes through time, and an update of the notion of 'normal' is required in order to maintain accurate anomaly detection.
- The definition of an anomaly depends on the application domain. Applying techniques developed in one domain to another might not provide good results.
- Often the data contains noise which might mistakenly look like anomalous behaviour. Noise removal might accidentally remove an informative outlier, which is not desired.

Anomaly detection is a complex problem to solve generally. Most anomaly detection techniques are built around solving one specific problem by leveraging the data properties and problem specifics. For example, some techniques are built specifically for dealing with time series data, where data samples have the *contextual attribute* of time and often time correlations between subsequent samples exist. Feature values belonging to one context might be considered anomalous, while in another context are considered completely normal. For example, if high temperatures are registered in winter they might be flagged as anomalous, while the same temperatures in summer might be considered normal behaviour. Another distinction between models might be that some models focus on flagging single samples as anomalous, while other models will only flag a *collective* of samples as outliers, with each sample on its own not being considered anomalous. An example of a collective anomaly is shown in Figure 3.13.

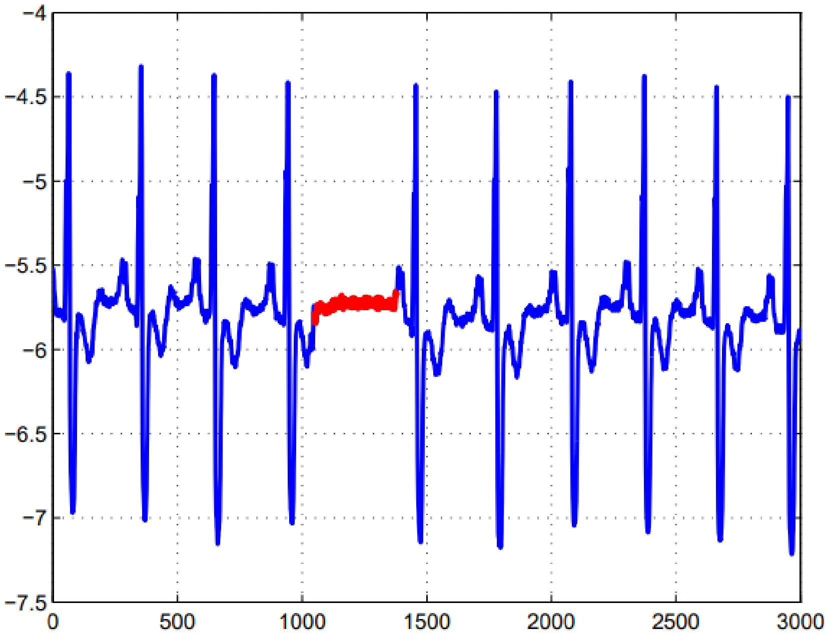

Figure 3.13: Collective anomaly corresponding to an Atrial Premature Contraction in a human electrocardiogram output. The red region denotes an anomaly because the same low value exists for an abnormally long time. The low value by itself is not considered an anomaly [23].

The machine learning distinction between supervised and unsupervised learning also exists for anomaly detection. Obtaining quality labeled data is often expensive and requires knowledge from human experts. Covering all types of anomalous behaviour during labeling is often not feasible, and it can occur that the anomalous behavior is dynamic in nature, e.g., new types of anomalies might arise for which there are no labeled training examples. Another difficult case is if anomalous instances correspond to catastrophic events and are therefore extremely rare or nonexistent. However, *if* one has the luxury of a fully labeled dataset, the complete arsenal of supervised learning techniques could in theory be applied to try and classify new samples correctly. One important problem with a predictive 'normal' vs. 'anomaly' approach, is that very often there is a severe class-imbalance between the two classes, which severely complicates a good optimization of the machine learning model. Dealing with class-imbalance is an important topic in machine learning, and more details are provided in [25].

The case of a training dataset where all anomalies are labeled is rare, and if it does occur, it resembles more a traditional classification problem than 'real' anomaly detection. Getting labels for normal behaviour only, however, is often within the possibilities of many datasets, since normal behaviour is generally more easily recognizable, more abundant, and requires less experience and knowledge of the specifics of the data than for labeling of the anomalous behaviour. It is therefore quite usual in anomaly detection to use a semi-supervised learning strategy, where normal behaviour is labeled, and anomalies are defined as anything that diverges from this pool of normal data. The techniques from semi-supervised anomaly detection are therefore more widely applicable than supervised techniques. It also addresses problems like dynamic behaviour of anomalies or extremely rare occurrences of anomalies, like for catastrophic events. The problems with class-imbalance are also conveniently avoided. A typical approach in semi-supervised anomaly detection is to build a model that recognizes normal behavior, and use the model to identify anomalies in the test data if a sample diverges from the learned normal behaviour.

Besides semi-supervised learning, dealing with completely unlabeled datasets is also very common in anomaly detection. When unlabeled data is available, the assumption is made that normal instances are (far) more frequent than anomalous instances. If this is not true, unsupervised anomaly detection techniques will suffer from high false alarm rates.

The output of an anomaly detection algorithm can be a continuous *anomaly score*, or a discrete label. Anomaly scores can be converted to discrete labels by setting up scoring intervals and creating a label for every interval. Scoring based anomaly detection techniques allow the analyst to use a domain specific threshold to select the most relevant anomalies. This is precisely what the proposed models from chapter 4 will do, based on a semi-supervised learning approach.

Some popular anomaly detection algorithms are:

- One-class support vector machines [26],
- Bayesian networks [27],
- Hidden Markov models [28],
- Cluster-based anomaly detection. [29]

All anomaly detection algorithms have their strengths and weaknesses with regards to the available data, so an algorithm has to be chosen with care.

# Chapter 4

# Anomaly detection for turbomolecular pump data

## 4.1 Turbomolecular pumps

The art of building and maintaining an operational fusion device involves making sure millions of separate parts, with just as many varying functions, form a coherent working entity. If one of the parts is broken or malfunctions, the reactor as a whole stops functioning. One example of an important chain in a tokamak fusion device is the vacuum pump system. For a good plasma operation, a high vacuum has to be obtained. This can be done by combining turbomolecular pumps and cryopumps to achieve very low pressures. The vacuum pump system is not only expected to give good operation in high magnetic field conditions, but also to produce the ultra-clean high vacuum necessary to *generate* the plasma. It has to remove all detrimental molecular constituents, and additionally make sure that the remainders of a plasma experiment are pumped away in a sufficiently short time. Another requirement for the pumping system, is that it must be able to withstand faultlessly the high working pressures and high gas throughput for long periods of operation. Turbomolecular pumps are suited as an important core part of the pumping system for such operational demands: the high compression ratio for gas particles ensures a clean vacuum by efficiently preventing any backstreaming of pollutants into the vacuum chamber. Still, operational failures happen, and good maintenance of the pumping system is required. In this chapter, a data-driven approach will be explored to support this maintenance and provide new information to operators about the condition of the turbomolecular pumps. The dataset used here is provided by the Culham Centre for Fusion Energy (CCFE) at Culham, near Oxford, where the JET tokamak is located.

### 4.1.1 Workings of a turbomolecular pump

In Figure 4.1, an example of a turbomolecular pump is illustrated. The main body consists of rotor blades and stator blades. The tilted rotor blades rotate at high frequencies to give kinetic energy to gas particles in the direction away from the vacuum chamber. The stator blades, which are tilted in the opposite direction of the rotor blades, help make sure that gas does not return to the vacuum vessel. Most turbomolecular pumps work in different stages, where each stage compresses the gas a bit more, until at the exhaust of the pump, the pressures are acceptable for the gas to be carried away by more conventional pumps. This is visible in Figure 4.1 as the angle of the blades shifts throughout the stages. Turbomolecular pumps operate at pressure ranges where the fluid approximation of a gas is often not applicable anymore, and it is more convenient to use the free molecular flow regime. The pumps do not necessarily ‘attract’ diffusing gas by creating an underpressure, but simply process the molecules that eventually hit the rotor blades.

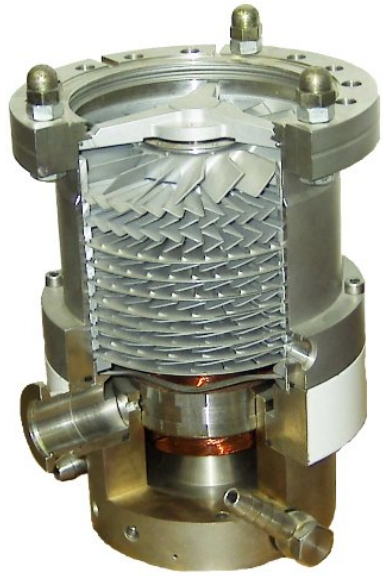

Figure 4.1: Interior view of a turbomolecular pump. Different rotor sizes can be seen.

Part of the performance of a turbomolecular pump is related to the frequency of the rotor. As the frequency increases, the blades give more kinetic energy to the gas molecules. To increase speed without causing deformation to the rotor blades, several stiff materials and blade designs are used. The standard turbomolecular pumps available on the market therefore have a metal rotor/stator assembly suitable for their intended use, with the rotor shaft supported by metal or ceramic ball bearings with an organic lubricant, or magnetic levitation bearings.

### 4.1.2 Adaptation to fusion conditions

When turbomolecular pumps are used in a fusion reactor, they are subject to some rather uncommon conditions. Magnetic fluxes leaking from field coils can cause heating of the rotor blades and reduce the rotational velocity of the pumps through Eddy currents. Magnetic fields can also disrupt magnetic levitation bearings. It is therefore desirable to place the turbomolecular vacuum pumps where they are least affected by the magnetic fluxes. The use of tritium in the fusion fuel imposes another problem: organic materials are contaminated by the radioactivity, and they cannot come into contact with the outside of the closed system. Also, high energy neutrons caused by deuterium-tritium reactions deposit energy in organic materials, causing

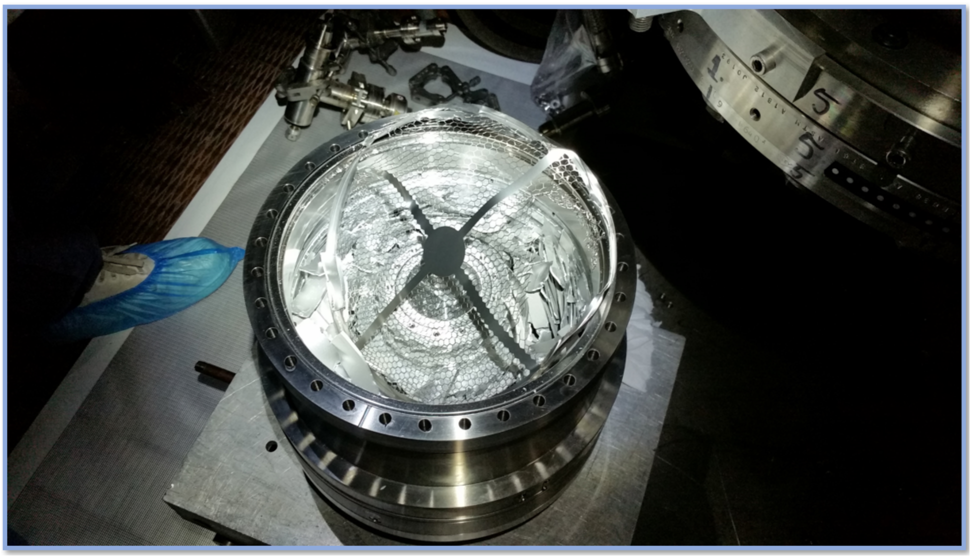

Figure 4.2: The first failure for the turbomolecular pumps: broken rotor blades. (UKAEA)

polymer chains to break and decrease performance of the pump. This accumulation of radioactivity and degradation of the pump performance is undesirable for maintenance of the pumps, and is hard to manage from a safety point of view. This implies the need for tritium compatible vacuum pumps in fusion research. To accommodate this need, changes to a commercially available turbomolecular pump can be made, so as to handle fusion conditions. The turbomolecular pumps studied in this case are such adapted vacuum pumps. The primary changes that were implemented are: replacing Viton O-rings with metal seals, changing wire insulation to silicon rubber, changing electrical lead-throughs to ceramic or glass, and removing any additional purge or vent ports. This adapted vacuum pump can then safely be used for fusion experiments with tritium gas.

### 4.1.3 Failures

There are nine turbomolecular pumps of the same model under study here. They have been adjusted for fusion conditions and have ceramic ball bearings (in contrast to magnetic levitation bearings). The period of operation of the vacuum pumps spanned about nine months, with sensor measurements of the pumps dating from December 2017 to August 2018. Without failures, only four pumps would have been required for the entire operational period, since there are only four positions around JET where the pumps are to be placed. Unfortunately, five critical failures occurred during the time the pumps were operational, and thus five replacements had to be purchased. Failures are manifested as the inability of the rotor to turn at the correct operational frequency: the rotational frequency of the blades rapidly declines to zero in a matter of seconds, mostly joined by excessive noise. In the case of the first failure, it was a rotor blade loss incident, illustrated in Figure 4.2, that caused the failure. The other four failures appear to be due to a failure of the upper and/or lower ceramic bearings, shown in Figure 4.3, inhibiting the rotor from turning. The root cause of each failure is currently unclear, but it is suspected that an

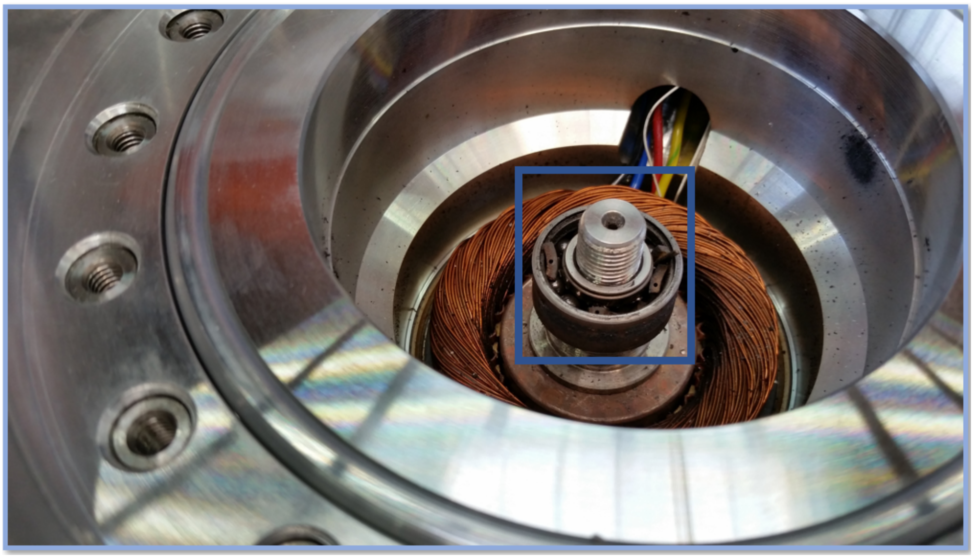

Figure 4.3: Example of a typical bearing failure for the turbomolecular pumps. The bearing is indicated with the blue rectangle. (UKAEA)

adaptation to fusion conditions might have introduced a weakness in the pump. These five costly and unexpected failures are investigated in the following sections, to see if machine learning can provide some new insights and, more importantly, if a general approach to preventing failures like these can be developed and used for future operations.

## 4.2 Data properties

### 4.2.1 Samples and feature description

Eleven sensors monitor each turbomolecular pump. The first six sensors are sampled every 30 seconds. These are the rotational frequency (Hz), the bearing temperature (°C), the body temperature (°C), the current (A), the power (W) and the voltage (V) of the pump, and temperatures are only measured in integers. The last five sensors are sampled every 5 seconds. These are two Penning gauge pressures (mbar) and three Pirani gauge pressures (mbar). There are four positions around the tokamak where the turbomolecular pumps are placed, called TT01, TT02, TT03 and TT04, each with their own sensors that provide readings to a controller unit. Data from all these sensors is available from December 2017 until August 2018, spanning about 6500 hours for every position. An example of a clean period of data for the first six sensors is provided in Figure 4.4 for the TT01 position. The fragment starts at March 1, 2018 and shows the next 100 days of sensor data. The pump is almost continually on, indicated by the 555 Hz rotational frequency, except for the last part; on June 2, 2018 (∼2246 h), a failure occurred in the installed turbomolecular pump, leaving it unable to operate any further. The failure can be recognized in Figure 4.4 as a sharp peak for the temperatures and smaller peaks for current and power. The frequency rapidly decreases, and so does the voltage.

The complete data from Figure 4.4 is an unusually good sequence of quality data and will be

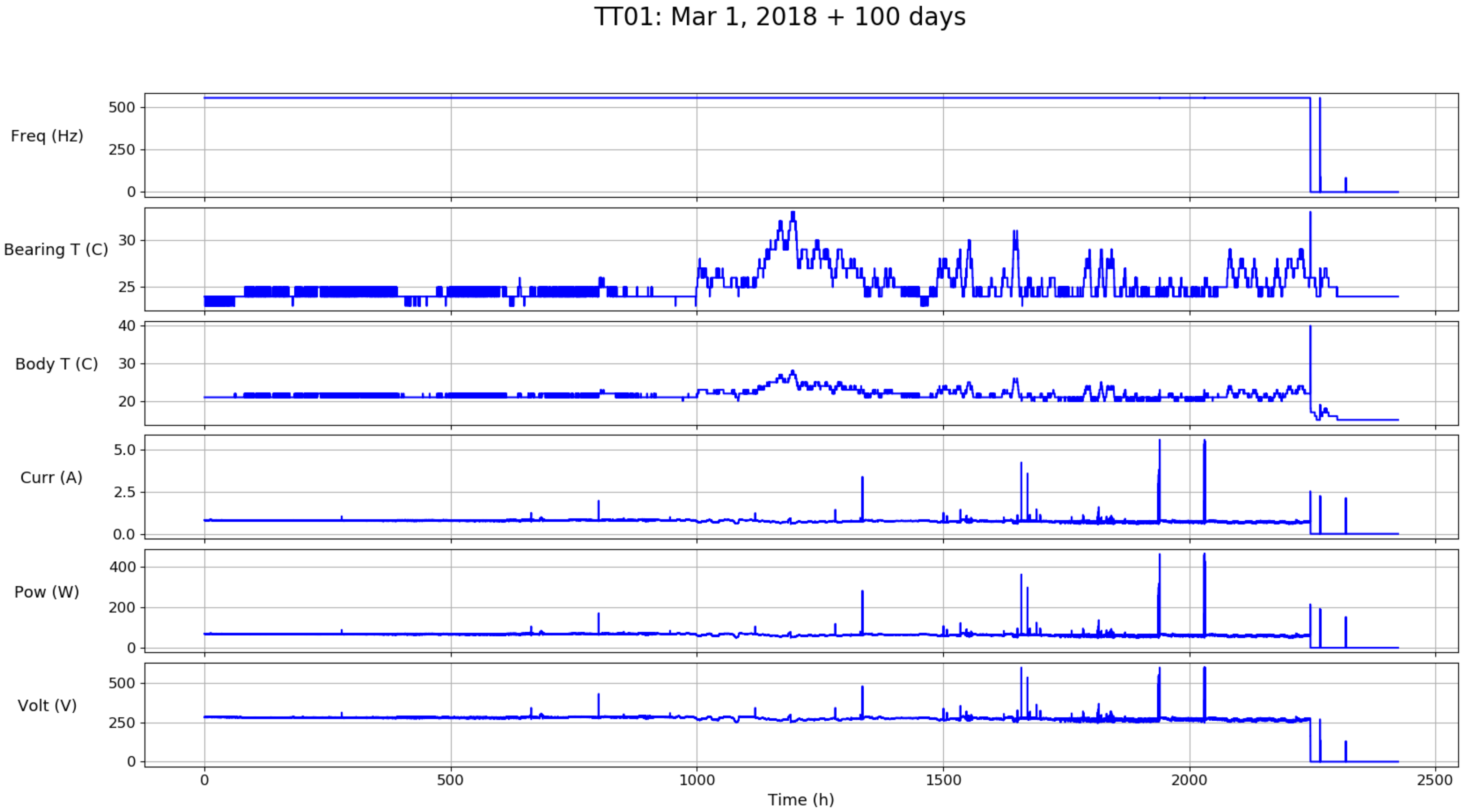

Figure 4.4: The first six sensor measurements (sampled every 30 s) for TT01 from March 1, 2018 and the following 100 days. The pump is mostly working, indicated by the 555 Hz rotational frequency, but a failure occurs around 2250 h, leaving the pump unable to operate any further.

used as the main sample during the exposition of the different machine learning approaches. It is not required to thoroughly study other data samples to follow the concepts in the remainder of this chapter, so for the sake of readability these are omitted here. The pressures are also omitted: they are not used in any of the models. The reason for this is to make the models easily extendable to new datasets without pressure readings and to avoid difficulties regarding electrical noise in the pressure data. A discussion on this decision is provided in the feature analysis section.

Useful data from other sequences will be presented throughout this chapter in accessible formats. Still, for the sake of completeness and to provide an overview of the full dataset to the interested reader, plots for every position and for every time period are presented in the appendix. A few main remarks to take away from the other sequences are:

- Extreme temperature spikes (that are not failures) sometimes occur. These are most likely due to a temperature read-back disconnection. If an open circuit occurs (e.g., a disconnected cable), then the output will look like an infinitely high temperature which the pump controller unit interprets as ~250 °C, its highest possible reading.
- Zero readings everywhere mean the controller has been powered off or disconnected.
- The Penning pressure gauge only works at pressures smaller than $10^{-3}$ mbar, so anything above this threshold can look like electronic noise. Above $10^{-3}$ mbar, one should use the Pirani pressure gauges for informative measurements.
- JET experiments – also called pulses – last about a minute. During and after this time, the pumps need to work harder to maintain vacuum vessel conditions. As a result, sudden peaks appear in the data, especially for the power/current and the pressures; higher pressures mean the pumps need to work harder and thus need more power. Some of the peaks in Figure 4.4 around 1700 h are examples of reactions of the pump to such pulses.

The data fragments from the first three remarks represent uninformative data, which are to be disregarded. These discontinuities impose some restrictions on the models we can use. The data from the pulses, however, represent a physical process and need to be handled accordingly in the analysis. We will come back to this in the next sections.

### 4.2.2 Inspection of the data for failures

The dates of the five failures are:

First failure: December 3, 2017 – Rotor blade loss incident at TT02
Second failure: January 22, 2018 – Bearing failure at TT01
Third failure: June 2, 2018 – Bearing failure at TT01
Fourth failure: August 2, 2018 – Bearing failure at TT01
Fifth failure: August 22, 2018 – Bearing failure at TT02

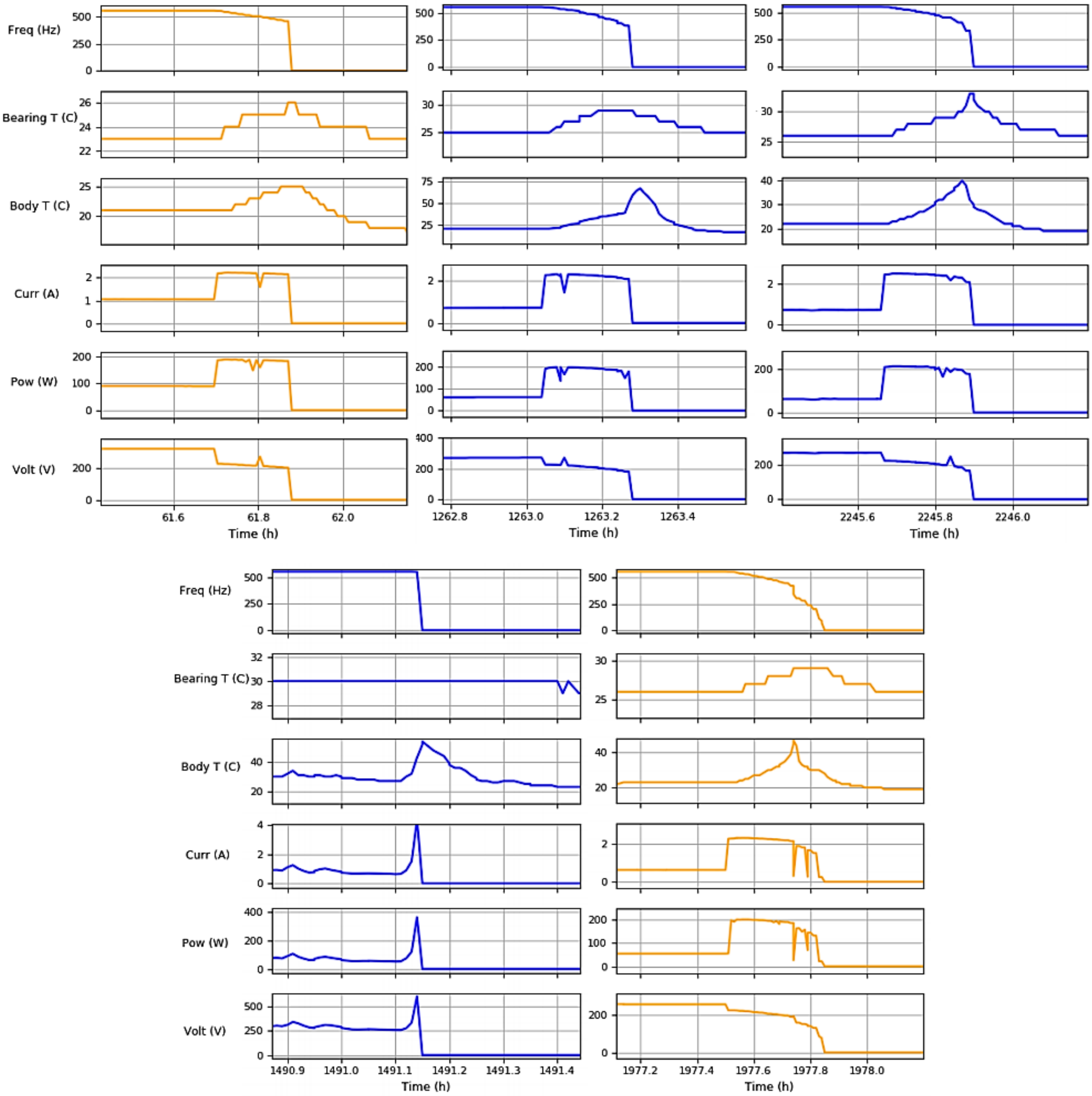

Figure 4.5: Close-up for all five failures, chronologically represented in reading order: December 3, 2017; January 22, 2018; June 2, 2018; August 2, 2018 and finally August 22, 2018. Blue lines represent position TT01, yellow lines TT02.

Let us take a closer look at how the failures unfold up close. In Figure 4.5, the failures are illustrated in chronological order. The second, third and fifth failures appear to be alike. Although the first failure is a rotor blade loss incident that happened almost immediately after starting the pump, the failure itself looks rather similar to these three failures. In fact, the data shows divergent behaviour for the fourth failure, compared to the other failures, which would indicate a different kind of malfunctioning based solely on visual analysis.

The three similar failures (2nd, 3rd and 5th failure) all have a spike in bearing temperature and body temperature. They have sharp increases for current and power, and a drop in voltage. An intuitive buildup to a failure can be seen in the data sequence leading up to the third failure in Figure 4.4. From 0 h to about 600 h, the data is quiescent (except for the single spike around 280 h). The period between 600 h and 1000 h could be called a transitional period. From 1000 h onward, increasingly variable behaviour is observed. The last two hundred hours before the failure show clearly visible fluctuations for all features, and before that, strong peaks in current, power and voltage are visible. A comparison between the quiescent period and the last variable period leading up to the failure is presented in Figure 4.6.

### 4.2.3 Feature analysis

Sensor readings are only informative when the pumps are on, otherwise zero readings or ambient temperatures are registered. Since the frequency is at a constant 555 Hz in Figure 4.4 when the pumps are on, this feature is not very informative. It is therefore left out in the feature analysis; we are more interested in the conditions leading up to the failure, rather than the aftermath. Rotational frequency might, however, be an informative feature for failure detection, since one of the main characteristics of a failure is the rapid decline of rotational frequency compared to a normal powering down of the pump, due to the inability of the rotor to turn correctly. Normally, while turning off the pump, the frequency should decrease gradually from 555 Hz to 100 Hz over 20-30 minutes. During most failures, it rapidly declines from 400 Hz to 0 Hz in a few seconds. The models for anomaly detection that will be presented in the next sections clearly recognize a failure anyhow, even without frequency as a feature. Another important remark is that, even though the pumps work at a constant 555 Hz, this can still mean the pumps are not working properly. In response to a degradation, the pumps could use more power to keep the rotors turning at the desired 555 Hz frequency, which can also bring about changes in the temperatures. A degradation pattern would then emerge in the non-frequency data.

The five remaining features (pressures not included) can be checked for correlations. A correlation matrix using the Pearson correlation coefficient is plotted in Figure 4.7. There seems to be a strong correlation between power and current, and both are also rather strongly correlated to the voltage. These correlations are visible on an intuitive level in the time series. The temperatures are correlated, and body temperature seems to correlate with current, power and voltage too.

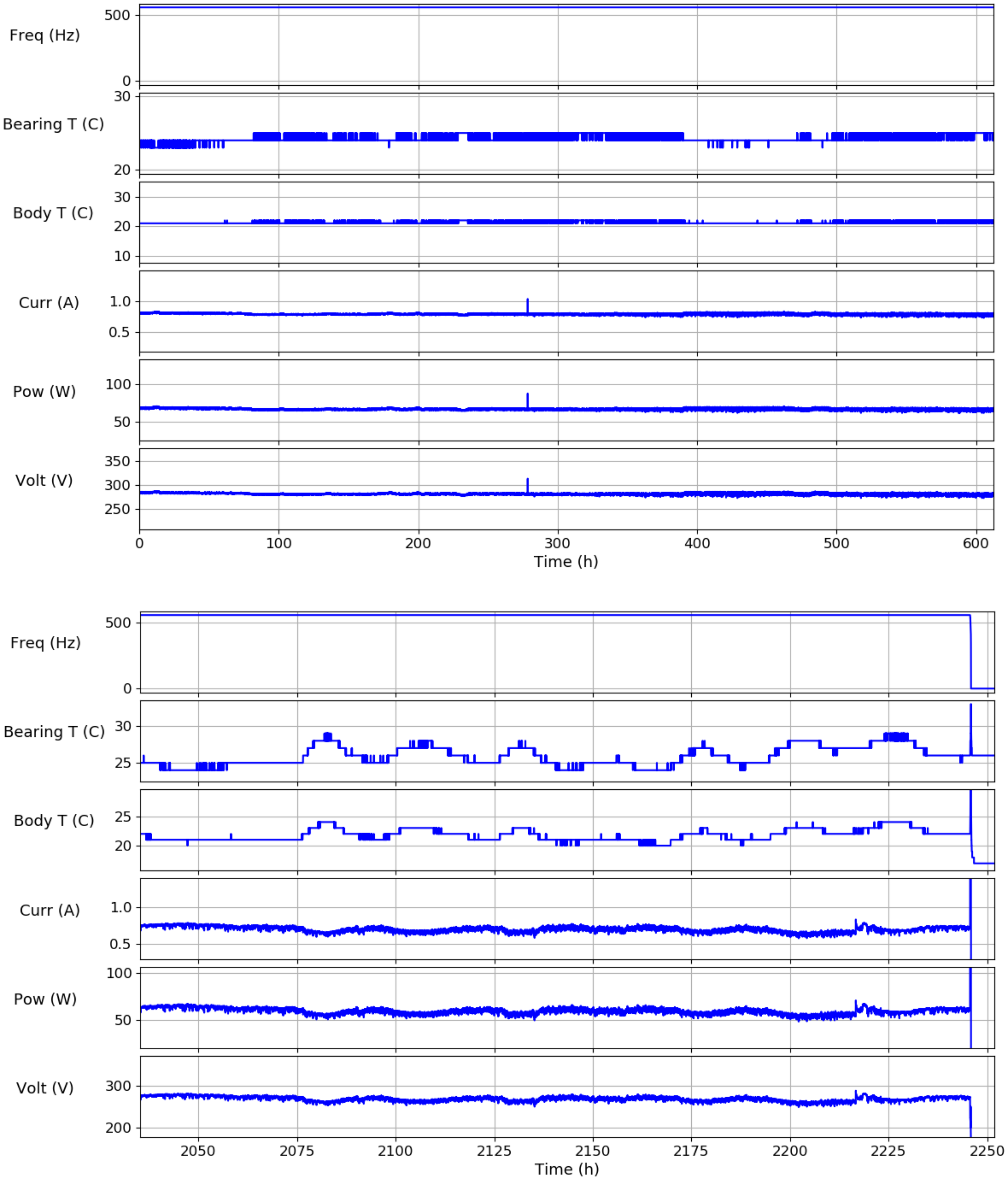

Figure 4.6: Top (0 h to 600 h): healthy behaviour of a pump after a fresh installment. Bottom (2050 h to failure at 2246 h): all features show non-subtle fluctuations and heavy noise.

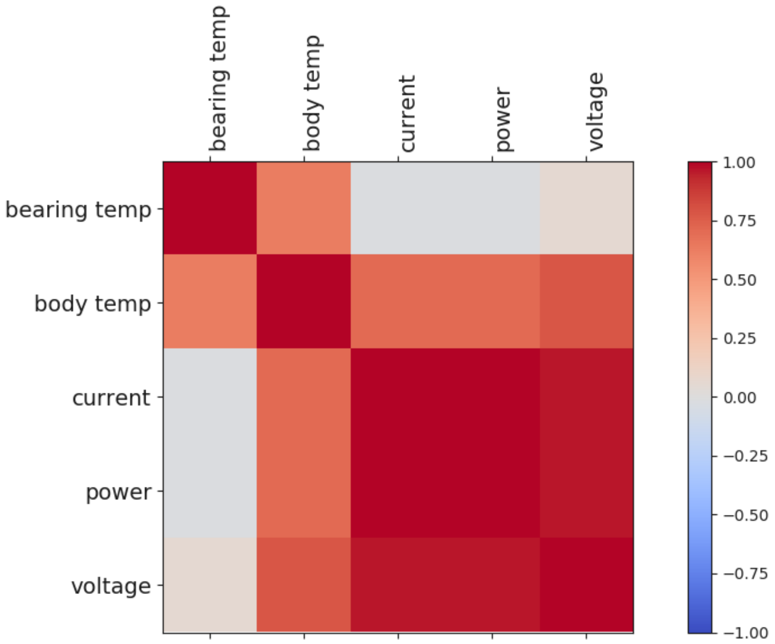

Figure 4.7: Correlation matrix for the sensor measurements.

## 4.3 Development of anomaly detection models

This section will describe the development of the models that were used for anomaly detection in the turbomolecular pump dataset. The choice of a model is dependent on three main factors: the problem definition, the quantity and quality of the data, and a trade-off between complexity and time/hardware constraints.

### 4.3.1 Data quality

Building a model to assess failures requires enough data to capture the degree of information that is needed to provide a sufficiently accurate answer to our problem. Besides the measured data (e.g., the sensor data for the pumps), information regarding the environment of the machine, mechanical properties and the way the machine is used are also valuable. Expert knowledge on physical processes that influence the workings of the machine can greatly weigh on certain model building decisions and help fine-tune a model where the data is too uninformative to use an automated approach. It is important to make note of the following:

- What does the failure process look like? Is it a slow degradation process or an acute failure? In our case, the failure seems rather acute, but prior to the failure, divergent behaviour can be seen.

- Are there different kinds of failures? Which ones are important to us? Most failures seem to be the same, with small variations. This would mean the model can reasonably be targeted to one kind of failure, or one can opt for a model that finds deviations from normal behaviour in general.

- How often and how accurate do the measurements need to be performed? Right now, the measurement sample rate is once every 30 seconds. Since we are interested in information on the scale of hours or days, this sampling rate should suffice. The accuracy of the temperature measurements, however, is very crude: only integers are provided. This is inconvenient and might disturb some subtle patterns in the data.

For degradation processes, often data has to be available for a longer period of time to capture the subtleties of the degradation. Besides the length of the data, the quality and quantity of the measurements are also of great importance. In an ideal scenario, data scientists work together with domain experts and machine operators to prepare a plan for data collection with the appropriate sensors that will serve as input to the best possible model for the problem at hand. Unfortunately, in real life, the data has often already been captured, and a model has to be built with what is at hand to address the specific problem. This does, however, provide an opportunity to use the experience from the limited dataset to plan data collection for the same – or new – problems that are foreseen for the future.

Depending on the quality and quantity of the available data, restrictions are imposed on what models can be built and with what accuracy.

### 4.3.2 The Holy Grail: estimating Remaining Useful Lifetime

One of the best possible scenarios for a predictive maintenance scheme would be to have an accurate estimate at every time step of how long it will take before the machine fails and cannot be used in operations anymore. This estimate is known as the *Remaining Useful Lifetime*, or RUL. The Remaining Useful Lifetime would provide very insightful information of when to schedule maintenance for the system. Unfortunately, our turbomolecular pump dataset does not provide the means necessary to build a good RUL estimator. The reason for this is that many degradation life times leading up to a failure are required to build a reliable RUL algorithm. When a statistically relevant amount of degradation life times are available, a notion of the remaining lifetime of the system can be obtained.

A practical use case of this technique is to predict the remaining useful lifetime for rechargeable batteries. If the amount of charging cycles a battery undergoes until it fails is recorded for thousands of batteries, one could calculate the probability of a new battery surviving or failing during its next energy depletion cycle. The prediction could be based on the prior probabilities of failure from the other batteries at that cycle, as illustrated in Figure 4.8. This is a very simple but effective estimation of remaining life time, where time is expressed as number of charging cycles. Once enough failures are recorded, a curve like the one in Figure 4.8 can be drawn. Operators can choose to replace batteries after a fixed amount of charging cycles when failure is becoming too likely based on a probability threshold. Having this kind of information can enable industry to implement preventive maintenance after an amount of cycles where the costs of replacing working batteries are less than the previous occurrences of battery failure during operations. Failures still happen, but much less frequently, and the large costs associated with frequent failures are avoided.

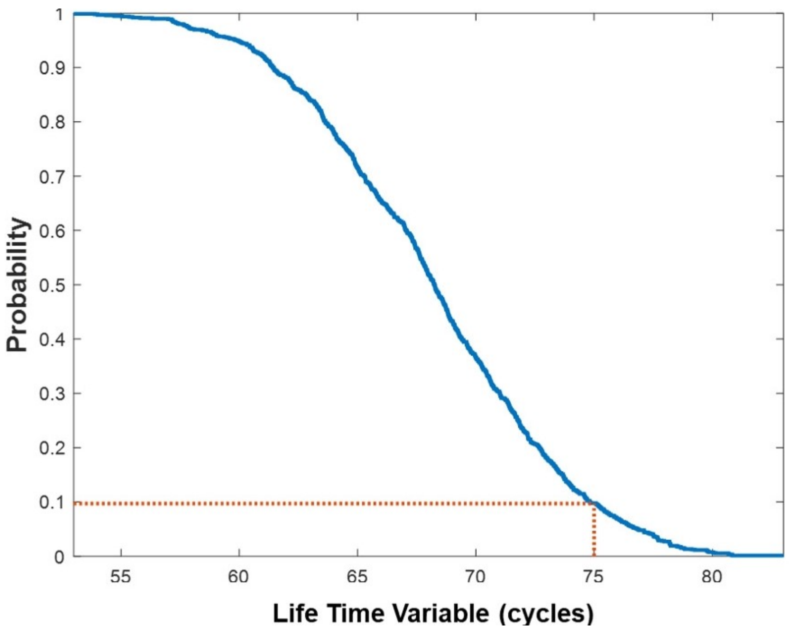

Figure 4.8: Probability of survival per charging cycle, based on the prior amount of batteries that survived that cycle. At the 75th cycle, the probability of the battery's survival is only 0.1 [30].

Another common case for RUL estimation is to use incoming sensor data from machine operations and build a *condition monitoring system*. In order to determine if a system is healthy or not, an appropriate *condition indicator* can be evaluated for the system, which can be as simple as a carefully chosen weighted sum of the input sensor values. Another more sophisticated example of a condition indicator is a reconstruction error of incoming sensor data. If an algorithm, trained to faultlessly compress and reconstruct healthy sensor data, signals an increase in reconstruction errors, the incoming sensor data probably do not represent a healthy condition of the system anymore[1]. If there is historical sensor data available from many similar systems, the condition indicator curves until failure can be stored for each of them. The initial curve from a new system can then be compared to the other curves, and the one that resembles the new curve the most can be taken for the prediction of the failure. This is illustrated in Figure 4.9. Another strategy is to fit a regression curve to the initial condition indicator values of an operational system, and extrapolate it to see when it crosses a threshold indicating unsafe operations. In the example of Figure 4.9, the threshold could be placed at 0.6, before any prior failure occurred. This threshold can be used as an indicator that maintenance is required. The estimated time until maintenance can then be obtained via the extrapolation.

Since for the turbomolecular pumps we are dealing with, at best, a few degradation cycles, our problem is not suited for RUL prediction. We can however, build a model using the next best thing: anomaly detection, which monitors the life time of every pump separately. Just like the previous case, a condition indicator based on the sensor data will be used to signal how healthy the system is. Although the obtained condition indicator curve cannot be linked

[1] Or the sensors are degrading themselves, which is a use case on its own.

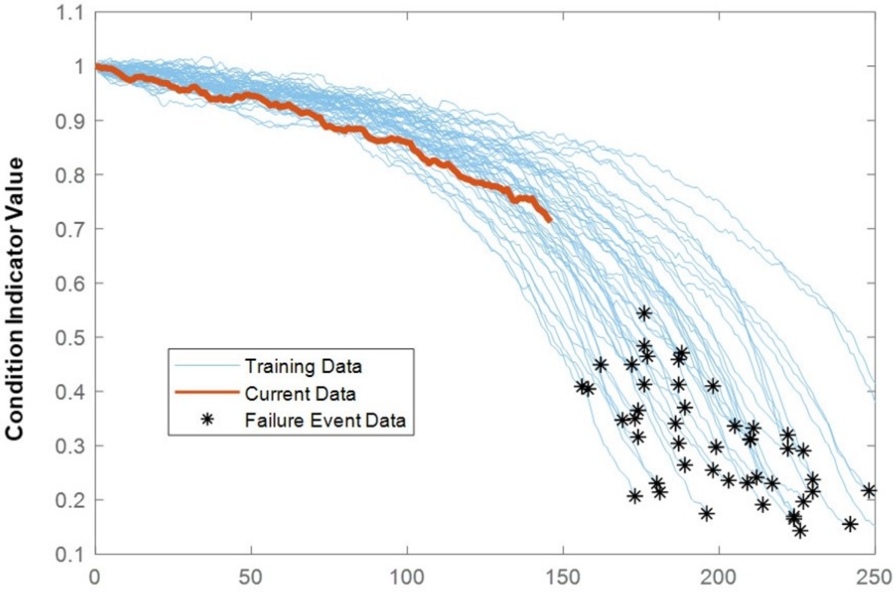

Figure 4.9: Condition indicator curves for several historical systems over time. The curve from an operational system can be compared to find an estimate of the RUL [30].

to a prediction of the RUL anymore, it still provides useful information to operators. To aid in the distinction between indications of healthy and unhealthy behaviour, a threshold can be applied. The threshold value does not contain any information on how much remaining lifetime the pump still has, but it does indicate a regime of anomalous behaviour or a 'danger zone' of operational conditions. This threshold, which separates the two regimes, is not categorized as a traditional classification method as it does not use labels for every time step during the pump's lifetime. Instead, we can designate a period, mostly at the beginning of the pump's lifetime, to be used as healthy data, and compare the data from every subsequent time to this healthy data by means of an appropriate condition indicator. This is a typical semi-supervised strategy used for anomaly detection, where anomalies are not put into designed classes, but are simply defined as being different from healthy data. In practice it is often possible to hand-pick a period of healthy data after the installation of a system, but not to differentiate the many complex regions that might or might not contain anomalies later on.

### 4.3.3 Failed approaches

Before describing the proposed models, we will briefly discuss some approaches that were considered, but eventually dismissed. One such approach was to build a recurrent neural network with memory, particularly a long short-term memory (LSTM) neural network [31], for time series forecasting. Essentially, the model is used to predict the next sensor measurement(s) based on the sensor data up to that point. The network is set up for the prediction of healthy behaviour, so if predictions of subsequent data samples significantly diverge from the actual measured sensor values, an anomaly is flagged. Divergent behaviour is defined by a setting a practical threshold. Unfortunately, our dataset contains too little and too erratic data to be used for training such complex models.

A classification approach was originally considered using labeled data split into two categories: ‘normal’ and ‘failure’. The main problem with training a classifier to make a distinction between these categories is that there is usually much less failure data than there is normal data, known as class-imbalance. Also, the classification of a failure during or after it has happened is not very useful for predictive maintenance. One could solve this by introducing gradations in the data, and labeling data based on categories between the limits of ‘healthy’ and ‘failure’, so that action could be taken if, e.g., many ‘near-failure’ classification results occur. The problem with this approach is that it was not feasible to accurately classify every sample in the turbomolecular pump time series without extensive knowledge of the pump and its conditions.

Another classification approach that was implemented but deemed not suitable, was to use a balanced dataset of healthy behaviour and near-failure behaviour, and train a classifier on separating both categories. If a soft classification approach is taken, with a probability as output instead of a discrete label, the increase in near-failure probabilities from 0 to 1 should in theory be observed during the pump’s lifetime. Although a shift in probabilities was indeed observed, using a simple logistic regression classifier, the results were subpar to the final models developed in this chapter, and they are therefore omitted.

A different strategy was based on the *matrix profile* technique [32], a recent time series motif discovery tool with a lot of potential. One of the great advantages of the matrix profile is that it is very general, and once it has been calculated for a time series it can be paired with many applications. The matrix profile values indicate for every time window in the time series how similar (or dissimilar) it is to the rest of the time window. The highest peaks in the matrix profile are called *discords*, and they indicate the time windows that are most uncommon. Discords can then be treated like anomalies, and it is assumed that a time period with high matrix profile values signals an upcoming failure. The matrix profile for the turbomolecular pump dataset was calculated, but deemed not informative due to it being too erratic (e.g., discords were not clear or intuitive). According to De Paepe et al. [33], the reason for the odd matrix profile is the flatness of the time series presented here. This flatness causes small perturbations around the flat equilibrium to be greatly enlarged in the matrix profile calculation, and thus the results of the matrix profile appear distorted. Extensions to counteract this unwanted phenomenon are proposed in the paper, but have not yet been implemented for the pumps dataset. More details about the very promising matrix profile technique are found on the site of the University of California [34].

### 4.3.4 Model 1: PCA and Mahalanobis distance

The first model is a combination of principal component analysis (PCA) [18] and multivariate Gaussian modeling of the resulting features, with the Mahalanobis distance to the center of the distribution as a measure for the anomaly score. It is a combination of several anomaly related techniques discussed in the anomaly detection survey of Chandola *et al.* [23]. From this paper, the idea originated to combine robust spectral anomaly detection with the Mahalanobis distance[2].

Principal component analysis is one of the most known dimensionality reduction techniques. It performs a linear mapping of the data into a lower-dimensional space, while maximizing the amount of variance that is kept. More on principal component analysis, including a visual representation, is given in the introductory chapter on machine learning. PCA actually converts possibly correlated features into a set of maximally linearly uncorrelated features, called the principal components, with the *same* dimensionality. To reduce the dimensionality of the data, a part of the tail of the principal components is dismissed, and only the components that capture the greatest amount of variance in the data remain. One has to keep in mind though that principal component analysis is an unsupervised technique, and that the components that capture the greatest amount of variance do not necessarily represent the most informative features for the task at hand. Nevertheless, PCA is a widely used dimensionality reduction technique, and will serve as the first step in this model. There are other linear and non-linear dimensionality reduction techniques available, and the interested reader is invited to explore the 'manifold learning' documentation from the popular scikit-learn machine learning library [35].

The dimensionality of each input sample is five, since only the bearing temperature, body temperature, current, power and voltage are kept as features. A practical reason for not using the pressures was already briefly discussed: measurements of the pressure gauges often contain electrical noise when the readings surpass a certain threshold. When one pressure gauge reading gets too large, the algorithm needs to switch to another gauge for reliable information. This is a challenge to implement in an algorithm, because it is often hard to separate noise from useful data and to have an algorithm learn the transition. If the model works without pressure measurements, and if it is kept as general as possible, this would be a good indication that the model is extendable to other pumps at JET and even other devices.

This model takes the features for each new time step as input, reduces their dimensionality with PCA, and further gives as output an anomaly score based on the Mahalanobis distance of the lower dimensional features to the center of the multivariate Gaussian distribution. If the score surpasses a threshold, it is registered as anomalous. The dimensionality reduction and the multivariate Gaussian modeling are based solely on healthy data. It is then assumed that unseen healthy data instances will be positioned in high probability regions of the distribution, while anomalies will occur in the (very) low probability regions, or are simply modeled by a different distribution. As mentioned earlier, a collection of healthy data has to be hand-picked to fit the Gaussian distribution on, so this is a semi-supervised technique. This selection of healthy data has to be reconsidered for every pump by an operator in the field. A general guideline, however, is to take data starting a little after the initial start-up of a new pump (to make sure that possible start-up fluctuations are not taken into account) and end the healthy data segment early enough, at a time where the operator believes the data is still healthy. This

[2] Chapter 9 from Chandola *et al.* shows the equivalence of summing the (eigenvalue weighted) squared projections of a sample on the principal components, to the Mahalanobis distance from the multivariate Grubb's test. This is a real mouthful, but basically comes down to the fact that squared Mahalanobis distances from randomly sampled data of a multivariate Gaussian follow a $\chi^2$ distribution.

is to minimize the chance that anomalous behaviour is brought into the healthy distribution, as this might distort the predictions of the model and make it less robust. The healthy data can in some cases be as little as a few percent of all the data from start to failure, especially for datasets where it is known that the degradation is already significant close to the moment of start-up, in contrast to systems where degradation mainly occurs during later stages in the system's life. It is of course important to have a statistically relevant amount of samples in the healthy data, and in some cases there is a trade-off between the purity of the healthy dataset and the amount of samples it contains.

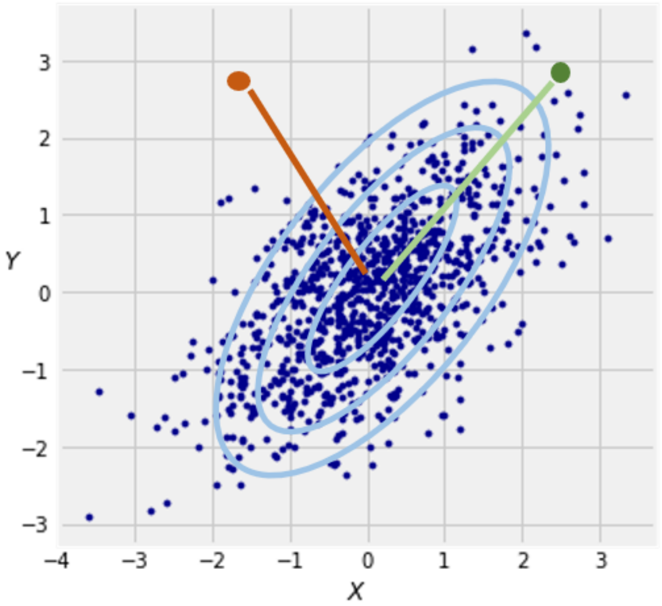

Figure 4.10: Intuitive visualization of the Mahalanobis distance for a two-dimensional Gaussian distribution. Both the orange and green sample have a similar Euclidean distance to the center of the distribution, but the orange sample has a greater Mahalanobis distance; the probability that it belongs to the multivariate Gaussian is smaller.

The main assumption that the data is structured according to a multivariate normal distribution is often a good approximation, even when the data is not strictly Gaussian. There are certain ways to check if this approximation is justified, like a visual inspection of the reduced data or checking if the squared Mahalanobis distance from the center follows a $\chi^2$ distribution. The multivariate Gaussian distribution for a collection of data is obtained by calculating the mean and covariance matrix of the samples. In theory, if large amounts of data are available, they should form ellipsoid shaped probability shells for a perfect normal distribution. Since the distribution is modeled on the healthy data, new data samples that are also healthy will have a high probability of being in proximity to the center of the multivariate distribution. Here, the distance measure used for this 'proximity' is the Mahalanobis distance. It is expressed as the Euclidean distance to the center of the distribution, divided by the width in the sample's direction of the characteristic ellipsoid that is spanned by the covariance matrix of the distribution. This characteristic ellipsoid can be seen as a multidimensional extension of the standard deviation. The mathematical definition for the Mahalanobis distance is

$$M_d(\mathbf{r}) = \sqrt{(\mathbf{r} - \boldsymbol{\mu})^T \ \Sigma^{-1} \ (\mathbf{r} - \boldsymbol{\mu})}, \tag{4.1}$$

with $\boldsymbol{\mu}$ the mean and $\Sigma$ the covariance matrix of the distribution. An intuitive visual example is given in Figure 4.10, where both the green and orange samples have a similar Euclidean distance to the center of the normal distribution, but the orange sample has a greater Mahalanobis distance: it is further from the center in terms of the spread of the ellipsoid in that direction. The Mahalanobis distance is a quantitative expression of the notion that the probability of the orange sample belonging to the multivariate Gaussian is smaller than the probability of the green sample.

Given the estimated multivariate Gaussian distribution for the healthy data, one can assess if new test samples are considered healthy by calculating their Mahalanobis distance and compare it with a specified threshold. The threshold can be defined with statistical methods, or fine-tuned empirically, e.g. with knowledge of domain experts. The squared Mahalanobis distances from randomly drawn samples belonging to a multivariate Gaussian follow a $\chi^2$ distribution, so a statistical approach to obtain a threshold might be to calculate the value for which the cumulative $\chi^2$ distribution amounts to 0.975 (which means a randomly drawn sample from a multivariate Gaussian has only a 2.5% chance of having a larger squared distance than this calculated value, see also Figure 4.11) and then take the square root of this value to get the Mahalanobis distance threshold. This is an approach frequently used in statistics. However, whether to take 0.975, or another value as a hard cut-off, is up to the expert, so in the end the decision again relies on domain knowledge, albeit perhaps more guided and explainable by the use of statistical confidence intervals.

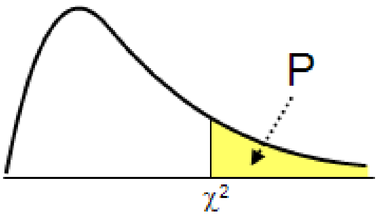

Figure 4.11: A $\chi^2$ distribution, where the percentage of the cumulative tail $P$ has to be specified in order to obtain a value for the Mahalanobis distance threshold.

We have built in the idea of only modeling the distribution on selected healthy data from the start. This is actually a simple form of *robust outlier detection*, as described in the paper by Rousseeuw *et al.* [36]. It contrasts with traditional outlier (anomaly) detection, where a distribution is modeled on the *complete* dataset, and the samples with the highest anomaly score are flagged. Including outliers in the distribution is dangerous, because it makes the anomaly detection sensitive to *masking*, which means the algorithm will not recognize some new outliers. This is not as obvious as it might seem at first, and the interested reader is referred to a more subtle explanation in the aforementioned paper by Rousseeuw *et al.* Besides avoiding the outlier masking problem, using a small healthy dataset for the distribution has a fundamental reason when used in predictive maintenance: we are not only looking for rare outliers to *one* distribution, but we are also trying to pick up a *shift* in the distribution from healthy to unhealthy. An illustrated example of this for the pump dataset will be given in the results.

### 4.3.5 Model 2: Auto-encoder neural networks

The second model is based on auto-encoder neural networks [19]. An auto-encoder neural network has the same amount of nodes in its input layer and output layer. In between the input and output layers are the so-called hidden layers. The first half of the hidden layers contain a gradually decreasing amount of nodes, so that the input data is compressed into a smaller subspace. The second half of the hidden layers then have increasing numbers of nodes, to reconstruct the data from the subspace to its original space. The auto-encoder neural network architecture has several possible applications, and more information on auto-encoders, and neural networks in general, is provided in the introductory chapter to machine learning.

We are interested in building an auto-encoder network that can reconstruct healthy data as precisely as possible. If new incoming test samples are compressed and reconstructed and deemed similar enough to the original, they are considered healthy. To get a measure of the similarity to the original input sample, the mean absolute error (MAE) between the input and output vectors is calculated. If this reconstruction error surpasses a certain threshold, it is flagged as an anomaly. The mean absolute error between an $n$-dimensional input vector $\mathbf{x}$ and its reconstructed output vector $\mathbf{x'}$ is given by:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x'_i - x_i|. \tag{4.2}$$

The anomaly flagging routine follows the same principles as for the statistical model with the Mahalanobis distance. The auto-encoder neural network is trained on healthy data until the reconstruction error is minimized. For new incoming test samples, the reconstruction error will be small if the sample is healthy, and large if it deviates from normal behaviour. A threshold to separate the two regimes is implemented. If the pumps are degrading or operate in dangerous conditions, anomalous behaviour should be flagged by the algorithm.

The threshold can be estimated with a heuristic method. Like the Mahalanobis distance, the reconstruction error can be considered a random variable, but now with an unknown underlying distribution. In the first approach, the assumption that the squared Mahalanobis distance follows a $\chi^2$ distribution was inherent to the model from the beginning. An appropriate statistical confidence interval could then be chosen by the operator and a threshold was obtained. Still, the boundary of the confidence interval posed a degree of freedom that replaced the freedom of choice of the threshold value. It is the same for the reconstruction error of the auto-encoder: an approximate distribution of the mean absolute error can be obtained (this is done in the results section with a simple error histogram), but will still only give an indication of a minimum value for the reconstruction error threshold. In reality, the threshold is ideally optimized with a data-driven estimate derived from many degradation cycles (which are not available for the pumps), or with fine-tuning based on experience.

One of the main strengths of neural networks is the ability to inherently model non-linear relations between the input variables. The network does not assume anything about the underlying distribution, but simply models an approximation of the real underlying function that describes the relation between input and output nodes. This is a great improvement compared to the linear mapping[3] and assumed normal distribution from the previous approach. At the smallest inner hidden layer, the auto-encoder network has effectively reduced the dimensionality of the input space. It has learned a (non-linear) relation that captures as much of the original information as possible. The following reconstruction part is then responsible for learning a second function that is tailored to the compressed information from the inner hidden layer. The network is most powerful when it can compress and decompress one kind of data. It can then really fine-tune its weights to make a correct abstraction of that data. The more underlying types a dataset consists of, the more complex the auto-encoder's architecture has to be to model all relationships. Avoiding over- or underfitting on the data is a constant challenge in machine learning, and this is no different here.

Our contribution is inspired by the 2016 paper from Kuzin and Borovicka [37] on early failure detection for predictive maintenance of sensor parts. The paper deals with several failure detection methods, from which the model based on auto-encoder neural networks was deemed appropriate for the turbomolecular pump dataset at hand. The general proposed outline of the model was implemented for the pump dataset and used as a baseline model from which modifications were made.

The algorithm from Kuzin and Borovicka uses raw sensor values from a sliding window as input to the neural network. If there are $n_s$ sensors and the sample window size is $n_w$, then the amount of input nodes is $n_s \times n_w$. With the aim of ingesting more temporal information, the model is extended by introducing the first and second time derivatives of the samples as new possible features. Each derivative can be used as the sole input, or they can be combined with the raw input values. If the raw values and both derivatives are used, the input size becomes $3 \times n_s \times n_w$.

It is important to stress that here, in contrast to the single-sample approach from the Mahalanobis model, a sliding time window is used. A time window is a simple way of taking time correlations into account[4]. Each sample now has the $n_w - 1$ previous samples attached to it, so that the context of a sample can help determine whether it is considered healthy or anomalous. Another additional feature that was implemented in the model is the possibility of varying the *hop size* $h$ of the time window. The hop size indicates how many samples are skipped between two time windows. This is another parameter to make the model as general as possible and to

[3] Although non-linear feature mappings were also possible for our first approach, here the neural network does all of that automatically without the need for extensive feature engineering.

[4] Advanced models, like long short-term memory neural networks, or attention-based models, are especially proficient at taking time correlations between samples into account; even between very distant samples. The simple sliding time window used here is restricted to shorter time correlations.

keep the possibility open for future applications. Window size and hop size are shown visually in Figure 4.12.

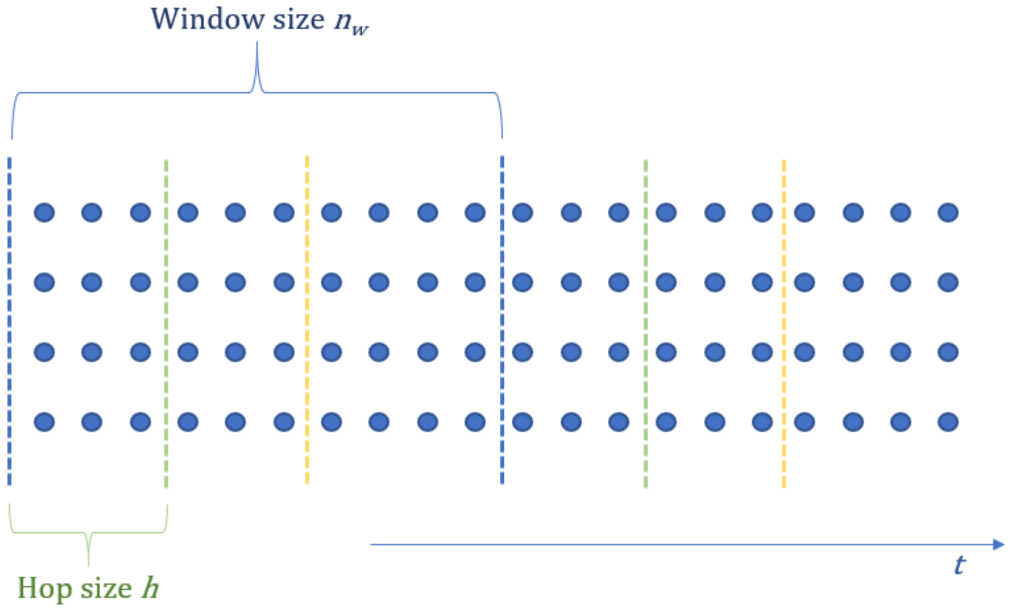

Figure 4.12: Illustration of the window size and hop size. Each dot represents a sensor measurement. The blue window is used as the first input sample to the algorithm, the green as the second, and the yellow as the third, and so on. For this figure, the window size is 10, the hop size 3 and the amount of sensors is 4.

## 4.4 Results

This section presents the results obtained with the models described in the previous section. Some thought is given to the model parameters, but an extended discussion of the model choices is kept for the discussion part in section 4.5. Results are shown for the third failure (data from March 1, 2018 until June 2, 2018, cf. Figure 4.4), since it is the most informative one. Results for another failure are presented in the appendix.

### 4.4.1 PCA and the Mahalanobis distance

The 5-dimensional feature input is reduced to two dimensions by means of principal component analysis. This is done because the correlation matrix in Figure 4.7 indicates that two engineered features could capture most of the information. Two dimensional data also provides a way to visualize the data. In Figure 4.13, this is done for healthy data samples, data samples that are already considered anomalous, and some samples right before the failure. A shift in the distribution can be seen from healthy to anomalous. Some data samples seem to form lines in the plot; these are caused by the discrete integer values from the temperatures. Before applying PCA, the original features were rescaled to values between 0 and 1.

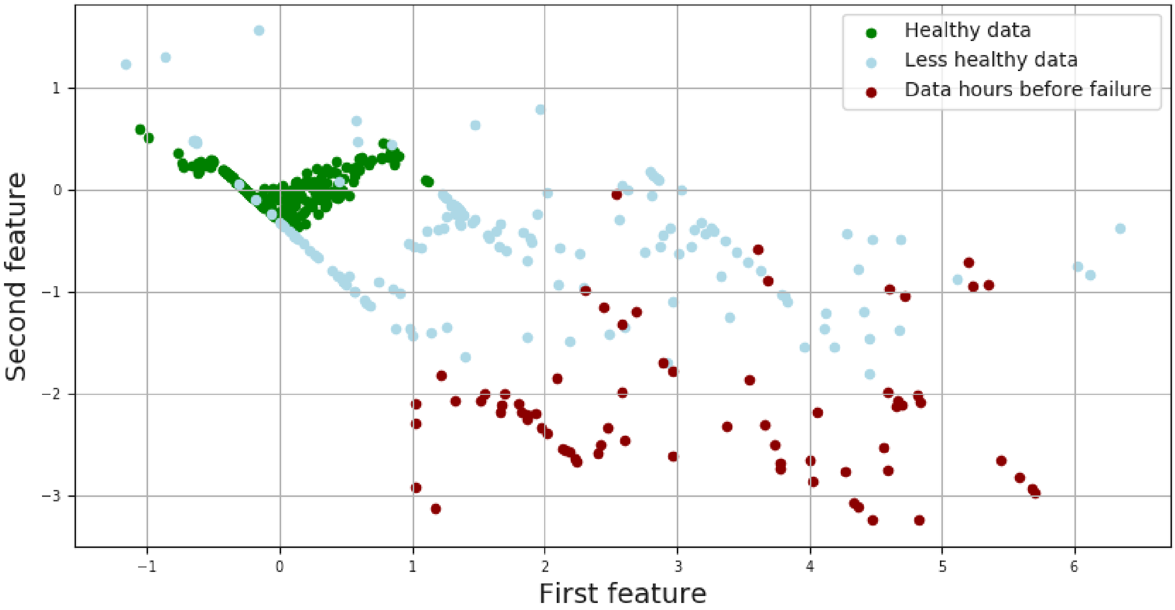

Figure 4.13: PCA reduced feature values for healthy data, less healthy data and data right before a failure.

As healthy data, the first 500 hours of the TT01 dataset from March 1, 2018 + 100 days (Figure 4.4) are used. This range is confirmed as healthy data, and represents the pump in operation not long after its installation. The data is downsampled to focus on the general trend, and the PCA transformation is learned on this healthy data. All subsequent samples are subjected to the same learned transformation.

Next, the mean and (inverse) covariance matrix of the multivariate normal distribution are fitted to the healthy data. For every sample up until the failure, the Mahalanobis distance to this distribution is calculated. A distribution for the Mahalanobis distance and for the squared Mahalanobis distance for the healthy data is shown in Figure 4.14. Based on the tail of the Mahalanobis distance for healthy data, a threshold of about 5 is taken. Next, all samples are checked against this threshold. The final anomaly flagging result is shown in Figure 4.15.

### 4.4.2 Auto-encoder neural network

While for the previous model only a variation in parameter values was possible for the dimensionality reduction and the threshold, for the auto-encoder neural networks more degrees of freedom are present. The amount of input nodes equals the input size of a sample, which depends on the window size, the amount of features and if time derivatives are taken into account. The neural network basic architecture follows the general structure suggested by Kuzin and Borovicka. It contains five layers: one input layer, a first hidden layer with 75% of the input nodes, an inner hidden layer with 50% of the input nodes, a reconstruction hidden layer with 75% of the input nodes, and a final layer with the same amount of nodes as the input layer. Model hyperparameters besides these were not specified by Kuzin and Borovicka.

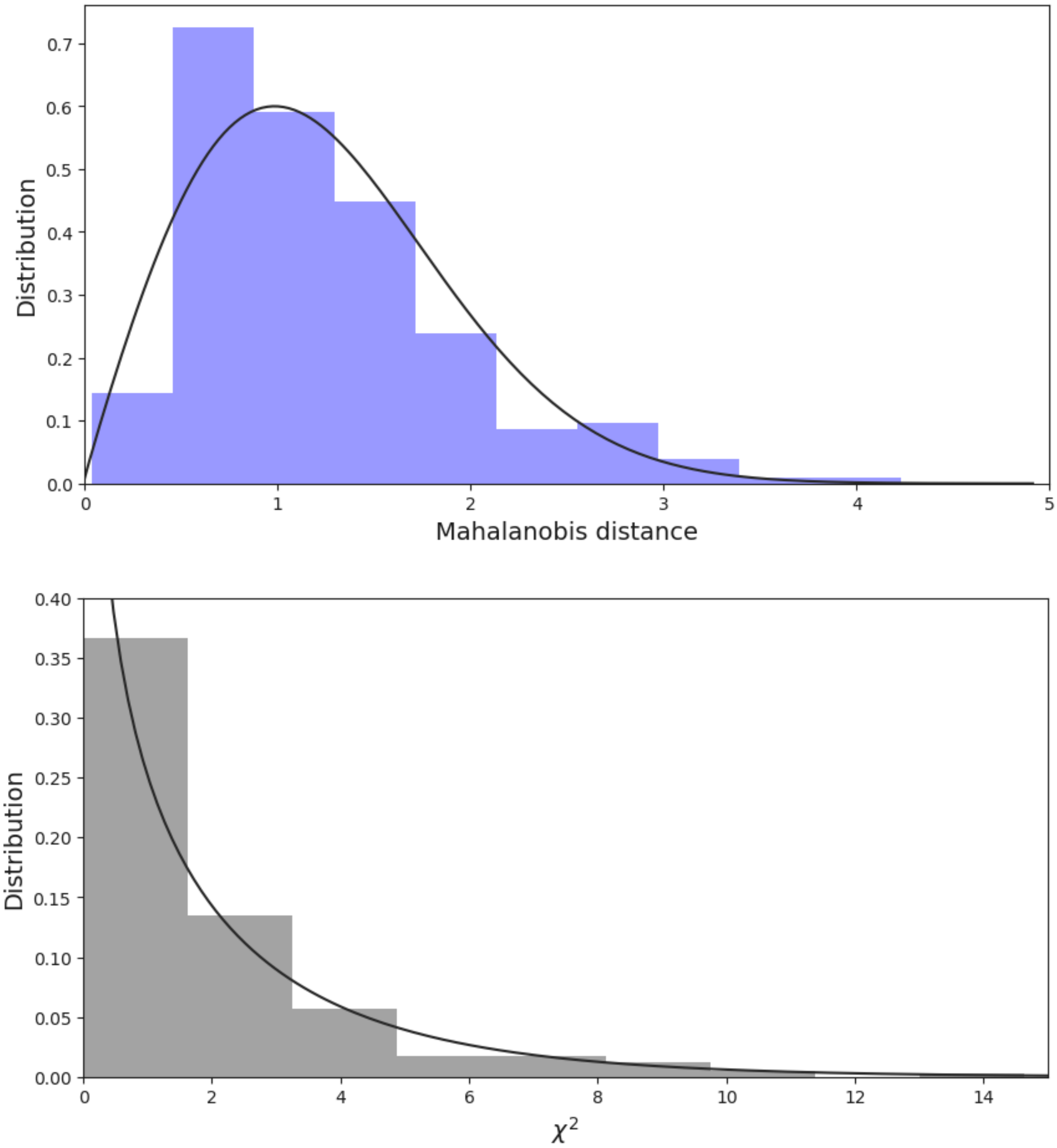

Figure 4.14: Probability distribution of the Mahalanobis distance (top) and *squared* Mahalanobis distance (bottom) for healthy samples. The dark lines are a fit to the data to guide the eye.

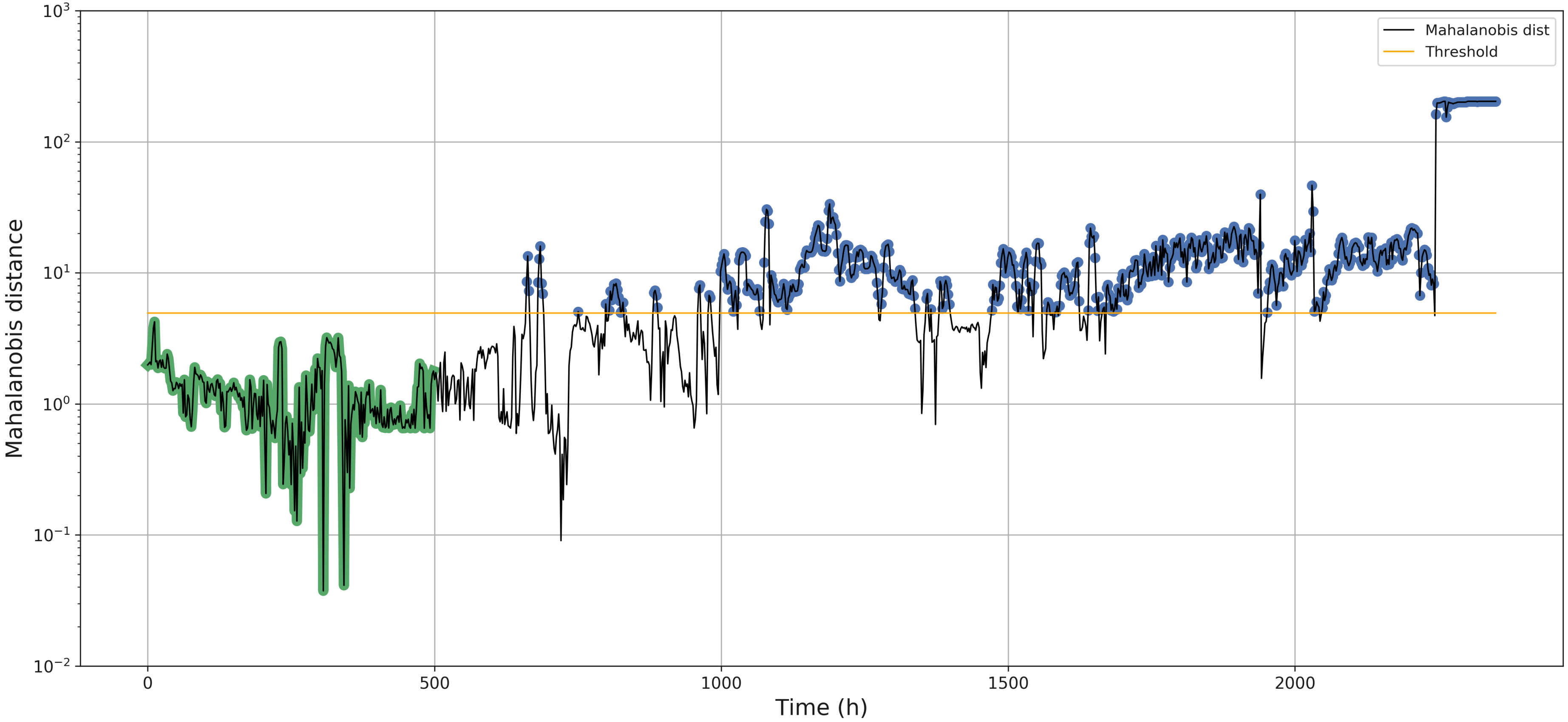

Figure 4.15: Anomaly detection by means of principal component analysis and multivariate Gaussian modeling. The Mahalanobis distance is chosen as the anomaly measure for the turbomolecular pump at position TT01 from March 1, 2018 until the failure on June 2, 2018. The green samples represent training samples, the blue dots are flagged as anomalies. The threshold is shown as an orange line.

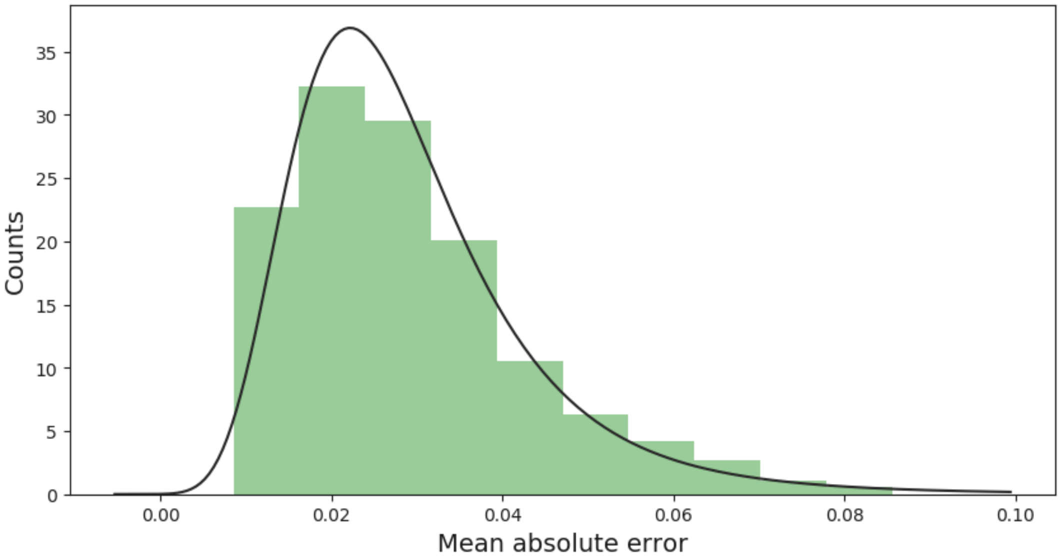

Figure 4.16: Probability distribution of the mean absolute error for healthy samples obtained with the auto-encoder neural network. The dark line is a fit to the data to guide the eye.

The activation function is the 'exponential linear unit' (ELU). The popular 'Adam' optimizer has been used for training the neural network. A train-test split of 4:1 is chosen. The loss function used is the mean squared error (MSE). This is because it is a differentiable function, and the mean absolute error is not. More discussion of the model parameters is provided in section 4.5. The learning rate and number of epochs depend on the input parameters, but generally a learning rate of about 0.001 and 120 epochs are used. The batch size is 16 samples. Features are again first rescaled to values between 0 and 1 and downsampled for noise reduction.

Finally, the results for the anomaly detection algorithm with auto-encoder neural networks are shown in Figure 4.17. A window size of ten hours and a hop size of two hours are applied. No time derivatives are included, only the classic input values. In Figure 4.16, the distribution for the mean absolute error is shown for the healthy samples (0 to 500 hours). The threshold is chosen to be 0.5.

Similar results for the second failure are shown in the appendix. The first, fourth and fifth failure are not eligible for analysis with the previous techniques, due to too little healthy data prior to the failure.

## 4.5 Discussion

This section analyses the results and assumptions made about the model. The multivariate Gaussian assumption from model 1 will be analyzed, and a comparison between the two models is made. The auto-encoder model is deemed most versatile and powerful, and the discussion continues focused on this model. Since model 2 depends on multiple parameters, their influence is discussed. Next, a possible application of the model in operations is explored. The model of course still has some shortcomings, and the origins of those are discussed together with possible

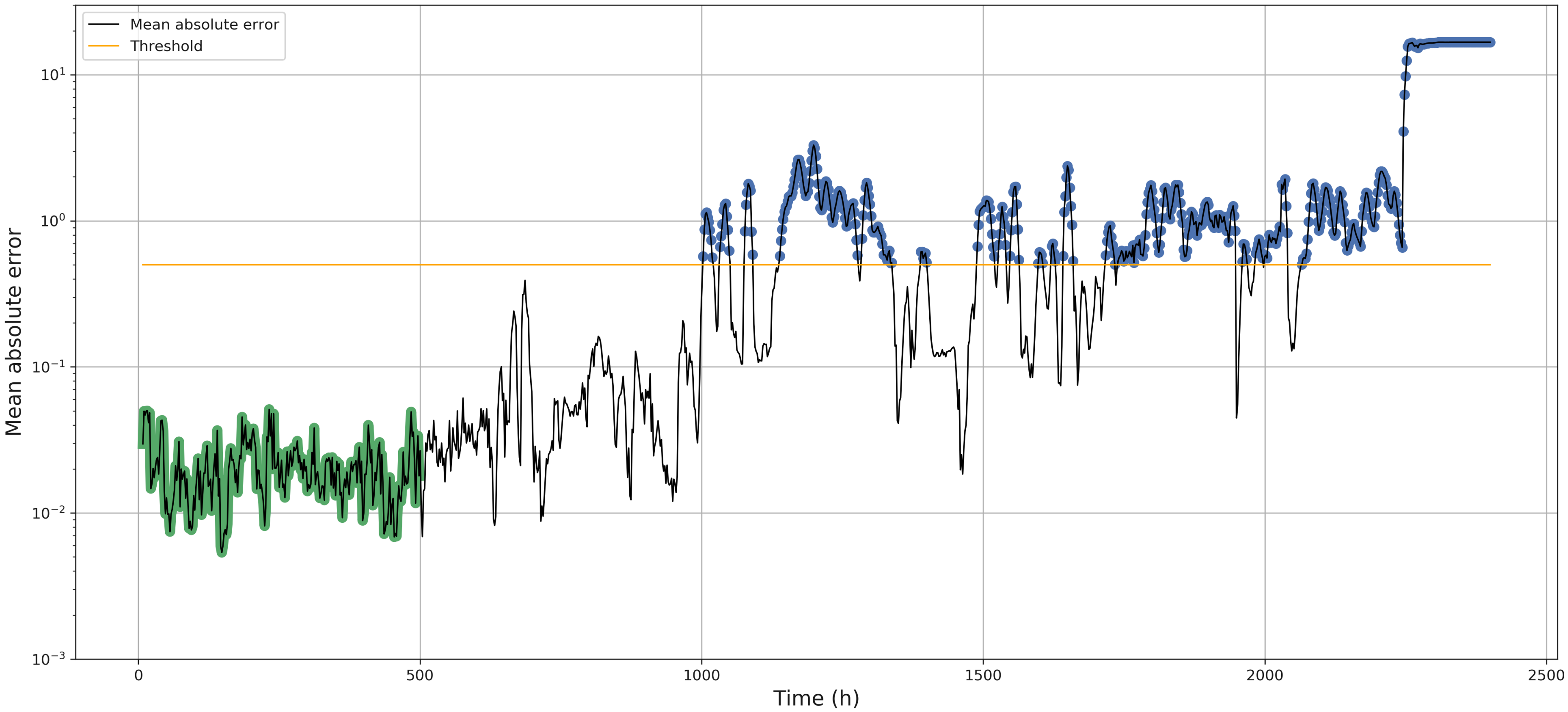

Figure 4.17: Anomaly detection by means of an auto-encoder neural network. The mean absolute error is taken as the anomaly measure for the turbomolecular pump at position TT01 from March 1, 2018 until the failure on June 2, 2018. The green samples represent training samples, the blue dots are flagged as anomalies. The threshold is shown as an orange line.

extensions.

### 4.5.1 Multivariate Gaussian assumption

The first model assumed that the data is distributed according to a multivariate Gaussian distribution, or at least approximates this. To get an idea of the soundness of this assumption, the distribution of the squared Mahalanobis distance for healthy data can be studied in Figure 4.11. A $\chi^2$ distribution depends on the dimensionality of the data, as illustrated in Figure 4.18. The fitted distribution from Figure 4.11 seems to correspond to a two-dimensional $\chi^2$ distribution, which validates our assumption of an approximate multivariate Gaussian distribution. This is, however, only a crude test, and the plot of the healthy data from Figure 4.13 shows that the underlying distribution might be described by an approximate joint normal distribution, but the discrete rounding of the temperature values already distort the symmetry of the distribution.

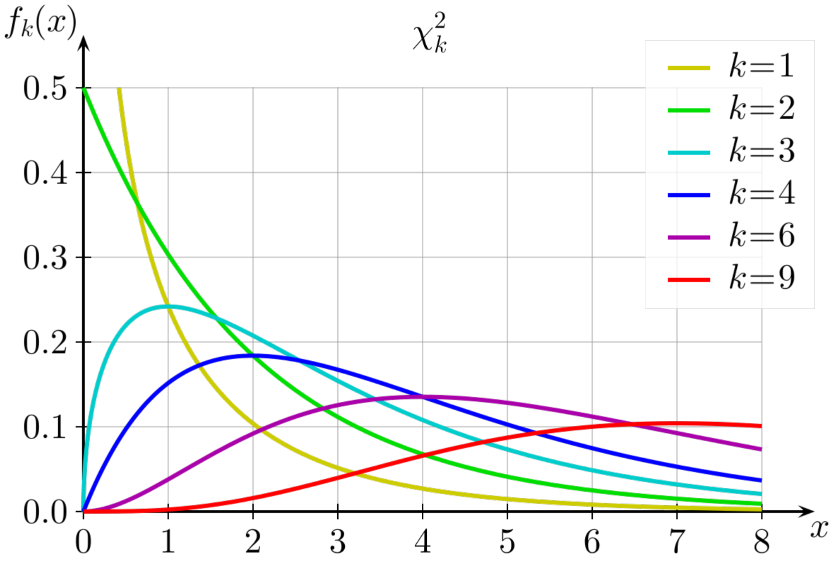

Figure 4.18: $\chi^2$ distributions for several dimensionalities $k$.

### 4.5.2 Comparison of the models

From the two-dimensional plots of different health regimes in Figure 4.13, a shift from a healthy regime to an increasingly unhealthy regime is visible in the two principal dimensions. This suggests that the statistical Mahalanobis distance model should be able to find some distinction between the samples. This is indeed the case, as the Mahalanobis anomaly detection results show in Figure 4.15: a general rising trend in the anomaly score can be seen, leading up to a strong peak at the end for the actual failure. Still, the distinction between normal and anomalous behaviour is rather modest. When the auto-encoder result in Figure 4.17 is inspected, the distinction between the regimes is much clearer. The anomalous regime begins only after about 1000 hours, which corresponds to a more intuitive notion of change in the sensor measurements, as seen in the plot of Figure 4.4. This suggests that the auto-encoder network has learned to recognize healthy behaviour quite well, and makes a more informed decision about flagging data as anomalous. This is not unexpected, since the auto-encoder network has the main advantage of taking time correlations into account, and also better captures the non-linear interactions between the variables, compared to PCA. The model also flags less false positives, in contrast to the first model. An increasing trend in mean absolute error is visible, eventually leading to the failure at 2246 h. Both models recognize the moment of failure really well, with a strong anomalous peak indicating behaviour that is not even remotely similar to what has been encountered before.

Based on the comparison between both models, the first model can be considered as the baseline model, while the auto-encoder neural network is more advanced and yields better results. The first model was interesting for exploring the data, being able to visualize what is happening under the hood and by working with intuitive concepts like normal distributions and the Mahalanobis distance. In contrast, what a black box model like the neural network actually *learns* is hard to explain, since knowledge is captured through adjusting the multiple weights. Even though the auto-encoder model is a more hermetic approach, it delivers clear results and is superior to the first model. From now on, we will focus on the second model for further analysis.

### 4.5.3 Model parameters for the auto-encoder neural network

There are several adjustable parameters present in the auto-encoder model. Their impact will be discussed here. First, the amount of nodes and layers in the neural network depends on the complexity of the task at hand. If a very complex relation is to be learned, or different relations are to be learned in the same model, the complexity of the neural network needs to increase. Generally, this leads to an increase of the amount of layers in the network. This is also one of the main ideas behind deep learning. Here, the option of a modest depth was chosen, as proposed by Kuzin and Borovicka [37], as the complexity of modeling the relations in the healthy data seems not complicated enough for deep architectures. As for the amount of nodes per layer, a similar reasoning applies. The main question to address is: 'how much data compression is needed to achieve the desired result?'. A strong compression means a smaller amount of hidden nodes. This forces the network to learn more abstract relations, and not have it rely on just memorizing samples. But if too much compression is required, the model will underperform, simply because it cannot capture enough of the complexity of the relations in the small amount of nodes it has. Again, the settings from [37] were implemented, since they provided good results compared to other settings.

The exponential linear unit (ELU) was used as an activation function for the hidden layers. The advantage compared to traditional ReLU activations, is that ELU allows negative values [38]. The ELU activation function is drawn in Figure 4.19. It is suggested [39] that ELU might give better results for reconstruction in auto-encoders.

The size of the sliding time window influences how many previous samples are taken into account. Increasing the size might give to some extent more context to the algorithm, but the neural network size increases at the same time, which is not optimal. Another effect of increasing time window size, is the smoothing of the results. This helps to show a more general trend with reduced noise. Of course, this effect is only helpful to some extent, since certain large-scale ‘noise’ is actually valuable information for analyzing the results, and thus not everything can be smoothed out.

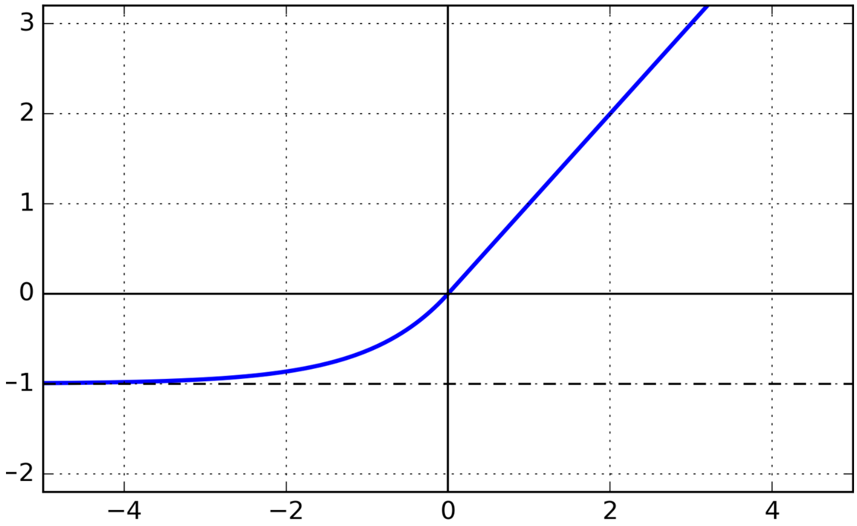

Figure 4.19: ELU activation function.

Another time window parameter is the hop size $h$ ($h \leq n_w$). Increasing the hop size, in our case, does not seem to alter the anomaly detection trend much, only the interval between the assessment of subsequent samples is increased.

The first or second time derivatives were not used for the results shown in the previous section (yet, they were implemented for generality purposes). The time derivatives on their own are not very informative, as shown in Figure 4.20 for the first time derivatives (the second derivatives look very similar). However, both derivatives combined with the original input values are shown in Figure 4.21 and provide meaningful results, albeit very similar to the results without derivatives. This would suggest that the neural network focuses mostly on the original inputs anyhow. Therefore, as the machine learning adage goes, the simplest solution was chosen that explains the observations well, which also has a three times smaller network size compared to the combination with the derivatives (and more importantly, about $3^5 = 243$ times less connections to optimize between neurons).

### 4.5.4 Application of the model in operations

In order to implement the anomaly detection model to aid in fusion operations, some adjustments need to be made. One way the algorithm would not be practical for real-life use, is when one or a few sporadic registered anomalies already trigger an alarm for the operator. To combat such false positives, a basic voting system can be implemented on top of the algorithm that only triggers an alarm when, e.g., 7 out of 10 subsequent samples are flagged as an anomaly. As a result, the model becomes more robust to possible false positives.

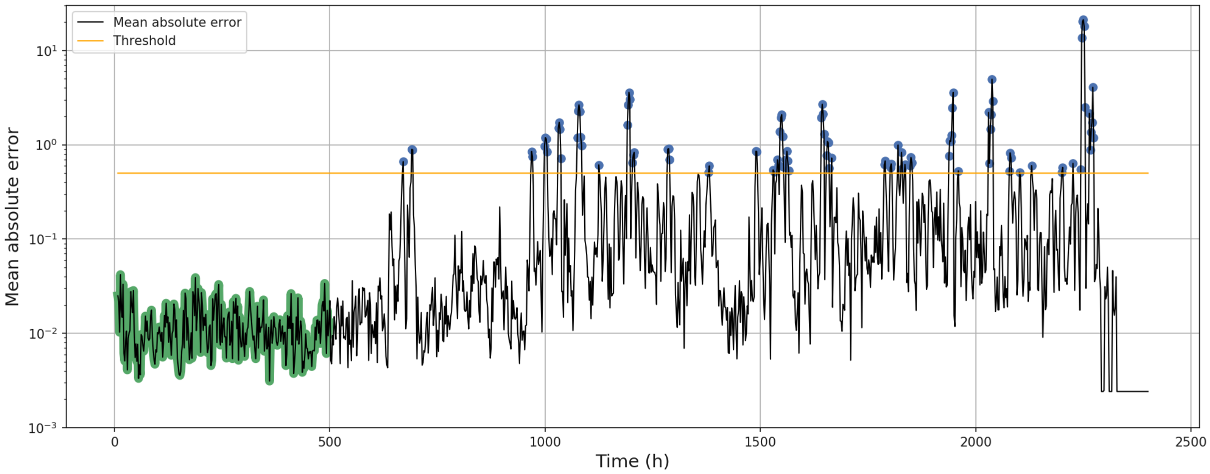

Figure 4.20: Anomaly detection results obtained with only the first time derivatives of the sliding window as input. Results are shown for position TT01 from March 1, 2018, until June 2, 2018.

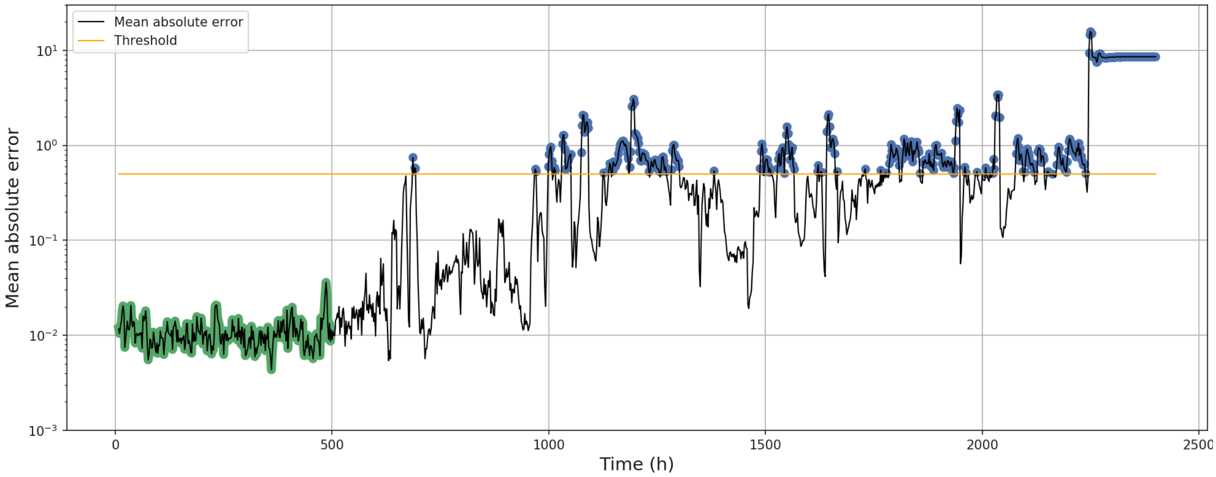

Figure 4.21: Anomaly detection results obtained with the combination of raw input values and the first and second time derivatives of the sliding window. Results are shown for position TT01 from March 1, 2018, until June 2, 2018.

After careful consideration of the results, it seems like the increase in anomalies in later life stages of a pump does not exclusively signal a process of degradation; it is also an indication of conditions in which it becomes dangerous for the customized pumps to operate in. When these ‘danger zones’ are combined with the subtle degradation of the pump, the chances of failure increase. As mentioned in section 4.3.2 on Remaining Useful Lifetime, equipment failure is a random process, with chances of failure increasing with every time step. However, the probability curves from the examples in 4.3.3 were calibrated on systems working in the same operating conditions. When the operating conditions change throughout a system’s lifetime, another probability curve applies. In this case, certain operating conditions increase the chance of failure compared to an earlier environment and shorten the expected life time of a pump. The regions of high anomaly scores appear to be linked with periods of experiments being performed.

To check this, a list of JET experiments with time stamps was used. This suggests that the extensive stress of multiple experiments being performed in a small period of time most probably increases the chances of failure. It also appears that the significant effects of the experiments linger on even after they have ended.

Although experiments appear to have lasting effects on the condition of the pump and its degradation, the question arises if they are to be flagged as anomalies themselves. Indeed, some of the peaks in the original sensor data are the pump responding to experiments (as was already mentioned in the data properties section) and these peaks are often flagged as anomalies. It comes as no surprise that experiments are seen as anomalies: to the algorithm, they represent seemingly random radical changes in the sensor data. So in a way, experiments are also false positives. Their expression in the sensor data is too erratic and does not represent normal – as in standard operational – behaviour. Therefore, a strategy to omit or soften the impact of experiments on the anomaly detection algorithm is adopted. This enables the model to focus on the lasting effects from the aftermath of experiments and spot a more general trend in the data towards failure. A few ways to implement this are:

- Create a separate classifier algorithm that detects experiments and passes this information to the primary algorithm, which then omits flagged anomalies around that time period.
- When JET starts a new experiment, an automatic signal can be sent to the primary algorithm, which then omits flagged anomalies around that time period. While the previous example is more illustrious, this approach is favored over the previous one; although it is not machine learning, it is highly accurate (∼100% correct), while a machine learning algorithm will report lower accuracies (say, about 90%) and thus might miss some experiments.
- As discussed already, an alarm will only be triggered when, e.g. 7 out of 10 subsequent samples are flagged as anomalies. As a result, even without using the previous proposals, a single experiment will not influence the alarm very much and cause a false positive (if it does not have too great of a lasting impact on the pumps, of course, because it *is* the intention to catch these changes).
- The auto-encoder algorithm uses a sliding time window as input. This means it also takes temporal correlations into account. If a time window contains one or more experiments, they will be softened by the other non-experiment data points. Sliding time windows seem to smooth the anomaly curve in general.

One last important element that influences the alarm is the threshold value. Since the auto-encoder algorithm learned to get very good at reconstructing healthy behaviour, the mean absolute errors for healthy samples are small. Consequently, the error distribution is much more dense, as shown in Figure 4.16, while the variable behaviour leading up to the failure is further removed from this dense distribution as a result[5]. It seems more natural then to only start

[5] In contrast to the Mahalanobis model, where both error regions seem to connect more.

flagging samples as anomalous if the reconstruction errors are further away from the healthy distribution, so as to not flag acceptable behaviour. Based on a visual inspection of the sensor data, together with the anomaly results of the second failure and the natural transition of the error pattern around ~1000 h, a threshold of 0.5 is obtained. The statistically obtained threshold of the Mahalanobis distance was also used as a guide. This is a heuristic way of determining the threshold, based on what is available from the data. It can certainly be fine-tuned by people with expert knowledge of the operations of the pumps. An even better approach would be to use data from more pumps with long-term operations that eventually led to a failure. The latter would be a rather costly gathering of new experimental data, if done for the sole purpose of fine-tuning a model parameter. However, if a general threshold could be obtained from many pump failures, and it provides a reliable average guess of the remaining useful lifetime[6], a more advanced form of predictive maintenance is attained. Another possibility might be to do away with thresholds as a whole, and simply use the error curve as extra information for operators to base decisions on.

### 4.5.5 Shortcomings and possible extensions

There are likely still more inventive ways to approach the problem presented here, but one hurdle in machine learning that is difficult to overcome with any approach, is the challenge of working with imperfect data. In an ideal world, data is always gathered specifically with analysis in mind. For the presented algorithms, it would be advised that in the future, each pump is first employed under quiescent conditions after installation, to gather at least an acceptable pool of healthy data to ground the algorithm on. Another possibility would be to gather data under different controlled conditions, each for a sufficiently long time, and compare the obtained models to see if different patterns emerge. This might eventually lead to more information for fine-tuning model parameters, or even do root cause analysis. Another interesting addition would be the implementation of a widely used predictive maintenance sensor: the vibrational sensor. An overview for predictive maintenance with vibrational sensors is provided in [40]. An interesting excerpt from the overview is the following:

> *"Interpreting the vibration signal obtained is an elaborate procedure that requires specialized training and experience. It is simplified by the use of state-of-the-art technologies that provide the vast majority of data analysis automatically and provide information instead of raw data. One commonly employed technique is to examine the individual frequencies present in the signal. These frequencies correspond to certain mechanical components (for example, the various pieces that make up a rolling-element bearing) or certain malfunctions (such as shaft unbalance [sic] or misalignment). By examining these frequencies and their harmonics, the [condition monitoring] specialist can often identify the location and type of problem, and sometimes the root cause as well. For example, high vibration at the frequency correspond-*

[6] It could, e.g., be set close to the failure, or multiple thresholds can be set: one indicating a near-failure, and an earlier one indicating suboptimal working conditions.

> *ing to the speed of rotation is most often due to residual imbalance and is corrected by balancing the machine. A degrading rolling-element bearing, on the other hand, will usually exhibit vibration signals at specific frequencies increasing in intensity as it wears. Special analysis instruments can detect this wear weeks or even months before failure, giving ample warning to schedule replacement before a failure which could cause a much longer down-time. Beside all sensors and data analysis it is important to keep in mind that more than 80% of all complex mechanical equipment fail accidentally and without any relation to their life-cycle period."*

The data from vibrational sensors is often combined with temperature and power data, so they would be a fitting addition to the existing arsenal of sensors and open up a range of new techniques.

Both presented models use direct sensor features. Although features are engineered through PCA and the encoding part of the auto-encoder neural network, several direct feature engineering options exist that could have been used on top of the original features. Some examples are: Fourier transforms, wavelet filters, window statistics, etc.

A drawback of both presented models is the need for healthy data at each new installment. Combining or cross-using sets of healthy data from different pump installments was tested and produced no sensible results. This might indicate that data properties are different for every pump installation, or this is simply a shortcoming of the models. Since the first, fourth and fifth failure have a short 'runway' before they fail, they were not eligible to be used by these models due to a lack of healthy data. The first failure was a rotor blade loss accident, and could be considered unrelated to an underlying degradation process. However, if we look at, e.g., the fourth failure, the fast build-up to a failure appears to be a reaction to immediate subjection to intensive experiments.

Another limitation presented by the data is the challenge to validate the model with an appropriate validation metric. Mainly heuristic arguments are used to assess if a model does well, largely based on the expectations imposed on the model. For example, the idea that the auto-encoder model is better than the Mahalanobis model is partly based on the notion that there should be a smooth transition into a more anomalous region, and partly by comparing it with what is seen visually for the raw sensor values and what is known from operation conditions and experiments. However, the core of the techniques presented here have proven their value on several datasets not unlike these. On top of that, they assume very little about the data and therefore generalize well to many settings. The main challenges seem to be dealing with limited data, fine-tuning the model parameters and correctly interpreting the results.

# Chapter 5

# Predictive maintenance for S1 current switches

Besides the turbomolecular pump failures from the previous chapter, other parts from JET also experience failures and might benefit from a predictive maintenance approach. In this chapter, the example of S1 current switches is discussed[1]. The available data is visualized, and based on an exploratory analysis, suggestions for future predictive maintenance strategies are provided.

## 5.1 S1 current switches

The S1 current switches (also known as circuit breakers) on JET interrupt the current flowing from the Poloidal Flywheel Generator Converter (PFGC), which provides power during a JET experiment, to the central solenoid (P1), which uses this power to drive a current into the plasma (see also the introductory chapter on nuclear fusion). A schematic overview of the circuit between the PFGC and the P1 central solenoid is given in Figure 5.1. When the S1 switch is expected to interrupt the current, a capacitor bank is discharged as a counter to the switch current to generate a brief ($\sim$1 ms) zero current period at the S1 switch to reduce the arc energy and make the switch more reliable at interruption time. This second opposite discharge creating the interruption window is called the *counter-pulse*. The operation of the S1 switch usually takes about 7 ms (5-8 ms) from the time a command is given, to when the switch interrupts. However, this interruption time varies throughout the lifetime of a switch, and depends on the number of previous executed interruptions and how recently maintenance was carried out. Following a maintenance, a *jitter measurement* is taken to get an estimate for the operation time and the variation on this estimate. The obtained opening time of the switch is then programmed into the JET control system to fall soon after the start of the counter-pulse. For good switch operation, the interruption of the current happens very shortly after the opening of the switch, so that the whole operation happens well in the time window provided by the counter-pulse. However, due to degrading health of a switch, the opening time is often

[1] The first and part of the second section of this chapter are largely based on a first analysis of the switches provided by J. Stephens [42].

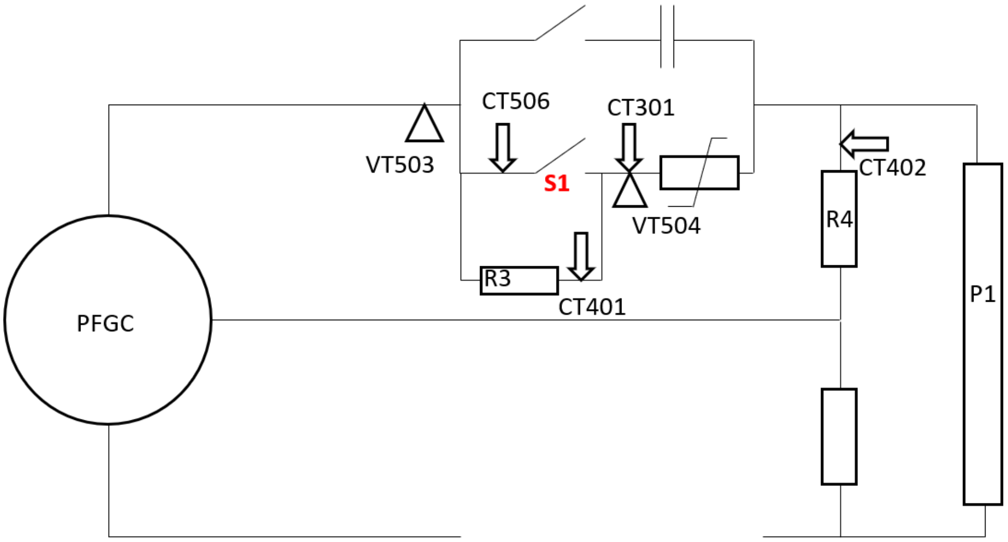

Figure 5.1: The circuit through which current is delivered from the Poloidal Flywheel Generator Converter to the central solenoid. The S1 current switch is visible in red [42].

later than planned. On top of that, a delay can happen between the opening and interruption time (typical delays are about ∼0.5 ms). Both the later opening time and the delay of the interruption can cause the interruption mechanism to fall outside the counter-pulse window. This results in the inability to interrupt the current successfully. The change in opening time can be programmed into the control system reactively over the operational life of a switch, so that the current interruption (hopefully) happens within the 1 ms time window. A trained operator can observe the data following an experiment and extract information on the health of the switch. However, this requires training, and time in between experiments, which is often not available, so an automated approach would be desirable. If failures and slow – but still successful – operations could be detected automatically in the future, this would provide useful information to the operator, signalling, e.g., signs of an aging switch, or multiple failures that need addressing.

## 5.2 Data properties

### 5.2.1 Signals

A number of signals are recorded from the circuit that are relevant to S1 behaviour:

CT506 records S1 current, sampled at 10 kHz,
CT401 records R3 current, sampled at 10 kHz,
CT301 records the sum of S1 and R3 currents, sampled at 10 kHz,
VT503 records the S1 voltage at the PFGC terminal, sampled at 50 kHz,
VT504 records the S1 voltage at the P1 terminal, sampled at 50 kHz.

CT signals represent a current, VT signals a voltage. Besides these signals, CT503 and CT402

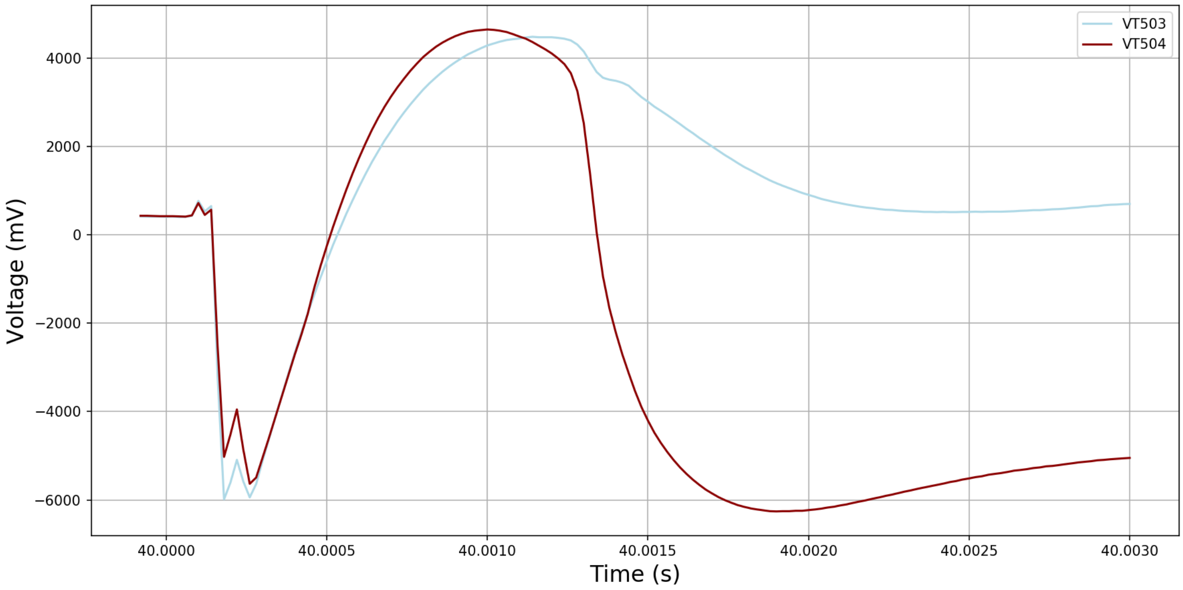

Figure 5.2: A successful current interruption. The switch opens at around 40.0004 s and clears the current promptly. Afterwards, the VT503 signal returns to the pre-counter-pulse voltage.

can be used for further validation or calibration of the above, but do not directly indicate S1 conditions. The higher sampling rate of the VT signals will provide more accurate information regarding the switch opening behaviour, and will be the main features used in the proposed models.

### 5.2.2 Switch behaviour

When the S1 switch is open, the VT503 and VT504 voltage signals will be separated by the voltage across the R3 resistance. In contrast, the voltage signals should be similar when the S1 switch is closed. This pattern is shown in Figure 5.2 for a successful current interruption. The time is expressed in seconds from the beginning of the JET pulse (experiment).

The VT503 signal is determined by the circuit condition relating to the PFGC, while the VT504 signal can either be related to the PFGC or to the P1 central solenoid, depending upon the position of the S1 switch. This dependence of the VT504 signal on the switch position is the key feature that can be used to asses the operation of the switch. If VT504 departs more suddenly from VT503, it can be assumed that at the same time there is a sharper transition in switch state. As the switch ages it is more common to see these abrupt current transitions, as the switch mechanism no longer takes full benefit of the counter-pulse. The opening of a switch is defined by the point at which the two VT voltages start to diverge. The interruption event is defined as the point when VT503 and VT504 each follow their independent curves (shown in Figure 5.2). An example of a slow opening and late interruption is given in Figure 5.3. This is not ideal behaviour, but still acceptable, since the current gets cleared and both voltages from then on follow their own independent curves.

The behaviour of the VT503 signal is similar for an early and late clearing of the current; it

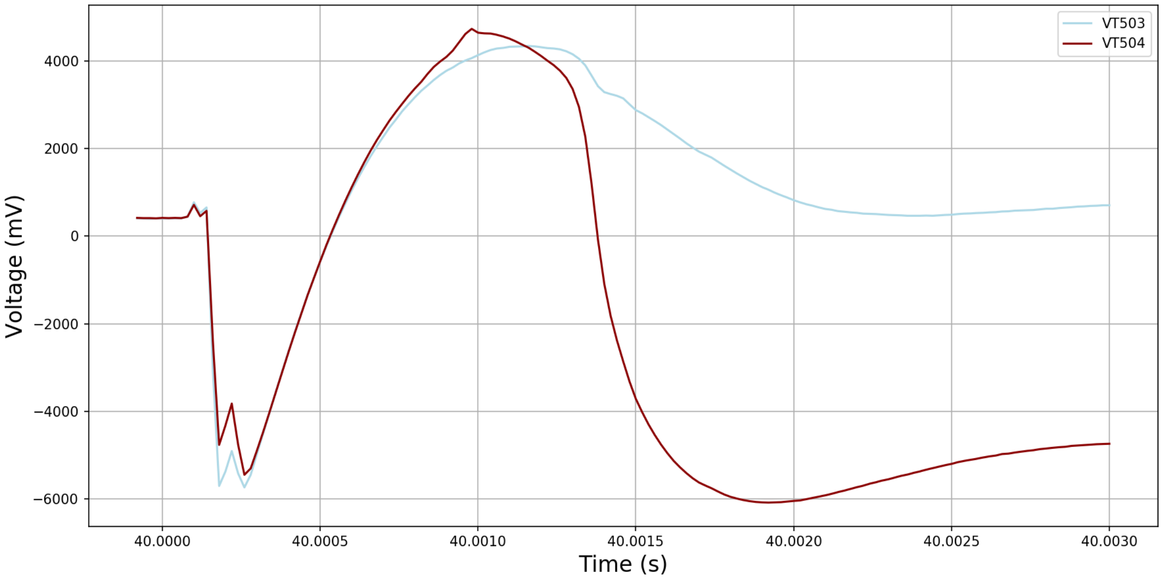

Figure 5.3: Slow opening and late interruption from the switch: the interruption is visible just before 40.0010 s. This is typical behaviour for an aging switch.

is the VT504 curve that sets them apart. For a switch *failure*, however, both VT503 and VT504 show a different behaviour. A failure occurs when the switch is unable to interrupt the current, or when a restrike[2] happens during the interruption. If at the end of the counter-pulse, VT503 does not follow a damped profile towards a similar voltage as the pre-counter-pulse level (like Figure 5.2 and Figure 5.3), a failure can be assumed. A typical failure where VT503 and VT504 show oscillating behaviour after the interruption window, is shown in Figure 5.4.

### 5.2.3 Inspection of the labeled available data

Thousands of pulses have been performed at JET the past few decades, with corresponding thousands of switch operations. An individual switch is used for ~1200 pulses before it is swapped out and refurbished. Sometimes a switch will work without failure for its entire lifespan, other times it has to be removed after only a few pulses due to constant failures. The ratio for successful vs. total failure operations is about 100-200:1 across every JET pulse to date. The ideal to marginal performance ratio is about 10:1 for the full switch history. By ideal performance, the opening of the switch soon after the beginning of the counter-pulse (~40.0004 s) is meant, while the opening at the end (~40.0008 s) is seen as marginal performance. When a switch opens very late, a failure becomes more likely.

The dataset used to conduct this exploratory research contains 291 samples (pulses), of which 128 are conveniently labeled by an expert. Out of the 128 samples, 7 were visually deemed uninformative outliers, and were removed from the dataset. The 121 remaining samples can roughly be labeled into four categories:

[2] A restrike is an interruption that was followed by a breakdown that restored the current flow, so the interruption was not successful.

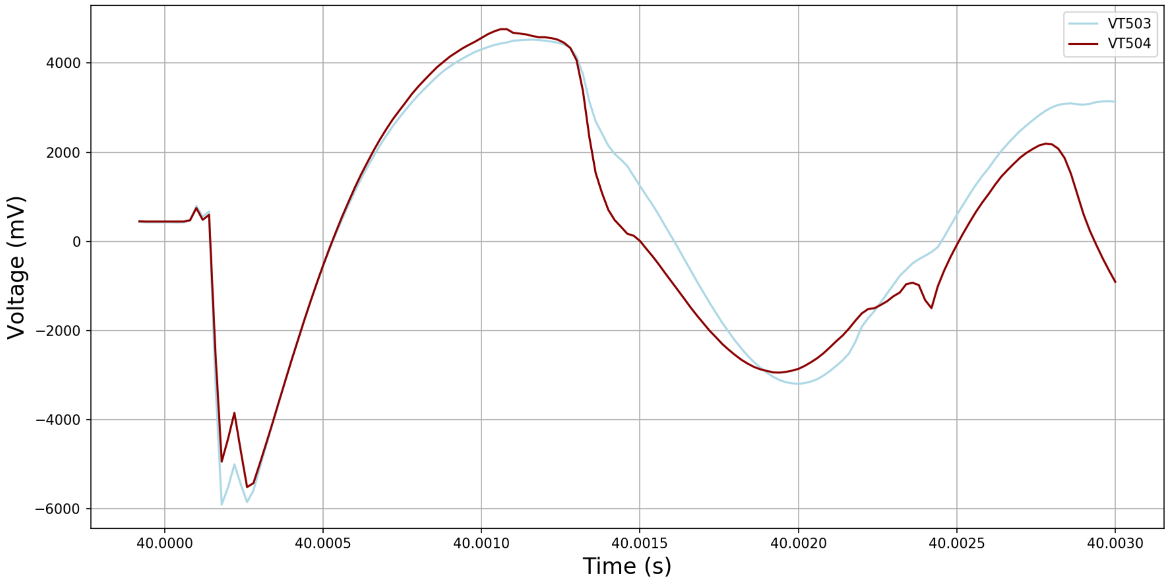

Figure 5.4: A failed interruption. Too late to open and too slow to clear the current, leading to arcing after the interruption window.

0 – Good: ‘A good operation. Fast to open and fast to clear current.’
1 – Intermediate: ‘Typical of an aging switch. A bit slow to clear current.’
2 – Slow: ‘Typical of an aging switch. Slow to clear current.’
3 – Failure: ‘A failed interruption. Too late to open and too slow to clear current.’

The class count is given in Figure 5.5. Note that the class imbalance for this dataset is artificial, and unfortunately not representative for the entire pulse collection. This dataset is labeled exactly because it contains a high percentage of failures and slow openings and was chosen for manual inspection to learn about aging switch behaviour.

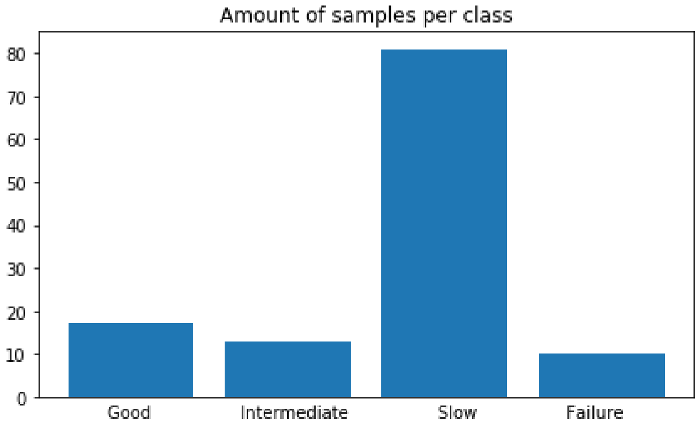

Figure 5.5: Histogram of the switch interruption classes.

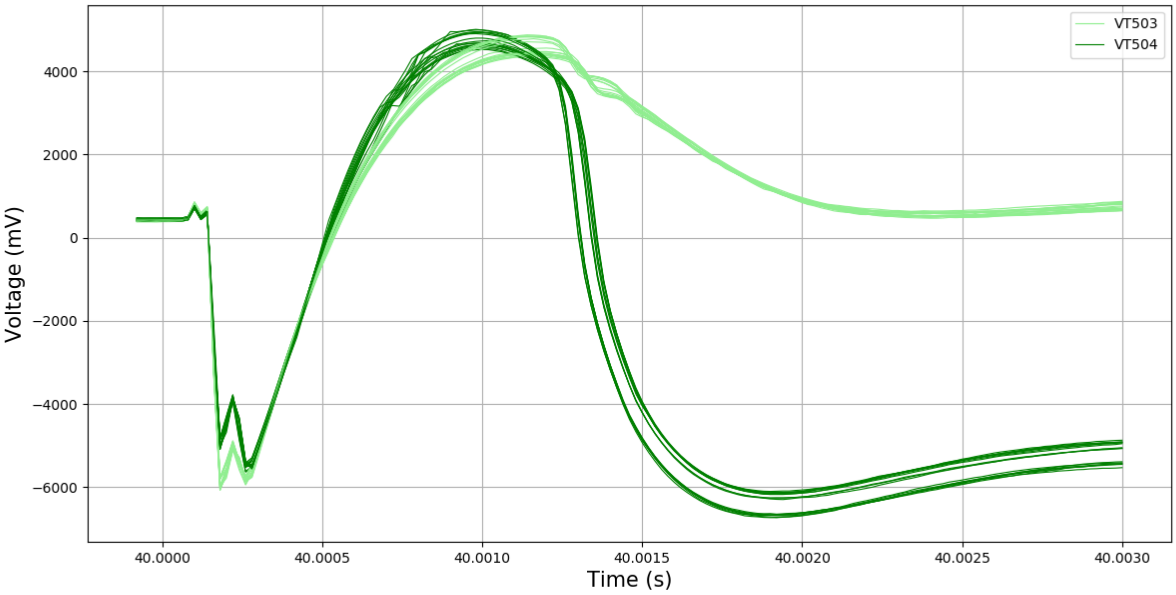

Figure 5.6: VT503 and VT504 voltage signals of good switch operations.

### 5.2.4 Comparison of classes

This section will give a comparison of the classes, so that a feeling of the differences between switch behaviour can be established.

In Figure 5.6, all 17 good operations are shown. Two dense line regions can be distinguished. This distinction is caused by the different circumstances under which switch operations take place, mainly different pre-magnetization currents. These circumstances essentially rescale the voltage vs. time pattern, but the behaviour of the switches stays the same. The noisy area of seemingly imperfect openings from 40.0007 s to 40.0009 s comes from operations where the switch was successfully opened, but still had a high resistance arc between the contacts, which shows up as a temporary closing of the voltage gap. Since these are still successful runs, a model will have to deal with this noise.

A comparison between the good switch operations, and the complete failures is provided in Figure 5.7. Again, several dense lines are formed, corresponding to different operating conditions. From now on, lighter colours represent VT503 signals, darker variants represent VT504 signals.

A comparison between slow and failure behaviour is given in Figure 5.8, and a comparison between good and slow behaviour is given in Figure 5.9. The data from the intermediate operations will not be shown here: they resemble slow behaviour, and can be treated as a small extra class besides the ‘regular’ slow operations. In the predictive maintenance discussion of the next section, they can always be added to the general framework as an extra class.

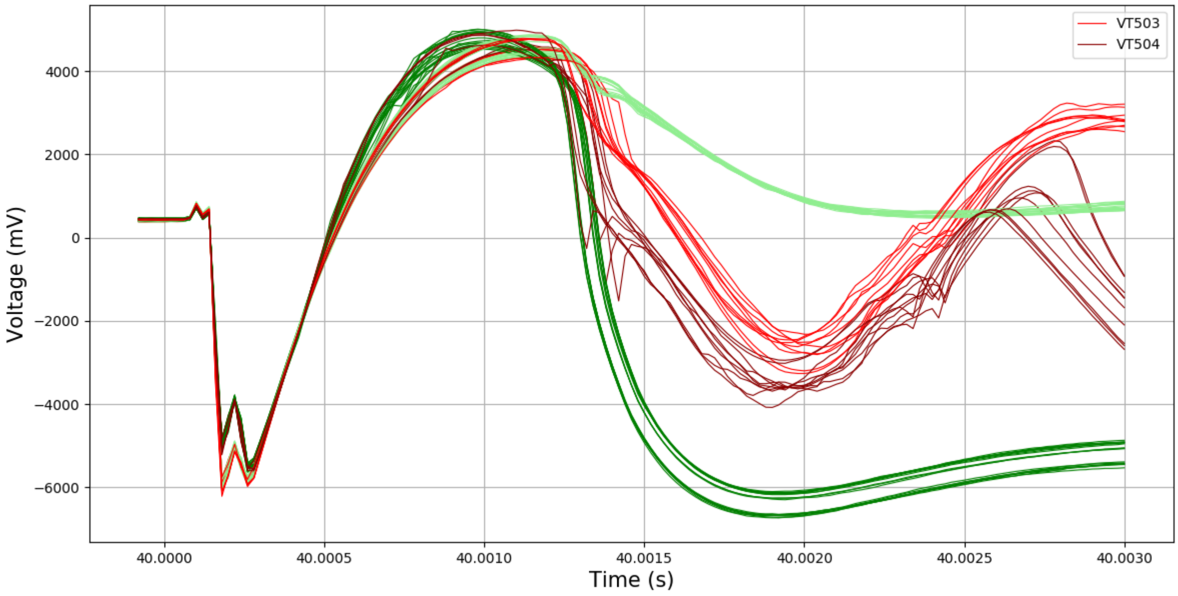

Figure 5.7: VT503 and VT504 voltage signals of good (green) vs. failure (red) switch operations.

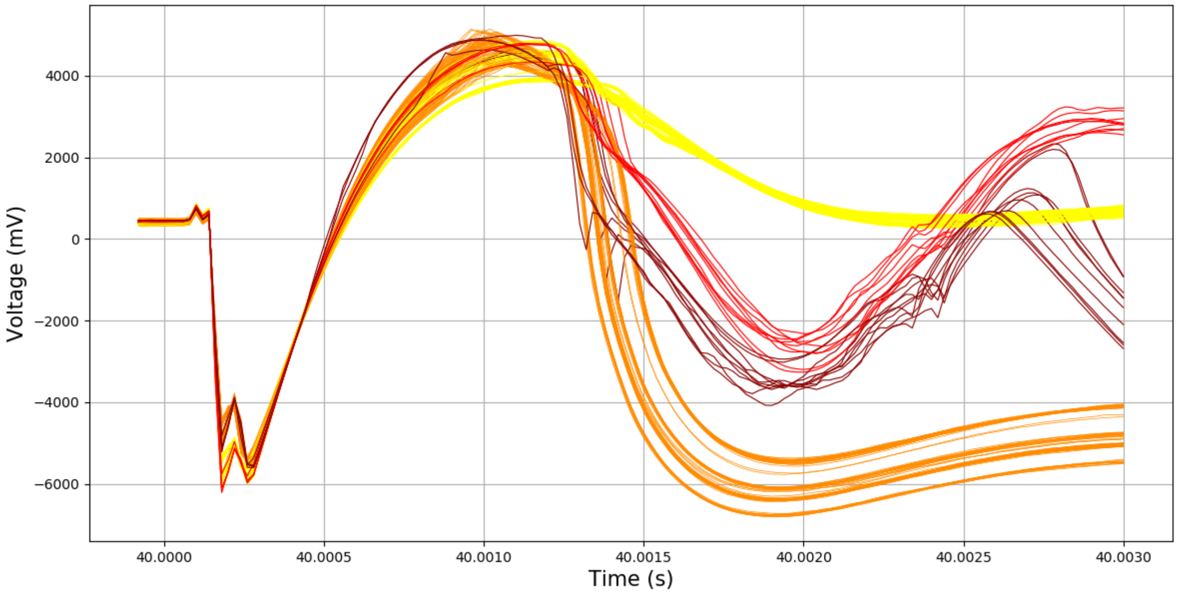

Figure 5.8: VT503 and VT504 voltage signals of slow (yellow/orange) vs. failure (red) switch operations.

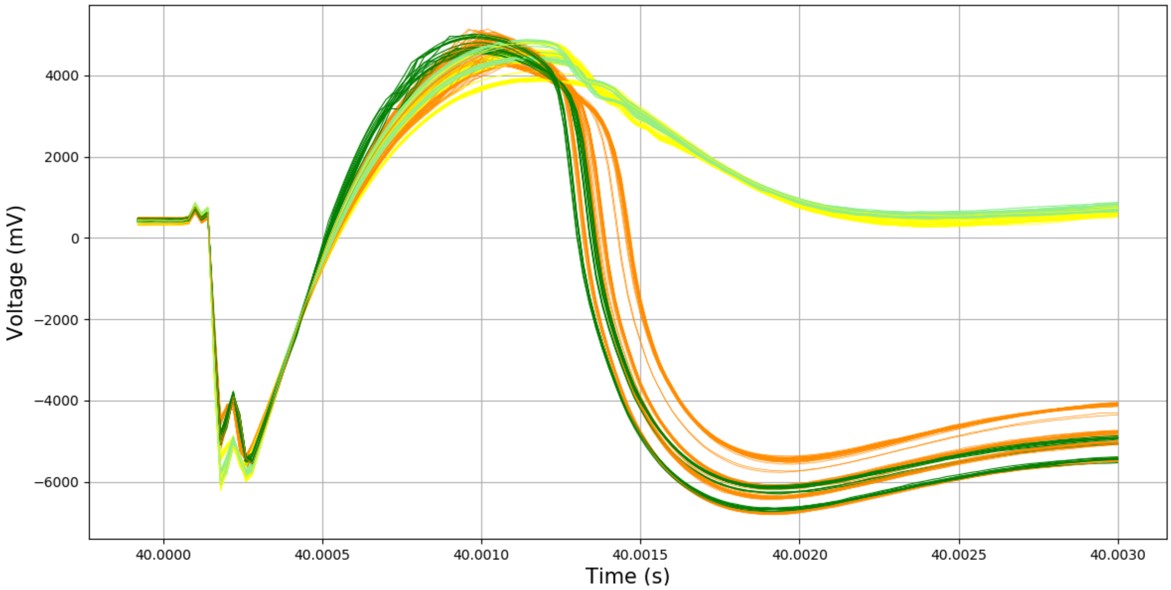

Figure 5.9: VT503 and VT504 voltage signals of good (green) vs. slow (yellow/orange) switch operations.

## 5.3 Suggestions for predictive maintenance

### 5.3.1 A simple classification model

When the signals from the three main categories – good, slow and failures – are compared, an immediate distinction between the failures and other signals is apparent (see Figure 5.7 and Figure 5.8) when the measurements from 40.0015 s and onward are considered. A simple threshold for the VT503 and VT504 signal at 40.0020 s would classify every failure vs. non-failure correctly. This simple solution can classify future failures without the need for more complex machine learning, assuming that the failures in the dataset are representative for the whole switch distribution, which is confirmed by [42].

Since the case ‘failure vs. non-failure’ is solved, the ‘good vs. slow’ classification still remains. This is more complex, as shown in Figure 5.9. Since there is very little data available, intricate solutions that learn the subtle nuances between the two regimes very well (like artificial neural networks), are out of scope. The problem is made even more challenging due to the different scaling of the curves, caused by the different operating conditions. One possibility is to find out how the initial conditions of a pulse relate to the scaling of the VT503/504 curves relative to others, and then rescale every curve accordingly so they become comparable. This might ease the classification, but requires research into the details governing the relation between operating conditions and the corresponding scaling. Another approach that does not require such knowledge, is to do a simple form of feature engineering by subtracting the VT504 with the VT503 signal and look at the relative growth of the voltage gap. The result of this subtraction for all good and slow switch operations is provided in Figure 5.10. The tails of the differences still are hard to distinguish correctly, but the time window from about 40.004 s until 40.0011 s

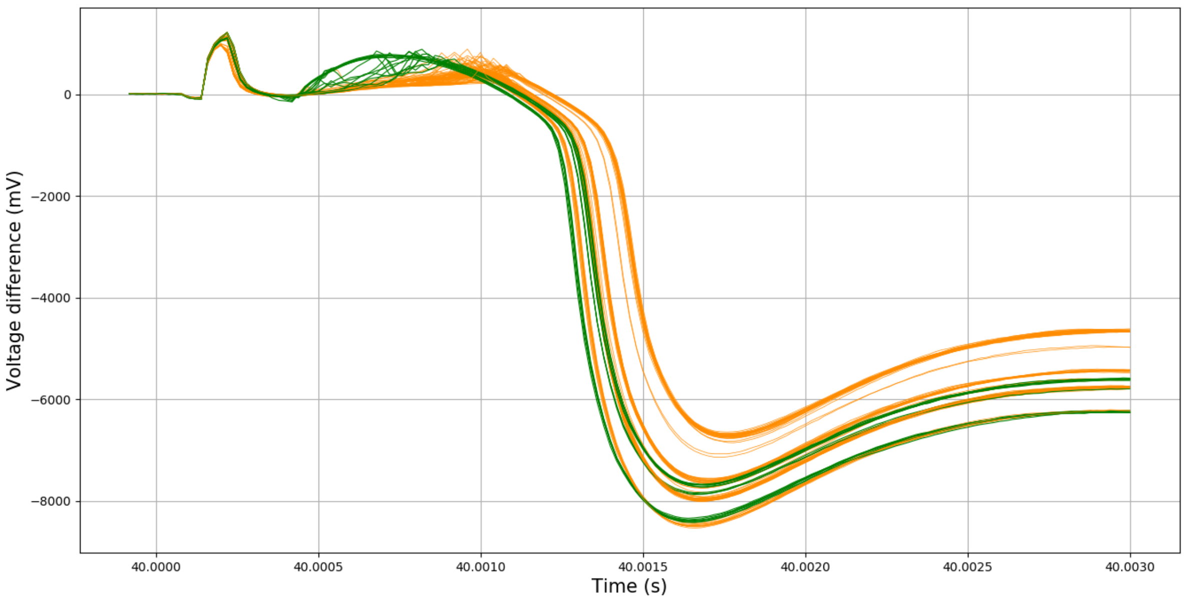

Figure 5.10: Difference between VT504 and VT503 voltage signals of good (green) vs. slow (orange) switch operations.

shows a clearer intuitive separation between the two classes. A close-up of this region is provided in Figure 5.11, where a separation between the two classes has emerged. Perfect openings and interruptions follow the arched concentration of green lines at the top. The green lines that show more irregular behaviour are the ones where the switch was successfully opened but still had a resistance arc between the contacts. We will leave these samples in the dataset to make the model more robust, since such occasions are likely to happen again in the future. Still, there is an intuitive difference visible even between the noisy good operations and the slow operations. A simple logistic regression classifier was trained on the time window between 40.0004 s and 40.00105 s, where every time step is considered as a new dimension in a constructed feature space. The resulting classification problem is 32-dimensional, due to the 50 kHz sampling rate. Every point in this 32-dimensional space represents a complete time window and thus one switch operation. It is assumed that the good operations will occupy a different region in this space than slow operations. The model was cross-validated and used ridge regression for regularization. It made a 50/50 stratified train-test split, so that the test results would hold some statistical significance. The resulting accuracy of the model is 98%. Since we are dealing with a strong class-imbalance, accuracy unfortunately is not a very good performance measure: a naive classifier could just assign every test sample to the 'slow' class and still get 82% accuracy. Therefore, results for more reliable performance measures are shown in Table 5.1. We are neither more interested in the recall, nor the precision, so the F1-score[3] is taken as the general performance measure for both classes. With only 10 good and 39 slow operations as training samples, attaining F1-scores of 0.92 and 0.99 respectively is already a promising result. Still, some caution is warranted, since the results are only based on 7 good and 42 slow test

[3] The F1-score is the harmonic mean of the precision and recall.

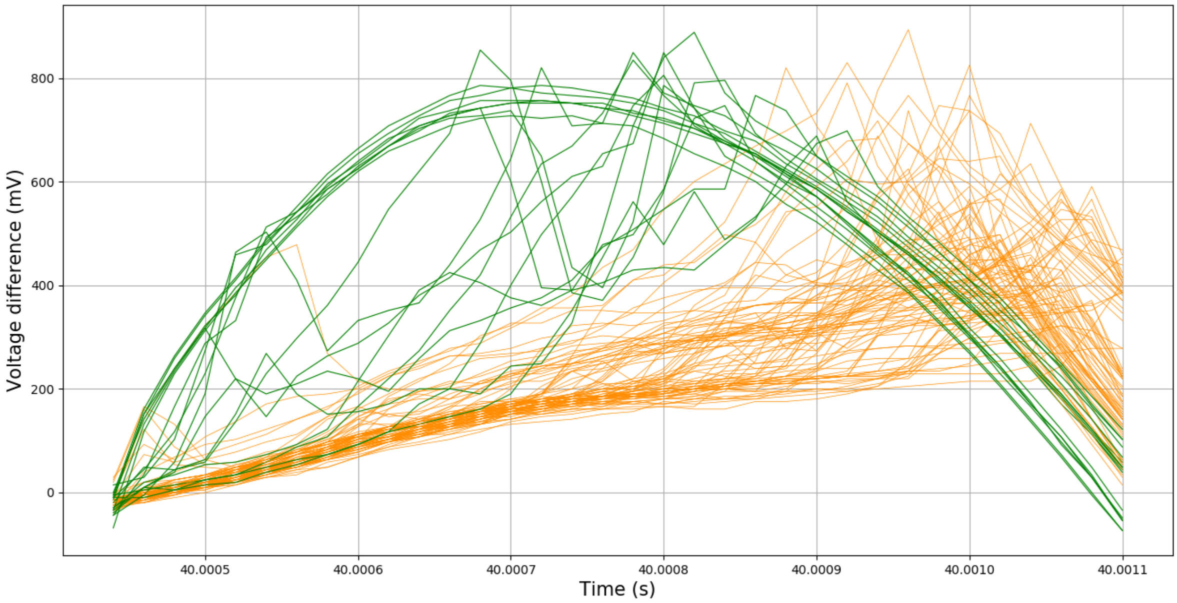

Figure 5.11: Difference between VT504 and VT503 voltage signals of good (green) vs. slow (orange) switch operations from 40.0004 s until 40.0011 s.

samples. Different dimensionalities, train-test splits, and cross-validations were tried, and all obtained comparable results. However, it would be desirable to try this approach on a dataset with at least three times the amount of good samples. In general, more data is always better.

Table 5.1: Precision, recall and F1-scores of the good and slow test samples.

| **Class** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| 0: Good | 1.00 | 0.86 | 0.92 |
| 2: Slow | 0.98 | 1.00 | 0.99 |

The final classifier is a hierarchical model, where the first layer classifies failures vs. non-failures (probably with ~100% accuracy), and if the sample is a non-failure, a further distinction can be made by the second classifier. For the first layer, the separation between failures and non-failures is even clearer for the VT504/503 difference plots, as shown in Figure 5.12.

### 5.3.2 Semi-supervised learning with historical pulses

The logistic regression classifier presented here has only a very limited dataset to train on. On top of that, the dataset is not representative of the real ratios between the categories. There is, however, a vast dataset of switch operations available, and these unlabeled samples can be used to enhance the performance of the simple classifier proposed in the previous section. This is a form of semi-supervised learning, and this section will explore the basic outlines of a few possible approaches.

One simple idea is to use the classifier from the previous section to classify all thousands of

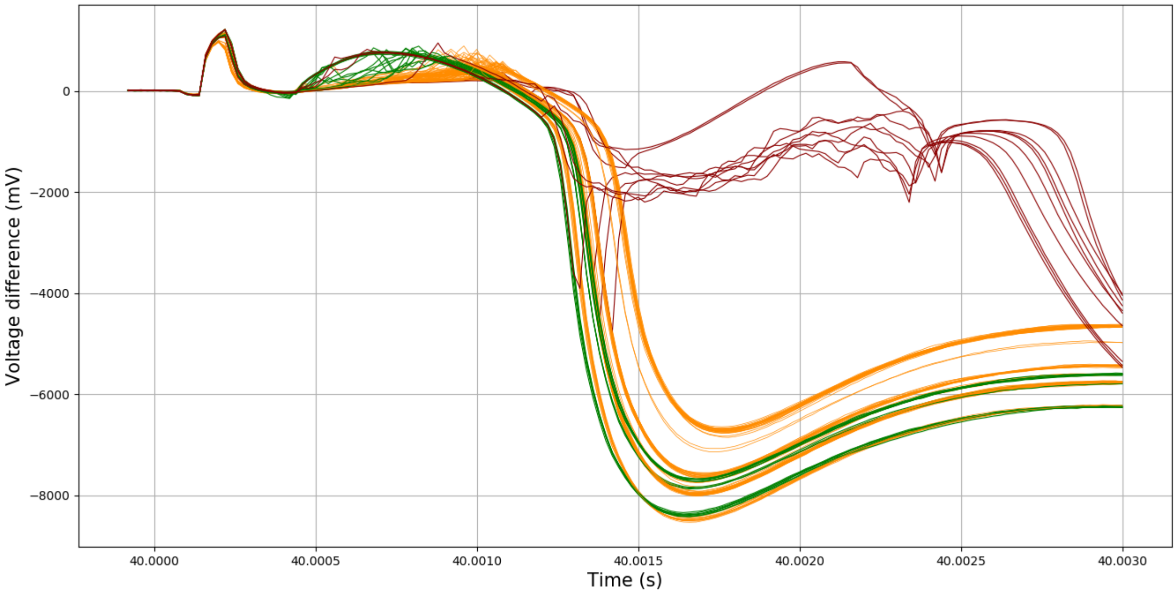

Figure 5.12: Difference between VT504 and VT503 voltage signals of failures (red) vs. non-failures.

remaining samples. By using soft classifications (probabilities as output), a part of the samples with very high class probabilities (high 'certainty') can be added to the labeled dataset. This larger labeled dataset can then be used to train a new classifier. This process is repeated until enough samples have been included and the classifier has been improved and made more robust. This is a simple form of an *expectation-maximization* [43] algorithm. The problem with this approach is that errors from the initial model might be ingrained more firmly in the resulting model by the process.

Another approach starts off with a clustering approach. If there are $k$ known classes in the data, an unsupervised clustering algorithm can search for $k$ clusters, which are afterwards classified by the human-labeled samples that each clusters contains. Then, a new classifier can be trained on this complete dataset which should obtain a better estimate of the decision boundary compared to only using the labeled data. An intuitive example is illustrated in Figure 5.13. Some assumptions are made when this form of semi-supervised learning is applied. Two often used assumptions are *smoothness* and *the cluster assumption*. Smoothness assumes that points that lie close to each other in feature space, are more likely to share the same label. The cluster assumption says that data points that belong to the same class are more likely to form a well-defined cluster in feature space. If these assumptions are true, it is likely that the classifier obtained with semi-supervised learning will outperform the one trained only on the small labeled dataset.

Some noteworthy clustering algorithms are k-means clustering [16] and expectation–maximization clustering using Gaussian Mixture Models (GMM) [17]. The k-means clustering algorithm starts with $k$ random cluster centers spread through the data. The samples closest to each cluster-center are assigned to this cluster. Now, the mean point in feature space for all the obtained

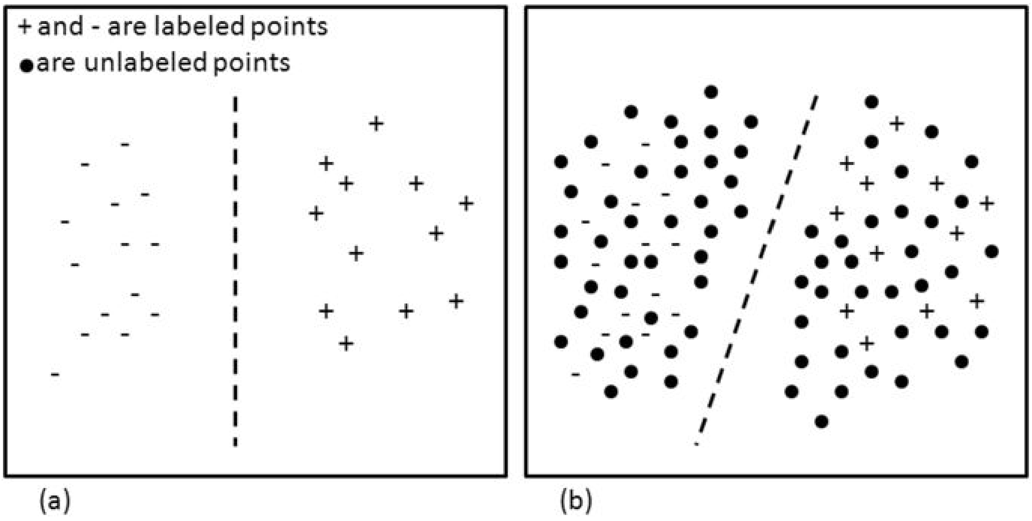

Figure 5.13: Semi-supervised learning tries to increase the generalization of classification performance by placing the decision boundary in between the dense regions in presence of both labeled and unlabeled data points. (a) The decision boundary in presence of labeled data points only, and (b) the decision boundary in presence of both labeled and unlabeled data. [44].

clusters is calculated, and these means are used as the new cluster centers. This process is repeated until convergence is attained. This alternation between the two steps of calculating the mean of a cluster and then defining new clusters based on this mean is another form of the expectation-maximization strategy. The GMM model uses the same approach, only more sophisticated. Instead of a simple mean, a multivariate Gaussian distribution is fitted to the clustered datapoints, and samples are assigned to probability distributions with a 'soft' probability score. A more complete and visual explanation of several popular clustering algorithms, including these two, is given in [45]. One advantage of the GMM model is that it models probability distributions. This means that samples are not restricted to one cluster, but are assigned to all clusters with varying probabilities. This might help to identify difficult cases: samples that have high uncertainty of belonging to one cluster can be manually inspected and labeled. This in turn will improve the final classification algorithm. Manually labeling only these few edge samples concentrates human efforts on the difficult samples and greatly reduces time requirements, while still obtaining a large reliable training dataset. The soft labeling can also address the more realistic scenario for the switches where samples are not strictly 'good' or 'slow', but somewhere in between.

The above approach supposes that the clustering algorithm has correctly converged to the real underlying clusters in a short amount of time. This is not always the case. A challenge for both models, especially the GMM model, is that in a high-dimensional space, the convergence of the algorithms is slow, and often many iterations of the algorithm are done using different random initializations to make sure that the algorithm does not converge on a suboptimal local minimum. A possible solution to avoid this time delay, is to do a *warm start* of the algorithm by using the means and possible standard deviations of the labeled examples as the initial cluster starting points. By guiding the clustering, a faster convergence can be attained. This might also help to avoid the clustering algorithms from converging on the 'wrong' clusters, by giving them a hint for where to start searching. In a highly-imbalanced dataset, the possibility also exists that the small cluster is not recognized by the clustering algorithm, and gets absorbed in a large cluster, especially when the distributions have a smooth transition into one another. Since we are dealing with a ~10:1 optimal to marginal performance, a class imbalance definitely exists. A warm start could help combat this problem. On top of that, the clusters can be weighted with the 10:1 ratio to hopefully avoid the problems posed.

### 5.3.3 Discussion and applications for predictive maintenance

Suppose a robust classifier has been built, and the historical switch data is reliably labeled. Then what are some of the possibilities for predictive maintenance? First of all, since the ratio of failures to non-failures is about 1:100-200, it would be meaningful for operators to receive a notification after every failure. On top of that, a condition monitoring system can be built, using an appropriate condition indicator signaling the health status of the switch. A degradation score is proposed based on the classification of the switch operations. Since there are about ten times more slow operations than failures, it would be reasonable to give a higher degradation score to failures than to slow operations. One implementation could be to add 1 to the score output for each registered slow operation, and, e.g., 5 for a failure. If the classifier measures one or more good operations, no degradation score should be added, but the degradation scores from previously detected slow and/or failure operations should not be dismissed immediately. To balance these considerations, an exponential decay is proposed for the degradation score. The strength of this decay function is determined by a set characteristic decay time $\tau_d$, but also needs to be weighted by the original height of the score, $s_0$. If not, decaying scores would rapidly converge to almost the same small value after just a few operations, as illustrated in Figure 5.14. The proposed decay function $D(\Delta n)$ is given by:

$$D(\Delta n) = s_0 \exp\left(\frac{-\Delta n}{s_0\,\tau_d}\right), \tag{5.1}$$

with $\Delta n$ representing integer numbers starting from the moment the first good operation is measured. From the historical data of the switches, an optimal threshold can be determined and tested for when to signal an alarm to the operators, indicating that maintenance is required.

There are several more possibilities to consider with the switch dataset. One example is survival analysis, e.g. with the Kaplan-Meier curve [46], where also the dependence of failures on operating conditions can be checked. Another possibility is Remaining Useful Lifetime (RUL) prediction, based on the similarity of a degradation curve to historical operational curves. The degradation score proposed in this section will probably not lend itself very well to this approach, so a new and smoother way of degradation should be devised for RUL predictions. Another workaround is to create a custom similarity measure between degradation curves, instead of relying on euclidean distance between the curves.

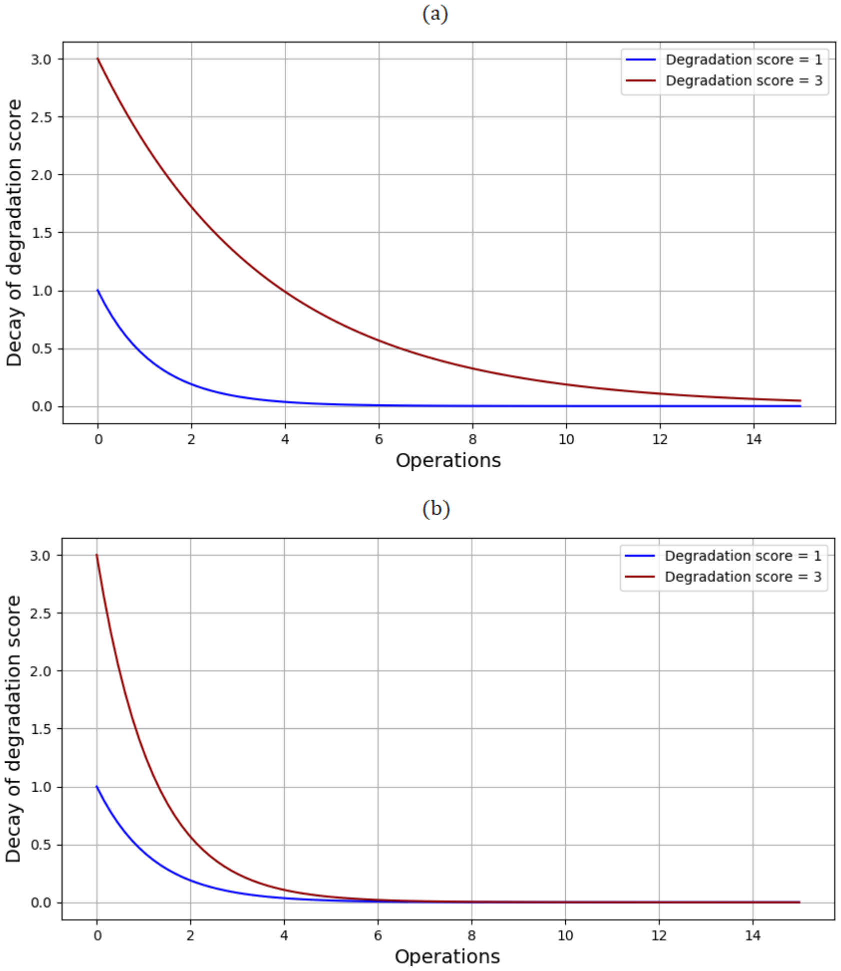

Figure 5.14: Decay of initial decaying scores of 3 and 1. (a) The decaying scores for the proposed decay in equation 5.1. (b) The decaying scores for the unadjusted decay $s_0 \exp(-\Delta n/\tau_d)$. The scores in (b) for both decays are almost the same after only 4 good operations. The higher original degradation scores for (a) linger on longer. To illustrate the principle, $\tau_D$ is set to 1.2 here, but a larger value is probably advised during actual operations.

One last application is a proposal for an automatic adjustment of the opening time programmed into the JET console. By once again using the difference between the VT504 and VT503 measurements, a threshold from 40.0004 s onwards can be implemented which signals the moment of switch opening. The program can take into account the classification of the operation, and might adapt the programmed opening time after a specified amount of opening times differed significantly from the programmed time.

# Chapter 6

# Conclusion and outlook

## 6.1 Conclusion

The worldwide effort on fusion research aims to realize a means of producing clean and safe energy for future generations. At the JET tokamak, extensive research is being performed to help accomplish this goal. By taking an in-depth look at two engineering problems at JET, the possibilities of data science as a valuable asset to fusion research were explored. As a result of the analysis, an automated approach was devised to aid researchers in managing these problems. The focus of the adopted methods was on anomaly detection in the context of predictive maintenance to predict equipment failures and avoid a possible setback for research operations. Next to the practical benefit of the models, the analysis of the available data and the discussion of the algorithms also provided insight into the underlying processes governing the problematic behaviour.

This section will provide brief overviews of the main points made in this thesis, and put forward the conclusions drawn from the results and discussions.

### 6.1.1 Turbomolecular pumps

The first use case handled the unexpected component failures for an important part in the JET vacuum system: the turbomolecular pumps. By analyzing the time series data from the pump sensor readings, it was established that Remaining Useful Lifetime prediction was not feasible with the available data. Two semi-supervised anomaly detection models were proposed and tested on suitable parts of the dataset.

The first model was based on dimensionality reduction with principal component analysis and multivariate Gaussian modeling, the second model used an auto-encoder neural network. Both models were trained on healthy data, and used an appropriate error measure to indicate the (dis)similarity of new data samples to the healthy data. For the first model, this resulted in an intuitive look at the evolution of the sensor data by means of the Mahalanobis distance and a visual inspection of the data distribution throughout time. The anomaly detection results identified a trend of anomalous behaviour leading up to a failure, but still flagged a number of probable false positives. The results from the auto-encoder model also provided a general trend towards failure, but with a clearer distinction between healthy and anomalous regions and with less false positives produced. Both models recognize the moment of failure well, with a strong anomalous peak representing behaviour that has not been encountered before. Since the auto-encoder model produced better results and was implemented with more general features, it is chosen as the main model for the turbomolecular pumps.

After consideration of the results, it can be concluded that the increase in anomalies in later life stages of a pump does not exclusively signal a process of degradation; it is also an indication of conditions in which it becomes dangerous for the pump to operate in. When these 'danger zones' are combined with the subtle degradation of the pump, chances of failure increase. The regions of high anomaly scores appear to be linked with periods of experiments being performed, and it is assumed that the extensive stress of multiple experiments being performed in a small period of time most probably is one of the major contributions to a pump failure.

Some suggestions to mitigate the effects of pulses (experiments) were provided, since pulses can be seen as a form of false positives to the anomaly detection algorithms. They manifest as a random sudden change in sensor measurements, and it is more informative to focus on the aftermath of the pulses and see how they have a lasting influence on the anomaly results of the model (and by extension, the condition of the pump). With a practical implementation for fusion operations in mind, managing these pulses is desirable and will lead to an improvement in the correctness of the results acquired.

In conclusion, the devised anomaly detection model provides indications of a deviation from healthy behaviour, and signals to operators that a pump is working under suboptimal conditions. The model is kept as general as possible to allow extensions to other situations with similar available data, in accordance with the proposed research goals. Information of the kind provided by the models can benefit operations by sending warning signals ahead of time, and possibly avoiding a costly and inconvenient failure of the pumps.

### 6.1.2 S1 current switches

For the second use case, another significant component of the JET tokamak was discussed: the S1 current switch, which interrupts a high electrical current from the poloidal flywheel generator converter to the central solenoid. As a switch ages, unintended slow and/or failed current interruptions occur. Two signals are important to assessing the operation of a switch: the VT503 and VT504 voltage signals. By comparing these signals for different kinds of switch operations, a classifier was built that can distinguish between good, slow and failed operations, with possible extensions to more fine-tuned categories. The classifier used a simple form of feature engineering: by subtracting the two voltage signals, a clearer distinction between classes emerged, almost independent from the operating conditions (in contrast to the original signals).

A semi-supervised approach to building a more precise classifier was proposed. By using the small available dataset of labeled switch operations, all samples from the large unlabeled historical switch operation dataset can in theory be reliably labeled. Clusters obtained with an unsupervised clustering algorithm for all thousands of samples could use the small labeled dataset for identification, and with this new large and labeled dataset, a better classifier can be built. Some problems with the clustering mechanisms can be addressed by providing a warm start to the unsupervised algorithms based on the small labeled dataset. The samples on the verge of two clusters can be assigned for manual labeling by an expert, creating a robust labeled dataset with little human effort. Finally, a proposal for a rudimentary predictive maintenance strategy was discussed based on a condition monitoring system that uses this robust classifier.

By carefully examining the switch dataset provided, the relations and distinctions between the different switch operations could be established. Simple feature engineering and visual representation of the VT503/504 signals paved the way to a classifier that will probably lie at the center of any predictive maintenance strategy. Even without an optimized predictive maintenance program, the automatic classification of switch behaviour will provide researchers with valuable information during operations, which might avoid failed experiments and effectively reduce downtime and costs.

## 6.2 Recommendations

The models proposed here are not necessarily complete. To further improve them, several steps can be taken. For the turbomolecular pump models, more data could provide a means to robustly test the model and apply statistically significant performance measures. Besides actually gathering this data, simulated data might also offer an intermediate solution. More data would also pave the way to a data-driven fine-tuning of certain important model parameters, like the anomaly threshold. It could also open up possibilities with regards to 'true' predictive maintenance, with potential applications in Remaining Useful Lifetime prediction.

Another data-related improvement for the anomaly detection models would be an adjustment of the way data is gathered. The coarse integer measurements of the current temperature sensors could be replaced by more precise sensors, and vibrational sensors could be added to the arsenal of pump monitoring. Vibrational sensors especially are suspected to hold great potential for the condition monitoring of the pumps, with many possible applications and analysis techniques available. Even root-cause-analysis might be possible, since for many standard mechanical components, like the bearings used in the turbomolecular pumps, lots of experience has by now been gathered on analyzing vibrational signals and relating specific patterns to common failures.

One major drawback of the proposed anomaly detection models is the need for a pool of healthy data for every pump installation. A model that could learn abstractions of healthy data could suffice with a healthy dataset that is gathered once, and could then be applied without restrictions to any pump operation, supposing that all normal samples are identically distributed from the same underlying distribution, or at least from very similar distributions.

As for the S1 current switches, a good first improvement of the model would be to train the initial classifier on a dataset with a balanced representation of the classes. Also, a larger dataset would be advised to obtain more statistically relevant results. Since the dataset already contains multiple slow and failure operations[1], only samples representing good operations need to be added, which are easy to find and recognize. Another possibility would be to further subdivide the classes into more nuanced categories. For example, failures could be further divided into 'regular failures' and restrikes, or different gradations of slow operations can be implemented.

Due to time constraints, the proposed clustering techniques and the subsequent rudimentary predictive maintenance strategy could not be implemented, and therefore remain mere suggestions. The only way to know for sure if these approaches would work, is to implement them and do a careful analysis of the results. This might be an opportunity for future work.

## 6.3 Final words and outlook

This work was carried out in the spirit of investigating the possibilities of data science in fusion operations. By tackling two interesting use cases head-on, an illustration of the potential of data science as a powerful framework was provided. It was shown that, even with limited data, results can be obtained that might help operators to evaluate the condition of fusion equipment in the future, therefore adding to the streamlining of fusion research in general. Hopefully this modest exploration will spark interest in other fusion enthusiasts to try and improve the methods proposed here, or even better, to extend the framework of machine learning to new exciting problems and help clear the path to achieve global fusion energy for all.

[1] As was shown in chapter 5, the classification of failures vs. non-failures was the easiest to accomplish. Therefore, not many failures are required. The real challenge lies in classifying good vs. slow behaviour.

# Appendix

## Turbomolecular pump data

### Frequency, bearing temperature, body temperature, current, power and voltage (sampled every 30s)

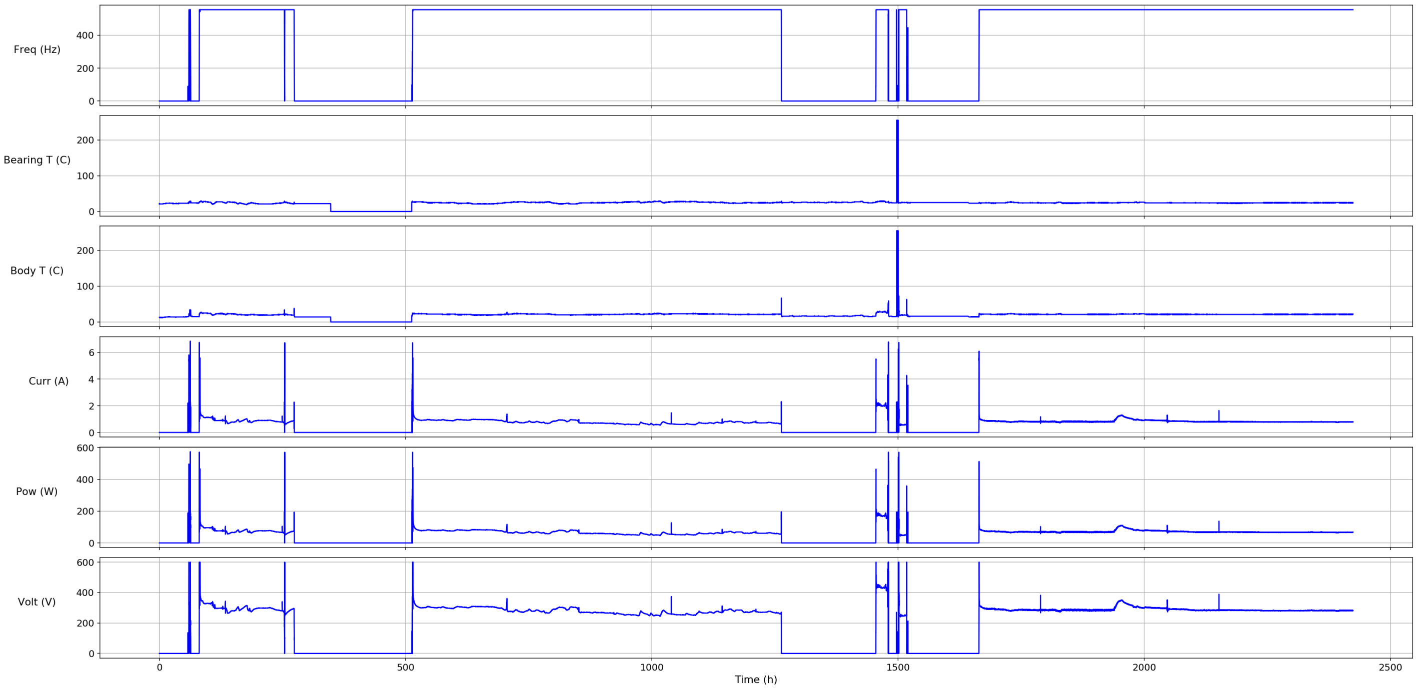

Figure 6.1: TT01 from December 1, 2017 and the following 100 days. The second failure is visible at about 1263 h.

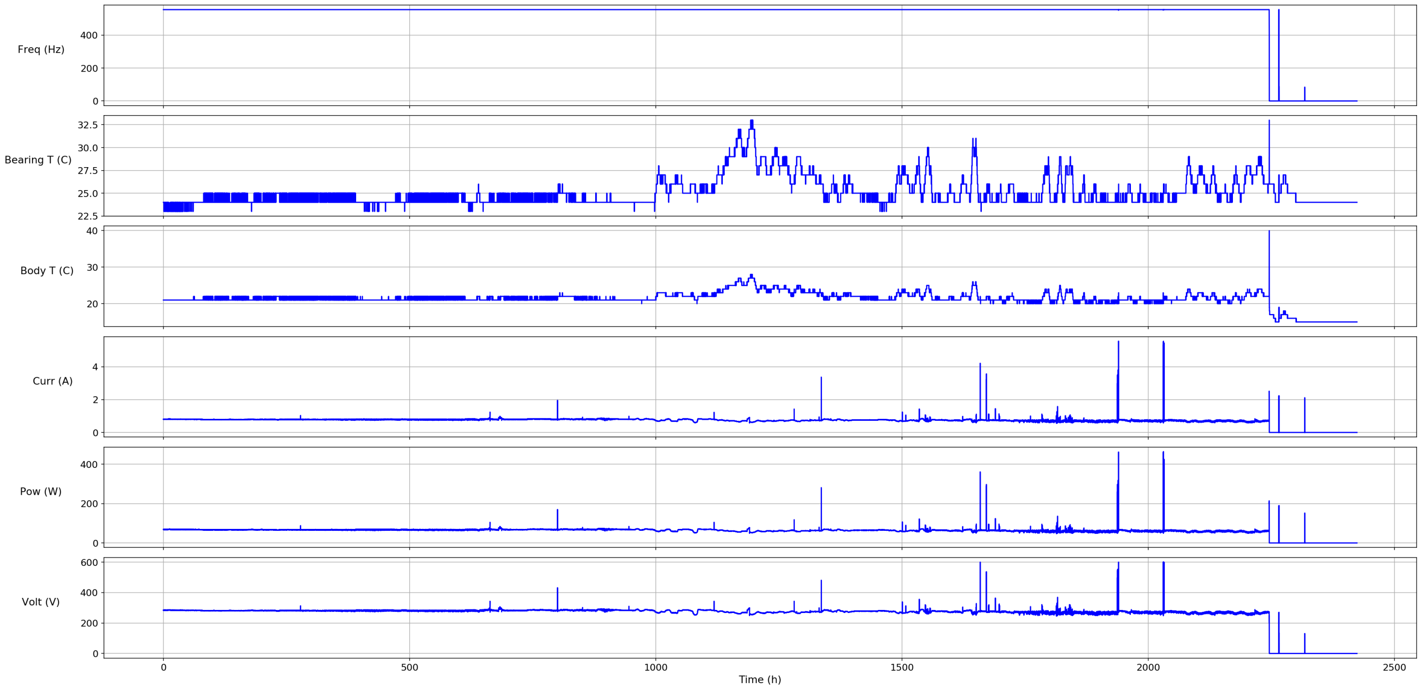

Figure 6.2: TT01 from March 1, 2018 and the following 100 days. The third failure is visible at about 2246 h.

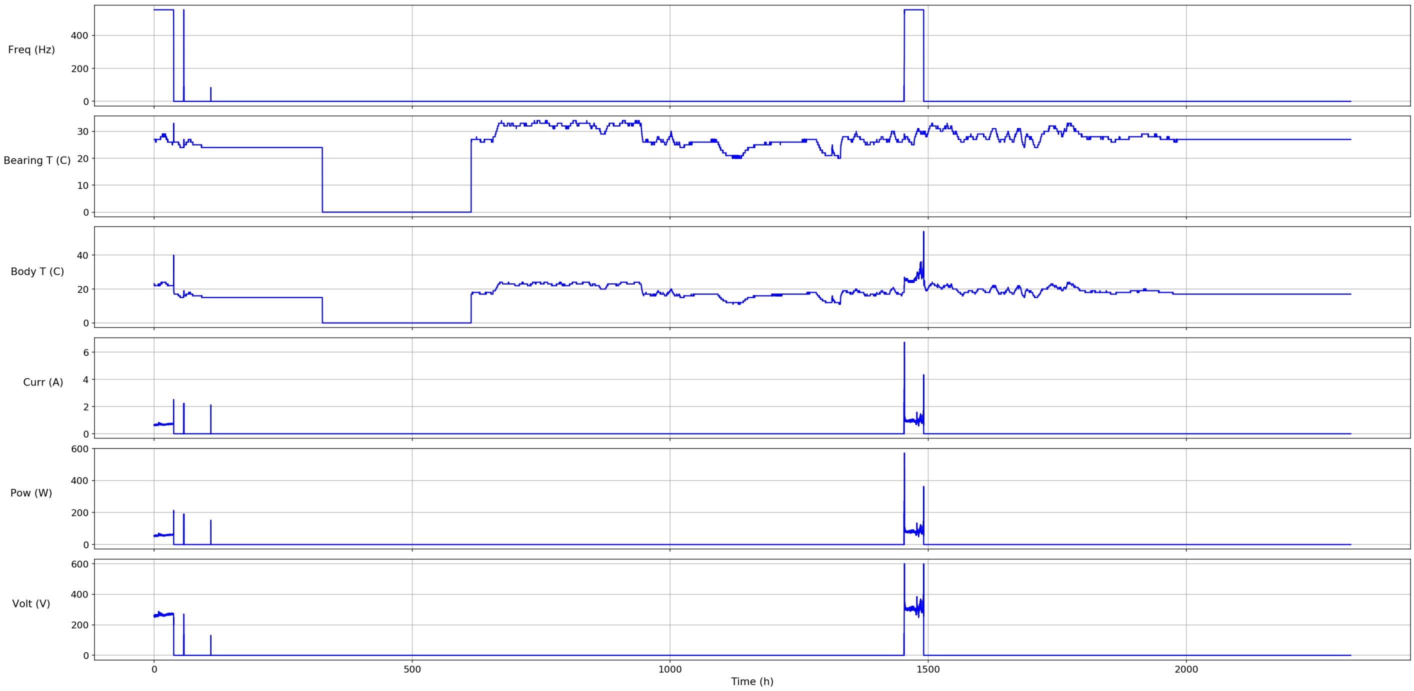

Figure 6.3: TT01 from June 1, 2018 and the following 100 days. The fourth failure is visible at about 1491 h.

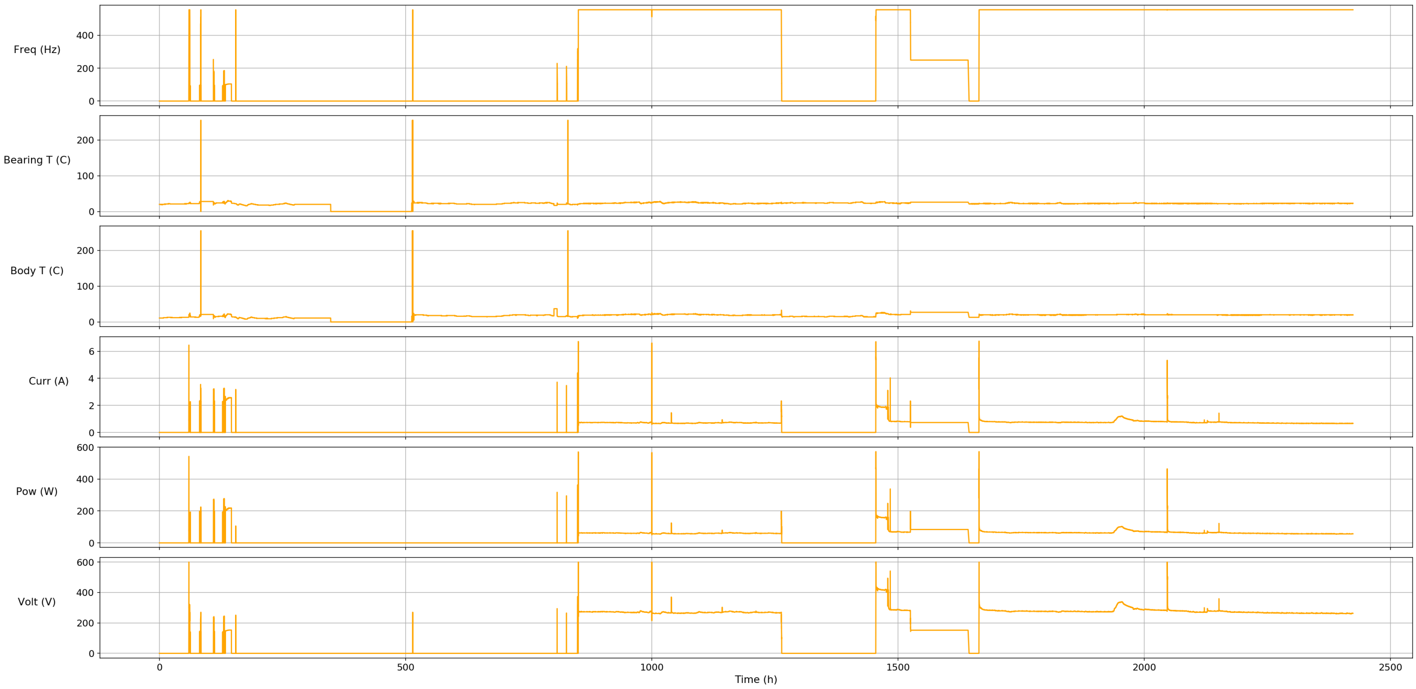

Figure 6.4: TT02 from December 1, 2017 and the following 100 days. The first failure is visible at about 62 h.

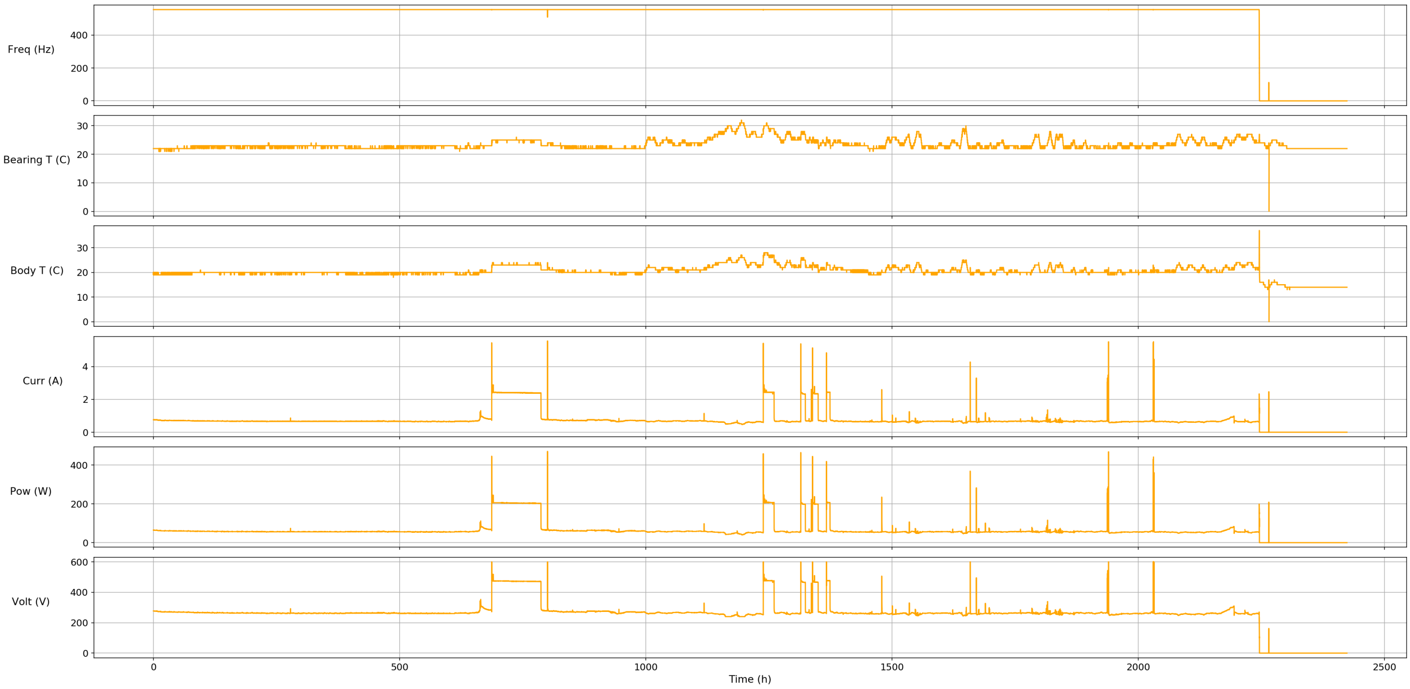

Figure 6.5: TT02 from March 1, 2018 and the following 100 days.

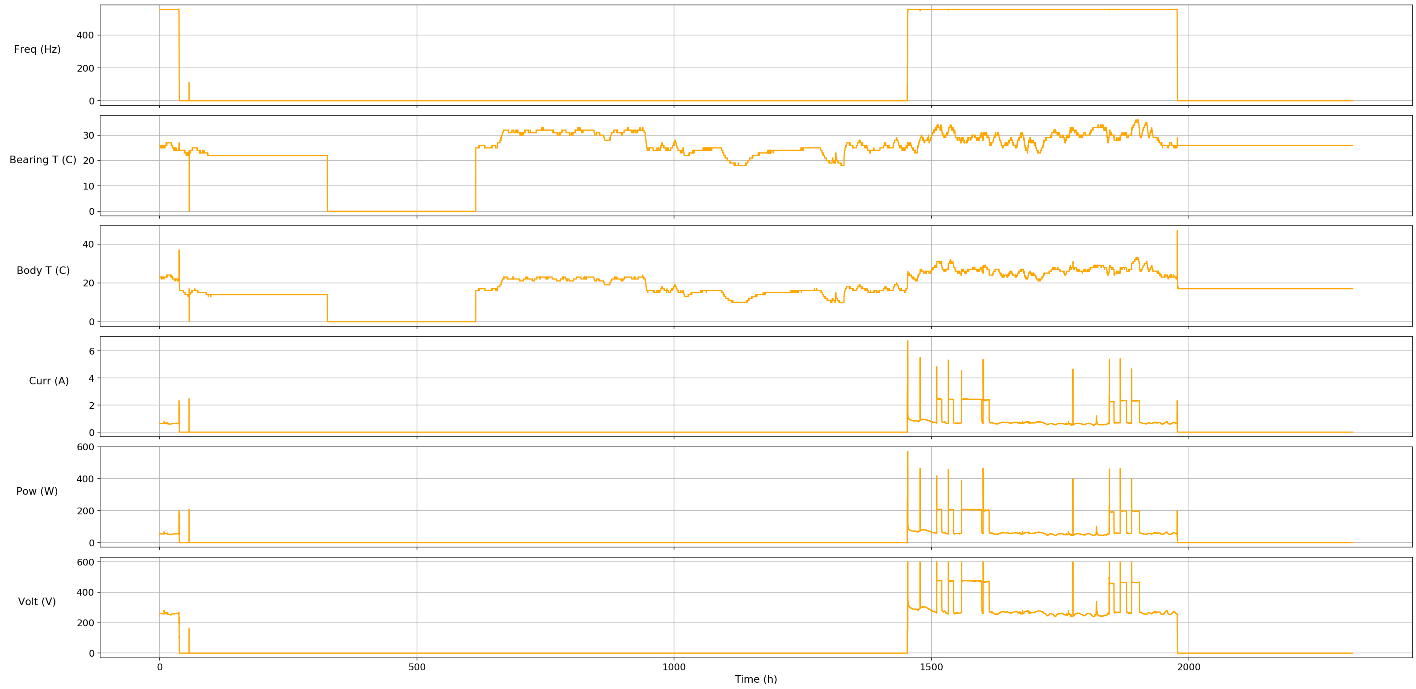

Figure 6.6: TT02 from June 1, 2018 and the following 100 days. The fifth failure is visible at about 1978 h.

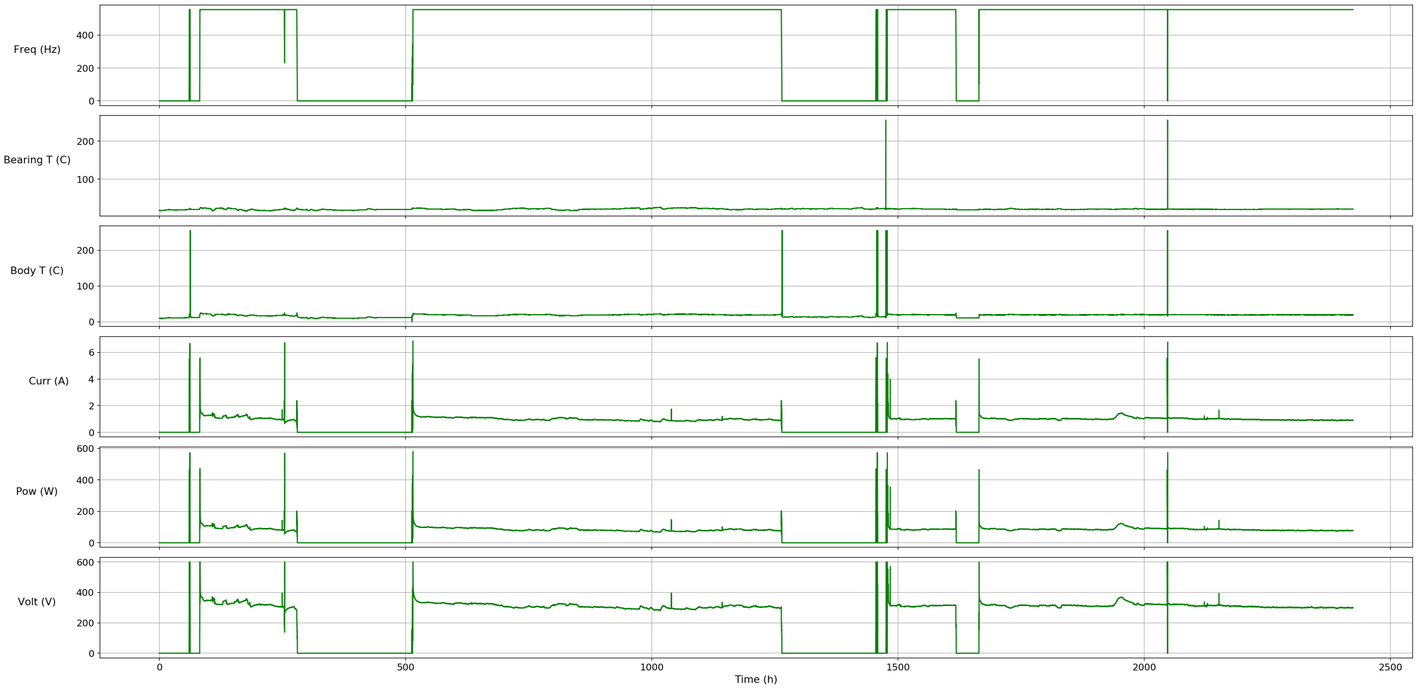

Figure 6.7: TT03 from December 1, 2017 and the following 100 days.

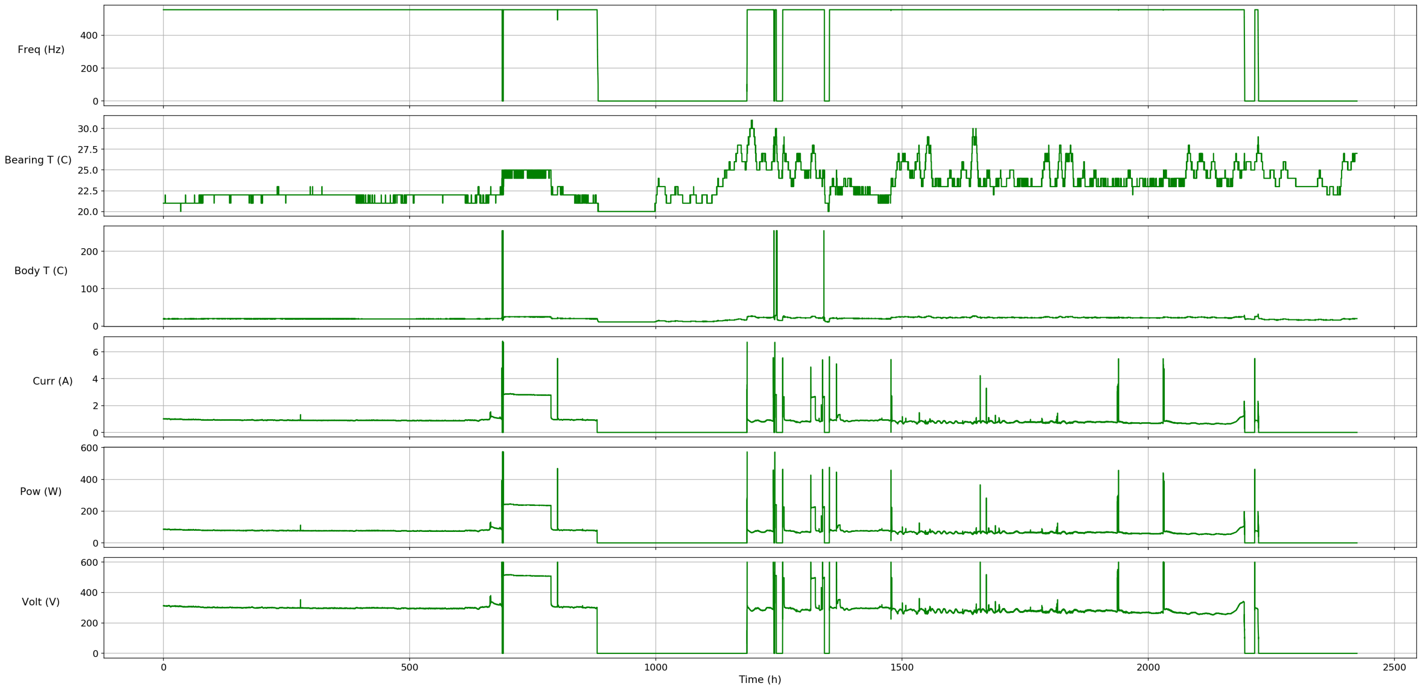

Figure 6.8: TT03 from March 1, 2018 and the following 100 days.

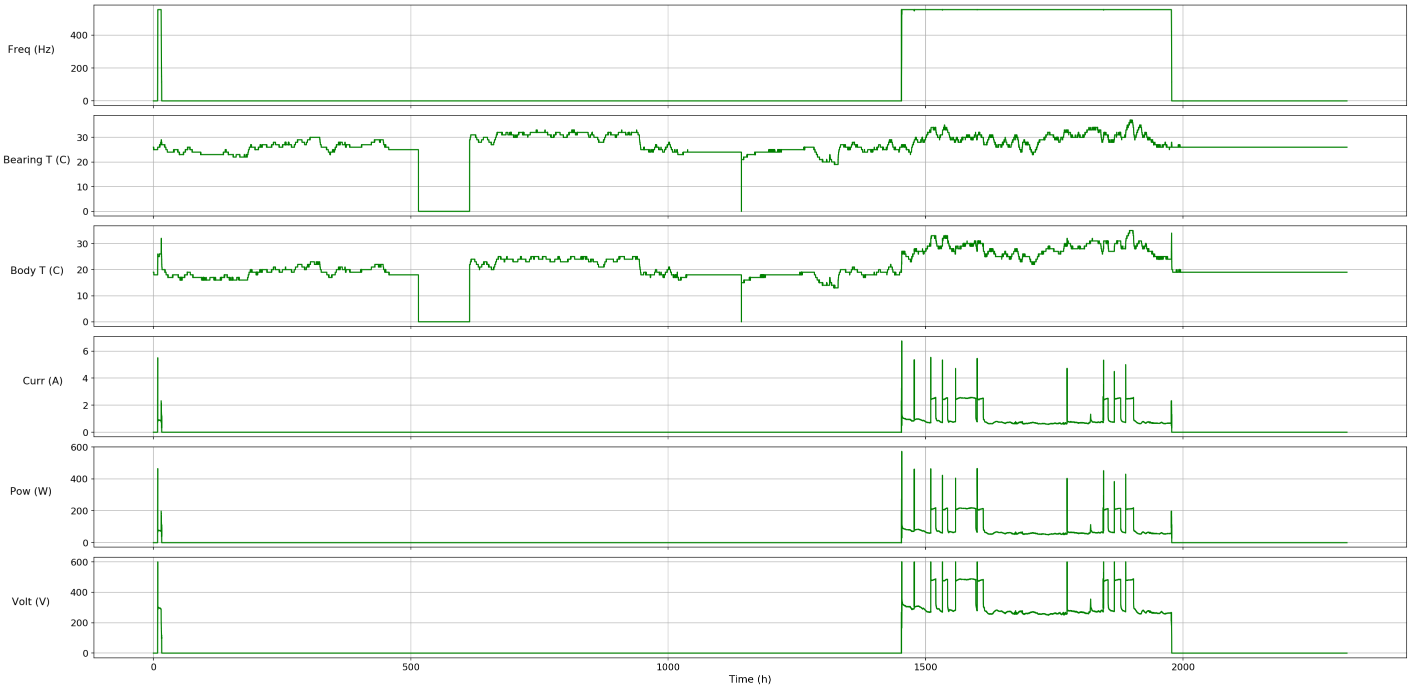

Figure 6.9: TT03 from June 1, 2018 and the following 100 days.

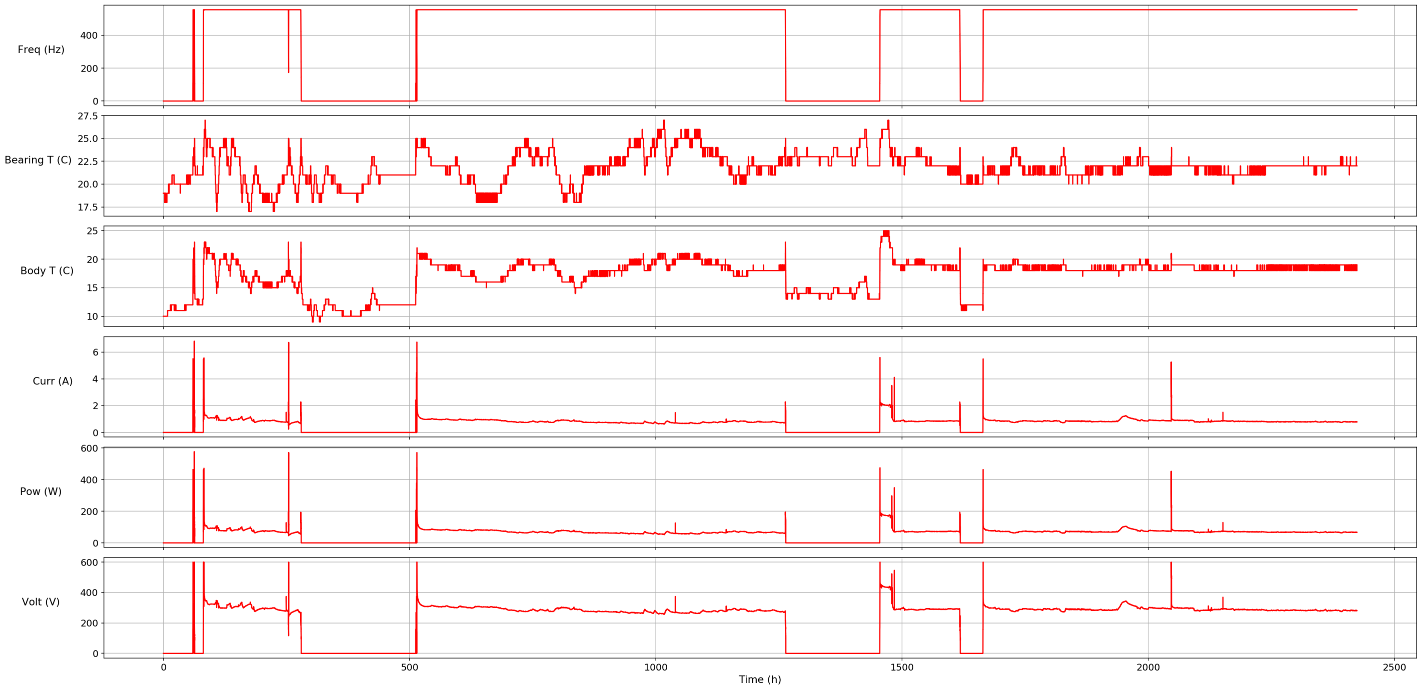

Figure 6.10: TT04 from December 1, 2017 and the following 100 days.

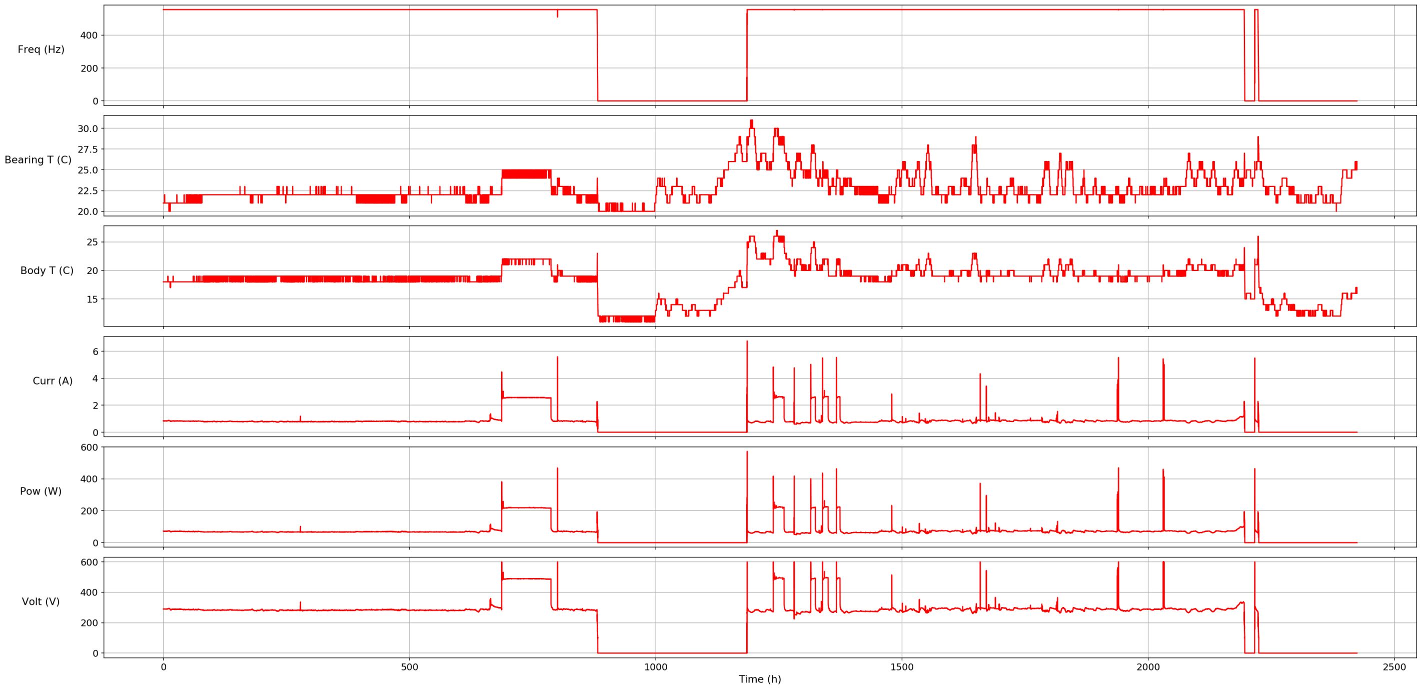

Figure 6.11: TT04 from March 1, 2018 and the following 100 days.

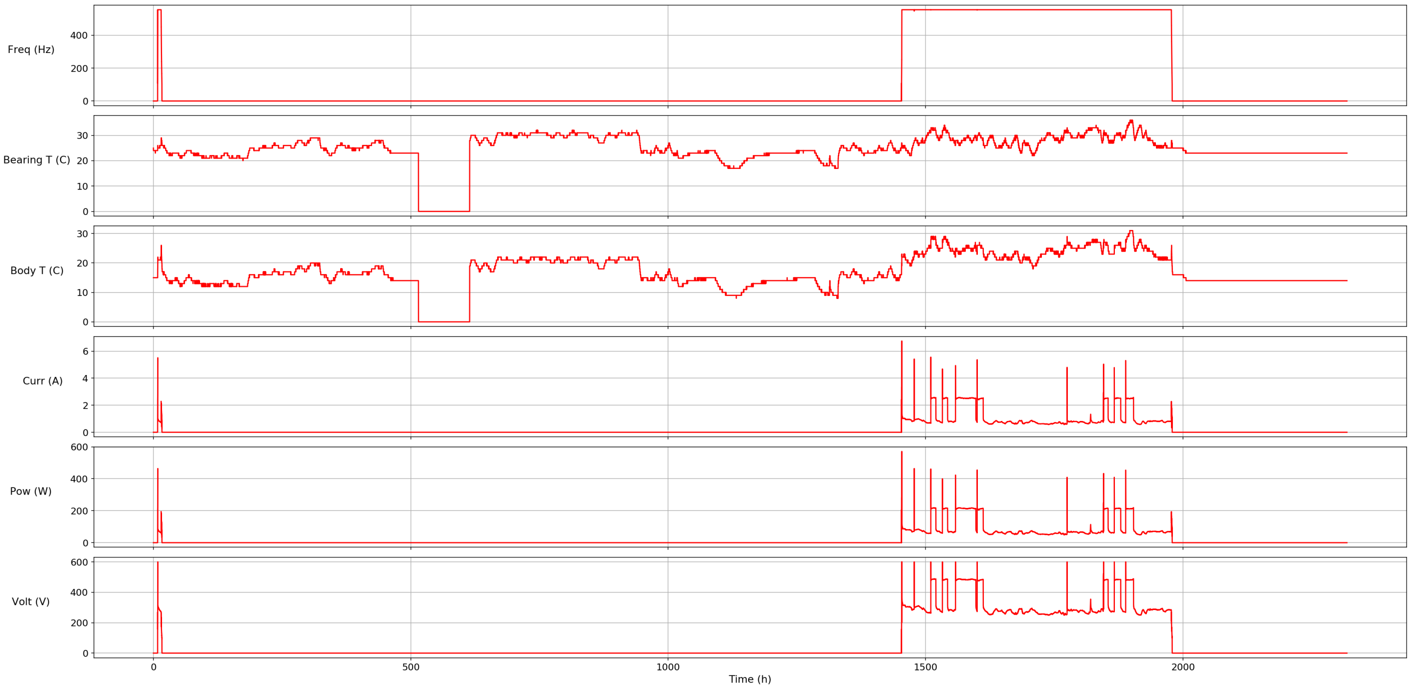

Figure 6.12: TT04 from June 1, 2018 and the following 100 days.

## Pressure sensor measurements (sampled every 5 s)

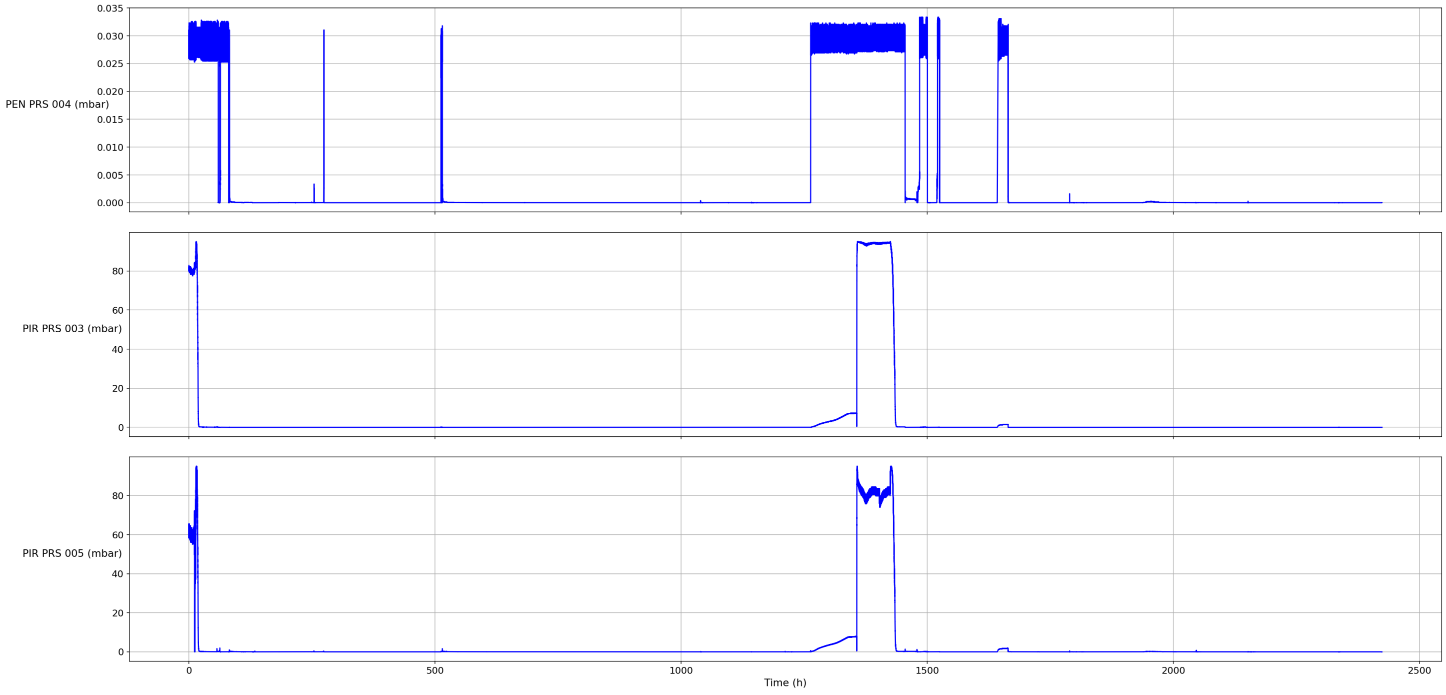

Figure 6.13: TT01 from December 1, 2017 and the following 100 days.

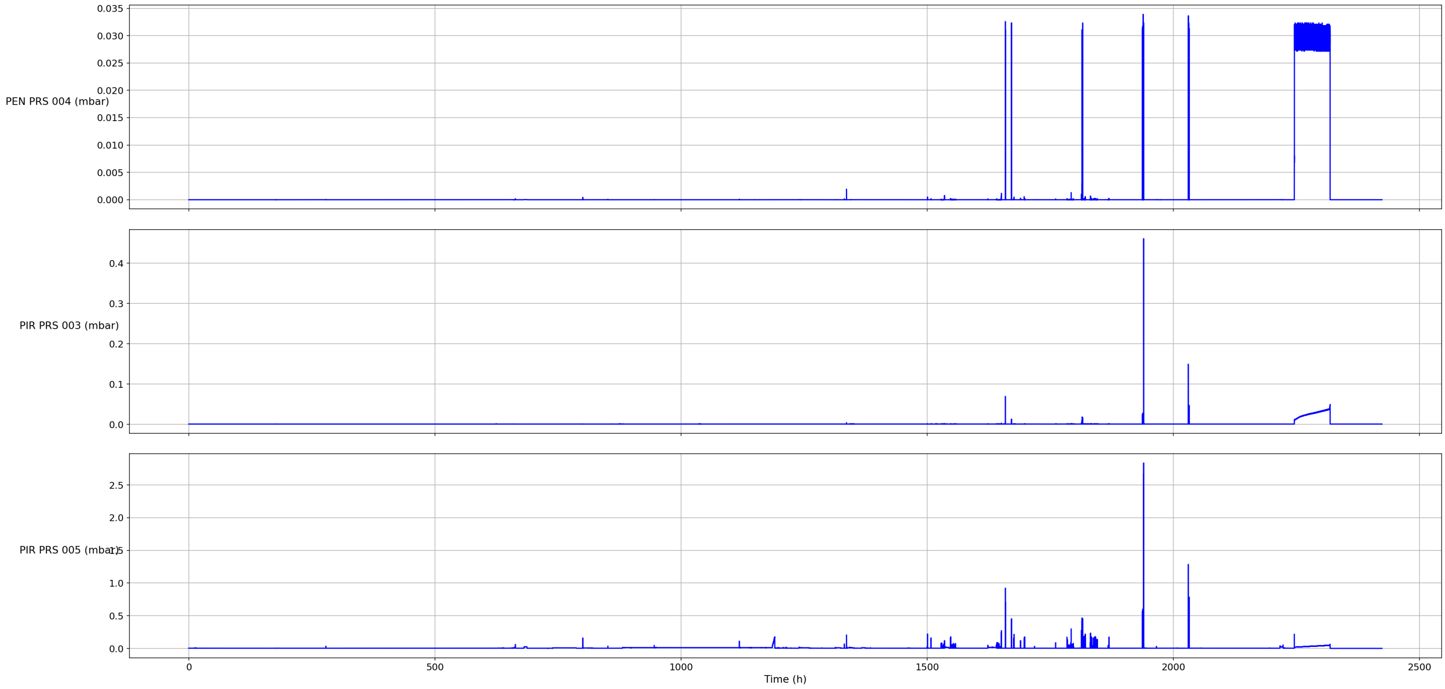

Figure 6.14: TT01 from March 1, 2018 and the following 100 days.

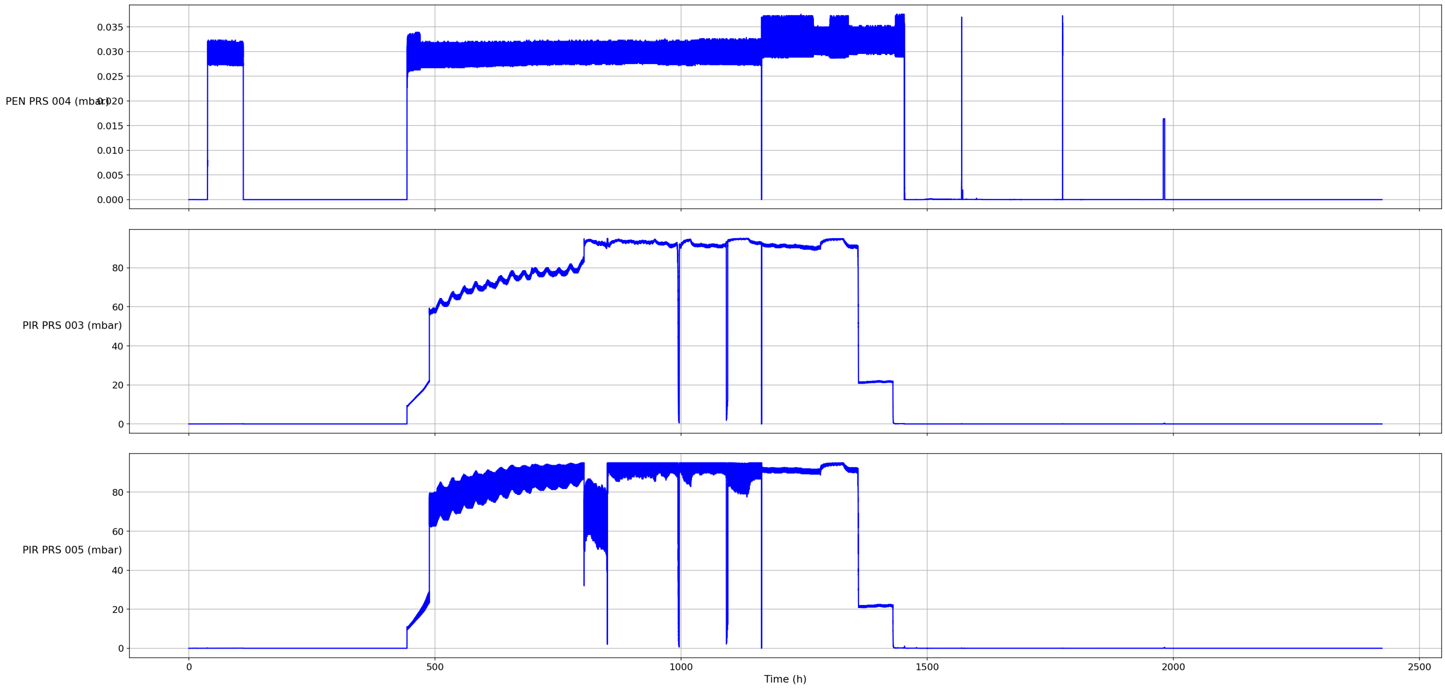

Figure 6.15: TT01 from June 1, 2018 and the following 100 days.

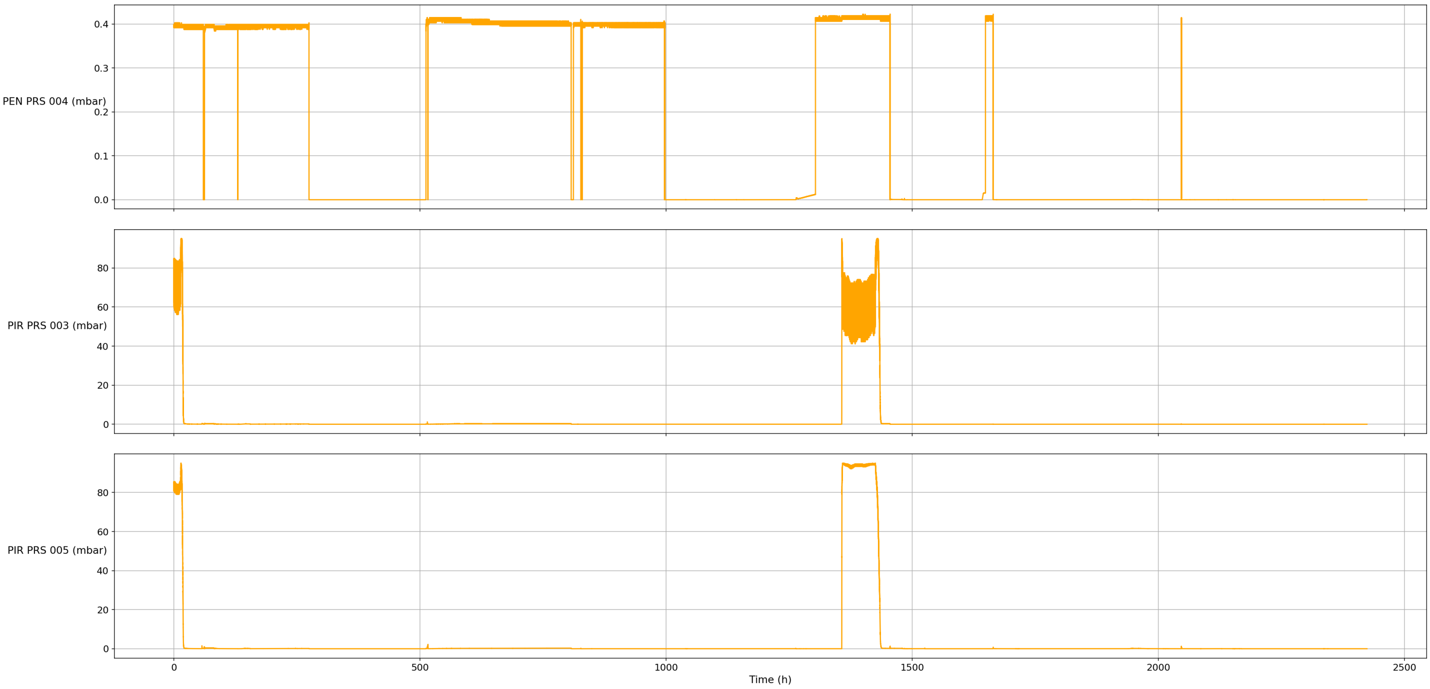

Figure 6.16: TT02 from December 1, 2017 and the following 100 days.

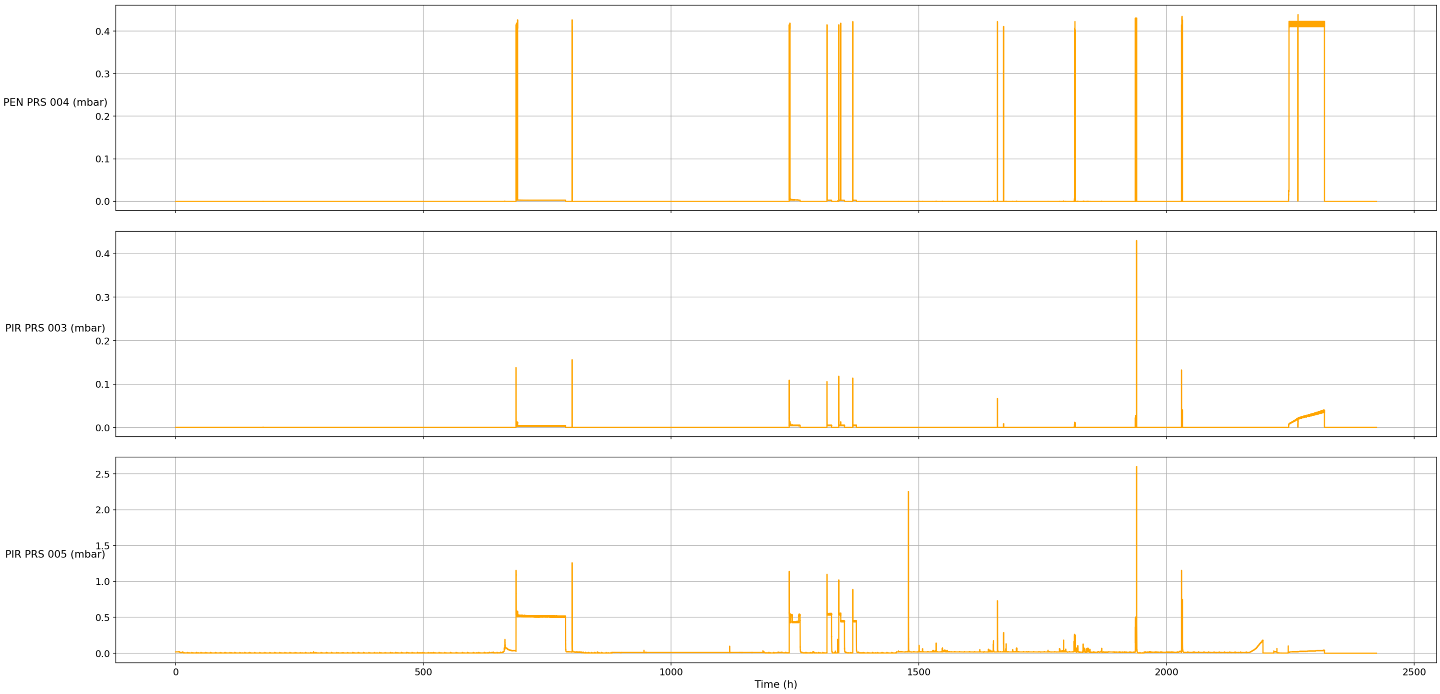

Figure 6.17: TT02 from March 1, 2018 and the following 100 days.

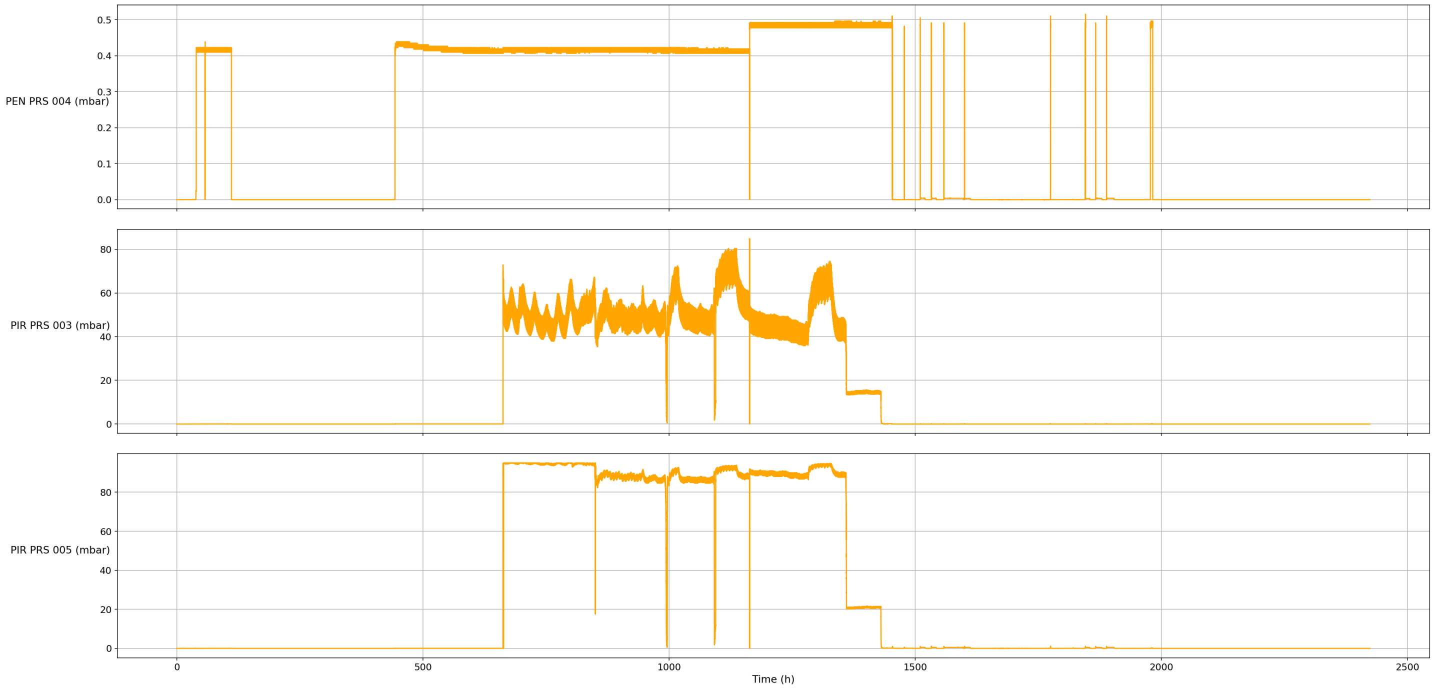

Figure 6.18: TT02 from June 1, 2018 and the following 100 days.

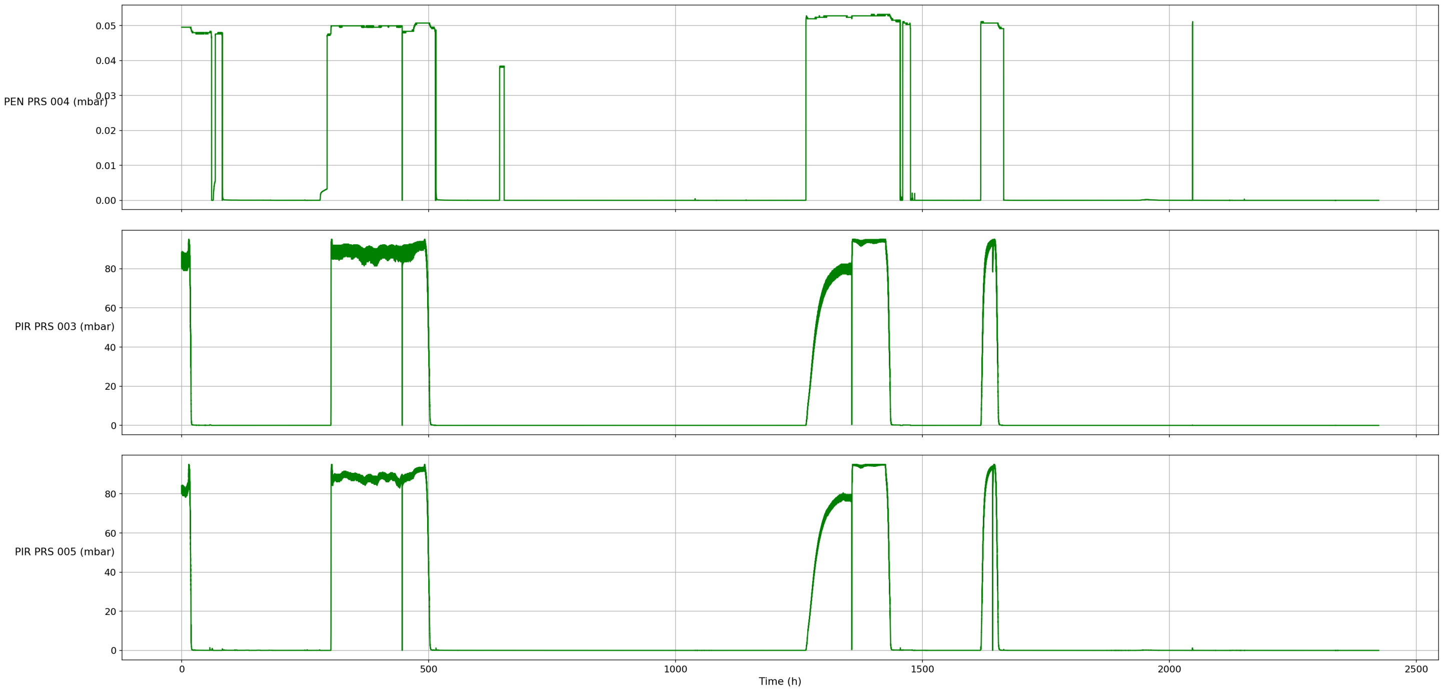

Figure 6.19: TT03 from December 1, 2017 and the following 100 days.

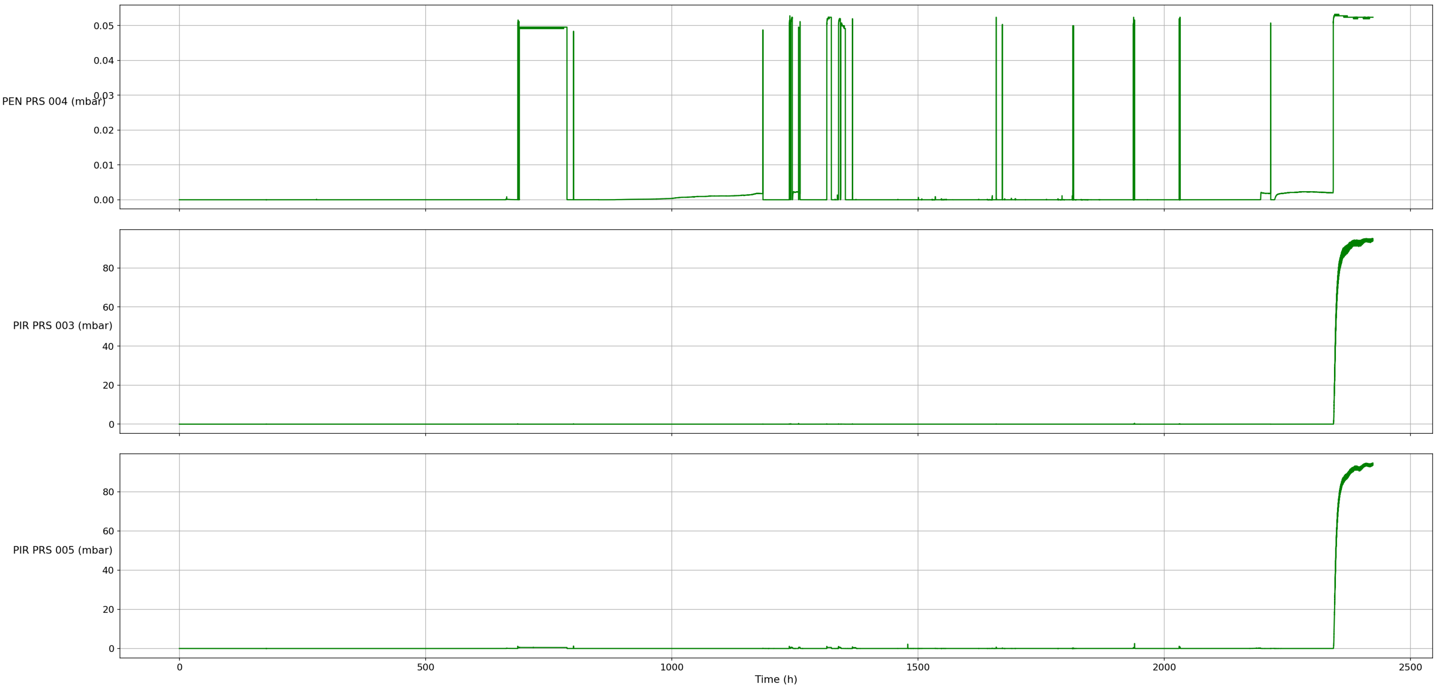

Figure 6.20: TT03 from March 1, 2018 and the following 100 days.

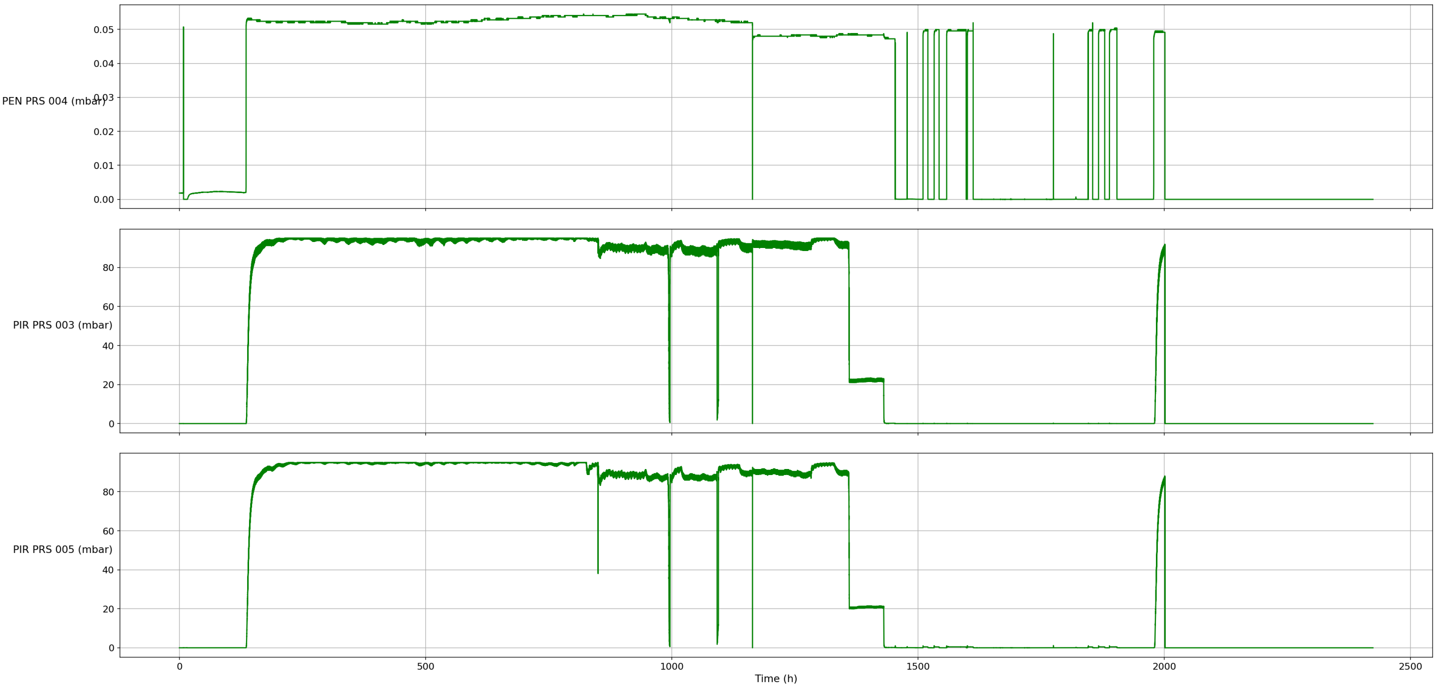

Figure 6.21: TT03 from June 1, 2018 and the following 100 days.

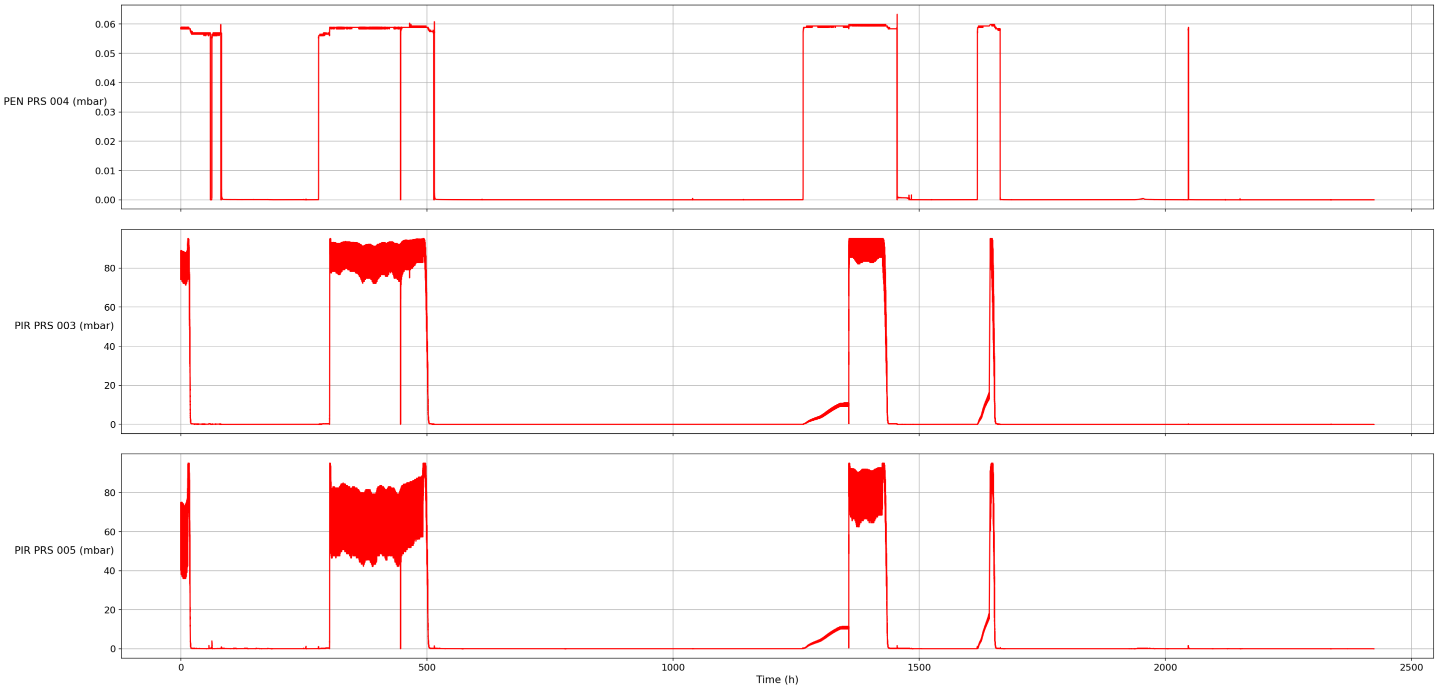

Figure 6.22: TT04 from December 1, 2017 and the following 100 days.

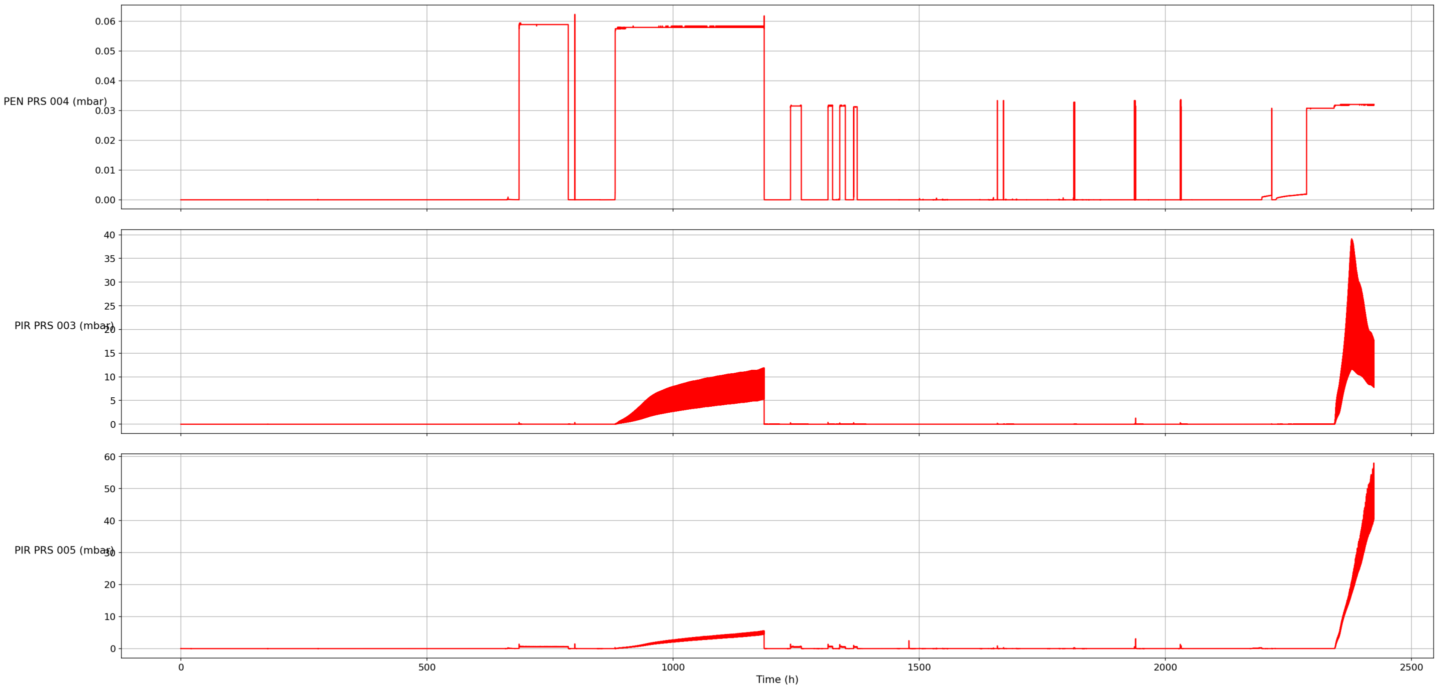

Figure 6.23: TT04 from March 1, 2018 and the following 100 days.

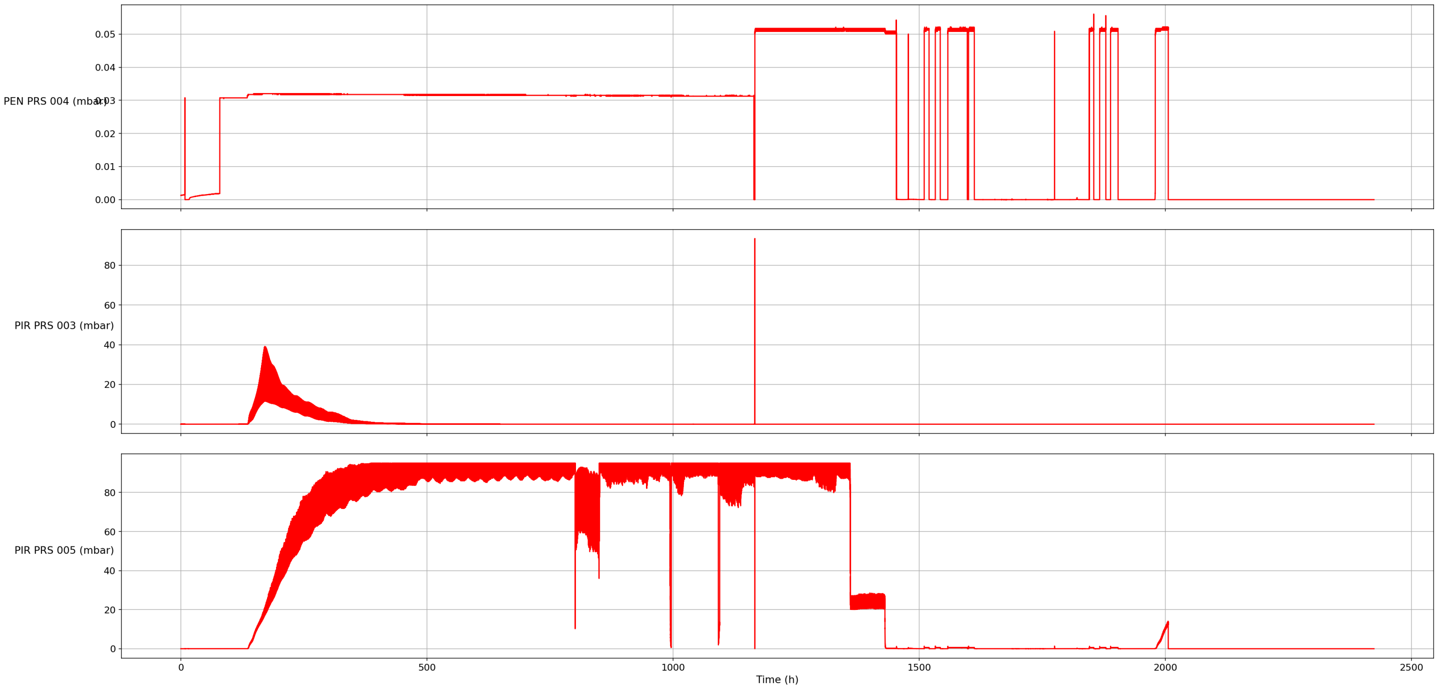

Figure 6.24: TT04 from June 1, 2018 and the following 100 days.

## Auto-encoder anomaly detection results for the second failure.

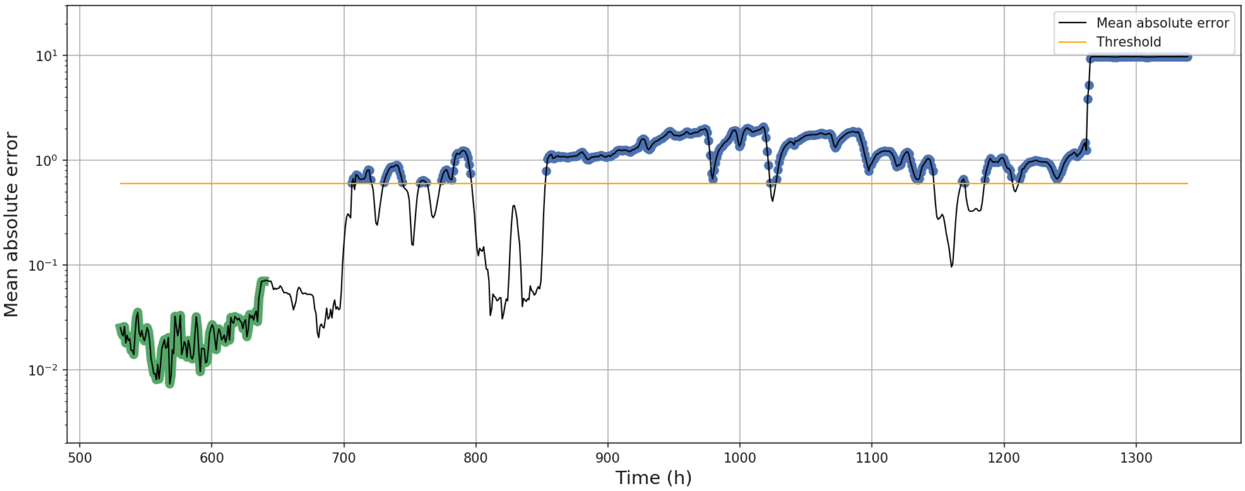

Figure 6.25: Anomaly detection by means of an auto-encoder neural network. The mean absolute error is taken as the anomaly measure for the turbomolecular pump at position TT01 from Dec. 23, 2018 until the failure on Jan. 22, 2018. The green samples represent training samples, the blue dots are flagged as anomalies. The threshold is shown as an orange line. It is worth noting that the prior period of healthy signals is shorter than for the third failure. To optimally train the neural network, the period of healthy data was chosen up to a moment close to the intuitive change in data patterns (∼700 h).

# Nederlandse samenvatting

*This summary is in Dutch.*

Wereldwijd onderzoek naar kernfusie is gericht op het realiseren van een schone en veilige energiebron voor toekomstige generaties. Uitgebreid onderzoek wordt momenteel gevoerd aan de JET-tokamak met oog op het bereiken van dit ambitieuze doel. Onderdelen aan de JET-tokamak gaan soms onverwacht stuk, net zoals bij elke complexe machine met veel componenten. In dit werk worden storingen bij twee JET-onderdelen behandeld met als doel het predictief onderhoud van deze componenten door gebruik te maken van anomaliedetectie en andere technieken uit machinaal leren.

Het eerste geval betreft verschillende mislukte werkingen bij de turbomoleculaire pompen in het JET vacuümsysteem. Een oplossing voor het tijdig opsporen van ongezond gedrag wordt voorgesteld met behulp van semi-gecontroleerde anomaliedetectie op basis van tijdreeksgegevens van sensorsignalen. Afwijkingen van normaal gedrag worden gesignaleerd wanneer binnenkomende sensorgegevens als te verschillend worden beschouwd van een verzameling gezonde trainingsdata. Een eerste model dat gebruik maakt van hoofdcomponentenanalyse en multivariate Gaussiaanse modellering wordt ontwikkeld waarbij de Mahalanobis afstand tot het centrum van de gezonde distributie gebruikt wordt als anomaliescore. De anomaliescores worden vergeleken met een drempelwaarde, en metingen met scores boven deze drempelwaarde worden gemarkeerd. Een soortgelijke benadering wordt gebruikt bij een tweede model, gebaseerd op auto-encoder neurale netwerken. In plaats van de Mahalanobis afstand wordt de reconstructiefout van het auto-encoder neurale netwerk gebruikt en wordt een glijdend tijdvenster ingezet om tijdscorrelaties mee te nemen in het model. Het netwerk wordt opnieuw alleen getraind op een verzameling van gezonde data, dus de reconstructiefouten zullen groter zijn voor metingen die afwijken van dit gedrag. Er wordt opnieuw een geschikte drempelwaarde ingesteld en als de reconstructiefout voor een tijdvenster boven deze drempel valt, wordt ze gemarkeerd. Beide modellen tonen een toename van de anomaliescores die leidt tot een sterke anomalie-piek die het moment van een gefaalde werking van een turbomoleculaire pomp voorstelt. Het auto-encoder neurale netwerk markeert echter minder vals-positieven en toont een duidelijker onderscheid en een vlottere overgang tussen gezond en abnormaal gedrag. Een bespreking van de resultaten en suggesties voor een implementatie in kernfusie-onderzoek worden gegeven, samen met mogelijke uitbreidingen van het model.

Het tweede scenario gaat over de S1 stroomschakelaar. Naarmate een schakelaar ouder wordt, komen fouten en te trage operaties steeds vaker voor. Op basis van de analyse van

twee spanningssignalen doorheen de tijd wordt een logistisch regressiemodel getraind om een onderscheid te maken tussen goede, langzame en mislukte operaties.

De resultaten van de classificator zijn veelbelovend, met F1-scores boven 0,9 voor alle categorieën. Het model wordt wel slechts getraind en getest op een kleine en ongebalanceerde dataset. Een semi-gecontroleerde clusteranalyse wordt voorgesteld om een meer robuuste classificator te bouwen door de kleine gelabelde dataset te combineren met de rest van de niet-gelabelde werkingen. Deze aanpak vereist weinig menselijke inspanning, terwijl er gebruik gemaakt wordt van alle beschikbare datasamples voor schakelaars. Ten slotte wordt een rudimentaire strategie voor predictief onderhoud voorgesteld met behulp van de ontwikkelde classificator samen met en een degradatiescoresysteem. De resultaten van beide scenario's tonen potentieel voor het gebruik van machinaal leren in kernfusie en dienen als uitnodiging om de voordelen van een datagestuurde aanpak voor de oplossing van problemen in machineonderhoud – en fusieonderzoek in het algemeen – verder te onderzoeken.

# Science popularization

*The following article is in Dutch. It is part of the effort of the Faculty of Sciences at Ghent University to communicate scientific research to the general public.*

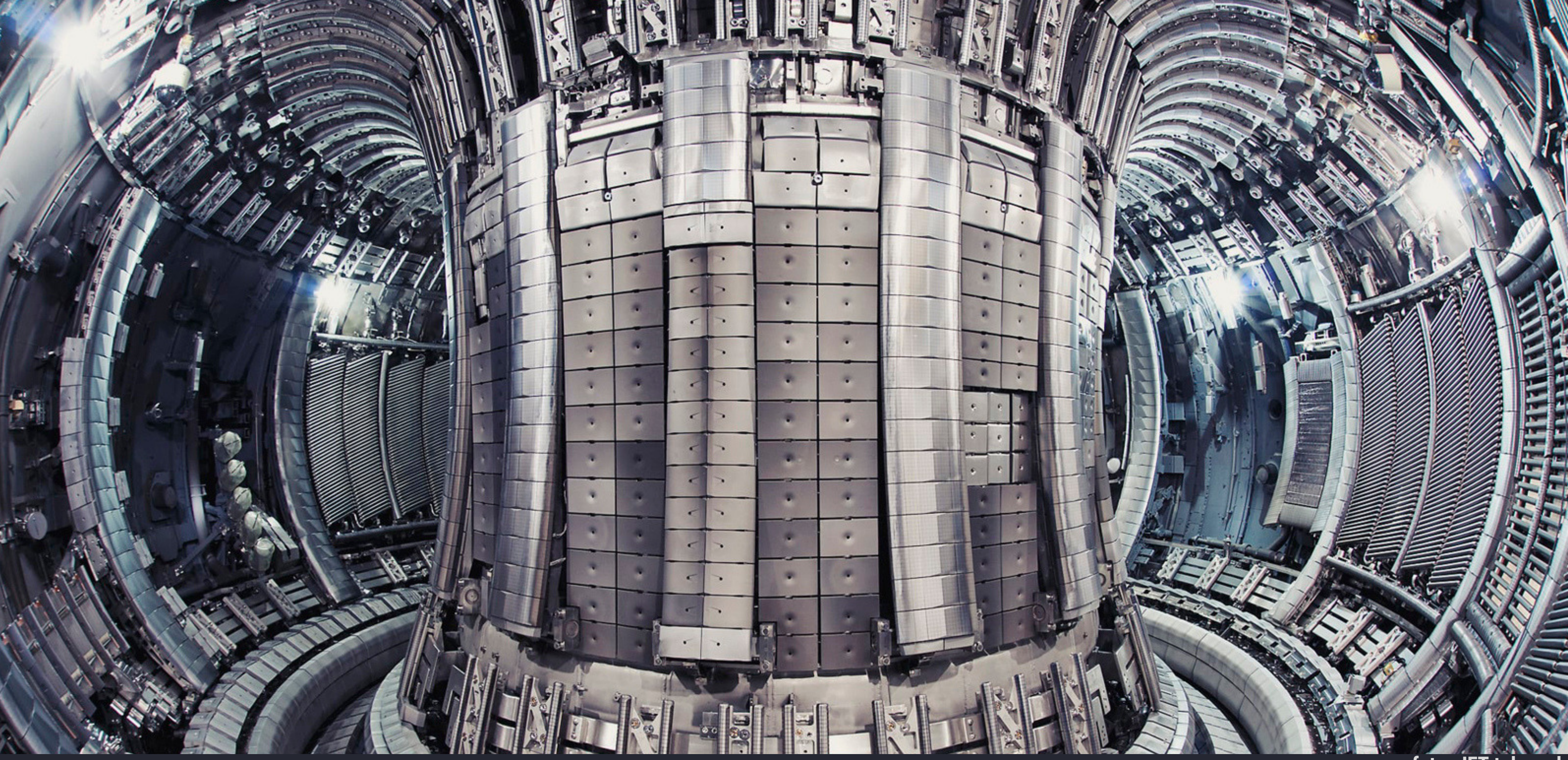

foto: JET tokamak

# OP WEG NAAR EEN BETROUWBARE KERNFUSIEREACTOR MET BEHULP VAN ARTIFICIËLE INTELLIGENTIE

ANDRIES ROSSEAU, 15 juni 2019.

**Hoe kunnen we onze snel groeiende wereld voorzien van schone en veilige energie? Het is een vraag die vandaag meer dan ooit relevant is. Kernfusiewetenschappers proberen een antwoord te bieden door de energieopwekkende processen uit de zon na te bootsen in een fusiereactor op aarde. In een ideale wereld zijn alle element in zo'n complexe machine perfect op elkaar afgesteld en doet elk onderdeel zijn werk naar behoren. Maar hoe voorkomen we dat er in de werkelijkheid toch iets fout loopt en een duur experiment mislukt? In dit onderzoek wordt artificiële intelligentie naar voren geschoven als nieuwe bondgenoot.**

De Joint European Torus, of JET, is de grootste werkende experimentele kernfusiereactor in de wereld, gelegen in Culham, nabij Oxford. Wat daar gebeurt, kan vergeleken worden met het recreëren van wat zich afspeelt in het binnenste van een ster. Al sinds 1983 wordt bij JET fundamenteel onderzoek verricht naar kernfusie door wetenschappers uit wel 28 verschillende landen. Kernfusie is het proces waarbij twee zware waterstofkernen samengebracht worden bij heel hoge temperaturen om ze te fusioneren tot helium. Tijdens dat proces komt heel veel energie vrij: een kilogram fusiebrandstof vormt het equivalent van ongeveer zeven miljoen kilogram olie. Deze brandstof kan voor een deel eenvoudig gewonnen worden uit zeewater, en wordt voor het andere deel geproduceerd in de reactor zelf. Bovendien is kernfusie ook nog eens een nagenoeg CO2-neutrale bron van energie en worden er geen langlevende radioactieve stoffen geproduceerd, zoals wel het geval is bij traditionele kerncentrales. De reden waarom we nog geen gebruik kunnen maken van deze veelbelovende energiebron, zit in de moeilijkheid om de fusiebrandstof lang genoeg op de ongelooflijk hoge temperaturen te houden die vereist zijn voor een winstgevende operatie van de reactor. Voor kernfusie spreken we dan ook over temperaturen van meer dan 100 miljoen graden Celsius. Een manier om ervoor te zorgen dat deze extreem hete deeltjes niet in contact komen met hun omgeving, is ze op te sluiten in een sterk magnetisch veld. Dat is ook precies wat in JET gebeurt: miljoenen onderdelen werken er minutieus samen om de fusiebrandstof op te warmen in het magnetische veld, en ze daarna ook weer weg te voeren uit de reactor. Soms gaat er iets fout in dit complexe proces en laat een onderdeel in de machine het afweten op een cruciaal moment. Dit leidt niet alleen tot een mislukt experiment, maar brengt ook vaak frustraties met zich mee bij de onderzoekers. Herstellingen van deze fouten kunnen veel geld kosten en nemen vaak een lange tijd in beslag, wat het onderzoek naar kernfusie uiteindelijk vertraagt.

> "Voor kernfusie spreken we dan ook
> over temperaturen van meer
> dan 100 miljoen graden Celsius."

**Artificial intelligence to the rescue**

Om dergelijke tegenslagen in de toekomst te voorkomen werd onderzocht of slimme algoritmes al dan niet in staat zijn om afwijkende datapatronen in de fusiereactor op te sporen. Wanneer deze algoritmes vervolgens zo'n afwijkend patroon vaststellen, kunnen de onderzoekers gewaarschuwd worden zodat tijdig kan worden ingegrepen.

Twee onderdelen van de experimentele JET reactor werden tijdens dit onderzoek onder de loep genomen, en voor elk van hen werd een specifiek algoritme gebouwd. Het eerste luik van het onderzoek focust op enkele frequente mankementen bij de JET turbomoleculaire pompen. Die vormen een deel van het systeem dat instaat voor het ultrahoge vacuüm in de reactor. Zo'n hoog vacuüm is nodig om ervoor te zorgen dat het kernfusiemengsel niet vervuild raakt met andere stoffen. Bovendien zorgt het vacuümsysteem er ook voor dat na een experiment alle fusiedeeltjes netjes verwijderd worden uit de reactor.

De eerste stap in het voorkomen van nieuwe fouten in het vacuümsysteem bestaat uit het bouwen van een neuraal netwerk. Neurale netwerken zijn geavanceerde machine learning algoritmes bestaande uit kunstmatige neuronen die samen informatie kunnen verwerken, gelijkaardig aan de manier waarop ons brein dat doet. Hier werd het neuraal netwerk getraind in het zo goed mogelijk herkennen van gezonde datapatronen afkomstig van sensoren die het systeem monitoren. Wanneer het neurale netwerk vervolgens na een tijdje nieuwe binnenkomende data niet meer herkent, wordt een signaal verzonden. Het resultaat is een algoritme dat onderzoekers kan waarschuwen wanneer afwijkend gedrag zich voordoet, zodat een opknapbeurt van het systeem ingelast kan worden en het vacuümsysteem niet op een onverwacht moment stopt met werken.

**Ook kernfusie heeft al eens wat warmte nodig**

Het tweede onderzochte onderdeel maakt deel uit van het centrale systeem dat de fusiebrandstof opwarmt. Om de hoge temperaturen te bereiken die nodig zijn voor kernfusiereacties is veel energie nodig. Die energie wordt geleverd vanuit een zware ronddraaiende schijf, een zogenaamde 'vliegwiel generator'. Voor de verplaatsing van de hoge elektrische stroom van duizenden Ampères zijn stevige geleiders en schakelaars nodig die tegen een stootje kunnen. Er is één belangrijke schakelaar die het vaak zwaar te verduren krijgt tijdens dit proces: de zogenaamde S1-schakelaar. Na vele experimenten werkt deze schakelaar soms niet meer, waardoor de stroomtransitie niet correct meer kan verlopen. Om onderzoekers te waarschuwen wanneer een schakelaar aan vervanging toe is, werd een algoritme ontwikkeld dat de typische ouderdomsverschijnselen van een vermoeide schakelaar leert herkennen. Hiervoor werd gekeken naar twee belangrijke spanningssignalen. Wanneer het verschil tussen beide signalen gedurende een specifieke halve milliseconde uitgezet wordt in een 32-dimensionale ruimte, kan het algoritme een duidelijk onderscheid maken tussen gezonde data en data geproduceerd door een vermoeide schakelaar. Als resultaat kan het algoritme nu opnieuw een waarschuwingssignaal versturen wanneer te veel onnauwkeurigheden worden waargenomen, om uiteindelijk erger te voorkomen.

**Wat nu?**

Met de resultaten van beide onderdelen wil dit onderzoek het potentieel van artificiële intelligentie aantonen in de zoektocht naar een operationele kernfusiereactor. Hopelijk voelen fusie-enthousiastelingen zich aangesproken om de fascinerende methodes uit de artificiële intelligentie verder toe te passen in het onderzoek naar kernfusie, en zo mee bij te dragen aan de ontwikkeling van een schone en veilige energiebron voor de hele wereld.

# References

[1] O. Hoegh-Guldberg *et al.,* ”Impacts of 1.5$^{O}$C Global Warming on Natural and Human Systems,” in: *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty,* In Press.

[2] T. Bruckner *et al.,* ”Energy Systems,” in: *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change,* Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2014.

[3] B. Marr, ”How much data do we create every day?,” *forbes.com,* May 21, 2018. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/. [Accessed: May 13, 2019].

[4] J. Desjardins, ”What happens in an internet minute in 2017?,” *World Economic Forum,* Aug. 31, 2017. [Online]. Available: https://www.weforum.org/agenda/2017/08/what-happens-in-an-internet-minute-in-2017. [Accessed: May 13, 2019].

[5] DOMO, ”Data Never Sleeps 5.0,” *domo.com,* n.d. [Online]. Available: https://www.domo.com/learn/data-never-sleeps-5. [Accessed: May 12, 2019].

[6] K. Smith, ”Facebook statistics,” *brandwatch.com,* Jan. 5, 2019. [Online]. Available: https://www.brandwatch.com/blog/facebook-statistics/. [Accessed: May 13, 2019].

[7] F. Richter, ”Smartphones cause photography boom,” *statista.com,* Aug. 31, 2017. [Online]. Available: https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/. [Accessed: May 12, 2019].

[8] C. Frey and M. Osborne, “The future of employment: how susceptible are jobs to computerization?” Oxford University paper, Sept. 17, 2013.

[9] J. Freidberg, *Plasma Physics and Fusion Energy.* New York: Cambridge University Press, 2007.

[10] F. Chen, *Introduction to Plasma Physics.* Springer US, 1974.

[11] S. Li *et al.,* ”Optimal Tracking for a Divergent-Type Parabolic PDE System in Current Profile Control,” *Abstract and Applied Analysis,* June 11, 2014.

[12] Euratom CEA, "Magnetic confinement," *Institut de Recherche sur la Fusion Magnétique - CEA,* Sep. 9, 2016. [Online]. Available: http://www-fusion-magnetique.cea.fr/gb/fusion/physique/modesconfinement.htm/. [Accessed: May 26, 2019].

[13] R. Arnoux, "ITER ... And then what?" *ITER,* May, 2014. [Online]. Available: https://www.iter.org/mag/3/22. [Accessed: May 26, 2019].

[14] P. Norvig and Stuart Russell, *Artificial Intelligence: A Modern Approach,* 3rd ed. Upper Saddle River, NJ : Prentice Hall, 2010.

[15] D. Wolpert, "The Lack of A Priori Distinctions between Learning Algorithms," in *Neural Computation*, pp. 1341-1390, 1996.

[16] S. P. Lloyd, "Least squares quantization in PCM," *Information Theory, IEEE Transactions on 28.2,* pp. 129-137, 1982.

[17] D. A. Reynolds, "Gaussian Mixture Models," in *Encyclopedia of Biometrics*, 2009.

[18] I. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," in *Philos Trans A Math Phys Eng Sci.,* April 13, 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4792409/. [Accessed: June 7, 2019].

[19] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," Parallel Distributed Processing. Vol 1: Foundations. MIT Press, Cambridge, MA, 1986.

[20] I. Shafkat, "Intuitively Understanding Variational Autoencoder," *Towards Data Science,* Feb. 4, 2018. [Online]. Available: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf. [Accessed: June 8, 2019].

[21] S. Perera, R. Alwis, "Machine Learning Techniques for Predictive Maintenance," *InfoQ,* May 21, 2017. [Online]. Available: https://www.infoq.com/articles/machine-learning-techniques-predictive-maintenance/. [Accessed: June 9, 2019].

[22] M. Barlow, "Predictive maintenance: A world of zero unplanned downtime," *O'Reilly,* Feb. 15, 2015. [Online]. Available: https://www.oreilly.com/ideas/predictive-maintenance. [Accessed: June 9, 2019].

[23] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection : A Survey," *ACM Computing Surveys (CSUR),* vol. 41, no. 3, Article No. 15, July 2009.

[24] E. Hannan, *Multiple time series,* Wiley series in probability and mathematical statistics. New York: John Wiley and Sons, 1970.

[25] J. Brownlee, "8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset," *Machine Learning Mastery,* August 19, 2015. [Online]. Available: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/. [Accessed: June 9, 2019].

[26] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, 2001.

[27] B. Gal I. "Bayesian Networks," in *Encyclopedia of Statistics in Quality and Reliability,* F. Ruggeri, R.S. Kennett, F.W. Faltin, Eds., John Wiley & Sons, 2007.

[28] O. Cappé, *Inference in Hidden Markov Models,* Springer, 2007.

[29] K. Mehrotra, C. MohanHua, H. Huang, *Clustering-Based Anomaly Detection Approaches,* Springer, 2017.

[30] A. Baru, "Three Ways to Estimate Remaining Useful Life for Predictive Maintenance," *MathWorks,* n.d. [Online]. Available: https://www.mathworks.com/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance.html. [Accessed: June 5, 2019].

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural Computation,* 1997.

[32] C. Yeh *et al.,* "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets," in *16th IEEE International Conference on Data Mining,* 2016.

[33] D. De Paepe, O. Janssens and S. Van Hoecke, "Eliminating noise in the matrix profile," in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods,* 2019. pp.84-93.

[34] University of California, "The UCR Matrix Profile Page," *University of California,* n.d. [Online]. Available: https://www.cs.ucr.edu/ eamonn/MatrixProfile.html. [Accessed: May 31, 2019].

[35] scikit-learn, "Manifold learning," *scikit-learn,* n.d. [Online]. Available: https://scikit-learn.org/stable/modules/manifold.html. [Accessed: June 2, 2019].

[36] P. Rousseeuw, M. Hubert and S. Van Aelst, "Multivariate Outlier Detection and Robustness," in *Handbook of Statistics, vol. 23: Data Mining and Computation in Statistics,* C.R. Rao, E. Wegman, and J.L. Solka, Eds. Amsterdam: Elsevier North-Holland, 2005, pp. 263-302.

[37] T. Kuzin and T. Borovicka, "Early Failure Detection for Predictive Maintenance of Sensor Parts," in *ITAT 2016 Proceedings, CEUR Workshop Proceedings,* vol. 1649, pp. 123–130, 2016.

[38] D. Clevert, T. Unterthiner and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations 2016* [Online]. Available: arXiv, https://arxiv.org/abs/1511.07289. [Accessed: May 3, 2019].

[39] Cross Validated, "Can I use ReLU in autoencoder as activation function?" *Stack Exchange,* April 19, 2017. [Online]. Available:

https://stats.stackexchange.com/questions/144733/can-i-use-relu-in-autoencoder-as-activation-function. [Accessed: June 3, 2019].

[40] S. J. Lacey, "The Role of Vibration Monitoring in Predictive Maintenance," *Schaeffler, INA, FAG,* n.d. [Online]. Available: https://www.schaeffler.com/remotemedien/media/_shared_media/08_media_library/01_publications/schaeffler_2/technicalpaper_1/download_1/the_role_of_vibration_monitoring.pdf. [Accessed: June 4, 2019].

[41] PWW, "How to use Condition Based Maintenance Strategy for Equipment Failure Prevention," *PWW,* n.d. [Online]. Available: https://www.lifetime-reliability.com/cms/free-articles/maintenance-management/condition-based-maintenance/. [Accessed: June 4, 2019].

[42] J. Stephens, "Switch Detection Automation," Unfinished manuscript, UK Atomic Energy Authority, 2019.

[43] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, 1977.

[44] M. Peikari, S. Salama, S. Nofech-Mozes and A. L. Martel, "A Cluster-then-label Semi-supervised Learning Approach for Pathology Image Classification," Scientific Reports 8, Article number: 7193, 2018.

[45] G. Seif, "The 5 Clustering Algorithms Data Scientists Need to Know," *Towards Data Science,* Feb 5, 2018. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. [Accessed: June 11, 2019].

[46] E. L. Kaplan, P. Meier, "Nonparametric estimation from incomplete observations," J. Amer. Statist. Assoc. 53 (282), 1958.

# List of Figures

# Abbreviations

**JET** – Joint European Torus
**ITER** – International Thermonuclear Experimental Reactor
**RUL** – Remaining Useful Life(time)
**PFGC** – Poloidal Flywheel Generator Converter
**EM** – Expectation-Maximization
**GMM** – Gaussian Mixture Model