Norwegian University
of Life Sciences

Master's Thesis 2022 30 ECTS
Faculty of Science and Technology

Predictive Maintenance in Hydropower Plants: A Case Study of Valves and Servomotors

Ernes Çereku
Environmental Physics and Renewable Energy

Preface

This thesis is written at the Faculty of Science and Technology at the Norwegian University of Life Sciences (NMBU) in the spring of 2022. The thesis is 30 ECTS and marks the end of a 5-year masters degree in Environmental Physics and Renewable Energy. The thesis is delivered on May 14^{th}, 2022. The author can be contacted by email at ernesc@proton.me. The code and models used during this thesis are uploaded on GitHub and can be accessed via this link: https://github.com/ernesc/masters_thesis_2022.

The thesis is done in collaboration with Statkraft, Norway's biggest renewable energy provider, as part of the Generic Life project. The goal of the Generic Life project is to create generic lifetime and maintenance frequency assessment systems for hydropower plant systems and components. The Generic Life project aims to make use of Functional Mock-up Interface, a free standard for dynamic models exchange. Each major system of a power plant is assigned to a Functional Moch-up Unit which collects standardized data to give an output of the remaining useful life of the equipment.

Acknowledgements

There are many people I am thankful for, that have helped and stood by me during my work on this thesis.

I'd like to start by thanking Bjørn Sonju-Moltzau, my professor during my bachelors degree in Renewable Energy, who passed away last year. With his experience from Arendals Fossekompani, Statkraft, Point Carbon, Reuters, Multiconsult, and NVE, he inspired me to specialize in hydropower. His engaging and inspiring lectures are some of the most memorable moments of my time at NMBU. He will be missed.

Jorge Mario Marchetti, my supervisor has put in a lot of time guiding me through my work on the thesis. He has followed closely every single draft of the thesis with notes and discussions on how to tackle problems from various angles. We even enjoyed a car ride to the Nore 1 hydropower plant. Jorge, thank you very much for your contributions! I am deeply grateful that I got the opportunity to work with such an extinguished professor!

I'd like to thank my external supervisor Erik Jacques Wiborg for giving me the opportunity to join the Generic Life project, as well as for his hard work on collecting the data needed for the analysis. The discussions and the weekly meetings have been insightful and of huge help. A well-deserved thank you also goes to Trond Bergan and Ole Gunnar Haug and the whole Statkraft team who welcomed us at Nore 1 HPP, gave us a tour of the power plant, and brainstormed with us several ideas around the various issues.

Thank you to Oliver Tomic, my co-supervisor, who has been of huge help with the data analysis and Machine Learning models, as well as the overall help with scoping the thesis.

Thor Arne Hvam Bruun provided the necessary data for the servomotor monitoring. Discussions with Thor Arne on the data were key results of the thesis, so thank you! Frode Halvorsen has been of help with the discussions we had and his insights from EDR & Medeso. Sondre Grevle Iveland has been an important source of information on this thesis with his experience within hydropower control from Enestor AS. I want to thank Truls Edvardsen Aarønes from Hymatek for his input on the servomotor monitoring and the combination of Machine Learning with physical limitations.

I am grateful to the Norwegian University of Life Sciences. My time at the university has been memorable. I want to thank Vilma Bischof and Tendai Chella Bengtsson from the student information center (SiT) who have been there for me since I moved to Norway. I want to thank the team at the writing center as well, especially Shelby Geitner who has gone thoroughly through the thesis!

There are many more people I want to thank, especially everyone I met at the PTK conference, as well as my colleagues at Norsk Gjenvinning who were always eager to hear about my work on this thesis.

Last but not least, I am deeply grateful to my family for supporting and encouraging me during my studies. My friends as well deserve a big amount of gratitude for their support. I am deeply grateful to each and every one of you!

"*Prevention is better than cure*"

Desiderius Erasmus

Abstract

Digitalization has opened the opportunity for a fourth industrial revolution and the hydropower industry is taking charge of enabling digitalization in their operation. There are a lot of studies on predictive maintenance, however, there are, to our knowledge no studies on system-specific predictive maintenance for hydropower. To bridge this gap, the idea of system-specific, Machine Learning driven Predictive Maintenance is explored. Two systems are chosen as a use-case for this thesis: valves and servomotors. With the increasing amount of intermittent renewable energy resources entering the power system, the need for flexibility in the power grid is unequivocal. Valves and servomotors are key components of hydropower control and thus will play a pivotal role in securing flexibility to the grid.

The first system assessed is the main valve. In order to make this analysis easily applicable, the data that is already being collected at Nore 1 hydropower plant is analyzed in order to assess the possibility of maintenance prediction from limited data. Unfortunately, this did not achieve the desired results for the data collected from the valve sensors. This is due to the fact that only one variable was measured, in this case, the opening and closing time-lag of the valve. However, this thesis presents a framework for data collection that allows the use of Machine Learning for predictive maintenance. Various sensors are suggested based on several published works on predictive maintenance.

The second system assessed is the servomotor that controls the guiding vanes in a Francis turbine. Servomotors are key components of hydropower control. Due to the data not being collected by Statkraft at the time of the study, this data was provided by one of Statkrafts suppliers. By making use of the historical data of pressure as a function of the piston position, a boundary for where new values should be expected is computed by making use of One Class Support Vector Machine. Another embodiment of this case is presented where force is given as a function of piston position, which yielded better results. When new values are being measured, the data is presented as a bullet chart that visualizes the distance of new values compared to the boundary computed by the One Class Support Vector Machine. This tool could easily be applied to other servomotors which perform other tasks such as controlling water injection to a Pelton turbine or opening and closing of the valve, whether they are butterfly or ball valves. Suggestions for further data collection are presented in order to make use of more data for the use of Machine Learning in Predictive Maintenance.

Sammendrag

Digitalisering har ledet frem til en fjerde industriell revolusjon og vannkraft bransjen er i ferd med å digitalisere sin operasjon. Under literaturstudien er det ikke funnet noen publiseringer innen systemspesifikk maskinlæringsdrevet predikativ vedlikehold. I denne masteroppgaven blir muligheten for bruk av systemspesifikk, maskinlæringdrevet predikativ vedlikehold innen vannkraftverk utforsket for å vekke interesse innen dette feltet. To av vannkraftverkenes maskiner er brukt som eksempler og utforsket: ventiler og servomotor. Økende mengder uregulerbar strøm er introdusert i kraftnettet og behovet for fleksibilitet øker. Ventiler og servomotor er nøkkeldeler av vannkraftverk regulering og spiller en stor rolle i å sikre flexibilitet til strømnettet.

Det første systemet som ble analysert er ventiler. For å gjøre analysen og resultatene enkelt anvendbare, blir data som allerede er innsamlet analysert for å utforske muligheten for predikativ vedlikehold med begrenset data. Analysene basert på data samlet inn fra sensorne montert på ventilene ble dessverre ikke konklusive. Det er utfordrende å forutsi fremtiden når man bare har en variabel å ta utgangspunkt i. Likevel presenteres det et prinsipielt rammeverk for innsamling av data som gjør det mulig å ta i bruk maskinlæring for predikativ vedlikehold. Ulike sensorer er foreslått, basert på relevant litteratur innen ventiler og maskinlæring drevet predikativ vedlikehold.

Det andre systemet analysert under studien er servomotorer som styrer vannet i en Francis turbin ved å regulere vinklingen til skovlene. Dataen innsamlet om servomotoren er en god indikator på tilstanden til servomotoren. Ettersom dataen var ikke samlet inn av Statkraft da studien ble utført, ble dataen hentet fra en av Statkraft sine leverandører. En One Class Support Vector Machine ble brukt for å beregne foventet verdi av differansetrykk over stempelkamrene, som funksjon av stempel posisjon. En kulegraf som viser avstanden mellom grensen og nye verdier er visualisert. En annen metode er også presentert hvor man regner ut kraft på begge sider av stempelkamrene gjennom trykk for å vise kraft som funksjon av stempel posisjon. Dette ga bedre valideringsresultater i forventet differansekraft over tempelkamrene. Verktøyet kan enkelt bli anvendt til andre servomotorer som styrer vannmengden i en Pelton turbin eller åpning og lukking av ventilene, uavhengig av om det er spjeld- eller kuleventiler. Forslag til videre data innsamling er presentert for å ta i bruk maskinlæring for predikativ vedlikehold.

Contents

List of Figures

List of Abbreviations

ADAM	**Ada**ptive **M**ovement estimation
ANN	**A**rtificial **N**eural **N**etwork
ARIMA	**A**uto**r**egressive **I**ntegrated **M**oving **A**verage
AT&T	**A**merican **T**elephone and **T**elegraph Company
CFD	**C**omputational **F**luid **D**ynamics
CNN	**C**onvolutional **N**eural **N**etwork
COP	**C**onference **O**f the **P**arties
DCBM	**D**ata-driven, **C**ondition-**b**ased **M**aintenance **F**ramework
FEM	**F**inite **E**lement **M**ethod
GHG	**G**reen **H**ouse **G**ass
HPP	**H**ydro**p**ower **P**lants
IEA	**I**nternational **E**nergy **A**gency
IoT	**I**nternet **o**f **T**hings
IPCC	**I**ntergovernmental **P**anel on **C**limate **C**hange
IRENA	**I**nternational **R**newable **En**ergy **A**gency
IRES	**I**ntermittent **R**enewable **E**energy **S**ources
KPI	**K**ey **P**erformance **I**ndicator
LSTM	**L**ong **S**hort-**T**erm **M**emory
MAS	**M**ulti-**A**gent **S**ystem
ML	**M**achine **L**earning
MLP	**M**ulti**l**ayer **P**erceptron
NASA	**N**ational **A**eronautics and **S**pace **A**dministration
NN	**N**eural **N**etwork
NREL	**N**ational **R**enewable **E**nergy **L**aboratory
NZE	**N**et **Z**ero **E**missions
PdM	**Pred**ictive **M**aintenance
PSH	**P**umped **S**torage **H**ydropower
PvM	**Pre**ventive **M**aintenance
R2F	**R**un to **F**ailure
ReLu	**Re**ctified **L**inear **u**nit
RBF	**R**adial **B**asis **F**unction
RHPP	**R**eservoir **H**ydro**p**ower **P**lant
RMSE	**R**oot **M**ean **S**quared **E**rror
RNN	**R**ecurrent **N**eural **N**etwork
RUL	**R**emaining **U**seful **L**ife
SARIMA	**S**easonal **A**uto**r**egressive **I**ntegrated **M**oving **A**verage
SCADA	**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition
SVM	**S**upport **V**ector **M**achine
UN	**U**nited **N**ations

Chapter 1

Introduction

1.1 Background

Around 52 billion tons of greenhouse gases (GHG) are being emitted every year [1]. According to the latest IPCC Assessment Report (The Physical Science Basis), it is unequivocal that the human influence has increased the temperature of land, atmosphere, and ocean [2]. Antonio Gueterres, the Secretary General of the UN, stated that this report is a "code red" for humanity and that the world is getting close to the point of no return [3]. IEA has established a Net Zero Emissions scenario, in order to cut emissions and keep global warming below 2 degrees [4]. In order to reach the aims of this scenario, renewable energy resources will play a key role. However, most of these resources have variable production. With the increasing amount of intermittent renewable energy sources entering the power systems, the need for flexibility in power generation is unequivocal. In the past decade or so, the amount of installed intermittent renewable energy sources (IRES) has increased exponentially [5].

Hydropower plants have been around for a long time. Harnessing the potential energy stored in water dates back to 13 B.C. [6]. According to the executive director of IEA, Fatih Birol [7], hydropower is a forgotten clean electricity giant that needs to be put back on the climate and energy agenda, in order to meet the net zero goals.

Reservoir Hydropower Plants store potential energy in water reservoirs at a certain height. When needed, the water is then led down through the tunnels and penstock to a turbine. The potential energy of the water is turned into kinetic and pressure energy which is later transformed into rotating, mechanical energy through turbines and finally later transformed to electrical energy by generators. Reservoir Hydropower Plants (RHPP) have played and will play, a pivotal role in regulating the intermittent power produced by IRES [8]. However, the increasing irregularity of production will generally lead to a reduced lifetime of the equipment.

Industries and governments have an increasing focus on refurbishment [9] and circularity[1] [10] in the recent years. It is important to increase the lifetime of equipment, as well as provide circularity to the equipment that are already in use at the end of their lifetime.

Although Predictive Maintenance has long been a goal for the industry, it is today often based on set frequency condition assessments. Condition monitoring presents

[1] Circularity means extending, or giving new life to products by refurbishing and recycling products.

information continuously and helps visualize faults without the need for frequent inspection. Predictive Maintenance gives the opportunity to have just-in-time maintenance based on the tools provided by the recent advances in sensor technology, IoT[2] and Machine Learning. This way one can provide maintenance when it is needed without the need for unnecessary waste of materials and expenses.

1.2 Thesis Aim and Structure

RHPPs are robust construction and have a long life cycle, however, there are various mechanical equipment that need to be maintained often. This thesis will go into the use of Machine Learning for the Predictive Maintenance of valves and servomotors. There is a huge amount of studies on the use of Machine Learning in Predictive Maintenance, as it will be shown in Chapter 2. To our knowledge, there are no studies on the use of Machine Learning driven Predictive Maintenance on valves and servomotors. This thesis aims to fill the gap in these studies by using Machine Learning as a tool for hydropower system-specific Predictive Maintenance by presenting two use-cases valves and servomotors.

For the valve use-case, data from Nore 1 hydropower plant is collected. The aim is to check if there is any way to notice previous maintenance performed based on historic data on the opening and closing time-lag of the valve. Later predictive algorithms are to be assessed and compared on their predictive accuracy based on the collected data.

Servomotor use-case data is obtained from one of Statkrafts suppliers. This use-case aims to present a condition monitoring mechanism based on the data obtained.

Finally, the thesis will present a framework for data collection that enables the use of Machine Learning driven Predictive Maintenance.

In Chapter 2 a review of relevant work will be presented, both on the mechanical equipment and Predictive Maintenance, including Machine Learning.

The theory of the mechanical equipment and Machine Learning models used during the data analysis and framework will be presented in Chapter 3

The methods for data analysis are presented in Chapter 4.

The analysis and results will be presented in Chapter 5. Challenges with the lack of data obtained so far will be discussed, as well as a framework for data collection and use of various sensors will be presented.

Finally, Chapter 6, will present the thesis' conclusions and introduce further work to be carried out.

[2]The term IoT represents the connectivity of equipment to internet, enabling communication between equipment and humans, thus making them "smart" [11].

Chapter 2

Literature Review

Various publications have been assessed in order to gather knowledge on both the mechanical equipment and the Machine Learning (ML) techniques that can be applied in the already acquired data as well as state-of-the-art techniques for future work.

The literature review can be divided into two main sections; mechanical equipment and Predictive Maintenance (PdM) combined with ML.

2.1 Mechanical Equipment

There is unfortunately little information that can be found on valves, probably due to their robustness and long lifespan. However, there are a few technical papers, like one by Arne Kjølle [12] which goes through various mechanical equipment in a HPP and one by Crane Company [13] that describes the various types of valves and some of their use cases.

Even though the valves analyzed in this thesis are butterfly valves, both spherical and butterfly valves are researched in order to see if measurements taken in spherical valves can be applied to butterfly valves. The operating systems and the physical properties are similar, so that makes it easier to compare and apply methods used in one type of valve to the other.

Shrivastava et Patel [14] present a Finite Element Method analysis for a large butterfly valve. A detailed description of the common pressure points and displacement of the valve housing is shown as well in this report [14]. High values of stress are noticed in the area between the support rib and disc area [14]. According to Azad et al [15], vortices are formed behind the valve disc during the opening in a CFD analysis. The results from the CFD analysis show that the opening angle has a huge impact on total head loss [15]. In an article by Fei Zhang et al. [16], a summary of several publications about the in-situ observations and test methods of spherical valve hydrodynamic closure is presented. The results show that vibrations and pressure pulsations are at their highest level when the valve is opened 34% [16]. The knowledge provided in these three publications [14]–[16] is useful in order to know what happens during the closing and opening process of the valve as well as help consider where to place sensors needed for further work.

Spherical valve behavior on a pumped storage hydropower plant (PSH) is investigated and modeled by Junginger et al. [17]. The models were simulated at various operation points. According to the simulations [17], the shaft torque was subject to overloads. The paper presents a closing law optimization and simultaneous closing process in order to avoid load rejection [17]. Even though the article focuses on two PSH power

plants with spherical valves, the principles can easily be implemented on a butterfly valve.

A computational model of the servomotor that operates the valve computed in a MATLAB/Simulink-based software, is presented in an article by Fang et al. [18], in order to compute the accuracy and effectiveness of the simulation system. The article describes thoroughly the physical principles of the servomotor and its relation to the valve. MATLAB [19] has developed its own toolbox within Predictive Maintenance for condition monitoring and remaining useful life (RUL) estimation.

2.2 Predictive Maintenance and Machine Learning

2.2.1 Maintenance Costs

An overview of costs related to maintenance is presented in order to inspire further work in this field due to cost reduction from PdM. For large hydropower plants, maintenance costs are assumed to be 2% to 2.5% according to IRENA [20]. However, in a decision making support paper by Heggset et al. [21], it is underlined that there are other hidden costs in maintenance planning due to the costs of choosing not to produce due to maintenance which might exceed the assumptions by IRENA [20], in times when power prices are high.

Other examples from the energy sector are offshore wind power and oil and gas. Maintenance costs account for 57%-71% for the offshore wind power sector according to an assessment by NREL in 2010 [22], while for the oil and gas industry these costs can vary from 15% to 70% [23]. Even though these costs might have come down in recent years due to continuous improvement of the technology, the materials price experienced an increase in 2021 [24], thus might have led to an increase in maintenance costs.

Downtime can give a good picture of maintenance costs due to failure. A survey from Uptime Institute [25] discovered that the costs of data center outages were between 100 000 and 1 million dollars for 47% of the companies and more than 1 million dollars for 15% of the companies surveyed. Another survey conducted by Nielsen Research [26], in 2005, showed that downtime costs in the automotive industry had an average of 22 000 dollars per minute.

2.2.2 Predictive Maintenance

The concept of Predictive Maintenance has been around for a long time. However, with the development of Industry 4.0[1], IoT and sensor technology, implementing Predictive Maintenance methods has become cheaper and more accessible [30]. These technologies have made condition monitoring, maintenance plans, fault prognosis, and diagnosis much more easy [31]. Ran et al. [32] present a good review of the latest development within the various systems and approaches for Predictive Maintenance. The report's [32] structure is easy to follow and groups the various Machine Learning algorithms in a simple way. The sources of this review [32] and the summary of

[1]The fourth industrial revolution, also known as Industry 4.0 is a product of a new way for the industry to manufacture, improve and distribute products [27]. The first industrial revolution came as a result of mechanization in 1733 [28]. New energy sources like electricity and synthetic products such as plastics, as well as mass production mark the second industrial revolution by the end of the 19^{th} century [28]. The third industrial revolution is marked by a rise in electronics and automation [29].

the different methods for Predictive Maintenance have heavily influenced and helped research during this thesis.

The industry has also taken a big interest in Predictive Maintenance. Companies like Seebo Interactive [33] and LimbleCMMS [34] have established guides for Predictive Maintenance. Even Microsoft [35], has developed a playbook on Predictive Maintenance for the use on their own platform, Azure. Recently, came EDR & Medeso with their approach to Predictive Maintenance with limited data by combining physical principles with digital twins [36]. Unlike other companies, EDR & Medeso explores the reasons for failure and maintenance in order to build the algorithms according to the specific issues [36]. A publication by Eker et al. [37] illustrates the drawbacks and advantages of the data-driven models and physical models. More methods are further described in books [38]–[42] which go into more detail on various assessments, vibration analysis, and decision support tools for Predictive Maintenance in various branches of the industry. The guides provided in these articles and the examples have heavily inspired the work on the framework that is presented in this thesis (see Section 5.3).

Predictive Maintenance is well established. However, real-world data is often stored in multiple formats, machines, and systems, thus making it extremely hard to make use of their data. Bekar et al. [43] presented an unsupervised ML technique for data pre-processing to structure the data using real-world data from the manufacturing industry. However the results from the study are very sensitive and in need of further work [43], thus this studies aim is to increase the use of ML in the industry, rather than present an actual solution. This is, unfortunately, the case for either the use-cases analyzed, where data collected is not enough for the use of ML-driven PdM.

A study by Zhang et Zhang [44] uses an LSTM-based model to predict the RUL of deteriorating systems, with data provided by NASA, and the predicted values are very close to the actual failures. A data-driven condition-based monitoring (DCBM) framework is also established for imperfect inspections in order to give probabilistic forecasting for maintenance decision making, as well as a regressor for performance evaluation, using a stacked autoencoder LSTM [44]. Figure 2.1 illustrates how the DCBM framework uses historical data in order to predict future failure.

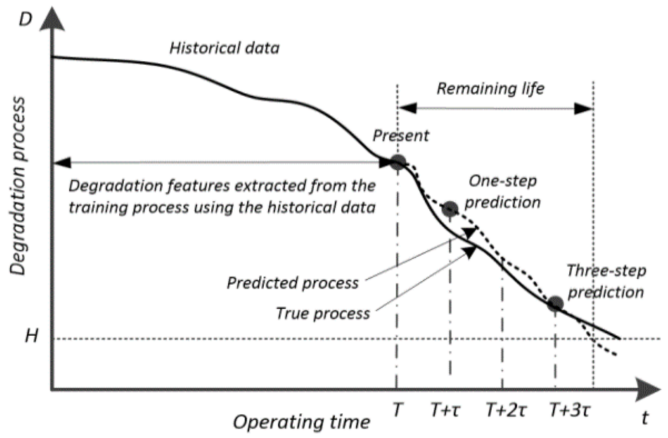


FIGURE 2.1: The DCBM process for the deteriorating equipment [44].

The information needed for Predictive Maintenance is often time dependent. Simple statistical algorithms, like ARIMA, have widely been used for time series prediction [45]–[46], however due to advances in computing, other deep learning algorithms, like

LSTM, seem to outperform ARIMA by 84%-87% [45]. Jason Brownlee [47] describes various deep learning tools for time series forecasting like MLP, CNN, LSTM [48] and a combination of these in order to increase accuracy. Brownlee [47], Hota et al. [49] and Wang et al. [50] have combined the Sliding Window Method with these deep learning tools in order to increase the accuracy further. Figure 2.2 by Kotu et al. [46] gives a nice illustration different pathways for time series forecasting.

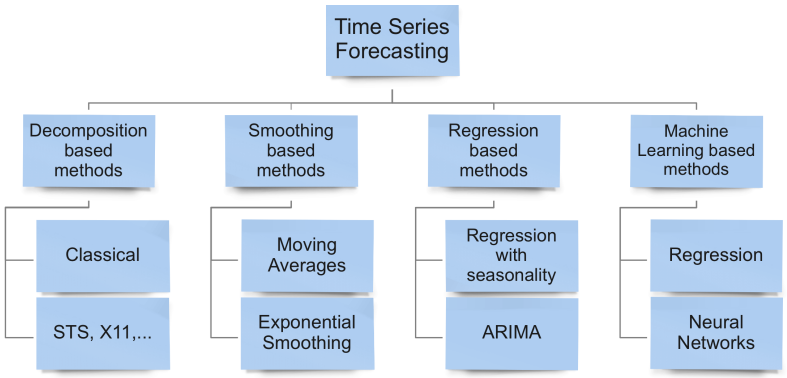


FIGURE 2.2: The taxonomy of time series (adapted from Kotu et al. [46] with permission from Elsevier).

Within the field of HPP, Jiang [51] attempted to apply a Multi-Agent System (MAS) and Artificial Neural Network (ANN) for Predictive Maintenance of the whole HPP based on data from several sensors. The model was applied in two HPPs [51]. However, this type of PdM introduces the Black Box issue. The Black Box issue is a term used in data science, to represent the lack of explanation of the results from an ML model. In the example by Jiang [51], one only knows that there is a need for maintenance in the power plant but not where or what type of maintenance. In a study led by Energy Norway [52], pressure data from the servomotor was analyzed and One Class SVM was used for data classification, which presented acceptable results. This inspired further work on the use of One Class SVM in this thesis (see Section 5.2).

2.3 Key Take-Aways

No cases of the use of system-specific ML-driven PdM within the hydropower industry were found during the literature review.

Research and modeling show that there are high-stress levels in various parts of the valve, as well as pressure pulsations. These issues generally lead to a reduced lifetime of the components.

Data gathered seems to be insufficient in most industries and there lacks a standardization for data collection. Multiple data-storage systems with various data formats present further challenges. Even high-performance algorithms have issues producing reliable results with such issues as lack of data and standardization.

Articles reviewed during the literature study show that Machine Learning models such as LSTM outperform conventional statistical models such as ARIMA. The Sliding Window Method combined with deep learning increases the prediction confidence of time series forecasting.

One Class SVM yields good results for condition monitoring of the servomotor.

Chapter 3

Theory

3.1 Mechanical Equipment

3.1.1 Valves

The conduits in most RHPP are provided with shut-off devices, valves [12]. According to the ISO/TS 81346-10:2015 standard, also known as the RDS standard, the main valve is part of the production unit (class A) and is defined as the system for controlling the water flow between the penstock and the turbine and is defined under class KA [53]. There are many different types of valves, depending on their function and requirements. According to Kjølle [12], the most relevant types of valves for hydropower are spherical, butterfly, gate, and ring valves.

In butterfly valves, the servomotor controls the opening of the valve. From measurements, at Nore 1 it takes around 60 seconds to open the valve. The first step of the opening process is to pressurize the downstream side of the valve. A by-pass pipe fills the downstream side of the waterway with water, as is shown in Figure 3.1.

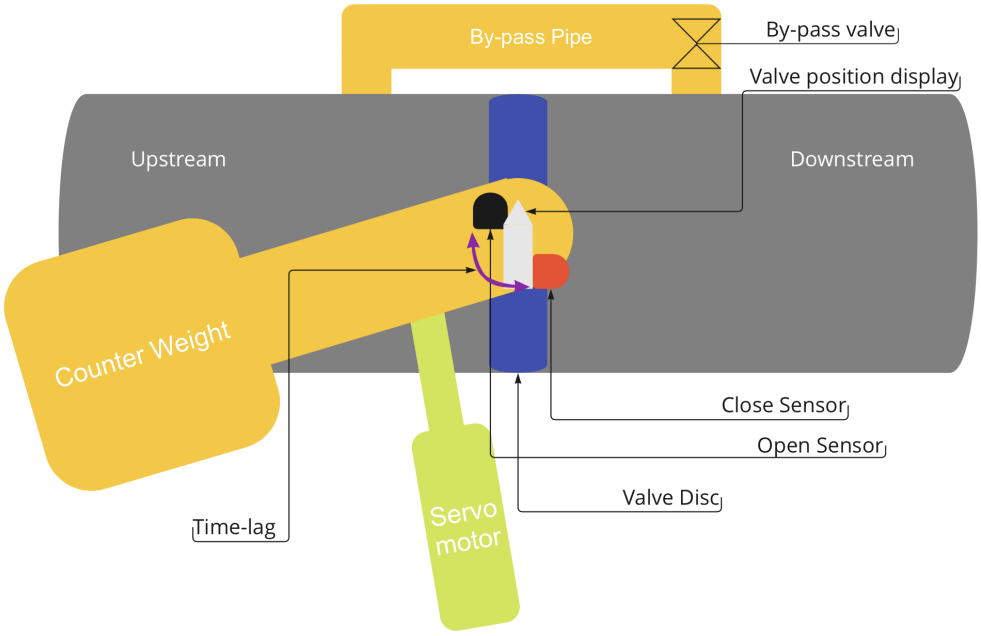


FIGURE 3.1: Illustration of a butterfly valve. Similar construction as the ones installed at Nore 1 HPP.

When the downstream side of the valve is filled with water, and the pressure is the same on both sides of the valve, the servomotor is pressurized and starts lifting the

counter weight, thus opening the gate. During the closing process, the pressure in the servomotor is reduced and the counter weight closes the valve. Figure 3.2 shows pressure measurements at an open valve at Nore 1 HPP, as earlier mentioned, pressure is equal on both sides when the valve is opened.

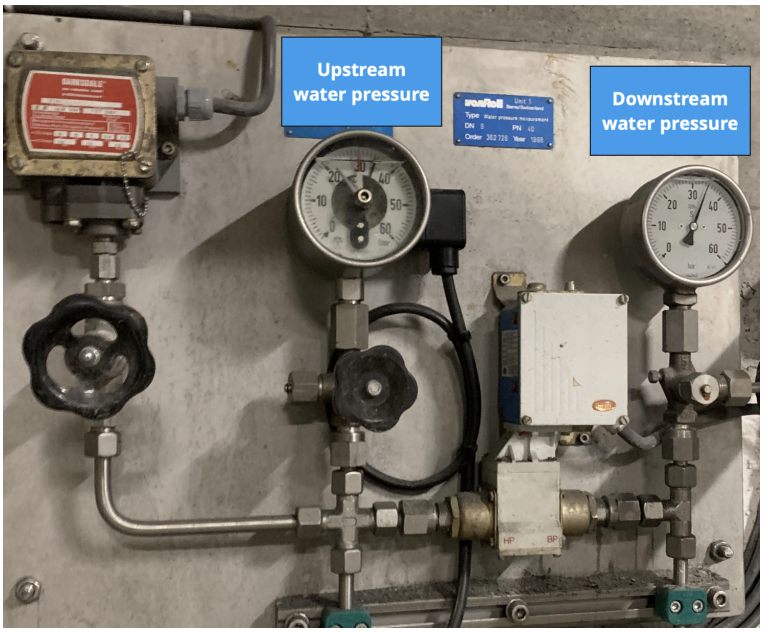


FIGURE 3.2: Pressure measurements downstream and upstream of valve 1 at Nore 1 HPP.

A flow meter is installed in the penstock, next to the valve, in order to detect anomalies in the flow so that emergency closure valves can close automatically in case of unexpected flow rise or penstock rupture.

There are two usual damages to be found on valves; normal wear and mechanical damages [12]. Valve inspection is difficult on the upstream side and its frequency should be determined on the base of material choice, probability of cavitation, vibrations or mechanical wear from sand, erosion, and so on [12].

According to interviews with operators at Nore 1 HPP, leakage inspection is issued every week. If the valve is not opened within five minutes, an error message is produced, however, it takes 30 minutes for an error message to come up in the SCADA system[1]. Maintenance on valves is initiated in case of failure.

3.1.2 Servomotor

Even though the valves might have different shapes, the operating mechanism is often a hydraulic servomotor operated either by water or oil pressure [12]. In butterfly valves the servomotor is pressurized in order to open the valve and lift the counter weight after the downstream side of the valve is filled with water. During the closing process the servomotor is depressurized and the counter weight closes the valve.

Figure 3.1 shows the placement of the servomotor in a the butterfly valves installed at Nore 1 HPP and Figure 3.3a) shows the operating system of the servomotor, also

[1]SCADA is a system of software and hardware elements that allows industrial organisations to record events, to interact with devices such as sensors, valves, pumps, and so on, control processes and gather real time data [54].

known as the hydraulic power unit (HPU). The HPU shown in Figure 3.3a) operates two servomotors at the same time. This pairing mechanism is in all the valves at Nore 1 HPP. The HPU is equipped with oil tanks, pumps, and an accumulator. The accumulator operates like a battery. Pressure, provided by the pumps is stored in the accumulator in order to be ready for use in the valve's piston when it's time for the valve to open.

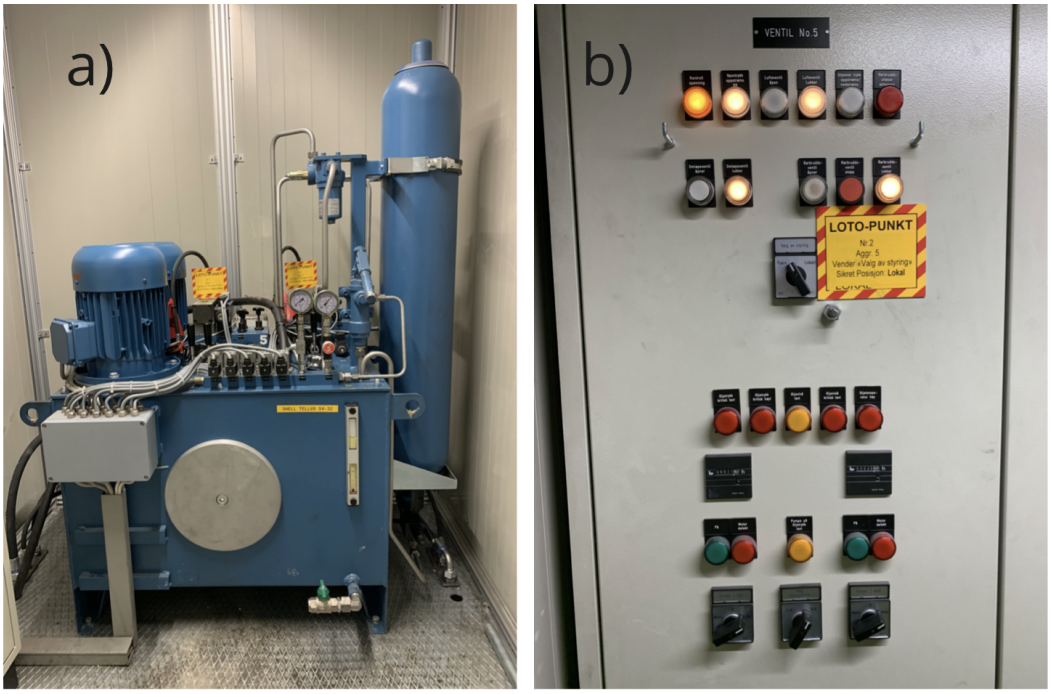


FIGURE 3.3: Pictures were taken at Nore 1 HPP. a) The servomotor's HPU with pressure sensors. b) A control cabinet for the valve on the upper part and monitoring system for oil amount and pressure in the servomotor in the lower part.

The amount of pressure per unit of time being provided to the accumulator is smaller than the amount of pressure per unit of time that the accumulator provides to the piston. Figure 3.3b) shows the control cabinet. On the upper part of the control cabinet the valve parameters, like pressure, are monitored. In case of pressure difference, an error message is sent to the SCADA system. In the lower part of Figure 3.3b), one can see that the oil amount and pressure are monitored.

At Nore 1 HPP, oil leakage in servomotors is inspected every month, while maintenance is issued every two years. During this maintenance oil and oil filters are changed and replaced. A detailed inspection of leakage is performed.

The servomotor is a key part of the hydro-turbine governing system [55]. In Pelton turbines, the servomotor controls the amount of water that is injected into the turbine. Figure 3.4 shows pictures taken from Nore 1. Figure 3.4a) shows how the controlling system is mounted on the turbine casing, while in Figure 3.4b), maintenance was being issued at the time so one could see inside the casing. Within the marked area, one can see the servomotors accumulators for the water injecting mechanism to the Pelton turbines.

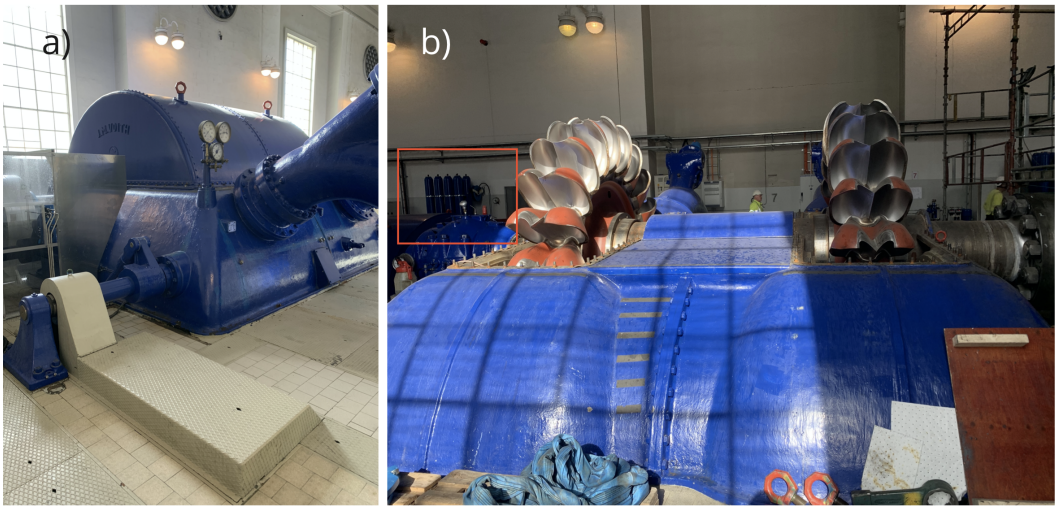


FIGURE 3.4: Pictures taken from Nore 1 HPP. a) Pelton turbine with pressure sensors and injecting mechanism. b) Pelton turbines under maintenance. Servomotor accumulators can be seen on the back.

Similarly, servomotors are also used to control the guide vanes in a Francis turbine. An illustration of the servomotors that control the guide vanes in a Francis turbine is shown in Figure 3.5. During this project, data from a similar servomotor was gathered for analysis.

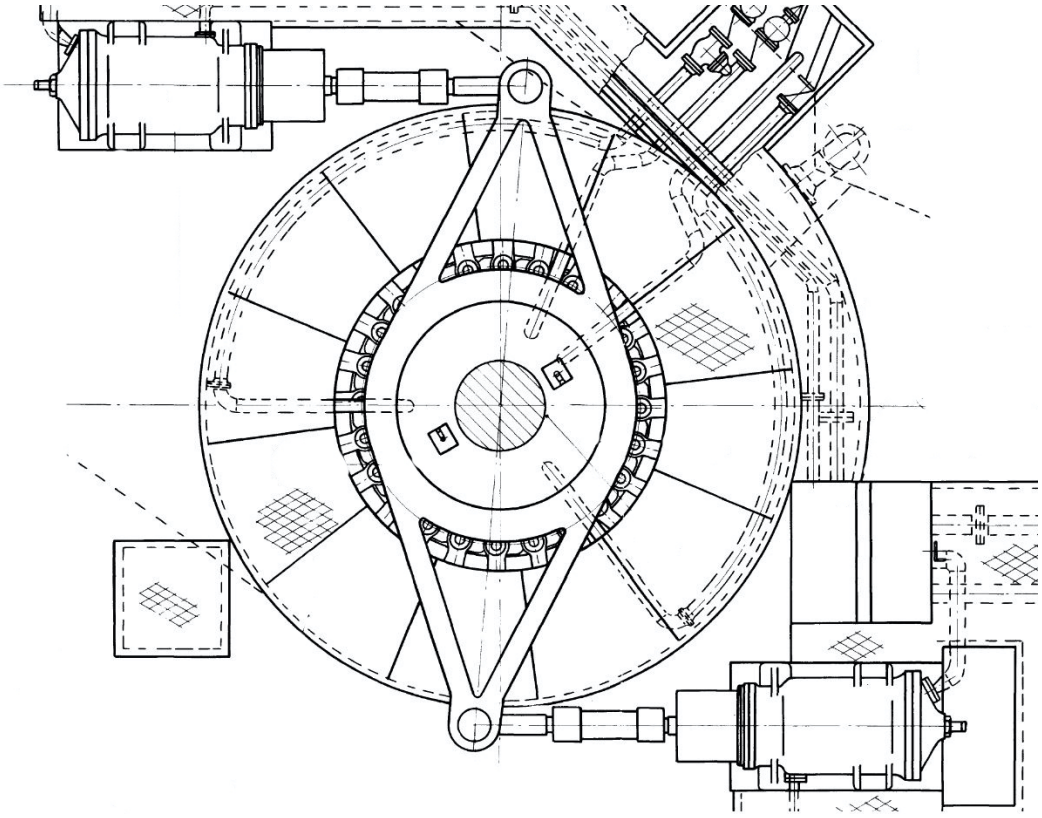


FIGURE 3.5: Illustration the servomotors that controls the guiding vanes of a Francis turbine. (Courtesy of Statkraft.)

3.2 Predictive Maintenance and Machine Learning Techniques

There are two methods used to predict a deteriorating system, physical-based methods and data-driven methods, also known as Machine Learning based methods [37]. This thesis will focus mainly on data-driven methods. Thus, in order to make it easy to follow, Predictive Maintenance (PdM) and Machine Learning (ML) methods will be presented together.

3.2.1 Predictive Maintenance

In various literature, maintenance strategies are divided into preventive, predictive, and breakdown maintenance or R2F. In the European Standard EN 13306:2017 [56], maintenance includes more definitions and thus is more complex. PdM is allocated under Condition based maintenance, which is allocated under Preventive Maintenance (PvM), as shown in Figure 3.6, inspired by EN 13306:2017 [56]. As shown in the literature review (in publications such as Ran et al. [32]) Corrective maintenance (also known as R2F in various literature) and predetermined maintenance are outdated maintenance methods and in this fourth industrial revolution, one should make use of sensors, IoT and Machine Learning in order to schedule maintenance when it is needed.

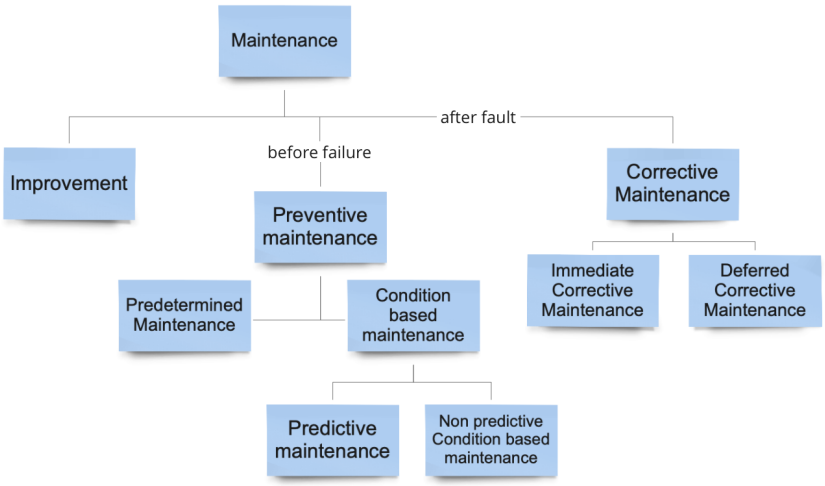


FIGURE 3.6: Maintenance methods and definitions. (Printed with permission from Standard Norway [56].)

Corrective maintenance, according to EN 13306:2017 [56], is issued after fault recognition. This leads to an unscheduled production stop and both material and water value[2] expenses.

Predetermined maintenance is a type of PvM, issued according to established intervals, either in intervals of time or the amount of unit of use [56]. Predetermined maintenance is done on critical equipment in order to avoid unexpected failure, however, this might lead to unnecessary costs (e.g. material, labor, and transportation costs) as well as material replacement.

[2]The water value is the estimated value of the water at any given time in the different reservoirs. RHPP can choose not to produce when the power prices are low but wait until the prices exceed the estimated water value (i.e. producing at a profit). Unexpected shutdowns may hinder production when profits would have been high. In a worst-case scenario, water could overflow the reservoir and its production value would be lost.

> Predictive Maintenance [is] condition-based maintenance carried out following a forecast derived from repeated analysis or known characteristics and evaluation of the significant parameters of the degradation of the item.
>
> -EN 13306:2017 [56]

Predictive Maintenance gives the possibility to schedule maintenance just before failure. Various Machine Learning techniques can be used to learn from past experiences and extrapolate this data in order to predict maintenance or failure. The following sections will go briefly through the theory of relevant techniques used for Predictive Maintenance.

3.2.2 Autoregressive Integrated Moving Average

Developed by Box et Jenkins [57] back in 1970, ARIMA has widely been used for time series forecasting. Having been around for a while, the implementation of ARIMA is relatively straightforward in most statistical tools and programming languages. Kotu et al. [46] expresses the ARIMA model as:

$$y'_t = I + \alpha_1 y'_{t-1} + \alpha_2 y'_{t-2} + ... + \alpha_p y'_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + ... + \theta_q e_{t-q}, \quad (3.1)$$

where:

- y'_t is the regression function at the time t
- I shows that the data values have been replaced with differenced values to obtained stationary data
- α represents the smoothing parameter
- θ represents the weighted moving average
- p is the predictor for the autoregressive part
- q is the lagged error for the moving average part
- e_t is the is the forecast error at a given time t
- d is a hidden parameter in equation 3.1 but is coded in the model and represents the order of differencing.

Another embodiment is Seasonal ARIMA, better known as SARIMA, which can take into account the seasonality of the data.

3.2.3 Multilayer Perceptron

Multilayer Perceptrons, also known as MLP, approximate a mapping function from input variables to output variables [47]. The neurons in an MLP are called perceptrons[3] [59]. MLPs are robust to noise and nonlinear models [47], giving MLPs a big advantage in the field of time series prediction, since the data is often nonlinear and the sensors are often faulty, thus generating noise in the data. According to Brownlee

[3]The perceptron combines the inputs with weights in order to compute the net input, which is passed on to the threshold function; after the error is calculated the weights are then updated in order to minimize the misclassification errors [58].

[47], a simple feed-forward NN, like MLP, should be evaluated as a base case for more advanced models. In time series forecasting MLPs take previous observations as input features in order to predict future time steps [47].

3.2.4 Long Short-Term Memory

LSTMs are a type of RNN and part of a complex area of Deep Learning [48]. LSTM was introduced by Hochreiter and Schmidhuber [60] back in 1997, to overcome the vanishing gradient problem. The study found out that LSTMs outperformed other RNNs at the time and managed to solve tasks that hadn't been solved so far, at that time [60].

> ... LSTM add the explicit handling of order between observations when learning a mapping function from inputs to outputs, not offered by MLPs or CNNs.
>
> -J. Brownlee, Long Short-Term Memory Networks With Python, 2017 [48]

The computational unit of LSTM, shown in Figure 3.7, is often called memory cell, memory block or just cell [47]. LSTM cells are comprised of weights and three types of gates. The forget gate, which decides which information goes through; the input gate, which updates the cell states and the output gate which allows for the updating of the hidden units [58], as shown in Figure 3.7. The structures of LSTM cells are complex, however, TensorFlow[4] has established easy methods in order to call LSTM cells [58], so more theory on LSTM is outside the scope of this study.

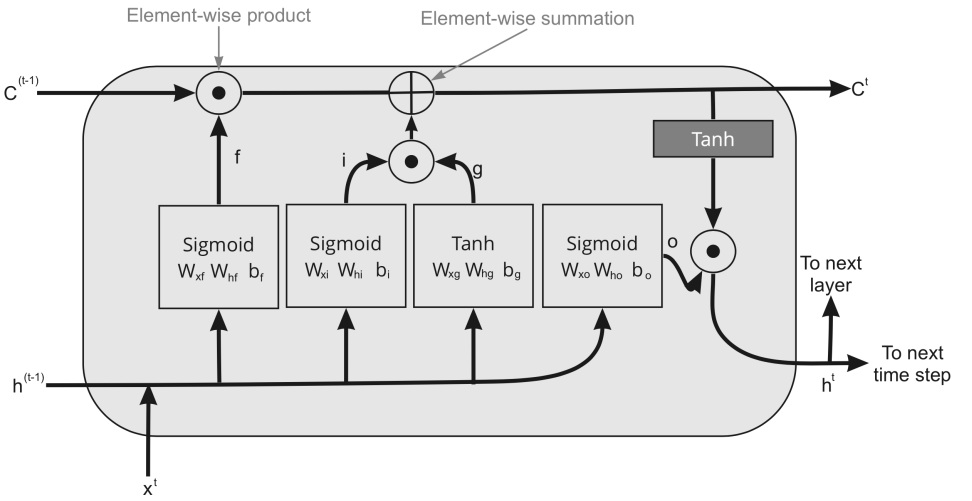


FIGURE 3.7: Representation of the LSTM memory cell. x^t refers to the input data at time t, C is the cell state, w is the weight, f is the forget gate, i is the input gate, h indicates the units and o is the output gate. (Printed with permission from Packtpub [58].)

LSTMs have various advantages compared to other time series estimators like their robustness to noise as well as the ability to work with nonlinear and multivariate data [48]. Compared to other RNNs, LSTMs lack the scaling problems, as well as vanishing gradient problems [62].

[4]TensorFlow is an open source platform for Machine Learning, developed by Google, with a comprehensive and flexible ecosystem of tools, libraries, and other resources [61].

3.2.5 Convolutional Neural Network

CNN is a type of ANN that mostly focus on pattern recognition and image processing. These types of NNs are compromised of convolutional layers, pooling layers and fully-connected layers [63]. CNN model operates directly on raw data and learn how to extract features from raw data which can be useful for the problem to be solved [58].

CNNs are composed of several convolutional layers and subsampling layers, known as pooling layers, which are connected in the end by a Dense layer [58]. The convolution is done on two vectors, one for the input data and one for the weights, and produces a third vector [58]. The problem in the mathematical expression for convolution is that it goes from negative infinity to positive infinity [64], while a finite amount of data is always used. In order to solve this problem, padding is introduced, where one adds a certain amount of zeroes on both sides of the input [58]. Subsampling is done by the means of pooling which reduce the size of the features and introduce some sort of variance [58].

Qiao et al. [65] and Brownlee [48] explore the idea of using CNN for time series prediction as well as combining CNN with LSTM in order to increase the confidence of time series prediction.

3.2.6 Support Vector Machine

Pattern recognition dates back to 1936 [66], by Ronald A. Fisher who described the first algorithm for pattern recognition. This work has been under continuous improvement ever since. Work done by Cortes and Vapnik at the AT&T Bell Laboratories presents another milestone in pattern recognition with the development of SVM [67].

SVMs are supervised[5] Machine Learning algorithms used for classification, regression, and outlier detection [68], which can be seen as a further development of the perceptron. Unlike the perceptron, the idea behind SVMs is to maximize the margin, while incurring a faulty observation when a sample is misclassified as illustrated in Figure 3.8.

Proof of work for several SVM embodiments can be found in Cortes' introduction of SVM [67], as well as recent work in handwriting recognition as well as image recognition [69].

During this analysis, a version of SVM is used, called One Class SVM. This is an unsupervised outlier detection algorithm and it produces a decision function that calculates the signed distance from the separating hyperplane [68]. The signed distance function gives positive values if the point x is within the decision boundary and negative if it is outside, as shown in the function below [70]:

$$x \begin{cases} \phi > 0, & x \in \Omega/\partial\Omega \\ \phi = 0, & x \in \partial\Omega \\ \phi < 0, & x \in R^2/\Omega, \end{cases} \tag{3.2}$$

where ϕ is the signed distance from the decision boundary, Ω is the boundary, ∂ is the partial derivative of the function, in this case the boundary function and R represents

[5] Most Machine Learning models make use supervised learning, where one has some input variables and some output variables based on input, where one can use the algorithm to learn a mapping function from input to output [47].

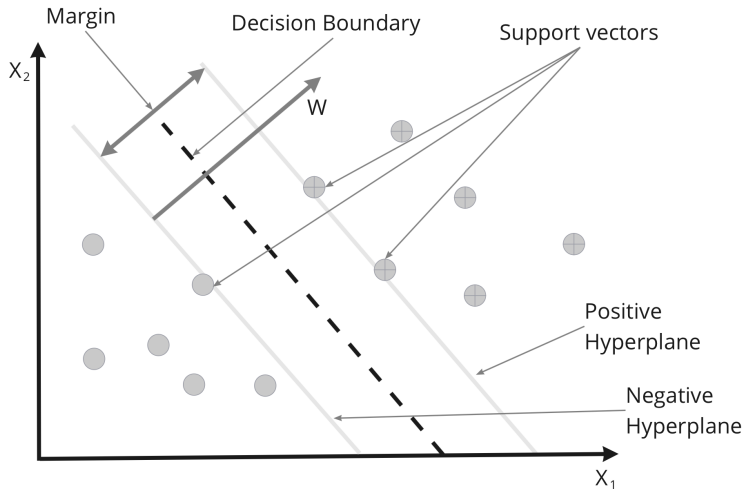


FIGURE 3.8: The margin is defined as the distance between the separating positive and negative hyperplanes. (Printed with permission from Packtpub [58].)

all the real numbers. One Class SVM will b used in order to define the required area for future measurements of pressure in the servomotor.

Chapter 4

Data and Methods

4.1 Time-lag of the opening and closing process of the valve

Data is gathered from Nore 1 HPP located at Rødberg municipality, which started producing electricity for the first time in 1928.

The opening and closing of the butterfly valve are controlled by the servomotor as has been illustrated previously in Chapter 3, in Figure 3.1. When the valve is closed, the disc is positioned perpendicular to the water direction. Here the positioning display is pointing to 90 degrees, as shown in Figure 4.1. After the downstream side of the valve is pressurized by the by-pass system, the servomotor lifts the counter weight, opening the valve, and placing it parallel to the water stream, i.e. at zero degrees, as shown in Figure 4.1.

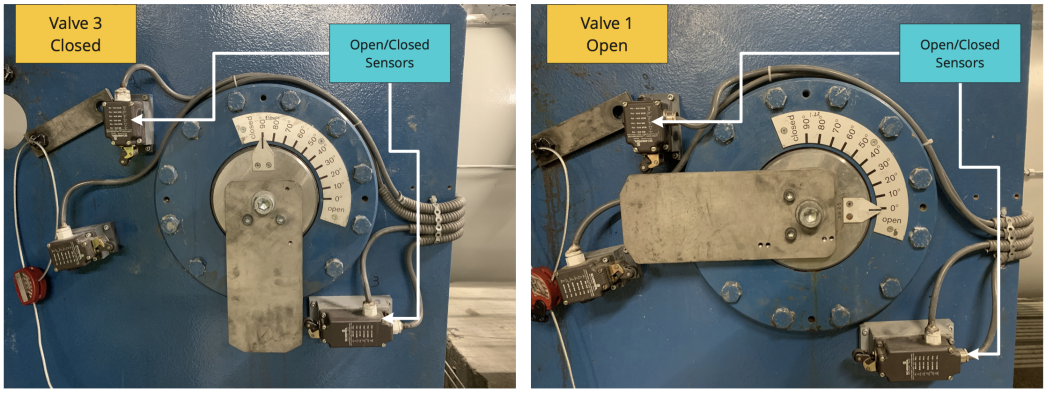


FIGURE 4.1: Valve 3, closed (left photo), and Valve 1, open (right photo). Photos taken by author at the Nore 1 HPP.

Two sensors are monitoring the valve position, one at fully opened and one at closed. These sensors have binary outputs, "On" or "Off". When the valve display system[1] points toward the open position and its back triggers the "Open Sensor," it gives an "On" signal, while the inactivated "Closed Sensor" gives an "Off" signal. When the valve position arrow moves away from the "Open Sensor", the signal is "Off" and as the valve position arrow's back reaches the "Closed Signal". The latter switches to an "On" signal.

[1]The valve display system is mounted at the rod of the valve disc and displays the position of the valve disc relative to the water direction.

The other sensor in the middle of the open and closed sensor is a fault sensor which activates in case the valve is not fully opened and sends an error message.

The data logged from these two sensors is logged in two separate tables. In order to calculate the time-lag, these tables had to be merged in order to calculate the time difference from when the "Open Sensor" turns off until the "Closed Sensor" turns "On", thus obtaining how long it takes the valve to close. After the time difference is calculated, two tables are produced; one for the time-lag of the opening process and one for the time-lag of the closing process, as illustrated in Figure 4.2.

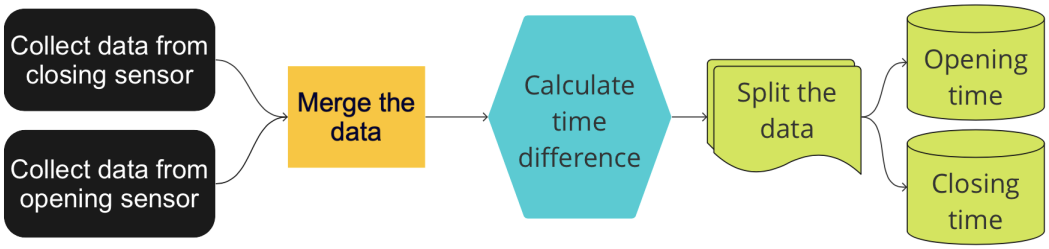


FIGURE 4.2: Flow chart for data preparation for the valves.

After the data was prepared, six data-sets were created in total; two data-sets each for the three valves assessed. Each of the 6 data-sets are first analyzed and then tested in various time series forecasting methods, both simple statistical methods like ARIMA as well as ML algorithms such as LSTM.

Models are trained for each valve for the opening and the closing process separately, as shown in Figure 4.3.

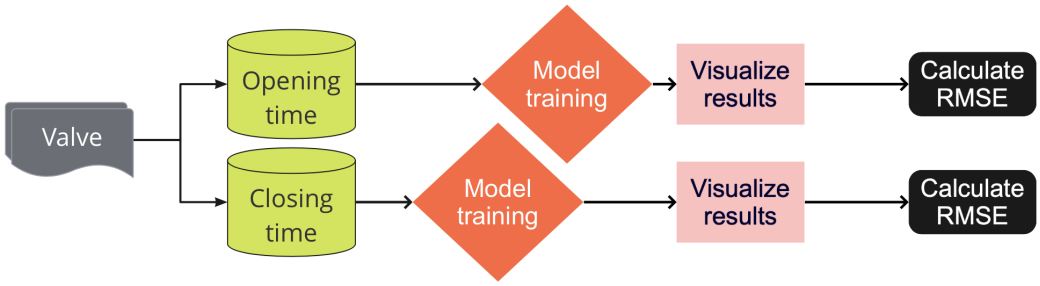


FIGURE 4.3: Flow chart for model training for the valves.

For all the Machine Learning models described in Sections 4.1.3-4.1.4, an optimizer called Adam is used. The adaptive moment estimation, better known as Adam optimization algorithm, is an extension to the stochastic gradient descent that is broadly adapted for deep learning [71]. The learning rate[2] in each weight update remains the same. Adam is used since Ruder [73] found it to have the best performance when tested against a gradient of optimization algorithms.

The mean squared error (MSE) is used as the loss function[3], calculated as follows:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y[i] - \hat{y}[i])^2, \tag{4.1}$$

[2]The change to the model during each step of this search process is called the "learning rate" is the most important parameter for a neural network in order to increase the performance of the model [72].

[3]The loss function evaluates how well the algorithm has adapted to the data-set [74].

where N represents the number of measurements, $y[i]$ is the value at time or cycle i and $\hat{y}[i]$ is the prediction. The activation function that is responsible for transforming the summed weighted input from the node into the activation of the node used for all predictive ML models is ReLU.

The rectified linear unit, usually called ReLU for short, is a piece-wise linear function that gives an output if the input is larger than, or equal to zero, else it will return zero, as shown in Figure 4.4. ReLu has become widely used since it is easier to train and has a better performance compared to sigmoid and hyperbolic tangent [75].

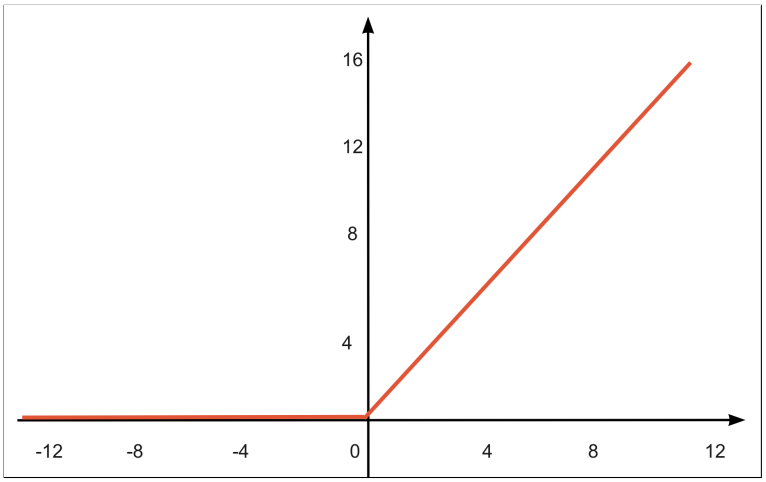


FIGURE 4.4: Representation of the rectified linear unit (ReLU)

4.1.1 Validation

In order to validate the model, the data is split into a training data-set and a validation data-set. Each valve operates at a different point in time, therefore cycle amounts vary during the 10 year time span of measurement. The training data-set is used in order to train the model and make predictions, while the validation data-set is used to test how well the prediction has performed. The validation data-set is set as a default value for the last 100 measurements for the opening and closing process of the valve. The data-sets vary from 350 to 400 data points, so the training data-set is comprised of the first 250-300 values and the validation data-set is comprised of the last 100 values, as shown in Figure 4.5.

In order to have a quantitative measurement of the performance RMSE is used to measure the error in all models. RMSE is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y[i] - \hat{y}[i])^2}, \tag{4.2}$$

where N represents the number of measurements, $y[i]$ is the value at time or cycle i and $\hat{y}[i]$ is the prediction. The lower the RMSE, the better the prediction. The RMSE returned in the end, is the mean of all predictions for each point, so it is important to visualize the prediction so that one can see whether the Machine Learning model is replicating the pattern or just producing a smoothed line.

In Figure 4.6 the loss is plotted against the number of epochs. One can see that the loss decreases as the number of epochs increases. The callback function saves the

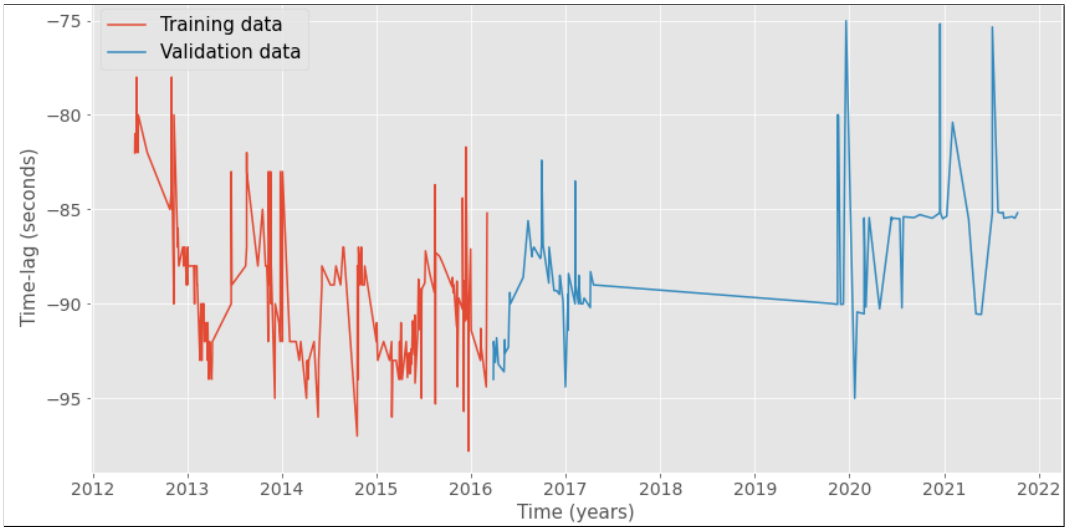


FIGURE 4.5: An example from closing time-lag of valve 1 of how the data is split into training and validation data-set.

model with the lowest loss from the validation data (test data). This function also looks at 100 epochs into the future to see if the loss decreases or not. If the loss does not decrease, the model stops and returns the results with the lowest RMSE.

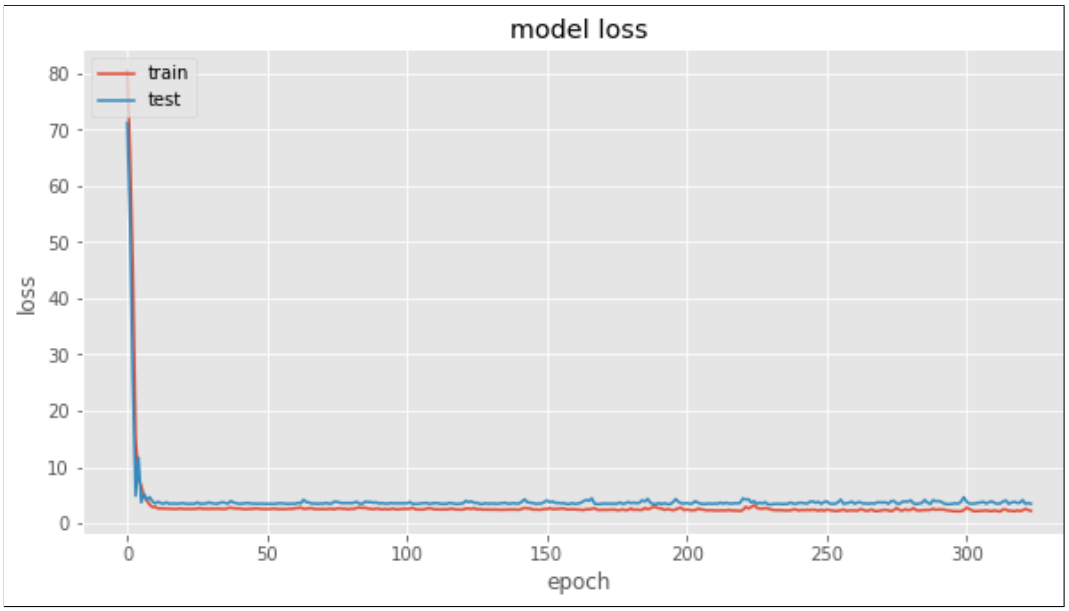


FIGURE 4.6: Example of model loss that depicts the validation of the data. This example is extracted from the CNN-LSTM model which will be presented in the following section.

4.1.2 ARIMA

ARIMA is a statistical model which is widely used for time series prediction. This model is usually used for univariate data, as is the case for this data-set. Since some seasonality is noticed in the data, as will be shown in the following chapter, SARIMA is used. It is fairly easy to add seasonality to the ARIMA model, thus enabling the use of SARIMA. The code for the ARIMA model includes an automatic function that chooses values for the different parameters of the ARIMA function. However tests are

done with both the automatic function and by trying various values of the differencing order d, predictor p, and lagged error q. In order to achieve high performance, the model is computed 50 times with various parameters.

4.1.3 Time-dependant LSTM

For better performance, data is scaled within a range of zero to one, with the help of the `MinMaxScaler` function provided by the preprocessing library of Scikit-learn [76]. The LSTM model built for this task is Stacked LSTM, which means that there are multiple hidden layers. In this case two LSTM cells with different amount of units are stacked and a Dense layer outputs the results as shown in the code below:

```
model = Sequential()
model.add(LSTM(units=100, activation='relu',
               return_sequences=True,
               input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=10, activation='relu'))
model.add(Dense(1))
```

Listing 4.1: Time-dependant stacked LSTM model.

The units in a hidden layer are inspired by neurons in the brain, however much less in amount and simpler than the biological brain [77]. A simple illustration of the code above is presented in Figure 4.7. The Dense layer is a fully connected layer that takes information from all units (see Figure 4.7) and predicts a value.

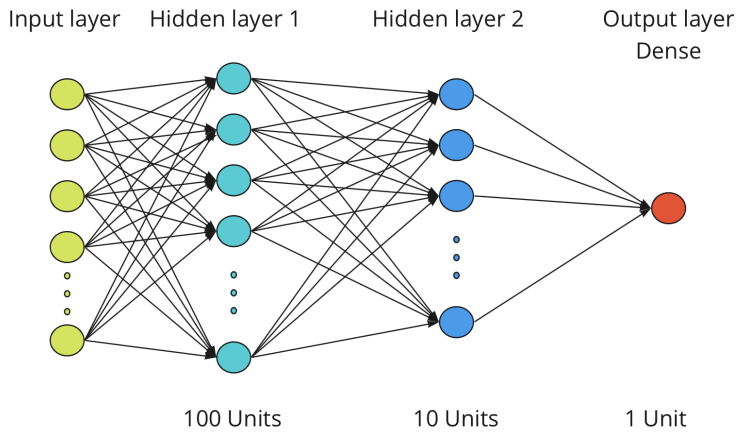


Figure 4.7: Illustration of the time-dependant stacked LSTM model.The amount of units in the Input layer depends on the size of the training data-set.

For each epoch[4] the model saves the number of epochs with the lowest root mean squared error (RMSE). The model has an early stopping function, which stops the new epochs if better results are not computed. This is done with the help of two functions in Keras' Callback library, `ModelCheckpoint`, which saves the best model, and `EarlyStopping`, which stops the epochs if better results are not computed [79].

[4]In one epoch, every sample in the data-set has updated the internal model parameters, the epoch is compromised of batches which is the number of samples performed before the model is updated [78]. The number of epochs is usually large, however in every ML task performed, the number of epochs is optimized.

4.1.4 Cycle Dependant ML

Due to the opening and closing process of the valve happening rarely within the time span, another approach where time is removed as a variable is assessed. In this approach, one looks only at the opening or closing cycle for a valve. Models like MLP, simple LSTM, Bidirectional LSTM, Stacked LSTM, and CNN LSTM are tested. Just as in the time dependent LSTM, the number of epochs and the savings, are optimized using the Keras' Callback functions. It is challenging to predict the future with only one variable, however, Blownlee [47], presents a sliding window method in order to predict future values based on previous observations.

Sliding Window Method

The sliding window is the basis for how we can turn any time series data-set into a supervised learning problem [47]. In the sliding window method, new points are predicted based on a given amount of previous points. The data-set is split into a matrix and a vector, as shown in example 4.3-4.4. In the sliding window method, one tries to turn an unsupervised learning[5] to a supervised one. The size of the window defines the size of the matrix. In example 4.3-4.4, a window of the size 10 is chosen.

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\ x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} \\ x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \\ x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} \\ x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix} \begin{bmatrix} x_{16} \end{bmatrix} \tag{4.4}$$

The ten first values give as output the eleventh one. Then, the next ten other values starting from the second one result in the twelfth one and so on, as illustrated in Figure 4.8. This mechanism turns the data-set into a supervised learning model by teaching the model how previous values (in this case the previous 10 values) affect the next one (in the first row, the 11^{th} value). In 4.4, one can see that the last vector predicts the future value. In this case, a batch of 15 values is used and one iterates over each value to predict the 16^{th} one. Various amounts of window sizes are assessed in order to find out which could yield the best results.

[5] Unsupervised learning does not make use of tagged data, however, it tries to sort the data and set probabilities based on pattern density [80].

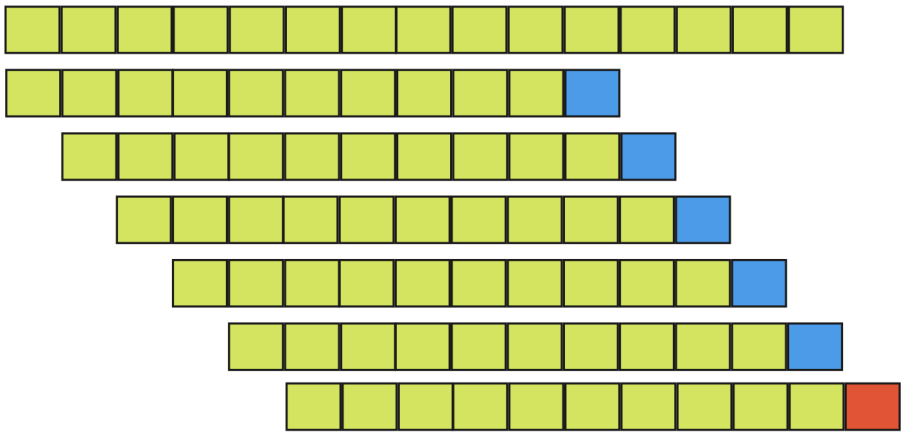


FIGURE 4.8: Illustration of the sliding window method. The first row represents the data-set. The rest of the rows illustrate the split. The green boxes represent the data that goes into the matrix while the blue, the data in the vector. The red box is supposed to be the predicted value.

4.2 Pressure analysis of the servomotor

The servomotors analyzed in this thesis are those which control the guiding vanes in a Francis turbine. However, the operating mechanisms and physical principles can be applied to other types of servomotors like those which open valves or control the amount of water injected into a Pelton turbine.

In the analyzed servomotors, two sensors are placed on each piston in order to continuously measure the pressure from each chamber. The pressure difference is then calculated for each piston. Figure 4.9 illustrates one of the pistons, with sensors at each chamber and piston.

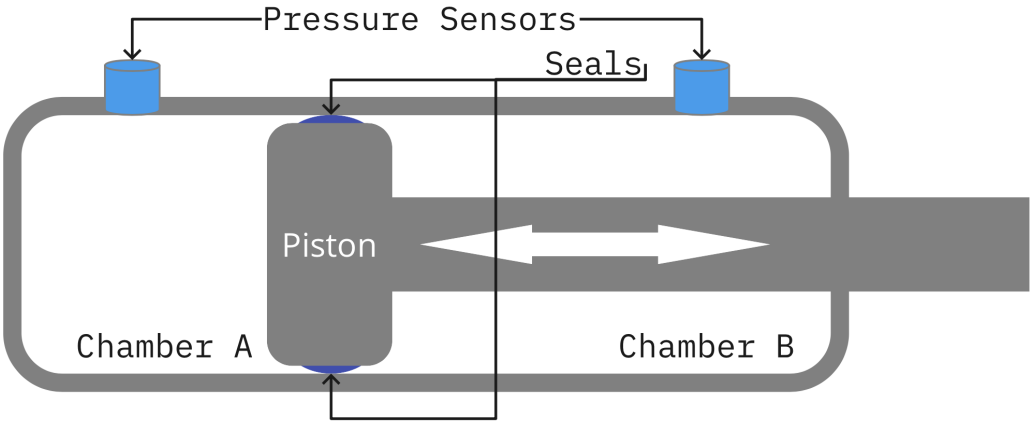


FIGURE 4.9: Simple illustration of the piston in the servomotor.

Two data-setts were provided, from one of Statkraft's suppliers, Andritz, one for the opening servomotor and one for the closing. The data was then merged together in order to plot the pressure difference against the piston position in the servomotor given as a percentage. The data is then processed in a One Class SVM algorithm. This algorithm is similar to the one used in case 9 in the MonitorX project [52]. One Class SVM defines a boundary based on the initial data. The data processing and classification are shown in the flow chart in Figure 4.10.

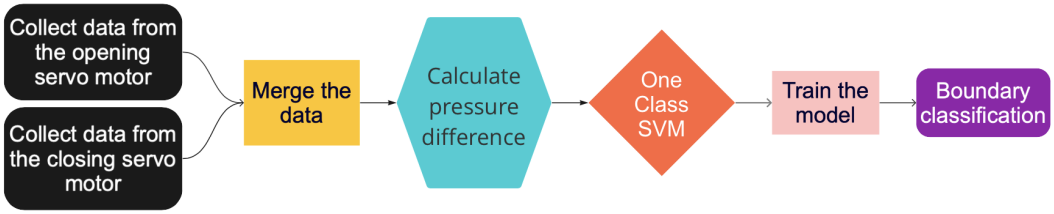


FIGURE 4.10: Flow chart of data processing and classification for data from the servomotor. Pressure is analysed as a function of piston position.

Force can be plotted against piston position as well. From the pressure data the force F can be calculated as well, as shown in the equation below:

$$F = P \times A, \tag{4.5}$$

where P is pressure and A is the area of the piston. The area of the piston in Chamber A can be calculated by its diameter which is 90mm, while for Chamber B the area is smaller due to the piston rod (see Figure 4.9), which has a diameter of 45mm. The area on both sides can easily be calculated as $\pi \times r^2$, where r is the radius. Force is plotted against the piston position in order to compare with the results from pressure. The flow chart in Figure 4.11 illustrates the data processing leading to the One Class SVM computation.

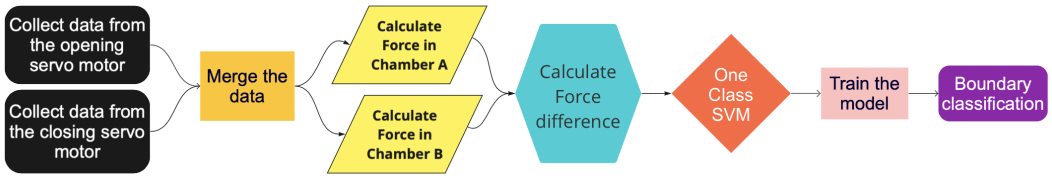


FIGURE 4.11: Flow chart of data processing and classification for data from the servomotor. Force is analysed as a function of piston position.

The One Class SVM is powered by a kernel[6]. In this case, the Gaussian kernel is used, which is one of the most used kernels, better known as RBF kernel [58]. Two variables are important in the One Class SVM algorithm: γ and ν. The γ parameter can be seen as a cut-off parameter for the Gaussian sphere [58]. A small value of γ gives a soft decision boundary of the RBF kernel, while an increase in γ leads to a tighter decision boundary. The ν parameter fine-tunes the trade-off between overfitting and generalization [68]. A low value of ν, yields a larger margin at the cost of training accuracy, on the other hand, a high value of ν, has a more narrow margin. Both γ and ν can have values between 0 and 1.

A condition monitoring key performance indicator (KPI) is the desired output of this task. Condition monitoring is an important part of maintenance activities and is intended to measure the parameters of the actual physical state of an item [56]. For this task, the most relevant KPI is the signed distance between a new measurement and the decision boundary established by One class SVM. This will be visualized by

[6]A kernel can be interpreted as a similarity function, where distance measure is inverted into a similarity score which is placed in a range from 0 to 1 [58].

the use of bullet graphs and can be integrated into the existing monitoring system or as a Functional Mock-up Unit (FMU)[7].

4.2.1 Validation

There are 18152 rows of data in the servomotor data-set. The data set is split into a training data-set and a validation data-set. 17152 data points are used to train the One Class SVM model and 1000 random data points in the data-set are used to test/validate the model. The reason random data points were chosen instead of consecutive ones is to generalize the model by having more representative test data than consecutive data points. The errors in the training and the validation data-set are presented in the figure output.

[7]The Functional Mock-up Interface is a free standard that allows the exchange of dynamic models. The Functional Mock-up Unit is a container that holds a unique model [81]

Chapter 5

Results and Discussion

In this chapter, the results of the analyzed data will be presented and discussed. Challenges in using the already acquired data will be introduced. As a result of these challenges, a framework for data gathering will be presented, including the implementation of other sensors that could give more information on the state of health of the various equipment. This information combined with various Machine Learning techniques could be essential for Predictive Maintenance of various components of a hydropower plant. The code and models used during this thesis are uploaded on GitHub and can be accessed via this link: https://github.com/ernesc/masters_thesis_2022.

5.1 Valve

One of this thesis' tasks had the aim to analyze if the time-lag for the opening and closing process could be used to predict when maintenance should be done in the valves and to see if there is any pattern in the data that suggests that the time-lag increases with the aging of the valve. The data was downloaded in unique files for each year and data from different sensors was in the same file. A function was therefore defined in order to extract and merge the data in order to make it easier to use (as shown in Figure 4.2 in Chapter 4). Figure 5.1 below shows the opening and closing time-lag for the three valves assessed at Nore 1 HPP. There are in total eight valves at Nore 1 HPP.

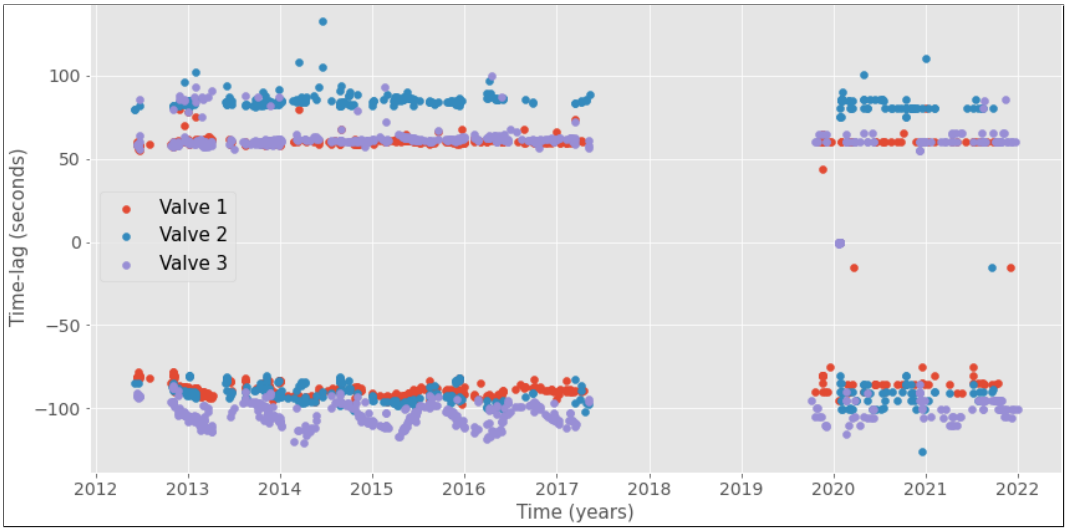


FIGURE 5.1: Data gathered from Nore 1 for the three valves. Data is visualized as a scatter plot in order to show that these values are unique and not continuous.

Time, in years, is plotted against the horizontal axis, while time-lag, in seconds, is plotted against the vertical axis. The data is visualized as a scatter plot since the values are periodical and not continuous. As shown in Figure 5.1, there are missing data from the beginning of 2017 until the end of 2019. It is uncertain the reason for this lack of data. In Figure 5.1, one can also see some outliers. Some outliers are due to sensor malfunction, others for reasons unknown. The outliers were removed before model training in order to increase the accuracy of the prediction. The closing process of valve 3 appears to have a seasonal variation during the years. The reason for this variation is still unclear, even after an inspection of the valve at the power plant with the operators. All valve types showed this type of seasonality as shown in Figure 5.2, however, the other valves do not show the same sensitivity as valve 3. In Figure 5.2, data is visualized as a line graph in order to make it easier to notice patterns and/or seasonality. One of the reasons for the seasonal sensitivity might be the water temperature which can change the viscosity of the water, thus making it harder for the counter weight to close the valve.

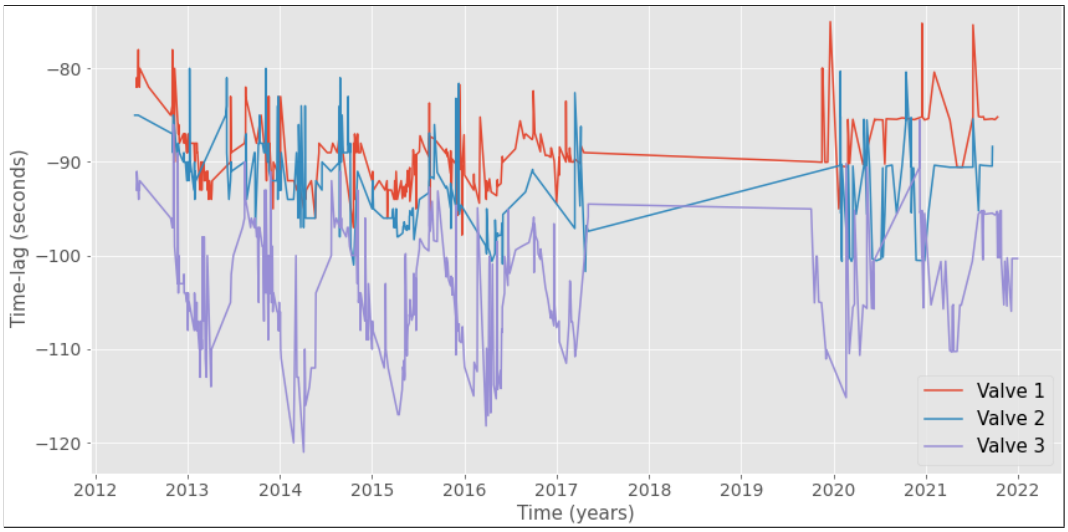


FIGURE 5.2: A closer look at the closing time-lag of the valves. A seasonality is noticed on the data. Data is visualised as a line graph in order to notice the seasonality in the data. Data is not continuous but periodical.

As mentioned earlier, each of the data-sets were tested in various time series forecasting methods, with both simple statistical methods like SARIMA as well as ML algorithms such as LSTM, MLP, and a combination of LSTM with CNN. In the following sections, the results will be presented, as well as discuss the strengths and weaknesses of each model.

5.1.1 Time dependant models

SARIMA is one of the most used tools for time series prediction, it was therefore used as a base case for the time dependant model. The SARIMA model seems to have a nice prediction and is on the same scale as the validation data, as we can see in Figure 5.3. SARIMA captures, to a certain degree, some of the patterns and seasonal variations. The root mean squared error (RMSE) obtained using SARIMA is computed to 0.838, which is good.

As mentioned in earlier chapters, LSTM is considered to be one of the better time series prediction algorithms. However as is shown in Figure 5.4, the prediction is of

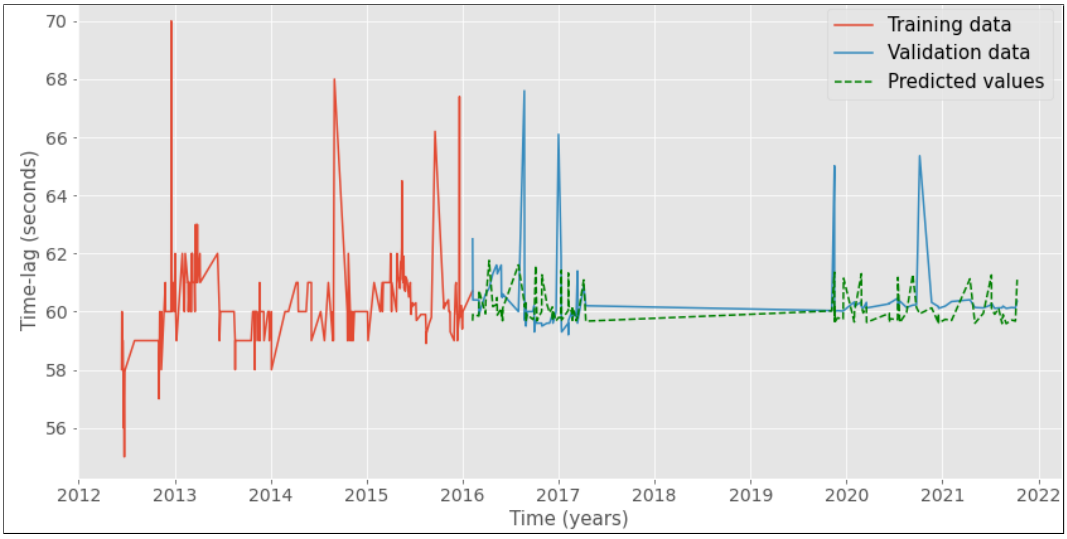


FIGURE 5.3: Results from SARIMA for the opening time-lag of valve 1. The computed RMSE is 0.838.

the same order however the prediction does not seem to capture the pattern from previous observations. The accuracy can be seen in the RMSE which is significantly higher than the one obtained with the SARIMA model, 1.340.

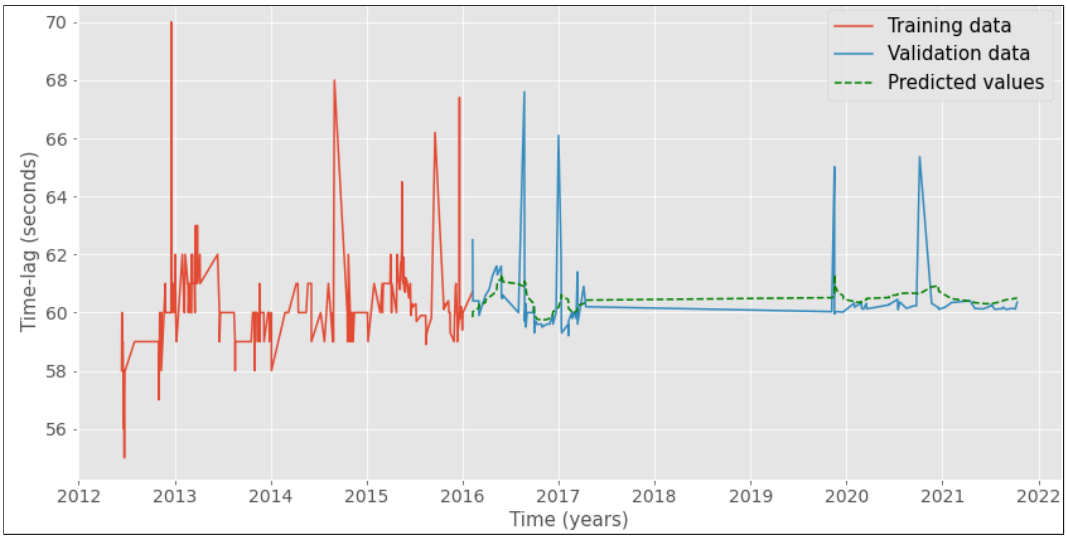


FIGURE 5.4: Results from time dependant LSTM for the opening time-lag of valve 1. The computed RMSE is 1.340.

Machine Learning (ML) algorithms perform better on large amounts of data. 200 to 300 data points are too few to be able to make an acceptable prediction using ML models, therefore simple statistical models like SARIMA perform better than ML algorithms such as LSTM. However, during the closing process, SARIMA did not perform quite as well as we can see in the Figure 5.5. The RMSE value increased to 3.874, which is much higher than the one for the opening process data-set. In the time-span from 2020 to the end of 2021 the difference from predicted values to the real ones increases much more than for the first predictions, from 2016 to 2017, as shown in Figure 5.5. This illustrates that The ARIMA model is not able to make reliable predictions when the data has experiences too many fluctuations.

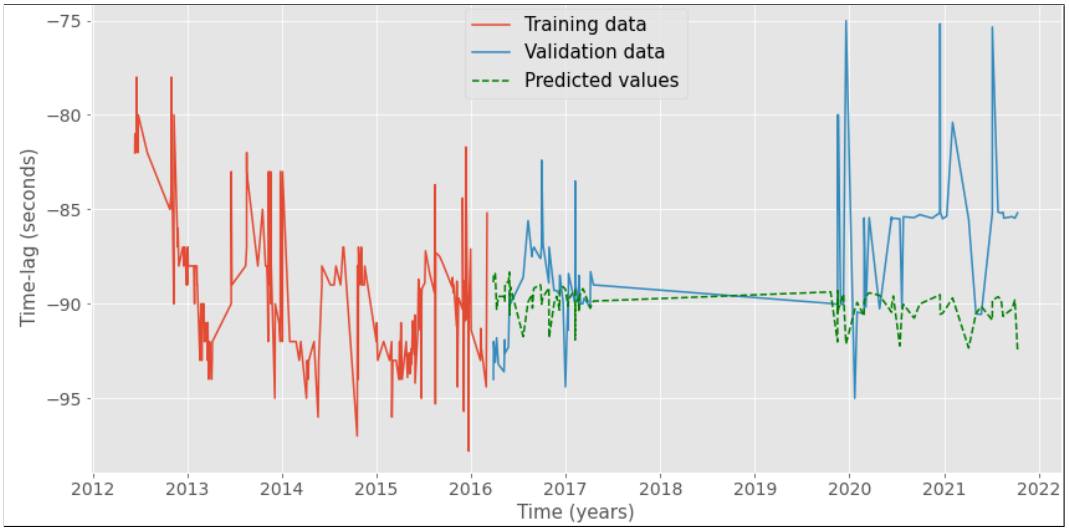


FIGURE 5.5: Results from SARIMA for the closing time-lag of valve 1. The computed RMSE is 3.874.

An increase in RMSE was noticed for the LSTM model as well. However, as is shown in Figure 5.6, the prediction fits quite well with the validation data. This shows that LSTM models are more robust to changes than statistical models like SARIMA. Even-though SARIMA had a low RMSE during the opening process, during the closing process the data had much more variation than in the opening process thus leading to a worse prediction.

LSTM on the other hand seems to generalize the data better and manage to capture the pattern nicely. The RMSE seems to be more stable with both predictions compared to SARMIA, where the RMSE increases by 4.6 times compared to the RMSE for the opening process. LSTM on the other hand experiences an increase of 2.7 of the RMSE for the closing time-lag compared to the opening time-lag, however, the prediction fits quite nice, as shown in Figure 5.6, compared to SARIMA, as shown in Figure 5.5.

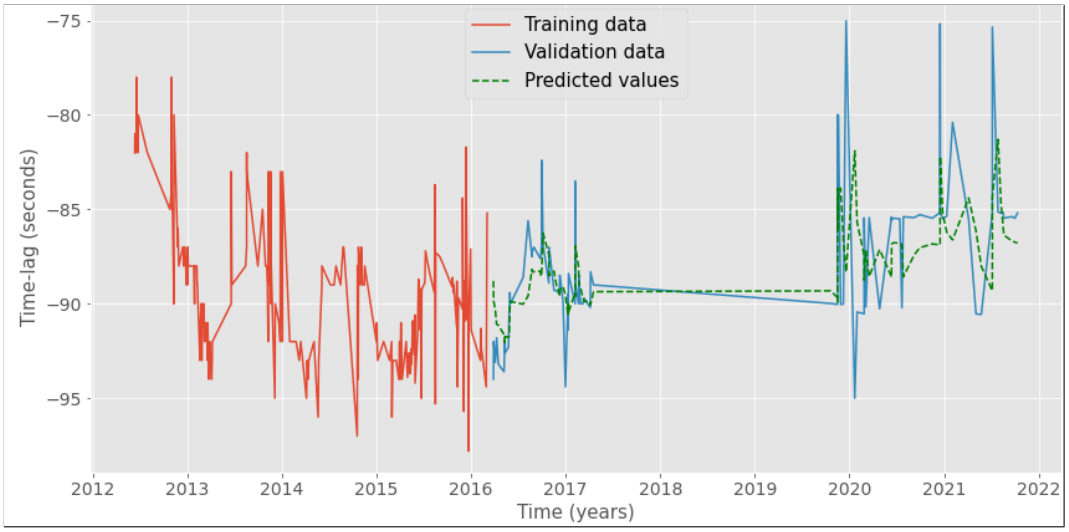


FIGURE 5.6: Results from time dependant LSTM for the opening time-lag of valve 1. The computed RMSE is 3.680.

Since these results were not satisfying, it was proposed to tackle the problem in another angle as it will be shown in the next sub-section.

5.1.2 Cycle dependant models

Brownlee [47], overlooks time as a variable and focuses only on the variable, in this case, opening and closing time-lag of the valve. This way the prediction is computed for the next cycle instead of the next time the valve opens or closes. As earlier mentioned, the Sliding Window Method transforms the data-set into a supervised learning data-set. This way, each window (set of data from the training data) returns the next value. After the model has iterated over all the training data-set, the other predictions are made based on previously predicted values. For each case, several window sizes are tested in order to test the accuracy at different window sizes. As shown in Section 5.1, the closing time-lag experiences seasonal variation so it is important that the window size captures this variation, since time is stripped as a variable.

A simple MLP model was first assessed in order to check the results with the time dependent SARIMA and LSTM models. This simple MLP model was used as a base case for the other tests where time dependency is taken out and we look only at the number of cycles. The MLP seems to be very smooth compared to the previous results and has a tendency to increase, as shown in Figure 5.7. Various MLP units were tested, even with multiple hidden layers, however, the predictions had worse results than the results presented in this thesis. Various window sizes were also analyzed but had little influence on the MLP results.

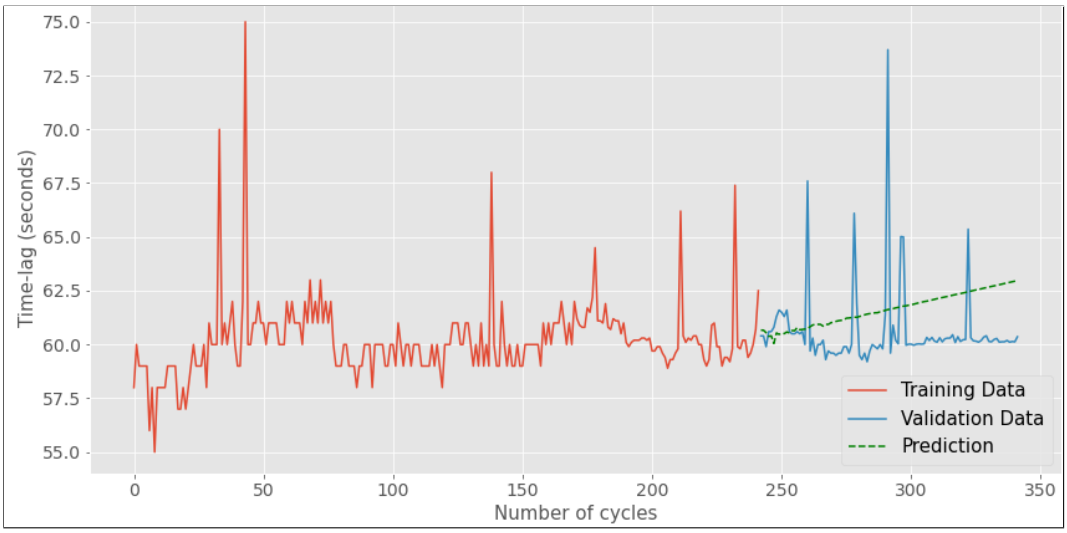


FIGURE 5.7: Results from MLP for the time-lag of the opening time of valve 1, with RMSE of 2.315 and sliding window size of 25. 75 LSTM units in the hidden layer.

For the closing process, a more than doubling of RMSE was noticed on the MLP, compared to the opening process. The number of units in the hidden layer is increased in order to give a better prediction. The opening process worked better with a relatively small amount of units than the closing process, probably due to little variation in the data during the opening process. The prediction seems to produce a sinusoidal-like curve for the first 50 predicted values, with a high error (see Figure 5.8). For the last 50 predicted values, the curve flattens out and represents more of a mean than an actual prediction, as shown in Figure 5.8.

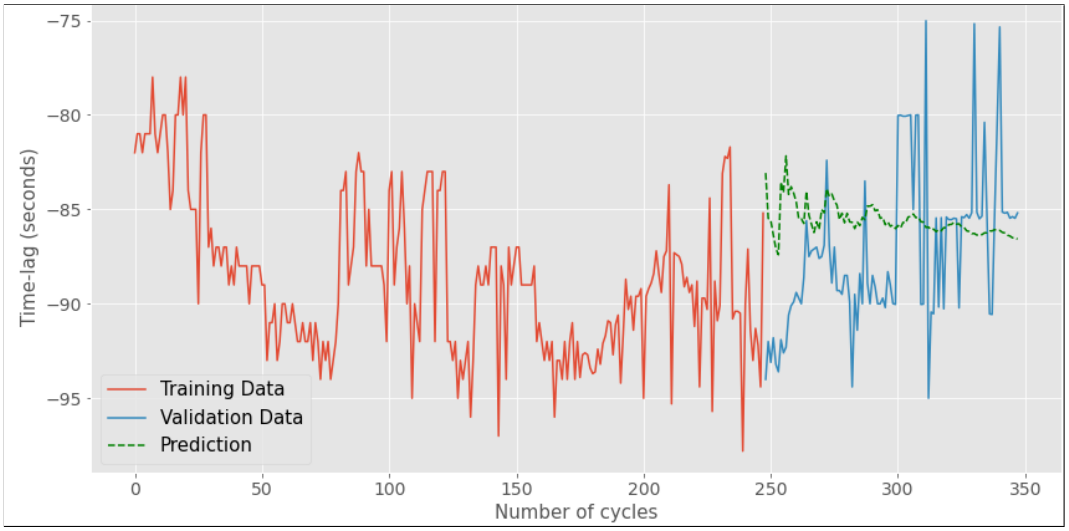


FIGURE 5.8: Results from MLP for the time-lag of the closing time of valve 1, with RMSE of 4.697 and sliding window size of 25. 150 LSTM units in the hidden layer.

A simple LSTM model without time dependency has a RMSE of 2.182. The model seems to have the same pattern of the training data but not the validation data. Figure 5.9, shows that the prediction converges with the validation data by the end, however, experiences a spread during most of the prediction.

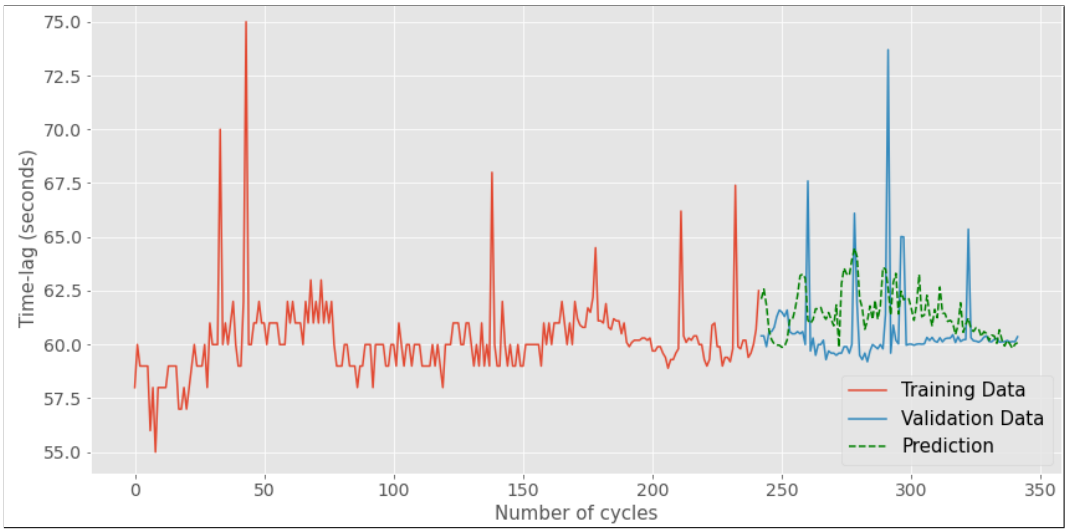


FIGURE 5.9: Results from simple LSTM for the time-lag of the opening time of valve 1, with RMSE of 2.182 and sliding window size of 30. 150 LSTM units in the hidden layer.

The results from the opening process of these two models show that the data is insufficient to provide an acceptable prediction for future values of the opening time-lag of the cycles.

The simple LSTM outperforms MLPs prediction with a lower RMSE of 4.489, as shown in Figure 5.10. The first 4-7 values have a high error, however, this is noticed in both time dependant models (see Figures 5.5 and 5.6) as well as the cycle dependant MLP model in Figure 5.8. The next values, up to the 300^{th} cycle are well predicted, but after that the prediction looks more like a moving average (see Figure 5.10).

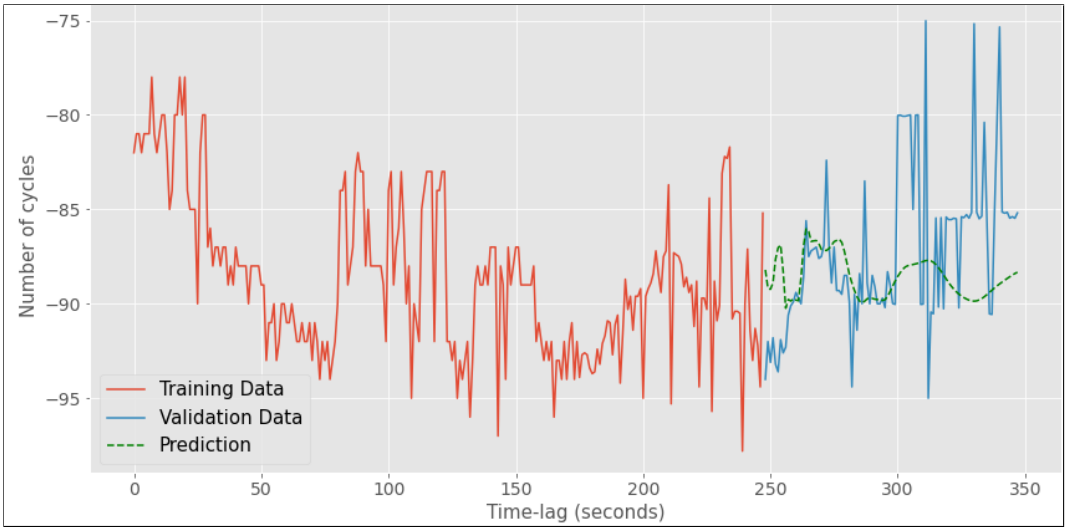


FIGURE 5.10: Results from simple LSTM for the time-lag of the closing time of valve 1, with RMSE of 4.489 and sliding window size of 25. 50 LSTM units in the hidden layer.

Bidirectional LSTM seems to perform worse than simple LSTM for the opening process. Just like the MLP model the Bidirectional LSTM produces a smoothed line, however in the opposite direction, predicting the time-lag to decrease in future cycles.

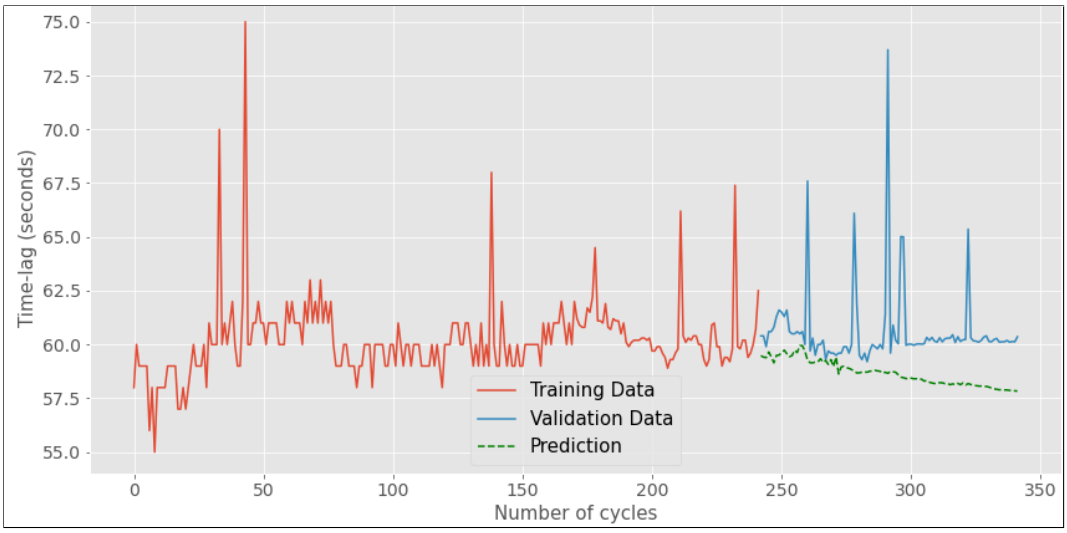


FIGURE 5.11: Results from Bidirectional LSTM for the time-lag of the opening time of valve 1, with RMSE of 2.730 and sliding window size of 50. 100 LSTM units in the hidden layer.

During the closing process, the RMSE increased to 8.111, and as is seen in Figure 5.12 the prediction seems to get worse after a few predictions and only converges with the validation data three times. The prediction shows that the closing time-lag of the valves has the tendency to decrease while the validation data shows some decrease in the closing time-lag.

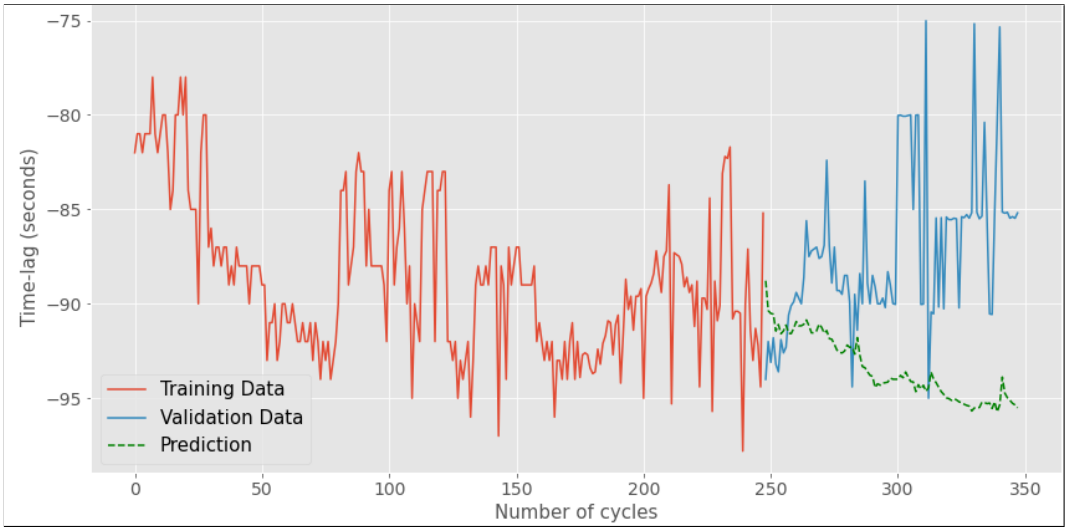


FIGURE 5.12: Results from Bidirectional LSTM for the time-lag of the closing time of valve 1, with RMSE of 8.111 and sliding window size of 50. 100 LSTM units in the hidden layer.

Stacked LSTM 5.13 gave a nice performance i terms of RMSE, however, it seems to have a tendency to fluctuate more than the validation data-set.

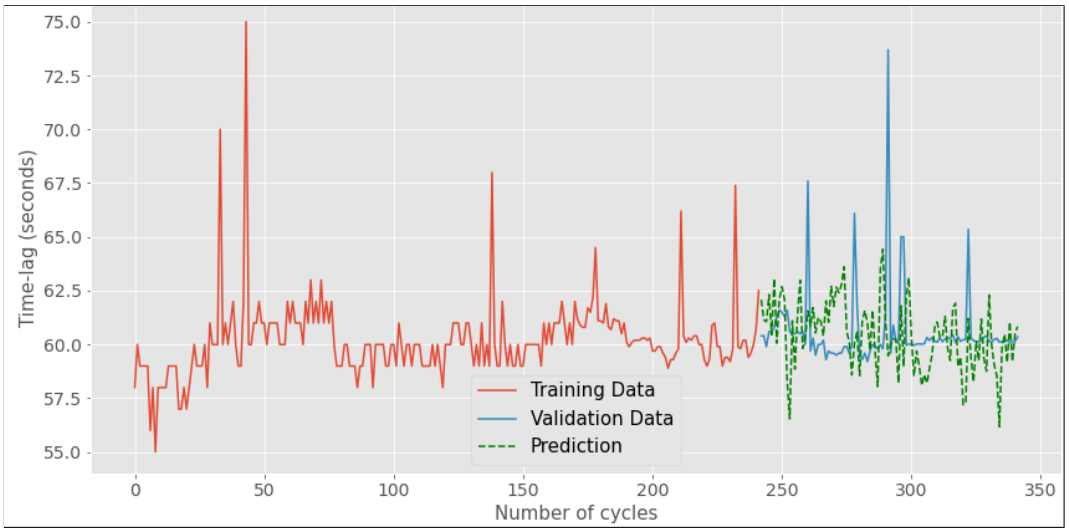


FIGURE 5.13: Results from Stacked LSTM for the time-lag of the opening time of valve 1, with RMSE of 2.494 and sliding window size of 40. Both hidden layers had 50 units each.

Stacked LSTM computes a prediction with a high RMSE for the closing process, just like Bidirectional LSTM. The prediction visualized in Figure 5.14 shows that the error for each prediction is quite high compared to the validation values.

A combination of CNN and LSTM seemed to yield good results. The model seemed to be trained very fast compared to the other cycle dependent models. With few units in a hidden layer, it took a small number of epochs to train the model, however, when a large number of units was applied, the model needed a larger number of epochs to train. the RMSE does not change much compared with the other results, however, the prediction looks smoothed for valve 1 as shown in Figure 5.15. The

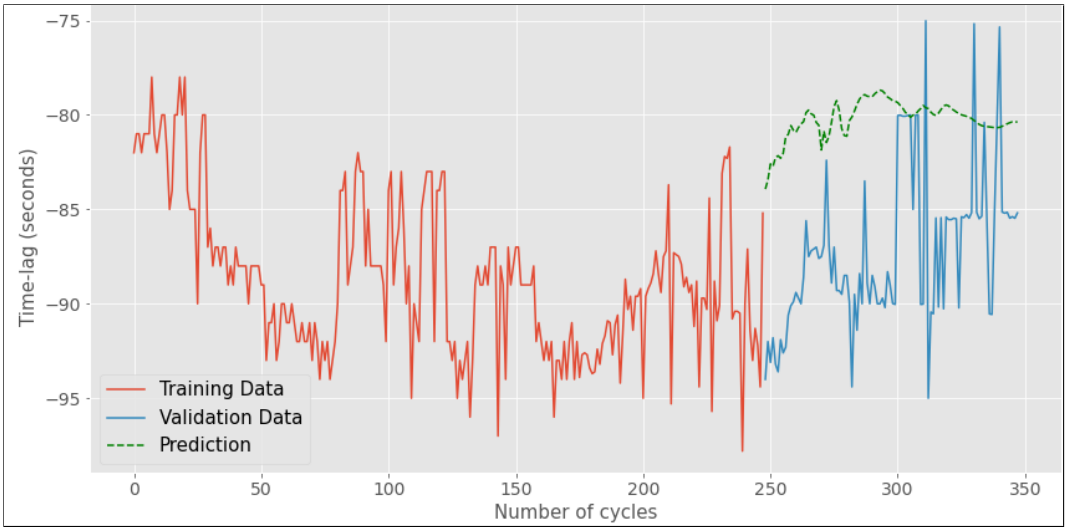


FIGURE 5.14: Results from Stacked LSTM for the time-lag of the closing time of valve 1, with RMSE of 8.039 and sliding window size of 50. Both hidden layers had 50 units each.

spread between the predicted values and the validation values seems to get smaller in the last 50 values.

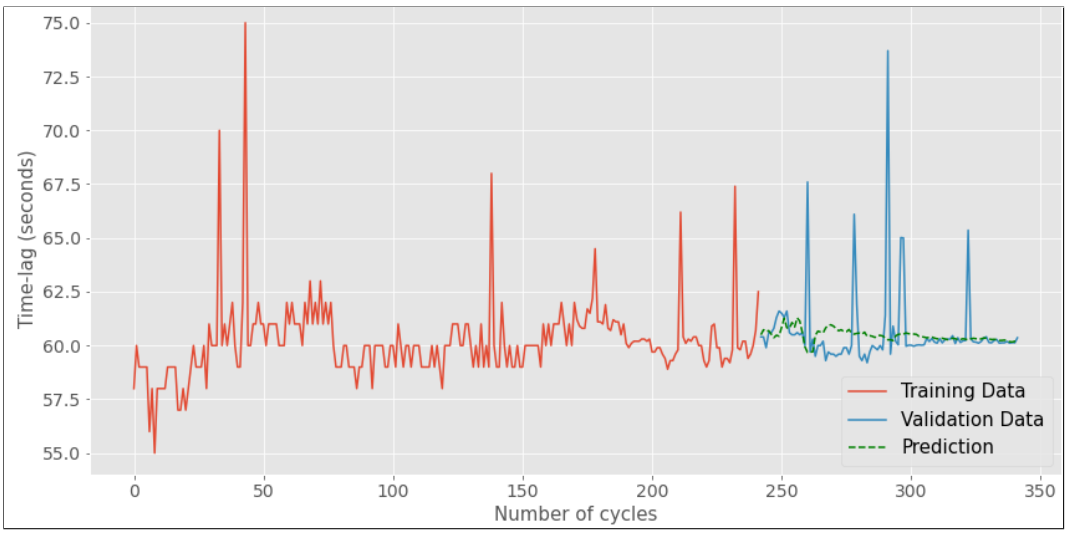


FIGURE 5.15: Results from CNN-LSTM for the time-lag of the opening time of valve 1, with RMSE of 1.939 and sliding window size of 40.

A smoothed line can be noticed in the closing process as well, however, the trend seems to be accurate as is shown in Figure 5.16. A doubling in the RMSE is also noticed.

Parameters such as window size, amount of units in a hidden layer, learning rate, and batch size could be optimized by performing a grid search. This was outside of the scope of this thesis due to time limitations, as well as a lack of any pattern for maintenance that could be proven useful.

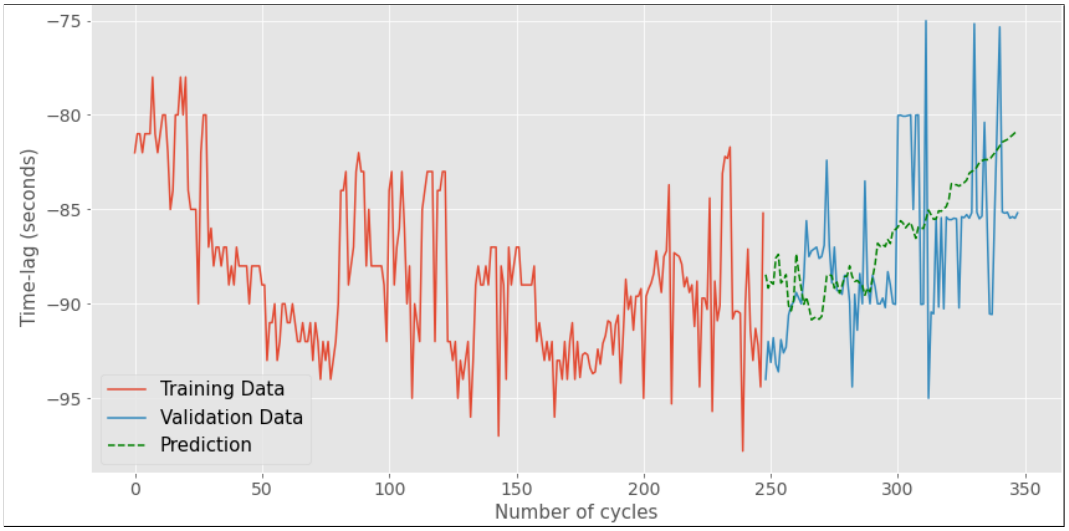


FIGURE 5.16: Results from CNN-LSTM for the time-lag of the closing time of valve 1, with RMSE of 3.982 and sliding window size of 40.

Table 5.1 presents results from all predictive models assessed. There are other models that due to time limitations were not performed. These methods vary from other simpler statistical methods to deep learning models. For example, LSTM could be combined with other ML models such as Autoencoder, as presented in [44]. Other common predictors are Naive Bayes models or even linear regression.

TABLE 5.1: Summary of the various models. *O* stands for the opening process and *C* for the closing process. The first two models are the time dependant ones, while the last five are the cycle dependant ones.

Model	RMSE	Comments
SARIMA	O: 0.838 C: 3.874	Performs well in terms of RMSE for the opening process, however experiences a lot of unnecessary fluctuations. Does not follow the pattern of the validation data.
LSTM	O: 1.340 C: 3.680	The opening process produces a smoothed prediction that does not capture all the fluctuations. The closing process prediction performs very good and replicates the pattern.
MLP	O: 2.315 C: 4.697	Even though it has a low RMSE the prediction looks more like a mean than an actual prediction. Experiences a doubling of RMSE during the closing process.
LSTM	O: 2.182 C: 4.489	Performs quite well during the opening process, even though some deviation is noticed. Performs good in some cases during the closing process.
Bidirectional LSTM	O: 2.730 C: 8.111	During the opening process the prediction is quite flattened and has a tendency to drop. An increase in the closing time lag is predicted unlike validation data.
Stacked LSTM	O: 2.494 C: 8.039	During the opening process the prediction fluctuates more than the validation data, but during the closing process the prediction is far off the validation data.
CNN-LSTM	O: 1.939 C: 3.982	Performs somewhat better than the other cycle dependant models, with lower RMSE and has the same tendency as the validation data.

5.2 Servomotor

The data acquired from Statkraft's suppliers was not enough to assess the possibility of Machine Learning for Predictive Maintenance, however, a condition monitoring system was developed for pressure analysis in the pistons.

After the two data-sets are uploaded, the needed data is extracted and merged together in a new table, as illustrated in Chapter 4 in Figure 4.10. Pressure differences are plotted against the position of the servomotor piston and given as a percentage, as shown in Figure 5.17.

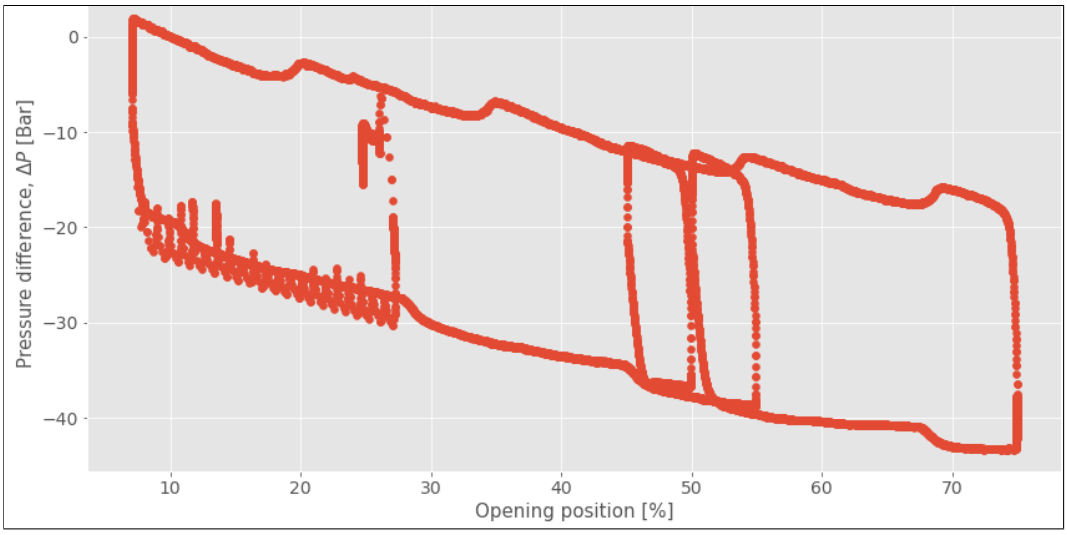


FIGURE 5.17: Pressure difference plotted against the position of the piston for both Servomotors.

One Class SVM is used in order to define a boundary from the data. Generic values were first chosen for the γ and ν. The results are shown in Figure 5.18.

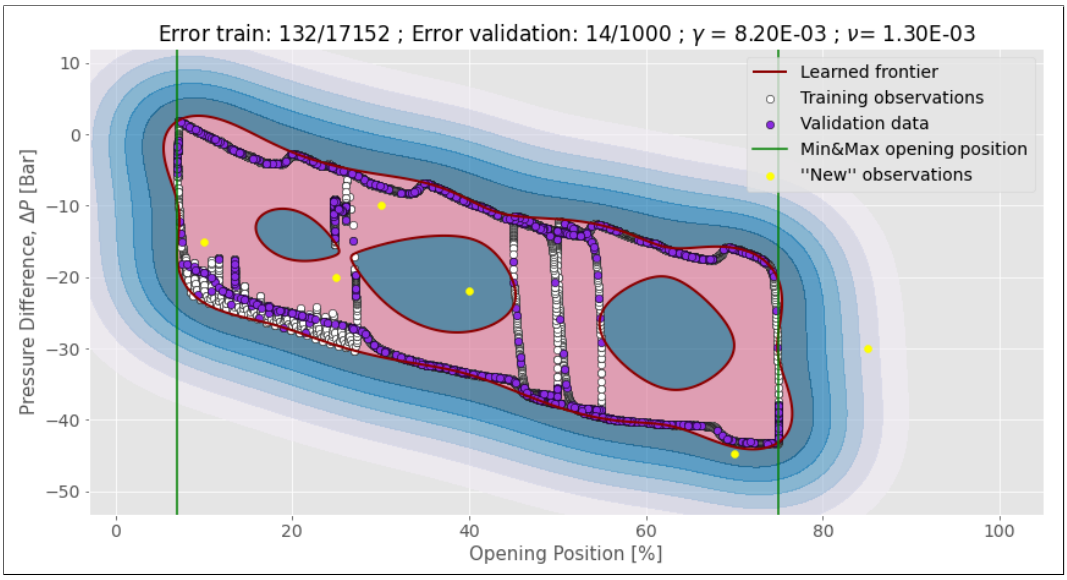


FIGURE 5.18: Preliminary results from the One Class SVM classification for pressure difference. The shades of blue are a visual aid in order to make the length of new points from decision boundary easier for the reader to notice. The size of these shapes is made arbitrarily and can be changed accordingly.

Values within the envelope are allowed as well, however, the One Class SVM presented in Figure 5.18 overlooks these values (as shown in the yellow point at the 40% opening position), therefore a new boundary must be computed which takes into account values within the envelope. Other parameters are thus chosen for γ and ν in order to achieve better results. Various parameters were tested. The ones which yielded the best results are $\gamma = 2.6 \times 10^{-3}$ and $\nu = 1.3 \times 10^{-4}$, as shown in Figure 5.19.

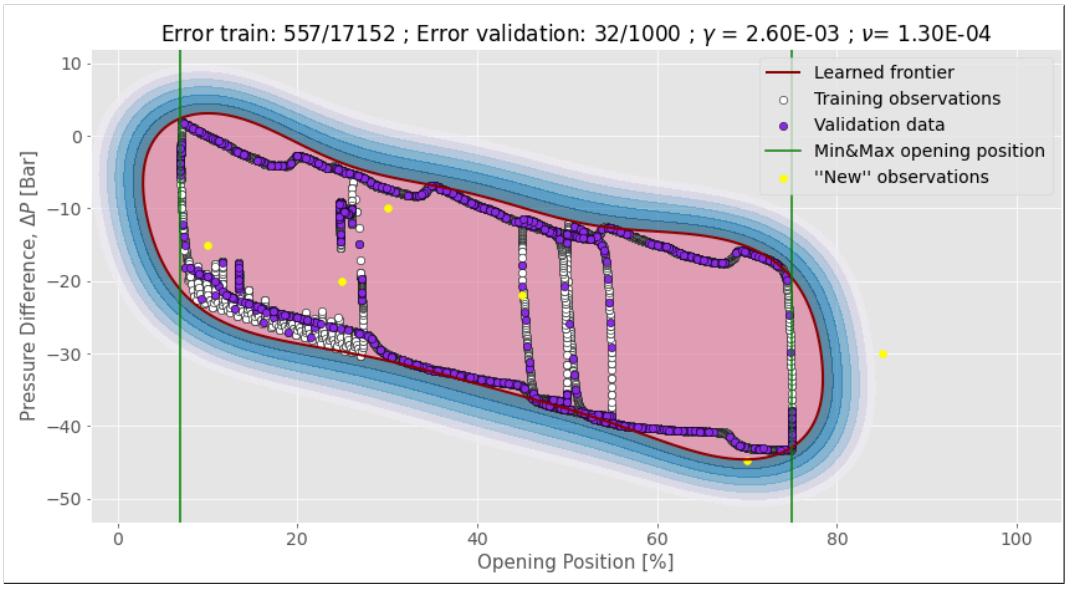


FIGURE 5.19: Desired results from One Class SVM classification for pressure difference.

The amount of data collected for this analysis is unfortunately too small to compute a better boundary for the envelope, with fewer errors in both training and validation data-set. The data analyzed is gathered during nominal operating conditions. In order to have a better understanding of the operation of the servomotor, more data on various operating conditions should be collected. One Class SVM gives a measure of the length a new value is from the computed boundary. In an HPP, one can do tests and decide the allowed outer values and define the scales of how far off can be allowed for new values. The green boundaries in Figure 5.19, represent the position boundary of the piston in the servomotor during nominal operating conditions. Any point that exceeds those green boundaries should be seen as an immediate error since this type of error is very unexpected, during nominal conditions.

A bullet chart (see Figure 5.20) is designed in order to control how far off new measurements are from the decision boundary computed by the One Class SVM. The bullet chart can be used in order to act as a KPI for the servomotor, and incorporated to the dashboards in order to monitor the condition of the servomotor. Various examples of some "new" fake points are produce in order to illustrate how the bullet chart works as shown in figures 5.20–5.23.

After some time of logging new data, the data can be assessed and analyzed in order to see if more outliers are appearing in the data. When an increasing amount of outliers is detected, might suggest that there is something wrong with the servomotor, thus maintenance should be considered.

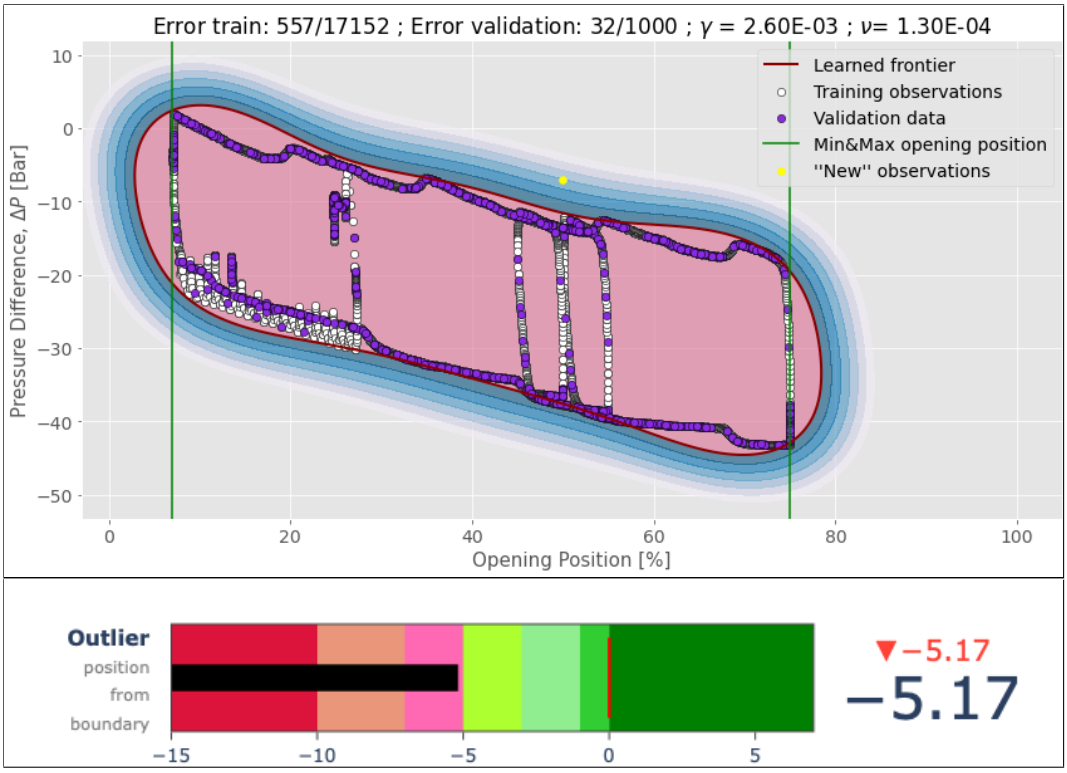


FIGURE 5.20: One Class SVM classification for pressure difference with KPI visualised as a bullet chart, when new value is outside of the boundary.

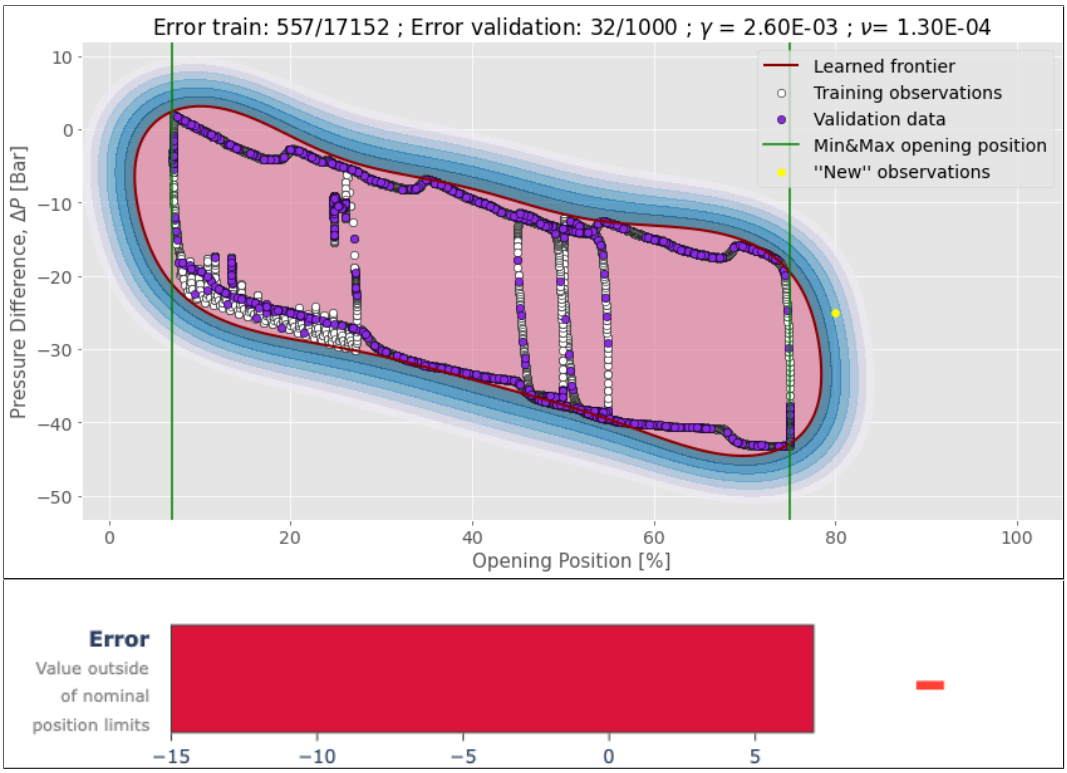


FIGURE 5.21: One Class SVM classification for pressure difference with KPI visualised as a bullet chart, when new value is out of nominal position boundary.

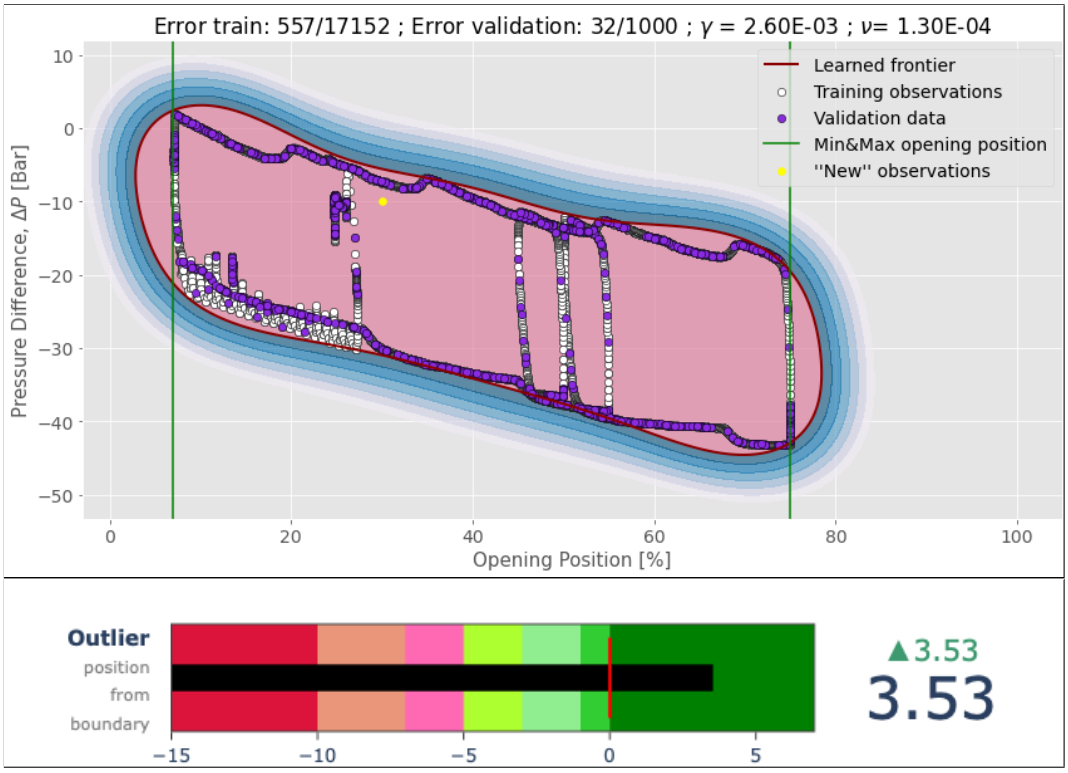


FIGURE 5.22: One Class SVM classification for pressure difference with KPI visualised as a bullet chart, when new value is within the learned frontier.

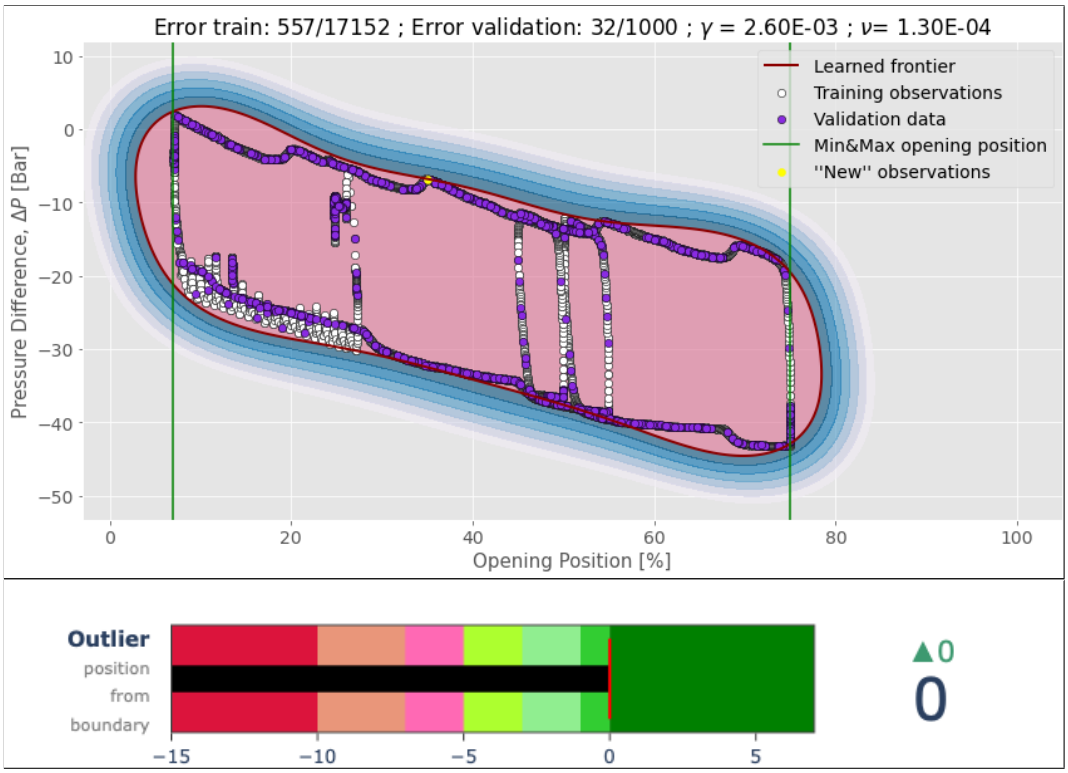


FIGURE 5.23: One Class SVM classification for pressure difference with KPI visualised as a bullet chart, when new value is overlapping with the learned frontier.

After a discussion with Thor Arne Hvam Bruun, Head Engineer at Andritz Hydro, the idea of plotting force instead of pressure was explored. Force is easy to calculate from pressure when one has knowledge of the specific servomotor. If one knows the area of the piston of both sides of the chamber, equation 4.5 can be used to calculate the force. A One Class SVM is performed again with γ and ν parameters and the validation results are much better compared to the previous embodiment. The training error decreases to 303 values compared to 557 where pressure is plotted against piston position. The validation error decreases as well to 16. As shown in Figure 5.24, the data presented as a function of force is more linear than the one presented as a function of pressure, therefore the One Class SVM fits more nicely. The boundary is fitted better to the data and as one can see in Figure 5.24, there is less need for the nominal operating boundaries (green lines) than the model where pressure is used (see Figures 5.19–5.23). There is, however, some divergence from the data noticed on the edges of the envelope. This is expected since pressure on one side is experienced on a bigger area than the other due to the piston cylinder which reduces the area. This issue is taken into account when force is calculated from pressure.

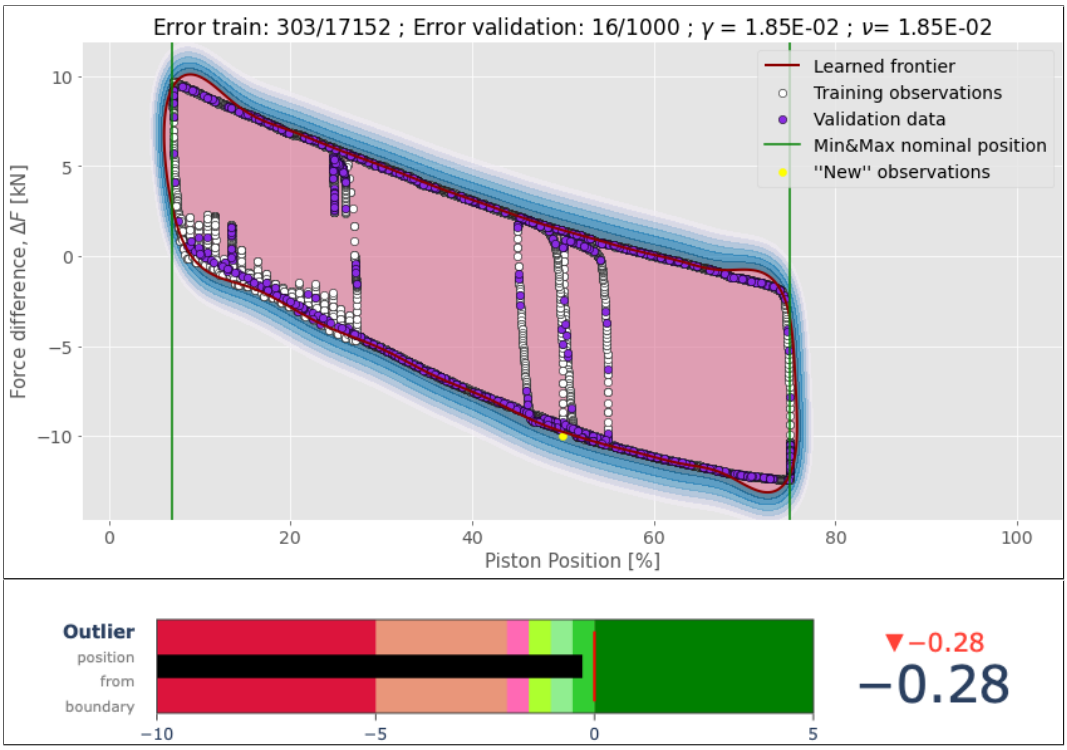


FIGURE 5.24: One Class SVM classification for force difference with KPI visualised as a bullet chart.

The calculating force from pressure might seem that makes the model less applicable to other servomotors, however, this is easily done and the computation takes little time. It could be somewhat challenging for operators to get used to working with force instead of pressure since most of the data being presented in various equipment is pressure data.

5.3 A Framework for data collection that enables the use of Machine Learning for Predictive Maintenance

This section will focus mainly on valves and servomotors since these two systems are the main focus of this thesis. However, these principles can be applied to other machinery in an HPP.

In order to achieve reliable results from Machine Learning algorithms, it is necessary to collect and store more data. On a tour at Nore 1 HPP, one can see a huge amount of pressure sensors around various parts of the power plant. However, these sensors are analog and are only consulted in the case of error detection. Digitalization of these measurements needs to be in place.

5.3.1 Valves

As mentioned in section 5.1, the data collected so far by most power plant operators is the closing and opening time. As is seen by the results, it is hard to predict the future with univariate data.

The most interesting measurements need to be taken during the opening and closing process of the valve. As shown by CFD analysis presented in Azad et al. [15], vortices, and pressure pulsations are observed during the opening and closing process, thus monitoring vibrations in several directions is important to understand what happens during this process and how this can affect the valves lifetime. Zhang et al. have installed several vibration, pressure and displacement sensors in spherical valves to monitor the effects of the opening and closing process [16]. Even though the research presented in Zhang et al. [16] is within a spherical valve, the same measurements can be performed in butterfly valves. During a presentation of the power plant at Nore 1, it was shown that there is some displacement of the valve housing during the opening and closing process. Zhang et al. [16], have installed displacement sensors on spherical valves as well.

In order to have enough data to correlate with failure, the opening time is not enough, therefore more sensors should be installed. These sensors could be vibration sensors installed in various positions (vertical and/or horizontal perpendicular and parallel to the water flow as well as displacement sensors. Digitalization of the current pressure monitors is necessary in order to monitor the pressure accurately during the opening and closing process and store the data. The valve disc angle should be monitored and logged in order to correlate the data with the position of the disc. The desired data-set for ML in order to predict failure and/or maintenance should look like the table in Figure 5.25.

	Displacement	Press_downstream	Date_Time	Valve_id	Failure_type	Press_upstream	Vertical_vib	Disc_Angle	Horizontal_vib_perp	Horisontal_vib_par	Maint_type
0	83	11	42	Nore1/Valve1	84	20	87	55	1	12	6
1	18	59	77	Nore1/Valve1	29	60	31	68	84	99	81
2	66	76	78	Nore1/Valve1	20	17	99	44	41	5	90
3	25	82	43	Nore1/Valve1	57	53	1	72	14	96	74
4	26	65	17	Nore1/Valve1	50	80	8	66	88	20	46

FIGURE 5.25: A "screenshot" of how a table with the required data for the use of Machine Learning should look like. The values are random and for illustrative purposes.

So to summarize the data that can be logged is presented in Table 5.2, where one can see the column description from Figure 5.25 in Table 5.2.

Table 5.2: Column description of the table in Figure 5.25 with required information for the use of Machine Learning.

Column name	Description
Valve-id	Standardized name of the valve
Date-Time	Date and time of the measurement
Press-Upstream	Pressure upstream from the valve
Press-Downstream	Pressure downstream from the valve
Disc-Angle	Rotational angle of the valve disc
Horizontal-vib-perp	Horizontal vibration monitoring, perpendicular to the water flow
Horizontal-vib-par	Horizontal vibration monitoring, parallel to the water flow
Vertical-vib	Vibration monitoring on the vertical direction
Displacement	Displacement of the valve housing
Failure-type	Standardised failure as a numerical value
Maint-type	Standardised maintenance type as a numerical value

The failure type and maintenance type, at Nore 1 HPP are being logged in an SAP system with a detailed description of failure or maintenance. The failure and maintenance which is already logged could be used to inspire some categories of maintenance or failure types so that it could be easier to standardize and be applicable for Machine Learning. However, there should be further assessment in this field for standardization of failure types and maintenance types so that one can standardize the input data in a Machine Learning algorithm. It is important that operators whom log failure and/or maintenance are heavily included during this process so that the standard is based on their experience with failure and maintenance as well as their expertise, desires, and wishes since at the end of the day, they are the ones going to log the data. It is hard to standardize something without the input of the people working hands-on with the equipment.

5.3.2 Servomotors

As mentioned earlier, servomotors have a wide range of use in an HPP. The data assessed in section 5.2 is obtained from a servomotor that steers the guiding vanes in a Francis turbine However the condition monitoring mechanism described in section 5.2 could easily be implemented in other types of servomotor like the ones which control the amount of water injected to a Pelton turbine or the ones which steer the valves.

More data could be gathered, however, from the HPU, like the state of health of the accumulators' pressures before and after the accumulators. Vibration sensors could give information on the sealings' state of health. The state of health on the oil pumps could prove to be interesting to assess, as well as continuous, digital, monitoring of the amount of oil in the hydraulic power unit. Table 5.3 lists some important data that could be gathered by the help of some sensors installed on the servomotor.

Table 5.3: Required data for the use of Machine Learning for the servomotor.

Data	Description
Vibration of the accumulator	Might indicate issues in the pressure storage
Vibration of the pump	Indicator of the SoH of the pump
Piston vibration	Might indicate wear of the sealing
Oil monitoring	Digitalization of the amount of oil in the valve could reduce maintenance
Pressure monitoring	Both before and after the accumulator could be interesting

5.4 Summary of results

One of the biggest drawbacks of Machine Learning is the black box explanation. The user has no knowledge of how the model is predicting the future or what is the basis for its predicting maintenance. Using a modular approach, i.e. using Machine Learning for each system gives a better picture of the specific system, and with methods like RENT [82], one can see which of the variables being measured has more of an impact on the maintenance decision. However, this is based on the assumption that one uses multivariate analysis instead of univariate analysis, as was the test case in this thesis.

With the valve use-case, it is shown that it is challenging to get a decent prediction with a limited amount of data as is the case with opening and closing time-lag. The results are also summarized in Table 5.1. The time dependent LSTM performed much better than the cycle dependent. An explanation for these results could be that the opening and closing processes happen so rarely within the year, that is hard to capture the seasonal variations. Therefore, time dependant LSTM, even though it had a relatively high RMSE value, the prediction fitted quite nicely to the data-set.

The servomotor use-case is a good example of how Machine Learning models can be combined with physical principles to produce a nice on-time visualization of the state of health of the servomotor. During discussions with operators at Nore 1 HPP, this kind of visualization, as shown in Figure 5.20, could be helpful to install on all servomotors. However one should work closely with the operators in order to qualify the various gradients, e.g. how far off the boundary one can expect new values and is not an issue and the other way around.

Finally, there is a need for further research where an "excessive" amount of sensors are installed and analysis is done in order to find out which of the information is the most relevant and to weigh this with an economical analysis.

Chapter 6

Conclusion

Machine Learning opens the door to new opportunities to Predictive Maintenance. Today very little data is being collected, making it challenging to predict the future.

Digitalization of the current sensors and the installment of new sensors will exponentially improve maintenance and failure prediction.

For the valve use-case, the opening and closing time-lag data by itself cannot give a maintenance prediction. Further information is needed in order to make good predictions. Data on previous failures and maintenance should be logged and structured according to some international standard in order to be easily assessed, analyzed, and correlated to other sensor information.

Position and pressure values from the sensor data are a user-friendly way to monitor live the health of the servomotors. One Class Support Vector Machine is a powerful tool for deciding the boundary of the values. The bullet chart presented from One Class Support Vector Machine boundary is a nice visualization tool for condition monitoring.

Further work should be done on failure and maintenance standardization in collaboration with the operators. Some of these ideas are presented in Section 5.3. This would make the data more accessible and easy to apply to Machine Learning models.

Bibliography

[1] Bloomberg Green, *Annual Greenhouse Gas Emissions*, https://www.bloomberg.com/graphics/climate-change-data-green/emissions.html, Accessed: 15.01.2022.

[2] Intergovernmental Panel on Climate Change, *Sixth Assessment Report*, https://www.ipcc.ch, Accessed: 15.01.2022.

[3] O. Milman, *António Guterres on the climate crisis: 'We are coming to a point of no return'*, https://www.theguardian.com/environment/2021/jun/11/antonio-guterres-interview-climate-crisis-pandemic-g7, Accessed: 15.01.2022.

[4] International Energy Agency, *Net Zero by 2050*, https://www.iea.org/reports/net-zero-by-2050, Accessed: 15.01.2022.

[5] H. Ritchie and M. Roser, "Energy," *Our World in Data*, 2020, https://ourworldindata.org/energy.

[6] K. Fasol, "A short history of hydropower control," *IEEE Control Systems Magazine*, vol. 22, no. 4, pp. 68–76, 2002. DOI: 10.1109/MCS.2002.1021646.

[7] IEA, "Hydropower special market report," *International Energy Agency*, 2021, Paris, https://www.iea.org/reports/hydropower-special-market-report.

[8] Statnett, "Verdien av regulerbar vannkraft," *OED Bistand*, Norwegian, https://www.statnett.no/contentassets/b82dcf206acc4762b2abcc3182e5bc52/verdien-av-regulerbar-vannkraft-statnett-mars-2021.pdf.

[9] Norwegian Directory for Water and Power, *Reinvesteringsbehov, opprusting og utvidelse*, Norwegian, https://www.nve.no/energi/energisystem/vannkraft/reinvesteringsbehov-opprusting-og-utvidelse/, Accessed: 25.01.2022.

[10] K. H. Karstensen, C. J. Engelsen, and P. K. Saha, "Circular economy initiatives in norway," in *Circular Economy: Global Perspective*, S. K. Ghosh, Ed. Singapore: Springer Singapore, 2020, pp. 299–316, ISBN: 978-981-15-1052-6. DOI: 10.1007/978-981-15-1052-6_16. [Online]. Available: https://doi.org/10.1007/978-981-15-1052-6_16.

[11] B. Holter, *The Internet of Things*. Sintef, Assessed 15.01.2022, https://www.sintef.no/felles-fagomrade/tingenes-internett/.

[12] A. Kjølle, "Hydropower in Norway: Mechanical equipment," in *Proceedings of the IAHR Congress*, 2001.

[13] Crane Co, *Flow of Fluids Through Valves, Fittings, and Pipe*, ser. Crane Company Technical paper no. 410-C. Crane Company, 1988. [Online]. Available: https://books.google.no/books?id=-fdRAAAAMAAJ.

[14] A. Shrivastava and A. Patel, "Structural design & fem analysis of large butterfly valve," *International Research Journal of Engineering & Applied Sciences, IRJEAS www. irjeas. org, ISSN (O)*, pp. 2322–0821, 2017.

[15] A. Azad, D. Baranwal, R. Arya, and N. Diwakar, "Flow analysis of butterfly valve using cfd," *International Journal Of Modern Engineering Research*, vol. 4, no. 6, pp. 50–56, 2014.

[16] F. Zhang, J. Fu, D. Liu, and S. Wang, "Spherical valve stability during hydrodynamic closure process," in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 304, 2019, p. 022 003.

[17] J. Junginger, A. Ruprecht, S. Riedelbauch, S Kolb, and S Vogel, "Investigation of the special behaviour of delayed load rejections on a 3x150 mw pump turbine power station," Oct. 2017.

[18] H.-q. Fang and Z.-y. Shen, "Modeling and simulation of hydraulic transients for hydropower plants," in *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific*, IEEE, 2005, pp. 1–4.

[19] MATLAB, *Predictive Maintenance Toolbox*. Help Center, https://se.mathworks.com/help/predmaint/, Accessed: 02.03.2022.

[20] International Renewable Energy Agency, *Hydropower*, 3/5. Renewable Energy Technologies: Costs Analysis series, vol. 1, https://www.iea.org/reports/renewables-2021.

[21] J. Heggset, E. Solvang, D. E. Nordgård, and A. O. Eggen, *Decision Support for Maintenance and Refurbishment Planning for Hydropower Plants*. Sintef, 2004, https://www.sintef.no/globalassets/project/beslutningsstotte_vannkraft/euromaintenance_2004.pdf.

[22] W. Musial and B. Ram, *Large-Scale Offshore Wind Power in the United States 2010*. NREL, https://www.nrel.gov/docs/fy10osti/40745.pdf.

[23] M Bevilacqua and M Braglia, "The analytic hierarchy process applied to maintenance strategy selection," *Reliability Engineering System Safety*, vol. 70, no. 1, pp. 71–83, 2000, ISSN: 0951-8320. DOI: https://doi.org/10.1016/S0951-8320(00)00047-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832000000478.

[24] International Energy Agency, *Renewables 2021*, https://www.iea.org/reports/renewables-2021.

[25] D. Bizo, R. Ascierto, A. Lawrence, and J. Davis, *Uptime Institute Global Data Center Survey 2021*, https://uptimeinstitute.com/2021-data-center-industry-survey-results.

[26] E. Vadal and C. Graham, *Downtime Costs Auto Industry $22k/Minute - Survey*. Thomasnet, 2006, https://news.thomasnet.com/companystory/downtime-costs-auto-industry-22k-minute-survey-481017.

[27] International Business Machines, *How Industry 4.0 technologies are changing manufacturing*. IBM, Assessed 15.01.2022, https://www.ibm.com/topics/industry-4-0.

[28] Britannica, T. Editors of Encyclopaedia, *Industrial Revolution*. Encyclopedia Britannica, https://www.britannica.com/event/Industrial-Revolution.

[29] The Institute of Entrepreneurship Development, *The 4 Industrial Revolutions*. iED, Assessed 15.03.2022, https://ied.eu/project-updates/the-4-industrial-revolutions/.

[30] K.-A. Nguyen, P. Do, and A. Grall, "Multi-level predictive maintenance for multi-component systems," *Reliability engineering & system safety*, vol. 144, pp. 83–94, 2015.

[31] J. Wang, L. Zhang, L. Duan, and R. X. Gao, "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing," *Journal of Intelligent Manufacturing*, vol. 28, no. 5, pp. 1125–1137, 2017.

[32] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv preprint arXiv:1912.07383*, 2019.

[33] Seebo Interactive, *The Complete IoT Industry 4.0 Knowledge Library*. Resources, https://www.seebo.com/resources/, Accessed: 22.02.2022.

[34] LimbleCMMS, *A Complete Guide to Predictive Maintenance*, https://limblecmms.com/predictive-maintenance/, Accessed: 22.02.2022.

[35] Microsoft Azure AI, *Predictive Maintenance Playbook*. Team Data Science Process, https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview, Accessed: 22.02.2022.

[36] J. C. Jønsson, *Reliable predictions with limited data*. EDR & Medeso, 2022, https://go.edrmedeso.com/iot-suite.

[37] O. F. Eker, F. Camci, and I. K. Jennions, "Major challenges in prognostics: Study on benchmarking prognostics datasets," in *PHM Society European Conference*, vol. 1, 2012.

[38] R. K. Mobley, *An introduction to predictive maintenance*. Elsevier, 2002.

[39] C. Scheffer and P. Girdhar, *Practical machinery vibration analysis and predictive maintenance*. Elsevier, 2004.

[40] J. Levitt, *Complete guide to preventive and predictive maintenance*. Industrial Press Inc., 2003.

[41] E. Lughofer and M. Sayed-Mouchaweh, *Predictive maintenance in dynamic systems: advanced methods, decision support tools and real-world applications*. Springer, 2019.

[42] J.-M. Nicod, B. Chebel-Morello, and C. Varnier, *From prognostics and health systems management to predictive maintenance 2: knowledge, reliability and decision*. John Wiley & Sons, 2017.

[43] E. T. Bekar, P. Nyqvist, and A. Skoogh, "An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study," *Advances in Mechanical Engineering*, vol. 12, no. 5, pp. 168–207, 2020.

[44] L. Zhang and J. Zhang, "A data-driven maintenance framework under imperfect inspections for deteriorating systems using multitask learning-based status prognostics," *IEEE Access*, vol. 9, pp. 3616–3629, 2020.

[45] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, IEEE, 2018, pp. 1394–1401.

[46] V. Kotu and B. Deshpande, "Chapter 12 - time series forecasting," in *Data Science (Second Edition)*, V. Kotu and B. Deshpande, Eds., Second Edition, Morgan Kaufmann, 2019, pp. 395–445, ISBN: 978-0-12-814761-0. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00012-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128147610000125.

[47] J. Brownlee, *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.

[48] ——, *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery, 2017.

[49] H. Hota, R. Handa, and A. Shrivas, "Time series data prediction using sliding window based rbf neural network," *International Journal of Computational Intelligence Research*, vol. 13, no. 5, pp. 1145–1156, 2017.

[50] J. Wang, W. Jiang, Z. Li, and Y. Lu, "A new multi-scale sliding window lstm framework (mssw-lstm): A case study for gnss time-series prediction," *Remote Sensing*, vol. 13, no. 16, p. 3328, 2021.

[51] W. Jiang, "Research on predictive maintenance for hydropower plant based on mas and nn," in *2008 Third International Conference on Pervasive Computing and Applications*, IEEE, vol. 2, 2008, pp. 604–609.

[52] T. Wetle and J. Foros, *MonitorX - final report*. Energi Norge, 2019, https://www.energinorge.no/fagomrader/forskning/forskningsprosjekter/produksjon/monitorx/.

[53] International Organization for Standardization, *Industrial systems, installations and equipment and industrial products — Structuring principles and reference designation — Part 10: Power plants*, 2015.02.05, Vernier, Geneva, Switzerland.

[54] Inductive Automation, *What is SCADA?* Resources, Assessed 15.03.2022, https://inductiveautomation.com/resources/article/what-is-scada.

[55] J. Tai, "Design on the digital electro-hydraulic servo system of the hydroturbine governor," in *2010 Asia-Pacific Power and Energy Engineering Conference*, 2010, pp. 1–4. DOI: 10.1109/APPEEC.2010.5449446.

[56] European Committee for Standardization, *Maintenance - Maintenance terminology*, Brussels, 16.07.2017.

[57] G. E. Box, "Gm jenkins time series analysis: Forecasting and control," *San Francisco, Holdan-Day*, 1970.

[58] S. Raschka, *Python machine learning*. Packt publishing ltd, 2015.

[59] T. Menzies, E. Kocagüneli, L. Minku, F. Peters, and B. Turhan, "Chapter 24 - using goals in model-based reasoning," in *Sharing Data and Models in Software Engineering*, T. Menzies, E. Kocagüneli, L. Minku, F. Peters, and B. Turhan, Eds., Boston: Morgan Kaufmann, 2015, pp. 321–353, ISBN: 978-0-12-417295-1. DOI: https://doi.org/10.1016/B978-0-12-417295-1.00024-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124172951000242.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[61] TensorFlow, *Introduction to TensorFLow*. Learn, Assessed 15.03.2022, https://www.tensorflow.org/learn.

[62] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[63] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, 2015. arXiv: 1511.08458 [cs.NE].

[64] S. W. Smith *et al.*, "The scientist and engineer's guide to digital signal processing," 1997.

[65] M. Qiao, S. Yan, X. Tang, and C. Xu, "Deep convolutional and lstm recurrent neural networks for rolling bearing fault diagnosis under strong noises and variable loads," *Ieee Access*, vol. 8, pp. 66 257–66 269, 2020.

[66] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[67] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[68] *SVM, Support Vector Machines, Scikit learn Library*, https://scikit-learn.org/stable/modules/svm.html.

[69] M. A. Chandra and S. Bedi, "Survey on svm and their application in image classification," *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, 2021.

[70] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 2, 2005, pp. 1164–1170.

[71] J. Brownlee, *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Machine Learning Mastery, 2017, Assessed 15.02.2022, https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[72] ——, *How to Configure the Learning Rate When Training Deep Learning Neural Networks*. Machine Learning Mastery, 2019, Assessed 15.02.2022, https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/.

[73] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[74] R. Parmar, *Common Loss functions in machine learning*. Towards Data Science, 2018, Assessed 15.02.2022, https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23/.

[75] J. Brownlee, *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. Machine Learning Mastery, 2019, Assessed 15.02.2022, https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[77] D. L. Poole and A. K. Mackworth, *Artificial Intelligence, Foundations of Computational Agents*. Cambridge University Press, 2017, https://artint.info/index.html.

[78] J. Brownlee, *Difference Between a Batch and an Epoch in a Neural Network*. Machine Learning Mastery, 2018, Assessed 15.02.2022, https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.

[79] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[80] G. Hinton and T. Sejnowski, *Unsupervised learning: foundations of neural computation*. The MIT press, 1999.

[81] Modelica Association Project, *Functional Mock-up Interface*. FMI-Standatd, Assessed 14.01.2022, https://fmi-standard.org/about/.

[82] A. Jenul, S. Schrunner, K. H. Liland, U. G. Indahl, C. M. Futsaether, and O. Tomic, "Rent—repeated elastic net technique for feature selection," *IEEE Access*, vol. 9, pp. 152 333–152 346, 2021.

Appendix A

Servomotor analysis and results

```
#!/usr/bin/env python
# coding: utf-8

# Servo Motor Analysis

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.font_manager
from sklearn import svm
plt.style.use("ggplot")
from ipywidgets.widgets import interact, Layout


# In[2]:


df_servo_open = pd.read_csv("TilSK/servoindikeringOPEN.csv",";",
    skiprows=(3))

df_servo_close = pd.read_csv("TilSK/servoindikeringCLOSE.csv",";",
    skiprows=(3))

df_servo_v1 = pd.DataFrame()
df_servo_v1 = df_servo_open[['Relative Time (ms)', 'POS;WG_PILOT', 'HWI
    ;WG_SIGNAL',
                             'HWI;Servo_pressure +', 'HWI;
    Servo_pressure -']]
df_servo_v1 = df_servo_v1.rename(columns={'Relative Time (ms)': '
    Relative Time',
                                          'POS;WG_PILOT': 'pos',
                                          'HWI;WG_SIGNAL': 'FCA',
                                          'HWI;Servo_pressure +': 'HWI;
    Servo_pressure +',
                                          'HWI;Servo_pressure -': 'HWI;
    Servo_pressure -'})
df_servo_v1 = df_servo_v1.set_index('Relative Time')
df_servo_v1 = df_servo_v1.iloc[1:]

df_servo_v1 = df_servo_v1.apply(lambda x: x.str.replace(',','.'))
```

```
df_servo_v2 = pd.DataFrame()
df_servo_v2 = df_servo_close[['Relative Time (ms)', 'POS;WG_PILOT', 'HWI;WG_SIGNAL',
                              'HWI;Servo_pressure +', 'HWI;Servo_pressure -']]
df_servo_v2 = df_servo_v2.rename(columns={'Relative Time (ms)': 'Relative Time',
                                          'POS;WG_PILOT': 'pos',
                                          'HWI;WG_SIGNAL': 'FCA',
                                          'HWI;Servo_pressure +': 'HWI;Servo_pressure +',
                                          'HWI;Servo_pressure -': 'HWI;Servo_pressure -'})
df_servo_v2 = df_servo_v2.set_index('Relative Time')
df_servo_v2 = df_servo_v2.iloc[1:]

df_servo_v2 = df_servo_v2.apply(lambda x: x.str.replace(',','.'))

df_servo_v3 = pd.concat([df_servo_v1, df_servo_v2])


df_servo_v3 = df_servo_v3.reset_index(inplace=False)

cols = df_servo_v3.columns

for col in cols:
    df_servo_v3[col] = df_servo_v3[col].astype(float)

df_servo_v3["diff"] = df_servo_v3['HWI;Servo_pressure +'] - df_servo_v3['HWI;Servo_pressure -']


# In[3]:


plt.figure(figsize=(12,6))
plt.scatter(df_servo_v3['FCA'], df_servo_v3['diff'])
plt.xlabel('Opening position (%)')
plt.ylabel('Pressure difference')
plt.savefig('servo_pressurediff_pos.png')


# In[4]:


from sklearn.svm import LinearSVC
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import make_classification
from sklearn import svm


# In[5]:


# Load training data
X_train = np.array(df_servo_v3[["FCA", "diff"]])

# Load test data
X_test = np.array(df_servo_v3[["FCA", "diff"]].sample(n=1000))


# In[6]:
```

```
99
100
101 X_train.shape
102
103
104 # In[7]:
105
106
107 gamma=26e-4
108 nu=1.3e-4
109
110 clf = svm.OneClassSVM(kernel="rbf", gamma=gamma, nu=nu, tol=.01)
111 clf.fit(X_train)
112 y_pred_train = clf.predict(X_train)
113 y_pred_test = clf.predict(X_test)
114 n_error_train = y_pred_train[y_pred_train == -1].size
115 n_error_test = y_pred_test[y_pred_test == -1].size
116
117 # create the grid for background colors
118 h = 0.02
119 x_min, x_max = X_train[:, 0].min() - 10, X_train[:, 0].max() + 30
120 y_min, y_max = X_train[:, 1].min() - 10, X_train[:, 1].max() + 10
121 xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max
        , h))
122
123
124 Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
125 Z_ = Z.reshape(xx.shape)
126
127
128
129 # In[8]:
130
131
132 pos1, press1 = 85, -30
133 pos2, press2 = 25, -20
134 pos3, press3 = 10, -15
135 pos4, press4 = 30, -10
136 pos5, press5 = 45, -22
137 pos6, press6 = 70, -44.8
138
139
140 # In[9]:
141
142
143 from decimal import Decimal
144 plt.figure(figsize=(12,6))
145 plt.contourf(xx, yy, Z_, levels=np.linspace(Z.min()+0.35, 0, 7), cmap=
        plt.cm.PuBu, alpha=0.6)
146 a = plt.contour(xx, yy, Z_, levels=[0], linewidths=2, colors="darkred")
147 plt.contourf(xx, yy, Z_, levels=[0, Z.max()], colors="palevioletred",
        alpha=0.6)
148 s = 40
149 b1 = plt.scatter(X_train[:, 0], X_train[:, 1], c="white", s=s,
        edgecolors="k")
150 b2 = plt.scatter(X_test[:, 0], X_test[:, 1], c="blueviolet", s=s,
        edgecolors="k")
151 c1 = plt.axvline(x=7, c="green")
152 c2 = plt.axvline(x=75, c="green")
153 n = plt.scatter(pos1, press1, c="yellow")
154 plt.scatter(pos2, press2, c="yellow")
155 plt.scatter(pos3, press3, c="yellow")
156 plt.scatter(pos4, press4, c="yellow")
```

```
plt.scatter(pos5, press5, c="yellow")
plt.scatter(pos6, press6, c="yellow")
plt.axis("tight")
plt.xlim((x_min, x_max))
plt.ylim((y_min, y_max))
plt.legend(
    [a.collections[0], b1, b2, c1, n],
    [
        "Learned frontier",
        "Training observations",
        "New regular observations",
        "Min&Max opening position",
        "New observations"
    ],
    loc="upper right",
    prop=matplotlib.font_manager.FontProperties(size=11),
)
plt.xlabel(
    "Opening Position"
)
plt.title("error train: %d/18152 ; errors novel regular: %d/1000 ; $\
    gamma$ = %.2E ; $\\nu$= %.2E"
    % (n_error_train, n_error_test, Decimal(gamma), Decimal(nu)))
plt.ylabel('Pressure Difference')

plt.savefig('OCSVM_servo_v1.png')
plt.show()


# In[10]:


def check_new_val(clf, position, pressure):
    import plotly.graph_objects as go
    from decimal import Decimal
    plt.figure(figsize=(10,5))
    plt.contourf(xx, yy, Z_, levels=np.linspace(Z.min()+0.35, 0, 7),
                 cmap=plt.cm.PuBu, alpha=0.6)
    a = plt.contour(xx, yy, Z_, levels=[0], linewidths=2, colors="
    darkred")
    plt.contourf(xx, yy, Z_, levels=[0, Z.max()], colors="palevioletred
    ", alpha=0.6)
    s = 40
    b1 = plt.scatter(X_train[:, 0], X_train[:, 1], c="white", s=s,
    edgecolors="k")
    b2 = plt.scatter(X_test[:, 0], X_test[:, 1], c="blueviolet", s=s,
    edgecolors="k")
    c1 = plt.axvline(x=7, c="green")
    c2 = plt.axvline(x=75, c="green")
    n = plt.scatter(position, pressure, c="yellow")
    plt.axis("tight")
    plt.xlim((x_min, x_max))
    plt.ylim((y_min, y_max))
    plt.legend(
        [a.collections[0], b1, b2, c1, n],
        [
            "Learned frontier",
            "Training observations",
            "New regular observations",
            "Min&Max opening position",
            "New observations"
        ],
        loc="upper right",
```

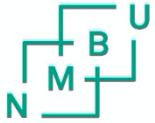
```
        prop=matplotlib.font_manager.FontProperties(size=11),
    )
    plt.xlabel(
        "Opening Position"
    )
    plt.title("error train: %d/18152 ; errors novel regular: %d/1000 ;
    $\gamma$ = %.2E ; $\\nu$= %.2E"
        % (n_error_train, n_error_test, Decimal(gamma), Decimal(nu)))
    plt.ylabel('Pressure Difference')
    plt.show()
    plt.savefig('OCSVM_newval1.png')
    new_val = clf.decision_function(np.c_[position, pressure])*100
    if position < 7:
        print('Error: Possible oil leakage in the servo')
        fig = go.Figure(go.Indicator(
        mode = "number+gauge+delta", value = 0,
        domain = {'x': [0, 1], 'y': [0, 1]},
        delta = {'reference': 0, 'position': "top"},
        title = {'text':"<b>Error</b><br><span style='color: gray; font
    -size:0.8em'>Piston</span></b><br><span style='color: gray; font-
    size:0.8em'>out of</span></b><br><span style='color: gray; font-
    size:0.8em'>range</span>", 'font': {"size": 14}},
        gauge = {
            'shape': "bullet",
            'axis': {'range': [-15, 7]},
            'bgcolor': "white",
            'steps': [
                {'range': [-15, 7], 'color': "crimson"}
                ],
            'bar': {'color': "crimson"}}))
        fig.update_layout(height = 250, width = 750)
        fig.show()
    elif position > 75:
        print('Error: Possible oil leakage in the servo')
        fig = go.Figure(go.Indicator(
        mode = "number+gauge+delta", value = 0,
        domain = {'x': [0, 1], 'y': [0, 1]},
        delta = {'reference': 0, 'position': "top"},
        title = {'text':"<b>Error</b><br><span style='color: gray; font
    -size:0.8em'>Piston</span></b><br><span style='color: gray; font-
    size:0.8em'>out of</span></b><br><span style='color: gray; font-
    size:0.8em'>range</span>", 'font': {"size": 14}},
        gauge = {
            'shape': "bullet",
            'axis': {'range': [-15, 7]},
            'bgcolor': "white",
            'steps': [
                {'range': [-15, 7], 'color': "crimson"}
                ],
            'bar': {'color': "crimson"}}))
        fig.update_layout(height = 250, width = 750)
        fig.show()
    else:
        fig = go.Figure(go.Indicator(
            mode = "number+gauge+delta", value = new_val[0],
            domain = {'x': [0, 1], 'y': [0, 1]},
            delta = {'reference': 0, 'position': "top"},
            title = {'text':"<b>Outlier</b><br><span style='color: gray
    ; font-size:0.8em'>position</span></b><br><span style='color: gray;
     font-size:0.8em'>from</span></b><br><span style='color: gray; font
    -size:0.8em'>boundary</span>", 'font': {"size": 14}},
            gauge = {
                'shape': "bullet",
```

```
                'axis': {'range': [-15, 7]},
                'threshold': {
                    'line': {'color': "red", 'width': 2},
                    'thickness': 0.75, 'value': 0},
                'bgcolor': "white",
                'steps': [
                    {'range': [-15, 0], 'color': "crimson"},
                    {'range': [-10, 0], 'color': "darksalmon"},
                    {'range': [-7, 0], 'color': "cadetblue"},
                    {'range': [-5, 0], 'color': "royalblue"},
                    {'range': [-3, 0], 'color': "cornflowerblue"},
                    {'range': [-1, 0], 'color': "aqua"},
                    {'range': [0, 7], 'color': "lightgreen"}
                    ],
                'bar': {'color': "black"}}))
        fig.update_layout(height = 250, width = 750)
        fig.show()
    return


# In[15]:


check_new_val(clf, 40, -25)


# In[ ]:
```



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway