Technische Universität München
Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. Wolfgang Kellerer

Master‘s Thesis

Machine Learning based Fault Detection Algorithms for Long Haul Elastic Optical Networks

Author:	Dick, Isabella Franziska
Address:	Theresienstr.53 80333 München Germany
Matriculation Number:	03659115
University Supervisor:	PD Dr.-Ing. Carmen Mas Machuca
External Supervisors:	M.Sc. Sai Kireet Patri M.Sc. Jasper Mueller
Begin:	15th of November 2020
End:	15th of May 2021

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

München, September 20, 2021

Place, Date

Signature

This work is licensed under the Creative Commons Attribution 3.0 Germany License. To view a copy of the license, visit http://creativecommons.org/licenses/by/3.0/de

Or

Send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

The findings and results are a propriety of ADVA Optical Networking SE until published in a scientific conference or journal. This work is partially funded by the German Federal Ministry for Education and Research as a part of project titled **OptiCON** (Grant **16KIS0989K**).

München, September 20, 2021

Place, Date

Signature

Kurzfassung

Die exponentiell wachsende Zahl an Datenübertragungen und der permanente Anspruch, die Margen der Betreiber zu minimieren, führten zu dem Konzept des Optical Spectrum as a Service (OSaaS) in optischen Netzwerken. Diese Methode ermöglicht eine optimale Nutzung und Aufteilung der optischen Netzwerkinfrastrukturen, indem eine flexible Ressourcennutzung ermöglicht wird. Da der optische Transceiver im Besitz und unter der Kontrolle des Service-Endbenutzers ist, analysiert und entwickelt diese Arbeit auf OSaaS Benutzer zugeschnittene Früherkennungs- und Identifikationsmethoden für Signalausfälle. Sie evaluiert in einer vergleichenden Studie mehrere Machine Learning (ML)-Algorithmen zur Elastic Optical Network (EON)-Fehlererkennung und -identifikation. Die Eingangsdaten, auf denen die Modelle aufbauen, bestehen nur aus Optical Performance Monitoring (OPM) Daten, die an den Transceiver-Modulen des Endanwenders vorliegen. Die zugrundeliegenden Daten basieren auf einem 1792 km langen europäischen Langstreckennetz, von dem für 45 Tage alle 30 Sekunden Optical Received Power (ORP) und Pre-Forward Error Correction Bit Error Rate (pre-FEC BER) gemessen wurden.

Für die Fehlererkennung wird zunächst der ORP vorhergesagt und mit dem gemessenen Wert verglichen. Die Vorhersage wird einerseits mit dem Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX)-Algorithmus und andererseits mit drei verschiedenen neuronalen Netzen, einem vollständig verbundenen, einem Long Short Term Memory (LSTM)- und einem Gated Recurrent Unit (GRU) Netzwerk durchgeführt. Liegt der ORP-Vorhersagefehler über einem dynamisch berechneten Schwellenwert, welcher nur außergewöhnlich hohe Fehler erkennt, wird dies als Fehler definiert. Als zweite Methode zur Fehlererkennung wird der One-Class Support Vector Machine (OCSVM) Algorithmus implementiert.

Mit OCSVM können Fehler zwar früher erkannt werden, aber der Ansatz mit neuronalen Netzen erreicht eine höhere Genauigkeit. Bei der Evaluation wurden sechs von sechs Fehlern erkannt. Darüberhinaus wurden zwei aus 13000 Datenpunkten fälschlicherweise als Fehler erkannt. Außerdem werden Fehler innerhalb von 47.54% der gesamten Fehlerdauer erkannt.

Für die Fehleridentifizierung werden der Naive Bayes-Klassifikator und der Dynamic Time Warping (DTW) k-means Algorithmus so angepasst, dass sie als Open-Set-Klassifikationsmethode fungieren. Die Anpassung ist notwendig, da für eine zufriedenstellende Lösung sowohl bekannte als auch unbekannte Fehler identifiziert werden müssen. Die Ergebnisse zeigen, dass DTW k-means eine deutlich höhere Genauigkeit hat als der Naive Bayes-Klassifikator. True Positive Rate (TPR) und False Positive Rate (FPR) liegen bei 0,76 bzw. 0,06.

Abstract

Exponentially growing traffic demand and the permanent claim of minimizing operator margins have resulted in Optical Spectrum as a Service (OSaaS), which enables leveraging the full value of optical network infrastructure by flexible resource utilization. Anticipating the high demand in OSaaS, where optical transceivers are owned and controlled by the service end user, this thesis deals with the question of how OSaaS users can detect and identify failures at an early stage. It therefore evaluates several Machine Learning (ML) algorithms for Elastic Optical Network (EON) failure detection and identification in a comparative study. Hence, the input data, on which the models are based on, only consist of Optical Performance Monitoring (OPM) data available at the end user transceiver modules. For this purpose, Optical Received Power (ORP) and Pre-Forward Error Correction Bit Error Rate (pre-FEC BER) are monitored every 30 seconds for 45 days from a 1792 km point-to-point link, part of a long-haul European live network.

For the failure detection, ORP is predicted and compared to the monitored value. The prediction is performed with the Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) algorithm, as well as with a fully connected, a Long Short Term Memory (LSTM) and a Gated Recurrent Unit (GRU) neural network. If the ORP prediction error lies above a dynamically calculated threshold, which only detects abnormally high errors, it is defined as a failure. As a second method for failure detection, the semi-supervised model One-Class Support Vector Machine (OCSVM) is implemented.

Failures can be detected earlier with OCSVM, but the Artificial Neural Network (ANN) approach achieves scores of higher accuracy. The ANN approach detects all six failures as such, and two out of 13000 faultless data-points were falsely classified as failures. Furthermore, failures are detected within 47.54% of the total failure duration.

For the failure identification, the Naive Bayes classifier as well as Dynamic Time Warping (DTW) k-means algorithm are adapted such that they operate as open-set classification algorithms. This is, because both known and unknown failures need to be identified for a satisfying solution. Results show, that DTW k-means has a significantly higher accuracy than the Naive Bayes classifier. True Positive Rate (TPR) and False Positive Rate (FPR) are 0.76 and 0.06, respectively.

Contents

1 Introduction 7

2 Background 10
2.1 Machine Learning (ML) . . . 10
2.1.1 Definition, Problem Description and Goal . . . 11
2.2 Elastic Optical Network (EON) . . . 12
2.3 Test Setup . . . 14
2.3.1 Network . . . 14
2.3.2 Monitored Parameters . . . 15
2.3.3 Considered Failures . . . 17
2.4 Related Work . . . 19
2.5 Methodology . . . 22

3 Selected Machine Learning (ML) Algorithms 25
3.1 Failure Detection . . . 25
3.1.1 One-Class Support Vector Machine (OCSVM) . . . 25
3.1.2 Detection with Prediction . . . 27
3.2 Failure Identification . . . 32
3.2.1 Naive Bayes . . . 32
3.2.2 Dynamic Time Warping (DTW) K-means . . . 33

4 Implementation 36
4.1 Data Analysis . . . 37
4.1.1 Correlation of the Monitored Parameters . . . 38
4.1.2 Seasonality . . . 42
4.1.3 Correlation to Ambient Temperature . . . 44
4.2 Data Preprocessing . . . 45
4.2.1 Interpolation for Equalized Time Intervals . . . 45
4.2.2 Feature Scaling . . . 47
4.2.3 Feature Selection . . . 47
4.2.4 Data Augmentation . . . 48
4.3 Failure Detection . . . 48

4.3.1 Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) . . . 48
4.3.2 Artificial Neural Network (ANN) . . . 50
4.3.3 One-Class Support Vector Machine (OCSVM) . . . 52
4.4 Failure Identification . . . 53
4.4.1 Naive Bayes Classifier . . . 54
4.4.2 Dynamic Time Warping (DTW) K-means . . . 56

5 Results 59
5.1 Prediction . . . 59
5.1.1 Accuracy . . . 59
5.1.2 Computation time . . . 61
5.2 Failure Detection . . . 62
5.2.1 Accuracy . . . 62
5.2.2 Time of Detection . . . 64
5.3 Failure Identification . . . 64
5.3.1 Accuracy . . . 64
5.3.2 Computation Time . . . 66

6 Conclusions and Outlook 67

Bibliography 73

Chapter 1

Introduction

Due to trends like autonomous driving, video streaming and online trading, a huge growth in data traffic has emerged in recent years. The many different requirements, which result from various data traffics, has lead to a change from fixed to flexible wavelength grids in optical networks. Elastic Optical Networks (EONs) enable a flexible allocation of resources in both time and frequency. Consequently, the optical spectrum can be efficiently adjusted to each demand, based on bit rate, transmission distance and spectral efficiency. [SR20]
Despite the many advantages of EONs, unintentional failures can happen occasionally. These failures can be divided into soft and hard failures. Hard failures have an immediate effect on the optical network and thus lead to an instantaneous signal disruption. In contrast, soft failures affect the signal quality through slowly varying phenomena that manifest themselves in anomalies in the Optical Performance Monitoring (OPM) data. [SMCT18]
EONs are more prone to such soft failures than fixed grid Optical Networks (ONs) because of the increased complexity of the individual network components as well as due to various link impairments [LFL$^+$20]. End-to-end connectivity can be impacted by failures caused by terminal or line equipment (ROADMs, amplifiers, etc.), but also by infrastructure components (fibres, joint closures, cable holders, etc). These failures can gradually raise the Bit Error Rate (BER), leading to potential service downtime. Hence, detecting, identifying and counteracting these soft failures can significantly reduce system disruptions and repair costs. [VRF$^+$17, SRCV19, SYW$^+$20, SMCT18]
Due to the high demand of leveraging the full potential of the infrastructure of EONs by flexible resource utilization, the concept of Optical Spectrum as a Service (OSaaS) has evolved. In OSaaS, optical transceivers are owned and controlled by the service end user, while the Open Line System (OLS), which is providing signal equalization, transportation and amplification, is controlled by the Communication Service Provider (CSP). With this method, more revenue can be achieved from already deployed fiber networks. Not only flexible resource utilization is enabled inside the dedicated customer spectrum of the OSaaS, but also unnecessary O/E/O conversions can be eliminated. [GFAS14]
As an example, let us consider the optical transmission procedure for international con-

nections. Without OSaaS, signal regeneration is required at each country border, but with OSaaS this can be avoided as optical transceivers are owned and controlled by the service end user. Furthermore, robust scalability in the future is ensured with OSaaS.
Anticipating a high demand in OSaaS, this thesis deals with the question of how OSaaS users can detect and identify failures in an early stage. This should therefore be achieved only with information that is available at the end user transceiver modules. The approach is completely independent of OLS elements telemetry data, as OLS related parameters and configurations are not known to the OSaaS service user and will thus not be considered in the scope of this work.

The proposed solution and the algorithms selected for the evaluation on real field data are presented in Figure 1.1. As shown, the work can be divided into two tasks: failure detection and identification.
As failures occur very rarely, the first part can be seen as an anomaly detection. A common method in anomaly detection is to predict the expected value of the parameter of interest and compare it to the monitored value. If the difference between prediction and monitored value is higher than a certain threshold, it is defined as a failure.
Since Machine Learning (ML) models can typically adapt to patterns without having detailed information about the triggering causes, they are perfectly suited for the underlying prediction. To make a ML model work, most often one needs to train it with many data points [KJ13]. In our work, the input data points are considered to be the OPM data, obtained from monitoring a 1792 km long-haul European live network for 45 days. The vast amount of data is another reason to apply ML models to this project, as they handle many data points efficiently and pick up on most important trends.
To compare the performance of ML algorithms to a statistical approach, the Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) algorithm is implemented and evaluated for Optical Received Power (ORP) prediction.
Furthermore, the One-Class Support Vector Machine (OCSVM) algorithm is implemented as it has been proven high accuracy scores for anomaly detection [Bra19]. With this ML model, the properties of normal data without failures are inferred and therefore samples that differ from the normal data can be predicted as outliers.
The second part of this thesis deals with failure identification, which needs to be seen as an open-set classification. A closed-set classification considers only limited amount of classes, where at least one sample for each class is known. As this work wants to identify both known and unknown failures, two common ML classifiers are adapted such that they operate as open-set classifiers.

This thesis is structured as follows:

- **Background:** In this chapter, all fundamental background knowledge necessary to understand the thesis is introduced. This includes information about EON, ML, the underlying test setup as well as related and corresponding work.
- **Selected Machine Learning Algorithms:** This chapter describes how the algo-

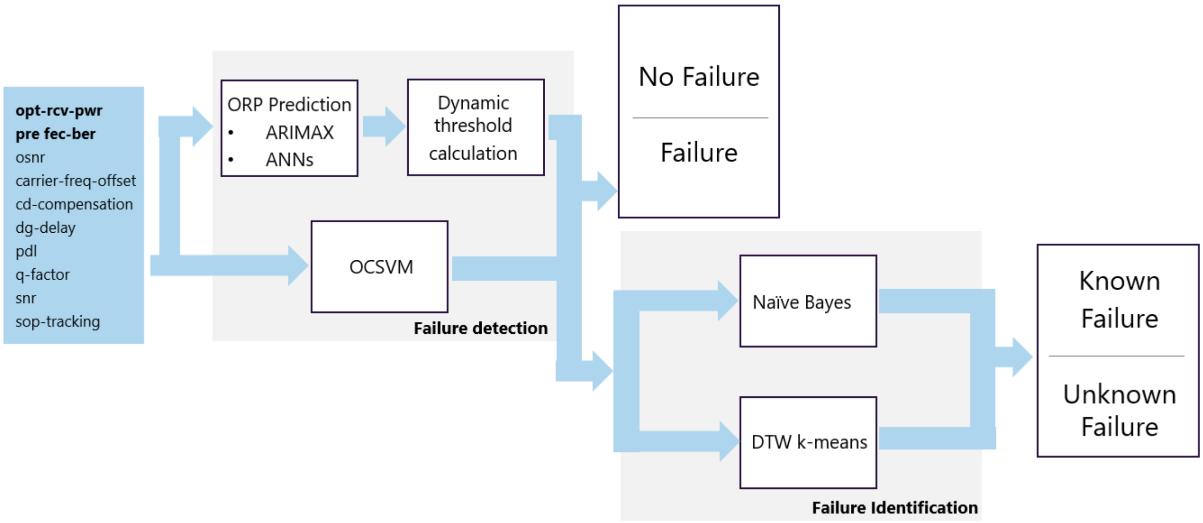


Figure 1.1: Proposed solution

rithms under study work and why they were implemented. The algorithms examined are divided into anomaly detection and failure identification.

- **Implementation:** While first explaining the required data analysis and pre-processing, the way of implementation is described by listing all hyperparameters of the utilized algorithms.
- **Results:** Subsequently, the results chapter provides the performances of all algorithms implemented based on various evaluation metrics and compares them to each other.
- **Conclusions:** Finally, the last chapter summarizes all findings from the evaluation, suggests an optimal pipeline and refers to future works.

Chapter 2

Background

This chapter gives an overview of all relevant theoretical foundations. Within the scope of this work, the following questions are being answered:

- What is ML and which types of learning are known?
- What are EONs, how did they emerge and what are the advantages to fixed grid ONs?
- What are the properties of EON used for data collection, which parameters are monitored and which failures are considered for evaluation?
- Which approaches exist concerning failure detection and identification in EONs?
- What is the goal of this work and how is it achieved?

To answer these questions, a definition of ML is given, followed by a description of EONs. Then, the test setup of the underlying network is described. Subsequently, a state of the art discussion in the field of failure detection and identification in EONs is carried out, especially focusing on ML approaches. The section is closed by stating the goal of the thesis.

2.1 Machine Learning (ML)

This section gives an overview of the concept, a basic definition as well as the main types of ML.

2.1.1 Definition, Problem Description and Goal

ML can be understood as artificial generation of knowledge from experience. The term was defined by Tom Mitchell in 1997 [Mit97] as follows:

> A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

ML consists of designing efficient and accurate prediction algorithms. These algorithms take a set of examples as a training input, build a statistical model from it, and generalize it. In this way, the system is able to asses unknown data and to make predictions or decisions without being directly programmed to do so. Besides time and space complexity of the calculation, the input sample size are critical measures of the algorithm quality. Since the success of a learning algorithm is completely dependent on the underlying training data, ML can be seen as data-driven concepts combining computer science with methods from statistics, probability and optimization. [MRT18, Gé19]

Machine learning approaches relevant for the underlying work can be divided into three broad categories, depending on the data available to the learning model:

- **Supervised Learning** includes every task in which the algorithm has access to labeled data. The goal here is to learn the correlation of the input to the output data.
- **Unsupervised Learning** covers all tasks that have no access to output values. Therefore the objective is to analyze similarities among the input data.
- **Semi-supervised Learning** defines tasks where both labeled and unlabeled data are present. A common use case would be a setting where only labels of few out of many classes are available.

This work compares supervised with semi-supervised learning approaches for the failure detection and supervised with unsupervised learning algorithms for the identification. ML includes many algorithms with different objectives and it can therefore further be structured into the task it pursues. The following describes some of the most commonly used learning tasks of ML, which are implemented later:

- **Classification** assigns a category to each input item. It needs labeled training data for learning and therefore usually belongs to supervised learning, but might also, if not all data points are labeled, belong to semi-supervised learning. A typical example would be the mapping of handwritten digits to classes from 0 to 9.
  The classification algorithm Naive Bayes is later used to identify the type of failure after it has been detected. Further, OCSVM is used to detect failures lying outside the defined non-faulty class.
- **Regression** predicts an output value for each set of input values and therefore is a

supervised method. Common examples of regression are stock market price predictions based on the historical stock market values as well as various other factors. Regression is applied to the OPM data in order to predict the ORP and then detect failures with the deviation to the monitored ORP.

- **Clustering** isolates subsets in a given set of input data points. The goal is to divide the data based on the characteristics of present patterns or structures. As labeled data is not needed, it belongs to the group of unsupervised learning methods.
  This work uses the clustering k-means algorithm for failure identification.

[KJ13, Gé19]

2.2 Elastic Optical Network (EON)

ONs are a communication medium that uses signals encoded in light to transmit information. The method of optical networks can be applied in various types of telecommunications networks, e.g. Local-Area Networks (LAN) or Wide-Area Networks (WAN). The majority of all human and machine-to-machine information as well as the Internet are based on ONs as they achieve extremely high bandwidth and can therefore transmit large amounts of data at once. The main enabling technologies are:

- Wavelength Division Multiplexing (WDM), which creates multiple channels within a single fibre with the help of a prism, which splits the light into different wavelengths. An optical channel is defined by an optical signal lying on a particular wavelength and carrying the transmission data by being modulated accordingly. Therefore, the beams of different wavelengths can be transmitted through the fiber simultaneously, being distinguishable from each other, because their frequency peaks are far enough apart from the peak of the neighboring beams.
- Reconfigurable Optical Add-Drop Multiplexers (ROADMs), which are the devices used for multiplexing and routing in WDM systems. A ROADM enables remote switching of traffic on the wavelength layer. In contrast to conventional multiplexers, ROADMs are flexible in rerouting optical streams and can bypass faulty connections without affecting traffic already passing the ROADM. This can be done remotely. The Wavelength Selective Switch (WSS) module of ROADMs allows one or more wavelengths to be added or dropped from a transport fiber. This works without converting the signals on all of the WDM channels to electronic signals and back again to optical signals. Therefore, if enough power and space is present, ROADMs are used as the preferred add-drop multiplexer technology. [Cla05]
- An optical amplifier, which amplifies the optical signal without conversion. It is compensating the loss of all optical elements along the signal path including the transmission fiber. An amplifier can be located at the input and output end of

> the system (boosters and pre-amplifier, respectively), but also between transmission fiber spans (in-line amplifier). Especially for long-haul transmission systems, in-line amplifiers are indispensable. The most common amplifiers are Erbium Doped Fiber Amplifiers (EDFAs) and Raman amplifiers. The EDFA works as a pump laser that excites the rare earth erbium atoms distributed in the fiber, which emit light, and thus boosts the optical signal. Raman uses the nonlinear interaction between the signal and a pump laser in the optical fiber to achieve amplification. [MRJP87]

In order to transmit data with as little energy as possible, the wavelength ranges are selected in which optical fibers have the lowest transmission losses. This low-loss wavelength region ranges from 1260nm to 1625nm and is separated into five wavelength bands called O-, E-, S-, C- and L-bands. Today, the C-band (conventional band: 1530-1565nm) is commonly used in many metro, long-haul, ultra-long-haul, and submarine optical transmission systems as it shows lowest loss of all bands. Therefore, it is also used in the underlying long-haul live production network to generate training data.

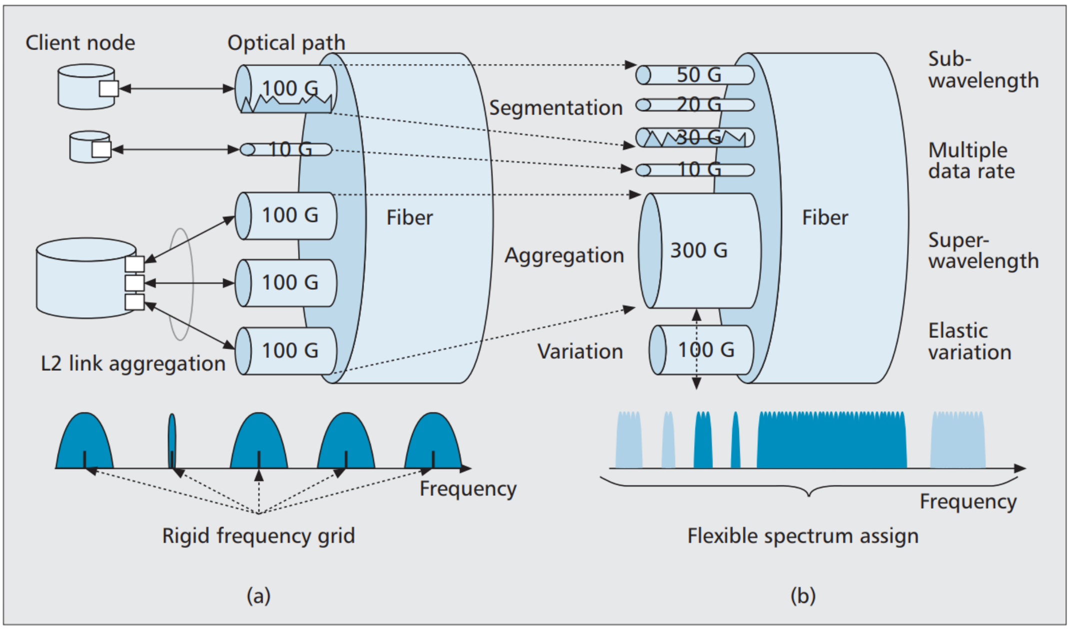


Figure 2.1: (a) Gridless architecture with arbitrary spectral widths and spacing. (b) Fixed grid spectrum assignment. [JTK+09]

Until 2009, fixed frequency grid spaces of 12.5 GHz, 25 GHz, 50 GHz or even 100 GHz were specified by the International Telecommunication Union (ITU). To meet increased data rate demands and, at the same time, stay within the fixed channel space, modulation formats with a higher number of bits per symbol were used. However, the rigid bandwidth and coarse granularity has led to inefficient and inflexible spectrum allocation. Furthermore, long distance transmissions were limited due to more interferences caused by the increased number of bits per symbol.

To overcome these issues, spectrum-sliced elastic (SLICE) optical path architecture was

introduced in 2009 [JTK$^+$09]. Using the SLICE approach, the network client services are allocated to the exact amount of optical spectrum that they require, as shown in Figure 2.1. The main advantages of the flexible over the fixed grid are that it has a denser granularity and that it can be scaled dynamically.
Figure 2.1 makes it clear, that EONs support sub- and super-wavelength as well as varying data rates. Sub-wavelength transmission means that the full bandwidth provisioning between the source and destination nodes is no longer needed. With super-wavelength, multiple continuous wavelengths can be combined into a super-wavelength and thus the interval bands can be saved. EON has a flexible grid so that appropriate spectrum can be assigned for different data rates and therefore it has higher resource utilization than fixed grids. [SR20]
Since this work investigates the ability of fault detection and identification with ML, an EON with the specifications described in the next section is used to generate OPM data.

2.3 Test Setup

This section provides an overview of the topology of the network from which the data is generated. Moreover, it also explains what the observed parameters track and describes the failures generated for evaluation.

2.3.1 Network

To collect data for training and evaluation of the ML models, a commercial pan-European live network operated by Tele2 Estonia spanning 2869 km [KRB$^+$20] is used, in which the majority of the C-band is filled with high-margin 100G QPSK channels. Besides the live channels carrying production traffic, five ADVA TeraFlex [Ter21] transceivers were installed to generate data for testing. Figure 2.2 shows that multiple optical lightpaths with lengths up to 5738 km are available, enabled by optical loopbacks in intermediate ROADMs nodes. To create training data for the underlying study, the second loopback location is used to create a link length of 1792 km using configurations of 200G QPSK and 16QAM. The five test channels are inserted into a custom add/drop port of the final ROADM using an 8-port splitter/combiner module. It is configured as OSaaS and occupies 400 GHz spectrum with central frequency of 193.95 THz within the C-band. Both Raman and EDFAs are used for amplification. To collect data for training as well as for testing and evaluating the implemented ML models, the parameters described in the following Section 2.3.2 are recorded for all five channels at the transceiver.

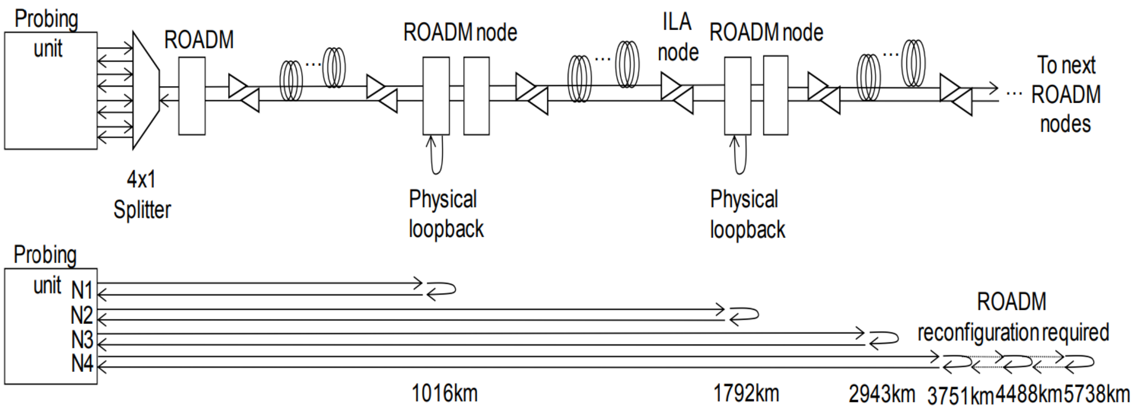


Figure 2.2: Test route [KRB+20]

2.3.2 Monitored Parameters

Modern optical transceivers support optical monitoring functions, which enable real-time monitoring of performance parameters.
OPM deals with obtaining performance information about an optical communication signal, it involves assessing the quality of data channel by measuring its optical characteristics without directly looking at the transmitted sequence of bits. Therefore, it is a potential instrument to improve the quality of transmission as well as the physical layer fault management in optical transmission systems. The following section describes all OPM parameters, which are monitored approximately every 30 seconds for all five channels of the network described in Section 2.3.1.

Carrier Frequency Offset (CFO)

Carrier Frequency Offset (CFO) denotes the mismatch between the frequency of the received signal and of the local oscillator at the receiver. It occurs when the two signals do not synchronize. Most often this arises due to two phenomena: first, a frequency mismatch in the transmitter and the receiver oscillators due to a difference between the manufacturer's nominal specification and the real frequency of that device. Second, the Doppler effect as the transmitter or the receiver is moving.

Chromatic Dispersion Compensation (CDC)

Chromatic Dispersion (CD) defines the frequency dependence of the phase velocity in a transparent medium. It occurs because of an inherent property of the silica fiber. With Chromatic Dispersion Compensation (CDC), this phenomenon is counteracted. The velocity of a light wave depends on the refractive index of the medium within which it is

propagating. In silica fiber, as well as many other materials, this index changes with wavelength. Hence, different wavelength channels travel at slightly different speeds within the fiber. This spreads the transmission pulse as it passes through the fiber. Furthermore, the difference between the velocity of two wavelengths depend on their locations.
All fibers have a wavelength at which CD is practically zero, called fiber zero dispersion wavelength. The further away a wavelength is from the zero dispersion position of its fiber, the larger the CD. If an optical channel has positive dispersion, its longer wavelength components travel slower than its shorter counterparts, and vice versa for an optical channel subjected to negative dispersion. A light pulse with dispersion can be corrected by traveling through an opposite dispersion medium to that of the fiber.

Differential Group Delay (DGD)

Differential Group Delay (DGD) refers to the difference in propagation time between the two eigenmodes X and Y polarizations of a signal. It occurs because the group velocities of different modes are usually different. It usually is a limiting factor for which transmission bandwidth can be achieved and thus also for the transferable data rate when multimode fibers are used.

Pre-Forward Error Correction Bit Error Rate (pre-FEC BER)

The BER is defined as the ratio of bit errors to the total number of transferred bits. The errors are can be caused by attenuation, ageing or temperature changes of the optical fiber. The Forward Error Correction (FEC) is used to correct these bit errors in the received data and if the Pre-Forward Error Correction Bit Error Rate (pre-FEC BER) stays below a certain limit, all bit errors are successfully identified and corrected and therefore, no packet loss occurs. The term pre-FEC BER, however, stands for the BER, which has not been corrected yet and therefore still contains all bit errors. This is why it is a good measure for early detection of failures.

Forward Error Correction (FEC) - Corrected Errors

The errors corrected by the FEC are counted with this metric.

Forward Error Correction (FEC) - Uncorrected Blocks

Uncorrected blocks counts the number of errors that were so corrupted by noise that they could not be corrected or recovered by the FEC algorithm since the last data request, hence within approximately 30 seconds.

Optical Received Power (ORP)

The ORP, also called Rx power, is the incoming power of the transceiver at the receiving end. This should not be confused with the transmit or Tx power, which is the power of the signal leaving the transceiver at the transmitting end. Both parameters are measured in dBm.

Optical Signal to Noise Ratio (OSNR)

The Optical Signal to Noise Ratio (OSNR) is defined as the ratio of optical signal power to the optical noise power, expressed in dB. A positive OSNR (>0 dB) indicates more signal than noise. OSNR is calculated with respect to the bandwidth, which is usually 12.5 GHz corresponding to 0.1 nm wavelength.

Polarization Dependent Loss (PDL)

Polarization Dependent Loss (PDL) measures the ratio of the maximum and the minimum transmission with respect to all possible polarization states.

Q-factor

The q-factor can be defined as a measure of the damping or energy loss of a system capable of oscillating. Therefore, higher Q-factor indicates less energy loss and the oscillations swings out more slowly. The parameter is dimensionless.

Signal to Noise Ratio (SNR)

Consequently to the OSNR, the Signal to Noise Ratio (SNR) is the measure of the ratio of electrical signal power to electrical noise power, also expressed in dB.

2.3.3 Considered Failures

End-to-end connectivity can be affected by terminal or line equipment failures (ROADMs, amplifiers, etc.), as well as by infrastructure components (fibres, joint closures, cable holders, etc) [RSAE18]. Besides the failures concerning direct equipment malfunction, failures can also occur due to misalignment, contamination, torsion or strain of the cables. There are many possible reasons for failures, but to evaluate the studied ML algorithms, the three typical failure patterns described in this section are used. That is, because on the one hand, they are common failures that were used in a similar manner for evaluation in other works

[VLR+19, VRF+17, LFL+20, SMCT18, WZW+17] and therefore results are comparable. On the other hand, the failures used are feasible to simulate within the underlying network described in Section 2.3.1 without disrupting the data transmission in the live network.

Power Degradation

Gradual power degradation can appear due to almost all causes listed above; gradual equipment failure because of e.g. aging lasers or amplified spontaneous emissions [LFL+20, SMCT18]. To depict these soft failures as real as possible, it is important to consider different gradients for the power degradation.

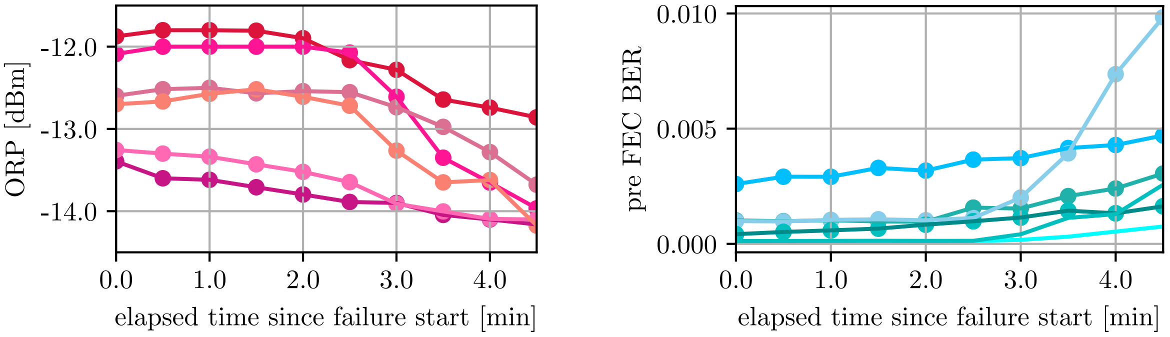


Figure 2.3: ORP and pre-FEC BER during Power Degradation

To simulate this, the transmit power was reduced by 0.2, 0.5 and 1 dBm every minute until no signal was received. Two different modulation formats were used for this purpose: 100G QPSK with 31.5 GBaud, starting at -5.5 dBm and 200G QPSK with 69.44 GBaud, starting at -2 dBm. The behavior of the ORP and the pre-FEC BER are shown in Figure 2.3. Note that for better comparability, the x-axis does not show the absolute time, but the time in elapsed minutes since the start of the respective failure. The failures did not occur simultaneously. While the ORP decreases gradually by one to two dBm, the pre-FEC BER rises by 0.001 to almost 0.01.
The Power Degradation Failure is referred to as Failure 1 in the below chapters.

Inter-channel Interference

Inter-channel interference occurs, when two neighbouring channels are too close to each other. This most often happens due to false configuration or due to an inaccurate laser of the central frequency [VLR+19, VRF+17].

This failure was introduced by shifting the frequency by 6.25 GHz of the central channel first to the left and then to the right and therefore reducing the channel spacing to the left

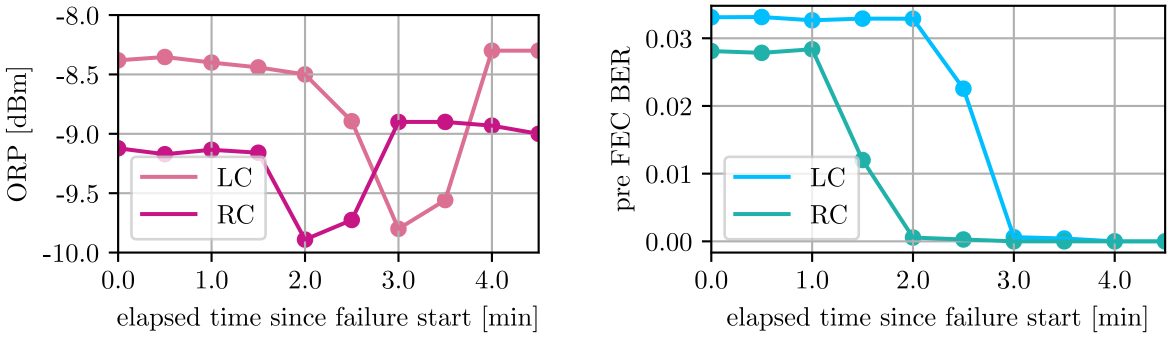


Figure 2.4: ORP and pre-FEC BER Inter-channel Interference

and the right neighbouring channel, respectively.

Figure 2.4 shows the behaviour of the ORP and the pre-FEC BER of the two neighbouring channels (Left Neighbouring Channel (LC) and Right Neighbouring Channel (RC)) during the time window which is given to the identification algorithm. It can be seen that the ORP drops about 1 dBm within 30 seconds and comes back up within one minute. The pre-FEC BER on the other hand, drops to complete zero and does not recover within the next 2.5 minutes. A complete zero pre-FEC BER implies that no more signal is received. This Failure is defined as Failure 2 within the context of this work.

Power Drop

A very steep drop in ORP can occur due to some kind of complete equipment failure, for example a fiber cut or a fault in the card [WZW⁺17]. In contrast to the previous considerations, this is a hard failure since it has an immediate impact on the network. Nevertheless, it is still considered for the evaluation, as the method's effectiveness on hard failures shall be demonstrated as well.

There is no exact documentation of this failure, as it is a real case scenario which was recorded while running the network under normal conditions. Figure 2.5 shows, that both ORP and pre-FEC BER drop immediately during the event; the ORP by six to eight dBm and pre-FEC BER to complete zero.

The Power Drop is called Failure 3 from here on.

2.4 Related Work

This section gives an overview of literature related to failure detection and identification in EONs. It mainly focuses on ML approaches in this field. Furthermore, the differences to

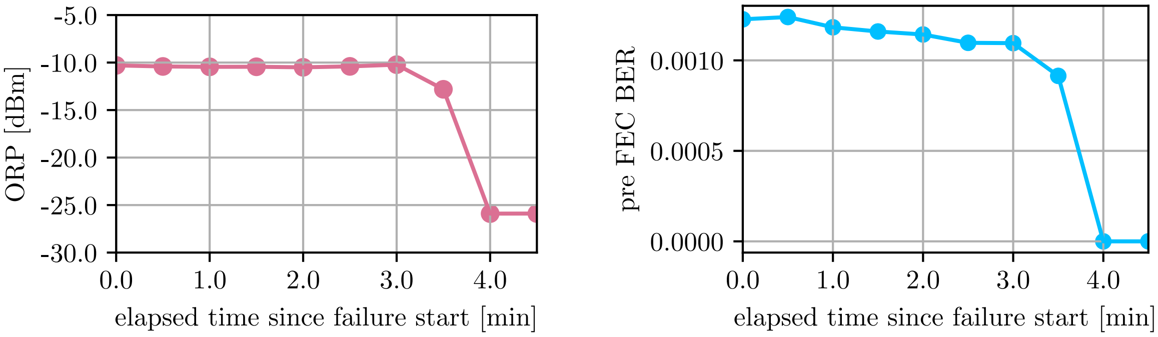


Figure 2.5: ORP and pre-FEC BER Power Drop

this thesis are pointed out, concluding with an explanation for the necessity of this works approach.

In recent years, predicting the Quality of Transmission (QoT) of unestablished lightpaths, meaning forecasting the reliability of the network before deployment, has been studied. [AT18, ATAT20] have shown that ML based methods such as k Nearest Neighbours (kNN), Support Vector Machine (SVM), Random Forest (RF) or Artificial Neural Networks (ANNs) have the potential to estimate the QoT fast and rather accurate. Similar to that work, [MP18] compares different ML models in order to predict SNR margins. [RBGT18] suggests to use a RF classifier to predict if the BER of unestablished lightpaths meet the required system threshold. [YMH+19] present a Q-factor based QoT estimation, using transfer learning ANNs, which reduces the training time enormously. In transfer learning, the knowledge learned from one ON can be transferred to a similar ON, resulting in only a few weights in the ANN having to be adjusted with a very small amount of training examples to fit the new network.
The work on this field clearly proves that ML is able to outperform other techniques for QoT estimation. Nevertheless, most of the approaches were not tested on real field data neither do they predict the quality of transmission based on the dynamically changing OPM data, as they tackle the problem of ensuring the performance of the undeployed networks. Hence, the main difference to the underlying work are the input features and the kind of data used for predicting failures in EONs.

Other research groups have studied fault detection and identification with Power Spectrum Density (PSD) as input. PSD gives the distribution of the signal power over the entire frequency range.
The authors of [LFL+20] have worked on a soft failure identification scheme based on a Convolutional Neural Network (CNN). The method is highly accurate, if only one of the four by the CNN known failures are given to identify. The identification is performing worse when given multiple failures at the same time and was not tested on unknown failures, as it is not designed for this use case.

[SRCV19] analyses three methods for detecting and identifying filter shift and tight filtering all using SVM. First, a multi-classifier SVM, which uses features extracted directly from the optical spectrum. Second, a One-Class SVM, which takes pre-processed features to compensate filter cascading. Third, a residual-based approach, which uses a residual signal computed from the difference of the actual signal and the expected signal, which is synthetically generated. The results show that the residual-based approach performs best in both accuracy and robustness.
The authors of [NUW$^+$21] have developed an anomaly detection method, where a CNN based autoencoder compresses constellation diagram images, extracting the most important feature information, and from which the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm then detects anomalies. The compression yields up to a 200-fold runtime reduction as well as improves the accuracy compared to applying the DBSCAN directly to uncompressed constellation diagrams.
The research results show, that on one hand, the PSD carries valuable information, that can be used for precise failure detection and -identification. On the other hand, it is extremely memory intensive and the computation time is therefore very high if the complexity is not reduced beforehand.

There also exists literature that studies failure detection and -identification with similar input features compared to the approach of this work.
As one of the first in this field, [VRF$^+$17] proposes an anomaly detection method where the latest BER value is compared to thresholds computed from the mean and standard deviation of the last n BER measures. The failure identification consists of a probabilistic algorithm based on ORP and BER that returns the most likely failure among a set of failure classes; signal overlap, tight filtering and drift.
[WZW$^+$17] uses SVM in combination with double exponential smoothing for equipment failure prediction. The input features are physical parameters of the equipment such as the ORP, the laser current, the environmental temperature, the module internal temperature, the central wavelength, the OSNR and the power consumption. The average prediction accuracy of the method was 95%.
[SMCT18] presents a BER anomaly detection based on monitored BER data. Therefore, different ML algorithms were implemented: ANN, SVM and RF. It was shown, that the SVM outperformed the other two methods slightly in terms of accuracy and training duration. Further, a sensitivity analysis has demonstrated how often BER values should be collected and analyzed. The shorter the time between two observations, the shorter window is sufficient to collect enough BER samples in terms of optimizing performance (100% accuracy is obtained for window duration of around 18 minutes if the distance of observations is 22s). The identification module presented is based on a binary SVM distinguishing between narrow filtering and reduced amplification. All experiments were executed on real field data.
[RSAE18] proposes a method for detecting, localizing and identifying potential faults based on Software Defined Network (SDN) integrated knowledge. The failure detection uses optical power level abnormalities only, and the localization works by network topology map-

ping. The failure identification is based on optical power level, temperature, amplifier gain as well as fan current draw and gives either a fan failure or a cooling unit failure as an output.

The work of [CLP^{+}19] consists of two stages in order to detect failures in ONs. First, DBSCAN is used to analyze patterns of the monitored OPM data. Second, the learned patterns are transferred to train a ANN for anomaly detection. The approach was tested on experimental data and shows a mean accuracy rate of 94.8% for the detection.

[SYW^{+}20] proposes a dual-stage approach for soft failure detection to reduce computational time. At the first-stage, only pre-FEC BER and ORP are taken as an input for a Gaussian distribution based anomaly detection algorithm. The second-stage of detection uses a One-Class SVM to distinguish true soft failures from false ones with additional digital spectrum features. Further, the authors assign the given failure to one of four possible errors, similar to [LFL^{+}20]. This is done with a SVM, which takes pre-FEC BER, ORP, spectral areas and asymmetries as an input. The work was trained and tested on simulated data and shows high accuracy scores (<4.44% False Positive Rate (FPR), <1.52% False Negative Rate (FNR) for detection and >84.58% acurracy for identification).

Even though some research works have appeared on the topic of soft failure detection in (elastic) ONs, questions are still open, whether all of the proposed solutions also work on real field data. Furthermore, it is not clear, which ML technique among the large set of already existing and well established tools is best suited for this purpose. Moreover, the approach of this work is carried out from a OSaaS point of view and therefore it is only based on OPM data available at the end user transceiver modules.

Although the area of fault identification has been investigated by few works, so far all classification methods use a closed set of predefined faults. Since it is not possible to cover all types of faults, which could be infinitely high, this is not satisfactory.

2.5 Methodology

This section explains the methodology of the underlying work. The problem description is divided in two parts; early soft failure detection and failure identification. For each task, different algorithms are selected and evaluated, compare Figure 2.6.

The failure detection can be treated as an anomaly detection in multivariate time series. That is, because the input consists of multiple parameters of OPM data monitored every 30 seconds. Furthermore, failures can be considered as anomalies since EONs are typically designed conservative with large operating margins. Therefore, failures occur very rarely and it is difficult to build a large enough failure data set to train a supervised ML model on data including failures [SYW^{+}20]. Hence, the failure detection part of this work considers supervised- and semi-supervised learning methods based on data without failures.

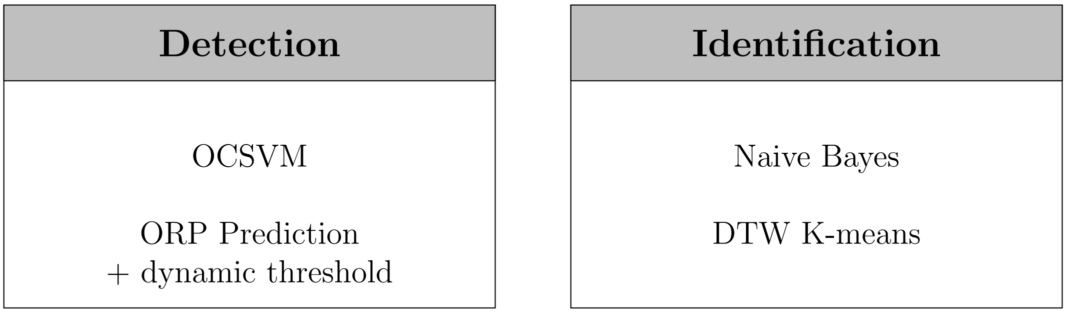


Figure 2.6: Methodology overview

The detection is implemented in two different ways, with an OCSVM and ORP prediction in combination with a dynamic threshold. OCSVM is performed because on the one hand, a comparative study [Bra19] shows, that besides neural network approaches, OCSVM has the highest performance in multivariate time series anomaly detection. On the other hand, many similar ON failure detection approaches [WZW$^+$17, SMCT18, SYW$^+$20] have proven high accuracy when applied to synthetic data. For this purpose, the faultless data points are used to define a hypersphere, which separates them from the failures.
The ANN prediction approach was also selected based on [Bra19]. In the comparative study, ANN approaches performed best for the majority of evaluation metrics. To show whether and to what extent ANN prediction is more accurate than a simple non-ML approach, the ARIMAX model is implemented as well. Further, the difference between the predicted and the monitored value is calculated, to assess whether a predicted ORP value is anomalous or not. If this difference is above a dynamic, unsupervised and non parametric computed threshold [HCL$^+$18], it is considered as an anomaly. For each of the five channels described in Section 2.3.1 a separate model is trained to predict ORP values for that channel. This is because the prediction accuracy of ANNs increases with less dimensional outputs [HCL$^+$18]. It also allows traceability down to channel level, and thus failures can later be assigned to the channel in which they occur. Moreover, having an own model for each channel ensures capturing the characteristics of that specific channel.

The underlying prediction is computed based on a time series $X = x^{(1)}, x^{(2)}, ..., x^{(n)}$, where each timestep $x^{(t)} \epsilon R^m$ is a m-dimensional vector $x_1^{(t)}, x_2^{(t)}, ..., x_m^{(t)}$, whose elements correspond to input variables, meaning the OPM parameters described in Section 2.3.2. In general, there are three variable parameters, that can be adjusted for the prediction:

- The amount and selection of the input features
- The sequence length l_s, which specifies the number of look-back timesteps to input into the model for prediction
- The prediction length l_p, that determines the number of steps ahead to predict. This is one in the underlying approach, as only one timestep ahead is predicted.

Furthermore, the number of dimensions d being predicted is $1 \leq d \leq m$, which is 1 for the underlying problem since the ORP of a single channel is predicted. Hence, a single scalar prediction $\hat{y}(t)$ is generated at each timestep t. [MVSA15]

The failure identification has to be seen as an open set classification or recognition task, because both failures that are known from training as well as unknown failures have to be identified [SRSB13].
The Naive Bayes classification is used to classify a univariate failure based on different stacking techniques, meaning transformations from multi- to univariate time series input. It is implemented because for each class and thus known failure, it gives the probability that the failure to be identified belongs to it. This then is done for multiple stacking methods. Indicators for an unknown failure are evenly distributed probabilities for all classes as well as different probabilities for different stacking methods. Therefore, a threshold is found that defines a probability beneath which all failures are classified as unknown failures.
The second algorithm implemented for the identification is k-means. K-means is a clustering algorithm, which measures the similarity between two data points or sets with a distance metric, commonly the Euclidean distance. In this case, as time series windows are compared, Dynamic Time Warping (DTW) is used as a distance metric. The smaller this distance metric, the more similar the time series are and the higher the probability that the underlying failure belongs to the class of comparison failures.
If a failure is classified as an unknown, five other versions of it are created with data augmentation, and from these samples a new class is created. Thus, in the future, a failure of the same type can be classified and assigned to the newly learned class.

The next chapter gives a deeper explanation about the computation of each algorithm.

Chapter 3

Selected Machine Learning (ML) Algorithms

This chapter describes the selected algorithms implemented for failure detection and identification. In the scope of this thesis, the following questions are being answered for each algorithm:

- How does the computation work in particular?
- Which hyperparameters can be adjusted?
- How can the algorithm be adapted to the underlying OPM data?

3.1 Failure Detection

For the failure detection, OCSVM as well as prediction of ORP with ARIMAX and with an ANN is explained. Furthermore, the dynamic threshold calculation is described.

3.1.1 One-Class Support Vector Machine (OCSVM)

SVM is a classification algorithm, which is a subset of supervised learning methods. It was originally invented to perform binary classification. The method is based on a set of labeled training objects, where each object is represented by a vector $\mathbf{x}_i$ labeled by y_i.

$$(\mathbf{x}_i, y_i) \mid i = 1, \ldots, m;\ y_i \in \{-1, 1\} \tag{3.1}$$

The SVM aims to find a hyperplane such that it divides the training objects into two classes. The distance between the hyperplane and the vectors that are closest to the hyperplane, also called support vectors, is maximized. Only these support vectors are sufficient to describe the plane mathematically accurately, as shown in Figure 3.1.

The hyperplane is described as followed:

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{3.2}$$

Where the normal vector $\mathbf{w}$ defines a straight line through the coordinate origin. Perpendicular to this vector are hyperplanes, where each intersects the vector at a certain distance $\frac{b}{\|\mathbf{w}\|_2}$ to the origin. This distance is called bias b. The combination of the normal vector and the bias uniquely determine a hyperplane, and for all points $\mathbf{x}$ belonging to it, 3.2 holds. For points, that are not on the hyperplane, the value is either positive or negative. The sign defines on which side of the hyperplane the point lies. If a class membership in the training examples is expressed by $y_i = \pm 1$, the class of a vector can be obtained by the formulatic condition (3.3):

$$y_i = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \tag{3.3}$$

For real world applications, most of the time the data can't be separated linearly. Because a hyperplane cannot be bent, the kernel trick is performed in such cases. The underlying concept behind the kernel trick is to transfer the vector space including the training vectors into a higher-dimensional space. In a space with a sufficient high number of dimensions, every vector set becomes linearly separable. Hence, a suitable kernel function is used to describe the separating boundary, which describes the hyperplane in the high dimension but still remains mathematically representable in the low dimension. [Jos20]
Hyperparameters, that can be adjusted include the kernel function (linear, polynomial with degree, sigmoid or radial basis) and the parameter ν, which defines the upper bound on the fraction of training errors as well as a lower bound of the fraction of support vectors. It can also be seen as the ratio of outlier samples in data set.
The concept of SVM can be extended to non-binary classification. At multiclass SVM, the problem is reduced to several binary classification problems. Whereas for OCSVM, the smallest possible hypersphere containing all data points of that one class is being obtained. Hence, for the multivariate anomaly detection a OCSVM is used to detect all failures lying outside of the hypersphere containing all faultless OPM datapoints, similar to [SRCV19, SMCT18, SYW$^+$20]. The time factor is not considered with this algorithm, hence only the values of the features at one timestep is given as an input to the OCSVM.

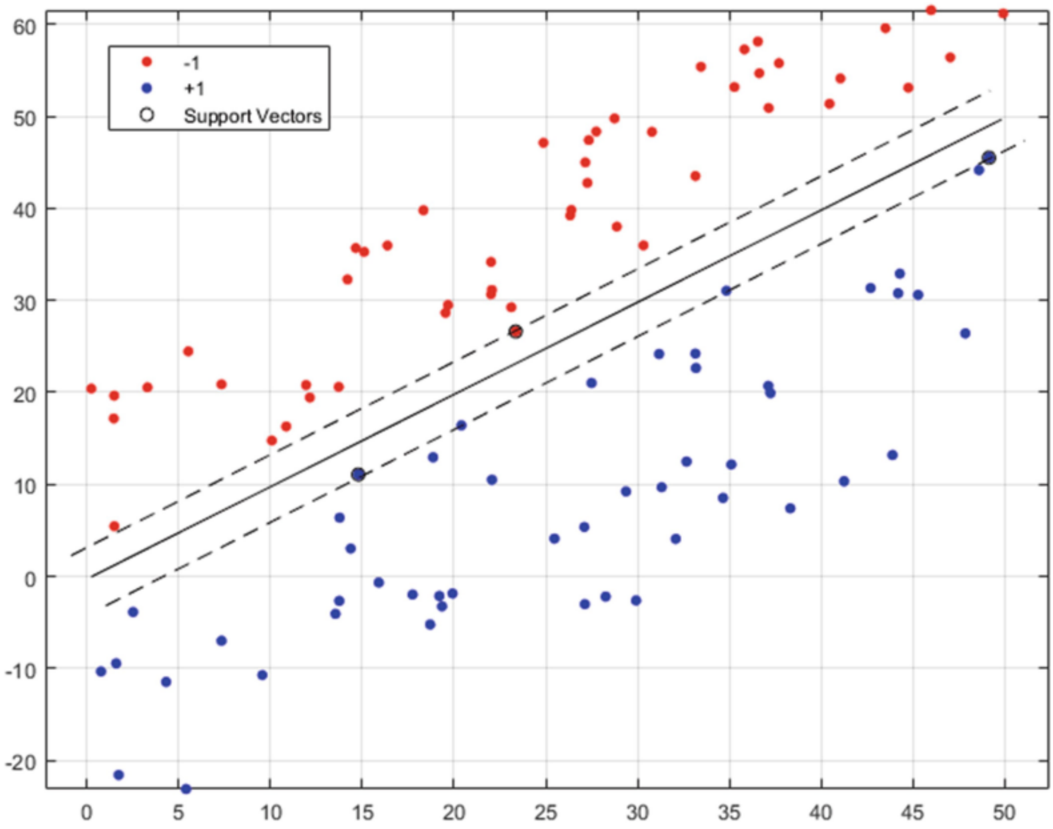


Figure 3.1: Linear binary SVM applied on separable data [Jos20]

3.1.2 Detection with Prediction

Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX)

A very common method in time series forecasting is the Autoregressive Integrated Moving Average (ARIMA) model, which aims to describe the autocorrelations in the data. It is fitted to time series data to predict future points in the series. The prediction is parameterized by three distinct integers:

- **p**: Number of autoregressive terms, incorporating the effect of past values into the model (Autoregressive term)
- **d**: Number of nonseasonal differences needed for stationarity (Integrated term)
- **q**: Number of lagged forecast errors in the prediction equation (Moving Average term)

The effect of the single parts of the ARIMA method can be explained best on an example: Considering the temperature prediction of the next days, the autoregressive term states

that if it was warm the last couple of days, it is likely going to be warm the next day. The integrated part of the model states that the temperature of tomorrow will be similar to today's, if the difference in temperature in the last days has been very small. The moving average term defines the error of the prediction as a linear combination of the error values from preceding timesteps.
Hence, if for one of the terms the effect it is representing is not given in the underlying data, the order of this terms is zero. Moreover, the higher the order of each term, the more values of the past are taken into account.

Mathematically, the model can be described as followed:

$$y^{(t)} = \alpha + \sum_{i=1}^{p} \phi_i y^{(t-1)} + \sum_{j=1}^{q} \sigma_j \epsilon^{(t-j)} + \epsilon^{(t)} \tag{3.4}$$

Where α is a constant, the ϕ terms denote linear combination lags of y (up to p lags) and the σ terms represent a linear combination of lagged forecast errors ϵ (up to q lags).
Further, y denotes the d^{th} difference of Y, meaning $y^{(t)} = Y^{(t)} - Y^{(t-1)}$ for $d = 1$, $y^{(t)} = (Y^{(t)} - Y^{(t-1)}) - (Y^{(t-1)} - Y^{(t-2)})$ for $d = 2$ and so on.

It is possible to extend the ARIMA model by b exogenous variables in order to fit multivariate data. In literature, the algorithm is often named ARIMAX. Equation 3.4 would then be extended by:

$$y^{(t)} = \alpha + \sum_{i=1}^{p} \phi_i y^{(t-i)} + \sum_{j=1}^{q} \sigma_j \epsilon^{(t-j)} + \epsilon^{(t)} + \sum_{m=1}^{b} \eta_m x^{(t-m)} \tag{3.5}$$

where $\eta_1, \ldots, \eta_b$ are the parameters of the exogenous input $x^{(t)}$. [HA18]

Since the underlying OPM data is a multivariate time series, ARIMAX is used to predict the next value of ORP. Thus, p, q and d are found for the best fit of the algorithm.

Artificial Neural Network (ANN)

Due to recent advances in ML, computational capacity and ANN architectures, performance breakthroughs in time series learning tasks have taken place. In particular, recurrent networks such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) have proven to be well suited for time series prediction [SVL14]. Therefore, this section focuses on the theory behind predictions with ANNs.
An ANN consists of a collection of connected nodes following the model of neurons in a biological brain. Each connection transmits signals, represented by real numbers, from one node to another. A node that receives a signal processes it and can further pass the modified signal to nodes connected to it. This output is computed by a non-linear function

of the sum of its inputs. Nodes and connections, also called edges, usually have weights which in- or decrease the impact of the signal. These weights adapt to the task during the training phase. This is done by first determining the prediction error, which is the difference between the processed output of the network and the target output. Then, the weights are adapting according to a learning rule defined by the optimizer using the error which ideally leads to a smaller error value in the next run. Typically, nodes are arranged in layers as shown in Figure 3.2. Each node of a certain layer performs the same transformation on its input. Signals proceed from the first (input) layer to the last (output) layer, possibly after passing through the hidden layers several times. [Gé19, KJ13]

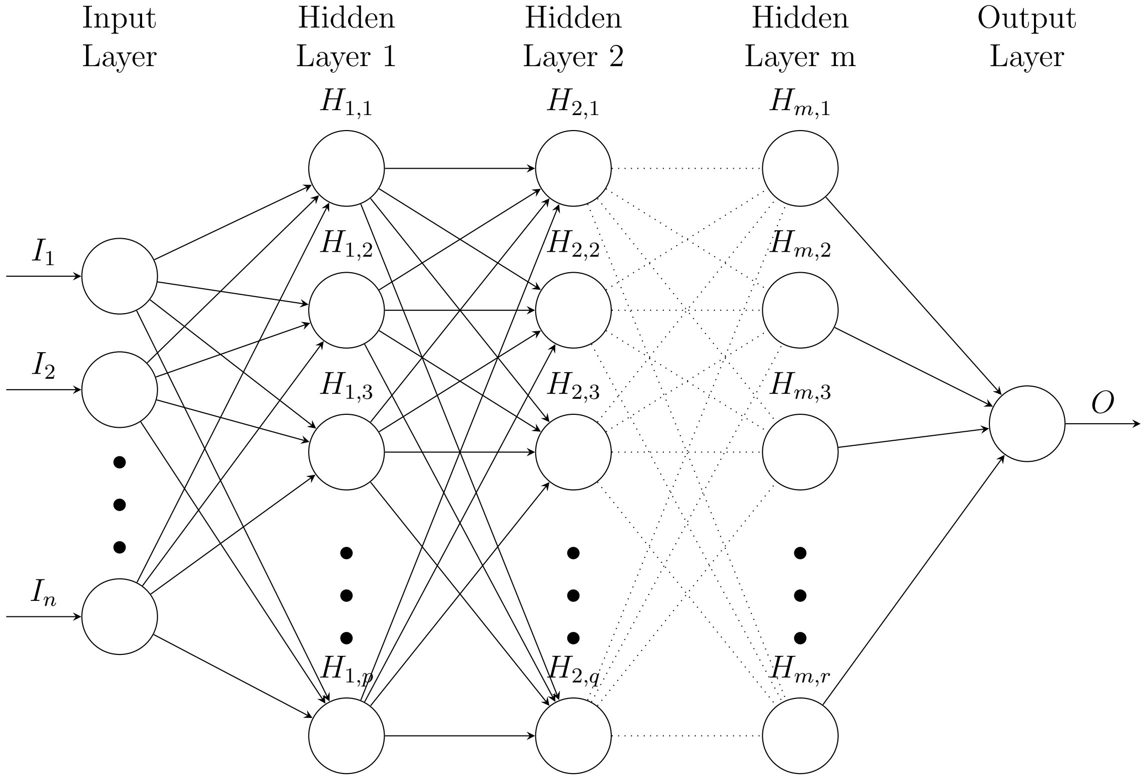


Figure 3.2: Structure of a ANN with m hidden layers

The amount of nodes in the input layer n depends on the how many features the data has. Designing the architecture of an ANN includes the decision on:

- The amount of layers m as well as their number of nodes.
- The types and connectedness of the hidden layers. Dense layers are fully connected (as depicted in Figure 3.2), whereas dropout layers set some random input units to zero during training time, which helps prevent overfitting. Further, there are flattening layers, which transform a two dimensional input into a one dimensional one by stacking it. In addition to that, there are also layers, that define the type of the network. Besides a fully connected layers, this work uses GRU [CvMG$^+$14] and LSTM [HS97] layers, as they are capable of learning the relationship between past

data values and current data values, representing that relationship in the form of learned weights [HCL+18]. Both LSTM and GRU belong to the class of recurrent neural networks. In contrast to feed-forward neural networks, which only pass information towards the output, recurrent neural networks cycle information through a loop and hence have the ability to remember values of the past. An LSTM network accomplishes that with using so-called LSTM units instead of regular nodes. The units usually consist of a cell, which remembers information over arbitrary time intervals, an input, output and forget gate, which control the information stream into and out of the cell. Hence, an LSTM unit can decide whether to keep the existing memory with its gates. It is therefore capable of transporting information assessed as important over a long distance and thus of capturing possible dependencies over long distances. Similar to the LSTM unit, a GRU contains gates controlling the information flow. The update gate determines how much information is passed and the reset gate controls how much information to forget. In difference to the LSTM unit, the GRU works without a memory cell. Comparative studies show, that LSTM and GRU networks perform equally better than fully connected networks when predicting time series. [CGCB14]

- The type of activation function to produce an output signal from the weighted sum of input signals in a node. The implemented ANN uses linear, Rectified Linear Unit (ReLU) ($ReLu(x) = max(x, 0)$), sigmoid ($sigmoid(x) = \frac{1}{(1+exp(-x))}$), softmax ($softmax(x) = \frac{exp(x)}{\sum_{j=1}^{k}(exp(x_j))}$) and tanh ($tanh(x) = \frac{((exp(x)-exp(-x))}{(exp(x)+exp(-x)))}$) activation functions [Cho21b].
- The learning rate, which defines the speed of learning at each step.
- The optimizer, which is an algorithm that computes how weights and learning rate have to be adapted in order to reduce the prediction loss. The most common optimizers are: Stochastic Gradient Descent (SGD), which is dependent on the first order derivative of the loss function; Root Mean Square propagation (RMSprop), which uses a moving average of squared gradients to normalize the gradient and therefore decrease the learning rate for large gradients; and Adam, which is a combination of RMSprop and SGD with second order momentum, which accelerates the gradient descent in the relevant direction as well as it dampens oscillations.
- The batch size, which is the number of samples to go through before updating the networks weights. A sample is a single set of data points of the dimension $feature\ size \times lookback\ timesteps$.
- The number of epochs, which defines the number of times that the learning algorithm will work through the entire training dataset.
- The train, validation and test split size. The complete data set is first split into test and non test data, where the test data is kept for evaluation finding optimal hyperparameters after completing the training phase. The non test set is further

split into train, used for training, and validation sets, used to evaluate if one training cycle has improved the model or not.

Since ANNs are able to take two dimensional and therefore multivariate time series as an input, no further adjustment is necessary in addition to hyperparameter selection.

Dynamic Threshold Calculation

The underlying work deals with automated monitoring of multiple channels, whose expected values vary according to changing environmental factors. This requires a rapid, general, and unsupervised approach to determine if predicted values are anomalous. The threshold computation presented in [HCL$^+$18] conquers diversity, non-stationarity, and noise issues with an algorithm that adapts to varying behavior and value ranges, and therefore only detects abnormally high or low prediction errors.

The main advantages over a constant threshold are, that it is not relying on expert knowledge, it is universally applicable instead of customized to the network topology and it is capable of not only covering point anomalies, but also contextual and sequential outliers.

The computation proceeds as follows. First, the prediction error is calculated by $e^{(t)} = |y^{(t)} - \hat{y}^{(t)}|$ where $y^{(t)} = x_i^{(t+1)}$ with i corresponding to the entry of the parameter to predicted, hence ORP in this case. Then from h historical error values, a one dimensional vector is created as $\mathbf{e} = [e^{(t-h)}, ..., e^{(t-l_s)}, ..., e^{(t-1)}, e^{(t)}]$. These errors $\mathbf{e}$ are then Exponentially-Weighted Moving Average (EWMA) smoothed ($\mathbf{e}_s$) to dampen spikes in errors that frequently appear with ANN predictions.

A set of candidate thresholds is computed by $\boldsymbol{\epsilon} = \mu(\mathbf{e}_s) + \mathbf{z}\sigma(\mathbf{e}_s)$, where $\mathbf{z}$ is an ordered vector of the k highest deviations of $\mathbf{e}_s$ above $\mu(\mathbf{e}_s)$. k is a tunable hyperparameter between two and ten. The actual threshold is then selected such that:

$$t = \frac{\frac{\Delta\mu(\mathbf{e}_s)}{\mu(\mathbf{e}_s)} + \frac{\Delta\sigma(\mathbf{e}_s)}{\sigma(\mathbf{e}_s)}}{|\mathbf{e}_a| + |\mathbf{E}_{seq}|^2} \tag{3.6}$$

with

$$\Delta\mu(\mathbf{e}_s) = \mu(\mathbf{e}_s) - \mu(\{e_s \in \mathbf{e}_s | e_s < \epsilon\}) \tag{3.7}$$

$$\Delta\sigma(\mathbf{e}_s) = \sigma(\mathbf{e}_s) - \sigma(\{e_s \in \mathbf{e}_s | e_s < \epsilon\}) \tag{3.8}$$

$$\mathbf{e}_a = \{e_s \in \mathbf{e}_s | e_s < \epsilon\} \tag{3.9}$$

$$\mathbf{E}_{seq} = \text{continuous sequences of } e_a \in \mathbf{e}_a \tag{3.10}$$

As a result, if all errors above the threshold t are removed, the greatest percent decrease in the mean and standard deviation of the smoothed errors $\mathbf{e}_s$ is achieved. [HCL$^+$18]

This can directly be applied, as the previous prediction computed in advance by ARIMAX and ANNs is providing the threshold calculation algorithm with the ORP value of one timestep.

3.2 Failure Identification

The failure identification can be conducted with two different algorithms, Naive Bayes and DTW k-means. Both are described in the following.

3.2.1 Naive Bayes

Naive Bayes is a probabilistic classifier that applies the Bayes' Theorem and assumes that all features are independent. It can be seen as a conditional probability model:

$$P(C_j \mid \mathbf{x}) = \frac{P(C_j)P(\mathbf{x} \mid C_j)}{P(\mathbf{x})} \tag{3.11}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector to be classified defined by n features and where C_j includes all possible k classes. Equation 3.11 gives a vector, called posterior probability, containing the probabilities for each class k that $\mathbf{x}$ belongs to that class. According to the chain rule for repeated applications of the definition of conditional probability, the numerator can be written as

$$\begin{aligned} &P(C_j)P(\mathbf{x} \mid C_j) = P(C_j, x_1, \ldots, x_n) \\ =&P(x_1 \mid x_2, \ldots, x_n, C_j)P(x_2 \mid x_3, \ldots, x_n, C_j) \ldots P(x_{n-1} \mid x_n, C_j)P(x_n \mid C_j)P(C_j) \end{aligned} \tag{3.12}$$

and assuming that all features are mutually independent,

$$P(x_i \mid x_{i+1}, \ldots, x_n, C_j) = P(x_i \mid C_j) \tag{3.13}$$

holds. Thus, 3.12 becomes

$$P(C_j, x_1, \ldots, x_n) = P(C_j) \prod P(x_i \mid C_j) \tag{3.14}$$

and further

$$P(C_j \mid \mathbf{x}) = \frac{1}{Z} P(C_j) \prod P(x_i \mid C_j) \tag{3.15}$$

with

$$Z = P(x) = \sum_j P(C_j)\ P(\mathbf{x} \mid C_j) = const. \tag{3.16}$$

Thus, when a failure is to be identified, the Naive Bayes probability model gives a probability for each class that the failure belongs to this class. Combining this model with a decision rule, which is

$$\hat{y} = \text{argmax} P(C_j) \prod P(x_i \mid C_j) \tag{3.17}$$

in the underlying case, gives the class $\hat{y}$ for the vector $\mathbf{x}$. [KJ13]
As the prior probability distribution, the likelihood of each feature is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{3.18}$$

where μ_y and σ_y are mean and standard deviation, respectively.
Because the input vector $\mathbf{x}$ has to be one dimensional, and the OPM data is two dimensional ($feature\ size \times lookback\ timesteps$), it needs to be flattened. This is done in two different ways:

- **Stacked**, where the features are concatenated one after another, such that the first entries are all timesteps of the first feature, then all timesteps of the second feature and so on. This results in a dimension of $1 \times (feature\ size * lookback\ timesteps)$.
- **Multiplied** where the normalized features of the same timestep are multiplied such that for each timestep there is only one value. Hence, a dimension of $1 \times lookback\ timesteps$ is achieved.

The main reason for investigating the Naive Bayes on the underlying problem is that it gives the exact probability for each class, thus when probabilities are relatively equally distributed, it can be defined as an unknown failure. Moreover, since the both training and execution time are short as the calculation is straight forward.
Although the assumption in 3.13 stating that all features are mutually independent, does not hold true for time series, there exist literature [AI20, Sch15], which applies the Naive Bayes classifier on time series data with acceptable performance.

The algorithm itself has no hyperparameters, so only the type of flattening and the probability threshold for a failure to be identified as a known one can be modified in this case.

3.2.2 Dynamic Time Warping (DTW) K-means

K-means is a clustering algorithm that divides n observations $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ into $K(\leq n)$ clusters $\mathbf{S} = S_1, S_2, \ldots, S_k$ with the minimum within-cluster variance. This minimum within-cluster variance is commonly measured with the Euclidean distance metric. It can be described mathematically as

$$\underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2 = \underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i \tag{3.19}$$

where $\boldsymbol{\mu}_i$ denotes the mean of all points in S_i. If a new observation is classified by an existing model, the distances of the observation to the center of the clusters are calculated and the class which distance is smallest is selected to be the one of the new observation.

[KJ13]
As this work deals with open-set classification, and also unknown classes shall be identified as such, a class probability is calculated by:

$$P(C_j \mid \mathbf{d}) = 1 - \frac{(k-1) \cdot d_j}{\sum_{i=1}^{k} d_i} \tag{3.20}$$

where $\mathbf{d}$ denotes the distances to the cluster centers and $\mathrm{argmax}_j(P(C_j \mid \mathbf{d}))$ is achieved for m with $d_m = \mathrm{argmin}(\mathbf{d})$. Hence, the probability is the complement of the ratio of the smallest distance to the mean of all distances.
Although the Euclidean distance metric works well for most clustering tasks, it is not ideal for time series clustering. This is, because it is not invariant to time shifts, as shown in Figure 3.3. Here, two univariate time series, which are identical and therefore belong to the same class, depicted in blue and red, are compared. In Figure 3.3 a), both time series are exactly above each other, which implies that they are synchronized. The black vertical lines, representing the distances of the two series at same moment, are all of the same length and thus the variance is zero. Figure 3.3 b) shows the same scenario, but the red time series is shifted two timesteps forward. Here, the variance is large as there are both small and large distances. Hence, two time series that are completely the same would potentially not be classified into the same class, if they have a small offset in time. This problem can be solved when using DTW, as shown in Figure 3.3 c).

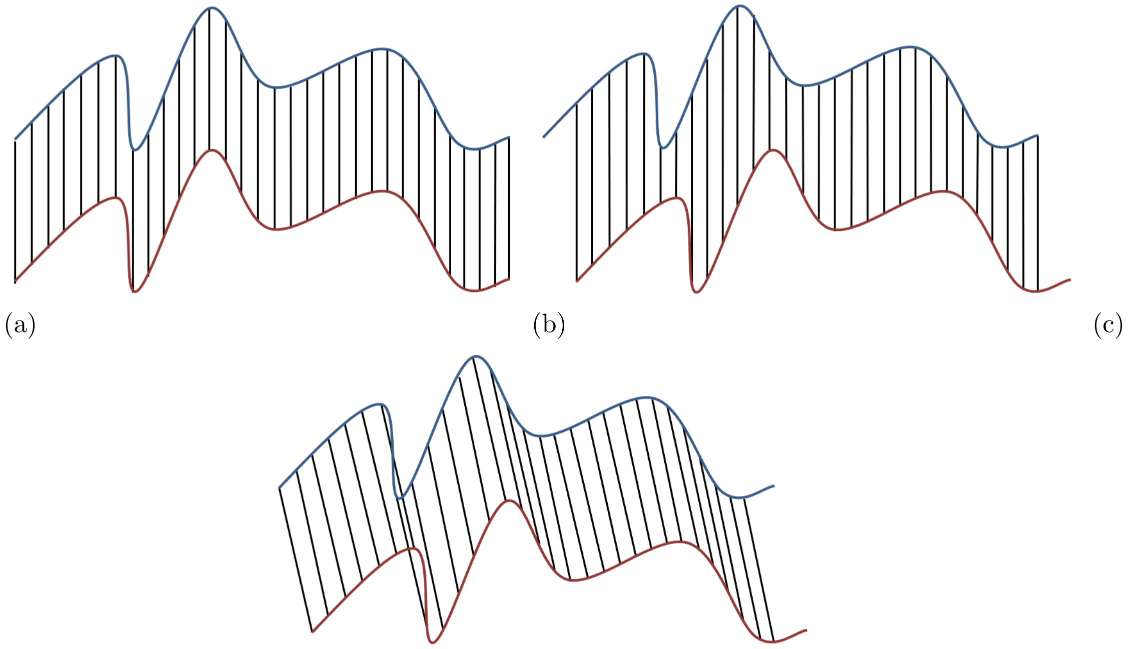

(a) (b) (c)

Figure 3.3: Distance measuring for k-means clustering with a) Euclidean distance without time shift b) Euclidean distance with time shift and c) DTW with time shift

The DTW of two time series $\mathbf{x} = (x_0, \ldots, x_q)$ and $\mathbf{y} = (y_0, \ldots, y_r)$ is determined by

$$DTW(\mathbf{x}, \mathbf{y}) = \min_{\pi} \sqrt{\sum_{i,j \in \pi} d(x_i, y_i)^2} \tag{3.21}$$

where $\pi = [\pi_0, \ldots, \pi_K]$ is a path satisfying the following properties:

- all entries are index pairs $\pi_k = (i_k, j_k)$ with $0 \leq i_k \leq q$ and $0 \leq j_k \leq r$
- $\pi_0 = (0, 0)$ and $\pi_K = (q, r)$
- for all $k > 0, \pi_k = (i_k, j_k)$ is related to $\pi_{k-1} = (i_{k-1}, j_{k-1})$ as $i_{k-1} \leq i_k \leq i_{k-1} + 1$ and $j_{k-1} \leq j_k \leq j_{k-1} + 1$

Hence, with DTW, not only time shifts, but also shrinking and stretching of time are taken into consideration when classifying time series. [TFV$^+$20]
Similar to Naive Bayes in Section 3.2.1, the input data needs to be one dimensional. Again, the two flattening methods stacking and multiplying are implemented.

Chapter 4

Implementation

This chapter analyzes the monitored OPM data with standard analysis methods in order to get a better understanding about the behavior of the data. Based on this, all preprocessing steps, that are necessary to implement the selected ML algorithms from Chapter 3, are described. Afterwards, the chapter explains how each algorithm is adapted to the underlying OPM data and which hyperparameters are used for this purpose. In order to address these issues, the following questions are being answered:

- How are the monitored parameters correlated to each other?
- What are the properties of the seasonal component?
- How does this seasonal component correlate to the ambient temperature?
- How is it ensured that the input data is continuous and consistent?
- How is the data scaled such that it is dimensionless?
- Which parameters from the OPM data are used as an input for the algorithms and why?
- Which methods are used to create enough training data for failure identification?
- How are the hyperparameters chosen for each algorithm?

All algorithms are implemented in Python 3.6. The ARIMAX model is based on the Statsmodels library [SP10], while the dynamic threshold calculation is implemented from scratch. The DTW k-means algorithm is implemented with the tslearn [TFV+20] library. The OCSVM and Naive Bayes are implemented using Scikit-learn (Sklearn) [PVG+11] and for the ANN models mainly the keras library [Cho21a] is used.
The training and testing of the models are performed on hardware components listed in Table 4.1

Artifact	Value
Processor	Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 2.70 GHz
Installed memory (RAM)	8GB
System type	64-bit Operating System, x64-based processor

Table 4.1: Hardware specifications

4.1 Data Analysis

The OPM data used for training and evaluation covers the period from the 17^{th} of November 2020 until the 2^{nd} of January 2021 and thus spans 46 days. This is, because within this time period, no rerouting took place and thus the same link with a length of 1792 km was used to log the data. Further, the OPM data within this time period was relatively stable and only few human interventions and tests were performed. As the parameters are monitored every 30 secs, about 130000 timesteps are recorded. 8000 datapoints, and thus about 67 hours, are removed to eliminate abnormal data sequences, which, for example, were recorded under test conditions not useful for this study.

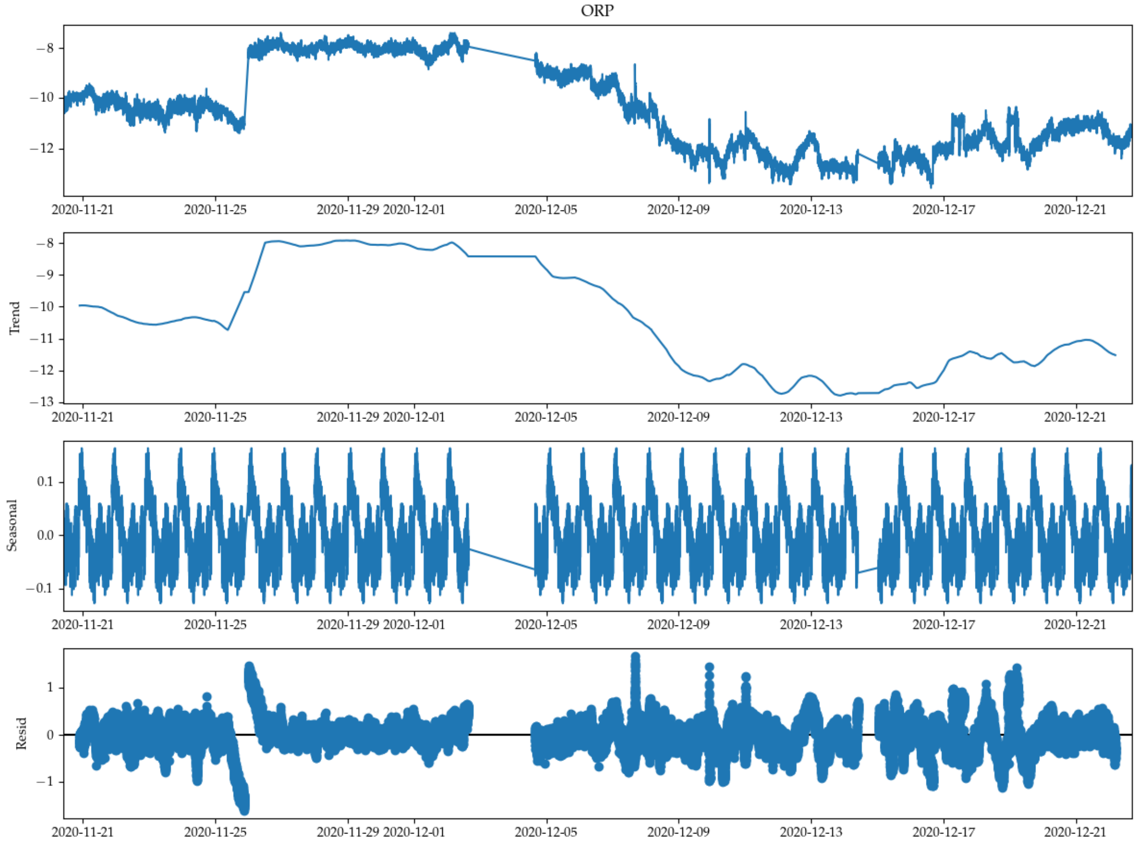


Figure 4.1: Seasonal decomposition

The first plot in Figure 4.1 shows the ORP within the described training period. The cut out data can be recognized by straight lines connecting the consecutive data points. The three bottom plots show the seasonal decomposition of the ORP. Seasonal decomposition is a mathematical procedure which transforms a time series into three different components; trend, seasonal and residual. The trend component describes the change in the data that is assumed to have a long-term effect, but which maintains a certain direction regardless of existing fluctuations. It is noticeable, that there are significant changes in trend within short time periods. The seasonal component includes all patterns that repeat during a fixed period of time. For the underlying ORP, this interval lasts 24 hours. The seasonality is analyzed in more detail in Section 4.1.2. The residual component can be interpreted as noise, it is calculated by subtracting the seasonal as well as the trend series from the original time series. This value is not constant, which indicates, that the ORP behaviour can not be perfectly described by only trend and seasonality.

4.1.1 Correlation of the Monitored Parameters

Correlation measures the relationship between two or more features, states or functions. For ML, knowledge of the underlying training data is important, as it can reduce both computation time and bias. If two variables are highly correlated, they carry the same information and thus, using only one of the variables as an input for a ML algorithm is sufficient to achieve the best result. In addition to that, knowledge about the correlation provides the basis for better interpretability.
The following section covers three different types of correlation methods: Pearson correlation, Spearman rank and Maximal Information Coefficient (MIC).

Pearson Correlation

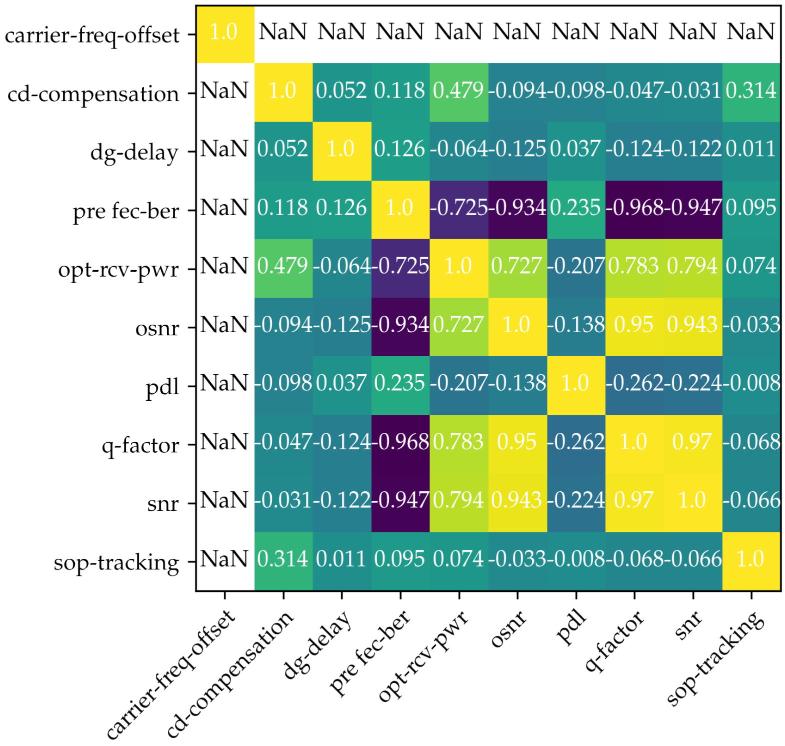
The Person correlation coefficient R_p determines the linear relationship between two variables. The coefficient can be understood as the standard deviation of all values plotted against each other (first variable on x-axis and second on y-axis) to the best fitted straight line of these points. Hence, the Pearson correlation of two vectors $\mathbf{x}$ and $\mathbf{y}$ is determined by

$$R_p(\mathbf{x},\mathbf{y}) = \frac{cov(\mathbf{x},\mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \tag{4.1}$$

where cov is the covariance and σ is the standard deviation. If n defines the length of the two vectors, it can also be expressed as

$$R_p(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i - \overline{\mathbf{x}})(y_i - \overline{\mathbf{y}})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{\mathbf{x}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{\mathbf{y}})^2}} \tag{4.2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean of $\mathbf{x}$. [Pea08]



	carrier-freq-offset	cd-compensation	dg-delay	pre fec-ber	opt-rcv-pwr	osnr	pdl	q-factor	snr	sop-tracking
carrier-freq-offset	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cd-compensation	NaN	1.0	0.052	0.118	0.479	-0.094	-0.098	-0.047	-0.031	0.314
dg-delay	NaN	0.052	1.0	0.126	-0.064	-0.125	0.037	-0.124	-0.122	0.011
pre fec-ber	NaN	0.118	0.126	1.0	-0.725	-0.934	0.235	-0.968	-0.947	0.095
opt-rcv-pwr	NaN	0.479	-0.064	-0.725	1.0	0.727	-0.207	0.783	0.794	0.074
osnr	NaN	-0.094	-0.125	-0.934	0.727	1.0	-0.138	0.95	0.943	-0.033
pdl	NaN	-0.098	0.037	0.235	-0.207	-0.138	1.0	-0.262	-0.224	-0.008
q-factor	NaN	-0.047	-0.124	-0.968	0.783	0.95	-0.262	1.0	0.97	-0.068
snr	NaN	-0.031	-0.122	-0.947	0.794	0.943	-0.224	0.97	1.0	-0.066
sop-tracking	NaN	0.314	0.011	0.095	0.074	-0.033	-0.008	-0.068	-0.066	1.0

Figure 4.2: Pearson correlation matrix

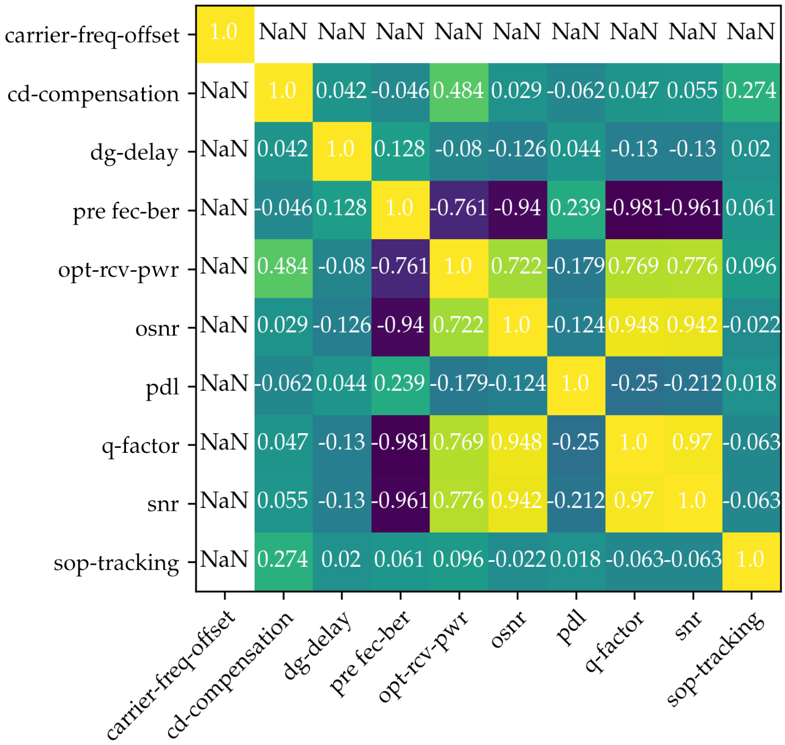
In Figure 4.2, the correlations are presented in a correlation matrix, where the correlation between two parameters can take values from minus one to one. Zero denotes no correlation at all, thus the two parameters are independent from each other. Positive one signifies completely positive and minus one completely negative linear correlation. Not a Number (NaN) implies that no significant statement could be made in terms of correlation.
It is shown, that the strongest positive linear correlations are between q-factor and SNR, OSNR and SNR, q-factor and OSNR. It becomes clear, that these three variables behave very similar in terms of linearity. The ORP is also following the same pattern, although it is not as strongly correlated. Looking at negative correlations, it is noticeable that pre-FEC BER is strongly negative linear dependant to the four previously mentioned variables. Besides the correlations of CDC with ORP and sop-tracking, only negligibly small linear correlations are present.

Spearman Rank

The Spearman rank R_s is a measure of correlation, which gives the statistical dependence between the ranks of two variables. It determines how well the relationship between two variables can be described using a monotonic function. It can be seen as the Pearson correlation of the rank between two variables, assessing monotonic linear and non linear relationships. Therefore it is computed by

$$R_s(\mathbf{x}, \mathbf{y}) = \frac{cov(rg_{\mathbf{x}}, rg_{\mathbf{y}})}{\sigma_{rg_{\mathbf{x}}} \sigma_{rg_{\mathbf{y}}}} \tag{4.3}$$

Analogous to the Pearson correlation, a Spearman rank of negative or positive one denotes a perfect monotonic function between the two variables being compared. [Spe08]



	carrier-freq-offset	cd-compensation	dg-delay	pre fec-ber	opt-rcv-pwr	osnr	pdl	q-factor	snr	sop-tracking
carrier-freq-offset	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cd-compensation	NaN	1.0	0.042	-0.046	0.484	0.029	-0.062	0.047	0.055	0.274
dg-delay	NaN	0.042	1.0	0.128	-0.08	-0.126	0.044	-0.13	-0.13	0.02
pre fec-ber	NaN	-0.046	0.128	1.0	-0.761	-0.94	0.239	-0.981	-0.961	0.061
opt-rcv-pwr	NaN	0.484	-0.08	-0.761	1.0	0.722	-0.179	0.769	0.776	0.096
osnr	NaN	0.029	-0.126	-0.94	0.722	1.0	-0.124	0.948	0.942	-0.022
pdl	NaN	-0.062	0.044	0.239	-0.179	-0.124	1.0	-0.25	-0.212	0.018
q-factor	NaN	0.047	-0.13	-0.981	0.769	0.948	-0.25	1.0	0.97	-0.063
snr	NaN	0.055	-0.13	-0.961	0.776	0.942	-0.212	0.97	1.0	-0.063
sop-tracking	NaN	0.274	0.02	0.061	0.096	-0.022	0.018	-0.063	-0.063	1.0

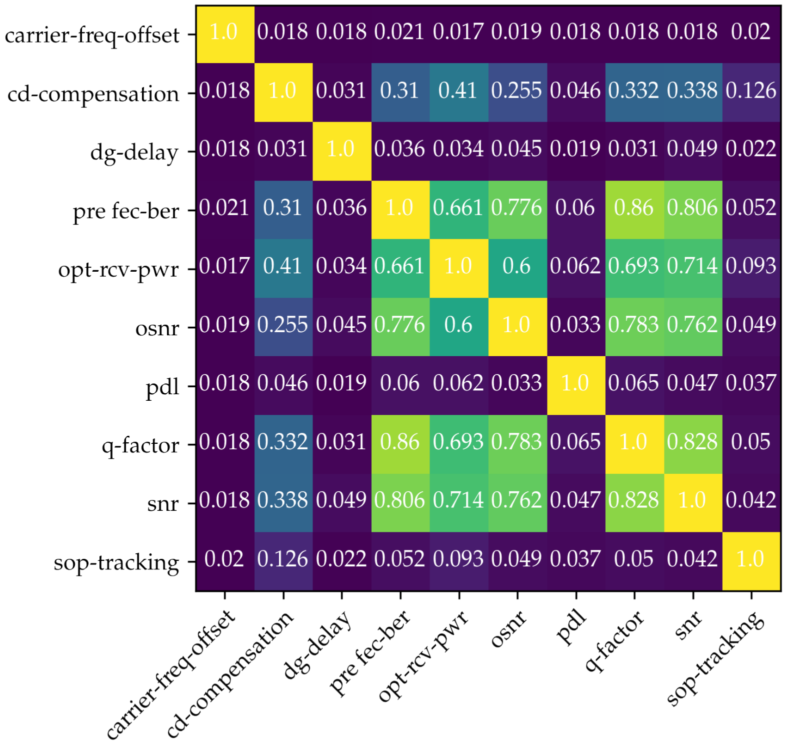
Figure 4.3: Spearman correlation matrix

From Figure 4.3 it becomes clear, that the Spearman correlations are quite similar to the Pearson ones. This implies that most of the correlations determined by the Spearman rank are linear correlations. The main difference to the Pearson correlation matrix is, that the correlation of pre-FEC BER with ORP, OSNR, q-factor and SNR has increased, especially evident with ORP. This indicates that there is some non-linear correlation that can be described by a monotonic function between ORP and pre-FEC BER.

Maximal Information Coefficient (MIC)

MIC is correlation metric capturing a wide range of both functional and non-functional relationships between variables. MIC takes values between zero, denoting statistical independence, and one, implying a completely noiseless relationship. No distinction is made

between positive or negative relation. MIC is the mutual information between $\mathbf{x}$ and $\mathbf{y}$, normalized by their minimum joint entropy.
MIC uses binning to apply mutual information on continuous random variables. Data binning a technique to reduce the effects of minor observation errors. The data values lying in a given small interval or bin are replaced by a value representative of that interval, often the mean value of that bin. In MIC, the bins are chosen such that the mutual information between the variables are maximal. Mathematically, this is achieved when $H(\mathbf{x}_b) = H(\mathbf{y}_b) = H(\mathbf{x}_b, \mathbf{y}_b)$, where H is the entropy of a variable. Thus, the bins have roughly the same size, as the entropies $H(\mathbf{x}_b)$ and $H(\mathbf{y}_b)$ are maximized by equal-sized binning. Further, each bin of $\mathbf{x}$ roughly corresponds to a bin in $\mathbf{y}$. [RRF$^+$11]



	carrier-freq-offset	cd-compensation	dg-delay	pre fec-ber	opt-rcv-pwr	osnr	pdl	q-factor	snr	sop-tracking
carrier-freq-offset	1.0	0.018	0.018	0.021	0.017	0.019	0.018	0.018	0.018	0.02
cd-compensation	0.018	1.0	0.031	0.31	0.41	0.255	0.046	0.332	0.338	0.126
dg-delay	0.018	0.031	1.0	0.036	0.034	0.045	0.019	0.031	0.049	0.022
pre fec-ber	0.021	0.31	0.036	1.0	0.661	0.776	0.06	0.86	0.806	0.052
opt-rcv-pwr	0.017	0.41	0.034	0.661	1.0	0.6	0.062	0.693	0.714	0.093
osnr	0.019	0.255	0.045	0.776	0.6	1.0	0.033	0.783	0.762	0.049
pdl	0.018	0.046	0.019	0.06	0.062	0.033	1.0	0.065	0.047	0.037
q-factor	0.018	0.332	0.031	0.86	0.693	0.783	0.065	1.0	0.828	0.05
snr	0.018	0.338	0.049	0.806	0.714	0.762	0.047	0.828	1.0	0.042
sop-tracking	0.02	0.126	0.022	0.052	0.093	0.049	0.037	0.05	0.042	1.0

Figure 4.4: MIC correlation matrix

Figure 4.4 shows the matrix of all MIC correlations. The main differences to the results of previous correlation analyses is, that the CDC is slightly correlated to the strongly correlated parameters mentioned earlier. Apart from that, the values of the correlations are slightly lower, which means that the correlation with entropy cannot be explained as well as with monotone linear and nonlinear relations.

4.1.2 Seasonality

The seasonal component of time series data is defined as variations that appear at specific regular time intervals. Seasonality may occur due to repetitive events such as vacation or rush hour. [Cip20]

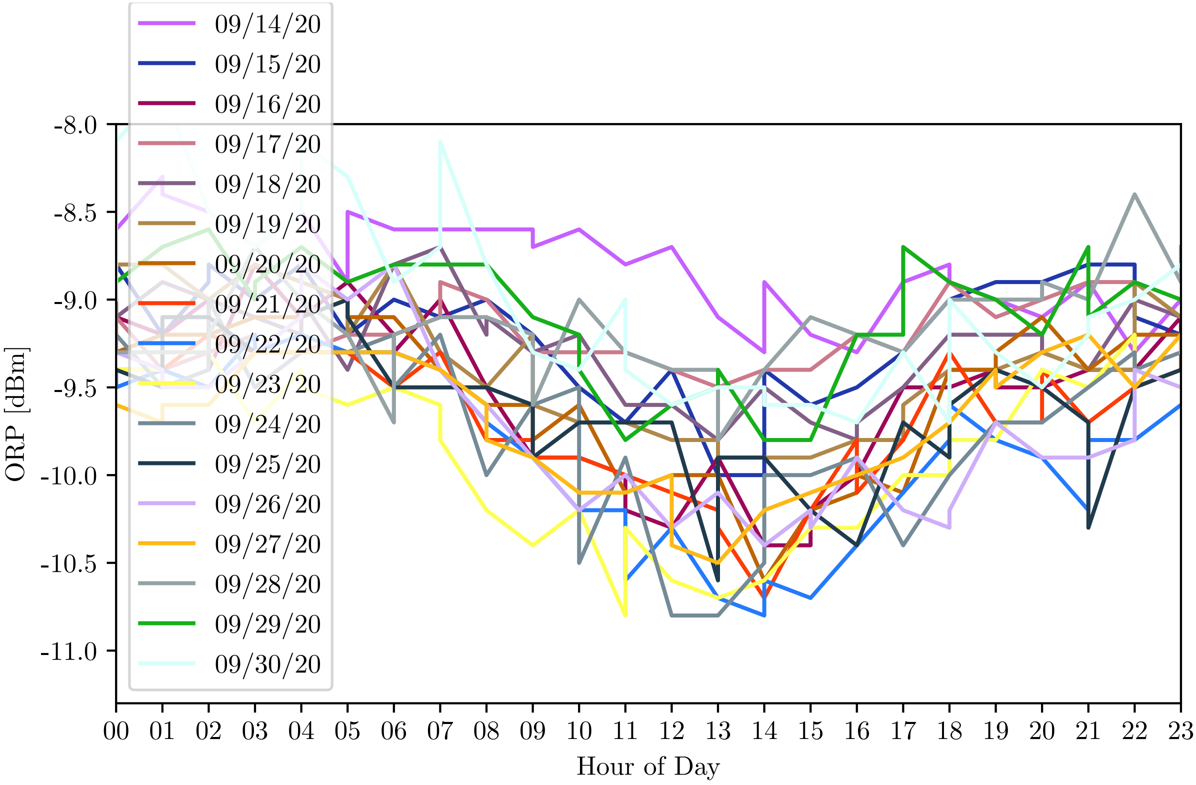


Figure 4.5: ORP for different days in September

Figure 4.5 shows the ORP of several days in September. It can be seen, that the underlying data is fluctuating within 24 hours and thus has a seasonality of one day.

Further investigations show, that this seasonality is not the same for all months. Figure 4.6 contains boxplots of the ORP distribution of each hour for several days for the months from August until December. Here, 50% of all datapoints lie in the coloured boxes. 25% of the data is represented by the black line above and the remaining 25% below the box. The median is shown by the line that divides the box into two parts. Half of the datapoints are greater and half are less than this value. The diamonds indicate outliers, which are very far away from the normal distribution.
From Figure 4.6, it is clear that the amplitude of the variations is lower in the winter months than in the summer months. The difference in mean power between midnight and noon in August and September lies in the range of 1.5-2 dBm, whereas this difference is about 0.5-1 dBm for November whereas for December, no seasonality can be recognized.

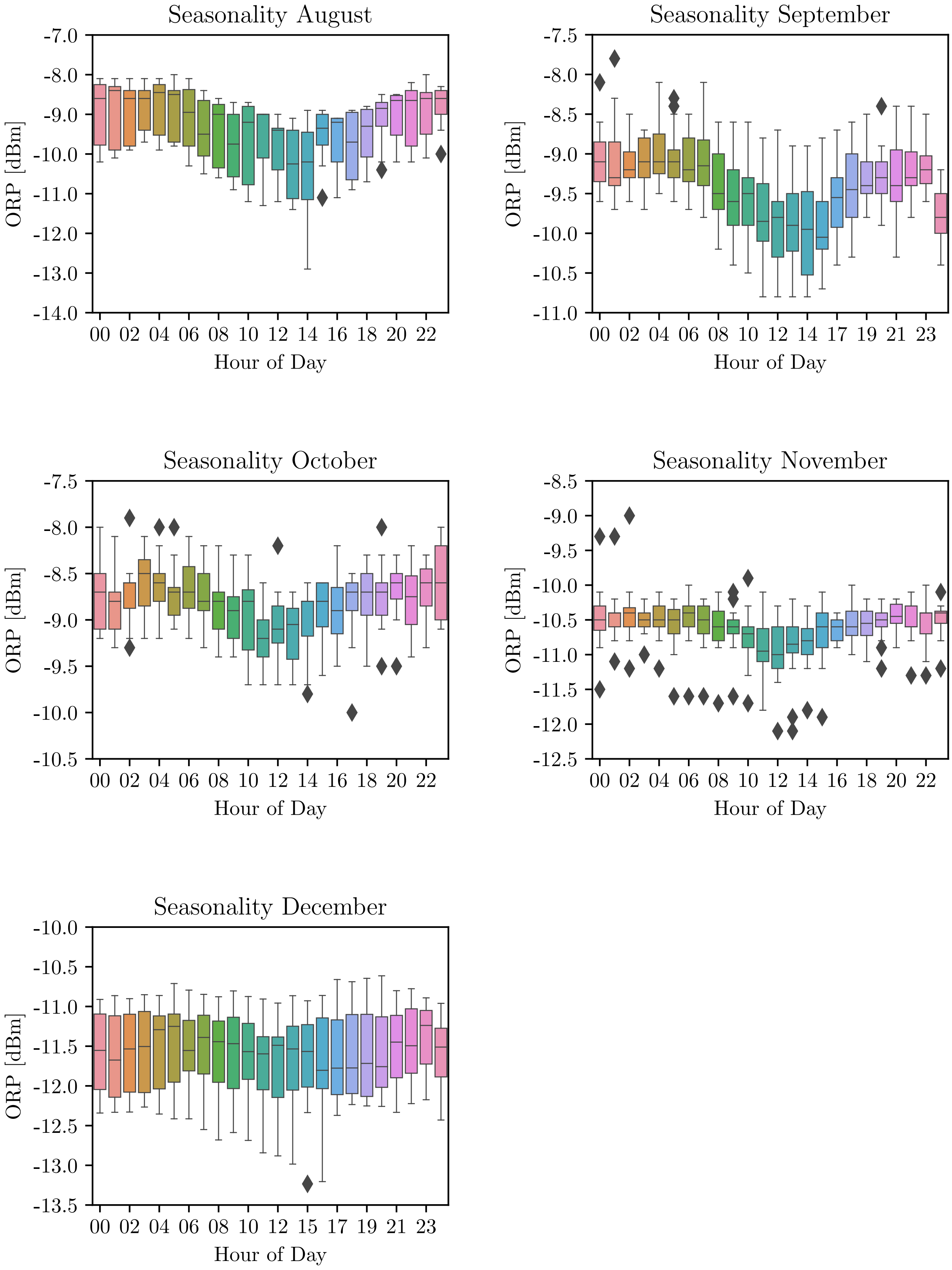


Figure 4.6: Hour-wise boxplot of the seasonalities of August, September, October, November and December

Moreover, there are relatively many outliers in October and November, while they occur very rarely in August and December.
One reason for this seasonality could be the ambient temperature, which is investigated in the following section.

4.1.3 Correlation to Ambient Temperature

This section focuses on whether the ambient temperatures of the locations through which the fiber passes are correlated with the seasonality and thus with the in- and decrease in ORP. Figure 4.7 shows the ambient temperatures of 27 sites for each hour in green and the ORP in pink. The time span covers two days in November, where the seasonality is relatively low (about 0.5-1 dBm) but present, compare Section 4.1.2.

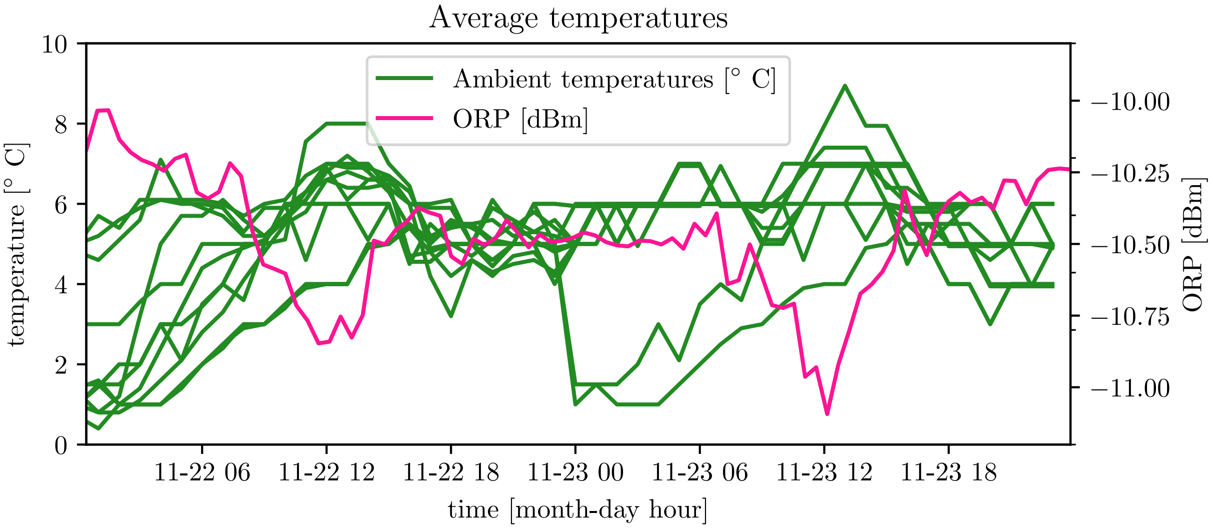


Figure 4.7: Temperatures and ORP for selected hours in November

From Figure 4.7, it is reasonable to assume that temperature and ORP are negatively correlated, since temperature is at its daily high just after noon, while ORP is at its daily low.
To generalize this theory, the minimum, maximum and average temperatures per day are plotted against the ORP in Figure 4.8. The Figure covers 26 days in November. Between the 17^{th} and 21^{st} of November, the average temperature rises from 3-4 °C to 10 °C and drops to nearly 0 °C shortly thereafter for most locations. Meanwhile, the ORP appears to respond to the temperature differences and therefore first decreases by about 1.5 dBm and then increases by 2 dBm, both taking place immediately after the temperature events described earlier.
An event which, however, is not completely consistent with the behavior described above,

is the temperature drop between the 6^{th} and the 9^{th}. During these days, ORP rises only by about 0.5 dBm, although the average temperature drop is 8 °C. If the ORP would be negatively correlated to the ambient temperature, one explanation for the poor increase in ORP could be, that the maximum temperatures fall far less steeply compared to the temperature drop of the 19^{th}.

The Figures 4.7 and 4.8 are only small sections of data examples, it is not possible to conclude from them whether there is a correlation between ORP and ambient temperature or not.

However, what speaks against this theory is the work of [LCT00]. Lewis et al. carried out an experiment, where a silica-fiber Raman amplifier pumped at a wavelength of 1455 nm, operating at a room temperature of 300 K was cooled down to 77 K with liquid nitrogen. Shortly after cooling, a decrease in the optical noise figure and correspondingly a decrease in the noise emission rate at wavelengths close to the pump was detected. Moreover, the amplifier gain decreased and underwent a change in spectral shape. Simultaneously, an increase in the distributed fiber loss, particularly at longer wavelengths, was measured. The increase in fiber loss would result in a decrease in ORP, which is the opposite behavior of what the underlying network exhibits.

Why this is stronger during the summer months, could be explained based on several factors. On the one hand, the temperature changes during one day are greater in summer than in winter. On the other hand, on the 22^{nd} of October, the loopback location of the EON was changed such that the length of network path was shortened from 3751 km to 1792 km. Therefore, the temperature differences around a light path are smaller because the locations it passes through are fewer and they are closer together. Further, the ORP fluctuations could be triggered by some in-line equipment, which are fewer in shorter network paths.

As this work deals with the analysis of OPM data only as a side issue, no further investigations have been taken at this point.

4.2 Data Preprocessing

This section explains which transformations of the initial monitored data are required to apply the algorithms described in Chapter 3.

4.2.1 Interpolation for Equalized Time Intervals

Since the OPM data is discrete real field data that is not always monitored at exactly the same time interval, it is transformed such that it provides values for all monitored parameters exactly every 30 seconds. This transformation is done by first creating values for every second with linear interpolation. Then, every 30^{th} value is selected for further proceedings.

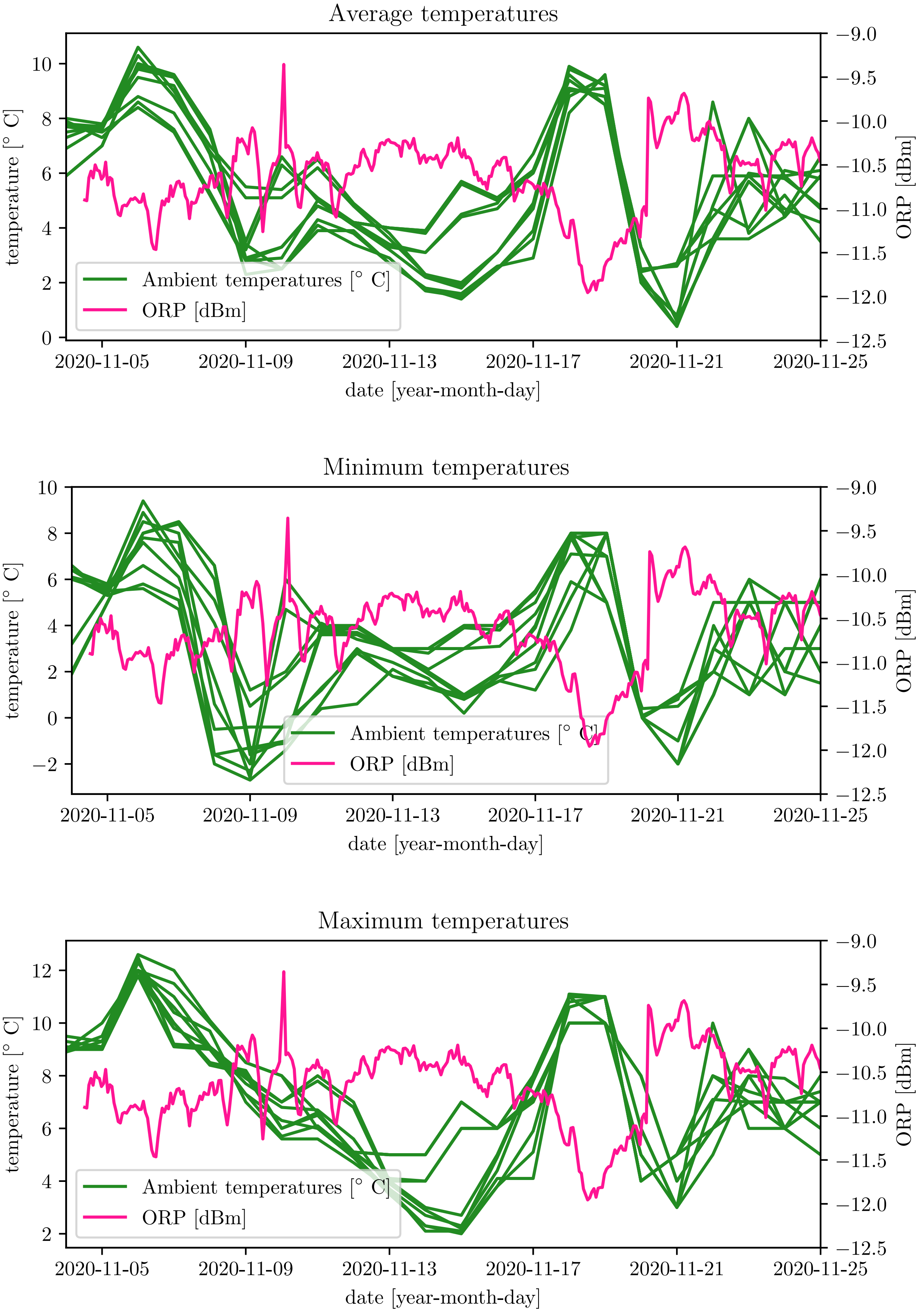


Figure 4.8: Temperatures and ORP for selected days in November

4.2.2 Feature Scaling

As all parameters monitored have different ranges, the data is scaled with normalization in order to have an unit-less input for the ML algorithms. This is necessary because most algorithms can only process scaled input data. In addition, the numerical stability is improved. [KJ13]
For the failure detection, the OPM data is transformed such that each feature individually ranges from zero to one instead of from its maximum to its minimum. Every entry x_i of each feature vector $\mathbf{x}$ is converted to [PVG⁺11]:

$$x_{i,MinMaxScaled} = \frac{x_i - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \tag{4.4}$$

For the identification, standard scaling is used. Applying this method, all features result in mean $\mu(\mathbf{x}) = 0$ and standard deviation $\sigma(\mathbf{x}) = 1$. Hence, every entry x_i of each feature vector $\mathbf{x}$ is transformed to [PVG⁺11]:

$$x_{i,StandardScaled} = \frac{x_i - \mu(\mathbf{x})}{\sigma(\mathbf{x})} \tag{4.5}$$

4.2.3 Feature Selection

Choosing the right parameters as an input plays an important role in terms of accuracy. Fewer parameters not only lead to decreased computational time and complexity, it might also result in a model with better interpretability. In addition, models can be significantly improved by omitting imprecise features or those with degenerate distributions. [KJ13]
Of all 11 monitored parameters mentioned in Section 2.3.2, only pre-FEC BER and ORP are selected as input features.
This is justified based on a combination of data analysis and expert knowledge. As one goal of the failure detection is to predict the ORP as accurate as possible, the historic values for the ORP are essential for the prediction. Furthermore, ORP is also important for detection with OCSVM and error identification, since all failures will at some point have an impact on ORP. Based on this, other parameters can be selected to add further information to the input data. Only features, which correlate to ORP are chosen, as they contribute to predicting ORP. Hence, pre-FEC BER, OSNR, q-factor and SNR would be good candidates. Moreover, it is sufficient to choose only one of these four parameters, since the correlation analysis from Section 4.1.1 shows that they all carry similar information due to their correlation with each other.
Finally, pre-FEC BER is selected as the second input feature besides ORP. This is, because it is known for sure that with this parameter, every Bit is being directly monitored, and not computed from another parameter. Thus it is reliable and up-to-date. Moreover, pre-FEC BER has both linear and nonlinear correlation to ORP, while the other parameters have mostly linear correlation. Therefore, it might carry more wide-ranging information.

4.2.4 Data Augmentation

Data augmentation increases the amount of training data by adding slightly modified versions of existing samples to it. It works as a regularizer and helps to reduce overfitting, since the performance of ML models rely heavily on the amount of labeled training data. [WSS+20]
In this work, data augmentation is needed only for the failure identification part, since sufficient fault-free data is available for the detection. Since failures are very rare and it is even less likely that the same fault will occur more than once in a short period of time, the different types of failures are replicated with data augmentation to increase class sizes. This is both helpful for the initial training with the three predefined failures as well as with failures which are added to the failure library and learned later. In particular, the augmentation consists of three steps:

- **adding noise:** between -50% and +50% of the standard deviation
- **window warping:** shrinking and stretching randomly selected slices of the time series
- **creating an offset:** shifting the time window by ± 0.5 to 1 min

4.3 Failure Detection

The following section explains how the failure detection algorithms described in Chapter 3 are implemented. The main focus lies on hyperparameter optimization for each algorithm. The results of the hyperparameter optimizations described in this section are summarized in Table 4.2. Only failure-free OPM data is used for both training and hyperparameter optimization, since the algorithms should not be overfitted to the known failures from Section 2.3.3. In the context of failures detection, the failures are only used for comparing and evaluating the different algorithms in Chapter 5.
For all algorithms that are predicting ORP, a span of 10 min, hence 20 timesteps, are taken to predict the next value.

4.3.1 Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX)

As explained in Section 3.1.2, when implementing the ARIMAX algorithm, values for the hyperparameters p, d and q have to be selected. To determine the most suitable ones, a grid search is performed in order to iteratively go through all possible combinations of the parameters. As the complexity increases with the order of p, d and q and thus also the required computing time, only values until four are taken into account. Chapter

Model	**Hyperparameter**	**Value**
ARIMAX	p	0
	d	1
	q	4
ANN	Type	
	# Layers	Flattening, Dropout and three Hidden Layers
	# Nodes	Funnel shaped layout: 5120,128,64,16
	Activation function	ReLU
	Optimizer	Adam
	Batch size	50
	# Epochs	100
	Test split	0.25
	Validation split	0.2
OCSVM	Kernel	Polynomial, degree = 4
	ν	0.0005
	k	5

Table 4.2: Selected hyperparameters (HPs) for failure detection

5 shows, that for the ARIMAX algorithm, computational time gets a critical parameter when predicting many ORP values.
The optimal set of parameters is defined by the set with the lowest Akaike Information Criterion (AIC) value. It is calculated as follows:

$$AIC = 2K - 2ln(\hat{L}) \tag{4.6}$$

where K is the number of estimated parameters and $\hat{L}$ is the maximum value of the likelihood function for the model. Thus, it gives a relative measure of whether the model is a good fit to the data, also taking into account how much complexity the model has. If two models fit the data equally good, the model with less complexity will be assigned a lower AIC score. [deL92]

The results of the grid search are shown in Table 4.3. Combinations that are not appearing in the Table are leading to numerical misspecifications and thus no AIC can be calculated. All values lower than 160000 are marked as bolts, underlining that all these combinations either have $p = 0$ or $q = 0$, while almost all $d = 1$. $d = 1$ signifies that differencing exactly one time is optimal to get stationary data. It is also clear, that combinations with $p = 0$ perform slightly better than combinations for $q = 0$, which means that the autoregressive term can be neglected. Hence, no lags of the ORP value are taken into account meaning that the algorithm disregards underlying the seasonality within 24 hours. One reason why the algorithm is not detecting the seasonality could be the very small ratio of time window taken into account for calculation (20 timesteps) to the time window of the seasonality (2880 timesteps). Nevertheless, as the aim of this work is to give a fast failure detection

(p, d, q)	AIC
(0, 0, 0)	242235.41
(0, 0, 1)	150510.48
(0, 0, 2)	90437.76
(0, 0, 3)	51319.24
(0, 0, 4)	21925.19
(0, 1, 0)	-144771.11
(0, 1, 1)	-159119.57
(0, 1, 2)	**-164499.76**
(0, 1, 3)	**-164606.00**
(0, 1, 4)	**-164643.79**
(0, 2, 0)	-66120.07
(0, 2, 1)	-133546.94
(0, 2, 2)	-142838.04
(0, 2, 3)	-148255.06
(0, 2, 4)	-148253.86
(1, 1, 0)	-150607.53
(1, 2, 0)	-91497.37
(2, 1, 0)	**-160000.20**
(2, 2, 0)	-115086.91
(3, 0, 0)	**-161069.65**
(3, 1, 0)	**-161424.25**
(3, 2, 0)	-125724.41
(4, 1, 0)	**-162676.02**
(4, 2, 0)	-132203.56

Table 4.3: ARIMAX HP optimization

method, the amount of timesteps taken into account is not increased. While $p = 0$ and $d = 1$, the higher q, the lower AIC. Setting q to four means that the model takes into account four preceding error terms.

4.3.2 Artificial Neural Network (ANN)

Since training a ANN takes a relatively long time compared to other algorithms and there are a large number of hyperparameters, a combination of knowledge about the general behavior of ANNs and randomized search is used to find the optimal hyperparameters.
Even though grid search is a good method for models with few hyperparameter combinations, it is not ideal for large search spaces. Trying out all possible combinations of many hyperparameters takes up too much time. Randomized Search however, evaluates a given number of random combinations by selecting a random value for every hyperparameter at

each iteration. Therefore, randomized search finds better models within a limited evaluation time by effectively searching a larger configuration space rather than going through each possible combination in a sequential manner. [BB12]
While the best values for activation function, batch size, amount of epochs, learning rate and type of optimizer are found by randomized search, the architecture of the ANN is being predefined.
While shallow networks with only one hidden layer might be sufficient to learn complex functions, deeper networks containing more hidden layers have a much higher parameter efficiency, resulting in less neurons and thus less computation time for training [Gé19] Therefore, three hidden layers are chosen in addition to a flattening and a dropout layer. The flattening layer is needed for dimensionality reduction of the time series and the dropout layer ensures that the model does not overfit by ignoring randomly selected nodes (15% of all nodes of that layer) at each iteration [SHK+14].
The amount of neurons in the in- and output layer is defined by the task, in this case the input is of dimension $feature\ size \times lookback\ timesteps\ = 2 \times 20$ and the output is one dimensional. The neurons of the hidden layers are selected to be decreasing for ascending layers so that they form a funnel. This is a common method because it allows many low-level features to merge into far fewer high-level features [Gé19]. The architecture of the ANN is shown in Figure 4.10.
To build the model, 90% of the complete data set is being used. 80% of this data is used for training and the remaining 20% is used for validating the model. The rest of the data, which is 10%, is held out for testing in a separate part of the pipeline. Before dividing the data into those three sets, it is shuffled to make the sets as homogeneous as possible [KJ13].

The remaining hyperparameters are found by a randomized search with 33 runs, which is visualized in Figure 4.10. Each line represents one combination of hyperparameters, which is used to train an ANN, and the colour of it indicates the R^2 score of its validation set. The R^2 score is used to evaluate the performance of a regression model by comparing the predicted to the actual value. When the predicted value is exactly the same as the actual value, the R^2 score is 1. The underlying math is explained more in detail in Section 5.1.1. The main results of the randomized search are, that large batch sizes as well as a low number of epochs are leading to good validation accuracy. Further, tanh activation function has proven not to be suitable for the underlying data. The best combination within this randomized search however, is given by: batch size = 50, amount of epochs = 100, learning rate = 0.03321, optimizer = Adam and activation function = ReLU. The resulting R^2 score is 0.9955.
It is assumed that the same hyperparameters (excluding the activation function) are ideal for the LSTM and the GRU model.

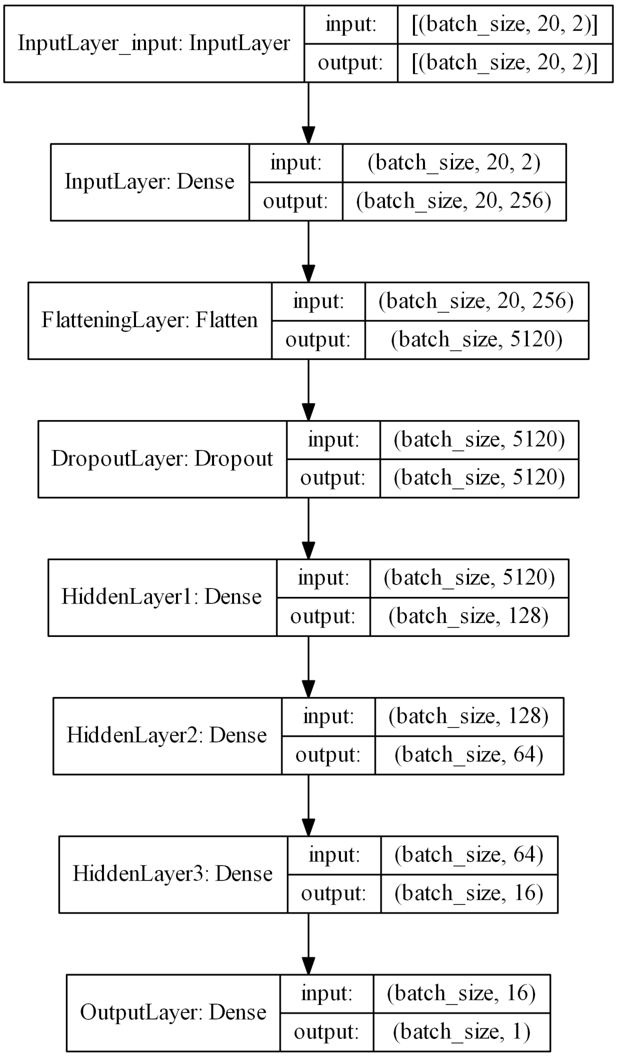

InputLayer_input: InputLayer
input: [(batch_size, 20, 2)]
output: [(batch_size, 20, 2)]
InputLayer: Dense
input: (batch_size, 20, 2)
output: (batch_size, 20, 256)
FlatteningLayer: Flatten
input: (batch_size, 20, 256)
output: (batch_size, 5120)
DropoutLayer: Dropout
input: (batch_size, 5120)
output: (batch_size, 5120)
HiddenLayer1: Dense
input: (batch_size, 5120)
output: (batch_size, 128)
HiddenLayer2: Dense
input: (batch_size, 128)
output: (batch_size, 64)
HiddenLayer3: Dense
input: (batch_size, 64)
output: (batch_size, 16)
OutputLayer: Dense
input: (batch_size, 16)
output: (batch_size, 1)

Figure 4.9: ANN architecture

4.3.3 One-Class Support Vector Machine (OCSVM)

To find the best values for the hyperparameters of the OCSVM algorithm explained in Section 3.1.1, a grid search with the validation split parameter k, the ratio of outliers ν and kernel function is performed. When executing a grid search, all possible parameter combinations are used to build a model and then the performance is measured based on a evaluation test set, which is not used for training.

Validation split is used to assess how a model can predict or classify new data of an independent data set not used for estimating it, in order to reduce overfitting and bias during the training. Therefore, the shuffled training data is randomly split into k sets of equal size. The model is fit using all samples except the ones of one set, called validation set, from which a evaluation metric is computed for the model. This is repeated until all k sets

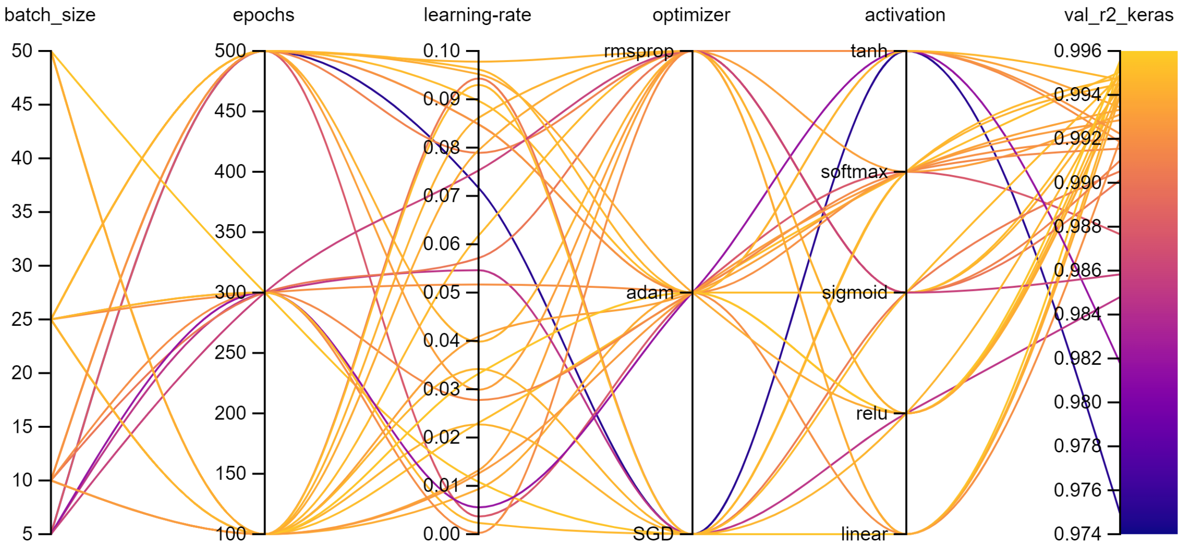


Figure 4.10: ANN HP optimization

are left out once. The model, which has the best performance on its validation set, is than selected as the final model.[KJ13]
Tables 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 and 4.10 show the number of evaluation data points lying outside the sphere computed by the OCSVM for the different HP combinations. These points would thus be misclassified as outliers, although they belong to the faultless dataset. Hence, the smaller the metric, the better the combination of hyperparameters. Each table shows the values for a different kernel function; Radial Basis Function (RBF), linear, polynomial with degree 2, 3, 4 and 5, and sigmoid function. These are the most common and widely-used kernel functions [BM14]. Across the columns, ν is iterated from 0.01% to 10% outliers. The rows show different values for k going from 2 to 50.
From the results in the Tables it becomes clear, that kernel functions with the best performance are polynomial functions with a degree of two and four. They both perform with 100% accuracy (no misclassified datapoints) for small ν. Table 4.8 shows, that the model is relatively stable, also if ν gets large. Therefore, the hyperparameters for OCSVM are selected as: kernel = polynomial with degree = 4, $\nu = 0.05\%$ and $k = 5$.

4.4 Failure Identification

This section explains how the failure detection algorithms from Chapter 3 are implemented. The main focus lies on selecting a flattening technique and a probability threshold, which defines the border between known and unknown failures, for both algorithms. The results of this section are summarized in Table 4.11.

ν \ k	2	5	10	20	50
0.0001	40	12	12	12	12
0.0005	12	12	12	40	40
0.001	40	40	40	40	40
0.005	40	40	40	40	220
0.01	220	220	220	220	220
0.05	220	220	220	220	220
0.1	220	220	220	220	220

Table 4.4: OCSVM HP optimization, RBF kernel

ν \ k	2	5	10	20	50
0.0001	4668	4668	6	6	6
0.0005	6	6	6	6	6
0.001	6	6	6	40	40
0.005	40	40	40	40	228
0.01	228	228	228	228	488
0.05	488	488	488	488	488
0.1	488	488	488	488	488

Table 4.5: OCSVM HP optimization, linear kernel

ν \ k	2	5	10	20	50
0.0001	50	50	0	0	0
0.0005	0	0	0	18	18
0.001	18	48	48	34	34
0.005	34	34	34	34	34
0.01	34	34	480	480	480
0.05	480	480	480	480	480
0.1	480	480	480	480	4668

Table 4.6: OCSVM HP optimization, polynomial kernel with degree = 2

ν \ k	2	5	10	20	50
0.0001	36	36	36	36	6
0.0005	6	6	6	2	4
0.001	4	4	4	28	28
0.005	28	28	28	28	130
0.01	130	130	130	130	526
0.05	526	526	526	526	526
0.1	526	526	526	526	526

Table 4.7: OCSVM HP optimization, polynomial kernel with degree = 3

4.4.1 Naive Bayes Classifier

As discussed in Section 3.2.1, when implementing the Naive Bayes classifier, only the type of flattening and the probability threshold for a failure to be identified as a known one have to be determined.

The mean probability of stacked and multiplied flattening has proven to be the best method, as the calculated probabilities for each class are very high in most cases. This holds not only for known failures but also for unknown ones. This behaviour can be eliminated by combining the two methods, because often the two different flattening methods give high probabilities for different classes when classifying an unknown failure.

The probability threshold, which defines whether a failure is classified as known or unknown, is specified by a trade off between sensitivity and fall-out rate.

Sensitivity, also called recall or True Positive Rate (TPR), defines the ratio of correctly identified known failures to all known failures and should thus be maximized. Mathematically, it is derived by:

$$TPR = \frac{TP}{TP + FN} \tag{4.7}$$

where True Positive (TP) are correctly and False Negative (FN) incorrectly identified known failures, see Table 4.12. In this case, as the identification is not binary, FN also

ν \ k	2	5	10	20	50
0.0001	526	526	526	0	0
0.0005	0	0	0	4	4
0.001	4	4	4	20	22
0.005	22	22	22	22	146
0.01	146	146	146	146	146
0.05	146	146	146	146	146
0.1	146	146	146	146	146

Table 4.8: OCSVM HP optimization, polynomial kernel with degree = 4

ν \ k	2	5	10	20	50
0.0001	146	146	146	146	146
0.0005	146	146	146	146	146
0.001	146	146	146	12	18
0.005	18	18	18	186	50
0.01	50	50	50	50	50
0.05	50	50	50	50	50
0.1	50	50	50	50	50

Table 4.9: OCSVM HP optimization, polynomial kernel with degree = 5

ν \ k	2	5	10	20	50
0.0001	6	6	6	6	6
0.0005	6	6	6	6	6
0.001	6	36	36	36	36
0.005	36	36	36	36	36
0.01	36	36	36	36	36
0.05	36	36	36	36	36
0.1	36	36	36	36	36

Table 4.10: OCSVM HP optimization, sigmoid kernel

includes known failures, which are classified as wrong known failures, e.g. if Failure 1 was classified as Failure 2. The fall-out rate or FPR, is calculated by

$$FPR = \frac{FP}{TN + FP} \tag{4.8}$$

and thus defines the proportion of misidentified unknown failures among all unknown failures. The aim is to minimize this metric. [Gé19]

Figure 4.11 shows the TPR and FPR for different probability thresholds. The curves are calculated by building the Naive Bayes model on all known failures except for one and the left out failure is then used to be identified as an unknown failure. This is repeated for all four failure classes; power degradation, inter-channel-interference, power drop and no failure.

As the FPR is ideal for small values, its y-axis is inverted. While TPR reaches almost 98%, meaning almost all known failures are identified as such, the FPR is relatively high for all thresholds. Hence, many unknown failures are misclassified, even if the probability threshold is very high.

The best trade off between TPR and FPR is defined by the intersection of the two lines plotted in Figure 4.11, as it maximizes TPR while minimizing FPR at the same time. At this point, the threshold is 0.72 and the values for TPR and FPR result in 0.94 and 0.57, respectively.

Model	**Flattening method**	**Threshold**
Naive Bayes	mean of stacked and multiplied	0.72
DTW k-means	stacked	0.42

Table 4.11: Selected HPs for failure identification

Identified \Actual	Known Failure	Unknown Failure
Known Failure	TP	FP
Unknown Failure	FN	TN

Table 4.12: True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) confusion matrix

Figure 4.12 shows the Receiver Operating Characteristic (ROC) curve, which is another way to plot the TPR against the FPR for different thresholds. In this plot, the best possible classifier would yield a point in the upper left corner at (0,1) of the ROC space, representing 100% sensitivity, hence no misclassified known failures, and 100% specificity, hence no misclassified unknown failures. The diagonal line represents the ratio of TPR and FPR for a random guess. All points lying above the diagonal line imply that the classification method used is better than random, and the further it lies to the upper left, the better. Again, Figure 4.12 makes it clear that TPR is really good while FPR is relatively high.

4.4.2 Dynamic Time Warping (DTW) K-means

Analogous to the Naive Bayes classification model, fitting the DTW k-means algorithm to the data consists of two decisions, choosing the flattening technique and the probability threshold.

As a flattening method, stacking the ORP and pre-FEC BER has emerged as the best method for DTW k-means.

For choosing the optimal probability threshold, the TPR and FPR curves are calculated the same way as for Naive Bayes, leaving out one class for every run. Figures 4.13 and 4.14 show that, exactly opposite to Naive Bayes, FPR is really good for most of the thresholds. TPR is not as high as for Naive Bayes, but still relatively good. The best fit is given by a probability threshold of 0.42, resulting in TPR = 0.76 and FPR = 0.06.

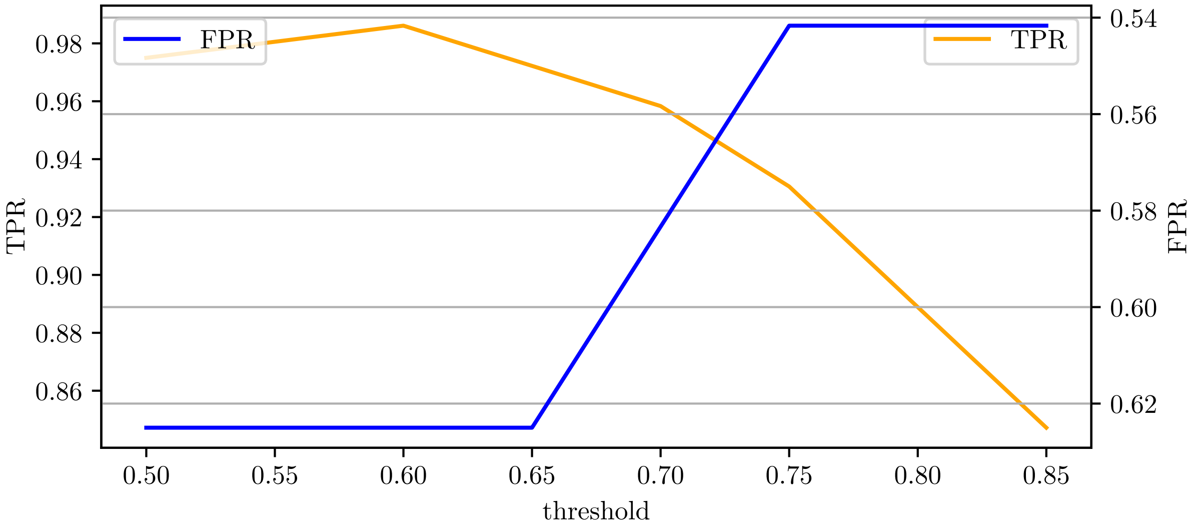


Figure 4.11: TPR and FPR of Naive Bayes

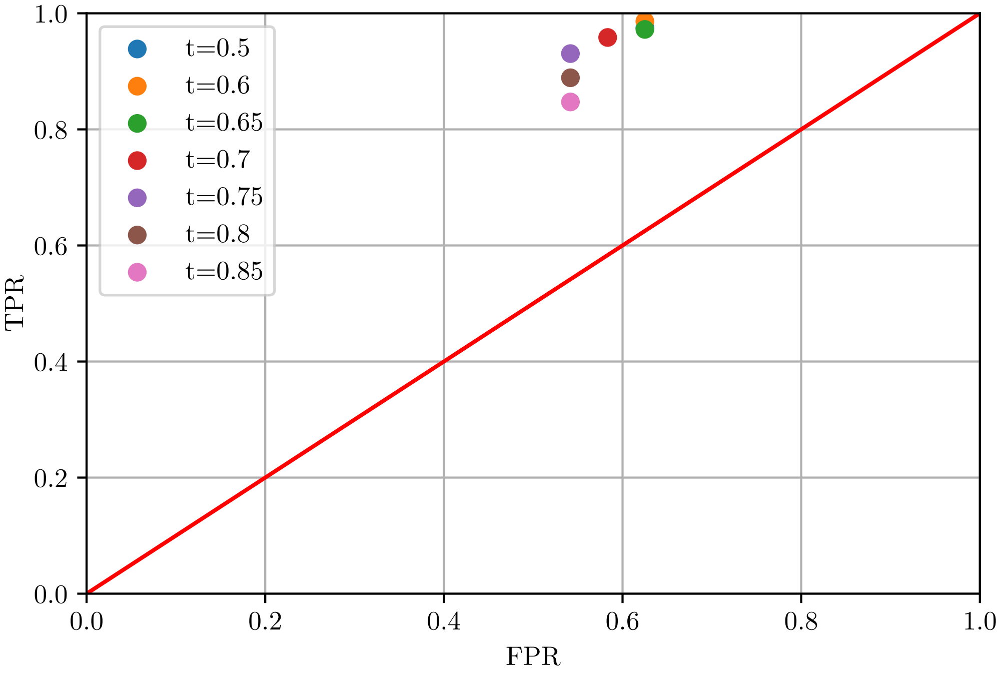


Figure 4.12: ROC curve of Naive Bayes

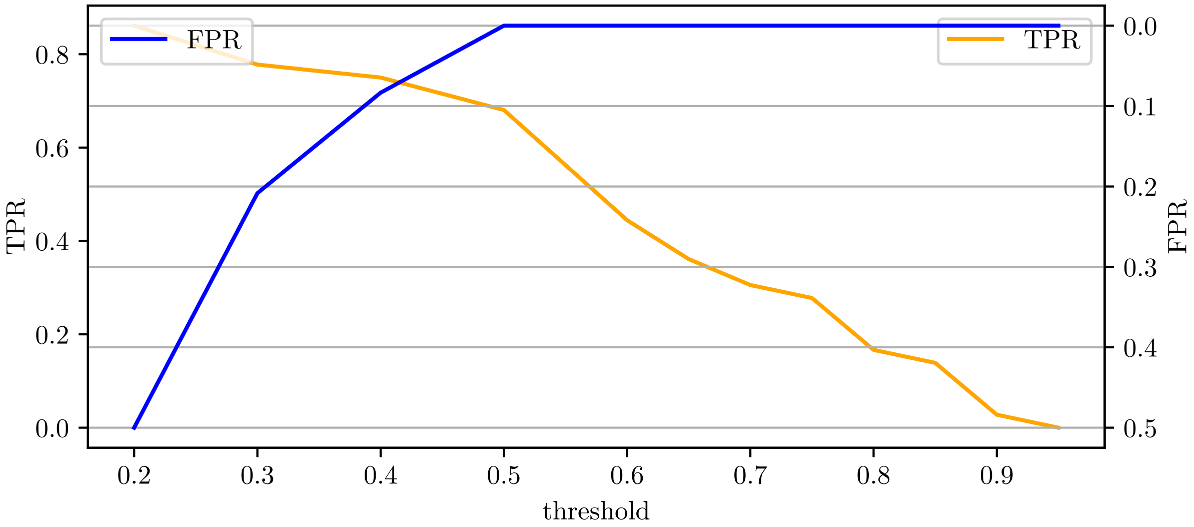


Figure 4.13: TPR and FPR of DTW k-means

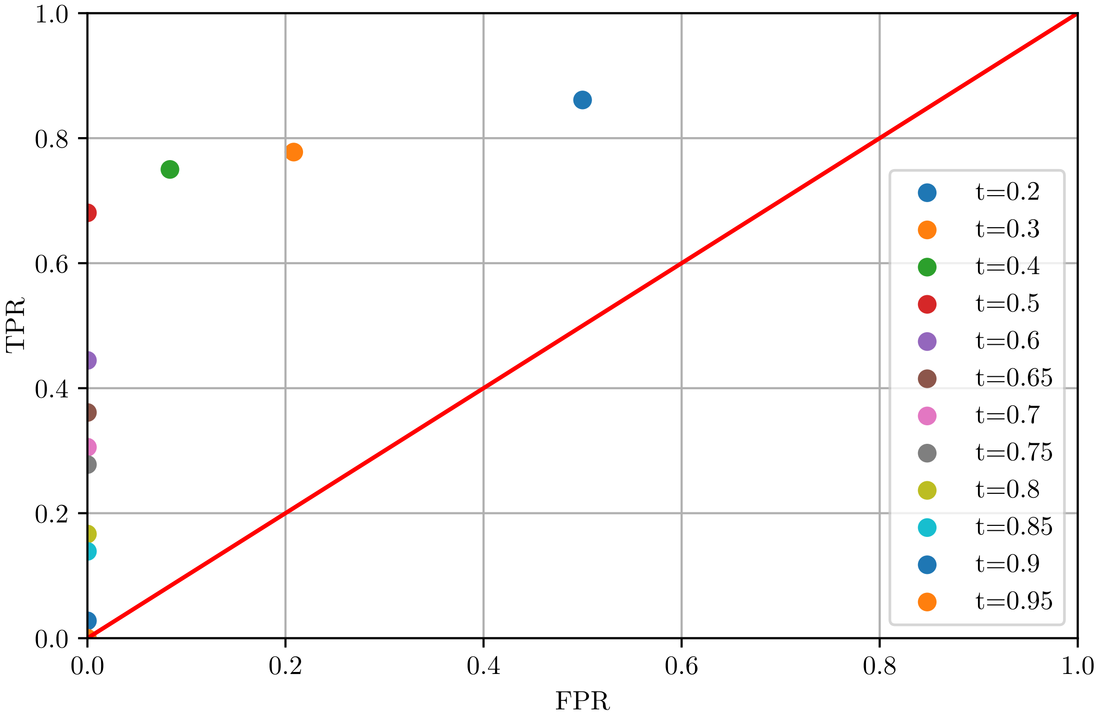


Figure 4.14: ROC curve of DTW k-means

Chapter 5

Results

In this chapter, the results of the different algorithms for ORP prediction, failure detection and failure identification are presented. The described algorithms of chapter 3 with the hyperparameters listed in Chapter 4 are executed and compared. Various accuracy metrics as well as the computation times are given for this purpose.

5.1 Prediction

5.1.1 Accuracy

To evaluate the accuracy of the ORP prediction approaches, R^2 score, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are computed with 10-fold Cross Validation (CV). Therefore, 90% of the shuffled data consisting of 130000 datapoints is used for training the model and 10% is used for testing. The results shown in Figure 5.1, 5.2 and 5.3 are based on the mean of ten different sets of unseen test data containing 13000 datapoints measured every 30 seconds, hence about 4.5 days of data.

The R^2 score defines the proportion of the variance in the dependent variable that can be explained by the model. It is calculated by the sum of squared prediction errors divided by the sum of the squared deviations of each value y_i to the mean:

$$R^2 = 1 - \frac{\sum_{i=0}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{m}(y_i - \overline{y}_i)^2} \tag{5.1}$$

where $\mathbf{y}$ denotes the actual, $\bar{\mathbf{y}}$ the mean and $\hat{\mathbf{y}}$ the predicted ORP. The closer R^2 is to 1, the closer the predicted value is to the actual value, and thus the higher the accuracy of the model. [KJ13]

As Figure 5.1 shows, the R^2 score is highest for the LSTM model. Its median, depicted by the green line, is about 0.05 higher compared to ANN and GRU. Furthermore, it is noticeable that when the outlier shown by the circle, is not taken into account, maximum and minimum values are almost the same for all models. The R^2 values lying 25% above and below the median, represented by the blue box, have a higher variance for GRU and LSTM.

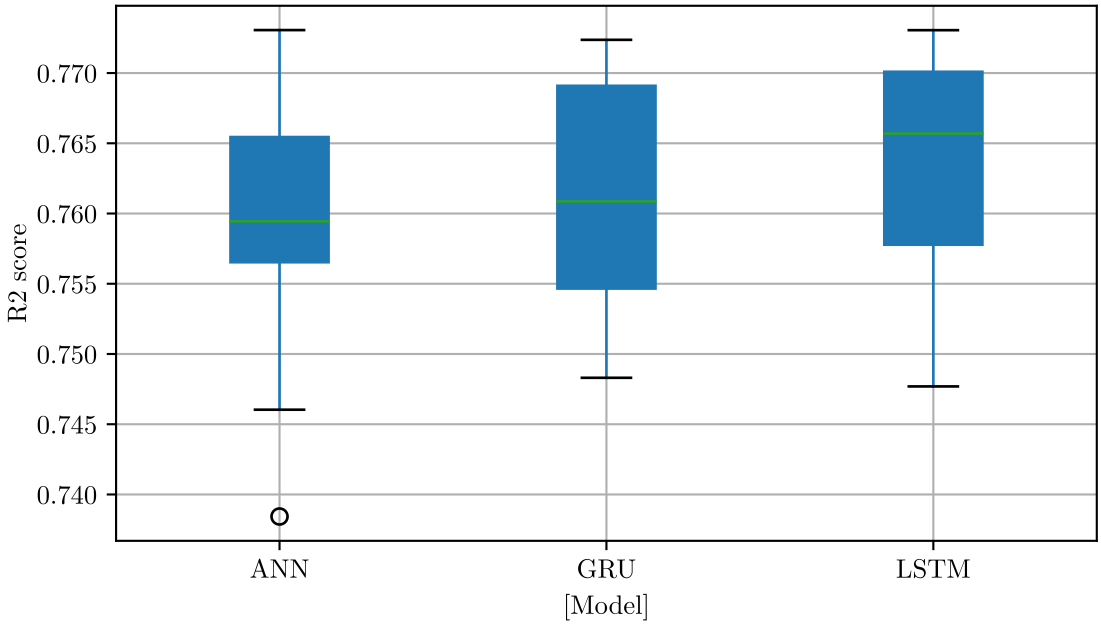


Figure 5.1: R^2 score for ANN, LSTM and GRU

As the name implies, RMSE is determined by the square root of the sum of the squared prediction errors divided by the number of samples [KJ13]:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=0}^{m}(y_i - \hat{y}_i)^2} \tag{5.2}$$

From Figure 5.2 it becomes clear, that again LSTM performs slightly better compared to the other two algorithms. This time, the spread of the 50% around the median is smallest for the LSTM model.

The MAE is similar to the RMSE, but instead of the squared error, the absolute error is calculated [dMGLGR16]:

$$MAE = \frac{1}{m}\sum_{i=0}^{m}|y_i - \hat{y}_i| \tag{5.3}$$

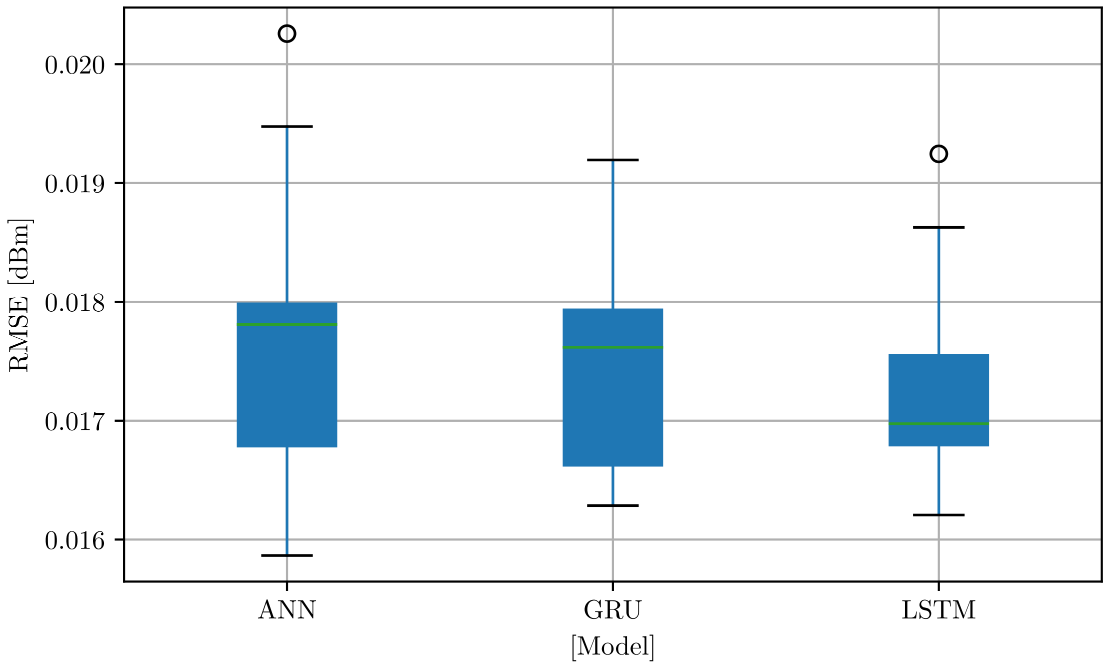


Figure 5.2: RMSE for ANN, LSTM and GRU

Figure 5.3 shows, that the LSTM model has the smallest MAE. The GRU outperforms the fully connected ANN by about 0.005 dBm, while GRU has the widest distribution of the 50% lying around its median.

In summary, LSTM outperforms the other two neural networks slightly for all three accuracy metrics. The GRU and the fully connected ANN are very similar in context of R^2 score and RMSE, but GRU has a slightly lower MAE.
Due to the fact that the ARIMAX model has very a high computation time for large datasets, which is shown in the next section, it can not be compared with the described dataset. Nevertheless, a piecewise comparison using five different datasets each of 1000 points, has revealed 17, 15 and 14 % higher RMSE values than ANN, LSTM and GRU, respectively. Furthermore, the MAE of ARIMAX is 16, 14 and 13% higher compared to ANN, LSTM and GRU, respectively.

5.1.2 Computation time

Predicting one datapoint takes on average 0.30, 3.08, 3.2 and 0.62 seconds for ANN, LSTM, GRU and ARIMAX, respectively. Figure 5.5 shows, how the calculation time for each model increases with increasing amounts of points to predict. Note that because the

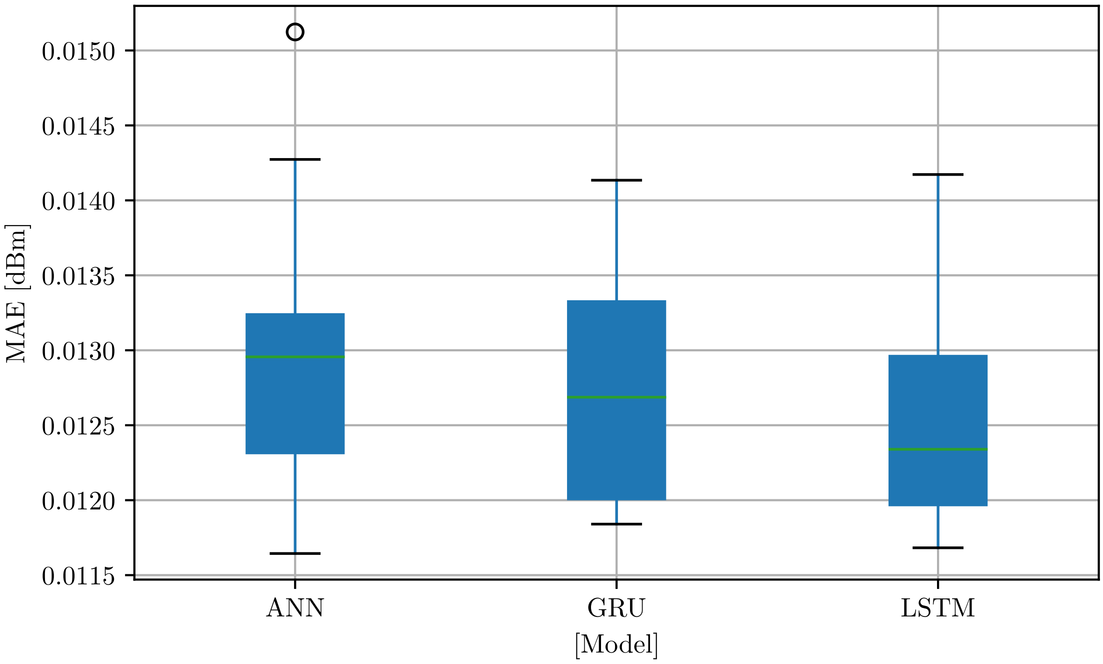


Figure 5.3: MAE for ANN, LSTM and GRU

calculation time for ARIMAX increases significantly faster than for the other models, it has a separate y-axis which is depicted in minutes while the other models times are shown in seconds. It becomes clear, that for GRU and LSTM, the computation time increases if the amount of predictions increases while the computation of ANN stays constant until 5000 predictions. The computation time of ANN for 50000 datapoints takes about 3.8 seconds, so the impact of data size is noticeable too, but much later.

5.2 Failure Detection

5.2.1 Accuracy

In terms of failure detection, the ANN model in combination with the dynamic threshold is compared to OCSVM. The ANN model is used for the prediction of the ORP because it is very fast in the calculation of many data points and has a sufficiently high accuracy. To calculate the detection accuracies, all six originally produced failures from Section 2.3.3 as well as 13000 unseen faultless datapoints are used.
Table 5.1 presents the detection rates for each failure and the faultless data. For failure 1, which is power degradation, only three out of six failures are detected by OCSVM. This is

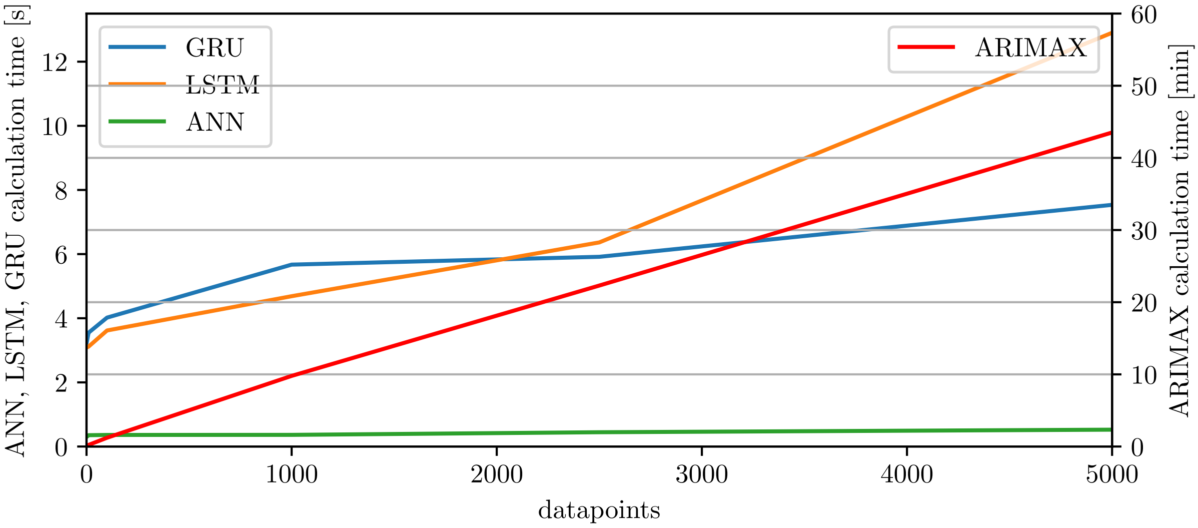


Figure 5.4: Calculation time ARIMAX, ANN, LSTM and GRU

because a correctly classified failure in this context is defined as a switch from detecting no failure to detecting one. For three measurements of the power degradation, the ORP and pre-FEC BER values before and after the failures occurred are already misclassified as failures by OCSVM and hence no failure is detected in that moment. All other failures are correctly detected by both algorithms. Within four and a half days of faultless data, two datapoints are misclassified as failures indicated by an ORP prediction error lying above the dynamic threshold. For the detection with OCSVM, the amount of misclassified datapoints is 112 and therefore much higher.

This is, because as analysed in Chapter 4, the trend of the ORP changes significantly within short periods of time. Due to the fact that the ANN approach detects anomalies based on the directly preceding data points, it can adapt to unseen data ranges. This is not the case for the OCSVM algorithm, as it detects all unseen data ranges as outliers. Hence, if the trend of the underlying data changes a lot, as it does for real field ORP data, OCSVM is not the best choice for anomaly detection.

	ANN with threshold	OCSVM
Failure 1	100%	50%
Failure 2	100%	100%
Failure 3	100%	100%
No Failure	99.98936%	99.44681%

Table 5.1: Failure detection rate for ANN with dynamic threshold and OCSVM

5.2.2 Time of Detection

Table 5.2 shows the time, at which the failures are detected with the two detection approaches. As each type of Failure 1 (power degradation) has a different gradient, the time difference between when it was triggered and when it was detected cannot be directly compared. A steeper slope results in a faster detection but also faster in a service break. Hence, it has to be analysed in combination with the degradation and put into relation with the time of service break, denoted as "End" in the Table. Figure 5.5 shows the total duration of each failure in comparison with the time until it is detected by each algorithm.

Failure	Degr. [dBm/min]	Start	End	ANN+threshold	OCSVM
1	0.2	14:20:00	14:49:00	14:30:00	14:22:30
1	0.5	08:02:00	08:14:00	08:05:30	08:05:30
1	1	08:32:00	08:38:00	08:35:00	08:34:00
1	0.2	13:04:00	13:25:00	13:14:30	-
1	0.5	14:25:00	14:34:00	14:29:00	-
1	1	15:00:00	15:04:00	15:02:00	-
2	-	10:06:00	10:47:30	10:32:30	10:19:00
2	-	11:14:00	11:43:30	11:31:00	11:19:30
3	-	-	22:00:00	22:00:00	22:00:00

Table 5.2: Time of failure detection for ANN with dynamic threshold and OCSVM

It is clear that OCSVM outperforms the ANN approach timewise. For all failures that are detected by OCSVM, the detection happens earlier or at the same time as the detection with ANN. The mean of the difference between failure start to detection divided by the total failure duration is 24.29% for OCSVM and 47.54% for the ANN approach. Hence, the ANN approach performs sufficiently good as well, detecting the failure almost always within the first half of the total failure duration.

The computation time is 0.0057 seconds for the detection with OCSVM and 0.7262 seconds for the detection with ANN including the dynamic threshold calculation.

5.3 Failure Identification

5.3.1 Accuracy

The accuracy of the two different failure identification algorithms is measured on four failure types: power degradation, inter-channel interference, power drop and no failure. Each of the failure types contain six samples. For the failure types 1 and 4 (power degradation and no failure), all six samples are from real monitored data. Failure type 2 (inter-channel interference) contains two actual monitored samples and failure type 3 (power drop) one

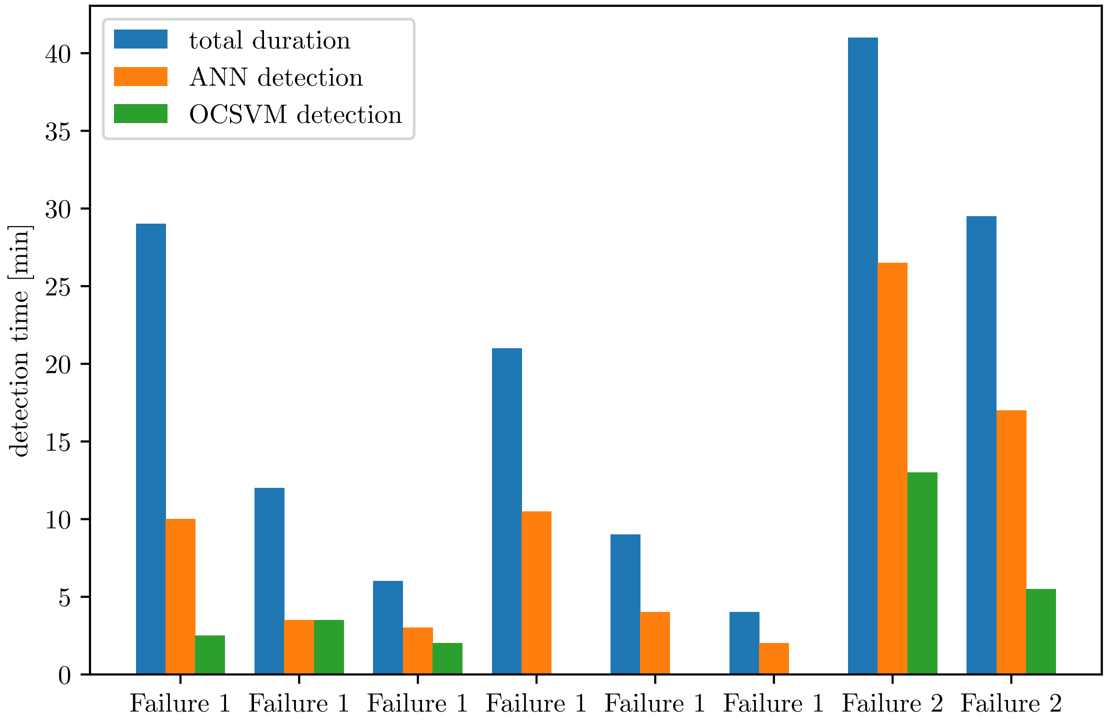


Figure 5.5: Detection time OCSVM and ANNwith threshold

sample. The remaining samples (four for failure 2 and five for failure 3) are produced with data augmentation, compare Section 4.2.4.
To compute the accuracy of the two methods of failure identification, two different ways of testing are used. In the first one, the identification model is built on all known failures types except for one. The left out failure is then used as an unknown failure type which should be identified as such. This is repeated for all four failure classes. The results in Tables 5.3 and 5.4 confirm the conclusions of Section 4.4. While Naive Bayes has a very high TPR, it often misclassifies unknown failures as known ones. DTW k-means on the other hand, has a lower TPR but therefore also has less misclassified unknown failures.

Identified \Actual	Known Failure	Unknown Failure	**Sum**
Known Failure	**68**	14	83
Unknown Failure	4	**10**	13
Sum	72	24	96

Table 5.3: Summed confusion matrix for Naive Bayes

In the second test, the model is built on all failure types, but only on five of six samples for each failure. The left out sample is then used for testing. This is repeated for all samples of each failure, hence six times. The results for the two models are shown in Tables 5.5

Identified \Actual	Known Failure	Unknown Failure	Sum
Known Failure	**55**	2	56
Unknown Failure	17	**22**	40
Sum	72	24	96

Table 5.4: Summed confusion matrix for DTW k-means

and 5.6. From those Tables it might seem, that the Naive Bayes classifier is more accurate, because it counts 17 TP while DTW k-means only has 14, but it has to be noticed that there are no unknown failures classified within this test.

This can be explained by the fact that supervised classification methods outperform unsupervised algorithms if only known classes are to be classified [GMR$^+$11, MMZT15]. From the results of this work we see, that if also unknown classes shall be identified, the unsupervised method DTW k-means is preferable. Moreover, only six samples of each failure are known to the algorithm. Increasing the training data would most likely increase the accuracy of both identification algorithms.

Identified \Actual	Failure 1	Failure 2	Failure 3	No Failure
Failure 1	6	0	0	0
Failure 2	0	2	0	0
Failure 3	0	0	6	0
No Failure	0	0	0	3
Unknown	0	4	0	3

Table 5.5: Confusion matrix, Naive Bayes

Identified \Actual	Failure 1	Failure 2	Failure 3	No Failure
Failure 1	6	0	0	0
Failure 2	0	2	0	0
Failure 3	0	2	4	0
No Failure	0	0	0	2
Unknown	0	2	2	4

Table 5.6: Confusion matrix, DTW k-means

5.3.2 Computation Time

The time needed to identify a failure is 0.0361 seconds for Naive Bayes and 1.1787 seconds for DTW k-means. Hence, the Naive Bayes method is a lot faster and more accurate for predicting known failures, but the DTW k-means method is also relatively fast and significantly better when also identifying unknown failures.

Chapter 6

Conclusions and Outlook

In this work, several EON fault detection and identification algorithms were compared and evaluated based only on the OPM data available at the end user's transceiver. Overall, eight algorithms were implemented and adapted to the OPM data from a 1792 km live production network running for 45 days. Furthermore, three different types of failures were recreated to evaluate the models.

For the failure detection, ORP values were predicted with three different neural network architectures as well as with the statistical approach ARIMAX. The training data for the prediction models did not contain any failures. Based on the resulting prediction error, a dynamic threshold was calculated to decide whether the given timestep is considered as a failure or not. Furthermore, as a second approach for failure detection, an OCSVM was built based on faultless data. The evaluation showed, that of all considered ORP prediction algorithms, the LSTM network has the highest accuracy, while the fully connected ANN has the shortest calculation time. When comparing the ANN approach with the OCSVM, it becomes clear, that failures are detected earlier with OCSVM, but with a higher accuracy using the ANN approach. While the ANN approach has a 100% and 99.989% accuracy rate for failures and faultless data, respectively, the OCSVM only detected 6 out of 9 failures. In addition, significantly more faultless data was misclassified, resulting in an accuracy rate of 99.4468% for faultless data. Both low performances can be explained by the fact that the OCSVM is very much adapted to the training data. The algorithm is not capable of adapting to new ORP ranges and trends, which is the case for the real field ORP data. Although literature [SYW^{+}20] has shown, that OCSVM shows very high performance when using synthetic data, this is not completely applicable to the underlying live production network data, since ORP ranges can vary strongly.

It becomes clear that ORP prediction combined with dynamic thresholding is much better suited to the problem, since the neural network predictions are computed from the ORP and pre-FEC BER values of the timesteps immediately preceding the prediction. Furthermore, the dynamic threshold ensures, that even if the prediction is slightly inaccurate, it is not directly detected as a failure, as only abnormally high prediction errors are detected.

Nevertheless, as external conditions, that are not tracked by the OPM data, might change over time, it is recommended to retrain the prediction ANN as soon as the standard deviation of one day exceeds 1 dBm. To overcome problems with false failure detection, one suggestion would be to define a detected failure as such only after two consecutive failures have been detected. This, on the other hand, would delay the time of detection by 30 seconds (one timestep).

For the failure identification, two classification models have been implemented and adapted such that they are capable of identifying both known and unknown failures. For both algorithms, Naive Bayes classifier and DTW k-means, an ideal probability threshold was found, which defines the decision border between known and unknown failures. For the Naive Bayes, the probability is a direct output of the classification method, and for DTW k-means, the probability is calculated from the distances of the failure to be identified to the cluster centers of each failures class. The evaluation revealed that DTW k-means has significantly higher accuracy than the Naive Bayes classifier when considering known failures that are assigned to the correct known class and unknown failures that are also classified as such. TPR and FPR are 0.76 and 0.06, respectively. The fact that the metrics are not as good as in the literature [LFL$^+$20, SYW$^+$20] can be explained by three reasons. First, to create the faultless data, a live production network was used and hence it is not as stable and ideal as in a simulated network. Second, only OPM data available at the end user transceiver was used and hence only information captured by the OPM parameters can influence the identification. Third, the biggest difference of the approach in this thesis to existing ones is the capability of also identifying unknown failures. Since the method proposed in this work covers the whole range of possible errors, it has a weaker performance than other methods, where by delimiting the range of input failures, all failures that would lead to worse accuracy are already sorted out.

Nevertheless, the approach, which identifies both known and unknown failures, has been shown to work with acceptable accuracy, even when using only the information contained in the OPM data available to OSaaS users. Furthermore, it was demonstrated, that it can distinguish between certain failures without having any knowledge about the OLS.

To conclude, this work has shown, that it is possible to detect and identify soft failures in an early stage only based on the OPM data at the transceivers end. Nevertheless, for an earlier detection and a more precise identification, more knowledge about the network is needed. Furthermore, the proposed failure identification only works, if the failures are separable from each other by ORP and pre-FEC BER.

Future work could include carrying on both, failure detection with an ANN and dynamic thresholing as well as the failure identification, further with additional input parameters. Especially the new way of identifying known and unknown failure could benefit significantly from further information. Hence failures, which are not separable from ORP and pre-FEC BER could potentially be identified with additional input features.

Furthermore, the the implemented models could be tested on other links and networks to see whether the approach is transferable and can be generalized. The aim would be to find out whether the information stored in the models from the training process is enough to detect and identify failures on links different to the training link. When comparing the accuracies of the detection algorithms on new links, the ANN approach will most likely have higher performance than the OCSVM approach. This is due to the better adaptability to new data of the ANN and the dynamic threshold.

Acronyms

AIC Akaike Information Criterion. 48, 49

ANN Artificial Neural Network. 4, 6, 20, 21, 23, 25, 28, 29, 30, 31, 36, 50, 51, 59, 60, 61, 62, 64, 67, 68

ARIMA Autoregressive Integrated Moving Average. 26, 27, 28

ARIMAX Autoregressive Integrated Moving Average with Exogenous Variables. 3, 4, 5, 8, 23, 25, 26, 28, 31, 36, 48, 49, 61, 62, 67

BER Bit Error Rate. 7, 16, 20, 21

CD Chromatic Dispersion. 15

CDC Chromatic Dispersion Compensation. 15, 38, 41

CFO Carrier Frequency Offset. 15

CNN Convolutional Neural Network. 20

CSP Communication Service Provider. 7

CV Cross Validation. 59

DBSCAN Density-Based Spatial Clustering of Applications with Noise. 20, 21

DGD Differential Group Delay. 16

DTW Dynamic Time Warping. 3, 4, 5, 6, 24, 31, 33, 34, 35, 36, 56, 64, 65, 66, 68

EDFA Erbium Doped Fiber Amplifier. 12, 14

EON Elastic Optical Network. 3, 4, 5, 7, 8, 10, 12, 13, 19, 20, 22, 44, 67

EWMA Exponentially-Weighted Moving Average. 31

FEC Forward Error Correction. 16

FN False Negative. 54

FNR False Negative Rate. 21

FP False Positive. 54

FPR False Positive Rate. 3, 4, 21, 54, 55, 56, 55, 56, 68

GRU Gated Recurrent Unit. 3, 4, 28, 29, 51, 59, 60, 61, 62

HP hyperparameter. 48, 49, 51, 53

ITU International Telecommunication Union. 13

kNN k Nearest Neighbours. 20

LAN Local-Area Networks. 12

LC Left Neighbouring Channel. 18

LSTM Long Short Term Memory. 3, 4, 28, 29, 51, 59, 60, 61, 62, 67

MAE Mean Absolute Error. 59, 60, 61

MIC Maximal Information Coefficient. 38, 40, 41

ML Machine Learning. 3, 4, 5, 8, 10, 11, 13, 14, 17, 19, 20, 21, 22, 23, 28, 36, 38, 45, 47

NaN Not a Number. 38

OCSVM One-Class Support Vector Machine. 3, 4, 5, 6, 8, 11, 23, 25, 26, 36, 47, 51, 53, 62, 64, 67, 68

OLS Open Line System. 7, 68

ON Optical Network. 7, 10, 12, 20, 21, 22, 23

OPM Optical Performance Monitoring. 3, 4, 7, 8, 11, 13, 14, 20, 21, 22, 23, 25, 26, 28, 32, 36, 44, 45, 48, 67, 68

ORP Optical Received Power. 3, 4, 8, 11, 16, 18, 19, 21, 23, 25, 28, 31, 37, 38, 40, 42, 44, 45, 47, 48, 49, 56, 59, 62, 67, 68

OSaaS Optical Spectrum as a Service. 3, 4, 7, 14, 22, 68

OSNR Optical Signal to Noise Ratio. 17, 21, 38, 40, 47

PDL Polarization Dependent Loss. 17

pre-FEC BER Pre-Forward Error Correction Bit Error Rate. 3, 4, 16, 18, 19, 21, 38, 40, 47, 56, 62, 67, 68

PSD Power Spectrum Density. 20

QoT Quality of Transmission. 20

RBF Radial Basis Function. 51, 53

RC Right Neighbouring Channel. 18

ReLU Rectified Linear Unit. 30, 48, 51

RF Random Forest. 20, 21

RMSE Root Mean Square Error. 59, 60, 61

RMSprop Root Mean Square propagation. 30

ROADM Reconfigurable Optical Add-Drop Multiplexer. 12, 14, 17

ROC Receiver Operating Characteristic. 55

SDN Software Defined Network. 21

SGD Stochastic Gradient Descent. 30

SNR Signal to Noise Ratio. 17, 20, 38, 40, 47

SVM Support Vector Machine. 20, 21, 25, 26

TN True Negative. 54

TP True Positive. 54, 65

TPR True Positive Rate. 3, 4, 53, 55, 56, 55, 56, 64, 68

WAN Wide-Area Networks. 12

WDM Wavelength Division Multiplexing. 12

WSS Wavelength Selective Switch. 12

Bibliography

[AI20] S. T. Aarthy and J. L. Mazher Iqbal. Time series real time naive bayes electrocardiogram signal classification for efficient disease prediction using fuzzy rules. *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[AT18] S. Aladin and C. Tremblay. Cognitive tool for estimating the qot of new lightpaths. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[ATAT20] Sandra Aladin, Anh Vu Stephan Tran, Stéphanie Allogba, and Christine Tremblay. Quality of transmission estimation and short-term performance forecast of lightpaths. *J. Lightwave Technol.*, 38(10):2807–2814, May 2020.

[BB12] James Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305, 03 2012.

[BM14] Abdenour Bounsiar and Michael G. Madden. Kernels for one-class support vector machines. In *2014 International Conference on Information Science Applications (ICISA)*, pages 1–4, 2014.

[Bra19] Mohammad Braei. *Master thesis*. PhD thesis, 12 2019.

[CGCB14] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[Cho21a] F. Chollet. Keras. `https://github.com/fchollet/keras`, accessed May 24, 2021.

[Cho21b] F. Chollet. Keras layer activation functions. `https://keras.io/api/layers/activations`, accessed May 24, 2021.

[Cip20] Tomas Cipra. *Seasonality and Periodicity*, pages 87–112. Springer International Publishing, Cham, 2020.

[Cla05] S Clavenna. Roadms and the future of metro optical networks. *Heavy Read*, 3:1–5, 2005.

[CLP+19] X. Chen, B. Li, R. Proietti, Z. Zhu, and S. J. B. Yoo. Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks. *Journal of Lightwave Technology*, 37(7):1742–1749, 2019.

[CvMG+14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[deL92] J. deLeeuw. *Introduction to Akaike (1973) Information Theory and an Extension of the Maximum Likelihood Principle*, pages 599–609. Springer New York, New York, NY, 1992.

[dMGLGR16] Arnaud de Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, Jun 2016.

[GFAS14] Klaus Grobe, Cornelius Fürst, Achim Autenrieth, and Thomas Szyrkowiec. Flexible spectrum-as-a-service. In *TERENA Networking Conference (TNC)*, May 2014.

[GMR+11] Luis Guerra, Laura M. McGarry, Víctor Robles, Concha Bielza, Pedro Larrañaga, and Rafael Yuste. Comparison between supervised and unsupervised classifications of neuronal cell types: A case study. *Developmental Neurobiology*, 71(1):71–82, 2011.

[Gé19] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems, second edition.* O'Reilly, 2019.

[HA18] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice.* OTexts, 2018.

[HCL+18] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. pages 387–395, 07 2018.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[Jos20] Ameet V. Joshi. *Support Vector Machines*, pages 65–71. Springer International Publishing, Cham, 2020.

[JTK+09] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *IEEE Communications Magazine*, 47(11):66 – 73, 2009.

[KJ13] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling.* Springer, 2013.

[KRB+20] K. Kaeval, D. Rafique, K. Blawat, K. Grobe, H. Grießer, J. Elbers, P. Rydlichowski, A. Binczewski, and M. Tikas. Exploring channel probing to determine coherent optical transponder configurations in a long-haul network. In *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, 2020.

[LCT00] S. Lewis, S Chernikov, and James Taylor. Temperature-dependent gain and noise in fiber raman amplifiers. *Optics letters*, 24:1823–5, 01 2000.

[LFL+20] H. Lun, M. Fu, X. Liu, Y. Wu, L. Yi, W. Hu, and Q. Zhuge. Soft failure identification for long-haul optical communication systems based on one-dimensional convolutional neural network. *Journal of Lightwave Technology*, 38(11):2992–2999, 2020.

[Mit97] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.

[MMZT15] M. Mohammady, H. R. Moradi, H. Zeinivand, and A. J. A. M. Temme. A comparison of supervised, unsupervised and synthetic land use classification methods in the north of iran. *International Journal of Environmental Science and Technology*, 12(5):1515–1526, 2015.

[MP18] R. M. Morais and J. Pedro. Machine learning models for estimating quality of transmission in dwdm networks. *IEEE/OSA Journal of Optical Communications and Networking*, 10(10):D84–D99, 2018.

[MRJP87] R. Mears, L. Reekie, I. Jauncey, and David Payne. High-gain rare-earth-doped fiber amplifier at 1.54 m. *[No source information available]*, 01 1987.

[MRT18] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning, second edition*. Adaptive Computation and Machine Learning series. MIT Press, 2018.

[MVSA15] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. 04 2015.

[NUW+21] C. Natalino, A. Udalcovs, L. Wosinska, O. Ozolins, and M. Furdek. Spectrum anomaly detection for optical network monitoring using deep unsupervised learning. *IEEE Communications Letters*, pages 1–1, 2021.

[Pea08] *Correlation Coefficient*, pages 115–119. Springer New York, New York, NY, 2008.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay.

Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[RBGT18] C. Rottondi, L. Barletta, A. Giusti, and M. Tornatore. Machine-learning method for quality of transmission prediction of unestablished lightpaths. *IEEE/OSA Journal of Optical Communications and Networking*, 10(2):A286–A297, 2018.

[RRF$^+$11] David Reshef, Yakir Reshef, Hilary Finucane, Sharon Grossman, Gilean McVean, Peter Turnbaugh, Eric Lander, Michael Mitzenmacher, and Pardis Sabeti. Detecting novel associations in large data sets. *Science (New York, N.Y.)*, 334:1518–24, 12 2011.

[RSAE18] D. Rafique, T. Szyrkowiec, A. Autenrieth, and J. Elbers. Analytics-driven fault discovery and diagnosis for cognitive root cause analysis. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[Sch15] Patrick Schäfer. Scalable time series classification. *Data Mining and Knowledge Discovery*, pages 1 – 26, 2015.

[SHK$^+$14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[SMCT18] S. Shahkarami, F. Musumeci, F. Cugini, and M. Tornatore. Machine-learning-based soft-failure detection and identification in optical networks. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3, 2018.

[SP10] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. *Proceedings of the 9th Python in Science Conference*, 2010, 01 2010.

[Spe08] *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY, 2008.

[SR20] Jane M. Simmons and George N. Rouskas. *Routing and Wavelength (Spectrum) Assignment*, pages 447–484. Springer International Publishing, Cham, 2020.

[SRCV19] B. Shariati, M. Ruiz, J. Comellas, and L. Velasco. Learning from the optical spectrum: Failure detection and identification. *Journal of Lightwave Technology*, 37(2):433–440, 2019.

[SRSB13] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 35, July 2013.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[SYW$^+$20] L. Shu, Z. Yu, Z. Wan, J. Zhang, S. Hu, and K. Xu. Dual-stage soft failure detection and identification for low-margin elastic optical network by exploiting digital spectrum information. *Journal of Lightwave Technology*, 38(9):2669–2679, 2020.

[Ter21] Teraflex. `https://www.adva.com/en/products/open-optical-transport/fsp-3000-open-terminals/teraflex`, accessed March 24, 2021.

[TFV$^+$20] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020.

[VLR$^+$19] S. Varughese, D. Lippiatt, T. Richter, S. Tibuleac, and S. E. Ralph. Identification of soft failures in optical links using low complexity anomaly detection. In *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, 2019.

[VRF$^+$17] A. P. Vela, M. Ruiz, F. Fresi, N. Sambo, F. Cugini, G. Meloni, L. Potì, L. Velasco, and P. Castoldi. Ber degradation detection and failure identification in elastic optical networks. *Journal of Lightwave Technology*, 35(21):4595–4604, 2017.

[WSS$^+$20] Qingsong Wen, Liang Sun, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *CoRR*, abs/2002.12478, 2020.

[WZW$^+$17] Zhilong Wang, Min Zhang, Danshi Wang, Chuang Song, Min Liu, Jin Li, Liqi Lou, and Zhuo Liu. Failure prediction using machine learning and time series in optical network. *Opt. Express*, 25(16):18553–18565, Aug 2017.

[YMH$^+$19] J. Yu, W. Mo, Y. Huang, E. Ip, and D. C. Kilper. Model transfer of qot prediction in optical networks based on artificial neural networks. *IEEE/OSA Journal of Optical Communications and Networking*, 11(10):C48–C57, 2019.