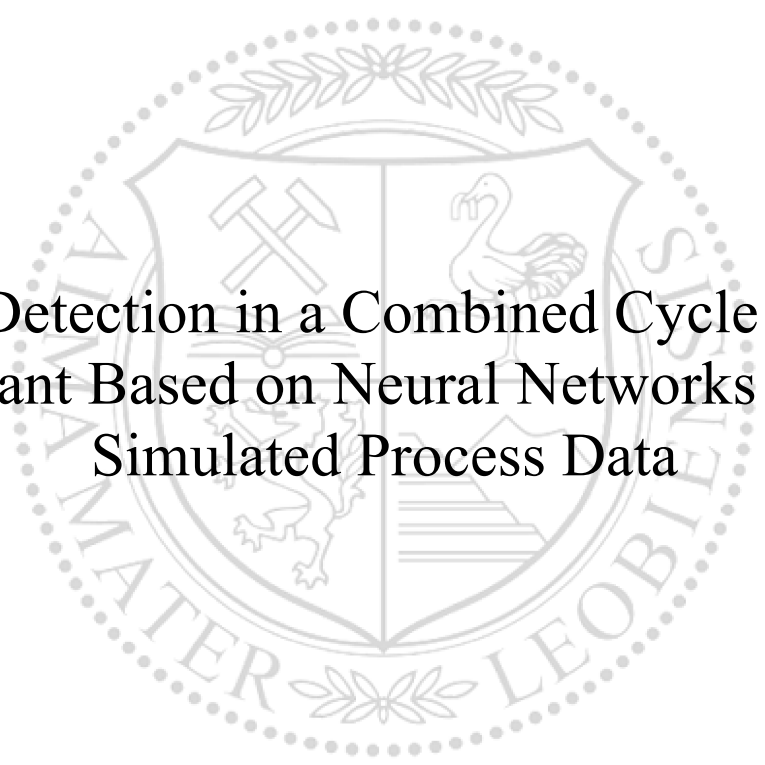




Chair of Energy Network Technology

Master's Thesis



Fault Detection in a Combined Cycle Power
Plant Based on Neural Networks of
Simulated Process Data

Philipp Rohrweck, BSc

February 2021



EIDESSTÄTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 23.02.2021

Unterschrift Verfasser/in
Philipp Rohrweck

ABSTRACT

Despite the progressing decarbonization of our power sources through the increase of renewable energy, conventional power stations like combined cycle power plants (CCPP) still contribute to the power mix by providing highly efficient backup power. To maintain an efficient operation of these plants, the early identification of occurring degradation is essential. In this context, this work deals with a novel approach of automated fault detection carried out by a neural network based on simulated process data.

The initial comprehensive literature research on failure modes in combined cycle power plants and their thermodynamical impact serves as a solid foundation for their realistic simulation. Due to the necessity to generate large amounts of data to constitute every failure mode under various plant operation conditions, an automated workflow of data generation, preparation and validation is introduced. The process of constructing a neural network and enhancing its performance by optimizing the underlying data structure and the networks' hyperparameters are shown. Finally, a statistical evaluation of different network models and their achieved results is conducted.

The networks' ability to detect both, the occurrence of single and multiple failure modes at a time, is evaluated.

It can be shown that the developed neural network is capable of detecting the failure modes with high precision, even when noise is applied to the simulated process data to mimic the scatter of real plant measurements.

KURZFASSUNG

Trotz der fortschreitenden Dekarbonisierung unserer Energiequellen durch den Ausbau erneuerbarer Energien, sind konventionelle Kraftwerke wie Gas-und-Dampf-Kraftwerke ein essenzieller Teil unserer Stromversorgung, da sie beispielsweise hocheffizient Reserveleistung bereitstellen. Um einen effizienten Betrieb dieser Anlagen zu gewährleisten, ist die frühzeitige Erkennung auftretender Degradationen unerlässlich. Aufgrund dessen wird in dieser Arbeit ein neuartiger Ansatz der automatisierten Fehlererkennung durch neuronale Netze auf der Basis von simulierten Prozessdaten behandelt.

Eine umfassende Literaturrecherche zu Fehlermöglichkeiten und den resultierenden thermodynamischen Auswirkungen in Gas-und-Dampf-Kraftwerken dient als Fundament für deren physikalisch korrekte Simulation. Eine enorme Menge an Simulationsdaten wird benötigt, um die einzelnen Fehlermöglichkeit in verschiedensten Anlagenbetriebspunkten darzustellen. Deshalb wird ein automatisierter Arbeitsablauf zur Datenerstellung, -aufbereitung und -validierung eingeführt. Das Erstellen von neuronalen Netzen sowie deren Optimierung durch Verbesserung der zugrundeliegenden Datenstruktur und deren Hyperparameter wird behandelt. Abschließend folgt eine statistische Evaluierung der erzielten Ergebnisse unterschiedlicher neuronaler Netze.

Die trainierten Netze werden auf die Fähigkeit, einzelne wie auch kombiniert auftretende Fehler zu erkennen, geprüft.

Es kann gezeigt werden, dass das entwickelte neuronale Netz in der Lage ist, die Fehler mit hoher Präzision zu erkennen, selbst wenn die simulierten Prozessdaten mit Rauschen überlagert werden, um die Streuung von realen Anlagenmessungen nachzuahmen.

ACKNOWLEDGMENT

My sincere thanks go to all persons, friends and colleagues who supported me in my years of study. Above all, I want to express my deepest gratitude to my girlfriend Juliane.

I am deeply grateful to the competent and nice colleagues of Enexsa who enabled this thesis and allowed me to work on such an interesting topic. I would particularly like to thank Dipl.-Ing. Dr. techn. Peter Pechtl, Dipl.-Ing. Dr. techn. Josef Petek and Dipl.-Ing. Martin Pözl. Without their professional expertise and persistent help this thesis would not have been possible.

I would like to express my deepest appreciation to Univ.-Prof. Dipl.-Ing. Dr. techn. Thomas Kienberger. His professional guidance and very valuable comments steered this work in the right direction.

Finally, I would like to offer my special thanks to my family, especially to my parents Waltraud und Herbert Rohrweck for their unfailing support and encouragement in all life situations.

CONTENTS

Nomenclature	I
List of figures	IV
List of tables	VI
1 Introduction	1
2 Task Assignment	2
2.1 Methodology.....	2
3 Theoretical Background	4
3.1 Combined Cycle Power Plants	4
3.1.1 Gas Turbine	4
3.1.2 Steam Turbine	6
3.1.3 HRSG	7
3.1.4 Combination of Brayton Cycle and Rankine Cycle	9
3.1.5 Failure Modes in CCPP	11
3.2 Artificial Intelligence	18
3.2.1 Neural Networks	18
3.2.2 Statistical Methods for Performance Evaluation	23
4 Model Design	26
4.1 Software	26
4.1.1 EBSILON	26
4.1.2 Python	27
4.1.3 Tensorflow and Keras.....	27
4.2 CCPP EBSILON-Model	27
4.3 Failure Mode Simulation.....	28
4.4 Data Generation.....	36
4.4.1 Data Validation	39
4.4.2 Data Preparation	40

Contents

4.4.3	Measurement Noise Integration.....	40
4.5	Neural Network.....	42
4.5.1	Building of the NN	42
4.5.2	Training of the NN	43
4.5.3	Performance Evaluation Process.....	45
5	Results.....	49
5.1	NN Model.....	49
5.1.1	Optimization	50
5.1.2	Clean Data Model	53
5.1.3	Noisy Data Model	56
5.1.4	Test on Two Active Failure Modes	61
6	Conclusions.....	65
7	Outlook	66
	Bibliography	67
	Appendix	71

NOMENCLATURE

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
CCPP	Combined Cycle Power Plant
CO	Carbon Oxide
CRH	Cold Reheat
CV	Cross-Validation
ES	EarlyStopping function
FI	Failure Impact
FM	Failure Mode
FN	False Negative
FP	False Positive
GT	Gas Turbine
GTCCPP	Gas Turbine Combined Cycle Power Plant
HP	High Pressure
HRH	Hot Reheat
HRSG	Heat Recovery Steam Generator
IDE	Integrated Development Environment
IP	Intermediate Pressure
LP	Low Pressure
LR	Learning Rate

Nomenclature

LRS	Learning Rate Scheduler
MC	ModelCheckpoint function
NH ₃	Ammoniac
NN	Neural Network
NO _x	Nitrogen Oxides
ReLU	Rectified Linear Activation function
ST	Steam Turbine
TN	True Negative
TP	True Positive

Formula Symbols

A	Heat transfer area	Square meter (m ²)
CAN	Nominal free cross-section	Square meter (m ²)
DP12RN	Nominal pressure drop	Bar (bar)
E	Energy	Joule (J)
ETAI	Isentropic efficiency	
ETAPN	Polytropic efficiency	
KAN	Nominal heat transfer factor	Watt per Kelvin (W/K)
M1N	Nominal mass flow	Kilogram per second (kg/s)
P	Power	Watt (W)
PFETA	Performance Factor on Efficiency	
PFFLOW	Performance Factor on Flow	
Q	Heat quantity	Joule (J)
Q̇	Heat flow	Watt (W)
T	Temperature	Degrees Celsius (°C)

Nomenclature

W	Work	Joule (J)
b	Bias of neurons	
c	Confusion matrix entry	
k	Heat transfer coefficient	Watter per square meter times Kelvin (W/m ² K)
\dot{m}	Mass flow	Kilogram per second (kg/s)
p	Pressure	Bar (bar)
w	Specific weight neuron connection	
x	Input value neural network	
y	Output value neural network	
z	Weighted sum	
η	Degree of efficiency	

LIST OF FIGURES

Figure 3-1: Gas turbine schematic, [15] S. 24	5
Figure 3-2: Rankine cycle schematic	6
Figure 3-3: Single pressure HRSG schematic, [15] S. 29	8
Figure 3-4: Triple pressure HRSG schematic, [15] S. 30	8
Figure 3-5: CCPP schematic, [15] S. 23	10
Figure 3-6: TS-diagram for the combination of Brayton and Rankine cycle, [15] S. 36	10
Figure 3-7: HP turbine and IP turbine with sealing package, [20]	13
Figure 3-8: Exhaust loss curve of a steam turbine, [29]	15
Figure 3-9: Condenser for steam power plants, [34]	17
Figure 3-10: 1-2-3-4: Ideal Rankine cycle; 1-1': Condensate subcooling	17
Figure 3-11: Biological neuron model, [37]	19
Figure 3-12: Perceptron: Neural network model consisting of one neuron, [35] S. 374	20
Figure 3-13: Neural network structure, [35] S. 386	21
Figure 3-14: Examples of fittings of a neural network to training data, [39] S. 253	22
Figure 3-15: Confusion matrix for binary classifications	23
Figure 3-16: Confusion matrix for a multi-label classification problem	25
Figure 4-1: Gas turbine expander tip clearance simulation	30
Figure 4-2: Steam turbine HP bowl to IP bowl leakage simulation	31
Figure 4-3: Letdown leakage simulation, shown on the example of the bypass for the LP section of the ST	32
Figure 4-4: Evaporator blowdown leakage simulation	32
Figure 4-5: Condenser incondensable gas accumulation and hotwell subcooling simulation	34
Figure 4-6: Histogram of failure impact distribution for 10,000 generated values	38
Figure 4-7: Effect of condenser incondensable gas accumulation on the power output of the steam turbines	39
Figure 4-8: Effect of the learning rate on the validation loss	44
Figure 4-9: Example of a prediction from the NN with applied threshold	46
Figure 4-10: Influence of the threshold on false predictions of the NN for FM 12	47
Figure 4-11: Influence of the threshold on correct predictions of the NN for FM 12	48
Figure 5-1: Predicted failure modes in 5000 examples for the actual active failure mode 7	50
Figure 5-2: Correct(blue) and False(orange) predictions for every class; Clean data model and clean test data	53
Figure 5-3: Correct(blue) and False(orange) predictions for every class; Clean data model and clean test data with FI > 5%	54

List of figures

Figure 5-4: Correct(blue) and False(orange) predictions for every class; Clean data model and noisy test data	55
Figure 5-5: Predictions for an actual failure mode 0.....	55
Figure 5-6: Correct(blue) and False(orange) predictions for every class; Noisy data model and noisy test data	56
Figure 5-7: Predictions for an actual failure mode 1.....	57
Figure 5-8: Dependence of sensitivity and precision on the threshold value.....	57
Figure 5-9: Number of TP predictions for failure mode 6 as a function of the threshold	58
Figure 5-10: Overall precision without FM 0 for a noise factor of 1.0	59
Figure 5-11: Number of overall TP predictions as a function of the threshold without FM0.....	59
Figure 5-12: Overall precision without FM 0 for a noise factor of 0.5	60
Figure 5-13: Overall precision without FM 0 for a noise factor of 2.0	60
Figure 5-14: Predicted probabilities for the actual active classes FM 21 (FI = 9.73%) and FM 23 (FI = 14.43%)	61
Figure 5-15: Predicted probabilities for the actual active classes FM 21 (FI = 17.91%) and FM 23 (FI = 40.12%)	62
Figure 5-16: Predicted failure modes for the actual active classes FM 21 and FM 23 on clean test data	62
Figure 5-17: Predicted failure modes for the actual active classes FM 21 and FM 23 on noisy test data	63
Figure 5-18: Mean overall probability for every failure mode for actual active classes FM 21 and FM 23 on noisy test data.....	64
Figure Appendix 1: Heatmap for the relative impact (scale up to 2%) of every failure mode on the process parameters	71
Figure Appendix 2: Heatmap for the relative impact (scale up to 15%) of every failure mode on the process parameters	72
Figure Appendix 3: Specification values of false predicted examples for failure mode 21	79
Figure Appendix 4: Predicted failure modes for the actual active classes FM 17 and FM 21 on clean test data	80
Figure Appendix 5: Predicted failure modes for the actual active classes FM 5 and FM 18 on clean test data	80
Figure Appendix 6: Predicted failure modes for the actual active classes FM 9 and FM 22 on clean test data	81

LIST OF TABLES

Table 4-1: Failure modes	35
Table 4-2: Model parameters.....	37
Table 4-3: Measurement tolerances	41
Table 5-1: Specification limits	49
Table 5-2: Hyperparameter test configurations.....	51
Table 5-3: Best hyperparameter combinations	52
Table Appendix 1: Result of clean data model and clean test data	73
Table Appendix 2: Result of clean data model and clean test data (FI>5%)	74
Table Appendix 3: Result of clean data model and noisy test data	75
Table Appendix 4: Result of noisy data model and noisy test data	77

1 INTRODUCTION

The Paris Agreement was, till now, ratified by 189 Parties. Countries worldwide signed the agreement to limit the global temperature increase compared to pre-industrial levels below 2.0 degrees Celsius by reducing greenhouse gas emissions. [1] The decarbonization of our energy system is one of the challenges we face in the upcoming decades to reach this goal. This drives the expansion of renewable energy, such as wind and solar power forward. Energy production from these technologies depends highly on weather and day time, hence requires immense storage capacities and a suitable power grid to compensate for fluctuations in power generation and demand. Energy storages balance demand in time, and the power grid balances demand between the locations of generation and consumption.

Today we only start to build these infrastructures and still need fossil fuels to back up the installed renewable power. In this transition period, it's essential to use highly efficient technologies to emit as little CO₂ as possible. Combined Cycle Power Plants (CCPP) represent one of these bridge technologies. CCPPs provide high electric efficiency and enormous fuel utilization rates. The key for assuring high efficiency over time is perfect maintenance, which directly affects fuel consumption and CO₂ emissions. Therefore, effective use of resources for revision planning and detection of degradation failure modes is necessary. Especially efficiency losses due to degradation are hard to detect because they usually don't cause forced outages of system parts or even the whole power plant. Instead, the impact of these failures gradually increases over time and affects the performance parameters of the power station. The performance and the power output of CCPPs also depend on environmental conditions as ambient temperature and air humidity or cooling water temperature. These interdependencies add even more complexity to the detection of equipment degradation.

Today physical performance monitoring systems are already established in CCPP applications. These systems track important process parameters in plant operation. [2] But they are restricted to observing the power plant and usually have only limited capability of predicting which failure led to a deviation in the operation parameters. Other systems provide the opportunity to predict failures only for single components. They are often based on pattern recognition in vibrations or other measurements. [3] However, none of the mentioned methods to date sufficiently addresses the issue of monitoring equipment performance of a total power plant by deriving present failure modes from available operational measurements. Hence this work deals with a novel approach to prove the potential of automatic failure detection based on simulated data.

2 TASK ASSIGNMENT

This work aims to prove the possibility of automated failure detection in combined cycle power plants using simulated data. The company ENEXSA GmbH uses EBSILON Professional to simulate thermodynamical processes and optimize thermodynamic cycle applications. An existing EBSILON model representing the characteristics of the whole CCPP process serves as the basis for further investigations. Different degradation failure modes (FM) shall be implemented in the model to enable the simulation and analysis of their effect on the power plant. Comprehensive literature research on failure modes ensures the physically accurate modeling of their impact on the affected plant component.

Subsequently, so-called labeled data sets for the different failures shall be created: Each data set represents the response of the power plant to an occurring failure, at various ambient and load conditions.

The variation of the quantity of all simulated failure modes leads to a multi-label classification problem, which shall be solved by utilizing a neural network (NN). The goal is to build and train a NN capable of detecting degradation failures using the before-mentioned simulated data. After optimizing the NN, statistical evaluation of the achieved performance follows.

2.1 Methodology

The first step is comprehensive literature research on failure modes in CCPPs. This research covers all major components of the power plant, such as the gas turbine (GT), the heat recovery steam generator (HRSG), the steam turbines (ST), and the condenser. Besides the systematical categorization of the failure modes, the analysis focuses on the predicted thermodynamical impact. Furthermore, research on the CCPP cycle and the highly interdependent sub-systems (GT, HRSG, ST, condenser) is necessary to gain a more profound knowledge of the underlying processes.

Next, a physically correct model of a CCPP is needed. The EBSILON®Professional software [4] from the Steag Energy Services company is utilized to modify an existing model of a CCPP from ENEXSA. After adopting the CCPP model, the integration of the failure modes follows. The usage of an EBSILON Add-In for Microsoft Excel enables the automated calculation of power plant simulation runs. The specifications for the simulation runs are set in the Excel file. Transmitting these calculation tasks to the ENEXSA servers, using ENEXSA's EbsGrid Distributed Calculation Technology, assures an efficient and fast way to execute a large number of simulations.

The data generation process is further automated by using Python scripts and a Python client, enabling access to servers of ENEXSA. Python scripts simplify data handling and analysis, especially for vast amounts of information. The programming takes place in the PyCharm integrated development environment (IDE) [5] and the web-based JupyterLab IDE [6]. Utilizing these environments and deploying the methods mentioned above, the labeled failure mode data can be generated and preprocessed before training the failure predicting neural network.

The NN is set up and trained with Python scripts by deploying Tensorflow [7] and the Keras application programming interface (API) [8].

Finally, statistical methods will help to judge the performance of the developed NN in its capability of detecting failures.

3 THEORETICAL BACKGROUND

This chapter represents the basic knowledge for the applied methods and empirical studies of this thesis. The physics of the investigated power plants and the theoretical background of neural networks are explained.

3.1 Combined Cycle Power Plants

The term combined cycle power plant could refer to any plant which uses two or more thermodynamic cycles to generate electricity. However, this work deals with CCPPs combining a Brayton cycle (also known as Joule cycle) and a Rankine cycle. These two cycles serve as thermodynamic models for a gas turbine and a steam turbine system. The resulting CCPP is known as a gas turbine combined cycle power plant (GTCCPP). The CCPPs mentioned in this thesis always refer to GTCCPPs. An HRSG acts as the linking structure between the systems. The HRSG utilizes the gas turbine's waste heat to generate high-pressure steam for the steam turbine. [9]

Combining these systems leads to the highest efficiencies of all commercially used power plants. The following list shows a few beneficial characteristics of GTCCPP: [10]

- Efficiencies from 50 % to above 60 %
- Lowest emissions of unburnt hydrocarbons, carbon oxide (CO), and oxides of nitrogen (NOx) of all thermal power plants
- Low construction costs ranging between 450 \$/kW and 650 \$/kW for CCPPs of 200 MW and above
- Construction time of 200 MW and above CCPPs could be less than 24 months

The following sub-chapters explain the basic functionality of CCPPs and their components. For further information on the described processes, the reader is referred to [9], [11], [12] and [13].

3.1.1 Gas Turbine

Today various types of gas turbines are available for stationary or mobile applications. Ships, aircraft, pumping stations, trains, and power plants run on gas turbines. [14] Determined by the field of operation, either aero-derivative turbines, heavy-duty industrial turbines, or other gas turbines are utilized. However, state-of-the-art CCPPs with a power output of 200 MW and above use heavy-duty industrial gas turbines as their core component. Gas turbines in simple cycle operation reach efficiencies of about 40 % and slightly above.

The following explanation describes the main components and basic functionality, as shown in Figure 3-1.

First, air enters the system and the compressor increases the pressure and temperature of the fluid. In the combustion chamber, fuel injection and ignition follow. The added heat increases the fluid temperature at a constant pressure. This high-temperature pressurized gas expands isentropically and powers the turbine (or expander). The majority of gas turbines operate in an open thermodynamic cycle, and therefore the flue gas is afterward emitted and not returned to the beginning of the cycle. The gained mechanical energy runs the compressor and the dedicated gas turbine application. In this case, the shaft transmits the mechanical energy to a synchronous generator to produce electricity.

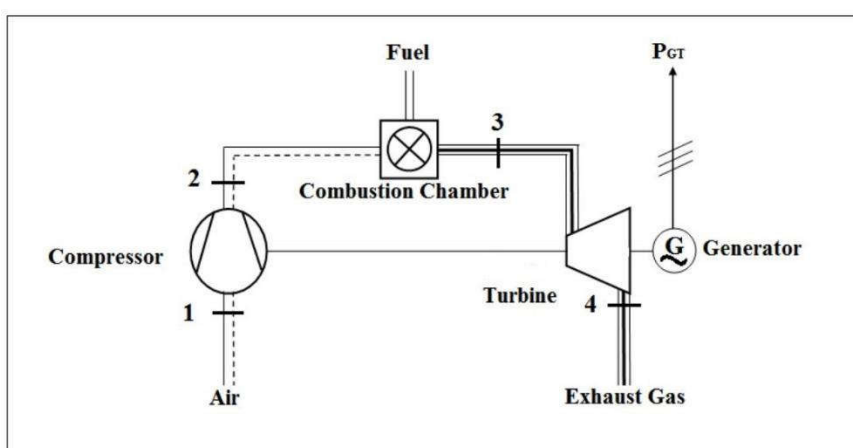


Figure 3-1: Gas turbine schematic, [15] S. 24

The following rating parameters of gas turbines are essential for the application in CCPPs as they influence the power plant's overall efficiency:

- Power in kW or MW
- Heat Rate in kJ/kWh or Btu/kWh
- Electrical Efficiency in kWh/kJ or kWh/Btu
- Exhaust Flow in kg/s or klb/s
- Exhaust Temperature in °C or °F

The power determines the gross electrical power output without auxiliary or step-up transformer losses. The heat rate indicates the amount of fuel needed to generate a defined amount of electric energy. The electrical efficiency is the inversion of the heat rate. And the exhaust flow parameters refer to the flue gas used to fuel the steam cycle of the investigated CCPPs.

The rating of gas turbines is always defined at reference conditions, and the mentioned performance parameters are achieved at this specific operation point. [16] At other

operating conditions, for example, changing ambient temperatures or air pressure, the power output and the other parameters differ. Therefore, the term Base Load is defined to describe if the gas turbine runs at full load at the given ambient conditions. Base Load is the load at which the gas turbine process is operated at its aerodynamic design point with maximum throughput and optimal flow pattern. Since the specific volume of the airflow varies with the conditions at the compressor inlet, the compressor exit pressure and the mass flow through the combustor will change accordingly. In addition, the pressure at the exit of the gas turbine limits the pressure ratio of the expander, and thus the Base Load power generated by the gas turbine is varying significantly with these parameters. Not reaching expected Base Load performance at specified conditions is a sign of degradation of the power plant.

The exhaust flow and especially the exhaust temperature of the gas turbine directly affect the CCPP efficiency. Modern heavy-duty gas turbines apply complex control schemes to optimize overall combined cycle performance through inlet guide vane positioning, firing temperature control, different burner schemes or bypass/blow-off mechanisms.

3.1.2 Steam Turbine

Steam turbines still represent the backbone of the power industry. [17] They are in operation in coal-fired power plants, nuclear power plants and combined cycle power plants, but also in solar thermic and geothermal applications besides others. All these plants operate on the same principle, the Rankine cycle (see Figure 3-2): The feedwater pump delivers water into the boiler and increases the liquid's pressure. In the boiler, a heat source adds energy to the water, evaporates and superheats it. As mentioned above, the heat source can be anything from nuclear fission to solar energy. The superheated steam enters the steam turbine, expands and afterward returns to its liquid state in the condenser. The condensate is again fed to the pump and closes the cycle.

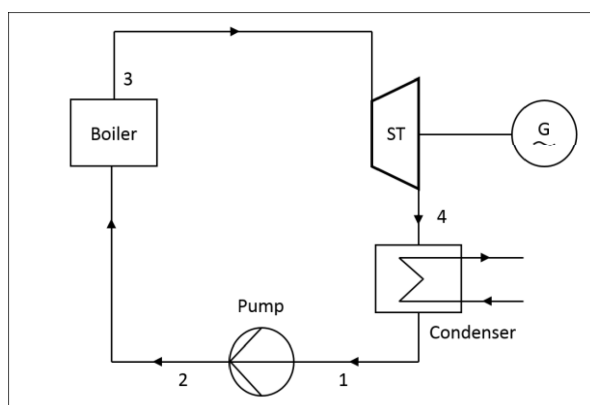


Figure 3-2: Rankine cycle schematic

Carnot's theorem describes the maximum efficiency of heat engines. The hot and cold reservoir temperature, between which the cycle operates, determines the maximal achievable efficiency of the engine (3-1). Therefore, a high mean temperature of heat addition and a low mean temperature of heat rejection lead to a high cycle efficiency. In steam power plants, the mean temperature of heat addition is low compared to other fossil-fuelled power plants like gas turbines. But the condenser assures low heat rejection temperatures as it operates at low pressures. This low turbine back pressure guarantees a high cycle efficiency.

$$\eta_{Carnot} = 1 - \frac{T_C}{T_H} \quad (3-1)$$

The methods to increase the efficiency of power plants that operate on a Rankine cycle differ among the various power plant types. To achieve high efficiency in CCPPs it's important to extract as much heat as technically possible out of the GT flue gas. Therefore, modern CCPPs operate on three pressure levels with corresponding turbine sections and various reheating and preheating cycles. An example of a three pressure process in CCPPs is explained in the following subsection.

3.1.3 HRSG

The heat recovery steam generator acts as the linking structure between the Brayton cycle and the Rankine cycle. The flue gas of the gas turbine surges through the HRSG and passes a series of heat exchangers. The extracted heat fuels the steam cycle of the CCPP. The HRSG can either be constructed vertically or horizontally. The HRSG shown in Figure 3-3 is aligned vertically and operates in counter-current mode, except for the evaporator. Feedwater, coming from the condenser, enters the economizer and is heated up close to boiling temperature. A few degrees of subcooling are beneficial to avoid evaporating in the economizer, which would cause excessive flow velocities in the pipes. The water's physical state changes from liquid to steam in the evaporator section. The liquid flows from the drum through the heat exchanger pipes and evaporates. The drum itself separates the steam from the liquid phase and contains a blowdown system. Impurities that accumulate in the drum can be extracted by blowing water out. In the superheating section, the extracted heat from the flue gas increases the steam temperature, and the superheated steam is led to the turbine.

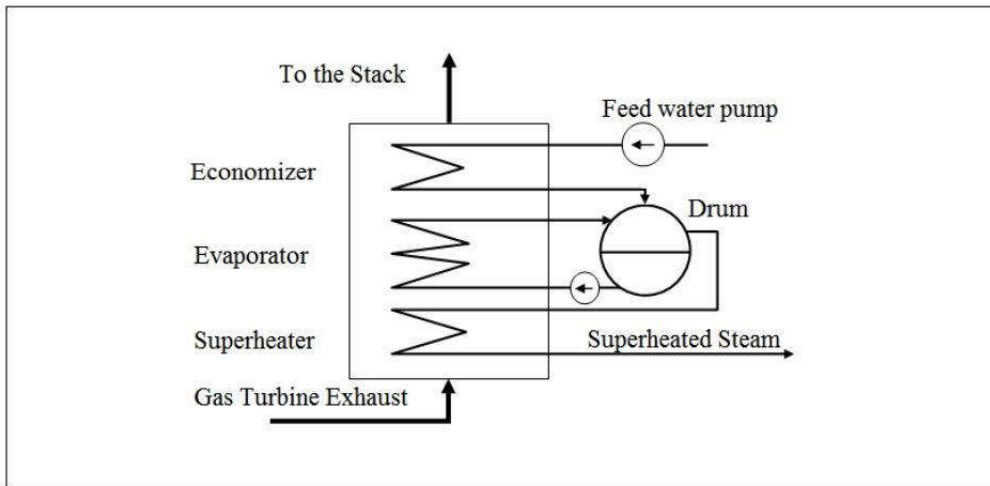


Figure 3-3: Single pressure HRSG schematic, [15] S. 29

CCPPs with a triple pressure HRSG run on the same principle, with an economizer, an evaporator and a superheater for each pressure level. The goal is to extract more heat from the flue gas and increase plant efficiency. For each pressure level (high pressure (HP), intermediate pressure (IP) and low pressure (LP)) a corresponding steam turbine section exists. The schematic diagram in Figure 3-4 shows a simple triple pressure HRSG without any reheat cycles.

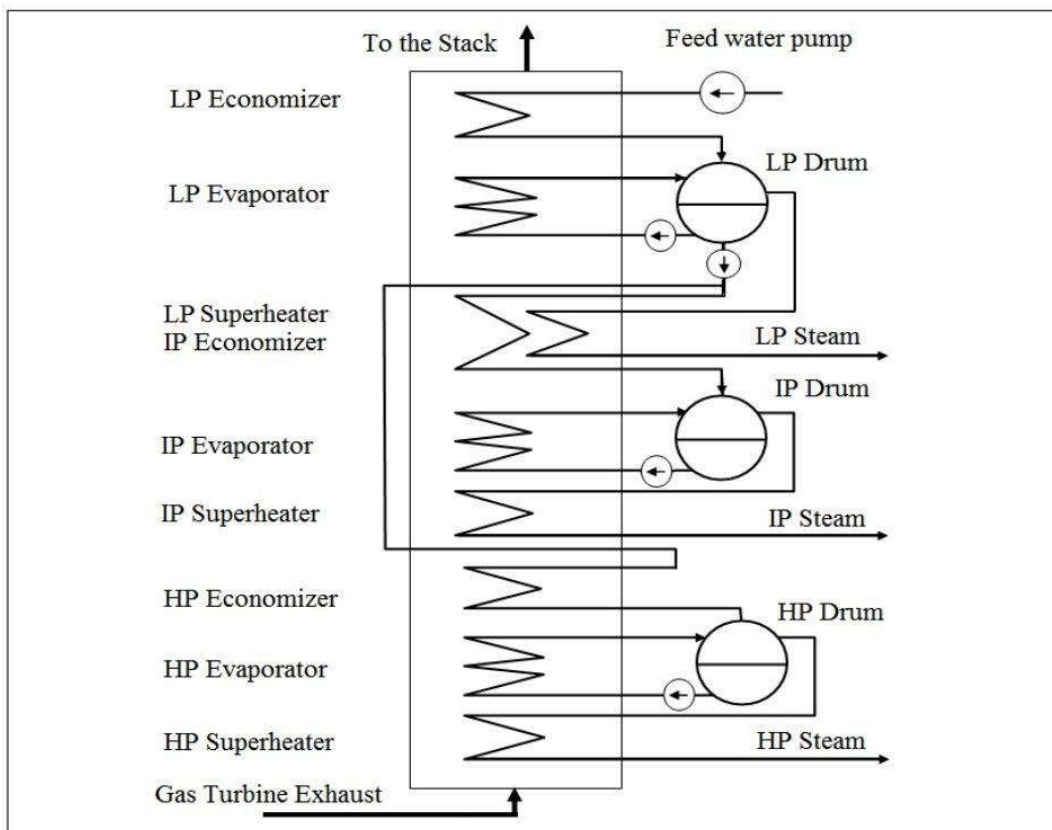


Figure 3-4: Triple pressure HRSG schematic, [15] S. 30

3.1.4 Combination of Brayton Cycle and Rankine Cycle

Gas turbine combined cycle power plants reach field-proven efficiency levels of 60% and above. [9] The reason for this performance is the combination of two thermodynamical cycles that operate at different temperature levels. CCPPs combine the high mean temperature heat addition of gas turbines with the steam turbines' low mean temperature rejection. Referring to their positions in the temperature-entropy (T-S) diagram in Figure 3-6, the Brayton cycle represents the "topping cycle" and the Rankine cycle the "bottoming cycle". The HRSG between them serves as the heat sink for the former and the latter's heat source.

The flue gas temperature and mass flow determine the achievable superheating temperature of the steam cycle. A high flue gas temperature leads to a higher HRSG- and Rankine cycle efficiency but also a lower GT efficiency. Therefore, HRSG- and GT design are critical for optimal design in this trade-off. However, from the TS-diagram in Figure 3-6 a simplified version of the overall plant efficiency (η_{CC}) can be derived (3-7). Q_{GT} symbolizes the gained heat quantity by burning the fuel. W_{GT} and W_{ST} present the amount of work output of the cycles. The HRSG transfers Q_{HRSG} from the exhaust gas to the Rankine cycle, and Q_{LOSS} describes the total heat loss of the process. The HRSG losses are combined in η_{HRSG} .

$$P_{GT} = \frac{dW_{GT}}{dt} \quad (3-2)$$

$$P_{ST} = \frac{dW_{ST}}{dt} \quad (3-3)$$

$$P_{GT} = \dot{Q}_{GT} \cdot \eta_{GT} \quad (3-4)$$

$$P_{ST} = (\dot{Q}_{GT} - P_{GT}) \cdot \eta_{ST} \cdot \eta_{HRSG} \quad (3-5)$$

$$\eta_{CC} = \frac{P_{GT} + P_{ST}}{\dot{Q}_{GT}} \quad (3-6)$$

$$\eta_{CC} = \eta_{GT} + \eta_{ST} \cdot \eta_{HRSG} \cdot (1 - \eta_{GT}) \quad (3-7)$$

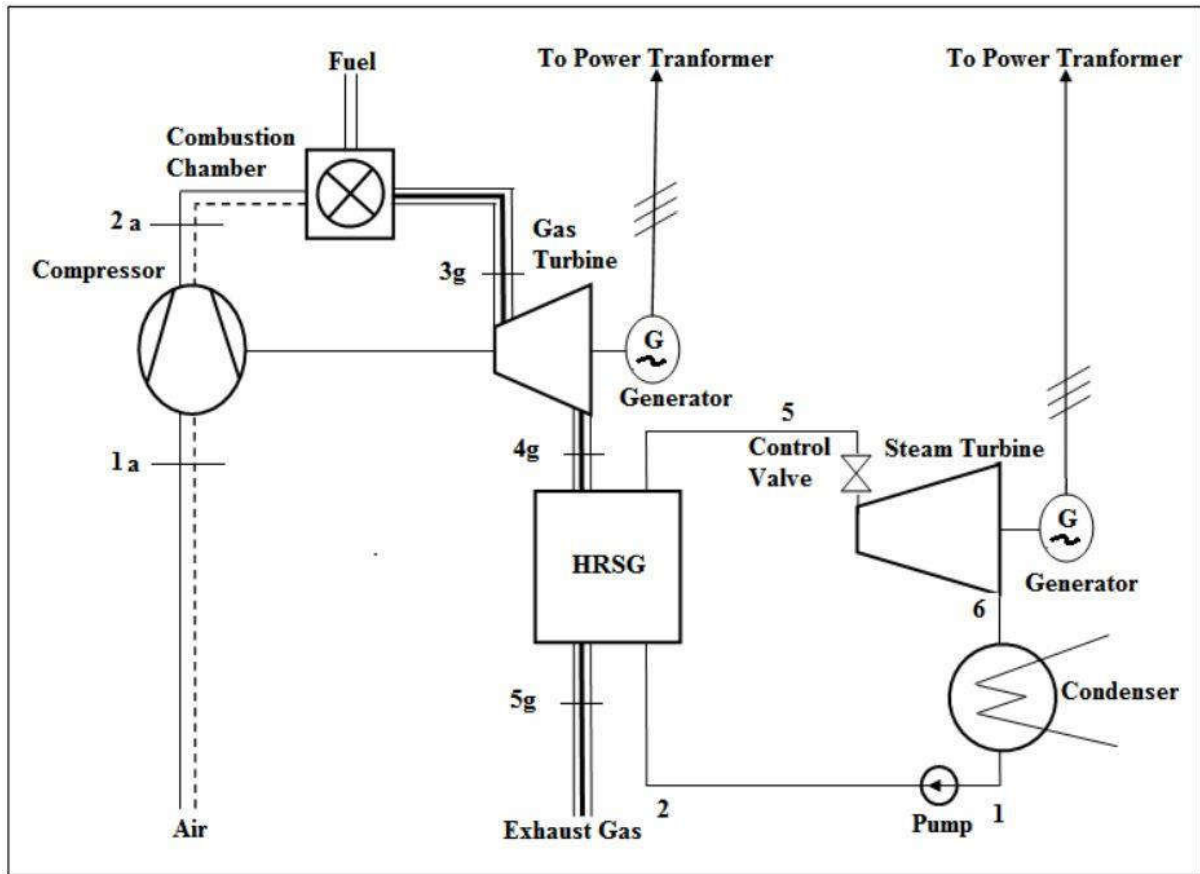


Figure 3-5: CCPP schematic, [15] S. 23

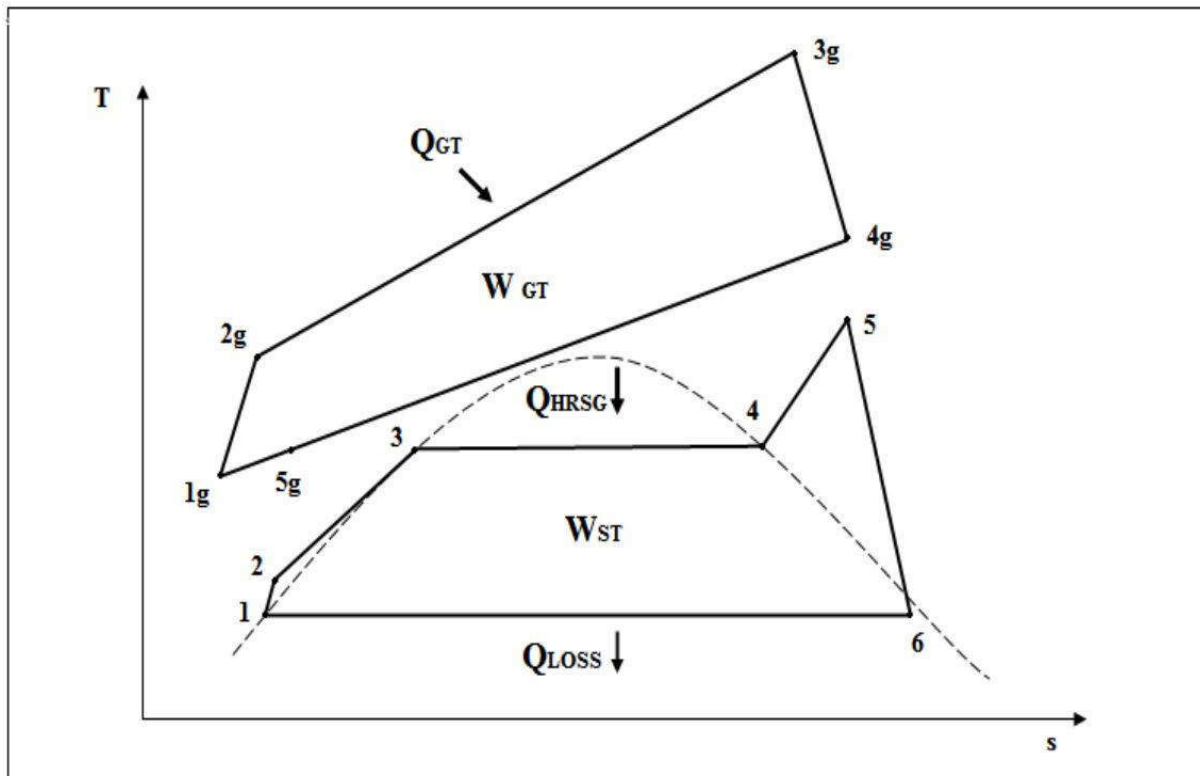


Figure 3-6: TS-diagram for the combination of Brayton and Rankine cycle, [15] S. 36

3.1.5 Failure Modes in CCPP

The following sub-chapter summarizes potential degradation failure modes in CCPPs, their causes, and the estimated impact.

GT Compressor Fouling

Air impurities that pass the inlet filter of the gas turbine for any reason may build up on the surface of the compressor blades. These deposits change the blades' inlet angles, the airfoil shape and increase the surface roughness. [14] [18]

Impact:

- isentropic efficiency reduction
- increased fuel consumption
- decreased flow capacity

GT Compressor Tip Clearance

Tip Clearance describes the distance between the rotating blades and the housing. Due to transient load conditions, the compressor blades may rub on the housing surface (because the blades heat up and expand faster than the housing) and the tip clearance increases. The increasing cross-section area leads to a rising leakage between the different pressure sections of the compressor.

Impact: [14]

- loss of compressor efficiency
- drop of pressure ratio
- decreased flow capacity

GT Expander Fouling

Turbine fouling or hot section fouling is caused by combustion products, which form deposits on the turbine blades. Contaminants leading to turbine fouling enter through the inlet air or liquid fuel, containing fuel additives. Fluids that are injected for NO_x control purposes are also part of the problem. [14] [18]

Impact:

- reduced power output and efficiency
- increased fuel consumption
- decreased flow capacity

GT Expander Tip Clearance

The increase of turbine tip clearance follows the same principle as the above-mentioned compressor tip clearance increase: The surface area between the rotor blades and the casing of the turbine increases because of blade rubbing caused by transient load conditions. An increasing mass flow bypasses the tips of the turbine blades leading to additional losses.

Impact: [14]

- increased exhaust temperature and flow capacity
- decreased turbine efficiency

GT Filter Clogging

The gas turbine's air inlet filter protects the compressor and the turbine from impurities in the ambient air. These pollutions include dust, sand, moisture, or anything else that might be suspended in the air. Filter clogging occurs as the impurities accumulate in the filter. [18]

Impact:

- increasing the pressure drop over the filter
- decreased plant efficiency and capacity

Steam Turbine HP Bowl to IP Bowl Leakage

Besides the leakage by design, seal damaging or weakening increases the leakage from the HP to the IP section of the steam turbine. The degradation of the sealing happens through misalignment, poor start-ups, or temperature excursions. Inner shell distortion or loose and overstretched bolting can also cause leakage from the HP bowl into to IP bowl at the horizontal joint. The direct mass flow from the HP to the IP turbine reduces the reheat cycle's mass flow. The reduced cold reheat flow causes overheating of the reheater tubes and may stress the spray coolers to control both HP and reheat steam temperatures in the HRSG and at the ST inlet. Once cooling flow limits are reached, load curtailing is necessary to avoid overheating. [19]

Impact:

- power-loss because of load reduction
- increased steam entrance through the sealing leads to higher measured (apparent) efficiency at the IP turbine [19]

Figure 3-7 shows the HP and the IP turbine section with the affected sealing package between them.

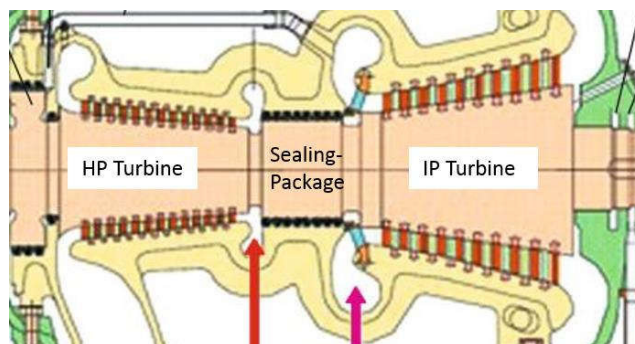


Figure 3-7: HP turbine and IP turbine with sealing package, [20]

HRSG Tube Failure and Leakage

Leakages in HRSG tubes occur due to various mechanisms: creep, fatigue, creep-fatigue, under deposit attack, flow-accelerated corrosion, mechanical erosion, stress-corrosion cracking, and overheating. These chemical and mechanical attacks on the tubes cause leakages and, therefore a loss of energy. [21]

Impact:

- decreased power output of the steam turbines
- decreased plant efficiency and capacity
- excessive make-up water consumption

Evaporator Blowdown Leakage

Evaporator blowdown leakage describes an unscheduled loss of mass flow and enthalpy caused by a failure of the drum's blowdown system. A broken valve, defect closing mechanism, or a sealing degradation are possible leakage reasons.

Impact:

- decreased power output of the steam turbines
- decreased plant efficiency and capacity
- excessive make-up water consumption

HRSG Fouling

Fouling is the term for deposit accumulation on surfaces, in this case on the heat exchange tubes of the HRSG. Fouling can occur on the inner or the outer surface of the heat exchanger.

On the inside of the tubes impurities of the feedwater may build up deposits, leading to under-deposit corrosion and overheating failures. [22] [23]

Impact:

- degradation of the heat transfer capability

On the outer surface, flue gas may attack the tubes. If fuel oil is used to fire the gas turbine, the sulfur content of the fuel can lead to ammonium-salt (Ammonium Bisulfate and Ammonium Bisulfite) deposits at components after the catalysator. An Ammoniac (NH_3)-slip, occurring from a not complete chemical reaction of all NH_3 molecules in the catalysator enables the formation of Ammonium. [24]

Impact:

- degradation of the heat transfer capability
- increased pressure drop on the gas side

Letdown Leakage

Letdown valves are part of the turbine bypass system which provides the opportunity to discharge steam directly into the emergency letdown to minimize transient stresses during a sudden load reduction. [25] In a triple pressure CCPP the process steam is lead in three different paths:

- Bypass for the HP-turbine: High-pressure steam → cold reheat (CRH); to maintain cooling for the reheater even if the turbines are shutdown
- Bypass for the IP-turbine: Hot reheat (HRH) steam → Condenser
- bypass for the LP-turbine: Low-pressure steam → Condenser

Malfunction of the letdown valves leads to a steam mass flow in the mentioned paths even during normal operation.

Impact:

- loss of process steam
- decreased power output of steam turbines
- decreased plant efficiency

ST Blade Erosion/Fouling

The expansion of the steam flow in the last stages of the LP turbine causes condensation. The resulting droplets damage the turbine blades and lead to erosion and a change in the surface roughness. In higher pressure stages thermal erosion, fouling, and collision of particles also cause abrasion and impact the blade surface. [26]

Impact:

- decreased steam turbine efficiency
- may increase vapor condensation in the turbine [26]

ST Tip Clearance

An increase of tip clearance occurs on steam turbines leads to the same effect as for gas turbines (see GT compressor and turbine Tip clearance). The rubbing of the turbine blades is induced by the highly dynamic operation of the power plant and transient load conditions.

Impact:

- decreased steam turbine power output
- decreased steam turbine efficiency

ST Low-Pressure Exhaust Loss

The term steam turbine exhaust loss (in kJ/kg) describes the kinetic energy of the steam that leaves the last stage of the LP turbine plus the energy loss through friction and additional losses through vortex generation in the exhaust diffuser and the hood. This energy will dissipate in the condenser and is not available to generate electrical power. The exhaust loss depends on the steam mass flow and the annular exhaust velocity which is determined by the turbine back pressure and the blade geometry (see Figure 3-8). Erosion and fouling change the surface roughness of the turbine blades and affect the characteristics of the exhaust loss behavior of the LP turbine. [27] [28]

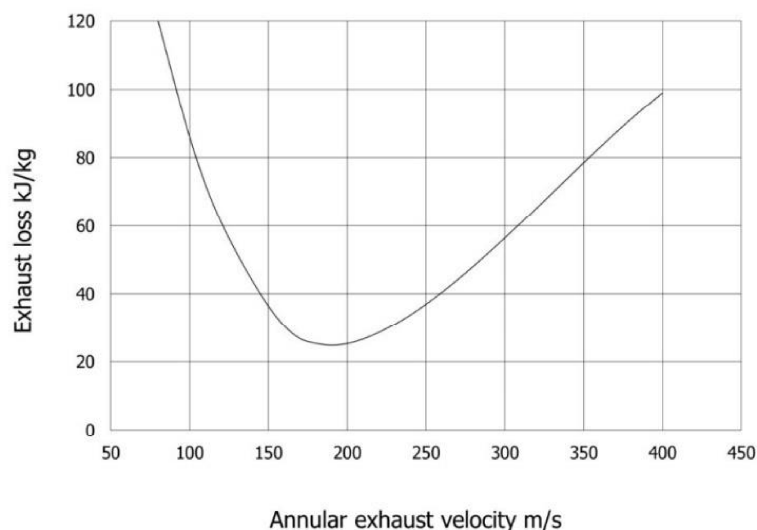


Figure 3-8: Exhaust loss curve of a steam turbine, [29]

Impact:

- increased exhaust loss
- decreased LP steam turbine power output
- decreased LP steam turbine efficiency

Condenser Incondensable Gas Accumulation

In the condenser, the physical state of the process steam changes to liquid water which is then returned to the deaerator. The condensate temperature and its corresponding steam pressure determine the back pressure of the LP turbine. Therefore, the temperature of the cooling water (water side of the heat exchanger) should be as low as possible, to maintain low pressure in the condenser. A condenser pressure above the corresponding steam saturation pressure of the condensate indicates the accumulation of other gases in the condenser.

Steam volatile chemicals (chemicals with a boiling temperature below the steam temperature) such as amines are used for boiler water conditioning. At the elevated operation temperatures in the HRSG, the amines break down to ammonia which causes damage to condenser tubes. Ammonia is incondensable in the field of the operation parameters of a condenser and relies on the extraction by the air removal system. [25] If the system doesn't work probably incondensable gases accumulate in the condenser and lead to higher back pressure for the LP turbine.

Impact:

- decreased LP steam turbine power output
- decreased LP steam turbine efficiency

Condenser Fouling and Scaling

Condenser fouling or scaling is a process where impurities form deposits on the heat exchanger tubes. These accumulations occur mainly on the cooling water side of the condenser and are accountable for the reductions in heat transfer capability that occurs in service. [30] Micro-fouling or scaling is caused by foulants as calcium carbonate, calcium sulfate, salts, and lime in the cooling water. [31] Additionally, biofouling may occur which means the deposition of micro-organism that built an organic film on heat exchanger surfaces. This type of fouling appears if seawater is used in the condenser. [32]

Impact:

- decreased heat transfer coefficient
- increased turbine back pressure because of the higher temperature in the condenser

Condenser Hotwell Subcooling

In the condenser, process steam condenses and covers the heat exchanger tubes. The heat transition continues as the condensate falls towards the bottom of the condenser, into the hot well. The temperature reduction beneath the condensation temperature is called subcooling. [33]

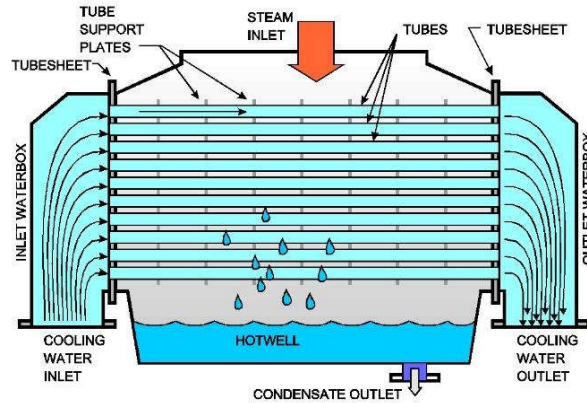


Figure 3-9: Condenser for steam power plants, [34]

Figure 3-9 shows a condenser without subcooling and a normal hotwell level. A certain degree of subcooling is normal but excessive subcooling indicates problems with the level control of the hotwell. The rising liquid covers the lower heat exchanger tubes of the condenser leading to a subcooled fluid and an increase in turbine back pressure due to a reduced remaining heat transfer area for condensation. The degrees of subcooling are shown in Figure 3-10 by the difference between the condensation temperature T_1 and the temperature in the hotwell $T_{1'}$. The dark blue area between the points 1 and 1' presents the additional heat rejection due to subcooling.

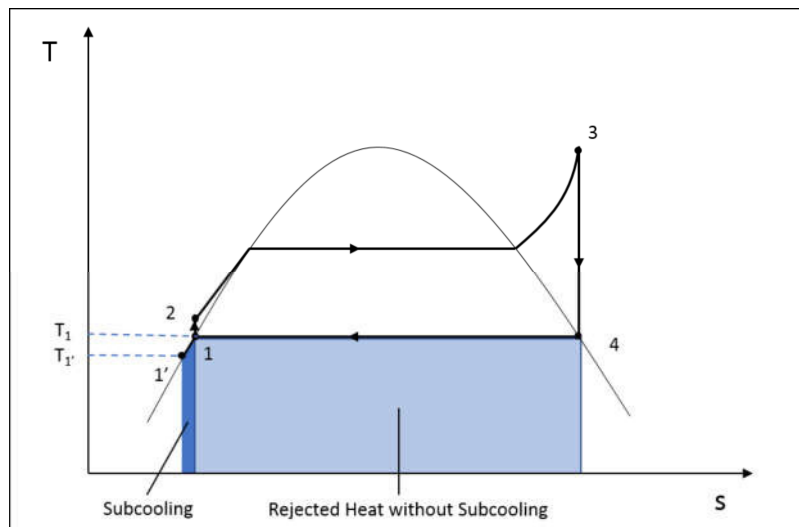


Figure 3-10: 1-2-3-4: Ideal Rankine cycle; 1-1': Condensate subcooling

Impact:

- decreased efficiency because reheating of the condensate will require more energy
- increase of LP turbine back pressure

3.2 Artificial Intelligence

The definition of artificial intelligence (AI) involves the philosophical question of what intelligence itself means and how to determine if an artificial entity is capable of thinking. This short introduction to artificial intelligence focuses on technical implementation rather than on philosophy. Alan Turing, a famous English mathematician and one of the first computer scientists developed the Turing test, an empirical test for artificial intelligence. In this test, an interrogator communicates with a real human being and an artificial entity at the same time. If the interviewer can't distinguish the human from the artificial intelligence by their answers to the posed questions, then the artificial entity is considered intelligent.

Since the introduction of the Turing test in 1950, the research field of artificial intelligence has grown and today includes various subcategories. This work deals with the methods of neural networks and their application. For further information on artificial intelligence, the reader is referred to [35] and [36].

3.2.1 Neural Networks

The first models of artificial neurons were developed in the 1940s resulting in the research field of artificial neural networks (ANN). The initiating idea was to model the neurons of biological brains and their information processing. In the 1950s neural networks lost scientific interest because of other approaches to build AI systems (for instance expert systems using deterministic reasoning or statistical systems such as Bayesian belief networks) and due to lack of computational power at that time. However, with the increasing power of computers, new training algorithms and the various application possibilities, neural networks gained popularity again. A large number of artificial neurons are assembled in an organized structure which build modern ANNs. Neural networks are used in applications like speech recognition or computer vision and other fields for pattern recognition. [35]

The Biological Model of Neurons

An illustration of a neuron in a human brain is shown in Figure 3-11. The dendrites collect input signals from other neurons and transmit them to the cell body. If the intensity of the accumulated signals exceeds a certain threshold, the neuron itself sends a signal out to other

neurons. The axon and the corresponding terminals serve as a linking structure to communicate with other neurons. [35]

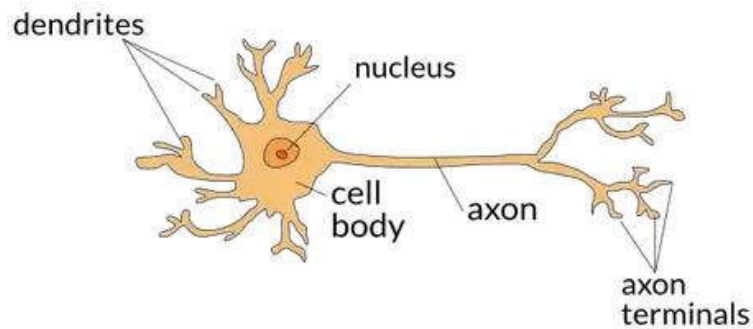


Figure 3-11: Biological neuron model, [37]

Artificial Replication of Neurons and Network Structure

The input of an artificial neuron is represented by a vector (x_1, x_2, \dots, x_n) . The elements of the vector are multiplied with specific weights (w_1, w_2, \dots, w_n) for each element and added up to a weighted sum z (3-8). Further, a neuron assigned bias (b_0) is added. Subsequently, an activation function f is applied on the sum value z . This function defines the output value y , which is passed to the following neurons or is displayed as the final result. Figure 3-12 shows the simplest neural network, the perceptron which is often referred to as the first machine learning algorithm. [38] It consists of only one neuron with different inputs and one output. The perceptron acts as a binary classifier, meaning that the output of the network only has two possible outcomes. With the chosen discrete activation function seen in (3-9), the output y can either be -1 or 1. [35]

$$z = b_0 + \sum_{i=1}^n w_i \cdot x_i \quad (3-8)$$

$$f(z) = \begin{cases} 1 & \text{if } z > 0 \\ -1 & \text{otherwise} \end{cases} \quad (3-9)$$

$$y = f(z) \quad (3-10)$$

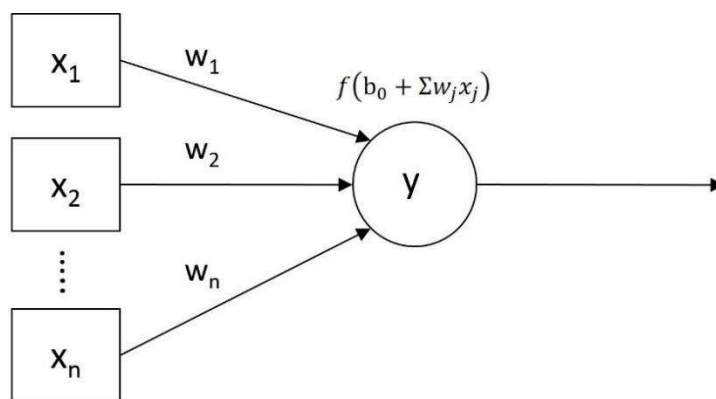


Figure 3-12: Perceptron: Neural network model consisting of one neuron, [35] S. 374

The activation function can also be a function resulting in a non-discrete value. Then the output y represents the probability for a certain prediction. For example, the probability that somebody has a disease. If the output value exceeds a defined threshold, the prediction result is true and otherwise false.

Complex problems like multilabel classifications lead to enlarged networks containing multiple hidden layers, besides the input and output layer. Hidden layers built the linking and data processing structure between the input and the output layer as shown in Figure 3-13. Every node in each hidden layer represents a neuron and gets input data from the neurons of the previous layers. The nodes run through the above-explained process of calculating a weighted sum, applying an activation function and transmitting data to the neurons of the following layer until the output layer is reached. However, choosing the optimal number of layers and neurons within the ANN for a specific application is not trivial and still the topic of research.

Multilabel Classification

Classification problems with more than two possible outcomes require neural networks that are capable of solving multi-label classification problems. The output layer of such networks includes one neuron for each class that shall be identified. The Softmax activation function (see 4.5.1), a function designed for multinomial classifications, calculates a probability value for each neuron of the output layer summing up to 100 percent in total. The output with the highest value represents the predicted class.

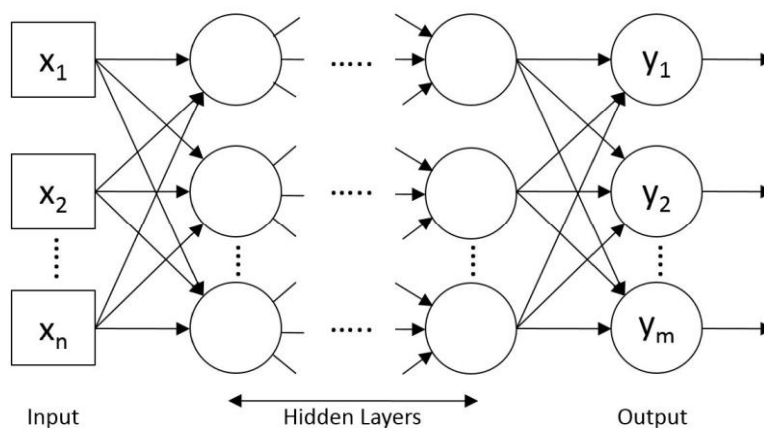


Figure 3-13: Neural network structure, [35] S. 386

Training of NNs

Neural networks produce predictions in the above-described feed-forward process. Initially, the weights of the connections and the neuron biases are set randomly. In every training loop, the weights and biases are adjusted to optimize the prediction accuracy. Thereby the input data passes through the neural network, resulting in a prediction. The difference between the prediction and the actual value yields to the so-called “loss”. The loss is calculated in the loss function chosen by the programmer. This loss is propagated from the output through the hidden layers to the input layer, called backpropagation. In every step backward the weights of the neuron connections and the biases of the neurons are adjusted to minimize the error for the next run. Different optimizers, which are chosen by the programmer, enhance the performance of the network by applying their corresponding mathematical optimization functions in the backpropagation process. This procedure is carried out iteratively until the prediction error reaches its minimum.

The goal is to train the neural network to recognize patterns in the input values and predict the consequent output. This training process is also called “fitting” the network to the data. If the network is not trained probably underfitting or overfitting may occur. An underfitted network is not able to recognize the characteristic pattern of the input data. An overfitted network “remembers” the patterns of the already seen training data, but can’t make accurate predictions when exposed to new data as it is too strongly fit to the training data (see Figure 3-14).

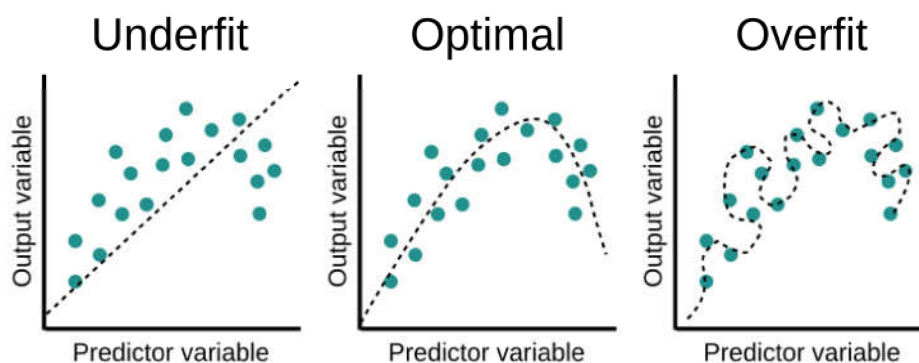


Figure 3-14: Examples of fittings of a neural network to training data, [39] S. 253

Hyperparameters

Some parameters of NNs, which are usually not modified through the training process, require initialization before the training process starts. These parameters are named hyperparameters. They include, among others, the number of layers, the number of neurons in each layer, batch size, learning rate and epochs. As mentioned before, no specific rules for setting hyperparameters exist, but methods for tuning them can be applied (see 5.1.1). The following listing describes the above-mentioned parameters.

- Epochs: The number of epochs determines how often each sample (pair of input and output data) is passed through the network in the training process. One epoch is finished when every pair of input and output data is processed once.
- Batch Size: The number of samples that are handed to the network to be processed at once. For example: If we have 100 training samples and we choose a batch size of 20, then the training data set is split into 5 batches of 20 samples. When all 5 batches are passed through the classifier, one epoch is finished.
- Learning rate: The step size for changing the internal variables of the layers during the optimization process. For example, the above-described weights are updated through the backpropagation process. For the perceptron, the new weight w_j of a certain connection is computed by subtracting the calculated loss gradient multiplied with the learning rate q from the old weight w_i of the previous learning step (see 3-11). This method of optimizing the weights and biases with the loss gradient is called stochastic gradient descent optimization.

$$w_j = w_i - q \cdot \frac{\partial Loss}{\partial w_i} \quad (3-11)$$

3.2.2 Statistical Methods for Performance Evaluation

Before we discuss the performance parameters for neural networks, some statistical terms are introduced. A statistical test with two possible outcomes is called a binary classifier. A disease test represents such a classifier: The test predicts if a person has a certain disease (=Positive) or not (=Negative). These predictions can represent four different states which are recorded in a confusion matrix (see Figure 3-15):

- True Positive (TP): The person is ill and is predicted ill
- False Positive (FP): The person is not ill and is predicted ill
- False Negative (FN): The person is ill and is predicted not ill
- True Negative (TN): The person is not ill and is predicted not ill

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Figure 3-15: Confusion matrix for binary classifications

The number of each of these possible outcomes (TP, FP, TN, FN) determines the performance of the applied test. The quality of classifiers can be evaluated by various parameters. The equations below (3-11, 3-12, 3-13, 3-14) describe some of the most common ones. It should be noted that these parameters are utilized for all kinds of statistical tests and machine learning applications, not only for binary classifiers.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3-12)$$

$$Precision = \frac{TP}{TP + FP} \quad (3-13)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (3-14)$$

$$Specificity = \frac{TN}{TN + FP} \quad (3-15)$$

Accuracy is the ratio of all correct classifications (TP + TN) to the total number of all classifications (TP + TN + FP + FN).

Precision is the ratio of all correct classified positives (TP) to all positive predictions (TP + FP). It defines the certainty for a positive prediction to be true.

Sensitivity is the ratio of all correct classified positives (TP) to the number of actual positives (TP + FN). It defines the ability of the network to identify an actual positive.

Specificity is the ratio of all correct classified negatives (TN) to the number of actual negatives (TN + FP). It defines the ability of the network to identify an actual negative.

The importance of each evaluation parameter depends on the purpose of the classifier. For a virus detecting test, high sensitivity is important, because a low rate of false-negative examples is desirable. A not detected contagious person may have contact with healthy persons and infect them. In contrast, for the detection of degradation failures in power plants, a high precision may be beneficial. Because a false positive possibly leads to a revision and opening of a faultless system, which would cause plant downtime and unnecessary maintenance costs. However, in most cases, high precision and high sensitivity are not achieved at the same time but setting a prediction threshold at a suitable level affects these parameters. A threshold is a barrier set at a defined level that has to be reached (or exceeded) by the classification sample to lead to a certain prediction. For example, a tested person is predicted as ill if the viral load of this person exceeds the set threshold. A low threshold leads to more false positives and fewer false negatives and a corresponding high sensitivity. A higher threshold level increases the precision and simultaneously decreases the sensitivity. Referred to the disease example, a test with a low threshold will detect persons even with a modest viral load but also diagnoses more healthy persons to be ill.

The explained measures are typically applied to binary classifications. For a multiclass classification problem, the parameters are calculated class-wise (3-16, 3-18) and a weighted mean is computed (3-17, 3,19) [40]. N symbolizes the total number of classes and c_{ij} stands for the confusion matrix entries with their corresponding index for the predicted class i (row index) and the actual class j (column index).

		Actual Class		
		$C_0 \dots C_{k-1}$	C_k	$C_{k+1} \dots C_n$
Predicted Class	$C_{k-1} \dots C_0$	TN	FN	TN
	C_k	FP	TP	FP
	$C_n \dots C_{k+1}$	TN	FN	TN

Figure 3-16: Confusion matrix for a multi-label classification problem

$$Accuracy_{overall} = \frac{\sum_{i=0}^N C_{ii}}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}} \quad (3-16)$$

$$Precision_{class} = \frac{TP_{class}}{TP_{class} + FP_{class}} \quad (3-17)$$

$$Precision_{overall} = \frac{\sum_{i=0}^N Precision_i \cdot (TP_i + FN_i)}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}} \quad (3-18)$$

$$Sensitivity_{class} = \frac{TP_{class}}{TP_{class} + FN_{class}} \quad (3-19)$$

$$Sensitivity_{overall} = \frac{\sum_{i=0}^N Sensitivity_i \cdot (TP_i + FN_i)}{\sum_{i=0}^N \sum_{j=0}^N C_{ij}} \quad (3-20)$$

4 MODEL DESIGN

This chapter deals with the design of the CCPP model and the procedure of simulated data generation and pre-processing. Further, the setup for the neural network and its training and testing methods are explained.

4.1 Software

The following descriptions provide a short insight into the software which was utilized in this work.

4.1.1 EBSILON

For generating an accurate model of the power plant, the EBSILON®Professional heat balance software ([4]) was used which has proven to be a valuable modeling tool for simulation-based design- and operation optimizations of all types of thermal power generation systems. It is used by utilities, engineering companies, equipment manufacturers and research organizations worldwide. Through individual model components, the gas turbine and/or engine performance characteristics can be integrated with a detailed plant model, and in-depth thermodynamic analysis can be performed benefiting from the features of EBSILON®Professional, such as:

- individual equipment characteristics in design and off-design mode
- full record of all gas, water/steam and electrical flows of the plant
- flexibility in equipment arrangement, plant configuration and mix of technologies
- a powerful, fast and reliable equation-based solver
- open architecture to include user-defined models for new technology or vendor data
- a state-of-the-art graphical user interface and a wide variety of output options in graphical and tabular formats and
- an interface to Microsoft® Office Excel®.

ENEXSA GmbH has further extended the usability of EBSILON by adding a framework for distributed calculation technology which allows for fast processing of very large simulation tasks such as the hourly re-calculation of the annual operating profile of a power plant. [41] This work uses this distributed calculation technology for the generation of the operating patterns of the 'healthy' power plant and with various failure modes, enhanced by an interface to Python for data pre- and post-processing.

4.1.2 Python

Python is a popular programming language for instance for handling huge data files. It allows automating data manipulations and repetitive tasks. The PyCharm IDE facilitates the creation and debugging of Python programs. Besides PyCharm, the web-based JupyterLab IDE is utilized for programming and executing python files directly on the ENEXSA webserver.

4.1.3 Tensorflow and Keras

Tensorflow is an open-source platform with a focus on machine learning. It provides various tools, libraries and community resources for programming state-of-the-art machine learning applications. [42] Keras is designed to ease the construction of deep learning neural networks. This API is built on top of Tensorflow and enables the utilization of the Tensorflow libraries. [43]

4.2 CCPP EBSILON-Model

In this thesis, a state-of-the-art CCPP model from ENEXSA was utilized. The power plant model includes a gas turbine, a triple pressure HRSG with one reheat cycle and three steam turbine sections (HP, IP and LP). The simulation in EBSILON is carried out in two consecutive stages: the design mode and the off-design mode.

Design mode: In the first step, a model is built in the design mode. The topology is set up, all design parameters and the physics of the components for the power plant is defined.

Off-design mode: The simulation of the plant operation under different environmental and load conditions is carried out in the off-design mode.

Before the failure mode simulation and the process of labeled data generation can be explained, some terms have to be specified:

- **Specification values (or parameters)** represent all values that have to be set to run the simulation. They are also called input values of the EBSILON model.
- **Result values** represent all values that are calculated in the simulation run. They are also called output values of the EBSILON model.
- **Input (or input values)** of the NN represent all values that are fed to the input layer of the NN. These values contain all result values of the EBSILON model and certain specification values, seen in Table 4-2.

Three elements are necessary to characterize a power plant model:

- Topology (used components and their linking connections)
- Physics of the components' behavior

- Parameterization

The topology and the physics of the components' behavior and their mathematical description can only be adopted directly in EBSILON (in design-mode). The simulation in the off-design mode uses the component characteristics specified by the user or already pre-defined in Ebsilon to determine off-design performance for every model component. In each simulation run the specification values, including gas turbine target power, ambient conditions, the failure mode and the failure impact, are set from the input array. The results of the simulation and the respective ambient conditions represent the input values for the neural network. By setting measuring points in the model, the results at a specific process stream (mass flow, temperature, pressure, enthalpy, or power/heat flow) can be extracted. Measuring points were only placed in the model where measurement equipment is typically in operation in existing CCPPs. Thus, the input for the neural network is representing information that can be gathered in the real-life application. No additional simulation results are required.

4.3 Failure Mode Simulation

Each of the described failure modes in 3.1.5 results with a specific impact on the thermodynamic cycle. The applied simulation methods focus on the most severe effects and can't replicate the exact physical impacts of every failure, but represent good approximations. Further, these simulations are limited to the static operation of the power plant. Hence shutdown, starting or other dynamic processes were not investigated.

For the implementation of the investigated failure modes, the specification macro of the model in which all specification values are set, has been expanded for the selection of the active failure modes. The failure modes can be activated by selection via a drop-down menu and the corresponding failure impact (FI) can be set. The failure impact is a unitless input value ranging from 0 to 1, representing a relative impact between 0 and 100 percent. The maximum impact of a failure mode is defined in a script of the specification macro, which is executed before every simulation run. This script overwrites a specific design value of the affected component with a value affected by the specific failure to the specified extent. After the calculation and the extraction of the results, another script resets all model modifications and returns the model into a faultless state.

Besides the mathematical operations and parameter manipulations of the macros, various modifications of the model topology have been implemented. The added system parts, like the letdown system, are simultaneously activated with the corresponding failure modes. The following section presents the modifications for each failure mode.

It should be noted that the design values are changed linearly as a function of the FI. This can be done because the goal is to simulate the resulting impact on the plant operation and not the exact process leading to it.

Normal Mode

The normal mode represents the power plant operation without any failures.

GT Compressor Fouling

Compressor blade fouling leads to a loss of efficiency. The isentropic efficiency (ETAI) of the compressor component can be directly affected by modifying the ETAI parameter (4-1).

$$ETAI = ETAI \cdot (1 - FI \cdot 0.2) \quad (4-1)$$

This means that compressor blade fouling in this study is assumed to lead to a loss of isentropic efficiency up to 20 % of its initial “faultless” value.

GT Compressor Tip Clearance

Tip clearance increase at the gas turbine compressor directly affects its flow capacity. The airflow reduction is modeled by manipulation of the control system of the gas turbine. The gas turbine requires a certain amount of fuel and a corresponding airflow to reach the target load. The calculated airflow value is manipulated which leads to curtailing of the inlet mass flow of the compressor (4-2).

$$\dot{m}_{Compr,inlet} = \dot{m}_{Compr,inlet} \cdot (1 - FI \cdot 0.03) \quad (4-2)$$

GT Expander Fouling

Expansion turbine blade fouling affects the efficiency and the power output of the gas turbine. The reduction in efficiency is directly simulated by manipulating the nominal polytropic efficiency (ETAPN) of the gas turbine component in the off-design run (4-3).

$$ETAPN = ETAPN \cdot (1 - FI \cdot 0.2) \quad (4-3)$$

GT Expander Tip Clearance

Expansion turbine tip clearance leads to a mass flow over the tips of the turbine blades. This effect is modeled by implementing a bypass system. Separating the gas turbine into different stages enables the implementation of this bypass. In front of each turbine element, a splitter divides the gas stream into two paths. The flow ratio of the streams defines the amount of bypassed mass flow (4-4).

$$\frac{\dot{m}_{Bypass}}{\dot{m}_{Turbine}} = FI \cdot 0.01 \quad (4-4)$$

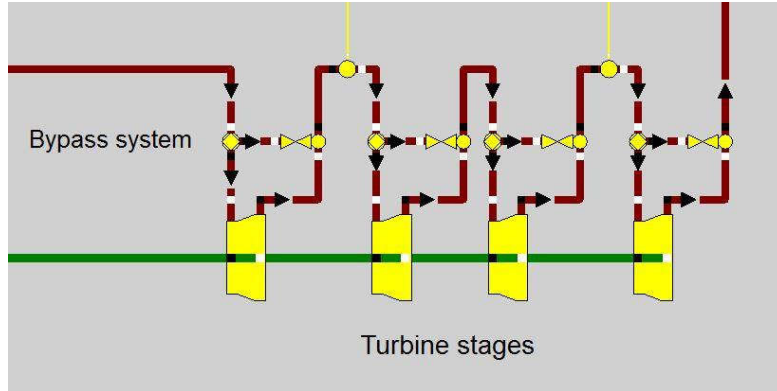


Figure 4-1: Gas turbine expander tip clearance simulation

GT Filter Clogging

The air inlet filter of the gas turbine is modeled by a pipe component. This structural element provides the possibility to define a nominal pressure drop ($DP12RN$) in a stream. In off-design mode, a pressure drop that depends on $DP12RN$ and the actual mass and volume flow is calculated. By manipulating the nominal pressure drop, the increased pressure drop due to filter clogging is simulated (4-5).

$$DP12RN = DP12RN \cdot (1 + FI \cdot 2) \quad (4-5)$$

Steam Turbine HP Bowl to IP Bowl Leakage

For the simulation of the HP bowl to IP bowl leakage (also called N_2 leakage) of the steam turbine, a seal-system has been implemented. Shaft sealing components (labyrinth seals) determine the amount of leakage mass flow. In design mode, the desired mass flow is set and the seal characteristics, like the nominal free cross-section (CAN), are calculated. In off-design mode, the seal characteristics and the pressure difference between the inlet and the outlet of the sealing determine the leakage quantity. To simulate the failure, the CAN value is manipulated in the off-design mode (4-6).

$$CAN = CAN \cdot (1 + FI \cdot 10) \quad (4-6)$$

The leakage mass flow is directly proportional to the CAN value. Therefore, the leakage quantity is varied from 1 to 11 times of the nominal mass flow, depending on the failure impact. In normal mode ($FI=0$) leakage by design occurs.

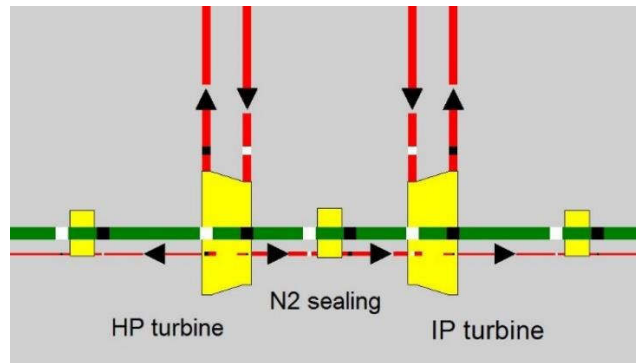


Figure 4-2: Steam turbine HP bowl to IP bowl leakage simulation

HRSG Fouling

The process of fouling directly affects the heat transfer coefficient (k) of the heat exchangers in the HRSG. In the design mode, the heat transfer coefficient and the heat transfer area (A) are calculated, resulting in the nominal heat transfer factor ($KAN = k \cdot A$). This factor is manipulated in the off-design mode to simulate the failure (4-7).

$$KAN = KAN \cdot (1 - FI \cdot 0.2) \quad (4-7)$$

Let-down Leakage

For the leakage simulation, a shaft sealing component is utilized. The underlying mathematics of a labyrinth seal enables a physical correct modulation (see Steam Turbine HP Bowl to IP bowl leakage). The leakage mass flow depends on the present inlet pressure and the sealing characteristics. Manipulation of the nominal free cross-section leads to a variation of the leakage quantity (4-8). In Normal mode, no leakage occurs and therefore the CAN value is set to zero.

$$CAN = CAN \cdot (FI \cdot 10) \quad (4-8)$$

The letdown steam is directly lead to the connected steam paths. Hence the implementation of a spraying system is necessary to avoid temperature excursions in the downstream plant sections.

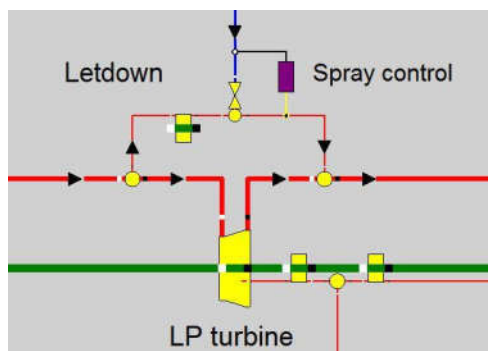


Figure 4-3: Letdown leakage simulation, shown on the example of the bypass for the LP section of the ST

Evaporator Blowdown Leakage

For the simulation of the evaporator blowdown leakage, a system that replicates the physical characteristics of a shaft sealing was implemented. The above-described shaft sealing component (see HP Bowl to IP Bowl Leakage) couldn't be used because of convergence issues in the simulation process. Instead, a piping element was deployed. In the design run, a nominal mass flow $M1N$ was set. According to the nominal mass flow and the appearing pressure drop between the drum and the ambient pressure, a mass flow in the off-design simulation is calculated. By manipulating the $M1N$ value the simulated leakage mass flow can be changed (4-9).

$$M1N = M1N \cdot (1 + FI \cdot 1.5) \quad (4-9)$$

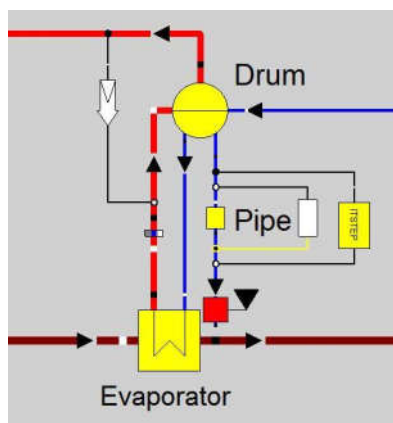


Figure 4-4: Evaporator blowdown leakage simulation

ST Blade Erosion and Fouling

Blade erosion and fouling result in a degradation of the steam turbine efficiency. This reduction is simulated by decreasing the performance factor on efficiency (PFETA) of the turbine component (4-10).

$$PFETA = PFETA \cdot (1 - FI \cdot 0.2) \quad (4-10)$$

ST Tip Clearance

An increasing tip clearance affects the flow conditions in the steam turbine and leads to an increased flow capacity. The mass flow through a turbine is determined by its design characteristics and the given pressure at the inlet and outlet. This relationship between pressure and mass flow is manipulated in the off-design simulation by varying the performance factor on flow (PFFLOW) of the turbine component (4-11). An increased PFFLOW leads to a higher mass flow at the same pressure difference.

$$PFFLOW = PFFLOW \cdot (1 + FI \cdot 0.2) \quad (4-11)$$

ST LP Exhaust Loss

In the off-design mode, the exhaust loss of the LP steam turbine is a function of the annulus exhaust velocity as the design characteristics are already set. The internal calculation of the exhaust loss can be manipulated to simulate a change in the characteristic exhaust loss behavior of the turbine (4-12). $E_{ExhLoss}$ describes the calculated exhaust loss before the manipulation.

$$E_{ExhLoss} = E_{ExhLoss} \cdot (1 + FI \cdot 0.2) + FI \cdot 5 \quad (4-12)$$

It should be noted that this modification can't replicate an exact modulation of the exhaust loss change through degradation. The applied manipulation simplifies a change in the exhaust loss behavior which further shall be detected by the neural network.

Condenser Incondensable Gas Accumulation

Accumulation of incondensable gases in the condenser leads to an increase in LP turbine back pressure. A separator component divides the stream coming from the LP turbine to the condenser. The separator provides the option to have two independent pressure levels, before and after that component. The value transmitter transfers the pressure level of the condenser, passing the separator, to the LP output stream. Within the transmitter, the offset pressure p_{Offset} can be defined. This offset represents the turbine back pressure increase (4-13).

$$p_{Offset} = FI \cdot 0.04 \quad (4-13)$$

The unit of measurement for the pressure offset is bar. Therefore, the maximum offset to the "faultless" condenser pressure equals 0.04 bar. Both the separator and the value transmitter are turned off in design mode and only affect the off-design simulation if the failure mode is activated.

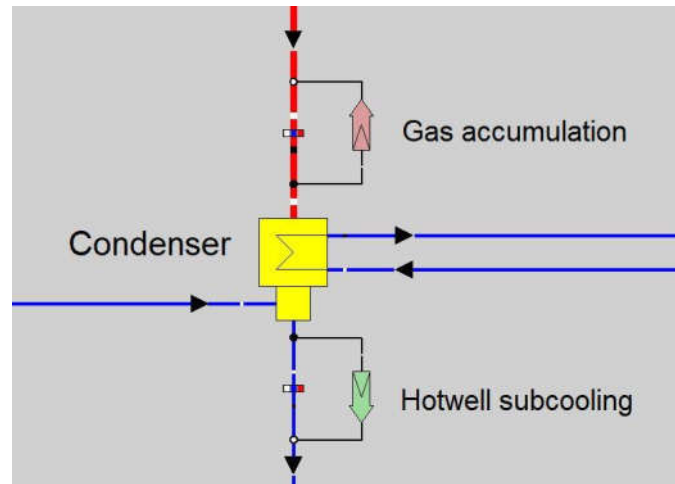


Figure 4-5: Condenser incondensable gas accumulation and hotwell subcooling simulation

Condenser Fouling and Scaling

Fouling affects the heat transfer coefficient of the condenser. The influence of this failure mode is modeled like the HRSG fouling, by manipulation of the KAN value in the off-design simulation (4-14).

$$KAN = KAN \cdot (1 - FI \cdot 0.4) \quad (4-14)$$

Condenser Hotwell Subcooling

The method of simulation for hotwell subcooling is similar to the incondensable gas accumulation (see Figure 4-5). But in this case, the temperature value is manipulated. A separator divides the stream from the hotwell to the feedwater pump and a value transmitter transfers the temperature of the hotwell, with an added offset, to the feedwater stream.

$$T_{Offset} = FI \cdot 5 \quad (4-15)$$

The unit of measurement for the offset is degrees Celsius. Therefore, T_{Offset} ranges from 0 °C to 5 °C (4-15), and the failure mode Condenser Hotwell Subcooling will thus produce condensate return temperatures that are lower by this offset compared to the operation of the “faultless” equipment.

As some of the presented failure modes occur on all three pressure levels of the HRSG components, their impact is simulated separately. Table 4-1 provides an overview of all simulated failure modes and their assigned numbers for the simulation.

Table 4-1: Failure modes

Number	Failure Mode	EBSILON Label
0	Normal Mode	Normal Mode
1	ST HP Bowl to IP Bowl Leakage	N2 Leakage
2	Evaporator Blowdown Leakage	Blowdown Leakage HP
3	Evaporator Blowdown Leakage	Blowdown Leakage IP
4	Evaporator Blowdown Leakage	Blowdown Leakage LP
5	HRSF Fouling	HRSF Fouling HP
6	HRSF Fouling	HRSF Fouling IP
7	HRSF Fouling	HRSF Fouling LP
8	Let-down Leakage	Letdown Leakage HP-CRH
9	Let-down Leakage	Letdown Leakage HRH-Cond
10	Let-down Leakage	Letdown Leakage LP-Cond
11	ST Blade Erosion and Fouling	ST Blade Erosion/Fouling HP
12	ST Blade Erosion and Fouling	ST Blade Erosion/Fouling IP
13	ST Blade Erosion and Fouling	ST Blade Erosion/Fouling LP
14	ST Tip Clearance	ST Tip Clearance HP
15	ST Tip Clearance	ST Tip Clearance IP
16	ST Tip Clearance	ST Tip Clearance LP
17	Condenser Incondensable Gas Accumulation	Cond Gas Accumulation
18	Condenser Fouling and Scaling	Cond Fouling/Scaling
19	Condenser Hotwell Subcooling	Cond Hotwell Subcooling

20	ST LP Exhaust Loss	ST LP Exhaust Loss
21	GT Compressor Fouling	GT Compressor Fouling
22	GT Compressor Tip Clearance	GT Compressor Tip Clearance
23	GT Expander Fouling	GT Turbine Fouling
24	GT Expander Tip Clearance	GT Turbine Tip Clearance
25	GT Filter Clogging	GT Filter Clogging

4.4 Data Generation

Training the neural network to predict failure modes in CCPs requires a large amount of data, that represents the operation characteristics of the power plant over the entire range of load and ambient conditions. The following listing represents the established workflow for data generation:

- I. Implementation of the failure modes in the EBSILON model
- II. Definition of the specification and result parameters in an Excel-file
- III. Defining the minimum and maximum values for the specifications
- IV. Import of the models' input and output configurations in a Python Script
- V. Creating equally distributed random values for the specification parameters within the set boundaries
- VI. Creating normally distributed failure impact values
- VII. Transmission of the EBSILON model and the specifications to the ENEXSA server
- VIII. Downloading the simulation results from the server and labeling the data with the corresponding failure mode

Table 4-2 shows a listing of the specification parameters and simulation results. The gray shaded values represent the input for the neural network. While the failure mode and failure impact are specification values (i.e. inputs) in the Ebsilon simulation, the NN shall determine the failure modes as results by “recognizing” a specific pattern in the simulated process data.

Table 4-2: Model parameters

Name	Type	Unit	Description
Ambient_Temp	Specification	°C	Ambient temperature
Ambient_Pressure	Specification	Bar	Ambient pressure
Ambient_Rel_Hum	Specification	unitless	Relative ambient humidity
Cool_Water_Temp	Specification	°C	Condenser cooling water temperature
P_Target_GT	Specification	MW	Gas turbine target power
Failure_Mode	Specification	unitless	Chosen failure mode
Failure_Impact	Specification	unitless	Failure Impact value
Gas_Massflow	Result	kg/s	Inlet gas mass flow
GT_Gen_Power	Result	MW	Gas turbine power generation
GT_FG_Temp	Result	°C	Gas turbine flue gas exhaust temperature
Comp_Out_Temp	Result	°C	Compressor output temperature
Comp_Out_Pressure	Result	°C	Compressor output pressure
HRSO_HP_Temp	Result	°C	HRSO HP stream temperature
HRSO_HP_Pressure	Result	bar	HRSO HP stream pressure
HRSO_HP_Massflow	Result	kg/s	HRSO HP mass flow
HRSO_HRH_Pressure	Result	bar	HRSO HRH stream pressure
HRSO_CRH_Temp	Result	°C	HRSO CRH stream temperature
HRSO_CRH_Pressure	Result	bar	HRSO CRH stream pressure
HRSO_IP_Temp	Result	°C	HRSO IP stream temperature

HRSG_IP_Pressure	Result	bar	HRSG IP stream pressure
HRSG_LP_Temp	Result	°C	HRSG LP stream temperature
HRSG_LP_Pressure	Result	bar	HRSG LP stream pressure
ST_LP_OUT_Pressure	Result	bar	LP steam turbine output pressure
ST_Gen_Power	Result	MW	Total generated power by the steam turbines (HP, IP, LP)
CW_OUT_Temp	Result	°C	Cooling water temperature after the condenser
Cond_OUT_Temp	Result	°C	Condenser output condensate temperature

Failure Impact Distribution

As degradation failures built up over time, detecting them at an early stage is beneficial. Therefore, the NN was trained on an overweight of simulation runs with low failure impacts. The failure impact values are normally distributed with a mean μ of 0.3 and a standard deviation σ of 0.3 (Figure 4-6). Additionally, the absolute value of all results was calculated and negatives were changed into positives. The resulting failure impacts above 1 were assigned with a new value. This ensures that the majority of the data sets contain low failure impacts, leading to high accuracy in this area.

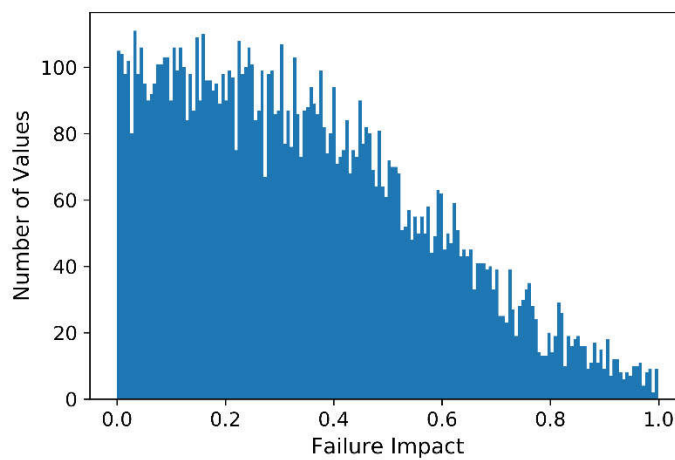


Figure 4-6: Histogram of failure impact distribution for 10,000 generated values

Target Power Correction

The baseload capability of a gas turbine depends on ambient conditions and performance level and thus is a result of the model simulation. Depending on the input for a target power output the model decides whether the target can be achieved at current conditions or if the target is beyond capabilities. Therefore, targeting to achieve the same ratio of part-load and baseload simulations at different conditions requires a dependency of the target power input and ambient conditions. Numerous calculations at baseload operation with varying ambient specifications were processed to derive linear correlations which were then applied as limits for the target power input for the data generation process.

4.4.1 Data Validation

The performance of neural networks depends on the quality of the underlying training data. Hence data validation is a critical point before building a NN. The simulated data have thus been validated to assure that the implemented failure modes on specific equipment impact the power output of the turbines and the overall plant efficiency as expected.

For the validation process, data sets with a variation of the failure impact between 0 and 100 percent were generated. All other specification parameters were held constant. Figure 4-7 shows for instance the impact of the accumulation of incondensable gases in the condenser on the power output of the steam turbines. The back pressure of the LP steam turbine increases and reduces the pressure difference between turbine section inlet and exit that can be utilized in the expansion process. Similarly, all failure modes have been investigated on their impact either on the steam turbine power or the gas turbine power and the overall plant efficiency.

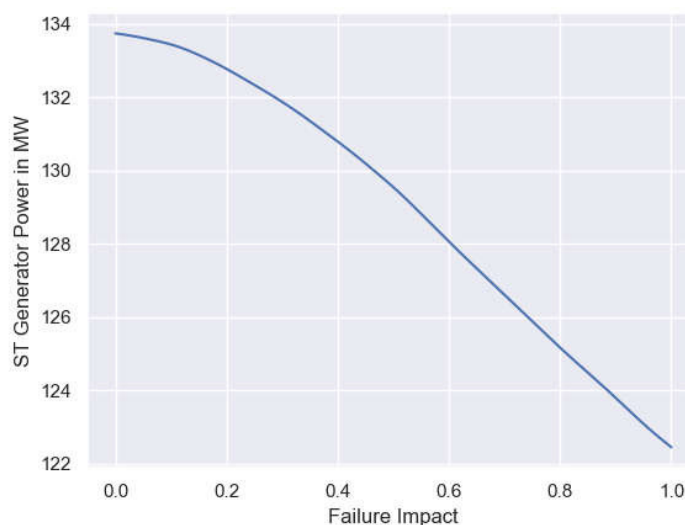


Figure 4-7: Effect of condenser incondensable gas accumulation on the power output of the steam turbines

Further, the influence of the failure modes on all measured process parameters, which are input values of the NN, was investigated. It was found that for example, condenser fouling has a strong impact on the LP steam turbine outlet pressure and the condenser outlet temperature (see Figure Appendix 2). For this analysis, all specification values were held constant and only the failure impact was varied. The difference between the maximum and the minimum value of each input parameter was calculated. The maximum relative impacts of the failure modes were computed and merged into a heatmap (see Figure Appendix 1 and Figure Appendix 2). This plot shows the characteristic pattern of each failure mode and its influence on the power plant process measurements.

4.4.2 Data Preparation

Before the simulated data sets could be processed by the NN, certain preparation was necessary. Each data set consists of over 30000 simulation runs (one run for every 15 minutes in a year with 8760 hours), representing the impact of one failure mode under various ambient conditions. In total 25 failure modes have been implemented, resulting in 26 data sets (including one set for normal mode). These data sets were split into training and testing sets. Afterward, the 26 training and 26 test sets were assembled into one large training and one large test set. Each set contains input and output values for the NN. As the units of measurements and the corresponding values differ among the input values, they needed to be normalized. Each input value was divided by a defined maximum value, leading to a data set of numbers between 0 and 1. The maximum values were set by searching for the highest occurring values in the dataset for each input and rounding them up.

Each column of the data sets represents one simulation run and therefore one sample of NN inputs plus the corresponding label for the simulated failure mode. After normalization, the columns were shuffled to mix the failure modes and equally distribute them across the whole set.

Subsequently, the data sets were utilized to build the failure mode predicting network. The NN learned certain patterns by processing the training data set. Evaluation of the achieved performance was carried out by feeding the test data set to the NN (see 4.5.2).

4.4.3 Measurement Noise Integration

EBSILON models deliver so-called clean data. The simulation follows mathematical expressions and solves equation systems by iterating until a defined deviation limit is reached. However, real-world measurements come with systematic and random errors. Therefore, the applied methods of failure mode detection were tested and trained using clean as well as noisy data.

Random errors embody an unpredictable variation of the measurement value. They occur in normally distributed patterns and can be minimized by taking the average of repeated measurements over a certain period. Random errors were simulated by adding noise to the input values of the network.

The noise was applied to the data sets after the normalization process. Each input value received a random normally distributed deviation within defined 3- σ intervals. Random values outside the 3- σ interval got a new value assigned. The 3- σ limits of the distribution were set according to measurement accuracies in CCPPs shown in Table 4-1. These values are based on the long-term experience and knowledge of the experts of ENEXSA.

To test the influence of different levels of noise on the network performance, a noise factor was introduced. The defined measurement accuracies were multiplied with the noise factor and the applied error values changed accordingly. For example, a noise factor of 2.0 results in a 2-times bigger maximum random error. The model in 5.1.3 was trained on data with an applied noise factor of 1.0 and evaluated on data with noise factors between zero and 2.0.

Table 4-3: Measurement tolerances

Measured quantity	Noise Factor	Measurement Tolerance/ Max. Random Error
Temperature	1.0	$\pm 0.5 \%$
Pressure	1.0	$\pm 0.5 \%$
Mass flow	1.0	$\pm 1.5 \%$
Power	1.0	$\pm 0.2 \%$
Humidity	1.0	$\pm 1.0 \%$

Systematic errors cause a constant bias from the true value. They can't be minimized with statistical methods and have to be detected and corrected for affected measurements by qualified engineers. This work assumes that systematic errors are identified and corrected in measurement data preprocessing and evaluation. Hence systematic errors were not implemented in the simulated data.

4.5 Neural Network

Tensorflow and Keras provide access to an immense library of machine learning tools. The library contains functions and classes for the required structures to build, train and test a neural network. The following sub-chapters explain the utilized methods to construct the failure mode predicting NN.

4.5.1 Building of the NN

The network itself is embodied by the so-called model. Inside the model, the structure of the NN is defined. It contains all layers, neurons, connections, and the internal variables of the network.

Input Layer

The input layer is the first layer of every network. It consists of one neuron for each input value that is passed to the NN. Therefore, the shape of the input layer is equal to the length of the input data vector. In this case, the first layer had 24 input values (gray shaded lines in Table 4-2). In the first layer, no activation function was applied to the data.

Hidden Layers

From the input layer, the input values are multiplied with the weights and passed to the first hidden layer. Keras provides various layer types that can be utilized as hidden layers. The following list names just a few of them:

- Dense layers
- Convolution layers
- Pooling layers
- Recurrent layers

In this research work, only dense layers were employed. Convolutional and pooling layers are used for computer vision applications and recurrent layers are an important component for forecasting programs. However, the term “dense” relates to the connection method of the neurons. A dense layer connects each neuron of the previous layer with each of its neurons.

Within the hidden layer, the number of neurons and the applied activation function were chosen. Defining a suitable number of hidden layers and neurons is discussed in 5.1.1.

Activation Functions

For each hidden layer and the output layer, an activation function has to be chosen. The activation function is applied to the weighted sum of each neuron (see 3.2.1). In the hidden layers, the rectified linear activation function (**ReLU**) was used.

The **ReLU** function results in 0 if the input x is negative, otherwise, its output $f_{ReLU}(x)$ equals the input (4-16). This was function was applied in all hidden layers.

$$f_{ReLU}(x) = \max(0, x) \quad (4-16)$$

In the output layer, the normalized exponential function (called **Softmax**) was used. The output layer consists of N values, one value for each predicted class. As the investigated classification problem for failure modes in CCPs consists of 26 different failure modes (including failure mode 0), the output vector has 26 output values. Each value z of the output vector is exponentiated and normalized according to (4-17). [36]

$$f_{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j^N e^{z_j}} \quad (4-17)$$

The **Softmax** function results in a vector representing the probability distribution among the predicted classes. Each value of the vector ranges between 0 and 1, summing up to 1 in total. The class with the highest probability represents the predicted class.

4.5.2 Training of the NN

After defining the NN structure with all layers, neurons and activation functions, the training of the NN followed. The training process itself required the following settings:

The **number of epochs** defines how often the whole training dataset is passed through the network before the training is finished. This number was set to 100,000 and above because the implemented callbacks, which measure the progress of the fit, prevent the NN from overfitting. The training was continued until no further improvement was measured.

The **batch size** was varied between 2000 and 4000. A smaller batch size decreases the computing speed, but increases the training progress per epoch. A larger batch size leads to faster computing but a decrease in progress per epoch.

The **learning rate (LR)** defines the step size in which the internal variables of the network are adjusted in the backpropagation process. Setting an appropriate LR is critical for training a NN. A high LR leads to unstable learning behavior, while a low LR slows down the training process significantly. Keras provides a very useful tool to set a suitable learning rate, the

learning rate scheduler (LRS). This scheduler was used to vary the learning rate during the training process and to monitor the response of the training loss. The LR was slowly increased in small steps after every epoch. Figure 4-8 shows that the learning rate reached a point where the training process got unstable. Therefore, the LR was set slightly below the unstable operation point to ensure a stable training process at a good training speed.

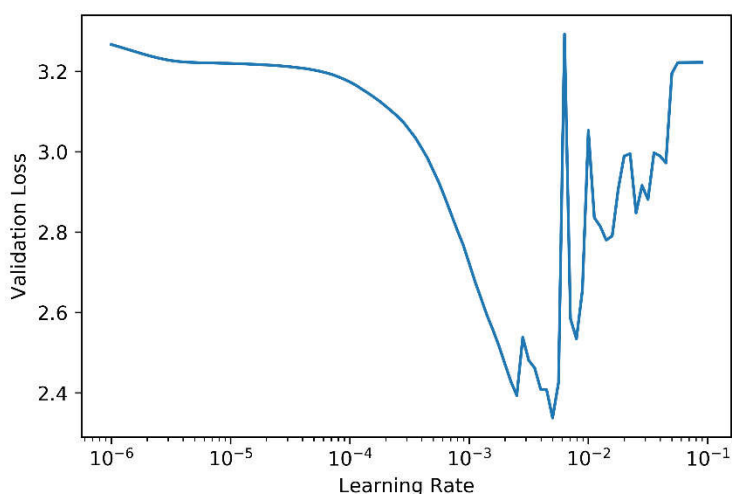


Figure 4-8: Effect of the learning rate on the validation loss

The **loss** function calculates the difference between the predicted output and the actual value. The goal of training the NN is to minimize this loss by adapting the internal variables of the network. To compute a loss value for this multilabel classification problem, the Sparse-Categorical-Crossentropy function was utilized. This function calculates the difference between two probability distributions. In this work, the predicted probabilities from the Softmax activation function of the output layer and the actual class were calculated.

Before the training, the training data set was split up into a training set and a validation set. The ratio between the number of training examples and validation examples was defined in the **validation split** parameter. The network was trained on the training set and the validation set was only used to check if the network still improves during the training process. The calculated training loss was utilized by the optimizer to adjust weights and biases, while the validation loss was used to monitor the progress of the training process. If the training loss still decreases while the validation loss rises, overfitting occurs.

The **optimizer** is the applied algorithm for adjusting the internal weights and biases. In this work, the **adam** optimizer is utilized. Besides weights and biases, this algorithm also adopts the learning rate during the training process leading to an efficient learning process.

Callbacks represent functions that are executed during the training process. These functions can either be self-written scripts or certain library implementations. The above-mentioned learning rate scheduler is one of them. The LRS was used in the first approach to find an appropriate learning rate. In the training process, the EarlyStopping (ES) and ModelCheckpoint (MC) functions were used. The ES algorithm compares the validation loss of the current epoch with the best value achieved so far. If no improvement of the model was achieved for a defined number of epochs, the training process stops and is finished. Additionally, the MC algorithm saves the new model weights and biases if an improvement occurs. With these two methods, the possibility of overfitting gets very low. Therefore, the number of training epochs doesn't have a huge impact on the model performance, because the training stops before overfitting happens.

The parameters and settings mentioned above were initialized before the training itself started. The training process was activated by applying the **fit** function to the model.

4.5.3 Performance Evaluation Process

As mentioned before, the initial simulation data sets for every failure mode were divided into training sets and test sets and further merged into one large training and one large test set. This enables the evaluation of the network with data it has never seen before. Otherwise, the NN could just "remember" the training data without learning the interdependencies between the input parameters and their resulting failure modes. The test set was utilized for the following analyzing methods.

In the first approach, the performance of the neural network was evaluated concerning its accuracy. This provided a quick insight if the trained network reaches acceptable accuracies or if fundamental problems occur. An accuracy test was carried out by calling the evaluate function for Keras models. The evaluate function simply processes the test data set and returns the overall accuracy of the NN. For further investigation on the performance, the confusion matrix for the NN was set up.

Confusion Matrix

To get predictions from the network the predict function from the Keras model was utilized. The test data was passed to the model that returns a prediction matrix. Every column of the matrix contains the probability values for one test example, visualized in Figure 4-9. As mentioned before, the class with the highest probability represents the predicted failure mode. The predicted result was compared to the actual value and the corresponding confusion matrix entry was added. All test examples were passed through this procedure

resulting in a complete confusion matrix. From this matrix, the precision and recall for the individual classes and the whole network were derived following the equations in 3.2.2.

Threshold

The influence of a threshold on classification problems was shortly discussed in 3.2.2. The following explanation provides a more detailed description of the threshold and its impact.

In Figure 4-9 an example of a probability distribution for a test sample is shown. Each bulk represents the probability of the corresponding failure mode, listed in Table 4-1. In this case, the threshold was set to 15 percent. Two values exceed this border and of course, the highest value represents the predicted failure mode 15.

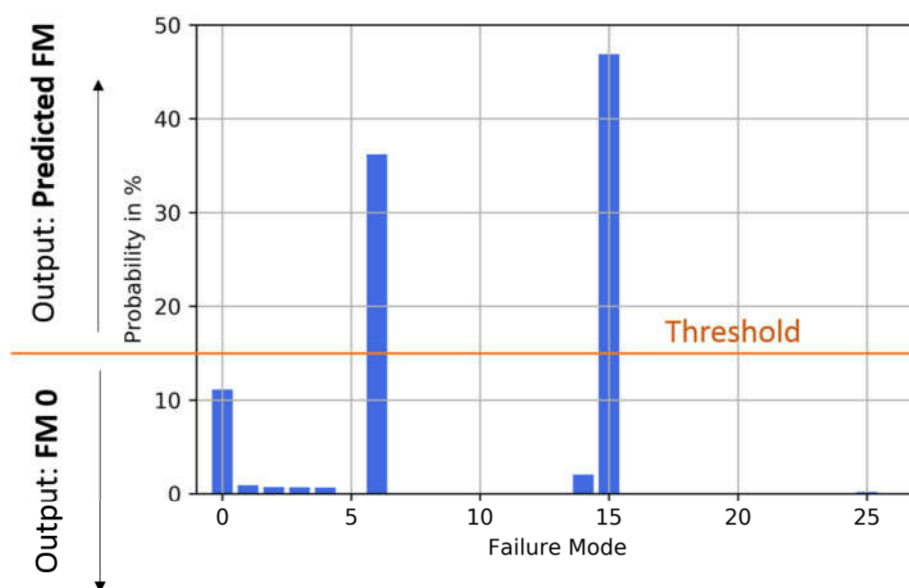


Figure 4-9: Example of a prediction from the NN with applied threshold

If none of the probability values is high enough to pass the defined threshold, the predicted output is set to failure mode 0. Failure mode 0 represents the normal operation mode. In other classification problems, this mapping of all weak results (results with a probability output below the threshold) on one defined other class wouldn't make sense. For example, if we try to classify different animals and every time a weak result occurs we say the animal is a mouse. But in this case of failure mode prediction, the mapping of a weak result to failure mode 0 has no impact on the real-life operation of the power plant. Because if no failure mode occurs (= failure mode 0) or if the network predicts a failure mode with low probability the result will be the same. The plant operator won't interfere. Therefore, every weak result is mapped on failure mode 0.

The second option of dealing with weak results would be to say the network can't predict an output with a high enough probability and therefore no classification is made at all. This approach would lead to a biased representation of the network performance because only simulation examples with a high output probability would be evaluated.

Setting a threshold directly affects the predictions of the network and therefore the resulting confusion matrix for the classification task. With the change of false positive, true positive, false negative and true negative results, also the performance parameters of the network vary. Hence the accuracy, precision and recall of the NN are functions of the threshold. The direct impact of the threshold on the results is explained by the following example. Failure mode 12 was the actual class and the threshold was set to 60 percent. Figure 4-10 shows a histogram of classifications in which any other failure than failure mode 12 was predicted. Figure 4-11 shows a histogram of classifications in which failure mode 12 was predicted correctly.

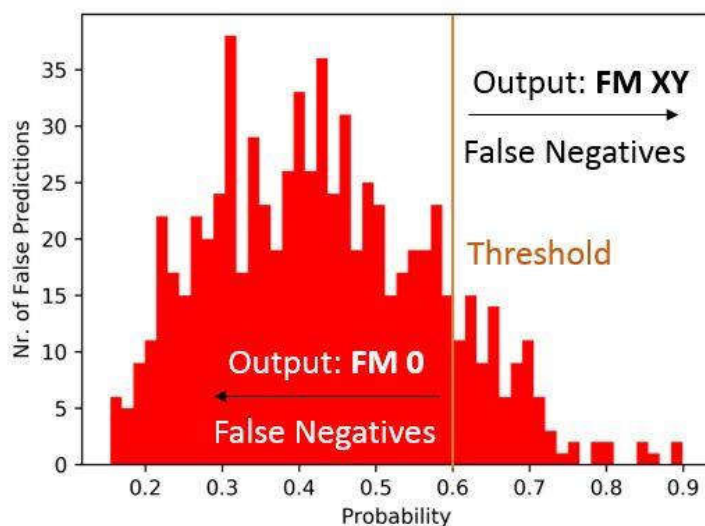


Figure 4-10: Influence of the threshold on false predictions of the NN for FM 12

Impact on false (false negative) predictions: Every other prediction than failure mode 12 causes a false-positive result in another class. For example, if failure mode 10 is predicted while failure mode 12 is present, failure mode 10 gets a false-positive result in the confusion matrix. With the specified threshold, all false-positive predictions that showed an output probability of less than 60 percent are mapped to failure mode 0. Because every prediction with a weak result gets mapped to normal mode. This leads to a reduction in false positives at all failure mode classes except for failure mode 0. All predictions with a probability above the threshold (FM XY in Figure 4-10) still cause false-positive results at the predicted failure modes.

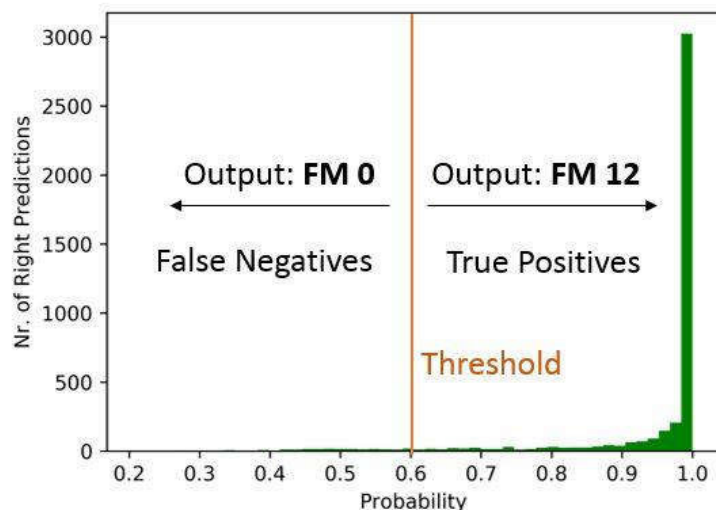


Figure 4-11: Influence of the threshold on correct predictions of the NN for FM 12

Impact on correct (true positive) predictions: All true positives which were predicted with a probability below the threshold are mapped to failure mode 0, causing a small decrease of true positives for failure mode 12. The results above the threshold still represent true positive predictions for failure mode 12. The optimum threshold value is thus a compromise between false positives for failure modes other than the actual and false negatives for this failure mode.

5 RESULTS

In this chapter, the results of the different network configurations are evaluated. The influence of the training data and the hyperparameters on the model performance is shown. The optimized model was tested on clean and noisy test data. Finally, the performance on two active failure modes was evaluated.

5.1 NN Model

All models were initially trained on datasets with an equal number of training examples for each failure mode. The upper and lower limits for the specification values of all simulation runs are shown in Table 5-1.

Table 5-1: Specification limits

Specification	Unit	Lower Limit	Upper Limit
Ambient_Temp	°C	5.0	45.0
Ambient_Pressure	bar	0.95	1.05
Ambient_Rel_Hum	unitless	0.0	1.0
Cool_Water_Temp	°C	10.0	20.0
P_Target_GT	MW	100	450
Failure_Mode	unitless	0	25
Failure_Impact	unitless	0.0	1.0

In a first attempt, a model with three hidden layers (128 Neurons, 64 Neurons and 32 Neurons) and ReLU activation functions was trained. This model reached an overall accuracy of 84 % until no further progress could be achieved. This result served as the benchmark for further improvements.

First Evaluation

To find areas in which the model performed poorly, the predictions for every failure mode were investigated. The model was tested on the test data set which contains 5000 samples

for each failure mode. The number of correct and false predictions for every failure mode showed in which cases the network didn't perform well.

After the identification of classes with low prediction accuracy, the reason for the underperformance was evaluated. Figure 5-1 shows the prediction for the active failure mode 7. This failure mode was often mistaken for failure mode 4, because these two failure modes have similar impacts on the measurements in the power plant. They both affect the low-pressure section of the HRSG. Their influence as well as the impact of all other failure modes on the process parameters of the CCPP can be seen in the heatmap (Figure Appendix 1 and Figure Appendix 2). This confusion between FM 4 and FM 7 reveals one limitation for the failure mode predictions with NNs. If failures have nearly the same impact and the relative impact is low, the NN may not be able to differentiate between them with high accuracy. Nevertheless, the NN could predict that one of the similar failures was present and hence the location of the failure was limited.

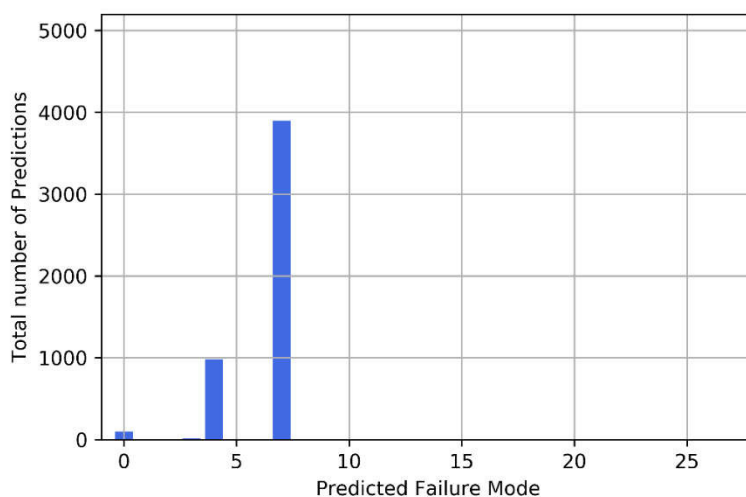


Figure 5-1: Predicted failure modes in 5000 examples for the actual active failure mode 7

The heatmap also shows that some of the failures have a higher impact on the plant operation than others. Failures with a severe influence were detected with higher accuracy and the result vector showed higher probability values. Cases with low impact also lead to weaker probability results and more false predictions. For example, gas turbine filter clogging (failure mode 25) leads to more false predictions than condenser gas accumulation (failure mode 17) which has a greater impact on the input values of the NN.

5.1.1 Optimization

After the first round of training and testing the model, areas of underperformance and the reasons were identified. Subsequently, methods for performance optimization were applied.

These methods were divided into two categories: Data Structure Optimization and Hyperparameter Tuning.

Data Structure Optimization

The neural network highly depends on the quality of the training data. As mentioned before, each failure mode was represented by the same amount of examples in the training data set. In the above-described evaluation, the failure modes with low prediction accuracy were identified. To increase the model performance particularly on these failure modes, more training data with these failures were generated. The new examples were mixed with the old training set. Therefore, a surplus of hard-to-predict failure mode samples was created, to train the model specifically in these areas.

The specification values for all false predictions were plotted (see Figure Appendix 3). An interface in which the investigated failure mode can be chosen was programmed. For the desired failure mode all false predictions under the specification values are displayed. Hence, circumstances that lead to a higher number of false predictions could be identified. The overhead on training data was then created in exactly these areas. For example: If one failure mode showed more false predictions on higher temperatures, then training examples for high temperatures were generated.

Hyperparameter Tuning

Besides the training data for the classification problem, the structure of the neural network and its corresponding hyperparameters were optimized. The performance of combinations of different batch sizes, activation functions, number of hidden layers and neurons within them was evaluated. As it would take enormous amounts of time to configure each combination of hyperparameters per hand and evaluate them afterward, the Scikit-learn library (see [44]) was utilized. This library provides the GridSearchCV class which enables the automated training and testing of all defined hyperparameters and their possible combinations. Table 5-2 shows the tested hyperparameters.

Table 5-2: Hyperparameter test configurations

Hyperparameter	Tested Configurations
Batch Size	1000, 2000, 3000
Activation Functions	ReLU, Sigmoid
One Hidden Layer (Neurons)	(128); (256); (512); (1024)

Two Hidden Layers (Neurons)	(256, 128); (512, 64); (512, 128); (512, 256);
Three Hidden Layers (Neurons)	(64, 64, 64); (128, 64, 32); (128, 64, 64); (128, 128, 128); (100, 80, 50); (512, 256, 128)
Four Hidden Layers (Neurons)	(64, 64, 64, 64)

The GridSearchCV function divided the training data set into five subsets and trained the model in each step on four of the subsets. The remaining set was used to test the performance. This procedure was continued until each set had been used as the test set. For each combination of sets, a test score was calculated. The five test scores were then used to calculate the mean test score for the investigated model. The mean test score is equal to the mean overall accuracy of the model. This procedure was carried out for all mentioned combinations of hyperparameters. Then, the different models were ranked by the mean test score which measures the achieved prediction performance. The five best combinations of hyperparameters are documented in Table 5-3. Each model was trained for 2000 epochs.

Table 5-3: Best hyperparameter combinations

Rank	Hidden Layers	Batch Size	Activation Function	Mean Test Score
1	(512)	2000	ReLU	0.742
2	(1024)	1000	ReLU	0.736
3	(512)	1000	ReLU	0.734
4	(1024)	2000	ReLU	0.731
5	(256)	1000	ReLU	0.721

The best configuration of hyperparameters ranked on number one (One hidden layer with 512 neurons, batch size of 2000 and the ReLU activation function), was used for all further

models. It should be noted that neural networks are probabilistic models and the weights are initialized randomly at the beginning of the training process. Therefore, a variation in the achieved performance occurs after every new training process, even if the same hyperparameters are applied.

With the above-described methods of data structure optimization and hyperparameter tuning the following performance of clean data and noisy data models was achieved.

5.1.2 Clean Data Model

The above-mentioned methods were utilized to build a model that was trained and optimized on clean data. This clean data model was created to evaluate the efficiency of the performance-enhancing methods and to see which prediction performance can be reached.

The neural network structure was set up according to the best-ranked model found with hyperparameter tuning. It was trained on the optimized training data. As the model name indicates, no noise was added to the training data. However, the performance evaluation was carried out on clean and noisy test data with 5000 examples for each failure mode.

Performance on Clean Test Data

Figure 5-2 shows the prediction results of the clean data model on the clean data test set. For every failure mode, the correct predictions (blue bulks) and the false predictions (orange bulks on top of the blue bulks) were plotted as the percentage of the test examples.

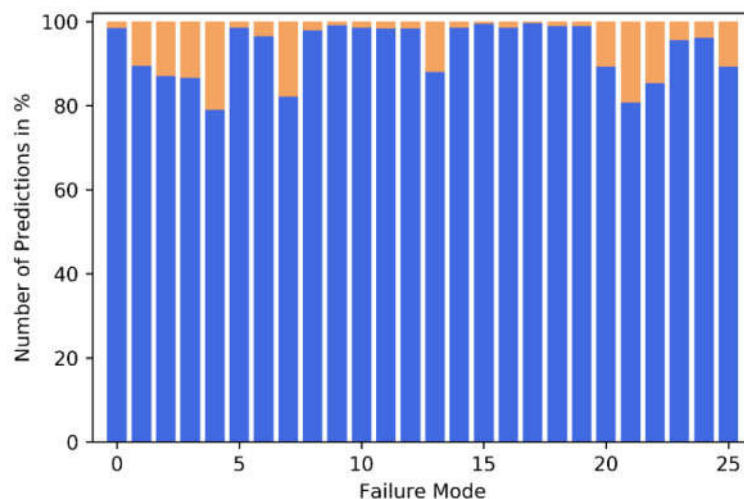


Figure 5-2: Correct(blue) and False(orange) predictions for every class; Clean data model and clean test data

The model achieved an overall accuracy of 93.4 %, an overall precision of 94.9 % and an overall sensitivity of 93.4 %. Table Appendix 1 shows the precision and sensitivity values for each failure mode. The model performed very well at most classifications, except for failure

mode 0 (normal mode). Although failure mode 0 was predicted correctly in 98.5 % of the cases, its precision only reached 53.1 % (see Table Appendix 1). The low precision value is caused by test examples of other failure modes with a low failure impact. These examples lead to a high number of false positives for failure mode 0 (see 4.5.3). This result indicates that failure modes are more effectively detected when they reach a higher level of impact on the power plant operation and therefore on the input values of the NN. To prove this assumption, the model was tested again on 5000 examples for each class, but with failure impact values from 5.0 % to 100 %. The resulting correct and false predictions are shown in Figure 5-3.

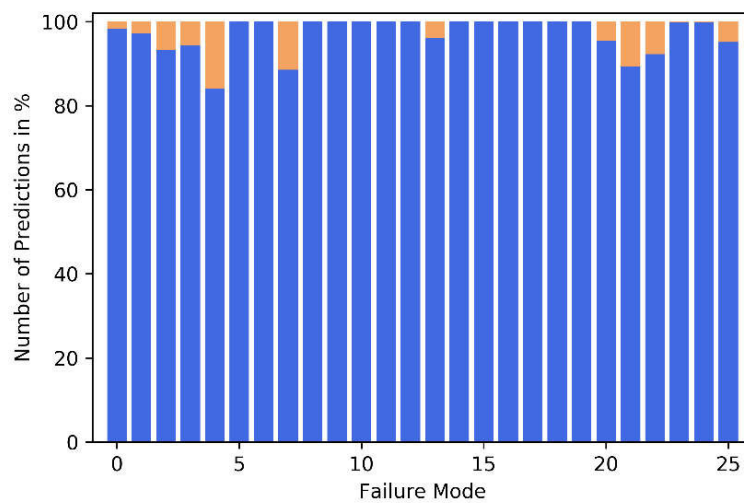


Figure 5-3: Correct(blue) and False(orange) predictions for every class; Clean data model and clean test data with FI > 5%

As expected the precision for failure mode 0 increased. It grew from the previously achieved 53.1 % to 86.3 %, because of fewer false positives for this class caused by excluding samples with a FI greater than 5 %. Samples with a low FI are more often mistaken for a plant operation without any failure and hence lead to false positives for failure mode 0. Besides the precision for failure mode 0, the overall accuracy (97.1 %), overall precision (97.2 %) and the overall sensitivity (97.1 %) were improved. Table Appendix 2 shows the results of the test with failure impact values above 5.0 %.

The improvement in performance parameters shows that the probability to detect a failure mode increases with its failure impact.

Performance on Noisy Test Data

It could be shown that the clean data model performs well on clean test data. Subsequently, this model was tested on noisy test data, to simulate conditions under real-life applications

with random measurement errors. The noise factor was set to 1.0, hence errors in the range of Table 4-3 have been applied to the test data. Figure 5-4 shows the prediction results for each failure mode of the noisy test data set.

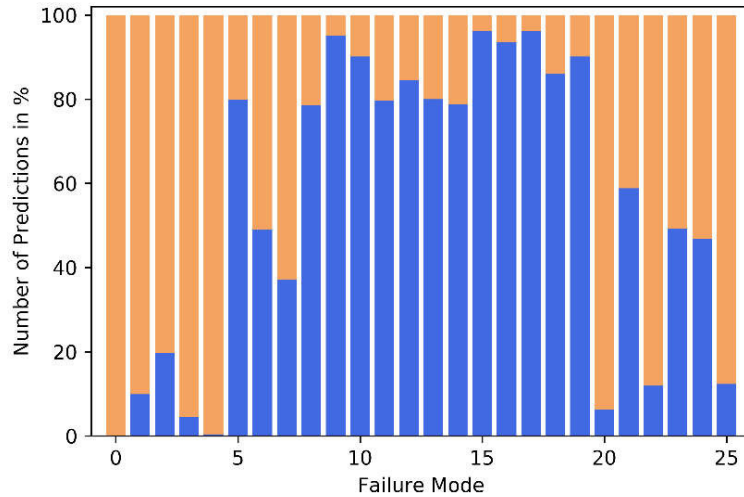


Figure 5-4: Correct(blue) and False(orange) predictions for every class; Clean data model and noisy test data

The model achieved an overall accuracy of 55.2 %, an overall precision of 51.5 % and an overall sensitivity of 55.2 %. This means a decrease in accuracy of 38.2 %, a decrease in precision of 43.4 % and a decrease in sensitivity of 38.2 % regarding the results on clean test data. The performance parameters for each class are documented in Table Appendix 3. This result shows that the clean data model was fit perfectly to the clean training data, but couldn't handle noisy measurements. For example, failure mode 0 wasn't even correctly identified once because the model is very sensitive to the small deviations in the input data created by the noise. Therefore, failures were predicted when failure mode 0 was the actual class. Figure 5-5 shows the predicted classes when failure mode 0 was present.

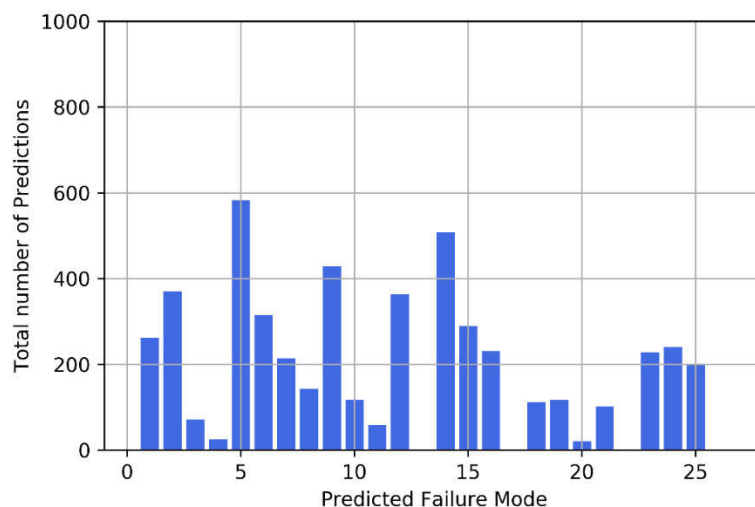


Figure 5-5: Predictions for an actual failure mode 0

In conclusion, the clean data model can't be used for real-life applications because of the occurring measurement errors and the overfitting of the model on clean data. Hence, the noisy data model was built.

5.1.3 Noisy Data Model

The noisy data model was trained on a data set with an applied noise factor of 1.0. The network structure was set up according to the best model found with hyperparameter tuning (see 5.1.1). The following results were achieved on a test data set with a noise factor of 1.0, except for the evaluation regarding the influence of different noise factors on the precision of the model. Figure 5-6 shows the correct and false predictions for each failure mode.

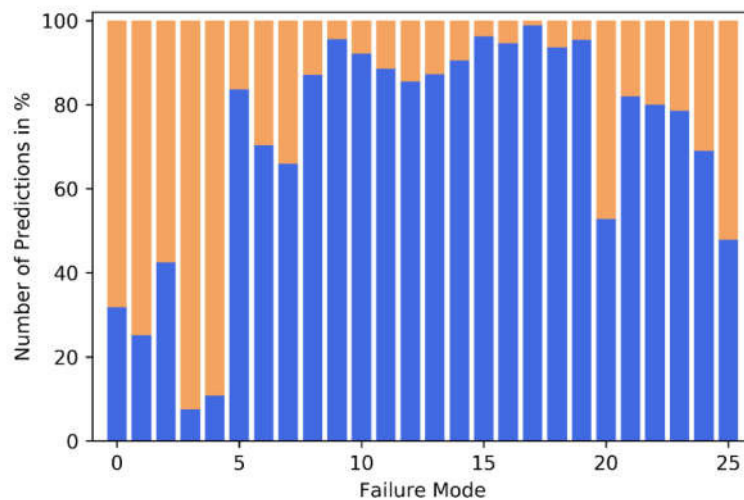


Figure 5-6: Correct(blue) and False(orange) predictions for every class; Noisy data model and noisy test data

An overall accuracy of 71.3 %, an overall precision of 71.2 % and an overall sensitivity of 71.3 % was achieved. The noisy data model performed much better on the noisy data set than the clean data model, showing an increase in accuracy of 16.1 %, an increase in precision of 19.7 % and an increase in sensitivity of 16.1 %. This model was more tolerant of random errors because it was already trained on data with noise. Nevertheless, failure modes with a small impact on the plant operation, like FM 1, 2, 3, 4, 20 and 25, were still detected with low precision and sensitivity. This was caused by the noise which leads to deviations near the maximum impact of these failure modes. Hence confusion in the prediction between the mentioned failure modes occurred. For example, failure mode 1 was often mistaken for others (Figure 5-7), because of its small relative impact on the input values of the network.

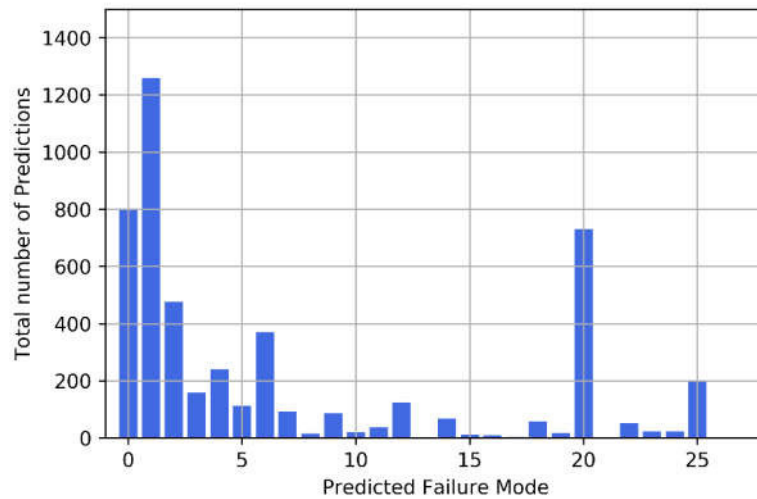


Figure 5-7: Predictions for an actual failure mode 1

Effect of Setting a Prediction Threshold

For further improvements, a threshold for the prediction probabilities was introduced. It should be noted that for the previously shown results, the threshold was set globally at the same level for every failure mode classification. Every result with a probability below the threshold value was mapped to failure mode 0, as explained in 4.5.3. The performance parameters were directly affected by the threshold. The resulting impact on the precision and sensitivity is shown in Figure 5-8 on the example of failure mode 6. The threshold reduced the number of true positive predictions but decreased the number of false-positive predictions stronger, leading to increased precision and a decreased sensitivity.

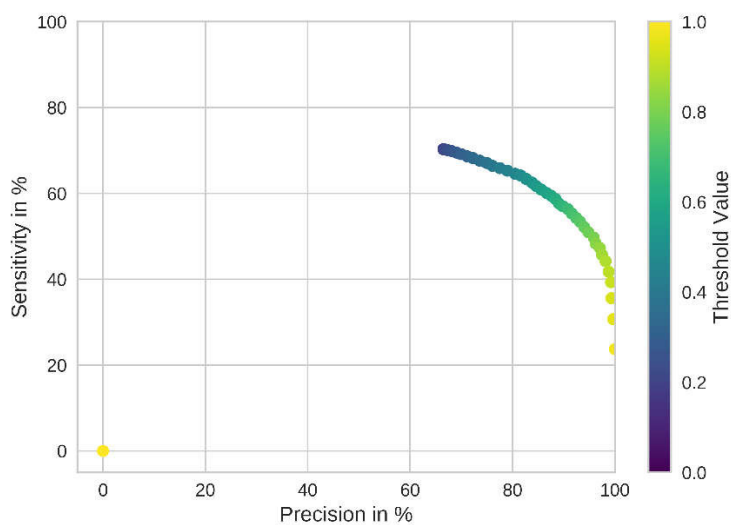


Figure 5-8: Dependence of sensitivity and precision on the threshold value

As mentioned before, the goal was to build a network with high **precision** for detecting failures. Therefore, the threshold value was tuned to achieve high precision for the cost of lower sensitivity. A precision value of 80.0 % and above, while keeping the number of TP predictions also over 80.0 % of the level with no threshold and hence providing good sensitivity, embodies an acceptable network performance.

The mapping of results below the threshold towards FM 0 reduced the precision for detecting FM 0 and the overall accuracy. The decreased precision for the FM 0 class is not a problem because the goal is to predict failures and not the normal plant operation. Hence the performance of the model was measured on the overall precision, excluding FM 0.

The increased precision for FM 6, shown in Figure 5-8 is caused by a reduction in false-positive results for this class. But with a higher threshold value, the true-positive results also decline (see Figure 5-9). Therefore, the threshold shouldn't be set too high, to keep the number of true positive predictions at an acceptable level.

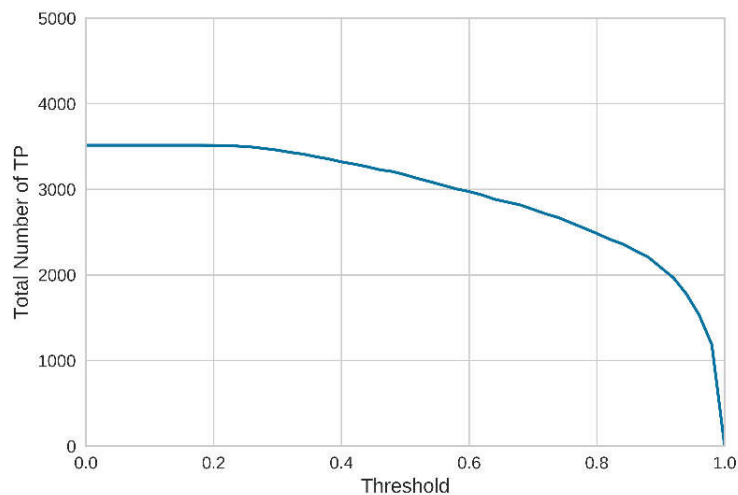


Figure 5-9: Number of TP predictions for failure mode 6 as a function of the threshold

A high precision value doesn't indicate good performance if the number of true positive values falls to near zero. This shows the importance of observing the declining number of true-positives together with the improvement in precision to find a suitable threshold value.

To evaluate the influence of the threshold on the model performance, the precision of the model was plotted as a function of the threshold value for test data with a noise factor of 1.0 (see Figure 5-10). Results for FM 0 were excluded.

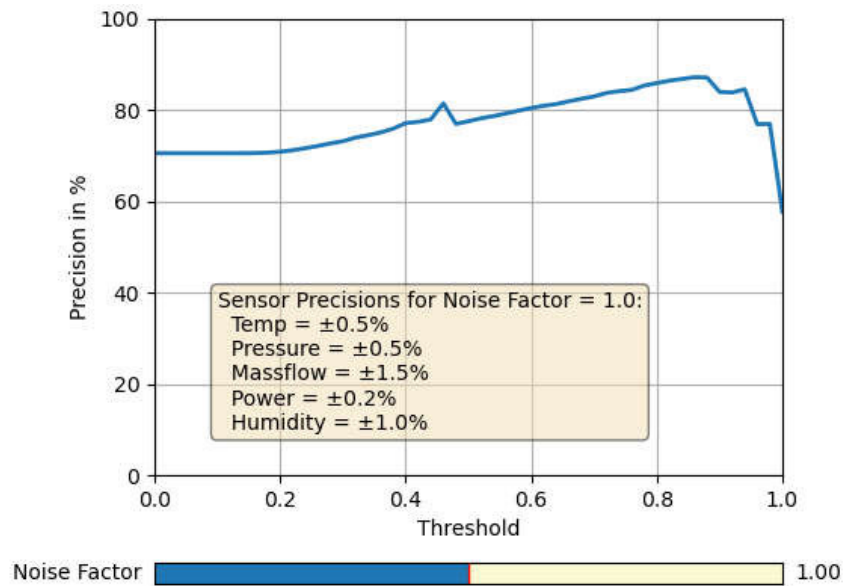


Figure 5-10: Overall precision without FM 0 for a noise factor of 1.0

It was found that precision can be significantly improved by setting the threshold high enough. This leads to the conclusion that predictions with high probability values also result more likely in true positives. Therefore the probability output of the model is not only a numerical result but also an indicator for the actual probability for the occurrence of a failure.

The small drop at a threshold of 0.4 is caused by the fewer number of true positives. The reduction of true positive predictions mainly affects the failure modes with a low impact on the input values of the NN. Additionally to the overall precision the total number of true positive predictions for all failure modes, excluding failure mode 0, is plotted (see Figure 5-11). This graphic shows the discussed effect of a higher threshold on the true positive predictions for all classes.

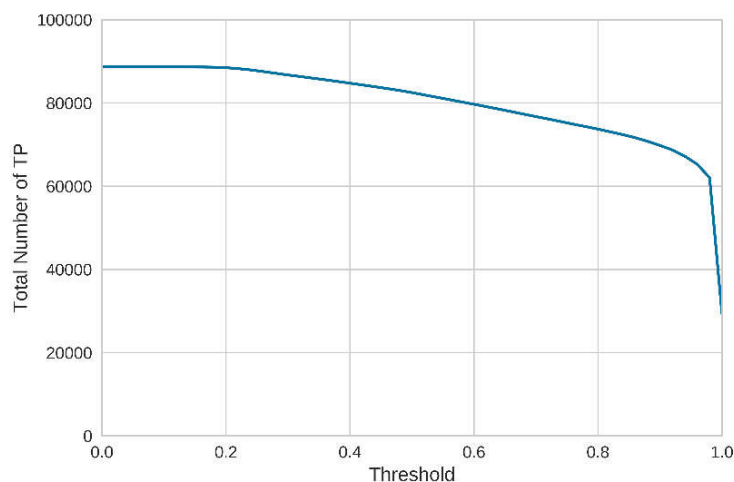


Figure 5-11: Number of overall TP predictions as a function of the threshold without FM0

Impact of Different Noise Levels

The noisy data model was tested on different noise levels to investigate the influence of measurement accuracy on the models' precision. Figure 5-12 shows the precision for a noise factor of 0.5 and Figure 5-13 the precision for a noise factor of 2.0.

A higher noise factor leads to more false-positive predictions between the failure modes and therefore to less precise results. With the high noise factor, failure modes with a small impact on the plant operation couldn't be predicted well. Hence a decrease in the overall precision occurred. Failures with a severe impact on the power plant and a high probability results were still predicted precisely, which lead to increasing overall precision at higher threshold values.

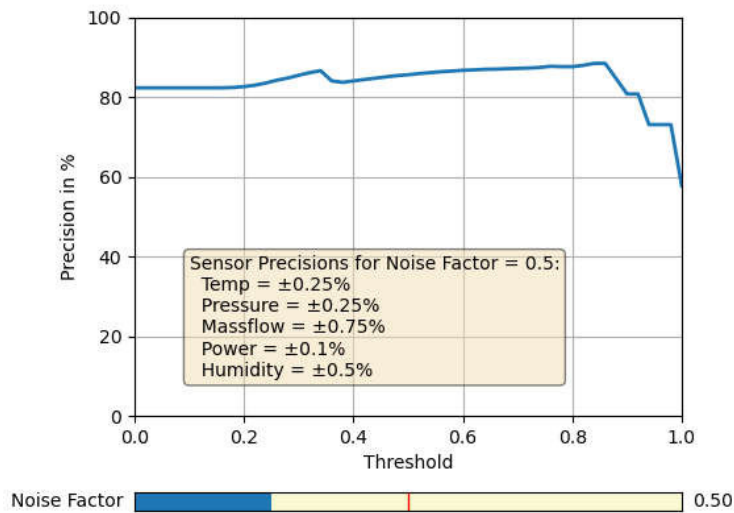


Figure 5-12: Overall precision without FM 0 for a noise factor of 0.5

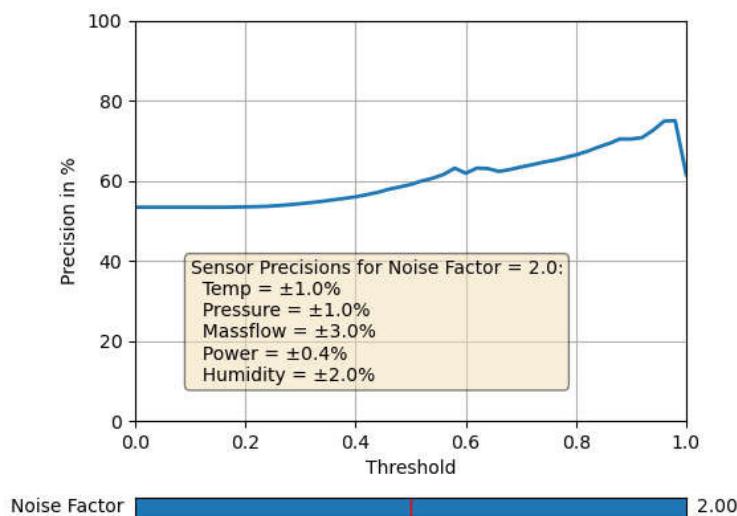


Figure 5-13: Overall precision without FM 0 for a noise factor of 2.0

5.1.4 Test on Two Active Failure Modes

The results above showed the performance of the model for detecting one activated failure mode at a time. The following evaluation deals with the response of the noisy data model on two activated failures. The possibility to detect more failures at once with a model that was trained only on single failures was tested. For this evaluation certain combinations of failure modes were chosen, as it was impossible to test all combinations of the 26 classes in a reasonable time.

The performance of the model on two failure modes is shown in the example of GT compressor fouling (FM 21) and GT turbine fouling (FM 23). These two specific failure modes were chosen because they have a similar impact on the plant operation (see Figure Appendix 1). This test should show if the network can detect both failure modes simultaneously or if the network only predicts one of them. The specification values for the generated test data set were set according to Table 5-1. The only difference to the examples above was the activation of a second failure mode and a corresponding failure impact. The failure impacts for the two failures were set according to the distribution explained in 4.4.

Performance on Clean Test Data

Initially, the model was tested on clean data. Figure 5-14 shows the results of a prediction with a failure impact of 9.73 % for FM 21 and a failure impact of 14.43 % for FM 23.

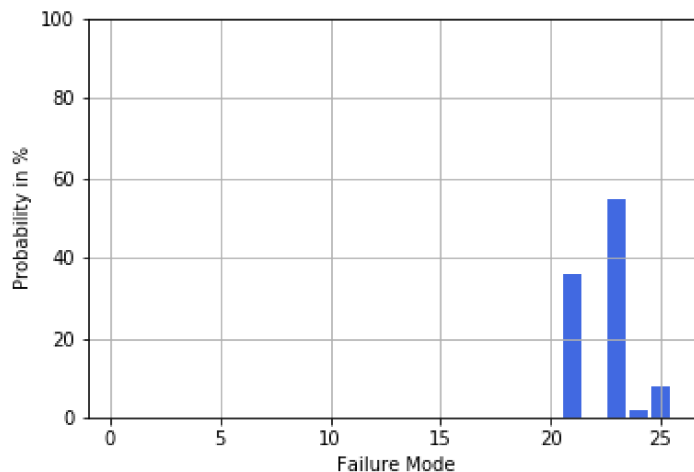


Figure 5-14: Predicted probabilities for the actual active classes FM 21 (FI = 9.73%) and FM 23 (FI = 14.43%)

Although the failure impacts were set to a low level, the two active failures were predicted correctly. The failure with the higher FI value was predicted with a higher probability, which presents a beneficial behavior since the failure with the highest impact should be detected

first to enable a fast intervention in real-life applications. Besides the active failure modes, FM 25 and FM 24 were incorrectly predicted, but with a small probability.

Figure 5-15 shows that higher failure impacts values (FI 21 = 17.91 % and FI 23 = 40.12 %) lead to more precise predictions. In this case, the probability values for other classes were negligible.

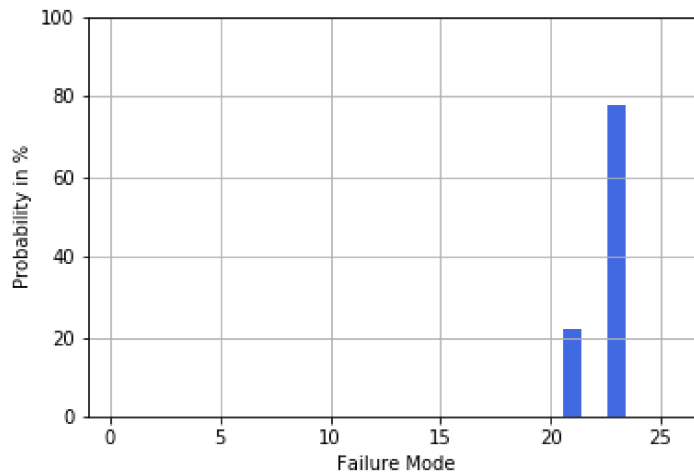


Figure 5-15: Predicted probabilities for the actual active classes FM 21 (FI = 17.91%) and FM 23 (FI = 40.12%)

In Figure 5-16 the total number of predictions for each failure mode for all 5000 test examples is plotted. The majority of predictions resulted in the actual classes FM 21 or FM 23. Besides the actuals failure modes, FM 25 was predicted in a few cases in which both failure impacts FI 21 and FI 23 were on a low level.

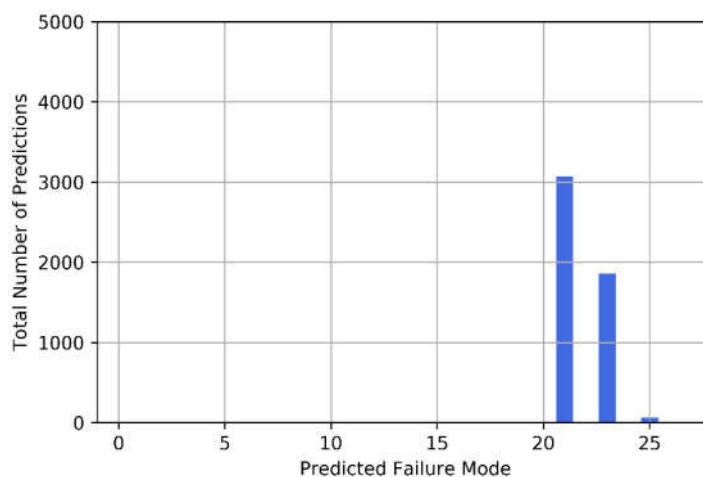


Figure 5-16: Predicted failure modes for the actual active classes FM 21 and FM 23 on clean test data

Additional to the combination of FM 21 and FM 23 which show similar impacts on the input value of the NN, the following combinations were tested:

- FM 5 (HRSG Fouling HP) and FM 18 (Condenser Fouling/Scaling)
- FM 9 (Letdown Leakage HRH-Condenser) and FM 22 (GT Compressor Tip Clearance)
- FM 17 (Condenser Gas Accumulation) and FM 21 (GT Compressor Fouling)

The total number of predictions for these combinations is shown in Figure Appendix 4, Figure Appendix 5 and Figure Appendix 6. The failure modes which were predicted with more precision in the single failure mode evaluation were also detected more often when two failure mode were activated.

Performance on Noisy Test Data

Additionally to the clean test data results, the performance of the model was tested on a noisy test data set with a noise factor of 1.0. Again FM 21 and FM 23 were chosen to enable a direct comparison with the results on clean test data. The noise for all measurements was applied randomly as discussed in 4.4.3. Therefore, only the total number of predictions for all examples was plotted (see Figure 5-17).

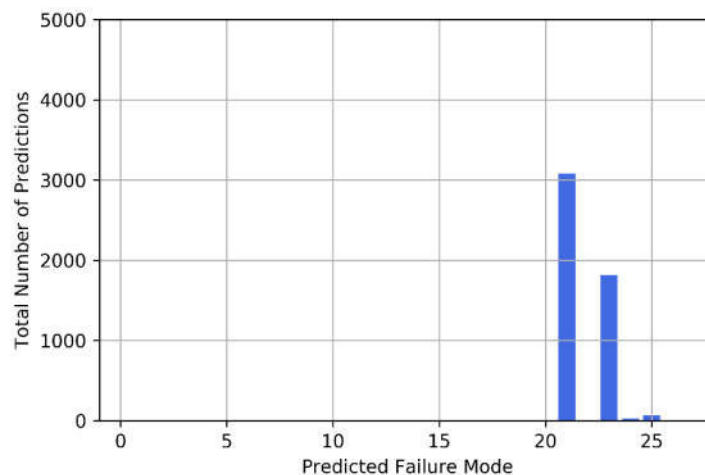


Figure 5-17: Predicted failure modes for the actual active classes FM 21 and FM 23 on noisy test data

The evaluation of single prediction examples wouldn't show the overall impact of noise on the model performance, because the value of noise differs between all test samples.

Additionally to the total number of predictions, the overall predicted probability for the failure modes was investigated. Figure 5-18 shows that the actual failure modes 21 and 23 were predicted with a high probability.

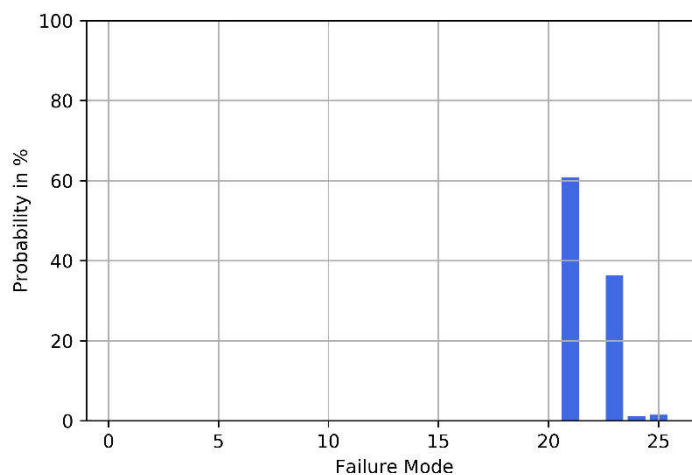


Figure 5-18: Mean overall probability for every failure mode for actual active classes FM 21 and FM 23 on noisy test data

The applied noise on the data set only showed a small effect on the performance of the model. The activated failure modes were predicted in most cases correctly and the number of false predictions only slightly increased. This leads to the conclusion that the combined effect of the two failure modes on the plant operation outweighs the disruption of the applied noise to a certain extent.

This evaluation of the noisy data model on two failure modes indicates that a network that was trained on single activated failure modes is also able to predict at least two failure modes at a time.

6 CONCLUSIONS

In this work, a failure detecting neural network for a CCPP was created based on simulated process data. The model shows the potential of automated failure prediction from process parameters in CCPPs. The trained neural network is capable of recognizing various degradation failure modes by identifying their characteristic impact on the measured process parameters in the plant.

The utilized performance-enhancing methods showed a significant impact on the models' prediction capabilities and thus represent a valuable tool for optimizing neural networks. The resulting model achieved high prediction precision for the majority of the investigated failure modes. Degradation failures with a high impact on process parameters were expectedly recognized with higher precision than low impact failures.

Additionally, limitations in prediction accuracy caused by measurement noise were analyzed to evaluate the potential of an application of neural networks derived from "clean" simulated data in real-life applications. The applied random errors lowered the prediction capabilities especially for failures that have a low impact on the plant operation. In conclusion, the measurement noise of sensors in real-life applications requires applying noise to the simulated training data as well. Otherwise, overfitting to clean simulated data occurs which decreases the prediction performance.

Although the neural network was trained on examples with one activated failure mode at a time only, the model was also able to detect multiple simultaneously activated failures. Its capability to perceive at least two present failure modes was demonstrated for combinations of two selected failure modes.

In conclusion, neural networks based on simulated process data provide a promising opportunity for automated failure predicting systems in the future.

7 OUTLOOK

The introduced workflow using Python scripts and the described methods in the field of artificial neural networks may serve as a basis for the future development of applications for failure identification and detection under real plant operating conditions. The statistical evaluation methods of the NN model performance are suitable for benchmarking upcoming failure mode predicting neural networks in CCPP applications.

As the next stage of development, the presented methods shall be tested on existing power plants. This requires the generation of thermodynamical models of selected plants which must be capable to reflect the plants operating characteristics. Although the process of modeling real plants, failure modes, generating training data and creating neural networks follows the steps described in this work, several additional steps of preprocessing online measured plant data will be necessary before such data can be successfully used for failure detection. For example, systematic measurement errors need to be taken care of, e.g. through data reconciliation techniques and steady-state criteria which must be defined to assure stable plant operation during evaluation periods. Additionally, random errors shall be minimized by applying adequate averaging and outlier treatment processes for every measurement.

For further improvement of the prediction accuracy, an individual threshold for every investigated failure mode may be applied. Optimal threshold values for every class will help operators to recognize the right moment of intervention to initiate maintenance activities.

BIBLIOGRAPHY

- [1] UNITED NATIONS FRAMEWORK CONVENTION ON CLIMATE CHANGE: *Paris Agreement - Status of Ratification*. URL <https://unfccc.int/process/the-paris-agreement/status-of-ratification> – Überprüfungsdatum 2021-01-08
- [2] McCLINTOCK, Michael ; CRAMLITT, Kenneth L.: Combined Cycle Plant Performance Monitoring. In: *ASME Power Conference (2004)*, S. 291–294
- [3] SHI, Licheng ; LONG, Yun ; WANG, Yuzhang ; CHEN, Xiaohu ; ZHAO, Qunfei: *On-line detection of porosity change of high temperature blade coating for gas turbine*. In: *Infrared Physics & Technology* 110 (2020)
- [4] STEAG ENERGY SERVICES GMBH: *EBSILON®Professional heat balance software*. URL www.ebsilon.com – Überprüfungsdatum 2021-01-08
- [5] JETBRAINS: *PyCharm*. URL <https://www.jetbrains.com/pycharm/> – Überprüfungsdatum 2021-02-03
- [6] PROJECT JUPYTER: *JupyterLab*. URL <https://jupyter.org> – Überprüfungsdatum 2021-02-03
- [7] GOOGLE LLC: *Tensorflow*. URL <https://www.tensorflow.org> – Überprüfungsdatum 2021-02-03
- [8] MIT: *Keras*. URL <https://keras.io> – Überprüfungsdatum 2021-02-03
- [9] GÜLEN, S. Can: *Gas Turbine Combined Cycle Power Plants*. Boca Raton, FL : CRC Press, 2020
- [10] BOYCE, M. P.: *Combined cycle power plants*. 2012
- [11] BOYCE, Meherwan P.: *Handbook for Cogeneration and Combined Cycle Power Plants*. 2nd ed. New York, NY : ASME Press, 2010
- [12] KEHLHOFER, Rolf ; HANNEMANN, Frank ; STIRNIMANN, Franz ; RUKES, Bert: *Combined-cycle gas & steam turbine power plants*. 3. ed. Tulsa, Okla. : PennWell, 2009
- [13] GÜLEN, S. Can: *Gas turbines for electric power generation*. Cambridge, New York, NY : Cambridge University Press, 2019
- [14] TAHAN, Mohammadreza ; TSOUTSANIS, Elias ; MUHAMMAD, Masdi ; ABDUL KARIM, Z. A.: *Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review*. In: *Applied Energy* 198 (2017), S. 122–144
- [15] MOHAMMED, Mohammed: *EXERGOECONOMIC ANALYSIS AND OPTIMIZATION OF COMBINED CYCLE POWER PLANTS WITH COMPLEX CONFIGURATION*. 2015

- [16] NOORDERMEER, Jim: *Understanding Gas Turbine Performance*
- [17] INTERNATIONAL ENERGY AGENCY: *World gross electricity production, by source*. URL <https://www.iea.org/data-and-statistics/charts/world-gross-electricity-production-by-source-2018> – Überprüfungsdatum 2021-02-03
- [18] THOMAS W. OVERTON: *A Primer on Gas Turbine Failure Modes*. URL <https://www.powermag.com/primer-gas-turbine-failure-modes/> – Überprüfungsdatum 2021-01-08
- [19] HOPSON, Warren: *Finding and fixing leakage within combined HP-IP steam turbines: Part I*. URL <https://www.powermag.com/finding-and-fixing-leakage-within-combined-hp-ip-steam-turbines-part-i/> – Überprüfungsdatum 2021-01-08
- [20] RADIN, Yu. A. ; GRISHIN, I. A.: *Axial Thrust Balancing in High-Temperature Cylinders of Steam Turbines during Transients in Combined-Cycle Units*. In: *Thermal Engineering* 66 (2019), Nr. 6, S. 409–414
- [21] TETRA ENGINEERING GROUP™: *HRSB Tube Failure Statistics*. URL <https://tetra-eng.com/resources/technical-white-papers/30-white-papers/177-hrsg-tube-failure-statistics> – Überprüfungsdatum 2021-01-08
- [22] BAIN, Douglas ; CHRISTOPHERSEN, David: *Some Common Mechanisms Leading to Failures in Heat Recovery Steam Generators*
- [23] HELALI, A. B.: *Effects of water contamination on sub-cooled flow boiling heat transfer*. In: *Energy Conversion and Management* 52 (2011), Nr. 5, S. 2288–2295
- [24] RAO, A. D.: *Natural gas-fired combined cycle (NGCC) systems*. In: *Combined Cycle Systems for Near-Zero Emission Power Generation* (2012), S. 103–128
- [25] MOORE, William: *Power station condensers their design and failure modes*. In: *Materials at High Temperatures* 34 (2017), 5-6, S. 407–414
- [26] DING, Hongbing ; LI, Yiming ; LAKZIAN, Esmail ; WEN, Chuang ; WANG, Chao: *Entropy generation and exergy destruction in condensing steam flow through turbine blade with surface roughness*. In: *Energy Conversion and Management* 196 (2019), S. 1089–1104
- [27] ZACHARY, Justin: *The long and short of last-stage blades*. URL <https://www.powermag.com/the-long-and-short-of-last-stage-blades/> – Überprüfungsdatum 2021-01-08
- [28] TANUMA, Tadashi: *Advances in Steam Turbines for Modern Power Plants*. Kent : Elsevier Science, 2016

- [29] WAN, Zhong-hai ; CAI, Wen ; YAN, Tao ; CHEN, Wen ; YANG, Hong-yan ; LU, Jin: *Calculation method of back pressure correction coefficient of steam turbine*. In: *IOP Conference Series: Earth and Environmental Science* 153 (2018)
- [30] MICHELS, H. T. ; KIRK, W. W. ; TUTHILL, A. H.: *The influence of corrosion and fouling on steam condenser performance*. In: *Journal of Materials for Energy Systems* 1 (1979), Nr. 3, S. 14–33
- [31] WESTECH ENGINEERING, INC.: *Condenser Tube Cleaning System*. URL <https://www.westech-inc.com/products/condenser-tube-cleaning-system> – Überprüfungsdatum 2021-01-08
- [32] EGUIA, E. ; VIDART, T. F. ; BEZANILLA, J. A. ; AMIEVA, J. J. ; OTERO, P. M.: *Elimination of biofouling in heat exchangers-condensers by different chemical methods* (1998). URL <https://www.witpress.com/Secure/elibrary/papers/MAR98/MAR98028FU.pdf> – Überprüfungsdatum 2021-01-08
- [33] ENGINEERS EDGE: *Large Steam System Condenser*. URL https://www.engineersedge.com/heat_exchanger/large_steam_condenser.htm – Überprüfungsdatum 2021-01-08
- [34] *Power plant surface condenser*. URL https://en.wikipedia.org/wiki/File:Power_plant_surface_condenser.jpg – Überprüfungsdatum 2021-02-07
- [35] NEAPOLITAN, Richard E. ; JIANG, Xia: *Artificial Intelligence: With an Introduction to Machine Learning, Second Edition* : CRC Press, 2018
- [36] GOODFELLOW, Ian ; BENGIO, Yoshua ; COURVILLE, Aaron: *Deep Learning* : <http://www.deeplearningbook.org> : MIT Press, 2016
- [37] SHARMA, Abhishek: *What is the differences between artificial neural network (computer science) and biological neural network?* URL https://www.engineersedge.com/heat_exchanger/large_steam_condenser.htm – Überprüfungsdatum 2021-01-08
- [38] VIEIRA, Sandra ; PINAYA, Walter H. L. ; MECHELLI, Andrea: *Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications*. In: *Neuroscience and biobehavioral reviews* (2017), S. 58–75
- [39] RHYS, Hefin I.: *Machine Learning with R, the tidyverse, and mlr*. Shelter Island, NY : Manning publications, 2020

- [40] KRÜGER, Frank: *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models*. 2016
- [41] PECHTL, Peter ; SCHEINECKER, Christian ; PETEK, Josef: Enhancing Prediction Accuracy in the Evaluation of Power Plant Uprates Utilizing a Novel 'Big Data' Approach. In: *Volume 2: Heat Exchanger Technologies; Plant Performance; Thermal Hydraulics and Computational Fluid Dynamics; Water Management for Power Systems; Student Competition* : American Society of Mechanical Engineers, 06242018
- [42] WIKIPEDIA: *TensorFlow*. URL <https://en.wikipedia.org/wiki/TensorFlow> – Überprüfungsdatum 2021-01-08
- [43] WIKIPEDIA: *Keras*. URL <https://en.wikipedia.org/wiki/Keras> – Überprüfungsdatum 2021-01-08
- [44] WIKIPEDIA: *scikit-learn*. URL <https://en.wikipedia.org/wiki/Scikit-learn> – Überprüfungsdatum 2021-01-08

APPENDIX

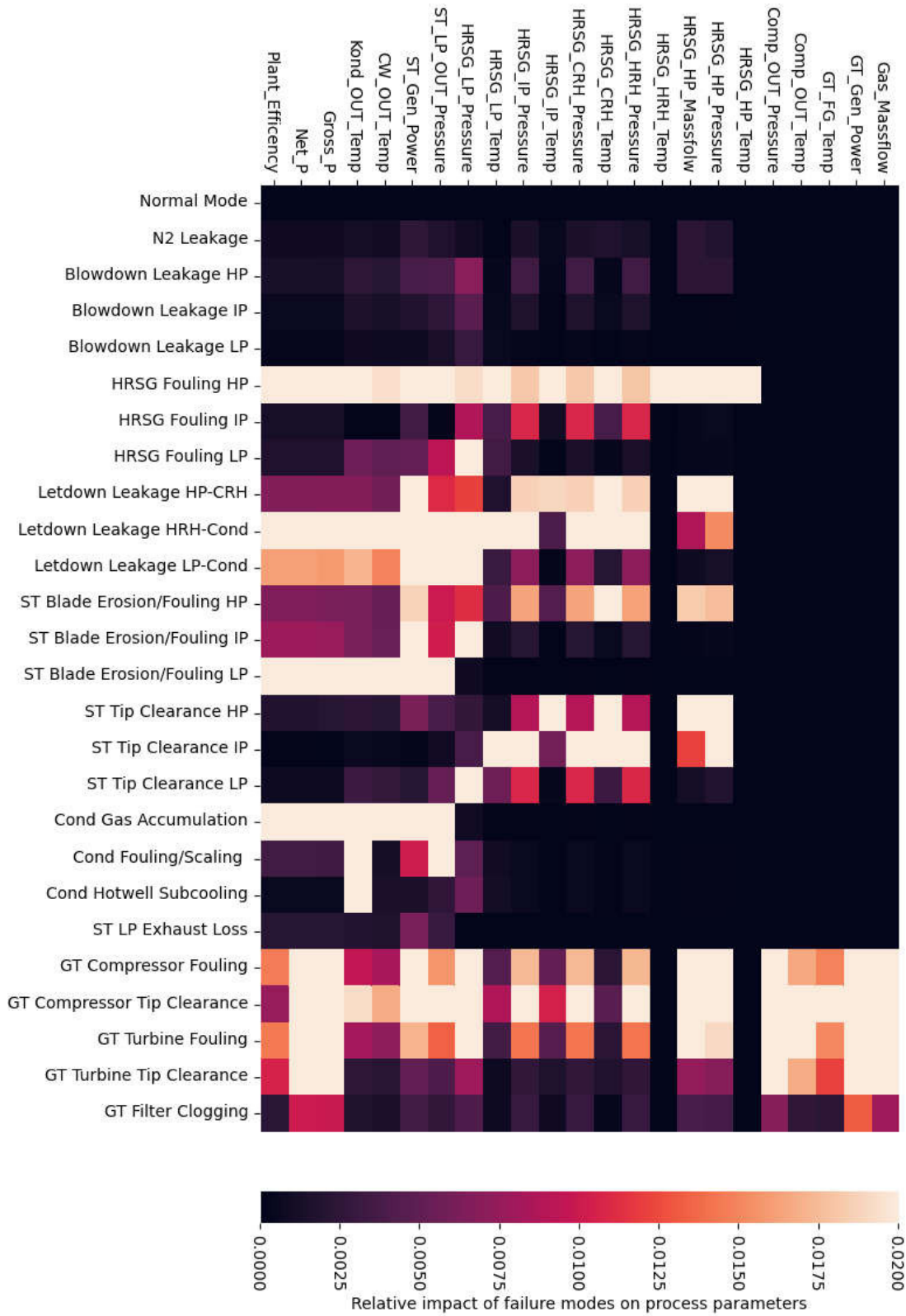


Figure Appendix 1: Heatmap for the relative impact (scale up to 2%) of every failure mode on the process parameters



Figure Appendix 2: Heatmap for the relative impact (scale up to 15%) of every failure mode on the process parameters

Table Appendix 1: Result of clean data model and clean test data

Failure Mode	Precision	Sensitivity
0	0.531	0.985
1	0.987	0.894
2	0.986	0.870
3	0.917	0.866
4	0.774	0.790
5	0.999	0.986
6	0.997	0.965
7	0.901	0.821
8	0.999	0.979
9	1.000	0.991
10	1.000	0.986
11	0.996	0.983
12	1.000	0.984
13	0.976	0.881
14	0.998	0.985
15	1.000	0.994
16	1.000	0.985
17	1.000	0.996
18	1.000	0.989
19	1.000	0.989

20	0.885	0.892
21	0.967	0.808
22	0.974	0.854
23	0.997	0.956
24	0.993	0.962
25	0.806	0.892
Overall	0.949	0.934

Table Appendix 2: Result of clean data model and clean test data (FI>5%)

Failure Mode	Precision	Sensitivity
0	0.863	0.984
1	0.998	0.973
2	0.989	0.933
3	0.925	0.944
4	0.845	0.841
5	0.999	1.000
6	1.000	1.000
7	0.880	0.886
8	1.000	1.000
9	1.000	1.000
10	1.000	1.000
11	1.000	1.000

12	1.000	1.000
13	0.969	0.961
14	1.000	1.000
15	1.000	1.000
16	0.999	1.000
17	1.000	1.000
18	1.000	1.000
19	1.000	1.000
20	0.958	0.954
21	0.962	0.893
22	0.981	0.923
23	0.998	0.998
24	0.997	0.999
25	0.897	0.951
Overall	0.972	0.971

Table Appendix 3: Result of clean data model and noisy test data

Failure Mode	Precision	Sensitivity
0	0.000	0.000
1	0.175	0.099
2	0.197	0.196
3	0.137	0.045

4	0.117	0.004
5	0.409	0.798
6	0.479	0.491
7	0.383	0.371
8	0.640	0.785
9	0.493	0.950
10	0.639	0.901
11	0.786	0.797
12	0.517	0.846
13	0.967	0.800
14	0.458	0.788
15	0.645	0.963
16	0.607	0.936
17	1.000	0.961
18	0.797	0.860
19	0.822	0.901
20	0.490	0.063
21	0.672	0.588
22	0.966	0.119
23	0.401	0.492
24	0.375	0.468
25	0.209	0.123

Overall	0.515	0.552
---------	-------	-------

Table Appendix 4: Result of noisy data model and noisy test data

Failure Mode	Precision	Sensitivity
0	0.192	0.318
1	0.273	0.252
2	0.359	0.425
3	0.218	0.075
4	0.177	0.108
5	0.778	0.836
6	0.665	0.703
7	0.571	0.660
8	0.897	0.871
9	0.953	0.955
10	0.956	0.921
11	0.881	0.885
12	0.835	0.855
13	0.977	0.872
14	0.816	0.905
15	0.967	0.962
16	0.956	0.946
17	0.996	0.989

18	0.940	0.937
19	0.975	0.954
20	0.448	0.528
21	0.870	0.820
22	0.848	0.801
23	0.721	0.786
24	0.808	0.690
25	0.445	0.478
Overall	0.712	0.713

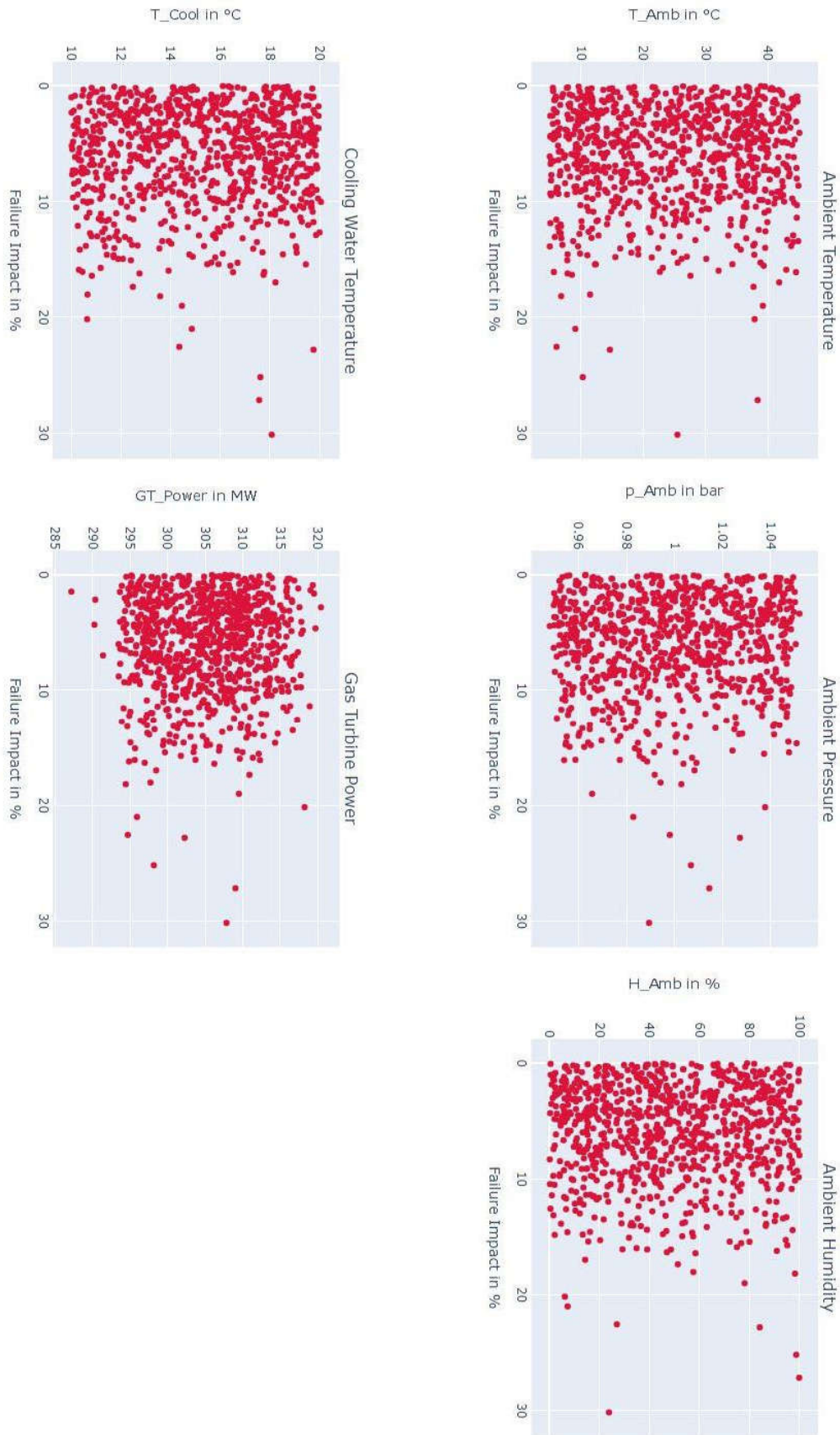


Figure Appendix 3: Specification values of false predicted examples for failure mode 21

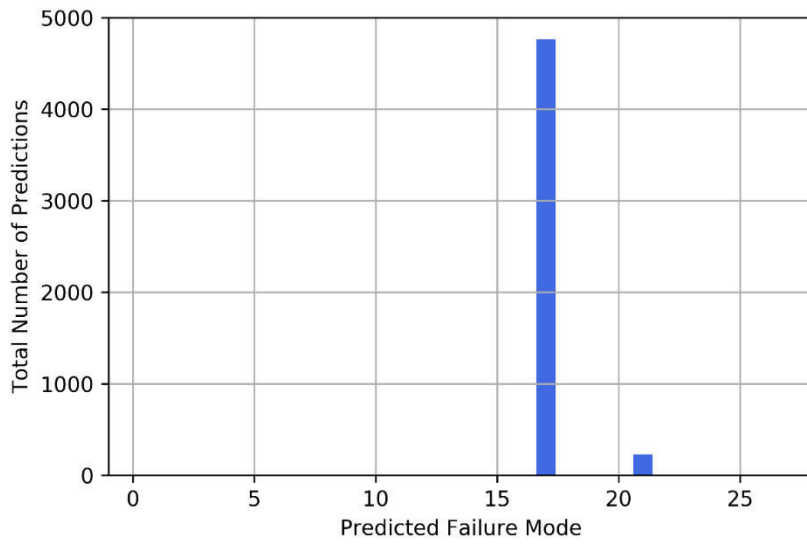


Figure Appendix 4: Predicted failure modes for the actual active classes FM 17 and FM 21 on clean test data

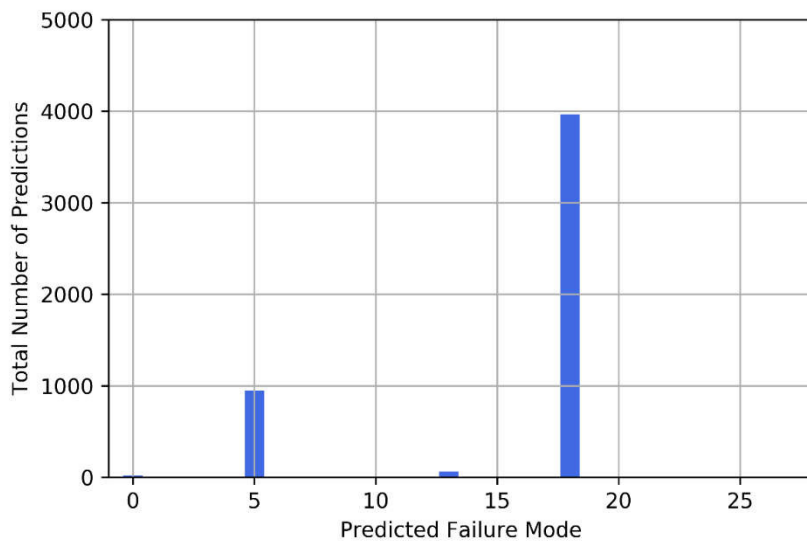


Figure Appendix 5: Predicted failure modes for the actual active classes FM 5 and FM 18 on clean test data

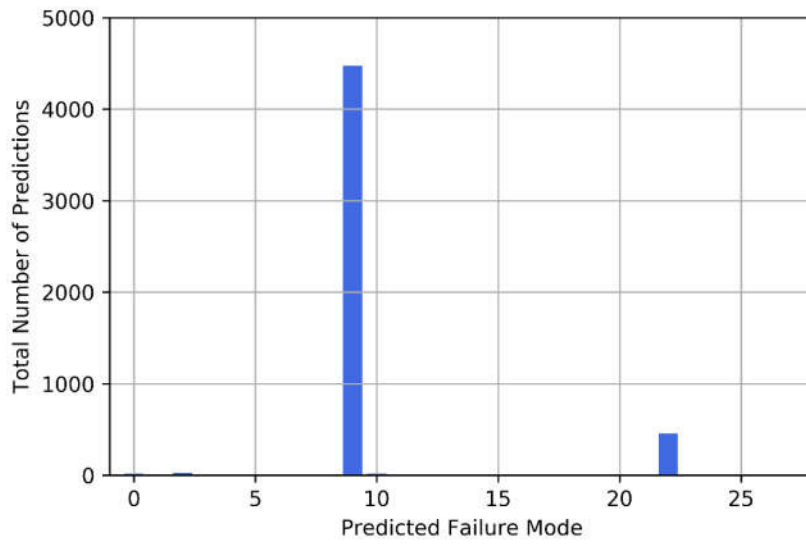


Figure Appendix 6: Predicted failure modes for the actual active classes FM 9 and FM 22 on clean test data