TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Max Filip Wakéus 221143IAFM

Condition monitoring and Predictive Maintenance of Ball Bearings

Master's Thesis

Supervisor: Gerry Nigro
MsC

Co-supervisor: Eduard Petlenkov
MsC

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Max Filip Wakéus 221143IAFM

KUULLAAGRITE SEISUKORRA JÄLGIMINE JA ENNETAV HOOLDUS

Magistritöö

Juhendaja: Gerry Nigro
MsC

Kaasjuhendaja: Eduard Petlenkov
MsC

Tallinn 2022

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Max Filip Wakéus ....

(signature)

Date: 09.08.2022

Abstract

This thesis explores the possibility of monitoring ball bearing health based on information from an accelerometer and explicitly targets the health issue of corrosion. The bearings are not affected by radial force, minimal axial force, and are commonly found in computer fans or anemometers. This is accomplished using an Artificial Neural Network approach, namely LSTM-classification on vibration data from an accelerometer. Data is generated by attaching an accelerometer to an anemometer at the optimal location of the bearings. A hairdryer generates the wind, and a Raspberry Pi collects the data and performs the classification. The bearings are corroded using an acid approach to 18-predefined levels.

The network predicts the classes with an accuracy of 94% on previously unseen test data. This project highlights the possibilities of automated classification of corrosion levels with minimum need for manual engineering or domain knowledge.

List of Abbreviations and Terms

DNN	Deep Neural Network
LSTM	Long Short Term Network
IoT	Internet of Things
GPU	Graphical processing unit
IDE	Integrated Development Environment
PdM	Predictive maintenance
DTW	Dynamic time warping
RNN	Recurrent Neural Network
ANN	Artificial Neural Network
NN	Neural Network
GPIO	General Purpose Input/Output
MSE	Mean Square Error
MAE	Mean Absolute Error
SoA	State of Art
BPTT	Back Propagation Through Time
GRU	Gated Recurrent Unit
LRP	Layer-wise Relevance Propagation
SGD	Stochastic Gradient Decent
ADAM	Adaptive Moment Estimation
GPIO	General Purpose Input and Output

Table of Contents

List of Figures

List of Tables

1. Introduction

This Master thesis targets the topic of Condition monitoring and Predictive maintenance in ball-bearings based on vibrations. It specifically looks at monitoring the health of the bearings due to corrosion.

Bearings are vital in most rotary equipment and often exist in multiple locations. Many different types of bearings are available on the market today, like roller bearings, ball thrust bearings, roller thrust, and tapered bearings. However, the most common is the ball-bearing due to its design and versatility.

Condition monitoring indicates that the goal is to monitor the state of health of the equipment by looking at information collected through sensors, production data, and performance. This information can then be used to proactively predict when a part needs to be replaced or maintained, which allows for avoiding unplanned stops and downtime, including a way to plan resources. This can be done by using a model that is either data-driven and basing its prediction on data or theory-based and built on the knowledge of the equipment and not necessarily on the data from it.

1.1 Background

Keeping the industry running with the highest efficiency is key in today's industrial market. Every time a machine stops means that it is no longer producing and generating an income and thereby is only a cost for the company. Therefore one of the most important parts of the product life cycle of machinery is the maintenance part, which will keep the machinery running at optimal capacity. Since bearings are at the heart of rotary machinery it is vital to keep them in perfect health and if the health dwindles to identify this well in time before something happens that would affect the performance. According to [1] up to 55% of the failures in rotary machinery are caused by bearings.

The health of bearings can be affected by a multitude of different causes since they operate in such different environments. A common failure is due to corrosion, especially those that operate under minimal load and thereby stay in use for a longer period of time. Manufacturers use a wide span of strategies to protect bearing from environmental effects such as encapsulation, corrosion-resistant material, grease, etc. This makes it hard for them to be visually inspected without being manually taken apart and investigated, so a cluster of different techniques can be used to monitor the health while the bearings are active. The most common is by looking at vibrations since that is one of the first symptoms of degrading health for bearings and is usually followed by noise, heat, and finally smoke. In the past this was done manually by measuring, looking, touching, and listening to the bearings in a process environment [2].

Industry 4.0 and digitization create the possibility to acquire data at a scale that has never been seen before by a decrease in production-cost of sensors in turn leading to an exponential deployment. This creates possibilities for monitoring the equipment and processes continuously in order for it to run as efficiently as possible for as long as possible. One thing that each and every manufacturer wants to avoid is sudden unplanned stops in production, this immediately accumulates costs. This has led to Predictive Maintenace, PdM becoming increasingly popular as an alternative to the preventative maintenance strategy i.e. maintenance when needed not based on an interval. With the use of IoT, Big Data, and Neural Networks it is today possible to predict maintenance at a much larger scale than before without having to rely on domain knowledge.

1.2 Problem

With Industry 4.0 and the lower cost of sensors, controlling the health state of bearings is easier than ever at least in the sense of gathering data. One of the major issues that have risen is the quality of data since such a large quantity is being generated.

The traditional approach of PdM was to generate the data, and then through statistical tools create the features assumed to be needed. These would then be analyzed and a selection of the most useful ones would be used to train the model. This is however a very manual approach that demands a high domain knowledge in statistics, type of data, and domain from which the data was extracted. This creates a very domain-specific model which is easily influenced by outer forces and thereby not very robust.

This trigger the question, is it possible to build a model that still is able to perform the task of monitoring the health of the bearing but without the need for any of the above-mentioned domain knowledge? In other words, predicting health issues for bearings and estimating the remaining useful life, RUL. To summarize this into three research questions that will be investigated throughout the rest of this thesis.

1. How can we classify corrosion levels of bearings that are not affected by radial load without statistical pre-processing of vibration signals using Artificial Neural Networks?

2. How can we use the pre-defined corrosion levels to estimate the Remaining useful life of the bearing in terms of time?

1.3 Purpose

The purpose of the thesis is to explore the possibilities of performing predictive maintenance on bearings without the need for domain knowledge, specifically looking at corrosion.

This thesis is done in collaboration with Santer Reply for the European research project Arrowhead. The desired output from this project for Arrowhead is a use-case where it is shown how the tools and architecture developed in Arrowhead can be utilized. The goal of Arrowhead is to "enable collaborative automation by networked embedded devices" [3]. This use case is developed within the sub-project of Smart buildings and architecture which will be described further in chapter 4 4.

This hypothesis will be illustrated in an experiment using an anemometer to generate vibration data on corroded bearings. The thesis will raise and discuss topics like Maintenance strategies, vibration analysis, Machine learning, and Artificial Neural Networks.

1.4 Goal

The aim is to provide an alternative approach to identifying non-optimal performance in ball bearings compared to the traditional method of manual feature extraction and selection. The objective also includes predicting the remaining useful life of the ball-bearing as well as showcasing how Arrowhead tools can be utilized.

The deliverables are listed below:

1. Condition monitoring model
2. Classification model
3. RUL model

1.5 Benefits, Ethics and Sustainability

This could benefit professionals that are looking for an introduction and practical approach to predictive maintenance through Artificial Neural Networks. Either if they are planning on applying it in their own industrial environment or already have implemented it but are looking for a way to increase the performance.

Some ethical issues that might arise could be the lack of insight into the decision algorithms. The data could be moved away from the factory where it usually has resorted to external providers. This moves the power to a few selected providers which could result in problems down the line when dependencies have been built up. With this comes also the movement of knowledge and expertise which could further increase the dependency on third parties. This highlights the question of security when moving data outside the factory and through that control. Is the data safe when it resorts to a different location or is being transferred there?

The environmentally sustainable part of this can be summarized within, the longer we can keep equipment in a healthy state, the fewer resources we use. Not to forget is the economic part as it provides another incentive to further invest in PdM for the industry. However, that shifts the focus from an economy that thrives on high consumption of products. This could lead to short-term downsides in the form that society needs to adapt.

Looking at the project itself from an environmental standpoint shows that multiple good bearings were on purpose destroyed in order to generate data. The data collection was quite extensive which lead to high energy consumption.

1.6 Methodology

For this thesis, a quantitative research method has been chosen to answer the research questions which is an optimal approach when developing an experiment and presenting the result through statistical analysis and numbers. A deductive approach will be used by looking at the current SoA and from that develop the hypothesis that will then be proven or disproven through an experiment. A method to generate data will be developed throughout this thesis and therefore primary data will be used to test the hypothesis. The data collected will be experimental data in the sense that it is created during the project and it will be influenced in order to test the hypothesis.

1.7 Stakeholders

- **Reply SpA** as the client I am making the project for.
- **Arrowhead** whom the project is for.
- **Tallinn University of Technology** as the university responsible for the thesis project.
- **EIT Digital** as an indirect stakeholder.

- **Universidad de Politecnica de Madrid** as an indirect stakeholder.

1.8 Delimitations

The project is limited to looking at the damages to bearings caused by corrosion through a specific type of acid approach where the original grease has not been manipulated. A proof of concept will be developed using the features of an accelerometer and wind speed. The data collection is limited to a fixed environment with minimum variation in environmental variables such as temperature, wind speed, humidity, etc. The test-bed setup is also fixed and no other applications will be investigated in this thesis. Data collection will only be collected at two speeds, 5 m/s, and 10 m/s. It is therefore needed to pay attention that this is a specific application and in order for this application to work in another setting further data collection is needed.

1.9 Outline

The thesis is structured as follows:

- **Chapter 2** will explored the theory in this field and the current state of the art.
- **Chapter 3** will explore the engineering perspective and the respective methods will be presented.
- **Chapter 4** will explain the actual experiment.
- **Chapter 5** will explore the result of the experiment.
- **Chapter 6** will then discuss the results and how this could be continued in future research.

2. Theory and State of Art

The theory for this thesis could be defined on different levels, firstly we will discuss the theory behind maintenance. Then we will continue with a deeper look at condition monitoring and predictive maintenance. Lastly, the field of AI and machine learning will be investigated in the realm of maintenance.

2.1 Maintenance

Maintenance is an increasingly interesting field, especially in times of financial hardship, this is the time when good maintenance strategies make or break companies. In 2019, 7 000 billion euros was spent on maintenance, repair, and renovation [4]. This highlights the possibilities of gaining business advantages by developing maintenance strategies.

2.1.1 History of maintenance

The definition of maintenance are many, but the main point remains the same, to restore the machinery to its intended state of performance. Machinery has always been subject to wear and tear which requires maintenance and throughout history, it can be boiled down to these three main strategies.

Corrective maintenance

Also called breakdown maintenance is the traditional way of performing maintenance. In this strategy, the machinery runs at full capacity until it simply breaks down and all focus is put on how to fix it and get it back up and running again. Fixing the issue on the spot demands extremely skilled technicians and a huge spare-part warehouse since there was no warning of what may break or when. This approach relies heavily on the technician's domain knowledge. This strategy is however still a very valid approach, [4] estimates that over 55% of the activities on an average facility still use this approach which includes simpler tasks such as cleaning and lubrication. It may sound counterintuitive, but it still plays a vital part in some maintenance strategies today. The benefits can be thought of as short-term savings. Considering new machinery, the likelihood that it breaks down, in the beginning, is fairly small so an adapted reactive maintenance approach is usually deployed. Meaning that no stops will take place and during this period no resources are wasted on stopped production or labor.

This short-term gain usually doesn't weigh up the downside of a reactive maintenance strategy due to a number of reasons. First of all, identifying the issue/s might not be as straight forward and this in itself could extend the downtime further. When the failure eventually occurs, the damage is usually greater than it would have been performing proactive maintenance. Consider moving rotary equipment, if a bearing seizes during operation, the forces will very likely damage other parts as well. Another downside is the need for an extensive spare part warehouse that should cover each and every possible broken part. If this is not supplied, there will be increased downtime since spare parts will need to be delivered. Not to mention the cost of express shipping.

Preventative maintenance

Is the type where a time frame or a specific number of cycles decides the maintenance interval. Meaning that the machinery is stopped and undergoes maintenance at an interval, often suggested by the manufacturer. This increase the lifespan of the machinery and its efficiency since it is kept in better condition. Another benefit is that it allows for a smaller spare-part warehouse since the stop is planned and the necessary parts can be ordered in time. Which spare parts are commonly known as well so there is little need for a technician with extensive domain knowledge and cheaper labor can be used. It also allows for an overview of the rest of the machinery, which can be inspected for unforeseen wear and tear. This can then be planned to be replaced for the next scheduled stop. Finally, it increases the possibility to plan the production stops which results in the expected yield.

However, it does come with its own downside, which is the increased downtime compared to reactive maintenance. More spare parts are used than actually needed since it is better to switch them out now rather than hope that they will last until the next stop. Because maintenance is performed no matter the condition of the parts but due to theoretical wear. The increased stops lead to loss of production capacity since start-up/shutdown takes time until optimal production capacity is reached again. Finally, having the personnel at the site to perform this increased number of stops led to an increase in personal costs as well.

Predictive maintenance, (PdM)

Is the type of maintenance that utilizes the power of data. Instead of running parts until they break or performing service on them no matter the state of health, PdM allows us to harness the power of data and monitor the health of the parts. This, in turn, allows us to perform scheduled maintenance when needed.

PdM takes the benefits from both reactive maintenance and preventative maintenance while minimizing their respective limitations. PdM allows the machinery to run for the maximum amount of cycles before the production is intervened and maintenance is performed. It mitigates the risk of breakdowns by performing preventative maintenance but at the time that best suits the production planning while exploiting the parts to their fullest performance life cycle. This leads to smaller spare-part warehouses, fewer planned stops, less maintenance personnel on call and it especially allows for preparation since the data shows where the issue originates from.

The possible cost benefits for general predictive maintenance are estimated as follows 1.

Type	**Saving**
Return of investment	10 times
Reduction of maintenance cost	25% to 30%
Troubleshooting	70% to 75%
Reduction in downtime	35% to 45%
Increased production	20% to 25%

Table 1. *Estimated savings by Predictive maintenance [4]*

There are however downsides with PdM as well and that becomes evident when the initial investment is to be made. It is hard to defend the invest short-term but the benefits are immense long-term. First generating the data needed is commonly done by the deployment of sensors and connecting already existing data producers. This demands domain knowledge and extensive planning. Then the whole communication architecture needs to be built and configured before PdM can start to generate yield. Finally, after data collection has been running for an initial time, models can be built to allow for condition monitoring and predictive maintenance. Not to forget the security needed around this to handle production data and all that comes with it such as protocols, servers, etc.

To summarize the different types of maintenance and connect them to the industrial evolution 2.

Industry revolution	**Industry 1.0**	**Industry 2.0**	**Industry 3.0**	**Industry 4.0**	**Industry 5.0**
Characteristics of the industrial revolution	Mechanization, steam power, weaving loom	Mass production, assembly lines, electrical energy	Automation, computers, electronics	Cyber Physical Systems, IoT, networks, cloud	Intelligence, human centric collaboration

Continues...

2 Table – *Continues...*

Industry revolution	**Industry 1.0**	**Industry 2.0**	**Industry 3.0**	**Industry 4.0**	**Industry 5.0**
Type of maintenance	**Reactive maintenance**	**Planned maintenance**	**Productive maintenance**	**Predictive maintenance**	**Intelligent maintenance**
Inspection	Visual inspection	Instrumental inspection	Sensor monitoring	Predictive analysis	Collaborative
Overall equipment efficiency	<50%	50-75%	75-90%	>90%	To be seen
Maintena-nce team reinforcem-ent	Trained craftsmen	Inspectors	Reliability engineers	Data scientists	To be seen

Table 2. *Industrial revolutions with respective maintenance strategies [4], [5]*

2.2 Data

PdM is heavily reliant on quality data which can be generated from multiple sources but there are mainly two groups, built in- and external sensors. The benefit of built-in sensors is that they are often in the optimal location for picking up the data needed. The downside is that the machine needs to be bought with sensors built in which in general demands a larger investment and is not applicable for legacy machinery. A retrofit is possible but that might require the machine to be taken offline which might be hard to justify in loss of production. In addition to that their location usually makes them harder to replace in case of failures and the parts themselves are more expensive. External sensors on the other hand have the benefit of being accessible and applicable where needed. The location will most likely not be as optimal as built-in so there is to strike a balance between the two of them.

Since bearings are one of the most crucial parts of machinery and exist in most moving parts, they are therefore often the cause in machine failures. Usually, the source of the failure in the bearings relates to over-loading, over-speeding, or lubrication fault [6]. But there are other causes for failures as well such as alignment issues and corrosion.

For bearings, vibrations have long been the method of choice for monitoring due to their proven track record but there are other interesting approaches as well such as heat sensors and sound sensors. The limitation with heat sensors is that when the sensors start to notice a heightened level of heat generated by the bearing it might already be too late to stop the dwindling performance. Sound is a fairly new approach and does work in many cases but the downside is that it could be difficult to filter the sound from the machine and the environment. Vibrations are commonly measured with an accelerometer which can measure the acceleration in one or 3 different directions, X, Y, and Z. That is why a 3-axis accelerometer will be used in this thesis to identify vibrations producing three features; X-acceleration, Y-acceleration, and Z-acceleration.

2.3 Models

There are two different types of PdM solutions in the form of models, model-based and data-based. The model-based needs extensive domain knowledge since the predictive model is designed based on theory. Data-driven models on the other hand learn from historical data and develop predictive models automatically. Since we are looking at building a model using the minimum amount of domain knowledge we will focus on data-driven models.

This model can further be divided into different levels as can be seen in figure 1 with AI as the highest level which can be summarized as the ability by computers to mimic human intelligence and through that solve tasks and problems. AI is the general concept of creating intelligence which highly depends on the definition of intelligence. Traditional AI is mostly related to the ability to do one task very well but is not very robust as if any parameter is changed the performance can drop significantly. This is where ANN makes the entrance, the main benefit of ANN is its adaptability, it will learn from the feedback. Similarly, like humans receive feedback when we do something and use that knowledge to update our general knowledge, the ANN uses the same principle. This way the computer can learn to generalize by being exposed to multiple scenarios [7].

The next main level is Machine learning which is a sub-category of AI meaning the tools and technologies used to perform tasks. "ML can be termed as an application of AI that offers systems the ability to learn and improve automatically from the experiences without explicit programming [8]." In [9] ML is defined as a sub-topic of AI with the main focus of identifying patterns and relationships within the data as opposite of AI that is general. Within ML there are 4 different categories.

Unsupervised learning identifies pattern and relationships within the data without knowing if it is doing well or not. This is due to that it only receives the input data and the output is unknown meaning it doesn't have the labeled data but tries to find clusters within the data. It therefore works with minimal guidance and explores the features for patterns, similarities and differences until it can cluster the data in different groups. The benefit here is that the data can be explored and there is no need for having labeled data which might not be available all the time.The downside on the other hand is that the algorithm doesn't know if the prediction is relevant or not and it could be difficult to reach an accuracy needed.

Reinforced learning learns by having an agent taking an action and then being rewarded for the action depending on the result using a time-delayed label. Hence the model isn't trained by using a pre-defined label as in the supervised learning.

Deep learning is essentially Neural Networks which will be covered in chapter 2.3.2. The number that is needed for the ANN to be considered a deep neural network compared to a Neural Network differs but can be generalized as if there is more than two hidden layers then it is a Deep Neural Network.

Supervised learning takes as input to the network features and tries to map them to a pre-determined label. It learns from history to predict the future. The model needs to be fed the input data, the labeled output data, and the model. It is then able to map the respective features to the correct label. Since it has the starting point, end point, and architecture it only needs to figure out the way in between. When the network has been trained and has mapped the features to correct labels, it is time for the prediction. This is done by letting the network see new input data and trying to predict the label with the knowledge it has previously learned. Supervised machine learning algorithms can be split into two different groups, regression which produces a real number, and classification, which produces a category [10]. The benefits are that it uses historical data to learn from and with more data it can make better predictions. The downside lies with the need of labeled data, which needs to be labeled manually. It also generally needs quite some time for training and sufficiently powerful hardware. Two common algorithms are SVM and ANN. For our case, a supervised learning model will be used since the data is generated within this project and thereby is pre-labeled and demands minimum extra pre-processing.

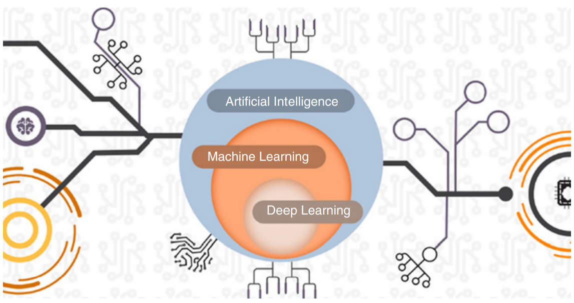


Figure 1. *AI, ML, DL [8]*

2.3.1 Traditional Machine Learning

Traditional machine learning algorithms are designed to solve specific tasks and require the input data to be pre-processed and analyzed by choosing which features will be used, etc. The benefits of traditional algorithms often come down to the ability to reach high accuracy with a limited amount of data and fast training times. Another important benefit of traditional ML is that they are easy to interpret, we can understand how the algorithm reaches the conclusion and thereby verify the logic. They don't require advanced hardware in general to perform the training which speaks to their advantage. However, in the world of time series, they require extensive pre-processing steps which in turn demands human input and domain knowledge. Some traditional models can handle time-series as input for example Dynamic Time Warping, DTW which matches time-series with each other by finding the optimal corresponding data point. This yield a measurement that can be used to classify time series based on similarities. However, this is more commonly used in anomaly detection by comparing the new time series against the baseline and having a threshold on the distance. The model doesn't learn anything but compares time-series to time-series so applying this to multi-label classes would be very computational heavy. The other downside is that the time series can be different within a class, so a generalized sequence needs to be identified for each class.

2.3.2 Neural Networks

In a world with big data where the amount of data presents an issue for the traditional algorithms, another approach is needed. This is where Neural Network models make their entrance since they need a large quantity of data for training. It is therefore one of the most common approaches for supervised learning methods to solve complex problems.

Introduction of Neural Network

Neural Network is a way of mimicking the way our brains work artificially using neurons. While the basic version of ANN can be dated back to 1940, it was first in the last couple of decades, the usage has increased exponentially. In the ANN, a neuron receives input from other neurons in the previous layer, and if that input is sufficient, the neuron will fire and send an impulse forward in the network. As in human intelligence, the ANN regulates the input to the neuron, while in the ANN, this is called weights and biases. It was the discovery of back-propagation that propelled ANN into the popular machine learning algorithm it is today. Back-propagation allows the network to update its weights and biases when the prediction doesn't match the desired output in supervised learning. This is the difference compared to Feed-forward ANN, which doesn't update its weights and biases i.e., doesn't use back-propagation. It can therefore be considered one of the simplest ANN due to it simply takes pre-determined weights and biases let the data run through the network from input layer, through the hidden layer, until the output layer as can be seen in figure 2.

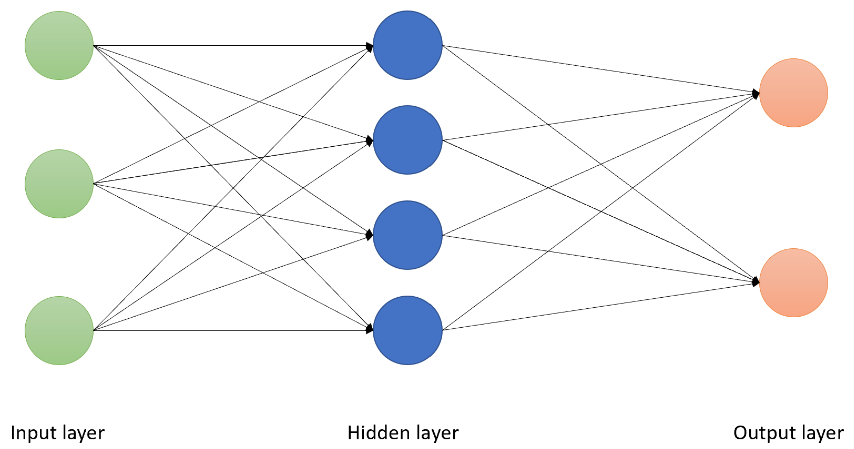


Figure 2. *Basic FFNN*

A neural network consists of a minimum of three layers, the input layer where the input data is organized, the hidden layer, and finally the output layer as can be seen in figure 2. In supervised learning, the Neural network maps the input to the output through repeated updating of the weights and biases based on the error rate, which is how the network adapts to the data it is fed. The learning process consists of adding initial weights and biases for each and every neuron. The network is then running data through the network until it reaches the output layer and a prediction. This predicted output is compared to the true output, and an error rate is calculated. This error rate is then fed back through the network and back-propagation the weights and biases are updated.

There are many benefits of using ANN over traditional ML, such as better performance and the ability to handle complex problems when a large quantity of good data is available. After the training which is computationally heavy and usually requires extensive time, the prediction is usually fairly quick, depending on the size of the model and the data. These models are usually transferable over different domains using transfer learning which allows the model to be trained on a large quantity of data and then adapted to a smaller dataset in a different domain where a large amount of data isn't available, warranting that the cases are similar enough. This also allows the model to be trained on simulated data and then applied to real data. It can then be trained on a smaller dataset since the majority of the optimization is already done. This flexibility also leads to that it can re-train itself if the environment and the input variable change which minimize the need for human intervention and leads to a more self-reliant system. Lastly, it can perform feature extraction and thereby removes the need for manual feature extraction and in extension removing the need for domain knowledge.

Recurrent Neural Network, RNN

The inputs of traditional feed-forward Neural networks are treated independently from each other which makes for a poor understanding of time-series and sequential data where the current data point is dependent on the previous data point. This need lead to the development of the Recurrent Neural Network, RNN which is characterized by its ability to remember. The ability of RNN to handle time series is well-known and this is due to the that neurons in RNN include adjacent time steps and thereby form cycles within each neuron. Which can be seen in (2.1) and (2.2) that displays the math behind the cycle where $h^{(t)}$ is the hidden layer activation at time t and consequently $h^{(t-1)}$ is at the previous time-step. x, y is the input and output, F, G are the activation function respectively, b_h, b_y are the bias and lastly W, U, V are the weight matrices [11].

$$h^{(t)} = F(Wh^{(t-1)} + Ux^{(t)} + b_h) \tag{2.1}$$

$$y^{(t)} = G(Vh^{(t)} + b_y) \tag{2.2}$$

The network can be trained through multiple time steps due to backpropagation through time, BPTT which is one way across the layers, and the weights are shared across the time steps.

The downside with RNN is the issue of exploding gradient decent and vanishing gradient decent. This leads to an inability to not remember very far back, RNN has more of a short-term memory. When the gradient is too small the weights get updated less and less until the gradient is zero and the weights aren't updated anymore. The opposite issue is when the gradient is too large and the weight updates become too large and create an unstable network. This leads to overshooting the minima and never actually converging.

Long Short Term Memory, LSTM

LSTM is a way of remedying the issue of vanishing/exploding gradient and increasing the ability to remember long-term data. The main difference between traditional RNN and LSTM is that the latter has replaced the neurons with memory cell structure which means

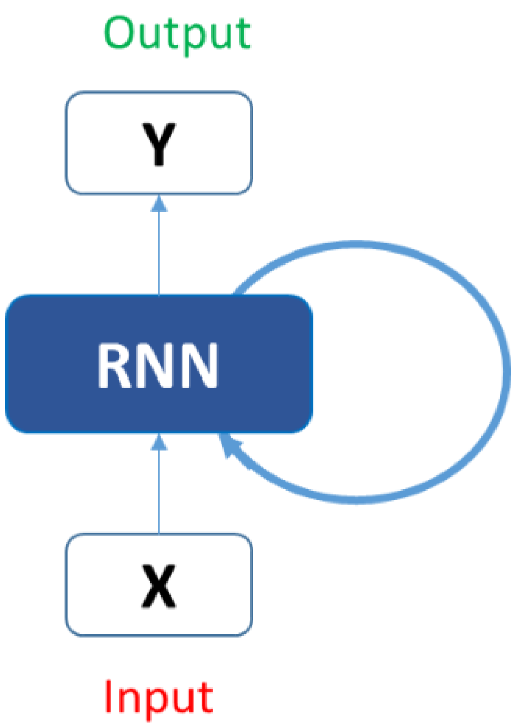


Figure 3. *Basic structure of RNN [12]*

that it has internal gates that can control the flow of information. GRU is a simplified version of LSTM to combat the heavy computations done in LSTM while still mitigating the exploding/vanishing gradient issue of RNN.

LSTM has 4 layers in the hidden layers controlled by three sigmoids in, out, and forget gate. Tanh is only used to update the cell state compared to RNN which only has tanh. The layers decide on what information should be passed on to the next layer and which should be discarded. The input to the next cell is the cell state and hidden state 4.

The hidden state from the previous time stamp relates to the encoding of the previous state. This doesn't relate to the output but the characteristics of that state. The difference between a hidden state and a cell state is that the cell state on the other hand looks at the complete data and not simply on the previous timestamp. The input from the cell state is passed through both the forget and input state so that the relevant knowledge extracted at this cell can be added to the cell state. Which features are added are controlled by weights in both the forget and input gate. These weights get multiplied with the input from the hidden state and the input data using either a sigmoid or a tanh function depending on the gate as can be seen in figure 4. The input gate is a bit more advanced than the forget gate since it first needs to generate features and then decide which of these features should be added to the cell state. The forget state has as an input already prepared features so it only needs to decide if it should add them to the cell state or not. These allow for storing information discovered earlier in the time series by adding and forgetting features at each time step. Due to LSTM's ability to handle long-term dependencies this will be the algorithm for extracting the features from the time series in this project.

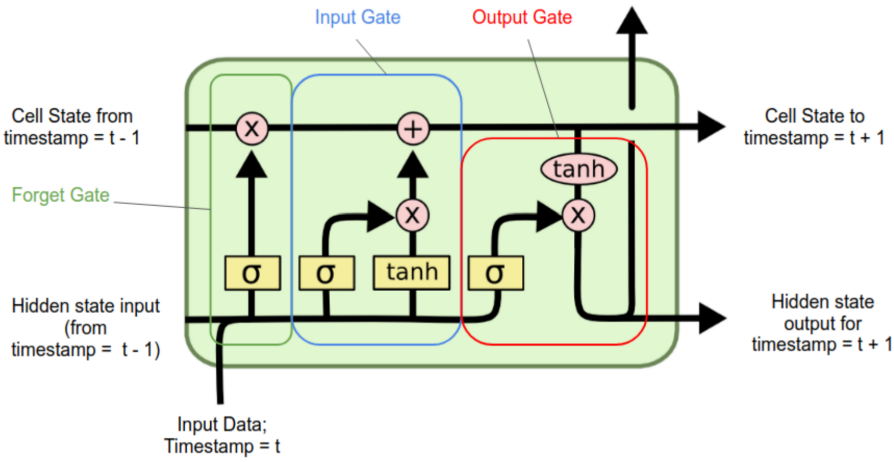


Figure 4. *LSTM cell [13]*

Over-fitting and Under-fitting

This refers to how well the model is learning and is monitored by having two datasets during the training phase of the model. The first one is used by the model to update the weights and biases and the latter to verify that it is actually learning and not just memorizing. This is done by monitoring the loss and the accuracy of both datasets and seeing how closely they are following each other. If the training accuracy reaches 100% while the validation accuracy only reaches 50% there is a clear sign of over-fitting which means that the model is simply remembering which data belongs to each class. On the other hand, under-fitting is when both the training and validation accuracy are bad, and the model doesn't learn anything.

In the case of under-fitting, the solution is pretty straightforward, increasing the complexity of the model and/or looking over the input data. By adding neurons layers and features the model receives more input to work with, another solution is to let the model train for a longer time. For over-fitting, the issue is the opposite and so are the solutions by removing features or stopping the training earlier before the training and validation accuracy diverges. Another common way is to use regularization techniques like dropout which randomly exclude part of the network during the training process. This limits the reliance on specific neurons within the network.

2.4 Features

Providing the models with correct data is commonly known as one of the most critical steps in machine learning.

2.4.1 Manual feature extraction

Historically vibration analysis has been done by extracting different features and from that stage selecting which features are the most interesting for the current dataset. Manual feature extraction is commonly done within the time domain, frequency domain, and time-frequency domain. This demands expertise in the process and makes the model relatively unstable with changes. In [11] the following types of features were extracted which is a mixture of features extracted through signal and statistical analysis. In order to transform the features from the time domain to the frequency domain Fast-Fourier transform is commonly used and to transform it to the Time-Frequency domain, Wavelet transform and Hilbert-Huang transform are common tools. These methods can also generate wavelet energy entropy, envelope, etc. [14].

Time domain

- Mean
- Variance
- RMS
- Entropy
- Skewness
- Kurtosis
- Shape factor
- Crest factor
- Upper bound of the histogram
- Lower bound of the histogram
- Impulse factor
- Margin factor
- Mean frequency
- Peak to peak

Frequency domain

- Frequency center
- Root mean square
- Standard deviation
- Spectral skewness
- Spectral kurtosis
- Spectral peak ratio (outer)
- Spectral peak ratio (inner)
- Spectral peak ratio (rolling element)
- Related similarity 2
- Related similarity 3
- Related similarity 4
- Related similarity 5
- Related similarity 6

Time and Frequency domain

- Wavelet energy ratio 1 (energy ratios of eight frequency sub-bands from Haar wavelet package transform)
- Wavelet energy ratio 2
- Wavelet energy ratio 3
- Wavelet energy ratio 4
- Wavelet energy ratio 5
- Wavelet energy ratio 6
- Wavelet energy ratio 7
- Wavelet energy ratio 8

The benefit of extracting and transforming features from bearing data comes in the form of interpretability. It is easier to understand how the algorithm learns from the data. It also allows choosing features that are deemed useful manually. Finally, it doesn't need as much data as other approaches.

The downside relates to the need for domain knowledge. To know which kind of data should be extracted, how it should be analyzed and then selected, which makes the algorithm application sensitive, the same model might struggle when applied to different scenarios.

2.4.2 Automated feature extraction

Another approach to this aspect is to use Neural Networks for extracting useful features by letting the network create and extract features it deems useful based on the data. CNN and RNN are common tools to be used for feature extraction.

The downside of this comes with the amount of data needed for the network to identify useful features. Another aspect is the interpretability of the features, this is commonly known as a black box. This is due to that it is hard to know what features the network have discovered and selected. Finally, there is also a need for domain knowledge here but in the form of a data scientist to build the network. This is different from the approach above since this is general knowledge and is not connected to the domain in the matter.

The benefits are that the network can adapt to new data and if there are changes in the setup, environment, etc, the network will evolve by looking at the new data. Thereby being less application specific and it doesn't have to rely on specific domain knowledge.

Using NN for feature extraction removes the need for manually defining features and extracting them from the vibration data. By allowing this, the network could identify features that humans might miss or do not consider important. This leads to a highly automatized end-to-end solution by eliminating as much of the human input as possible. Granted it can also miss features that humans might have found [15].

2.5 Related work

Article [16] proposes a combination of SVDD and PCA to identify corrosion pitting which is generated with a bearing under axial load. The data consisted of 203 healthy samples and 21 faulty samples and from this 13 statistical features were extracted. PCA was then performed on these features resulting in 6 features that explain 95.1 % of the total variability. Applying SVDD to these features results in an accuracy of 92.85% which is an increase from 96.2% for an SVDD without the PCA.

In the article, [17] a method to identify under-lubricated bearings that are not affected by any radial force and minimal axial force. The experiment was performed with different levels of lubrication as well as different levels of speed and temperature. The data collected was vibration, temperature and speed at four levels of grease, 100%, 50%, 25% and 0%. By using time domain and frequency domain features the different levels of lubrication were successfully identified.

In the article, [18] the frequency spectrum was analyzed using Fast-Fourier Transform and the envelope function. The damages explored of the bearings were inner race, outer race, and ball defects which were created through polishing with the grease-based compound of grit 400, 280, and 100. An accelerometer was used to collect vibration data which was then pre-processed. Since each bearing faults cause different frequency component the fault components was calculated such as ball pass frequency of inner race, ball pass frequency of outer race, and ball defect frequency. By applying FFT it was proven possible to identify different bearing faults using 3-D plots and having some understanding of signal analysis.

In the article, [15] it was proposed using a CNN to identify bearing faults based on vibration data. This way there is no need for manual feature extraction but it is handled automatically end-to-end in the algorithm. The network consists of a feature extraction layer and a classification layer using ADAM as an optimizer function and batch normalization. The feature extraction layer in turn consists of a convolution layer, activation layer, max pooling layer and the classification layer of a dense layer. The activation functions used was ReLu in the network and SoftMax at the end for generating a probability of class. This resulted in an accuracy of 96%.

In the [19] a Deep Learning algorithm was proposed to monitor the health of the bearings in wind turbines using time-series data and an LSTM-Autoencoder. The goal of the report is to develop an anomaly detection algorithm that can warn before a critical failure is occurring. The data was collected from 4 wind turbines in the West African Golf of Guinea in 2016. Three different types of data were collected; failures, alarms, and operational data from sensors which results in 83 features and over 20 700 samples. The data was loaded and pre-processed in the form of Min-Max scaling which transformed the features to a range of 0-1. Due to the abundance of features, PCA was performed to reduce the features and examine correlation. This data was then loaded into an LSTM-autoencoder which will try to identify the essential elements of the data. This is done on the healthy state of data and thereby if the autoencoder fails it can be assumed that the data provided isn't in a healthy state. The reasoning for choosing LSTM is due to its ability to remember data making it applicable for time series data. Utilizing the principal components made for poor performance so coefficients for all the features were considered and the highest was used. It was discovered that the more time-steps the model was allowed to use the better performance but it came with a cost, which is the training time of the model. The best accuracy achieved was at 83 %.

In the [20] a novel method is proposed to predict an RUL on a set of bearings. Since bearing faults usually generate periodic impulses but suffer from slippage when the bearing runs under load. So a spectral correlation technique was chosen to analyze the vibration signal. Performing a two-dimensional Fourier Transform allows it to distinguish bearing fault signatures from the interference. A Wasserstein distance is used to distinguish two probability distributions but can also transform one probability distribution into another, combined with a linear rectification to reduce the fluctuations over the degradations of the bearings. Through these steps, the model can show deviations from the modulations since the growth of the modulations is a characteristic of bearing failures. To connect this to RUL the result from the previous computation is divided into two stages; healthy and unhealthy, by the 3σ criterion mbased method. In order to calculate RUL a GRU network is built which is a refined version of RNN and a lighter version of LSTM, through its gate; update and reset. This decides how much of the historical information learned will be remembered moving forward and it solves the issue of vanishing/exploding gradient descent that RNN is suffering from. One of the major issues with using an ANN is the hyperparameters which require time and expertise to be manually tuned so a Bayesian optimization algorithm is deployed.

In [21] a statistical approach for identifying bearing faults is proposed through a theoretical model combined with a data-driven. A simulation model is built in order to generate the vibration data due to the lack of publicly available data. One of the issues is that the vibration signals characterized for bearing faults aren't happening at a constant interval due to slipping and sliding within the bearing. Therefore the envelope detection algorithm combined with the Hilbert-Huang transform is chosen. By applying this the small time differences become invalid and by analyzing the data in the time domain through spectrum analysis it's possible to identify the specific faults i.e. an inner ring, outer ring, etc. Another approach was displayed using Kurtosis, by calculating the Kurtosis for higher frequencies it was shown to be a clear distinction between healthy bearings and unhealthy bearings.

In the article, [22] both time domain features and frequency domain features were extracted from vibration data generated from a bearing. PCA was then performed to extract the principal components to reduce redundancy in the features. Since degradation of bearing health is time dependent an RNN was chosen due to its ability to remember distant information. However, RNN suffers from vanishing/exploding gradient so LSTM was chosen instead of traditional RNN. The LSTM predicts gradual degradation. Data from

Xi'an Jiatong University is used for the model which records vibrations from a healthy state to failure through an accelerometer. The features generated within the time domain were RMS, variance, absolute average value and within the frequency domain the variance together with the average variance. Only the first principal component was then used since it explained the original data with 93.58 %. The model generates an RMSE of 9.0 so it is concluded in the report that PCA is a useful tool in reducing dimensionality while still keeping the error low and LSTM is useful when predicting bearing degradation trends.

A larger part of the faults in induction motors is due to faults in the rotor and within those, the main culprit is the bearings. This is caused by dust and corrosion, in the article [23] an approach is proposed to identify bearing faults in induction motors. Even though vibrations are the most common metric when identifying bearing faults an alternative approach is proposed here, stator current harmonics measurements. Both Park transforms and the Concordia transform uses induction motor stator current patterns processing and has been successful in induction motor faults. The results show that Park transforms performs better but have the disadvantage of relying on the speed sensor than the Concordia transform.

In the article, [24] a fault detection method is used based on both the vibration and acoustic signal through a Hilbert-Huang transformation. This is applied to a single-phase induction motor where both signals are collected and then passed through empirical mode decomposition which extracts the intrinsic model functions and can be fed into the Hilbert-Huang model. This model achieved to reach an overall accuracy of 74.75% classifying 5 different classes. The classes were healthy, contaminated lubrication, outer race fault, inner race fault, and ball damage

In the article, [25] a method was proposed to identify the low level of lubrication in the differentials in a Light Armoured Vehicle, LAV. This is done by collecting vibration signals from an accelerometer and then analyzing at which frequency different levels of lubrication can be identified. By displaying the vibration data in a power spectral density, PSD plot the different frequencies can be identified. To start the range was between 0 kHz to 51.2 kHz which was later narrowed down to 15 kHz to 24 kHz. By calculating the RMS value it was possible to identify if the differential is fully lubed, half lubed, or empty.

In the article, [14] it is argued the benefit of using Deep Learning instead of classic methods for fault diagnosis in rotary equipment. With the rise of big data, the need for manual feature extraction is limited since DL can map the input to output by extracting the necessary features independently. This allows for transfer learning that helps in solving highly imbalanced data where it isn't possible to access fault data in rotary machinery. By training the model on simulated data, and then partially retraining the model on actual real data it is possible to create models where data is scarce. An example of this could for example be in helicopters where the rotary machinery simply isn't allowed to fail.

In [26] three different options of signal acquisitions, existing sensors, test sensors, and lastly injection of signals were presented and compared for predictive maintenance. The first category using existing sensors focuses on monitoring the performance of the process, process-to-sensor, and the sensors. The main sensors include RTDs, thermocouples, and pressure sensors. The second category using test sensors meaning sensors that aren't built in but can be incorporated retroactively. This also includes wireless sensors which allow for deployment where cables aren't possible. These includes accelerometer, ambient temperature, pressure and humidity. The last category is the only active monitoring signal acquisition meaning that it doesn't require the process to be running in order to generate data as opposite of the two categories above. Instead of only reading data, this category injects a signal and measure the outcome to identify different faults. This is used to identify cable anomalies and sensor faults.

Article [27] presents a method for predictive maintenace of a bearing in the TATA steel process. The bearing doesn't have a sensor connected to it and can neither be heard nor seen therefore another way of predicting bearing health is deployed. The data that is fed to the model include set-up sheet, warehouse-sheet and two different sensor-based data sources for system tracking. This include features like roll diameter, and roll condition before and after, of the bush before and after. This data is logged every 4 weeks and is logged manually. Warehouse data on the other hand relates to procurement data like order date, batch ID, etc. Finally the two sensorbased systems IBA and EMASS produces process data like speed, distance current etc. Through domain expertise 13 features were chosen;Total Length, Scrap Length, Total Surface, Mean Tension, Minimum Tension, Maximum Tension, Median Tension, Skewness Tension, Kurtosis Tension, Standard Deviation Tension, RMS Tension, Remaining Bush Width Days, Roll Diameter. Between these three models were evaluated, PSLR, ANN and Random Forest, PSLR provided the best result through the lowest RMSE. It was concluded that PSLR are a preferred method when sample size are small with high dimensionality. It was also concluded that Random Forest and ANN should be re-evaluated when more sample data is available.

In the article, [28] a novel model is proposed using a Maximum Correlated Kurtosis Decomposition, MCKD with LSTM on vibration signals to detect bearing faults. An heuristic search algorithm namely Cuckoo Search is performed to identify the optimal parameter filter length and deconvolution period. Correlated Kurtosis takes advantage of the periodicity that usually occurs in bearing faults. MCKD selects the finite impulse response filter which maximizes the correlated kurtosis which yields a high kurtosis and thereby a periodicity. The time series is then denoised through deconvolution which is fed to the LSTM which yields an increase of 25% compared to using the original time series

In the article, [29] an approach for identifying bearings faults and predicting RUL is proposed. The model consists of using CNN and RNN, specifically GRU together. The identification of faults is done through classification and the RUL through regression. This report applies the method to a dataset of bearings in a high-speed train and it consists of data from an accelerometer and a tachometer. The RUL is calculated according the mathematical formula (2.3) where day_{first} is the day when fault first occur and $day_{failure}$ when the bearing fails.

$$RUL = \frac{day_{failure} - day_i}{day_{failure} - day_{first}} \tag{2.3}$$

The data is split into training, validation, and testing set, and each is normalized individually. The model reaches a result of 99.72% of accuracy and an MSE of 1.326 of the RUL prediction.

Article [11] presents a method to investigate the "black box" of an LSTM algorithm used in an RUL prediction based on the health of a bearing using Layer-wise Relevance Propagation, LRP. The RUL model is built on vibration data from a test bed and each sample consists of 20 480 data points. These time-series was then transformed within the time domain and frequency domain generating 35 features 2.4.1. These were split into batches of 15 and fed into a Neural Network that contained one LSTM layer and three Dense layers. The model predicted 4 different levels with MSE as the loss function and ADAM as the optimizer. After training the model had an accuracy of 90.07%. The LRP distinguishes which of the original features had the largest impact on the result and does this by layer-wise propagation. Looking at the whole model it was fairly clear that 3 features had major influence overall and when this is explored on a deeper level it is clear that different features influence each label differently. The LRP method is compared to SAto verify that LRP identifies the correct features, the most valued features were removed and the model was re-trained which showed that the accuracy decreased relative to the removal of features in the case of LRP which was better than SA. LRP also identifies that the model assigns more weight to the latter information. By applying LRP it is possible to identify which layer doesn't provide useful information and thereby it is possible to decrease the size of the model which makes it less computational heavy. This can be extended by analyzing which neurons are used in each layer and as shown in the report two hidden layers could decrease the number of neurons with 40% and 20% respectively. This action reduced the accuracy by approximately 1% but reduced the trainable parameters by approximately 40%. Lastly, a method of identifying questionable predictions were developed by looking at individual heatmaps of the relevance. This showed large number of pixels with a relevance over 0.1 on the wrongly classified samples. Setting a threshold of a maximum of three pixels with a relevance over 0.1 resulted on 80% of the wrongly classified samples being identified as questionable.

3. Method

To answer the research question and sub-sequentially the secondary question a quantitative approach was chosen. This is due to that data will be of a primary sort and be generated through an experiment since there is no publicly available data that fit the research question.

The research question:

1. How can we classify corrosion levels of bearings that are not affected by radial load without statistical pre-processing of vibration signals using Artificial Neural Networks?
2. How can we use the pre-defined corrosion levels to estimate the Remaining useful life of the bearing in terms of time?

As presented in chapter 2 the majority of the research on the topic of condition monitoring of bearings uses a secondary source of data due to multiple publicly available datasets related to bearing failure. These datasets however are not related to corrosion but other failures which meant that the data of the sort needed for this project wasn't publicly available and thereby generated within this project. The project didn't have access to any domain knowledge of the bearings or the process and nor was it necessary for the success of the project. There are also many well-established ANN algorithms within this field that produces measurable metrics that can be used to analyze bearing health.

A qualitative approach was unfortunately not an option since research papers were scarce on the topic and no domain knowledge was available. It would have been possible to gain access to this in the form of interviews but that would have warranted multiple contacts as subjects for interviews which wasn't feasible. The overall plan can be concluded as generating data by corroding bearings to different levels, collecting vibration data and finally using said data to predict the different levels of corrosion based on vibrations. Based on these corrosion levels a prediction of the RUL can be estimated combined with the input such as corrosion depths and corrosion rate.

3.1 Data collection

Data is generated specifically for this project through a multi-step process.

1. Corrode bearings to different corrosion levels
2. Collect vibration signals for each level
3. Classify each corrosion levels

Therefore the data is from a primary source and the independent variables are managed throughout the project.

3.1.1 Corrosion of bearings

Corrosion of refined metals can differ immensely depending on the environment. Since these bearings are of the material 51200-Chrome steel they are treated in a way that hinders corrosion compared to carbon steel through the incorporation of chrome. Adding to that factor that they are also pre-greased increases the corrosion resistance. Therefore letting them corrode at a normal rate was not an option due to time constraints in the project so to speed up the corrosion a type of acidity was chosen. It was a two-step process, first, a type of acetic acid was used to make the bearings more susceptible to corrosion by ionizing them. Followed by a reactive oxygen species that in reaction combined with Sodium Chloride causes the sped-up corrosion rate. This approach was chosen by comparing different methods related to time, corrosion rate, and feasibility through trial and error.

3.1.2 Collection of vibration signals

An accelerometer collected the vibration data by measuring the acceleration in X, Y, and Z directions.

An accelerometer always provides a value larger than zero in one or more of the directions due to gravity in the opposite direction of gravity. In free fall on the other hand the accelerometer would read zero in each of the directions. Therefore the data will show one variable in the range of approximately 1000 mm/s. A mechanical accelerometer was used due to the need for high-frequency readings due to small movements through vibrations. The test location remained fixed in order to not create bias in the dataset through external vibrations and characteristics including the design of the setup.

3.2 Model

The model consists of an LSTM network that extracts and selects features and then classifies them into levels of corrosion based on the vibration data from the sensors.

3.2.1 Pre-processing

Due to the choice of using primary data in the project, it can be optimized to fit the task, here classification. Meaning that it's possible to limit the pre-processing to a minimum by having a fixed number of samples for each of the different classes. Imbalanced data isn't an issue and thereby neither up-sampling nor down-sampling is needed since the data is generated in a controlled environment.

The data is re-scaled and there are two ways of doing this, standardized and normalized. The reason for re-scaling is due to the difference in the ranges of the features since the data consists of windspeed and three-axis acceleration and it is not known in which direction the acceleration is larger if any. The measurement units for acceleration and windspeed are also different which could prove to have different ranges.

Normalized re-scaling of the dataset arranges all values between 0 to 1 3.1. While standardization on the other hand re-scales the dataset to have a mean of 0 and a standard deviation of 1 3.2 which was chosen in this project through trial and error.

$$x_{norm} = (x_i - x_{min})/(x_{max} - x_{min}) \tag{3.1}$$

$$x_{stand} = (x_i - \bar{x})/s \tag{3.2}$$

Where:

- x_i = The i^th value in the dataset
- x_{min} = The minimum value in the dataset
- x_{max} = The maximum value in the dataset
- s = The sample standard deviation
- $\bar{x}$ = The sample mean

ML networks usually struggles with categorical labels so a method called one-hot encoding is deployed which transforms the labels to binary format. It is structured such as a label is '0' in each place and '1' in the place of the correct label. Each corrosion level including the baseline are therefore transformed to this encoding.

Selecting the correct features for the model is a crucial part of pre-processing. Often this demands domain knowledge but by deploying ANN, the network itself is generating the different features and selecting the ones needed to perform classification.

3.2.2 Model

Artificial Neural networks have been applied in a variety of constellations but for time series, it is mainly one algorithm that stands out, LSTM. The network consists of different characteristics which will be explored in this chapter.

Layers

There are three variants of layers, input, hidden, and output. All NN has input and output, depending on how many hidden layers there are in the network it either constitutes a shallow network or a Deep Network. The latter is defined as having three or more hidden layers. An ANN is grouped together in layers that consist of multiple neurons and can be of different types throughout the network depending on the requested outcome.

Dense layer is the simplest and consists of fully connected neurons meaning that each neuron in the previous layer is connected to each and every neuron in the next layer.

LSTM layer is a type of RNN that targets the datatype time-series due to the ability to remember historical data through its hidden state within the cell structure which is used instead of the classical neuron as for the dense layer.

Dropout layer is a regularization layer used to prevent over-fitting meaning reducing the risk of memorization and forcing the algorithm to generalize. It does this by randomly forgetting part of the learned parameters.

Neurons

The decisions are made in the neurons, which can be thought of as mini-processing units. They get inputs from the previous neuron, which originally comes from the input layer. They then make decisions if this information should be passed forward or not. The input to the Neuron is input x, weights w, bias b and the output is y as is highlighted in 3.3 after passing through an activation function. As can be seen in figure 2 there could be multiple inputs to each neuron. The neural network handles this by assigning a weight for each input but keeps only one bias. In 3.4 it is shown how neurons handle multiple inputs as in a dense layer.

$$y_{single} = x * w + b \tag{3.3}$$

$$y_{mult} = x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + x_4 * w_4 + ... + b \tag{3.4}$$

LSTM layer doesn't have regular neurons but a cell structure instead. In formulas 2.17-2.20 the mathematical formulas behind LSTM are displayed where $f_i^{(t)}, s^{(t)}, g_i^{(t)}, q_i^{(t)}, h_i^{(t)}$ are the forget gate, cell state, input gate, output gate and hidden state respectively and calculated from the current input and previous output $x^{(t)_j}, h^{(t-1)_j}$. Both the input weight and recurrent weight are updated independently as well as the bias for each of the gates and state units with the activation function sigmoid [11]. This way the LSTM cell can decide to what extent let values in and out by activating the different gates. This solves the issue with exploding/vanishing gradient by having the cell decide on what to let in and out.

$$f_i^{(t)} = \sigma(b_i^f + \Sigma_j U_{i,j}^f x_j^{(t)} + \Sigma_j W_{i,j}^f h_j^{(t-1)} \tag{3.5}$$

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma(b_i + \Sigma_j U_{i,j} x_j^{(t)} + \Sigma_j W_{i,j} h_j^{(t-1)} \tag{3.6}$$

$$g_i^{(t)} = \sigma(b_i^g + \Sigma_j U_{i,j}^g x_j^{(t)} + \Sigma_j W_{i,j}^g h_j^{(t-1)} \tag{3.7}$$

$$q_i^{(t)} = \sigma(b_i^o + \Sigma_j U_{i,j}^o x_j^{(t)} + \Sigma_j W_{i,j}^o h_j^{(t-1)} \tag{3.8}$$

$$h_i^{(t)} = tanh(s_i^{(t)}) q_i^{(t)} \tag{3.9}$$

Weights and biases

The weights and bias tell the neuron how much value it should put on the input which is combined with the output of the previous neuron. These are regression parameters and are updated during the training process if the network utilizes backpropagation. When the model has initialized these weights and biases are assigned randomly.

Backpropagation

The method where the network sends information about the error rate backward through the network. It is through this the gradient descent can be calculated at each weight and bias.

Activation function

If the Neuron should fire or not is decided by the activation function depending on the weights and biases together with the input from the previous neuron. It also ensures non-linearity thereby it can learn relationships that aren't linear. The input to the activation function is the sum of the input of the neuron as shown in 3.4. There are numerous amount of types and each has its own strengths but the most common is the ReLu (3.10) or its sibling Leaky ReLu (3.11). Other common functions are sigmoid (3.12), tanh (3.13) and linear (3.14) where $\bar{z}$ is input from the previous neuron combined with the weight and bias and α is a small value in order to not have zero [29].

$$f(x)_{ReLu} = max(0, \bar{z}) \tag{3.10}$$

$$f(x)_{LeakyReLu} = max(\alpha\bar{z}, \bar{z}) \tag{3.11}$$

$$f(x)_{Sigmoid} = \frac{1}{1 + e^{-\bar{z}}} \tag{3.12}$$

$$f(x)_{tanh} = \frac{e^{\bar{z}} - e^{-\bar{z}}}{e^{\bar{z}} + e^{-\bar{z}}} \tag{3.13}$$

$$f(x)_{linear} = \bar{z} \tag{3.14}$$

For the output layer, a common variant for categorical classification is softMax, which takes as an input a set of scores where the highest is the predicted class. The issue is that these numbers are hard to interpret so what SoftMax does is that it takes those numbers which equal the number of classes the model is trying to predict and converts them into a probability range using this formula.

$$Probability_{SoftMax} = \frac{e_i^x}{\sum_{i=1}^{N} e_i^x} \tag{3.15}$$

Loss function

The metric to calculate the distance between the predicted output and the true output. This will tell us how well the model is performing and the model will aim to minimize this loss by updating the weights and biases in the direction of gradient decent. There are a multitude of options and each fit different tasks. For classification, the two main ones are the cross-entropy, binary(3.16) and for multiple classes (3.17). For regression on the other hand Mean squared error (3.18) or Mean absolute error (3.19) are often chosen.

Where N, i is the total number of samples with the sample index and $y_i, \hat{y_i}$ is the ground truth with the predicted value.

$$L(q)_{BinaryCE} = -\frac{1}{N}\Sigma_{i=1}^{i=N} y_i * log(\hat{y_i}) + (1 - y_i) * log(1 - \hat{y_i})) \tag{3.16}$$

$$L(q)_{CategoricalCE} = \Sigma_{i=1}^{i=N} y_i * log(\hat{y_i}) \tag{3.17}$$

$$MSE = -\frac{1}{N}\Sigma_{i=1}^{i=N}(y_i - \hat{y_i})^2 \tag{3.18}$$

$$MAE = -\frac{1}{N}\Sigma_{i=1}^{i=N}|y_i - \hat{y_i}) \tag{3.19}$$

Optimizer

The main purpose of the optimizer is to minimize the loss by updating the weights and biases They do this by calculating the gradient decent to find the global minima and thereby the lowest loss. There is the issue of local minima which the optimizer needs to ignore but sometimes ends up in without any ability to recover from on its path to the global minima. The most common ones are Stochastic gradient descent, SGD, and Adaptive Moment Estimation, Adam. The main difference between the two is that ADAM has adaptive learning rates for each parameter while SGD has the same for all [30]. This makes ADAM more flexible since it performs larger updates on uncommon parameters and smaller on common parameters. ADAM also includes a sort of momentum allowing for it to take larger steps similar to SGD Momentum. ADAM uses exponentially decaying average of past squared gradient (3.20) and exponentially decaying average of past gradient (3.21) but due to bias to zero these were corrected in (3.22) and (3.23) where β_1, β_2, g_t are exponential decay rate for the first and second momentum respectively and the gradient vector. In (3.24) the ADAM update rule is defined where Θ, η, ϵ are the update, learning rate, and a small number to prevent division by zero.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.20}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.21}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.22}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.23}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{3.24}$$

3.2.3 Remaining useful life

The most common for estimating RUL is by comparing the day of the first fault to when the bearing fails as in [29]. This demands data from the beginning of the product life cycle to failure which might not be available. To estimate the remaining useful life without the bearing's whole lifespan the corrosion rate needs to be defined. A way to calculate corrosion rate is to measure how far the corrosion has reached by looking at the weight loss, density, area, and time. The corrosion depth at each level can be measured by looking at the weight loss and when through domain knowledge a corrosion depth can be defined as a broken bearing.

This allows for connecting each corrosion level with a corrosion depth and since it is known at which depth the bearings are considered broken it is possible to estimate a health index for each level. Combined with the corrosion rate this is possible to estimate the remaining useful life.

3.3 Evaluation

The performance of the model is judged through the prediction of classes by the model on previously unseen data. This way the model doesn't have the chance to memorize the characteristics of the training data but is forced to learn the general characteristics of the classes.

The predicted class is compared with the actual class for each sequence in a confusion matrix 5, which shows if the model classified the sequence correctly or not. This is however not the only approach to evaluating the performance of the algorithm, common metrics like accuracy, and recall are also used. In 5 there are 4 different variables that need to explain in order to calculate evaluation metrics. Positive and negative in the case of binary evaluation are '1' and '0'.

True positive is when both the predicted value and the ground truth are the same, a positive value in the case of binary evaluation. **True Negative** is when both the prediction and the ground truth are negative i.e. the same as TP but for the negative value. **False Positive** is when the prediction is positive but the ground truth is actually negative. **False Negative** when the prediction is negative but the ground truth is positive i.e. the opposite of FP.

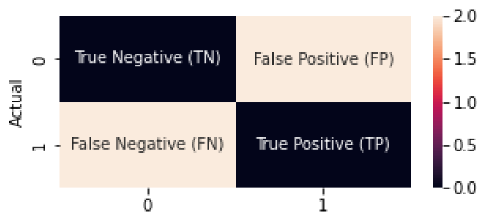


Figure 5. *Binary confusion matrix*

Accuracy is the most common metric and it gives a good indication of the model's performance but it does not show the whole truth. As can be seen in the formula 3.25 the total correct classifications are compared with the total number of predictions. But there are caveats that shouldn't be forgotten, this is a model that works well when the dataset is well-balanced. Unbalanced data on the other hand could lead to good accuracy even if it only predicts everything in the majority class, for example in anomaly detection. The data is well-balanced since it is primary which makes accuracy a great way to check the performance of the model.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.25}$$

Sensitivity and recall is the measurement often used when interested in a specific class. As can be seen in 3.26 it measures how often a class is correctly classified divided by the total amount of said class. In the context of PdM, the class of possible issues could be the main interest in for example a flight engine where the cost is too high to take any risks.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.26}$$

Specificity is the inverse of Sensitivity due to it only looks at the times the model classified a class wrong as can be seen in 3.27.

$$Specificity = \frac{TN}{TN + FP} \tag{3.27}$$

Precision is the total number of correct predictions compared to all the predictions for that class as can be seen in 3.28. It is therefore a measurement of how well the model predicts a class no matter how many are wrong of a different class.

$$Precision = \frac{TP}{TP + FP} \tag{3.28}$$

F1-score is the last of the common metrics and is a combination of sensitivity and precision. F1 is commonly used when there is an imbalanced dataset with no extreme values such as neither precision nor sensitivity is zero because then F1 will also be zero as can be read from the formula 3.29.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.29}$$

3.4 Software

The following software tools have been used for the duration of the project:

- Programming language: Python 3.10.2
- Programming environment: Visual Studio Code
- Main libraries: Tensorflow-Keras, Numpy, Pandas, and Sklearn
- Operating system: Pi, Ubuntu, and Windows

Respectively the following hardware tools have been used:

- Raspberry Pi 4B
- Davis Anemoemter 6410
- Steval STWINKT1B
- Digicom 4G Literouter Plus

4. Project

This project is part of the European Research project Arrowhead and developed at Reply Santer who partakes in multiple sub-projects. The role of this project relates to developing a use-case where the tools developed in Arrowhead are show-cased. Since this project is part of a larger project, some parts like the test-bed setup are inherited.

This project aims to highlight the possible usage of Arrowhead framework through the use case that will be explored in this chapter. The different sub-goals are defined in the list below.

1. Corrode bearings
2. Identify anomaly from normal operation
3. Identify corrosion levels without pre-processing vibration analysis
4. Predict intermediate levels previously unseen by the algorithm.
5. Based on corrosion level, predict the percentage of corrosion on the bearings and the RUL.

4.0.1 Arrowhead

Arrowhead is an open source framework that provides an architecture for building automation systems using IoT. Arrowhead framework enables interoperability between the majority of IoT devices and their respective protocols available in the market today [3]. The number of sensors and actuators deployed has grown exponentially lately, and the systems managing these have grown in unison to handle the added devices.

The idea behind Arrowhead is to utilize local clouds 6 that allow for:

- Real-time data handling
- Data and system security
- Automation system engineering
- Scalability of automation systems

The core systems are the service registry system that allows the provider to publish the data and the consumer to find the data. The authorization system allows the service provider to select which consumers can access the data, and finally the orchestration system allows the provider to decide which kind of data the consumer can access 6.

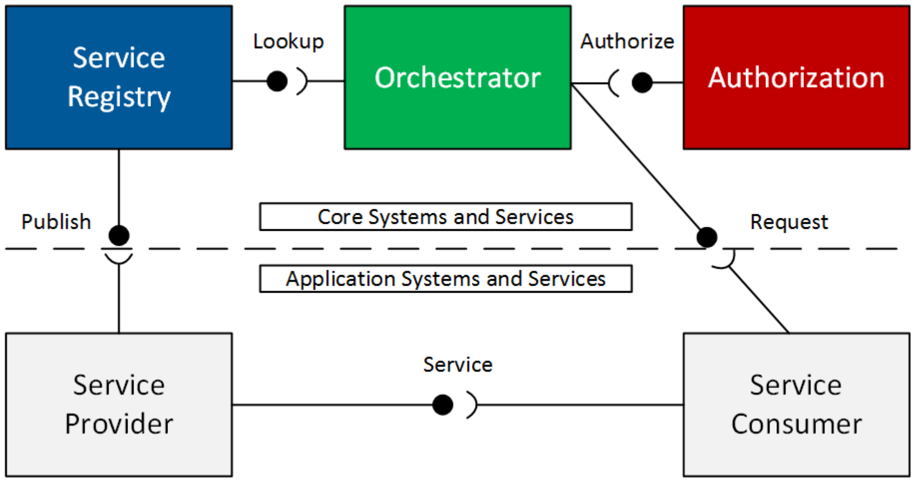


Figure 6. *Local cloud architecture [31]*

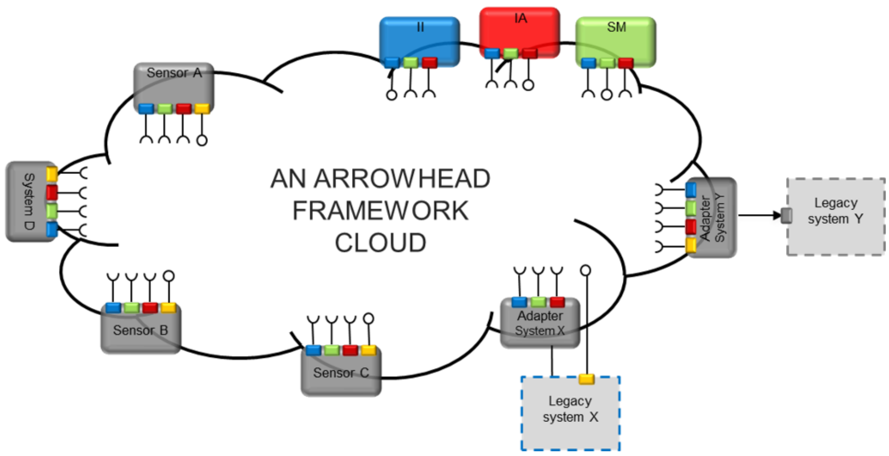


Figure 7. *Arrowhead framework cloud [31]*

This allows the legacy structures to be incorporated and communicate in a safe and fast manner as can be seen in 7. The project started in 2017 and consists of 5 pilot domains where demonstrators within the project will be made to show use-cases. As highlighted in the list above this project is connected to workpackage 2 with the subcategory of predictive maintenance.

- WP1 - Production
- **WP2 - Smart buildings and infrastructure**
- WP3 - Electromobility
- WP4 - Energy production and end-users services
- WP5 - Virtual market of energy

4.1 Test-bed

The test bed consists of the anemometer of the brand DAVIS, two Raspberry Pi 4B, and a sensor of the brand STEVAL. It was decided on the anemometer since it is a common device in the smart city context and the sensor is common within condition monitoring. The bearings in this anemometer operate with no radial force and minimum axial force, due to the wind-cups hanging in the bearings. This setup is not that common in the context of predictive maintenance, since most cases are based on bearing with radial force as has been shown in chapter 2.5. The reasoning for using an anemometer lies within the previously mentioned work package two from Arrowhead, Smart buildings, and infrastructure. Using an anemometer that is actually deployed in a smart city context connects this project to a real-world application.

Figure 8. *Davis anemometer.*

The anemometer can measure the wind speed and the wind direction using two separate parts. The wind direction is not of interest for this project since it doesn't use the bearings and will therefore be omitted moving forward, and it was detached from the test-bed. The sensor is attached at the optimal location in relation to the bearings in order to get as correct readings as possible. Attaching the sensor to the anemometer was solved through 3D-printing an intermediate piece that was attached to the sensor mechanically and to the anemometer through zip-ties that allows for easy removal as can be seen in 9

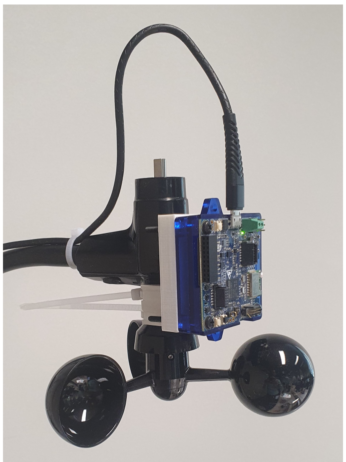


Figure 9. *Davis anemometer with Steval sensor.*

Both the Steval and the anemometer were further on connected to the raspberry pi through a USB cable from the Steval and a GPIO connector from the anemometer. Allowing the raspberry to read data from both the Steval and the anemometer. The raspberry pi then transmitted the pre-processed data through Bluetooth to a new raspberry that acted as a gateway and converted the low-energy protocol to an MQTT protocol so it could be published to the Vital IoT. The dashboard then subscribed to the sensor through Vital IoT and was able to visualize the data at any location.

Figure 10. *Raspberries for data management and gateway.*

- DAVIS 6410 - Anemometer that measures the wind direction and wind speed up to 89 m/s.
- STEVAL STWINKT1B – is an IoT device with multiple embedded sensors allowing for measuring a multitude of variables. Those sensors are an accelerometer, magnetometer, gyroscope, humidity, pressure, temperature, and sound. It can communicate through a variety of protocols like WIFI, Bluetooth, and Micro-USB. It is powered through either the USB cable or with a stand-alone battery.
- Raspberry Pi 4B – A mini-computer with a high level of connectivity due to its many interfaces, USB, Ethernet, Mini-HDMI and GPIO.
- Wind generator – Hairdryer of Philips with two-speed settings with an advertised

maximum wind speed of approximately 36 m/s.

- Extra sets of bearings for corrosion. The bearings came from the manufacturer pre-greased by Kyodo Yushi Multemp SRL grease and are of material 51200 chrome steel.

4.2 Data collection

The anemometer counts each lap of the wind-cups and in order to convert this to velocity with the unit m/s the following formula is used. According to documentation from Davis instruments the conversion rate is 1600 cycles = 1 mph [32]. This means that a window of time needs to be fixed which is decided as 10 seconds to allow for multiple cycles of the wind cups.

$$Speed = ((C * D)/T) * S \tag{4.1}$$

Where:

- C = Cycles
- D = Factor from Davis instrument
- T = Time in seconds,
- S = Conversion to m/s.

The Steval is configured to send signals to the raspberry with a rate 50 Hz and since the time frame was set to 10 seconds each sample resulted in 501 data points. These readings were combined and stored in a CSV file locally on the raspberry. Two scripts were created to read the data, one for the anemometer and one for the Steval.

4.2.1 Data collection

The anemometer was dissembled and the original bearings were replaced with a new set so the type of bearings nor the age didn't influence the prediction. First, the baseline was acquired according to the acquisition process, and thereafter the different corrosion levels. Each level of corrosion was completed from start to finish including data acquisition before starting the next level.

Corrosion process

Due to fumes during the corrosion process, this had to be conducted in a well-ventilated area and was therefore moved outside. Environmental variables affected the corrosion rate so in order to combat the variability a few independent variables were chosen before the corrosion process could start. The data collection only took place when the environmental variables were verified and then a new mixture was created according to the recipe below. The process is defined in two steps, first cleaning the bearings in vinegar and then allowing them to dry. The second step involved dipping the bearings in the mixture followed by resting, this allowed for the corrosion to develop and was repeated depending on the level 3.

- Variables
  - Temperature: approx 30 degrees
  - Humidity: approx 72 %
  - Sunlight: Direct and continuous
  - Position: Horizontally, allowing for airflow
- Mixture
  - 150 ml Hydrogen Peroxide 3%
  - 20 ml Vinegar 6%
  - 0.5 teaspoon of salt
- Process
  - V&R - Vinegar & Rest
    - Dip bearings in clean vinegar
    - Rest in direct sunlight for 10 minutes
  - M&R - Mixture & Rest
    - Dip in mixture
    - Rest in direct sunlight for 10 minutes

After the corrosion process was completed the bearings were immediately installed in the anemometer and the data acquisition started since the corrosion continues even after the process above was finished. The following levels were decided on as can be seen in table 3.

Table 3. *Bearings level based on corrosion*

Level	**V&R**	**M&R**	**M&R**	**M&R**	**M&R**	**M&R**	**M&R**	**M&R**	**M&R**
Baseline	-	-	-	-	-	-	-	-	-
Level 1	OK	OK	-	-	-	-	-	-	-

Continues...

Table 3 – *Continues...*

Nr	**Requirement**	**Importance**							
Level 2	OK	OK	OK	-	-	-	-	-	-
Level 3	OK	OK	OK	OK	-	-	-	-	-
Level 4	OK	OK	OK	OK	OK	-	-	-	-
Level 5	OK	OK	OK	OK	OK	OK	-	-	-
Level 6	OK	OK	OK	OK	OK	OK	OK	-	-
Level 7	OK	OK	OK	OK	OK	OK	OK	OK	-
Level 8	OK	OK	OK	OK	OK	OK	OK	OK	OK

Data acquisition

After installation the anemometer was allowed to run for 10 minutes at approximately 5 m/s for the bearings to "normalize" so the readings would be relatively steady. Meanwhile the speed was to calibrated to be 5 m/s with a margin of +-0.5 m/s. The data collection started with reading the speed during 10 seconds which was then reused for each sample. The number of samples was decided on 1 000 resulting in 500 000 data-points. After the collection at 5 m/s hereafter referred to as low-speed the speed was again calibrated and this time to 10 m/s hereafter referred to as high-speed and the data collection started again. This was repeated for each data collection process. By collecting data at two different speeds the algorithm becomes less vulnerable if different wind-speeds are presented during the prediction.

4.3 Condition monitoring

The raspberry now contained 9 classes of data consisting of wind speed and acceleration in three directions. Each class had two wind speeds and a total of 2 000 samples and 1 million data points. This data was exported from the raspberry to Google Colab for training the algorithm. The objective of this project was to predict each of the 9 classes mentioned above, and since the data is in the format of time series an ANN of LSTM had been chosen as the predictive algorithm due to its ability to remember historic information.

4.3.1 Pre-processing

The collected data were pre-processed by splitting the data into training, validation and testing datasets with ratios is 70, 10, and 20 percent. The reason for this is that the algorithm actually learns the features and not only memorizes the time series. Each dataset is normalized independently and within each feature by using StandardScaler in Keras.

Checking the ratio of each class verifies that the datasets are balanced so no further pre-processing is needed. The labels are converted into one-hot encoding using the package sklearn.

4.3.2 Training

The python package Keras is used to train the network together The ANN architecture is displayed in figure **??** which can be divided into two different parts, feature extraction, and classification.

The input size for the feature extraction is (501, 4) due to the length of the time series and the number of features. The feature extraction consists of bidirectional LSTM layers due to these layers extracting features from two directions, and the first layer is of the size 1024. This is followed by the same type of layers with different sizes ranging from 512 down to 128 creating a Deep Neural network. After each LSTM layer, a Dropout layer is added in order to prevent over-fitting.

The classification layer consists of a dense layer with a size of 128 meaning that it is a fully connected layer with the activation function LeakyRelu. This is followed by another dense layer with size 18 and activation function SoftMax providing probability for each of the 18 labels.

Callbacks have been used in the training process to keep the algorithm from over-training. An early stop was used to monitor the validation loss with a min delta of 0.001 and patience of 50. Model Checkpoint is used to only save the model with the lowest validation loss which means that if the model is over-fitting it won't write over the previously lowest validation loss.

- Learning rate: 0.001
- Batch size: 128
- Epoch: 500
- Optimizer: ADAM
- Loss function: Categorical CrossEntropy

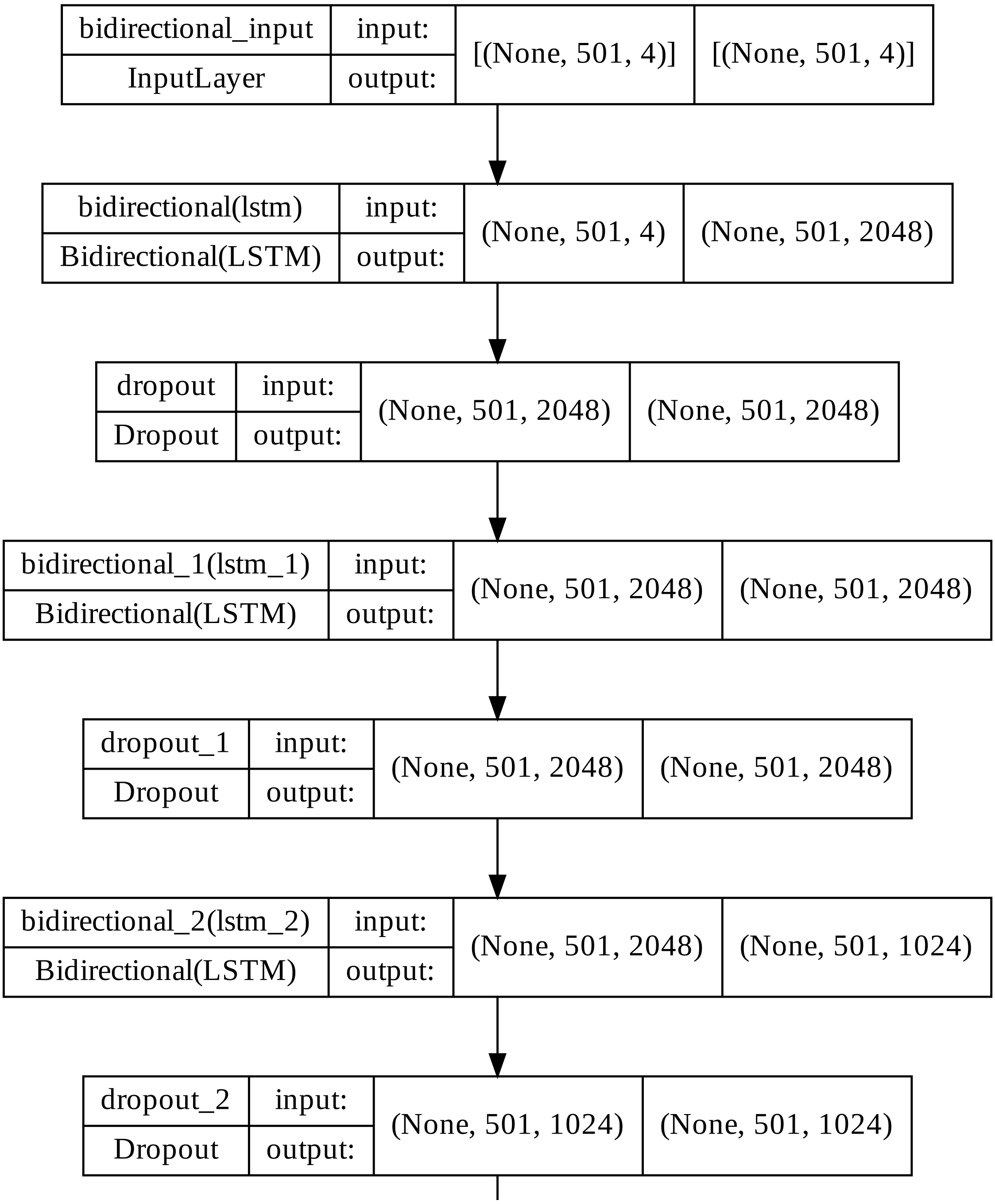


Figure 11. *LSTM network part 1.*

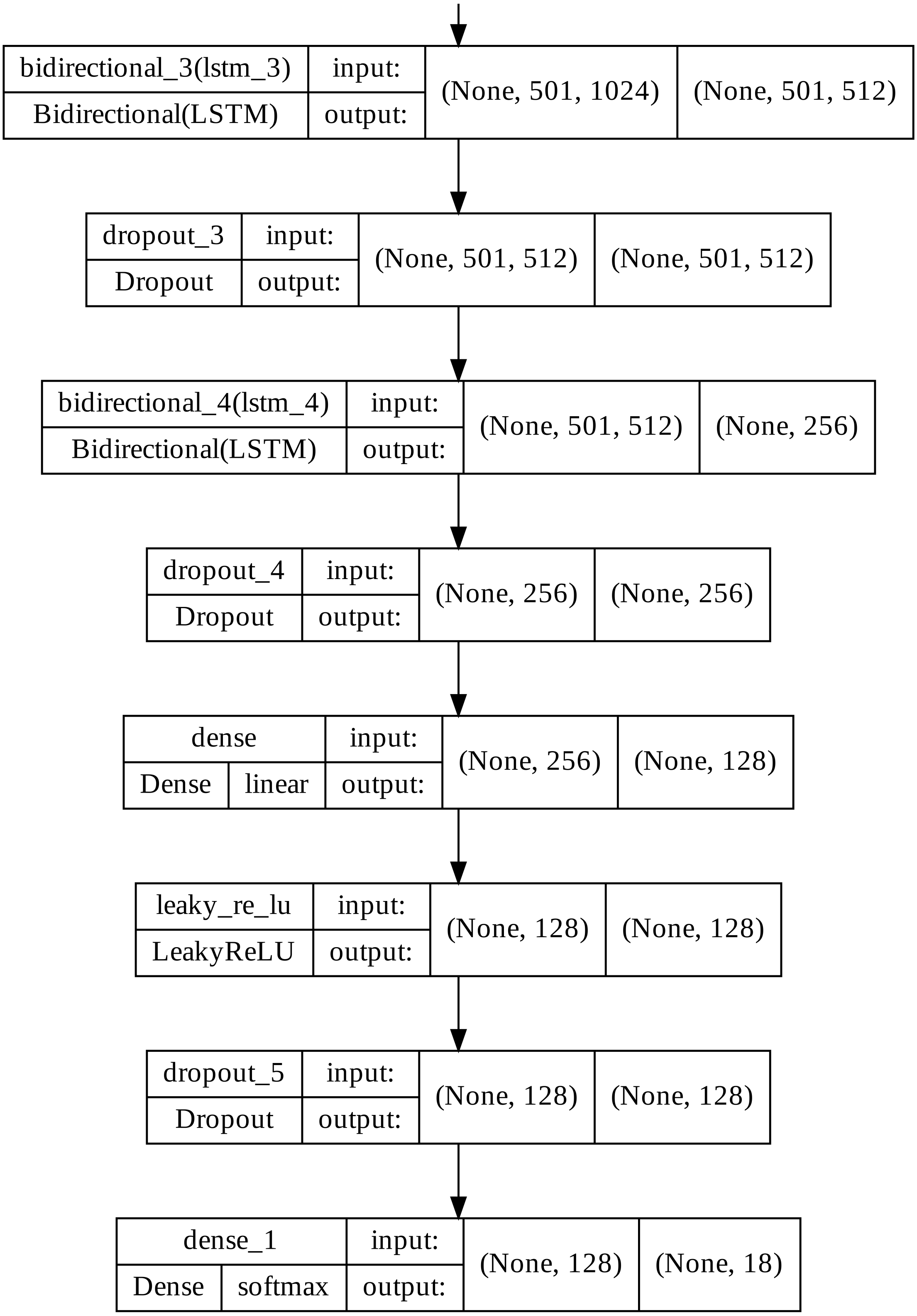


Figure 12. *LSTM network part 2.*

4.3.3 Remaining useful life

A common way to measure corrosion rate is to measure how far the corrosion has reached by looking at the weight loss. Unfortunately, this approach is not possible in this case since there are variables that can't be controlled like the bearing grease. If some of the grease falls off that would affect the weight loss immensely. Therefore a theoretical number will be used for calculating the corrosion rate.

This article uses the corrosion rate of 00.51 mm/year in an atmospheric marine environment for steel piling [33]. The bearings, in this case, are greased chrome-steel and not carbon steel which adds to corrosion protection, so estimation is made at 0.051 mm/year. Since the balls of the bearing have a diameter of 1.588008 mm it is assumed that if the corrosion has a depth of 0.5 mm then the bearing is broken. In this case level 8, with an estimated health level of 0% is deemed as a broken bearing. This is hypothetical and used as a proof of concept of this approach.

4.4 Result

The results of collecting the vibration data on corroded bearings and using that to create nine corrosion levels are presented in the following chapter.

4.4.1 Results

Figure 13 shows the accuracy and loss for both the training and validation sets. While training, the network works to minimize the validation loss. As shown in the training graph 13 the training loss and the validation loss dropped fairly quickly with only a few spikes which can be assumed to be local minimas. When the loss reached around 0.4, signs of over-fitting emerged as the training loss continued to decrease while the validation loss stagnated. The accuracy for both the training and the validation dataset followed each other up to 90% until they started to divert, which is a good sign since the over-fitting didn't affect the accuracy as much.

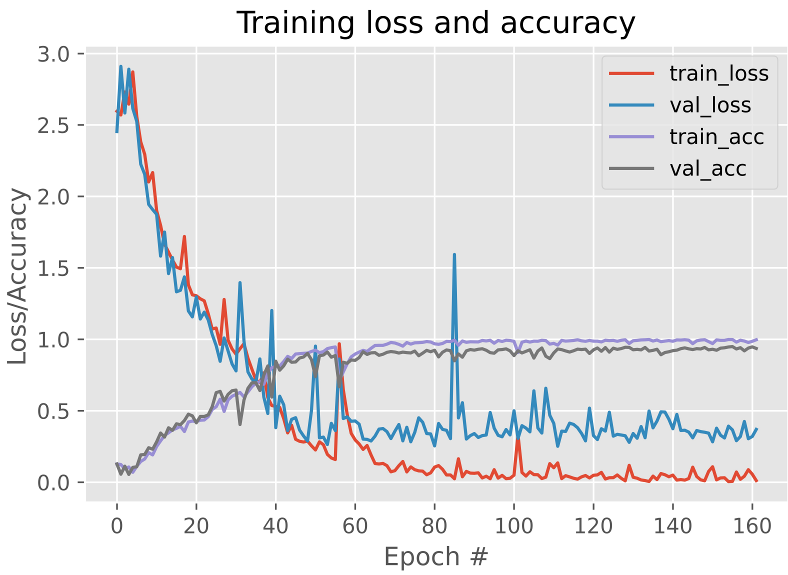


Figure 13. *Training with loss/validation loss and accuracy/validation accuracy.*

Making predictions using previously unseen data for the model with 200 samples per class generates the following classification report 4. This is the classification report of the model without it being trained, this will be used as a baseline, HS and LS respectively refer to High-speed and Low-speed.

Table 4. *Classification report baseline*

Class	**Precision**	**Recall**	**F1-Score**	**Samples**
HS Baseline	0.00	0.00	0.00	200
HS Damaged L1	0.00	0.00	0.00	200
HS Damaged L2	0.00	0.00	0.00	200
HS Damaged L3	0.00	0.00	0.00	200
HS Damaged L4	0.00	0.00	0.00	200
HS Damaged L5	0.03	0.01	0.01	200
HS Damaged L6	0.06	0.06	0.06	200
HS Damaged L7	0.00	0.00	0.00	200
HS Damaged L8	0.14	0.01	0.02	200
LS Baseline	0.17	0.01	0.03	200
LS Damaged L1	0.00	0.00	0.00	200
LS Damaged L2	0.00	0.00	0.00	200
LS Damaged L3	0.06	0.65	0.11	200
LS Damaged L4	0.00	0.00	0.00	200

Continues...

Table 4 – *Continues...*

Class	Precision	Recall	F1-Score	Samples
LS Damaged L5	0.05	0.25	0.08	200
LS Damaged L6	0.02	0.01	0.01	200
LS Damaged L7	0.11	0.09	0.09	200
LS Damaged L8	0.00	0.00	0.00	200
Accuracy			0.06	3600
Macro avg	0.04	0.06	0.02	3600
Weighted avg	0.04	0.06	0.02	3600

Comparing this to the classification report after the model has been optimized, it can clearly be stated that the model has evolved from the baseline.

Table 5. *Classification report*

Class	Precision	Recall	F1-Score	Samples
HS Baseline	99%	99%	99%	200
HS Damaged L1	100%	99%	100%	200
HS Damaged L2	99%	100%	99%	200
HS Damaged L3	98%	93%	96%	200
HS Damaged L4	100%	98%	99%	200
HS Damaged L5	93%	97%	95%	200
HS Damaged L6	92%	91%	92%	200
HS Damaged L7	87%	87%	87%	200
HS Damaged L8	97%	100%	99%	200
LS Baseline	97%	99%	98%	200
LS Damaged L1	99%	94%	96%	200
LS Damaged L2	89%	71%	79%	200
LS Damaged L3	90%	93%	92%	200
LS Damaged L4	75%	88%	81%	200
LS Damaged L5	90%	92%	91%	200
LS Damaged L6	97%	97%	97%	200
LS Damaged L7	82%	89%	85%	200
LS Damaged L8	98%	92%	95%	200
Accuracy			93%	2800
Macro avg	94%	93%	93%	3600
Weighted avg	94%	93%	93%	3600

In table 5 it is possible to read out Accuracy is 94% which indicates that the model classified each class correctly on average 94% of the time. Since the dataset is balanced, which can be read from the Samples column, we know that precision, recall, and f1-score will be similar, which the table confirms. It is also possible to see which classes the model had easiest to distinguish, High-Speed Damaged Level L1 and L4, for example. And which classes it struggled with, such as Low-Speed Damaged Level 4.

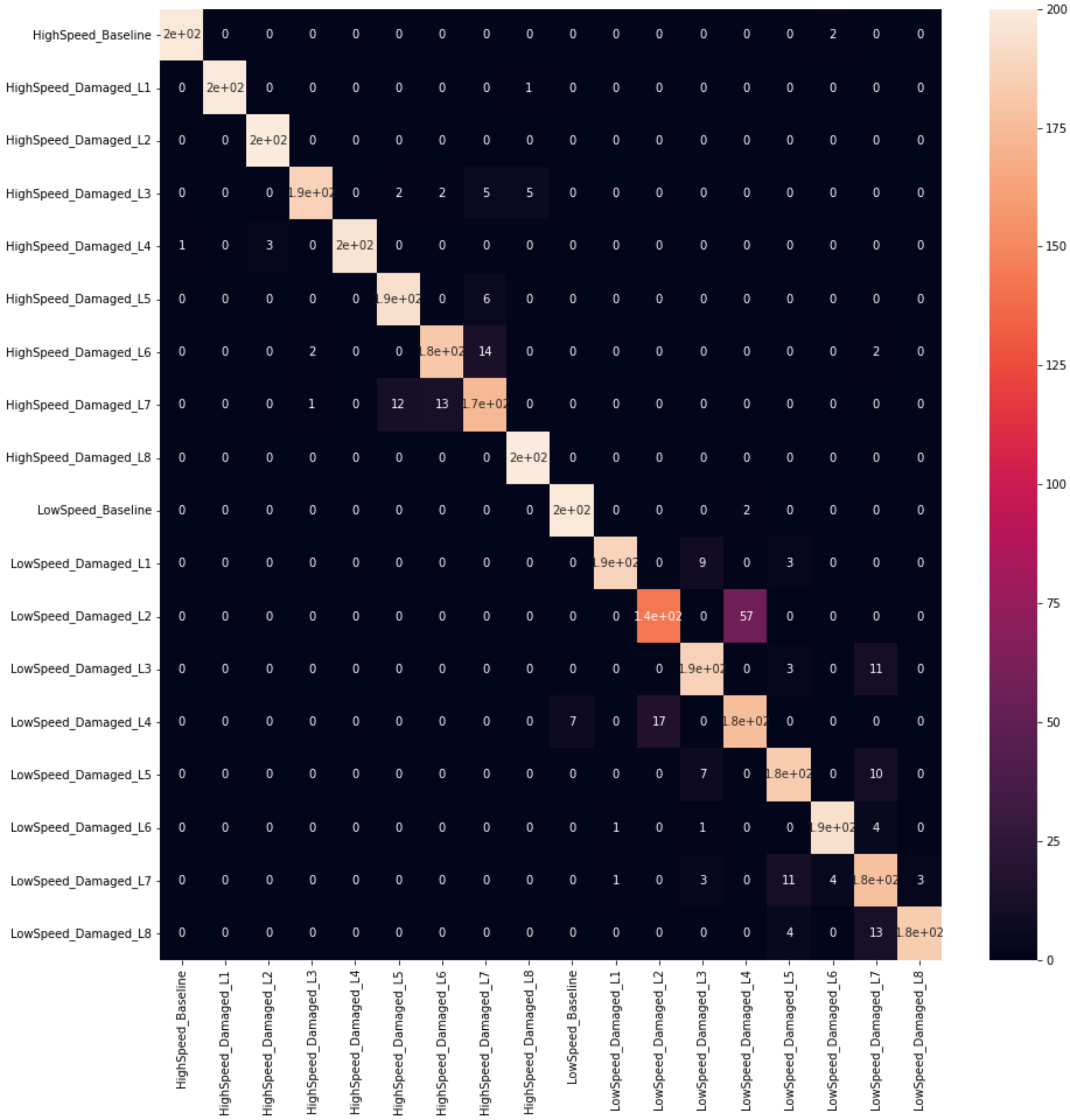


Figure 14. Confusion matrix heatmap

Figure 14 shows how the model classified each sample and the true class through color-coding with the true values on the y-axis and the predicted on the x-axis. Here it becomes even more clear how the model struggled with classifying Low-Speed Damaged L4 since it classified 57 of those samples in Low-Speed Damaged L2. This could be a bit hard to distinguish, so a way to highlight the dispersion is by converting it to a percentage like in figure 15. Even though the model, on average, produces an accuracy of 94%, five classes produce a result <90%.

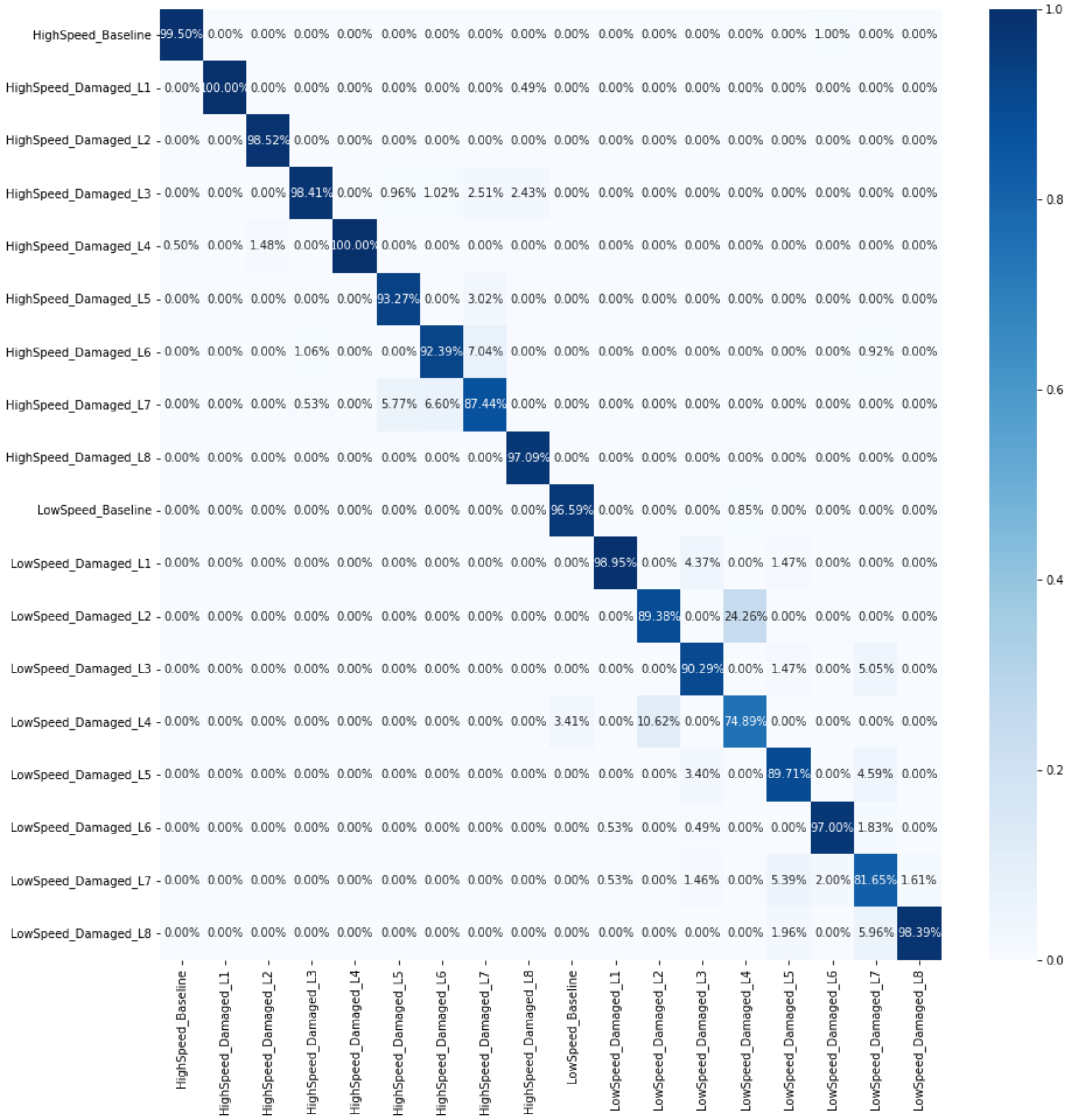


Figure 15. Confusion matrix heatmap in percentage

The RUL model can calculate the remaining useful life, providing it gets the correct data. In 6 the RUL for each level of corrosion is calculated by adding a theoretical corrosion depth and rate.

Table 6. *Bearings health level*

Level	Health level	Corrosion level	RUL in year
Baseline	100 %	0 mm	10 year
Level 1	90 %	0.06375 mm	8.75 years
Level 2	80 %	0.1275 mm	7.5 years
Level 3	70 %	0.19125 mm	6.25 years
Level 4	60 %	0.255 mm	5 years
Level 5	50 %	0.31875 mm	3.75 years
Level 6	40 %	0.33825 mm	2.5 years
Level 7	20 %	0.33825 mm	1.25 years
Level 8	0 %	0.51 mm	0 years

Since one of the objectives was to showcase Arrowheads tools, the predictions are presented in a dashboard as can be seen in **??**. This allows for access to live and historical data related to vibrations and wind speed by subscribing to the sensor. This data is displayed by the corrosion level, estimated RUL, and average accuracy.

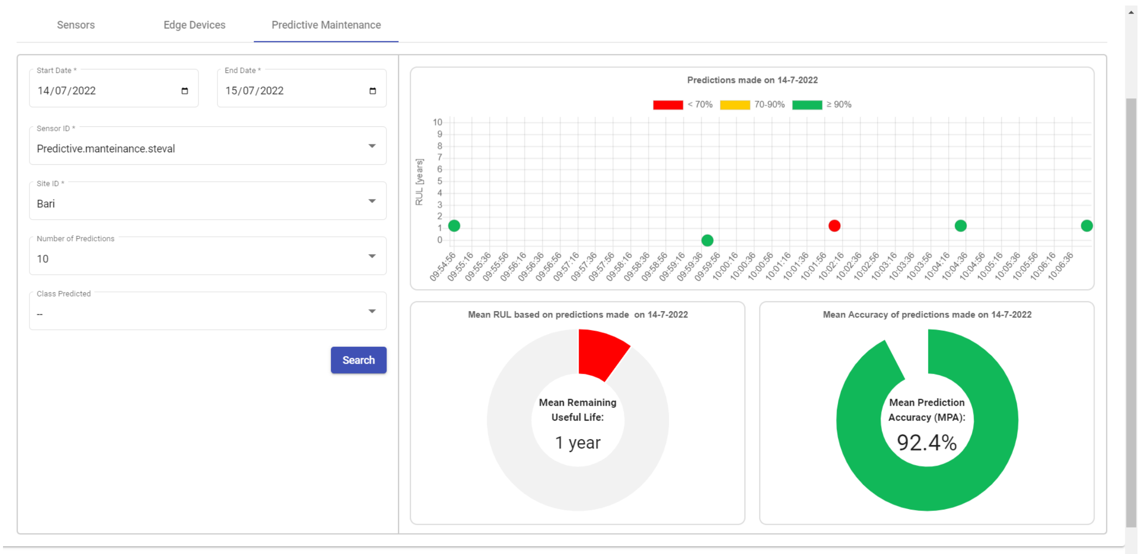


Figure 16. Dashboard

5. Discussion

The objective of this project was to deliver

1. Condition monitoring model
2. Classification model
3. RUL model

The proposed model can monitor the bearings' health by using labeled data in the format of different corrosion levels, including healthy bearings. With relatively high accuracy, the model can also classify the levels, which means that this model satisfies two of the above-mentioned deliverables. A proof of concept to calculate the RUL model has been developed, which satisfies the last deliverable.

The results from predicting previously unseen data prove that the algorithm is well equipped to identify classes it has been trained on. Thereby it is deemed by the author to be proven that this algorithm can in fact be used to identify different levels of corrosion as long as it has seen the data before.

The majority of the wrongly classified corrosion levels reside in the Low-speed spectrum as shown in 14 which leads to the conclusion that the algorithm performs worse at classifying Low-speed, especially at the higher corrosion levels. Since the wind speed was one of the fixed parameters we know that it wasn't due to slower cycles. Another possible cause for difficulties in distinguishing these could be the warm-up time. With higher corrosion levels, the warm-up might need to be longer. Since Low-speed was collected before High-Speed, this seems like a likely culprit. One aspect that has been noticed is the influence of the environmental variables in the corrosion process. For example, while performing corrosion it was identified that direct or indirect sunlight, as well as temperature, makes a big difference. This can propably be solved by using more data for training created under different scenarios. There is a risk that this might lower the accuracy of the model, but this could be handled by lowering the number of levels to for example healthy, functioning, and damaged. For the RUL, both the corrosion rate in mm/year and the corrosion level in mm are purely hypothetical. With the correct equipment, the corrosion rate can be calculated by measuring the level of corrosion in a specific time period. In that sense, this model would be applicable.

A restriction that this project had to adhere to is the limited resources available. Even though Google Colab Pro was used to train the model, it struggled to handle the size of trainable parameters and had limited training periods.

5.1 Further research

For future research, it is proposed to optimize the model in order for it to be able to identify intermediate levels of corrosion i.e. levels that it hasn't been trained on. This includes intermediate corrosion levels as well as different wind speeds. This could also extend into investigating different types of corrosion. The author proposes an environment with high humidity, salinity, heat, and direct sunlight with a longer data collection time frame to get a closer connection to the reality of decay of the bearing. Another interesting topic could be if different algorithms or combinations could be better at predicting intermediate corrosion levels such as CNN, combined LSTM, CNN, etc.

Due to large quantities of data and a deep network, larger resources can be used to improve performance of the algorithm.

For the RUL, this method could be replicated with an added approach of measuring the corrosion at each corrosion level when the data collection has finished. Then combining this with an experiment of letting a bearing corrode naturally for a period of time and then measuring the corrosion level would in theory generate a corrosion rate. This however only generates a linear corrosion rate and the true rate may be exponential. It should also be noted that the lubrication level affects the corrosion rate so using an acid approach for corroding the bearings might affect the grease in a different way than natural degradation this is a variable that should be taken into consideration in future research.

6. Conclusion

In this thesis, an approach has been developed and verified to identify corrosion levels in bearings with minimal domain knowledge. The method utilizes a deep-learning approach through a recurrent neural network, specifically LSTM on time-series of vibration data. The method was verified by collecting vibration data through an accelerometer attached to an anemometer. The bearings were purposely corroded to 9 different levels and the bearings run under no radial load and minimal axial load. Data collection was done at two different speeds, 5 m/s and 10 m/s, to increase the models' ability to identify intermediate wind speeds. The pre-processing was kept to a minimum by only normalizing the data and transforming the labels to binary which is required by the neural network. The model was built in Python using the Keras library. The model can classify the classes with an accuracy of 94%.

7. Summary

The goal of this thesis was to show a proof of concept of classifying corrosion levels of bearings that are not affected by radial load without statistical pre-processing of vibration signals using Artificial Neural Networks and to predict the RUL of the bearings based on these corrosion levels. This thesis has been developed within the project of European research project Arrowhead to showcase the usability of Arrowhead tools which provides a platform and architecture to connect and manage multiple types of sensors.

It was performed by building a test bed consisting of an anemometer and an accelerometer which measures the vibrations of the bearings during use. These time series are then standardized and fed into an LSTM network that has through supervised learning been trained to identify 9 levels of corrosion at two different speeds. The model manages to predict these classes with an accuracy of 94% on previously unseen test data. Using these corrosion levels a model for predicting an RUL value has been developed based on the corrosion depth at each level and the corrosion rate as well as at what corrosion depth the bearings are considered faulty.

References

[1] Jian Duan et al. "A Novel Bearing Health Prognostic Method Based on Time-frequency Analysis and LSTM". In: *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*. 2019, pp. 1–6. DOI: 10.1109/PHM-Qingdao46334.2019.8942821.

[2] Riadh Euldji, Mouloud Boumahdi, and Mourad Bachene. "Decision-making based on decision tree for ball bearing monitoring". In: *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*. 2021, pp. 171–175. DOI: 10.1109/IHSH51661.2021.9378734.

[3] *Fusion Forge Arrowhead Framework Wiki*. https://forge.soa4d.org/plugins/mediawiki/wiki/arrowhead-f/index.php/Main_Page. Accessed: 2022-06-14.

[4] P. Poór, J. Basl, and D. Zenisek. "Predictive Maintenance 4.0 as next evolution step in industrial maintenance development". In: *2019 International Research Conference on Smart Computing and Systems Engineering (SCSE)*. 2019, pp. 245–253. DOI: 10.23919/SCSE.2019.8842659.

[5] Dietmar P. F. Möller, Hamid Vakilzadian, and Roland E. Haas. "From Industry 4.0 towards Industry 5.0". In: *2022 IEEE International Conference on Electro Information Technology (eIT)*. 2022, pp. 61–68. DOI: 10.1109/eIT53891.2022.9813831.

[6] Pratik Phalle and Sangram Patil. "Fault Diagnosis of Rolling Element Bearing Using Artificial Neural Networks". In: *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*. 2021, pp. 1–4. DOI: 10.1109/ICNTE51185.2021.9487751.

[7] *Techopedia What is the difference between artificial intelligence and neural networks?* https://www.techopedia.com/2/27888/programming/what-is-the-difference-between-artificial-intelligence-and-neural-networks. Accessed: 2022-06-19.

[8] Abdulrahman Yarali. "Applications of Artificial Intelligence, ML, and DL". In: *Intelligent Connectivity: AI, IoT, and 5G*. 2022, pp. 279–297. DOI: 10.1002/9781119685265.ch16.

[9] K. Mohana Sundaram et al. "Deep Learning for Fault Diagnostics in Bearings, Insulators, PV Panels, Power Lines, and Electric Vehicle Applications—The State-of-the-Art Approaches". In: *IEEE Access* 9 (2021), pp. 41246–41260. DOI: 10.1109/ACCESS.2021.3064360.

[10] Luis P. Silvestrin, Mark Hoogendoorn, and Ger Koole. "A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance". In: *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2019, pp. 760–767. DOI: 10.1109/SSCI44817.2019.9003044.

[11] H. Wu, A. Huang, and J.W. Sutherland. "Layer-wise relevance propagation for interpreting LSTM-RNN decisions in predictive maintenance". In: *The International Journal of Advanced Manufacturing Technology*. 118. 2022, pp. 963–978. DOI: 10.1007/s00170-021-07911-9.

[12] *Analytics Vidhya Essentials of Deep Learning : Introduction to Long Short Term Memory*. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/. Accessed: 2022-06-19.

[13] *Medium LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras*. https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2. Accessed: 2022-06-19.

[14] Cheng Zhang et al. "A Method of Fault Diagnosis for Rotary Equipment Based on Deep Learning". In: *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. 2018, pp. 958–962. DOI: 10.1109/PHM-Chongqing.2018.00171.

[15] Chiao Wei Yeh and Rongshun Chen. "Using Convolutional Neural Network for Vibration Fault Diagnosis Monitoring in Machinery". In: *2018 IEEE International Conference on Advanced Manufacturing (ICAM)*. 2018, pp. 246–249. DOI: 10.1109/AMCON.2018.8614967.

[16] Yonglai Zhang et al. "Vibration analysis approach for corrosion pitting detection based on SVDD and PCA". In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2015, pp. 1534–1538. DOI: 10.1109/CYBER.2015.7288173.

[17] Ranjith-Kumar Sreenilayam-Raveendran et al. "Detection of under-lubricated ball bearings using vibration signals". In: *2013 IEEE Conference on Prognostics and Health Management (PHM)*. 2013, pp. 1–4. DOI: 10.1109/ICPHM.2013.6621431.

[18] Pornchai Nivesrangsan and Dutsadee Jantarajirojkul. “Bearing fault monitoring by comparison with main bearing frequency components using vibration signal”. In: *2018 5th International Conference on Business and Industrial Research (ICBIR)*. 2018, pp. 292–296. DOI: 10.1109/ICBIR.2018.8391209.

[19] Mahi Ayman et al. “Fault Detection in Wind Turbines using Deep Learning”. In: *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. 2022, pp. 272–278. DOI: 10.1109/MIUCC55081.2022.9781749.

[20] Qing Ni, Jinchen Ji, and Ke Feng. “Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network”. In: *IEEE Transactions on Industrial Informatics* (2022), pp. 1–1. DOI: 10.1109/TII.2022.3169465.

[21] Erkki Jantunen et al. “Predicting the remaining useful life of rolling element bearings”. In: *2018 IEEE International Conference on Industrial Technology (ICIT)*. 2018, pp. 2035–2040. DOI: 10.1109/ICIT.2018.8352501.

[22] Chen-tong Shao. “LSTM Network with PCA for Prediction of Bearing Performance Degradation”. In: *2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*. 2021, pp. 1–6. DOI: 10.1109/SDPC52933.2021.9563586.

[23] I. Y. Onel and M. E. H. Benbouzid. “Induction Motors Bearing Failures Detection and Diagnosis: Park and Concordia Transform Approaches Comparative Study”. In: *2007 IEEE International Electric Machines Drives Conference*. Vol. 2. 2007, pp. 1073–1078. DOI: 10.1109/IEMDC.2007.382825.

[24] Michael Angelo R. Alicando, Gabriel M. Ramos, and Conrado F. Ostia. “Bearing Fault Detection of a Single-phase Induction Motor Using Acoustic and Vibration Analysis Through Hilbert-Huang Transform”. In: *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*. 2021, pp. 1–6. DOI: 10.1109/HNICEM54116.2021.9732034.

[25] J.C. Banks, K.M. Reichard, and M.S. Brought. “Lubrication level diagnostics using vibration analysis”. In: *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*. Vol. 6. 2004, 3528–3534 Vol.6. DOI: 10.1109/AERO.2004.1368169.

[26] H. M. Hashemian. “State-of-the-Art Predictive Maintenance Techniques”. In: *IEEE Transactions on Instrumentation and Measurement* 60.1 (2011), pp. 226–236. DOI: 10.1109/TIM.2010.2047662.

[27] X. Chen et al. “Application of data-driven models to predictive maintenance: Bearing wear prediction at TATA steel”. In: *Expert Systems with Applications* 186 (2021), p. 115699. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.115699. URL: https://www.sciencedirect.com/science/article/pii/S0957417421010836.

[28] Leilei Ma et al. “Fault Prediction of Rolling Element Bearings Using the Optimized MCKDndash;LSTM Model”. In: *Machines* 10.5 (2022). ISSN: 2075-1702. DOI: 10.3390/machines10050342. URL: https://www.mdpi.com/2075-1702/10/5/342.

[29] F. M. Bono et al. “A deep learning approach for fault detection and RUL estimation in bearings”. In: *NDE 4.0, Predictive Maintenance, and Communication and Energy Systems in a Globally Networked World*. Ed. by Norbert G. Meyendorf, Saman Farhangdoust, and Christopher Niezrecki. Vol. 12049. International Society for Optics and Photonics. SPIE, 2022, pp. 71–83. DOI: 10.1117/12.2607084. URL: https://doi.org/10.1117/12.2607084.

[30] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2016. DOI: 10.48550/ARXIV.1609.04747. URL: https://arxiv.org/abs/1609.04747.

[31] *FusionForge Arrowhead - Technical architecture*. https://forge.soa4d.org/plugins/mediawiki/wiki/arrowhead-f/index.php/Technical_architecture. Accessed: 2022-08-08.

[32] *Davis Instruments support (SPEC SHEET) Anemometer for Vantage Pro2 Specifications (6410)*. https://support.davisinstruments.com/article/ffu8exjr1y-spec-sheet-solar-power-kit-heavy-duty-solar-power-kit-specifications-6612-6614. Accessed: 2022-06-15.

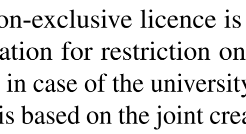
[33] *Caltrans, California Department of Transportation Division of Engineering Services Materials Engineering and Testing Services Corrosion Branch Corrosion guidelines Version 3.2, May 2021*. https://dot.ca.gov/-/media/dot-media/programs/engineering/documents/mets/corrosion-guidelines-a11y.pdf. Accessed: 2022-06-15.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Max Filip Wakéus

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Condition monitoring and Predictive Maintenance of Ball Bearings”, supervised by Gerry Nigro and Eduard Petlenkov
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

09.08.2022



[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student’s application for restriction on access to the graduation thesis that has been signed by the school’s dean, except in case of the university’s right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.