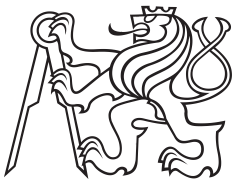
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Measurement

Deep digital twin for prognostics and health management of electromechanical actuators

Bc. Milan Zongor

Supervisor: prof. Ing. Radislav Šmíd, Ph.D.
Field of study: Cybernetics and robotics
Subfield: Cybernetics and robotics
January 2022

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Zongor Milan** Personal ID number: **465883**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Deep digital twin for prognostics and health management of electromechanical actuators

Master's thesis title in Czech:

Digitální dvojče s DNN pro prognostiku a management stavu elektromechanických aktuátorů

Guidelines:

1. Study the deep digital twin architecture for prognostics and health management (PHM) [1-3]
2. Suggest the deep digital twins for PHM of linear electromechanical actuators (EMA) [4] With supervised and unsupervised health indicators using sensor data and control signals
3. Implement the approaches and evaluate them on real data from the EMA testbed.

Bibliography / sources:

[1] W. Booyse, D. N. Wilke, S. Heyns, Deep digital twins for detection, diagnostics and prognostics, Mechanical Systems and Signal Processing, Vol 140, 2020, ISSN 0888-3270
[2] X. Li, X. Li, H. Ma, Deep representation clustering-based fault diagnosis method with unsupervised data applied to rotating machinery, Mechanical Systems and Signal Processing,
Vol 143, 2020, 106825, ISSN 0888-3270,
[3] M. Azamfar, J. Singh, I. Bravo-Imaz, J. Lee, Multisensor data fusion for gearbox fault diagnosis using 2-D convolutional neural network and motor current signature analysis, Mechanical Systems and Signal Processing, Volume 144, 2020, ISSN 0888-3270
[4] E. Balaban, P. Bansal, P. Stoelting, A. Saxena, K. F. Goebel and S. Curran, "A diagnostic approach for electro-mechanical actuators in aerospace systems," 2009 IEEE Aerospace conference, 2009, pp. 1-13, doi: 10.1109/AERO.2009.4839661.

Name and workplace of master's thesis supervisor:

prof. Ing. Radislav Šmíd, Ph.D., Department of Measurement, FEE

Name and workplace of second master's thesis supervisor or consultant:

prof. Ing. Radu Grosu, Dr.rer.nat., Institute of Computer Engineering, Vienna University of Technology

Date of master's thesis assignment: **17.07.2021** Deadline for master's thesis submission: **04.01.2022**

Assignment valid until:
by the end of winter semester 2022/2023

prof. Ing. Radislav Šmíd, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student's signature

Acknowledgements

First of all, I would like to thank to my supervisor prof. Ing. Radislav Šmíd, Ph.D.. for his patience, calm and polite approach, and professional advice.

Secondly, I would like to thank to prof. Ing. Radu Grosu, Dr.rer.nat. for the opportunity to write my thesis at Vienna University of Technology and to be a part of his department.

I would also like to thank to CTU and FEE, that allowed me and supported me on my university exchanges.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular, that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague, 4. January 2022

Abstract

This master thesis aimed to study, design and implement supervised and unsupervised deep digital twin for linear electromechanical actuator DSZY1. The thesis contains a detailed linear electromechanical actuator structure with its most common failure identification. It possesses an overview of potential digital twin approaches resulting with our proposed solutions. Our supervised implementation outputs a health indicator with the possibility of failure classification. Moreover, the unsupervised solution is trained just on healthy current and voltage operational data with the ability to create an overall state health indicator, which makes it easily deployable for heterogeneous fleets.

Keywords: deep digital twin, health indicator, GAN, linear electromechanical actuator

Supervisor: prof. Ing. Radislav Šmíd, Ph.D.
Praha, Technická 2, A3-324

Abstrakt

Cieľom tejto diplomovej práce bolo štúdium, dizajn a implementácia kontrolovaného a nekontrolovaného hlbokého digitálneho dvojčaťa pre lineárny elektromechanický aktuátor DSZY1. Táto práca detailne popisuje štruktúru lineárneho elektromechanického aktuátora s identifikáciou jeho najčastejších závad. Takisto obsahuje prehľad potenciálnych prístupov digitálneho dvojčaťa, končiac s nami navrhnutými riešeniami. Naša implementácia pomocou kontrolovaného učenia generuje indikátor zdravia s možnosťou identifikácie chyby. Navyše, naše riešenie pomocou nekontrolovaného učenia je natrénované iba na zdravých prúdových a napäťových prevádzkových dátach so schopnosťou generovať indikátor celkového zdravia. Toto robí naše riešenie jednoducho nasaditeľné pre heterogénne zariadenia.

Kľúčové slová: hlboké digitálne dvojča, indikátor zdravia, GAN, lineárny elektromechanický aktuátor

Preklad názvu: Digitálne dvojča s DNN pre prognostiku a management stavu elektromechanických aktuátorov

Contents

1 Introduction 1
1.1 Motivation 2

2 Linear electromechanical actuator 3
2.1 Construction 3
2.2 Parameters 5

3 Failures in electromechanical actuators 7
3.1 Motor 7
3.1.1 Stator 7
3.1.2 Commutator 7
3.1.3 Rotor 8
3.1.4 Connector cables 8
3.2 Gearbox 9
3.2.1 Gears 9
3.2.2 Bearings 9
3.3 Actuator body 10
3.4 Voltage source 10
3.5 Power Drive Electronics 10
3.6 Position sensor 11

4 Health management approaches 13
4.1 Model-driven health management 14
4.1.1 Mathematical model 14
4.1.2 Observers 15
4.1.3 Parity space 15
4.1.4 Parameter estimation 15
4.1.5 Bond graph 15
4.2 Data-driven health management 16
4.2.1 Supervised methods 16
4.2.2 Unsupervised methods 17

5 Artificial neural network health management methods 19
5.1 Deep digital twin 19
5.2 Artificial neural network basics . 20
5.3 Convolution neural network 22
5.4 Long short term memory networks 23
5.5 Variational autoencoders 24
5.6 Generative adversarial network . 26
5.6.1 Loss functions 26
5.6.2 Wasserstein loss 27
5.6.3 GAN evaluation 27
5.6.4 GAN training problems 28

6 Feature extraction 29
6.1 Signal processing techniques 30
6.1.1 Time-domain techniques 31
6.1.2 Frequency domain techniques 32

7 Dataset and implementation 33
7.1 Teststand 33
7.2 Dataset gathering 34
7.3 Raw data measurement 35
7.4 Data preprocessing 35
7.5 Data normalization 36
7.6 Dataset transformation 36
7.7 Data graphs 37
7.8 Software 38
7.9 Hardware 39

8 Unsupervised approach 41
8.1 Training dataset 41
8.2 Testing dataset 41
8.3 Unsuccessful approaches 42
8.4 Network architecture 43
8.5 Evaluation 43
8.5.1 Discriminator evaluation 44
8.5.2 Generator evaluation 45

9 Supervised approach 49
9.1 Dataset 49
9.2 Network architecture 50
9.3 Evaluation 52

10 Reduced unsupervised approach 55
10.1 Training dataset 55
10.2 Testing dataset 55
10.3 Network architecture 56
10.4 Evaluation 56
10.4.1 Generator evaluation 57
10.4.2 Discriminator evaluation . . . 58

11 Conclusion 61

A List of acronyms and symbols 63

B Figures 65

C Bibliography 71

Figures

2.1 Typical EMA construction - 1 DC motor, 2 gearbox, 3 lead screw, 4 drive nut, 5 cover tube, 6 actuator cover, 7 limit switches [13] 3
2.2 Block schematic of EMA [24] 4
2.3 Linear actuator DSZY1 produced by Drive-System [8] 4

3.1 Gear broken tooth from our experiment 9
3.2 Diagram of most common failure modes 12

4.1 The evolution of maintenance strategies [36] 13
4.2 Categories of models based on knowledge [43] 14
4.3 Bond graph of a car shock absorber example [41] 16

5.1 Example of digital twin for an airplane engine [11] 20
5.2 Neuron j model in layer k [3] . . . 21
5.3 Artificial neural network architecture [3] 21
5.4 Example of CNN for image classification [45] 22
5.5 Example LSTM cell chain [5] . . . 23
5.6 Network topology of an autoencoder [42] 24
5.7 Network topology of a variational autoencoder [42] 25
5.8 Applying reparametrization trick [42] 25
5.9 GAN architecture in block diagram [12] 26

6.1 Time detectability prior to the failure [48] 29
6.2 Typical frequencies for sources of vibrations [48] 30
6.3 Bearing schematic spectrum of vibration signal [10] 32

7.1 Teststand for EMA measurements 33
7.2 Model input as sequence array [22] 37
7.3 Current graph for load 80N and triangle motion profile 37
7.4 Body vibrations graph for load 30N and sine motion profile 38

8.1 Example of stabilized GAN loss 42
8.2 GAN model architecture 43
8.3 Decisions (HI) of trained discriminator 45
8.4 Boxplot of trained discriminator 45
8.5 Histogram of measured and generated force values 46
8.6 Histogram of measured and generated voltage mean feature values 46
8.7 Histogram of measured and generated force values by collapsed model 47

9.1 LSTM model (None stands for arbitrary batch size) 50
9.2 LSTM model loss 52
9.3 Evaluation of real and predicted health indicator 52

10.1 Example of RGAN loss 56
10.2 Histogram of measured and generated values for current crest factor 57
10.3 Histogram of measured and generated values for current mean . 57
10.4 Decisions (HI) of trained reduced GAN discriminator 58
10.5 Boxplot of trained reduced GAN discriminator 59
10.6 Current graph for load 150N and triangle motion profile 60

B.1 Decisions (HI) of collapsed discriminator 65
B.2 Boxplot of collapsed discriminator 65
B.3 Generator model (None stands for arbitrary batch size) 66
B.4 Discriminator model (None stands for arbitrary batch size) 67
B.5 Reduced generator model (None stands for arbitrary batch size) . . . 68

Tables

B.6 Reduced discriminator model (None stands for arbitrary batch size) 69

2.1 Parameters of tested actuator DSZY1 [8] 5

7.1 Sensors used in teststand 34
7.2 Measured scenarios for failures, loads, and motion types 35
7.3 Raw measurement output for duration 40 seconds 35
7.4 Computed features for certain signals 36

8.1 Average health indicator 44

9.1 Health indicator / labels vector meaning with examples 49
9.2 Set of tuned parameters 51
9.3 Top 5 configurations based on grid search 51

10.1 Average health indicator 58

Chapter 1

Introduction

Over the past decades, the industry has changed rapidly. As the human labor price is rising in the developing countries with the economic growth, companies are trying to reduce the number of workers to gain more profit. In many countries and industries, automation has gone so far, that the only human intervention to the production is needed by the remote control workers and the truck drivers bringing the goods for production to the factory. These fully automated factories reached the stage of minimum possible workers with current technology. To reduce their costs, they need to search for other options than cutting the labor work.

And the answer for cutting expenses is predictive maintenance and health monitoring of the technical equipment. For example, if an assembly line needs to be stopped for an hour because of stuck bearing in one manipulator. This problem wouldn't make a big cost on human resources, but a great loss of money on the production.

Nowadays, the fully automated factories are collecting gigabytes of data from the production per day, but in most cases, there are some remote engineers to notice if there is some unexpected behavior or to stop the line to prevent crashes.

With the rise of artificial intelligence, more intelligent methods for predictive maintenance are being developed, that could be easily implemented to the current technical equipment. The great advantage of these technologies is, that they cannot only cut the cost of production, but also save plenty of ecological resources. Giving the right maintenance, not a new component, saves a significant amount of carbon emissions. And this would be one of the biggest challenges over the past years.

The idea of predictive maintenance and fault detection doesn't have to be related only to mass factory production. It can be applied also for instance to the airline industry. In this industry is the cost of repairment enormous and not even talking about a cost of a breakdown.

To increase reliability, reduce aircraft weight and fuel consumption was developed a new concept called More Electric Aircraft (MEA). Its main idea is to replace hydraulic, pneumatic and partly mechanical systems with electric ones.

In our case, we have chosen a linear electromechanical actuator (EMA).

This specific part could be widely used from moving the wing flaps to the toilet door opener. The aim of this thesis is to develop a deep-digital twin for prognostics and health management (PHM) from the data measured on electromechanical actuator DSZY1 produced by Drive-Systems. Develop and evaluate supervised and unsupervised learning approach on the measured data and compare these approaches.

1.1 Motivation

My personal motivation is to widen my horizons in the area of signal processing using neural networks (NN). The NNs used for visual data processing are widely used nowadays even in the real industry. They are bringing a new level of efficiency in production. However, I consider visual data processing area as quite good implemented nowadays in the real factories.

On the other hand, I see a great potential of smart predictive maintenance and fault detection in the production process. The neural networks for this area are still novel and need to be developed more to be able to be implemented in real factories. This thesis can bring me a greater understanding of data processing, selecting the right features, and the architectures of suitable neural networks for PHM. All these fields are related to my future career aims.

Chapter 2

Linear electromechanical actuator

A linear electromechanical actuator is a device, which is transforming electricity into motion force in a linear direction. It is used for applications, where the straight-line movement is needed. There are several types of linear electromechanical actuators. For non-critical use, it is an active part of a door opening system or window automation. The critical use would be for instance in the airplane industry for the wing flaps or in the assembly line as robot manipulators. For the purpose of our research, we will inspect into detail the linear EMAs for non-critical applications containing DC motors. In the next section, we will explain the structure and common parameters of EMAs.

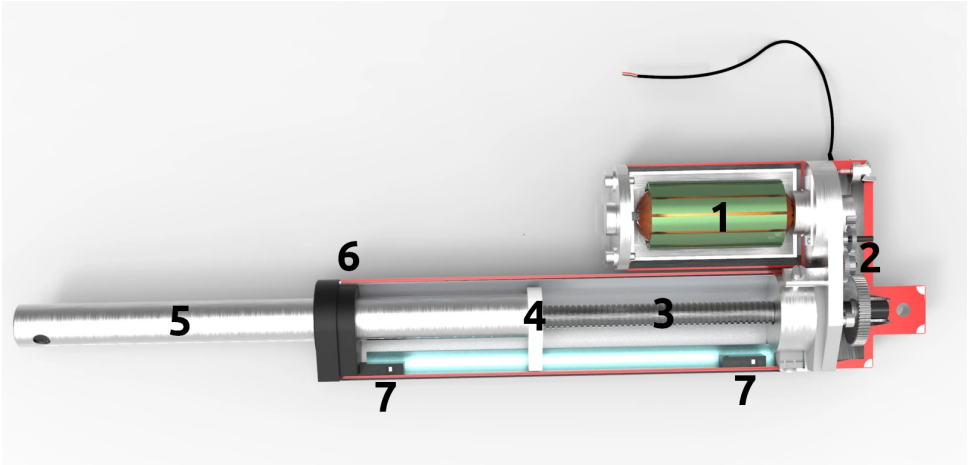


Figure 2.1: Typical EMA construction - 1 DC motor, 2 gearbox, 3 lead screw, 4 drive nut, 5 cover tube, 6 actuator cover, 7 limit switches [13]

2.1 Construction

The linear EMA consists of two main parts - a motor and an actuator 2.1. These two parts are connected with the gearbox to convert the rotation movement of the motor into a linear of an actuator. The gearbox transfers the power from fast-rotating motor to the slower linear movement of the actuator. The lead screw is firmly mounted in the actuator cover. By rotating

the lead screw, the cover tube is either sliding out of the actuator cover or hiding in it. To prevent overextending or overretracting, there are two limit switches, which are cutting the current to the motor, when the limit position is reached. The position sensor can be also attached externally to give feedback to the control system.

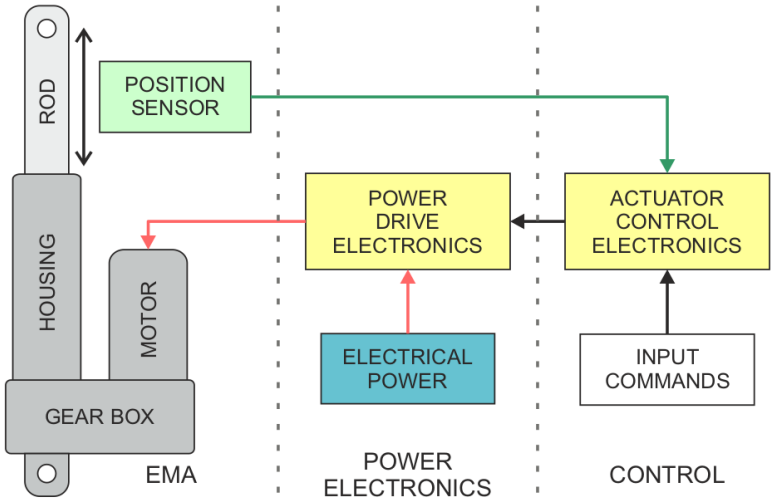


Figure 2.2: Block schematic of EMA [24]

In the figure 2.2, we can see the principal block schematic of EMA. An actuator is usually controlled by the actuator control electronics (ACE). The external sensor is giving feedback to the control unit and computes the action for desired rod movement. The actuator control unit then sends the signals to the power drive electronics (PDE). The right amount and direction of current are then sent to the motor by the H-bridge.

Figure 2.3: Linear actuator DSZY1 produced by Drive-System [8]

2.2 Parameters

For our closer inspection, we have chosen the DSZY1 linear actuator produced by Drive-System. The actuator can be seen in figure 2.3. For different types of applications, different parameters are needed. First, we need to look for the input voltage to meet the voltage level of the placed environment. Secondly, we need to know, what is the maximum load/force applied to the actuator during its runtime. It is also necessary to check if the speed at a certain load meets our expectations. For our non-critical use, we have chosen a certain variant of DSZY series with parameters written in the table 2.1.

Parameter	**DSZY1**	**Units**
motor voltage	24	V
maximum load	150	N
stroke	200	mm
gear ratio	5:1	-
typical current	2.0	A
max speed at 24V	40-45	mm/s
motor speed at 24V	6000	RPM

Table 2.1: Parameters of tested actuator DSZY1 [8]

Chapter 3

Failures in electromechanical actuators

In the previous chapter, we have explained the basic function and components of an electromechanical actuator. In this chapter, we will inspect all components to analyze the most common failures on EMAs. This would help us design feature indicators for the DDT further on. The most common failures can be seen in the figure 3.2.

3.1 Motor

A motor is a fundamental part of EMAs. It gets an electrical input from PDE and creates a rotation movement. Motors are often operated at high rotation rates, where the temperature rises quickly. This helps the development of several failures. Motors are considered as the most problematic part of EMAs [9]. In our case, a DC motor, that consists of three fundamental parts - stator, rotor, and commutator.

3.1.1 Stator

A stator of DC motor consists of permanent magnets mounted to the motor cover and the commutator carbon brushes. Due to its simple structure, stator doesn't suffer from many failures.

The failure arises when a magnet is shifted or even released from its stable position. This moves the neutral stator magnetic field. The efficiency of a motor is decreased and temperature rises. If the temperature reaches $160°C$ and more, the neodymium magnets start to demagnetize. Unfortunately, this causes permanent degradation, and cooling down doesn't bring magnetization back [15].

3.1.2 Commutator

A commutator is a rotary electrical switch, which periodically reverses the electrical current direction in the rotor coils. Its segments are mounted at the end of a rotor and are connected by stator carbon brushes. During motor operation, friction between commutator and brushes causes gradual wear

abrasion. Also, high currents are flowing through the commutator and this makes it a common source of failures.

Commutators often suffer from grooving, threading, sparking, or bar edge burning [35]. These failures are usually caused by overloading, uneven contact surfaces, loose commutator bars, or contamination with dirt. It leads to vibration gain, motor power decrease, sparkling, or rise of temperature. We can observe these failures due to the gain of motor resistance [24].

3.1.3 Rotor

A rotor is a moving component of a DC motor. It consists of shaft, windings, and commutator. Its rotation is caused by the interaction of the windings magnetic field with the stator permanent magnets. This produces a torque of DC motor. A rotor is usually operating at high rotation rates. In our case up to 6000 RPM, which causes high mechanical stress. The most common rotor failure modes are winding breakdown or coil short circuit.

Winding breakdown

Winding breakdown occurs when one or more coil contacts break and create an open circuit. The coil is not able to generate the magnetic field after the disconnection. The motor loses its power and if it stops in this place, it is not able to start rotation. It is usually caused by mechanical stress, thermal stress, dirt, or high centrifugal force to coil contacts. It can be detected by nonuniform current consumption [47].

Coil short circuit

When one or more turns within the winding coil lose the isolation layer and get shorted together, the coil short circuit failure occurs. The short circuits within 2 coils are the most critical. If this happens in the low rotation rates, the breaking effect could be strong enough to stop the motor. The fault phase current will increase and the resistance of the phase will decrease [47].

3.1.4 Connector cables

Connector cables secure connection between the electronic blocks - a computer with ACE, ACE with PDE, and PDE with a motor. Cables can be soldered or mounted by a compatible connector. This simple part suffers from either disconnection or short circuit failure. These failures occur due to high temperatures, excessive mechanical stress, or vibrations. If the current levels rise above the cable limit, the temperature rises, and the isolation or connection degrade or is even melted. The degraded connection has higher input resistance, which causes motor power loss [24].

3.2 Gearbox

A gearbox connects the motor with the actuator body. It transmits the quick motor rotation in ratio 5:1 to the slower lead screw rotation. It consists of gears and bearings. Gearboxes in EMAs are theoretically greased for life [13]. However, lubrication loss is not a rare event. This causes fast teeth or bearing degradation. If the problem is not solved, it can end up with broken teeth or stuck bearing. We can observe it by the changes in vibration signal.

3.2.1 Gears

DSZY1 gearbox consists of 3 stainless steel gears. Gears' surface is often stressed by dirt contamination or lubrication loss and can lead to permanent deformation. Shock overload above the yield point can cause a tooth break 4.3. These failure modes can be spotted by gearbox acoustic emissions or vibrations measurement [37]. Cheaper versions of EMAs contain plastic gears, which are even more vulnerable to the mentioned failures. In some applications with low actuator load and non-critical use, plastic gears can be sufficient, however, for our testing purposes the stainless steel was the best option.

Figure 3.1: Gear broken tooth from our experiment

3.2.2 Bearings

Bearings are located in EMAs, where the rotation movement is needed - DC motor output, gear axes, and lead screw. In classical electrical motors, which are in a long-term operation, the majority of failures come from bearings. According to the article [1] 51% of motor failures are due to bearings. However, in EMAs, the DC motor has a much lower operational rate, which significantly decreases the bearing failure rate. Bearing degradation is caused by particle contamination, misalignment, improper lubrication, or electric arc corrosion. These errors lead to instability of bearings and cause overheating, excessive vibrations, or worsening of rotation ability.

3.3 Actuator body

The biggest part of an EMA is an actuator body. It is a mechanical part, which is performing the final straight-line movement. An actuator body consists of rod housing, lead screw, and drive nut.

When the gasket breakdown occurs, the first failure is lubrication loss. Dry friction slows down the transmission and requires more electric power. If the failure is not detected soon, the temperature rises and the drive nut starts spalling. This would cause the failure called backlash - vulnerability between the lead screw and drive nut causing a delay in movement [24]. In this case, the failures can be spotted by measuring the temperature of the components, monitoring the vibration gain on rod housing, or following the trend of rising peak currents.

3.4 Voltage source

An ideal voltage source can maintain fixed voltage independent of output current or load resistance. In our case, it should operate on 24V and be able to supply inrush current, when an EMA is being turned on. The inrush current is several times higher than the normal operating current. It is also necessary to have sufficient protection of motor circuit to prevent the burn of electronic parts and other failures.

Even if the motor is in optimal operating conditions, these 3 components tend to degrade quicker - diodes, power MOSFETs, and electrolytic capacitors.

In the voltage sources, there are used quick Shottky diodes for switching. Due to the high currents, voltages, and temperatures, they degrade quicker. The side effect is the significant rise of reverse current.

The weakness of electrolytic capacitors is evaporation and drying out the electrolyte. This leads to the linear decrease of capacity and exponential serial resistance gain for the constant temperature condition [24].

3.5 Power Drive Electronics

The main component of PDE is H-bridge. It is powered by the voltage source and controlled by signals from ACE. Its main goal is to switch polarity. In our case, the VNH7070AS from STMicroelectronics is used. It is a fully integrated circuit containing thermal fuse, short-circuit, and cross-conduction protection [44].

Regular overloading leads to a breakdown of one or more power transistors in the H-bridge. A damaged transistor stays either in a permanently opened or permanently closed state. This ends up in the loss of controllability. If the state is permanently opened, and the second transistor in the branch is opened too, it creates an electrical short circuit. This state is dangerous due to exposure of high current to the integrated circuit. The failure mode can

be detected by comparing the direction and level of current flowing through the H-bridge with the control signals.

3.6 Position sensor

A position sensor is a fundamental part of the control unit. It provides feedback of position and is crucial for appropriate control signals. If the position information is not correct and the limit switches fail, the damage of EMA by overextending or overretracting threatens.

Generally, sensors suffer from noise, offset, drift, nonlinearity, hysteresis, and gain issues. This issue can be solved by dynamic or static redundancy [9].

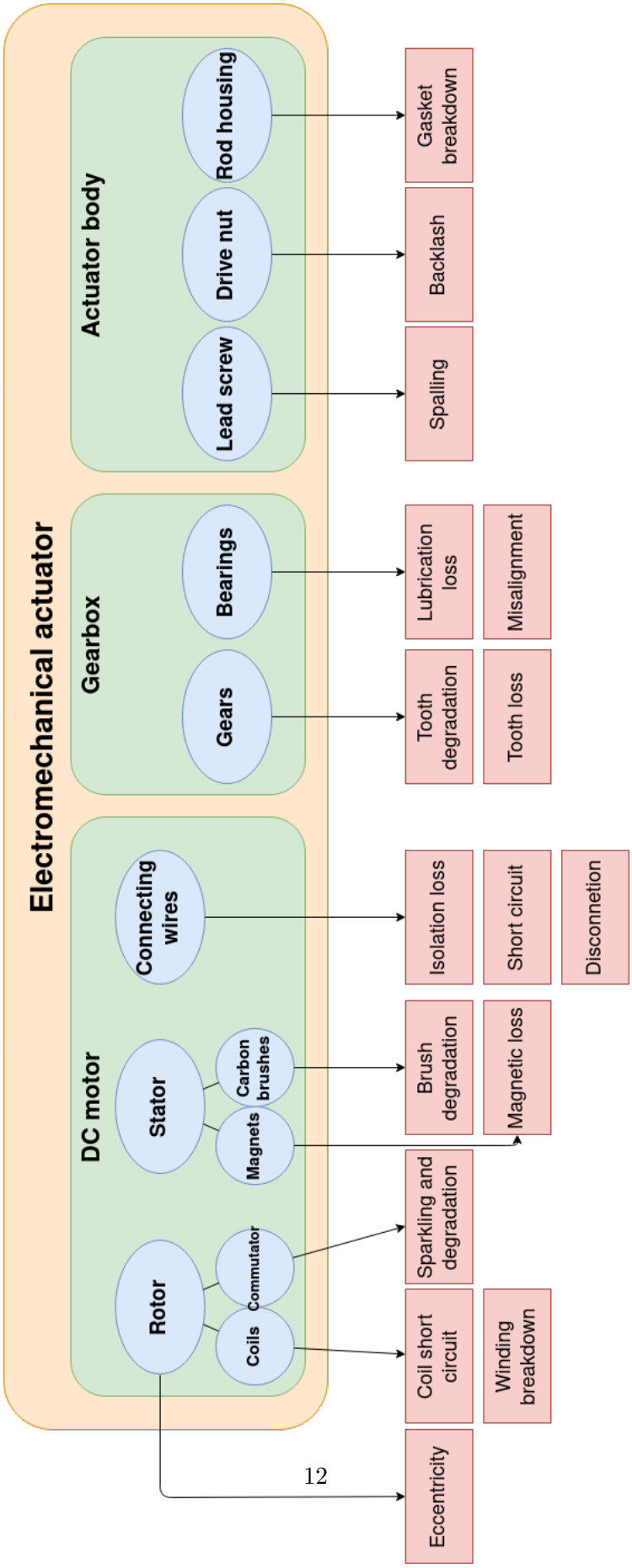


Figure 3.2: Diagram of most common failure modes

Chapter 4

Health management approaches

Health management is a wide area that covers everything that maximizes the machine runtime. Its subclasses are predictive maintenance, fault detection, or degradation monitoring.

Maintenance can be done by several approaches. The most basic one is reactive. It means that we provide a maintenance of a machine only if a failure occurs. This can have great money and time cost so preventive maintenance was then historically introduced. Prevention is based on time slots for checking the machine. However, in some cases the hidden failure can occur right after the maintenance, or the time window is set to be small. This brought us the third evolved approach - predictive maintenance. This technique employs the output signal analysis to recognize the changes in health behavior and schedule the maintenance appropriately for every single machine.

Figure 4.1: The evolution of maintenance strategies [36]

Fault detection (FD) is a field that classifies the type, locates, and determines the occurrence time of a fault. This field is divided into two main subfields - model-based FD and signal processing based FD.

Degradation monitoring tracks the continuous wear of a machine or machine parts from a health/initial state. For example, a battery capacity is very useful information provided for the health monitoring unit to decide on a component repair or change.

Health management can be divided into two fundamental categories. The first is data-driven and the second one is a model-driven approach. In this

chapter, we will discuss the advantages and disadvantages of each and look a bit closer to their subcategories.

4.1 Model-driven health management

The model-driven approach employs a mathematical dynamic model of the system that is directly tied to the physical processes that drive the health of the component [6]. It requires a deep understanding of the physics of the system and the degradation evolution.

The health monitoring is based on a comparison of the model-generated data with the real-model measurements. In order to achieve high reliability, the mathematical model needs to describe the model with high accuracy. However, this is often a difficult task, so in practice are described just parts of complex systems.

The main advantages of this approach are the high precision, deterministic approach, system-oriented - propagation of the failure in the whole system, and possibility of simulations of several failures. On the other hand, the main drawbacks are the high cost of implementation, the need for a degradation model, or as mentioned before, the difficulty of applying to complex systems [33].

According to the knowledge about the system, we can divide them into 3 subcategories 4.2.

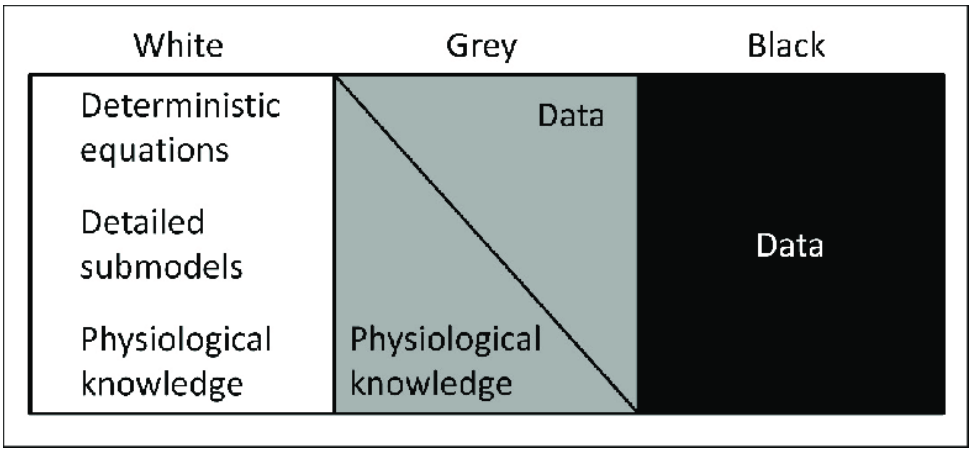


Figure 4.2: Categories of models based on knowledge [43]

4.1.1 Mathematical model

White box model

A white box is a theoretical model, where all knowledge about the system is known. All equations, states, and relations are certainly written. However, to describe a complex system into detail is not possible, so this concept is just theoretical.

Black box model

The basics of black box model are that the relations between inputs and outputs are completely unknown. We cannot describe a system by equations or principles of physics, so we need other methods for model creation. We usually just have data of a system and we try to create a similar model based on inputs and outputs.

Grey box model

The most common case of how to describe a real system is the grey model. As shown in the figure 4.2, it is a combination of a white box and a black box. Some subsystems can be certainly described by equations and some have to be created based on data.

4.1.2 Observers

The observer method relies on that the behavior of the whole system can be deduced from the measured output signals. It means that the condition for a system is to be observable. Residuals can be reconstructed using system unknown intern states and the process output is estimated by an observer. There are several observer-based methods, that are robust in model disturbances and uncertainties, but on the other hand more sensitive to a slow time constant decrease [28].

4.1.3 Parity space

Parity space is based on the transformation of the state-space model of the plant to obtain the parity relations by observing the system on a finite horizon [14]. The redundancies between system variables are used for acquiring parity relations. They are obtained, because they provide equations dependent just on the known variables or data. Either known variables or inputs and outputs. It can be used for both, static and dynamic systems.

4.1.4 Parameter estimation

The principle of fault detection via parameter estimation relies on that the faults of the monitored system with mathematical model states and specific parameters of the system [46]. Estimation is that we possess the basic model structure and the process parameters or state variables are obtained based on input/output knowledge. The parameter estimation is especially effective for multiplicative faults detection [28].

4.1.5 Bond graph

Bond graph (BG) is a unified graphical description that presents a domain-independent and energy-based methodology for modeling the dynamic behavior of physical systems from different domains (electrical, mechanical,

hydraulic, thermodynamic ... etc.) [28]. It is composed of bonds representing an instantaneous flow of energy and single to multi-port elements. BGs are used for systematic residual generation. They are especially appropriate for energy exchanges representation.

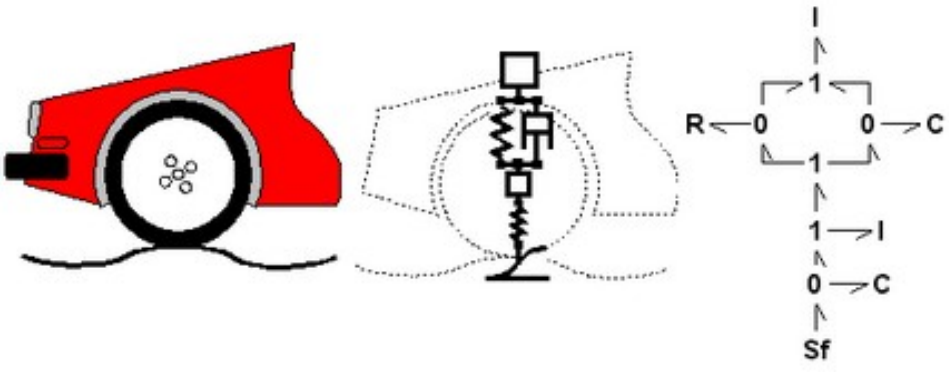


Figure 4.3: Bond graph of a car shock absorber example [41]

4.2 Data-driven health management

A data-driven approach is suitable in the opposite case - we cannot describe the system by equations and we possess plenty of data. It uses statistical methods to inspect the correlation between features and to find outliers and anomalies. The main advantage is the lower cost. On the other hand, it is less precise, the experimental data for the degradation needs to be obtained, it usually has a variability of test results even if the same component is in the same conditions and this approach is more component-oriented rather than system-oriented [33].

Within the data-driven approaches, we can distinguish between two basic categories - supervised and unsupervised learning. The supervised requires to have the data divided into classes based on historical labels. For example class for healthy data, a class for failure mode 1, failure mode 2, etc. On the other hand, the unsupervised methods let the neural network decide for a final class without any prior class information.

4.2.1 Supervised methods

Bayesian networks

A Bayesian network (BN) represents knowledge in graphical form. It belongs to the family of probabilistic graphical models and is represented by a directed acyclic graph [34]. The node represents a random variable probabilistic distribution and an arcs determines the probabilistic relationship between two connected variables. BNs are built on the Bayesian decision theory and are built in 3 steps - dependency among variables is determined, prior probability distribution estimation and condition probability distribution estimation. Its

main advantage is that the node can be updated in any state and it provides a pictorial view of the entire process [34].

Artificial neural networks

Artificial neural networks (ANN) are inspired by biological neural networks. They are collections of nodes and connections in between - like neurons and synapses in our brain. They work especially well in estimating the non-linear functions which are unknown. The drawback is that they require a vast amount of data for proper training and reasonable results. There are several architectures that can be either supervised, unsupervised or semisupervised. The ANNs for PHM will be further discussed in the following chapter.

4.2.2 Unsupervised methods

Control charts

Control charts or Shewhart charts is an old method proposed by Shewhart in 1931. It was designed to determine if the manufacturing process is in a state of control. For a time-series variable, the standard deviation is made and then the upper control limit (UCL) and lower control limit (LCL) are computed. UCL is usually 3 times the standard deviation above the average and LCL is 3 times lower. However, the conditions can be adjusted based on application. The control charts are effective in detecting large changes. A drawback is that when a state out of control is detected, it is not obvious to identify the fault [28].

Principle component analysis

Principle component analysis (PCA) is a multivariate statistical analysis method. Let's assume having a lot of features in a feature space. However, the dimensions are higher than wanted and some features might be dependent. In this case, we use the PCA for feature reduction, that the essential information retain and the new data is more readable than the original. It captures the maximum variance and correlation between the features [28]. This method is linear but many features from real processes are not linear, so the nonlinear PCA approach was developed. For example, the Nonlinear PCA is based on associative neural networks [27] or the Kernel PCA based on nonlinear kernel functions [30].

Chapter 5

Artificial neural network health management methods

As mentioned in the previous chapter there are developed several predictive and health monitoring methods. We have chosen the data-based direction and an artificial neural network (ANN) approach. In this chapter, we will also introduce deep digital twin concept which is closely connected to neural networks (NN) and discuss a few state-of-the-art PHM approaches.

5.1 Deep digital twin

Deep digital twin (DDT) is a deep neural network model, an up-to-date representation of a real physical asset in operation [32]. The model is trained on historical data of a certain asset and is continuously updated by the actual data. DDT is used to predict future behavior, reflect current health conditions, refine the control or optimize asset operation. DDT can be a component, group of components, whole system, or even a group of systems.

In our case, it is the actual health representation of a certain EMA in an airplane. However, it could be the state of all EMAs in the plane, whole plane, or a state of a fleet of planes. DDT reflects the properties, age, environment conditions, degradation, wear, and configuration of a certain asset [23].

DDTs are mostly used for **predictive maintenance**. They can predict remaining useful life (RUL), plan the service of an asset, or the component replacement can be scheduled.

They can be also used for **anomaly detection**. If the real asset behaves significantly different from the digital one, the system can notify the executives to take an action or even stop its operation in order to prevent a catastrophic scenario.

Series of anomalies can trigger fault isolation and its identification. Then the system can take appropriate action or send a notice for service [32]. This use of DDT is called **fault isolation**.

DDT can be also applied to the **operation optimization** by running plenty of different scenario simulations. This can help to set appropriate operating parameters, mitigate risk, optimize control of system during operation or reduce operational costs.

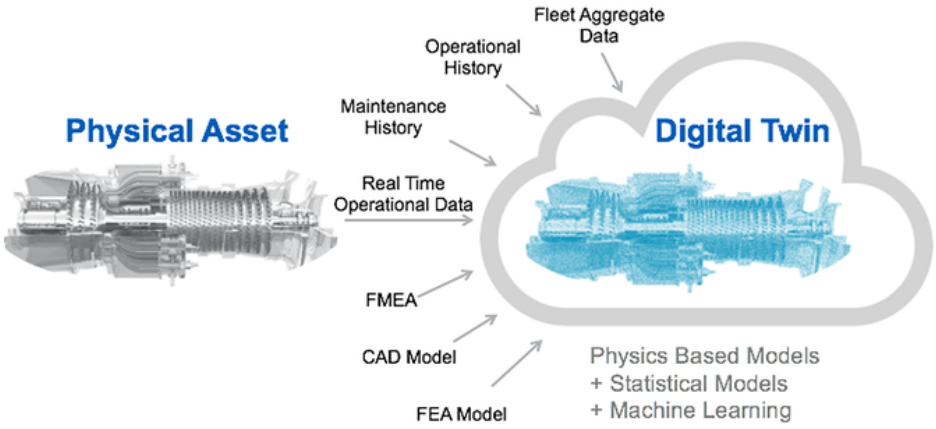


Figure 5.1: Example of digital twin for an airplane engine [11]

In the model approach, a digital twin is given the same inputs as the real asset and generates the virtual output. The output is then compared with the real-world asset output and if some deviations are found, they are further automatically investigated. In theory, the model-based model during the healthy operation should approach zero deviations from real asset output. However, for the more sophisticated and complex systems, it is really challenging to build a model, which so closely approaches reality.

And this is one of the reasons why we decided to implement the statistical model based on historically measured data, the so-called deep digital twin. Usually, the time and cost of collecting the right dataset for industrial machines are expensive. Their operation is planned for several years or even decades and to gather a representative dataset for not only a healthy state, but also a series of failures becomes unrealistic. According to the article [2], to make DDT fast deployable and rapidly reduce time for data collection, its learning should be based on just healthy data. Compared to the model-based model, it is not supposed to replicate the output of a real asset, but the NN model learns the representation of healthy data and its output is the health indicator (HI). HI represents a metric of deviation from the healthy asset data as the NN model tends to learn the space of healthy operation. If the data are further from this space, it decreases the HI, where values close to 1 represent the healthy operation. As the values decrease down to 0, it indicates the unusual working condition, fault, or failure.

5.2 Artificial neural network basics

ANNs are sophisticated computational systems used for solving complex regression, classification, or anomaly detection problems. Like our brains, they consist of neurons and synapses, so-called nodes and connections in

between. Each neuron can have several connections which bring information 5.2. The transmitted information represents a real number which is multiplied by a connection weight. Weights are crucial in the learning process, where are they continuously changed to inform the importance of the connection. So for example the connection bringing the information of house color would have low weight for the house price estimation, but the size will have high weight. All connections to a neuron are weighted and summed and adjusted by an activation function. This information flows to another layer of neurons, where connections exist.

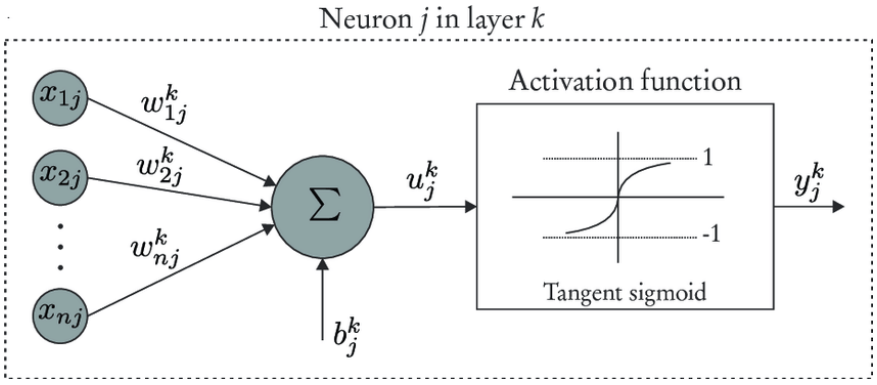


Figure 5.2: Neuron j model in layer k [3]

The input, so-called features, are transmitted to an input layer which dimensions are identical to the dimensions of the input data. The layers in between the input and the output layer are called hidden layers. Information flows through them to an output layer. The output layer can be a binary number, real number, vector, or multidimensional matrix. Depends on the desired output of a certain application. An example of architecture can be seen in figure 5.3.

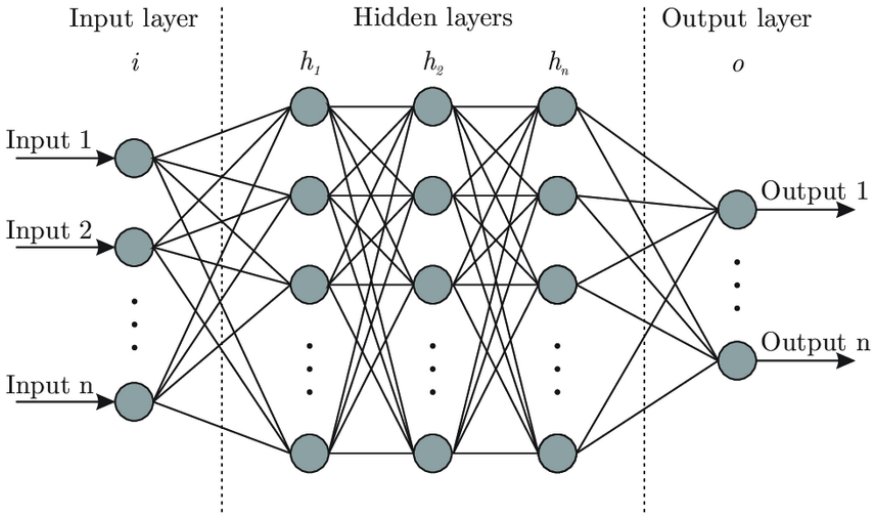


Figure 5.3: Artificial neural network architecture [3]

Each layer can contain a different number of neurons and can execute a different transformation. The architecture of neural networks is a way how the neurons are connected in between and structured in a layer.

This was just a brief introduction to ANNs. For more information about ANNs and related information such as activation functions, hyperparameters, weight initialization, optimizers, and others please refer to the book Deep Learning [17].

5.3 Convolution neural network

A convolution neural network (CNN) is a deep learning architecture that is mostly used in computer vision. CNNs were inspired by the visual perception of living creatures. After a discovery that visual cortex neurons are responsible for detecting light in receptive fields by Wiesel and Hubel [7], first, several CNN examples were proposed.

As mentioned, they are well suitable for computer vision and are especially good performing in object detection and image classification. It implies that its input is usually an image (not always). For black and white or greyscale pictures just 2-dimensional and for colored RGB 3-dimensional.

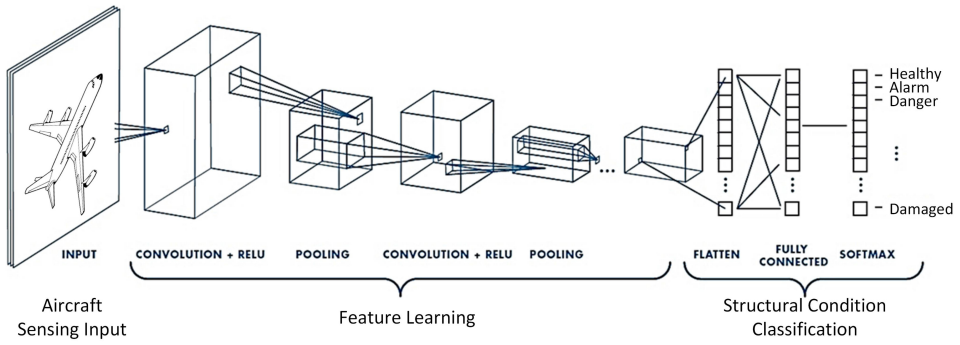


Figure 5.4: Example of CNN for image classification [45]

CNNs are usually constructed with 3 different layer types - convolution, pooling, and fully connected 5.4. To extract the information from the input, convolution filters are used. Filter depth fits the depth of input, but width and height shapes are usually smaller (i.e. 4x4). The filter goes over the whole image and creates a so-called feature map by computing the convolution operation. The filter output is a number or vector according to the depth size.

After each convolution layer usually follows the max or average pooling layer. The max-pooling filter takes the maximum of its size. It performs a down-sampling in width and height [45]. By taking the most significant feature it reduces the network complexity and helps prevent overfitting.

Lastly, the flatten and fully connected layers convert transmitted information for the desired output. This, however, depends on the purpose of the NN.

Recently, CNNs started to be used for health monitoring due to their ability to learn time-series features. It is caused by convolution operation, which compress data information from more features and time steps.

5.4 Long short term memory networks

Long short term memory (LSTM) neural networks are a special type of recurrent neural network (RNN). The RNNs are designed for learning time-series and time-dependent data, not only the single data points. They are applicable to weather prediction or text, speech, and video sequences. In contrast with the feed-forward architecture, RNNs connections can be cyclic. That means that the cell output can be connected within the same layer or even to the previous layers.

The cyclic connections ability creates a kind of long-term memory and the network is able to preserve older data. A good example would be weather prediction. Imagine having hourly temperature measurements dataset. Inputed to the RNN, the network is able to understand the periodicity of data. Colder in the night and winter, warmer in the day and summer, and many other relations, which are for people unnoticeable. And according to these long-term experiences learned, it could forecast the future.

Learning of RNN is not using the basic backpropagation optimization algorithm, but the backpropagation through time method. During the back-propagation, RNNs suffer from vanishing and exploding gradient problems. This ends up with good short-term memory, but poor long-term memory. Improving long-term memory led to new architecture development such as LSTM.

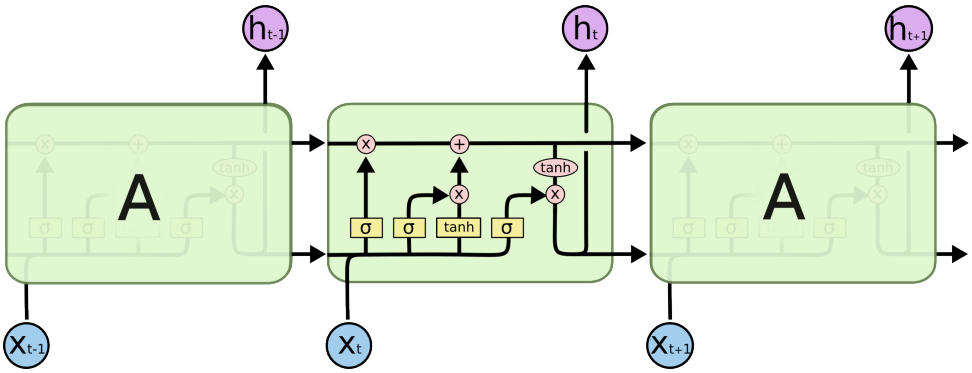


Figure 5.5: Example LSTM cell chain [5]

LSTM neural network consists of LSTM cells 5.5. Each cell has its cell state, which saves long-term information. To control the flow of information transmitted, there are 3 gates - input, output and forget gate. The forget gate decides, if the information should be thrown away or kept. Current input with the previous hidden state is passed through the sigmoid function. Sigmoid function has output in range (0;1). 0 stands for fully forget and 1 for fully remember. Then the input gate follows. It is responsible for the

information saved in the cell state. Following the output gate, which decides for the next hidden state [5].

Not only the weather forecasting, but in terms of health monitoring the remaining useful life (RUL) indicator or anomaly detection applications of LSTM are useful.

5.5 Variational autoencoders

In order to understand variational autoencoders (VAE), we need to understand their predecessors - autoencoders. It is a specific neural network architecture, which is built by two basic parts - encoder and decoder. The connection between encoder and decoder is called latent vector or bottleneck. The input vector enters the input layer of an encoder and forces the size reduction to the shape of the latent layer, which is smaller than the input. This forces to creation of a compressed representation of the input data [26]. The output vector of the encoder is then the input for the decoder. Decoder tries to recreate the original input from the encoded latent vector.

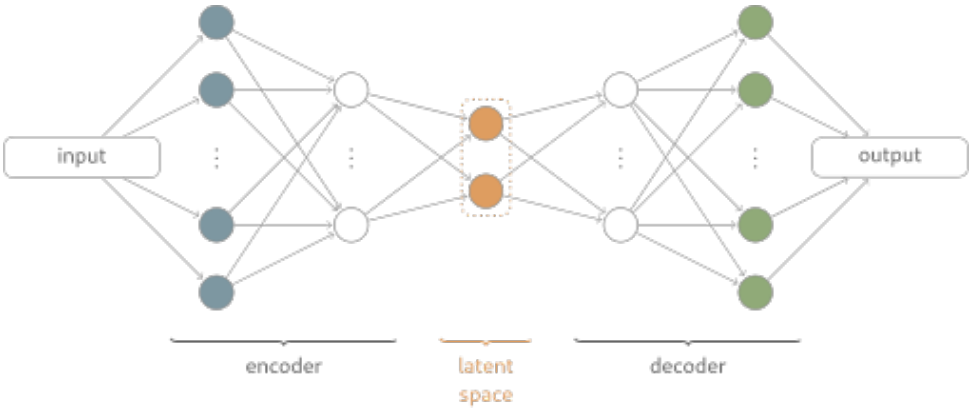


Figure 5.6: Network topology of an autoencoder [42]

Autoencoders are trained by minimizing the reconstruction error. The main part of this loss function is the difference between the original input and recreated output vector. Moreover, more parameters can be added to the loss function.

AEs are mostly used for image compression, anomaly detection, or noise reduction. The latent space of standard AEs doesn't follow any specific distribution. It is a problem for the generative models' application. In generative models, the knowledge of latent space distributions is needed. So then the meaningful output is created out of an artificially adjusted latent space. This change creates a new subclass of AEs - **variational autoencoders**.

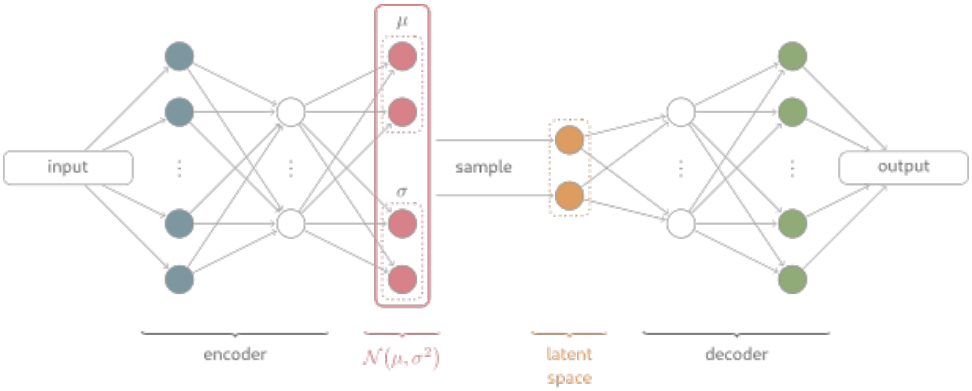


Figure 5.7: Network topology of a variational autoencoder [42]

The significant change in the architecture is in the bottleneck. The latent vector is not directly passed from the encoder's output to the decoder's input. It is divided into two vectors, which represent mean μ and standard deviation σ of a normal distribution $\mathcal{N}(\mu, \sigma^2)$. A latent vector is then created by sampling this distribution.

The result of this is that the decoder model learns to generate similar output to input from the latent vector that is generated from a normal distribution.

Moreover, the loss function needs to be adjusted. We need to add Kullback–Leibler divergence which expresses the distance between our learned distribution and normal distribution $\mathcal{N}(0, 1)$. During training, we force the learned distribution as close as possible to normal distribution. Then the loss function consists of reconstruction loss and the Kullback–Leibler divergence.

(a) : Before **(b) :** After

Figure 5.8: Applying reparametrization trick [42]

However, it is not able to perform backpropagation and compute gradient from the normal distribution in the latent space. In order to solve it, so-called reparametrization trick is used. The fact that every normal distribution can be expressed by standard normal distribution 5.1is applied [42]. By applying this rule, the latent vector can be expressed 5.2.

$$\mathcal{N}(\mu, \sigma^2) \sim \mu + \sigma^2 \mathcal{N}(0, 1) \tag{5.1}$$

$$z = \mu + \sigma^2 \epsilon, \quad \epsilon \longleftarrow \mathcal{N}(0, 1) \tag{5.2}$$

5.6 Generative adversarial network

Generative adversarial network (GAN) is a special type of neural network architecture that was introduced by Ian Goodfelow in 2014 [16]. It is performing well in domain adaption, denoising, data augmentation, super-resolution creation, and others.

It is composed of two fundamental blocks - generator and discriminator. The generator tries to generate new examples which are supposed to be as close as possible to the real data. Then the discriminator is classifying whether the input is real or fake - measured or generated. During the training, they are playing the zero-sum adversarial game until the generator is generating the plausible examples that are classified as real ones. This process usually finishes when the discriminator is tricked in about half cases [4].

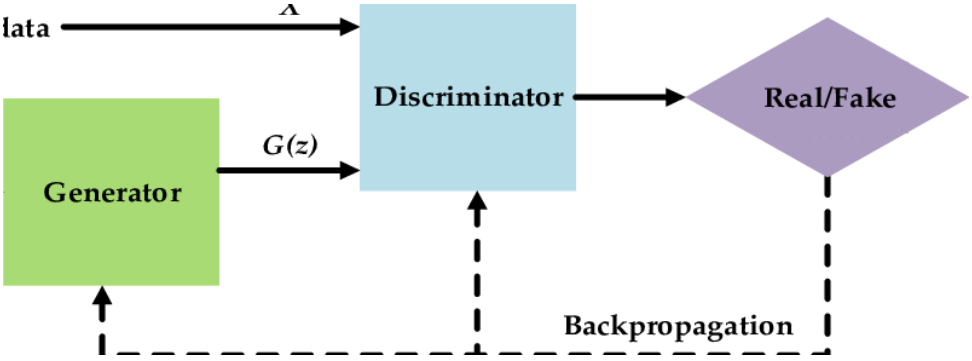


Figure 5.9: GAN architecture in block diagram [12]

The input for a generator is a fixed-length random vector and output are plausible new examples from the problem domain. The vector is a seed for a new example and is randomly picked from a Gaussian distribution.

The discriminator inputs the generated and real examples and usually outputs binary classification whether the input was real or fake. When a discriminator is rewarded for the correct classification, the generator is penalized, which is forcing the update of generator parameters. In the opposite situation, the same opposite is performed - the generator is rewarded and the discriminator penalized. In most common cases after the training process, the generator is kept for generating new examples from the input domain.

5.6.1 Loss functions

As GANs try to replicate real data probability distribution, their loss functions should reflect the distance of two probability distributions. The closer distributions of real and generated data are the better GAN performance. The most common loss functions used in GANs are cross-entropy loss, minimax loss, and Wasserstein loss.

Minimax loss

The minimax function 5.3is derived from cross-entropy between the real and generated distributions, where the discriminator tries to maximize the function and the generator tries to minimize it [20].

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))] \tag{5.3}$$

Where x refers to the real data and z refers to a random latent vector, from where are fake data generated. E_x and E_z have estimated values for real and fake inputs.

In the early stages of the GAN training, the job of the discriminator to distinguish real and fake is easy. To prevent overpowering the generator, modified minimax loss is used [20], so the generator is not minimizing the $log(1 - D(G(z)))$, but maximizing the $log(D(G(z)))$ with the flipped labels.

5.6.2 Wasserstein loss

Wasserstein loss seems to be the most used loss function, for training the GANs. It is designed to prevent common GAN problems as vanishing the gradients and mode collapse. It calculates the distance between two probability distributions using Earth Mover's distance. It represents the cost of turning one distribution into another [31]. The generator maximizes the output of the discriminator on generated data and the discriminator tries to maximize its output distance between the real and fake data.

5.6.3 GAN evaluation

GANs are mostly used for image data. Usually, they were evaluated in terms of how well they are doing relative to the opponent - discriminator VS generator. However, there were no objective metrics on how to evaluate the quality and diversity of generated output and the human-eye evaluation was not certain enough. So the inception score and frechet inception distance were proposed.

Inception score

Inception score is a metric that includes the image/generated output quality and also diversity. It computes the Kullback-Leibler divergence of two probabilities 5.4 (KL(conditional||marginal) in the equation). Conditional class probabilities $p(y|x)$ for every generated image with expected low entropy, which reflects image quality. The marginal probability $p(y)$ is expected to have high entropy, which means that the model generates varied images [40].

$$exp(E_x \cdot KL(p(y|x)||p(y))) \tag{5.4}$$

The lowest score is 1.0 and the highest is the number of the classes. In our case it is just 2 - real or fake, however, GANs can be used to learn more than

binary classes. To conclude, the highest score reached, the better performing model.

There are also a few limitations of the inception score. There is no measure for diversity within the class, so if the generator generates just one image for each class, we can reach the high inception score.

Frechet inception distance

Frechet inception distance (FID) was proposed as an improvement of the inception score. FID is more robust to noise and has also the opposite slope as the IS - the lower FID scores the better, where the best value is 0. It measures the distance between feature vectors of real and generated data [25]. Its big advantage is that FID is sensitive to the GAN mode collapse.

5.6.4 GAN training problems

Training a GAN is not an easy task. They suffer from various failure modes and are often unstable. Some improving techniques were recently proposed, but many of them are still in active research.

Vanishing gradients

At the beginning of the training, the task for the discriminator is easy, while the generator goal to fool the discriminator is very difficult. It can result in the discriminator not providing enough information for a generator to be able to progress. This can fail due to generator vanishing gradients. To prevent this, Wasserstein loss or modified minimax loss functions are used [18].

Mode collapse

If the generator oversimplifies the generated output, it is called the mode collapse. It is generating just one plausible output, while ignoring the other important ones from the real data distribution. For example, we can observe a mode collapse when GAN is generating just one type of dog image, even though it was taught to generate a variety of breeds of dogs. According to the [18], unrolled GANs and Wasserstein loss are designed to prevent the mode collapse.

Fail to converge

GAN training ends, when the discriminator is correct in approximately 50% cases. However, GAN training is very unstable, and reaching this point is difficult. it means, that the discrimination decisions start to behave randomly. During the training, the discriminator feedback gets less informative. As the generator is learning from the feedback, its quality is decreased. Two proposed techniques to prevent converge failure are penalizing discriminator weights and adding noise to discriminator input [18].

Chapter 6

Feature extraction

For fault detection, degradation and prognostics it is necessary to continuously monitor the device and collect the operational sensor data. There is a great variety of sensors and physical quantities, that can be captured - voltage and current consumption, temperature, acoustic emissions, load, position, smoke or gas detection, angular or linear velocity, and others. However, in many applications, it is necessary to extract features from a signal in order to harvest important information.

All signals are beneficial in certain applications but one of the most used and most informative is vibration signal and vibration analysis [39]. The great advantage of vibration signals is that we are able to detect the degradation or fault in a really early stage. According to the [48] it is detectable months before the catastrophic failure 6.1.

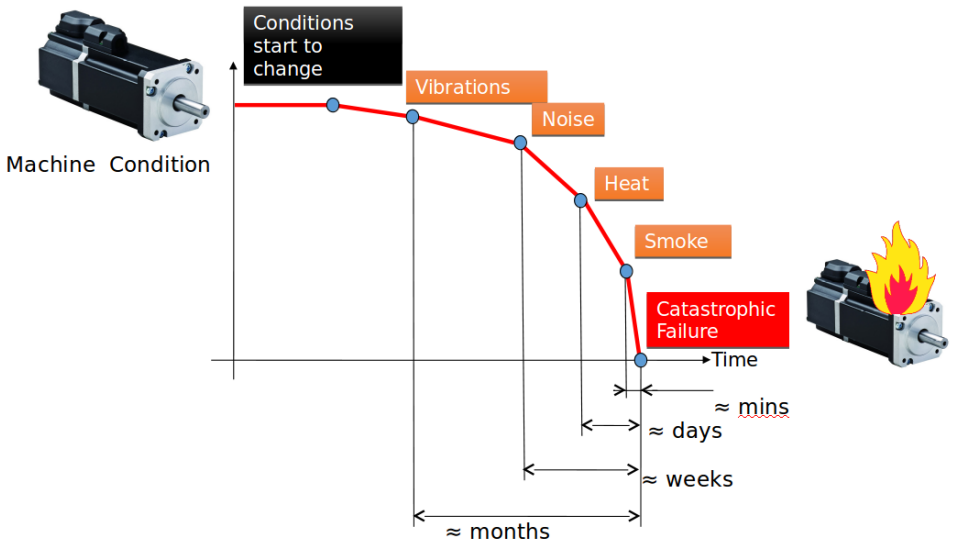


Figure 6.1: Time detectability prior to the failure [48]

All rotational devices (e.g. motor, turbine) or their parts (e.g. gearbox, shaft, bearing) are creating vibrations. This signal contains a great amount of information and the health state can be certainly extracted. Rotary devices are creating vibrations also in the healthy state and the signal tends to

change continuously along with the runtime. Using suitable signal processing techniques and analysis, we are able to extract key features for detecting degradation and faults.

Other highly beneficial signals are current and voltage. They can be used to detect faults in the electrical part such as winding breakdown by nonuniform current consumption, coil short circuit by increased fault phase current and decreased resistance [47], and others. Moreover, current and voltage can be used to detect also non-electrical failures such as dry friction, by increased power consumption. For more accurate detection, other physical quantities (e.g. power) and features need to be derived from them.

Depending on signal characteristics we also need to care about the sampling rate. For example, in case of a fault, the temperature will change in seconds. The temperature sampling frequency could be slow, i.e. in order of hertz. On the other hand, the bearing vibrations are typically around 10kHz (depicted at 6.2).

The measured signals flow to the AD converters which sampling frequencies should pass the Nyquist theorem. It means that the sampling frequency must be at least two times bigger than the maximal frequency of the input signal.

$$f_s \gg 2f_{max} \tag{6.1}$$

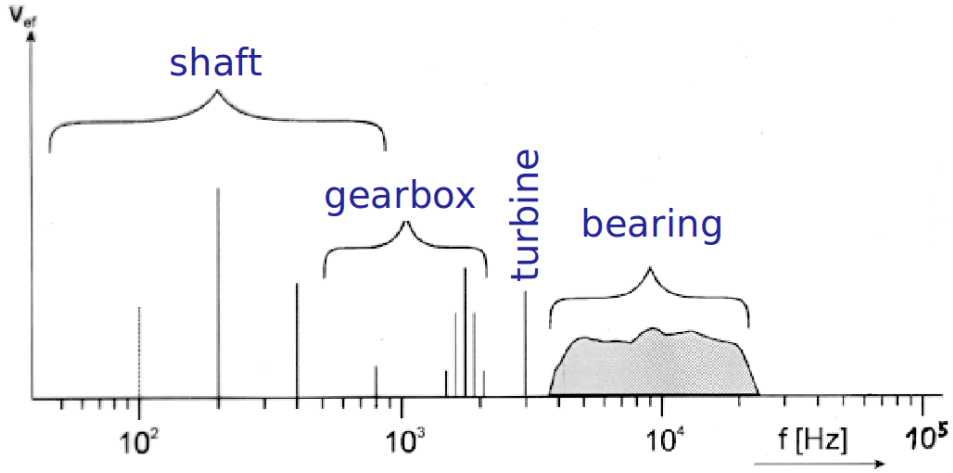


Figure 6.2: Typical frequencies for sources of vibrations [48]

6.1 Signal processing techniques

The signal processing techniques can be divided into 3 fundamental categories - time domain, frequency domain, and time-frequency domain. Usually, before feature extraction, the signal has to be preprocessed. Mean removal, filtering, amplification, or time-synchronous averaging is often used to preprocess the raw signal [29]. A lot of features can be then extracted directly from the preprocessed signal (time-domain). However, for rotary machinery are usually more powerful techniques describing the signal in the frequency domain.

6.1.1 Time-domain techniques

Root Mean Square

It is one of the easiest features which describes the power of vibrations. This feature is good to track noise levels, but cannot detect the location of the component. Its disadvantage is that it is not sensitive in the early stages of a fault. On the other hand, it is performing well in the major faults like out-of-balance and misalignment [29].

$$u_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} u[i]} \tag{6.2}$$

Crest Factor

This feature tells us more about the impulsivity of the signal. It is computed by dividing the peak level of a signal by its RMS level. Direct signal or square signal has the Crest Factor equal to 1. In the motor signals, this factor is usually in the range 2 to 6 [29]. Detected higher values can show us anomaly behavior. Gearbox tooth damage or defect on the outer race of a bearing can be detected using this factor. It is also useful to observe to long-term trend of the Crest Factor.

$$CF = \frac{|u_{max}|}{u_{RMS}} \tag{6.3}$$

Mean, Variance

Mean 6.4 and variance 6.5 are the first and the second statistical moments. Statistical moments describe the probability density curve and its deviation from Gaussian distribution. They might be very useful in fault detection by neural networks, as they help to describe the distribution of a dataset.

$$\mu = E[X] = \frac{1}{N}\sum_{i=1}^{N} x[i] \tag{6.4}$$

$$VAR[X] = \sigma^2 = E[(X-\mu)^2] \tag{6.5}$$

Skewness

Skewness is the third of statistical moments. It describes the skew of the distribution. Skewness is also a good measure when data has outliers. When the value is negative, the distribution is skewed negatively (left). It is caused by extreme values in the data, that move the mean on the left of the distribution and vice versa for the positive skewness values.

$$SKEW[X] = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{6.6}$$

Kurtosis Ratio

It is the fourth statistical moment and as power four it cannot reach negative values. It describes the data distribution in terms of flatness or peakness. Number 3 indicates a normal distribution of data. This also indicates a healthy state of gears or bearings. If the value grows higher, it means more peak values and closer to a failure mode [29].

$$KR = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] \tag{6.7}$$

6.1.2 Frequency domain techniques

Frequency domain techniques provide more complex information, especially for rotation parts. Rotation machinery or its elements are working periodically, so when the fault occurs, it is also periodically repeated in the signal and it is possible to capture the fault in the spectrum. Frequency techniques are closely connected with Discrete Fourier Transform 6.8 (DFT).

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{2\pi j}{N} kn} \tag{6.8}$$

Due to high time complexity of DFT $\mathcal{O}(n^2)$, Fast Fourier Transform (FFT) is often used. FFT time complexity is decreased to $\mathcal{O}(n \cdot log(n))$.

Each rotary element and its fault contribute to the signal spectrum. Techniques for fault detection in the frequency domain are based on magnitudes, phase shifts, or the presence of certain frequencies. They are performing well in bearing and gearbox defect detection, their misalignment, imbalance, and looseness [48]. In general, a disadvantage is their higher computational cost in comparison to the time-domain techniques.

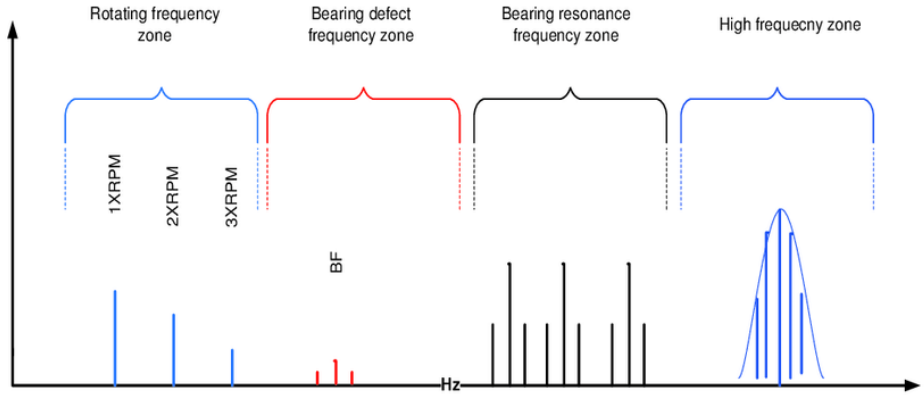


Figure 6.3: Bearing schematic spectrum of vibration signal [10]

Chapter 7

Dataset and implementation

In this chapter will be presented the process of gathering the dataset, data description, and later data post-processing to be suitable for the neural network. Later in this chapter, a brief overview of used software and hardware is provided.

7.1 Teststand

To ensure stability and repeatability of measurements, an automated testing system built by Ing. Ondřej Hanuš 7.1 was used.

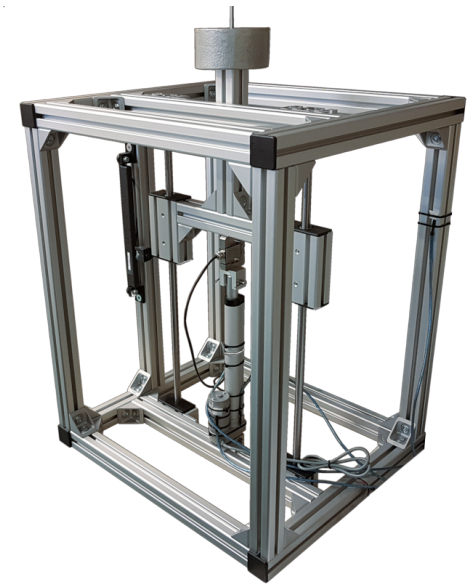


Figure 7.1: Teststand for EMA measurements

Its skeleton is constructed of lightweight aluminum panels with outer dimensions of 680x488x560mm. The tested EMA, in our case DSZY1, is placed in the center and is surrounded by two linear guides, which ensure stable movement for excessive loads. The range of movement within the teststand is 480mm.

Several sensors are placed on EMA itself or on parts of the teststand. Sensors available on the teststand can be seen in the table 7.1. In our case - position, force, voltage, current, vibrations, rotation, and temperatures were measured. To suppress noise, shielded cables are used in the whole system, and control circuits are placed in shielded metal boxes.

Measurement	Sensor	Range	Sample rate
Force	Honeywell Load Cell 151	-500N/+500N	1kHz
Position	Honeywell SPS-L225-HALS	225mm	1kHz
Motor current	VNH7070AS analog output	15A	200kHz
Motor voltage	Resistive voltage divider	-32V/32V	200kHz
Motor temp.	NB-PTCO-155	-30°C/200°C	5Hz
Gearbox temp.	NB-PTCO-155	-30°C/200°C	5Hz
Rod housing temp.	NB-PTCO-155	-30°C / 200°C	5Hz
Ambient temp.	NB-PTCO-155	-30°C / 200°C	5Hz
Motor vibrations	Bruel&Kjaer 4507-B-004	peak 70g, 0.3-6000Hz	51.2kHz
Gearbox vibrations	Bruel&Kjaer 4507-B-004	peak 70g, 0.3-6000Hz	51.2kHz
Motor rotation	Optical encoder 5421-EP111	200CPR	16kHz

Table 7.1: Sensors used in teststand

Teststand is controlled by a program designed in LabView. It can be operated in 3 modes - basic motion, position control, and load control. Motion profiles are controlled by the PID regulator. It provides us 4 motion profiles - sine, rectangular, triangular, and saw. The option of measuring only a subset of available signals is also available.

7.2 Dataset gathering

One of our goals was to avoid simulated data and use just data from a real-world measurement. Another goal was to gather data of EMA DSZY1 in different conditions and for several types of failure modes. Static loads of 30N, 80N, 150N, and 180N were captured for 3 types of motion control - sine, triangle, and rectangle. Each measurement was captured for 40 seconds.

The 4 most common failures of EMAs were injected. For the mechanical part, it was a gearbox tooth deformation and a missing tooth. For the electrical part, a change of the coil resistance in the motor and short of motor winding was injected. Our assumption was, that some failure modes can be more visible within different conditions. Raw measurements were captured and provided to us by Ing. Ondřej Hanuš. For each failure type, every variation of load and motion control scenarios were measured 7.2. For the healthy EMA, two same-sized raw datasets were measured.

Failure	Load	Motion
no defect	30N	sine
motor winding short	80N	triangle
motor coil resistance increase	150N	rectangle
gearbox tooth deformation	180N	
gearbox missing tooth		

Table 7.2: Measured scenarios for failures, loads, and motion types

7.3 Raw data measurement

One raw measurement captured in LabView generated 5 files in tdms format and one text file with measurement setup. The text file contains movement PID constants and upper and lower movement limits.

File contains	Format	Sample rate	Size
measurement setup	.txt	-	$\approx 95B$
motor, rod-housing, gearbox, ambient temp.	.tdms	5Hz	$\approx 8kB$
positions and force	.tdms	1kHz	$\approx 3MB$
increment rotary encoder	.tdms	16kHz	$\approx 4MB$
gearbox and motor vibrations	.tdms	51.2kHz	$\approx 36MB$
motor current and voltage	.tdms	200kHz	$\approx 280MB$

Table 7.3: Raw measurement output for duration 40 seconds

The problem is that every sensor is working on a different sampling rate - from the slowest temperature sensor operating on 5Hz to the fastest current sensor working on 200kHz. A drawback of very different sampling rates is the size of files. While the measurement on temperature sensors could last for days and would barely reach 10MB, the current and voltage was reaching 500MB just in one minute of measurement. This encouraged us to make rather shorter measurements with many distinct scenarios. It was also supported by the fact, that EMAs are typically working in a few second intervals during their operation in contrast to classic electric motors which are running continuously.

7.4 Data preprocessing

Raw data were kept as measured in terms of data cleaning. There was no outlier removal, filtering, or smoothing to test the ability of NNs to learn from imperfect real-world data. Our NN model was aiming to operate in time domain, so the increment rotation encoder data were dropped. Their information gain would be much higher using the frequency-domain signal processing methods.

The big advantage of time-domain features is in general less computational complexity. Units sampled with far distinct sampling rates were merged to a

sampling rate of 1kHz. It means that the slow signals remain unchanged (force, position), the fast (current, voltage) and vibration signals were described by features. 200 time values to 1 feature value for fast signals and 51.2 time values to 1 feature for vibrations (51 for 4 time windows and 52 for the 5th one). RMS, crest factor, and 4 statistical moments (mean, variance, skewness, kurtosis) were computed for almost all downsampled signals 7.4.

40 second measurement is too short for detecting the failures from temperature indicators in this application. It highly depends on the outside heat and the time is too short for heat propagation to the parts of EMA. However, it might have information gain in future experiments, so the temperature values were kept and linearly upsampled from 5Hz to 1kHz.

After we obtained the upsampled temperature data and the features from fast and vibration signals, we created one unified dataset containing 28 columns. The memory needed for storing one measurement decreased approximately from 320MB to 15MB in h5 format.

Signal	RMS	Crest	Mean	Variance	Skew	Kurt
motor current	X	X	X	X	X	X
motor voltage	X	X	X	X	X	X
gearbox vibrations	X	X		X		X
motor vibrations	X	X		X		X
position	kept original sampling 1kHz					
force	kept original sampling 1kHz					
temperatures	linearly upsampled to 1kHz					

Table 7.4: Computed features for certain signals

7.5 Data normalization

A good practice is to normalize the data for the neural network. They become less human-readable, however, it generally helps for faster learning and speeds up convergence. Each feature was normalized to range <-1;1> between its minimum and maximum value.

Firstly, the <0;1> normalization was used. But later we observed that <-1;1> normalized dataset has better learning performance. It is due to activation functions that are used in the GAN structure. Generally, ReLU and tanh activation functions are performing better, when also negative values are provided. The supervised LSTM approach was normalized to range <0;1>.

7.6 Dataset transformation

Most information in our data is hidden in the time relationships. That is why we rearranged our dataset to sequence array 7.2 with optimal window size 28. For example to make a sequence array with step 1 from a classic 2D array of

size (100, 20) with window size 15 means taking (1-15, 20), (2-16, 20), (3-17, 20), etc. This will create a 3D array containing same-sized, time-shifted 2D arrays. Then is NN also able to learn time relations between the features.

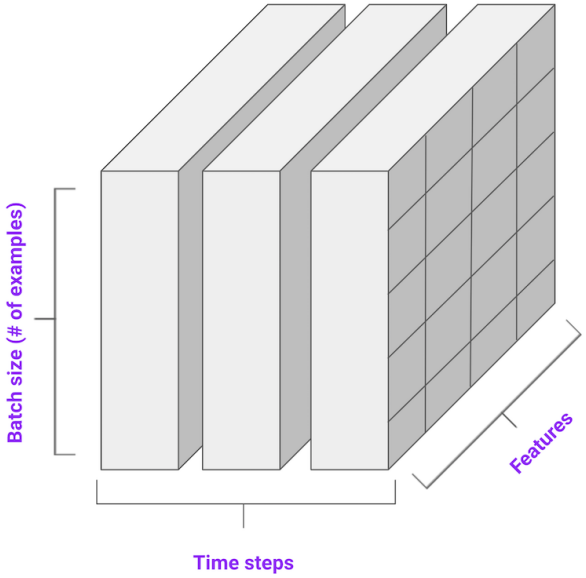


Figure 7.2: Model input as sequence array [22]

7.7 Data graphs

Our dataset contains 10 physical quantities and 26 features derived from them. To visualize all of them would take a lot of space and doesn't bring much information. Therefore we have chosen the following graphs which can help us understand changed behavior during the failure modes. Graphs show us easily human notable signal changes. Signal changes are not visible in every measured configuration.

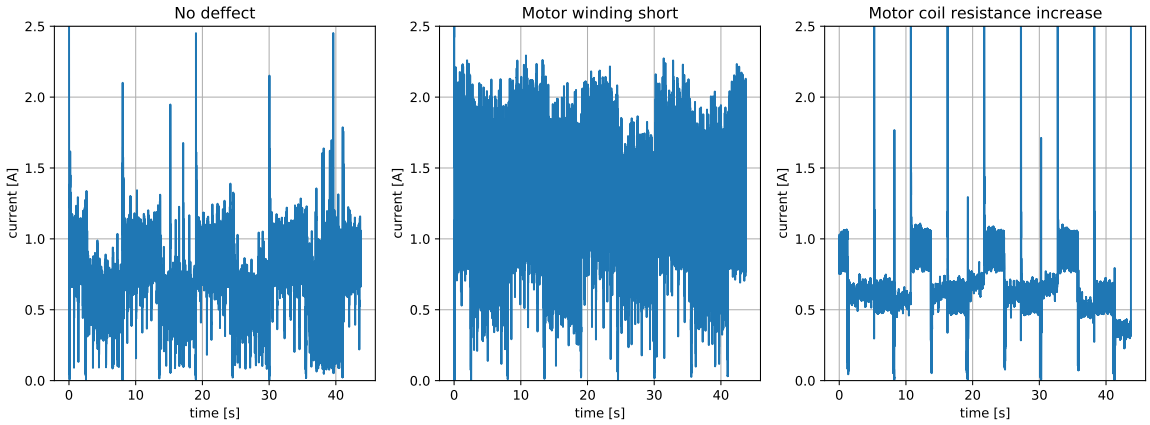


Figure 7.3: Current graph for load 80N and triangle motion profile

In the figure 7.3 are depicted 3 motor current graphs. All 3 of them were measured for the same load and motion profile conditions - 80N load, triangle motion profile. The first one shows us the healthy state. In the second one we can see the increased current consumption. This belongs to the short of motor winding, which causes almost doubled current flowing through the motor. In contrary, the third graph has decreased motor current. In this case, the increase of motor coil resistance failure was injected, which prevents flowing higher currents.

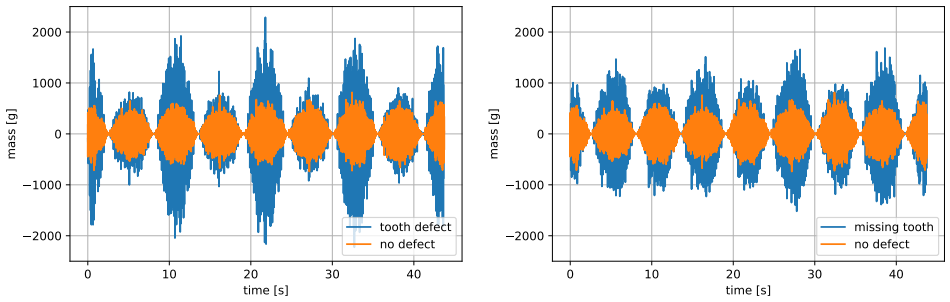


Figure 7.4: Body vibrations graph for load 30N and sine motion profile

The figure 7.4 shows us body vibrations for load 30N and sine motion profile. Orange color shows the lowest vibrations, which stands for the healthy operation. The first graph shows us vibrations belonging to the tooth deformation failure. We can see that every second vibration wave reaches higher values. This is caused by uneven tooth degradation. If the EMA is extending, the worse part of the tooth is employed, which causes higher vibrations. For retracting movement, degradation is lower and vibrations aren't so significant. In the case of missing tooth in the second graph, the vibration signal is reaching slightly higher values and is more even. This can be explained by the same tooth surface conditions for extending and retracting movement.

7.8 Software

All software parts used for the thesis were written in Python 3.6. In comparison to MatLab, it is a free, open-source programming language providing numerous open-source libraries.

For the data preprocessing, mainly NumPy library was used. It implements various functions to work with matrices and compute linear algebra operations over them.

For storing and manipulating the dataset, Pandas library was used. It contains data analysis and database-like functions for data management.

Neural network model structure and training were written using the TensorFlow library. It is an open-source API for machine learning, developed mainly by Google Brain Team. It provides a user-friendly high-level API

called Keras, but also supports a low-level API for custom NN structure creation.

When a NN is designed by a user, a computation graph is created. During the execution, the computation graph is filled with data, and training is performed. For debugging purposes, an eager execution is used, which evaluates operations immediately without building computation graphs. It supports graphic or tensor processing unit accelerators for faster computation. In our case, the TensorFlow version 2.7.0 was used.

7.9 Hardware

For debugging purposes, the code was first written and executed in Google Colab Jupyter notebooks. The Google Colab platform allows each Google account user to perform a free cloud execution on CPU, GPU, or TPU accelerator. We connected to a GPU session with 12GB Nvidia Tesla K80 and 80GB of disc space.

Google Colab was not sufficient for the hyperparameter search, so when was our model debugged, we switched to the Czech Technical University Cantor GPU servers. They possess 256GB NVIDIA GTX 1080Ti with 500GB SSD disc space, which made the computation faster.

Chapter 8

Unsupervised approach

For the unsupervised DDT approach, we have chosen a GAN architecture. According to the [2], it is more sensitive and has better performance than VAE. GANs are using unsupervised learning algorithms that use a supervised loss (real/fake) as a part of the training. GANs are especially good at learning the dataset distribution. Our aim was to teach a GAN the healthy dataset distribution to produce the health indicator in the range <0;1>. For inputed healthy data sequences the HI should approach number 1 and for the faulty HI should reach lower numbers down to 0.

8.1 Training dataset

Our training dataset consists of 12 measurement setups for different load and motion profiles. The duration of each measurement is 40 seconds. In raw format, it is 7.2GB of data. Dataset was acquisitioned on new healthy DSZY1 EMA. Gearbox and body vibrations, motor voltage and current, force, position and motor, gearbox, rod housing and ambient temperature were measured.

After the data preprocessing, normalization and sequence transformation were executed. The dataset was shuffled and converted to a batched TensorFlow dataset. Its memory requirements were decreased to 1.5GB. The first assumption was, that shuffling the dataset will disable learning some time relations between the data. However it showed up, that shuffling the dataset is crucial in order to prevent overfitting and increase the model robustness.

8.2 Testing dataset

The testing dataset consists of 4 failure datasets merged with the second healthy one 7.2. Each dataset is containing 12 measurement setups for different load and motion profiles.

8.3 Unsuccessful approaches

At the beginning, 3 open-source GAN architectures were tested. The model structure was taken as a core code and adjusted to our dataset. Different numbers and types of layers, activation functions, kernel initializers, optimizers, and other hyperparameters were tested with no reasonable results.

The most notable is a TF-GAN tutorial [21], which uses the Tensorflow GAN library proposed by Google. The GAN library implements a convenient infrastructure for training a GAN as a set of training features, losses, and evaluation metrics. Its main element is a GANEstimator. The programmer defines generator and discriminator models, losses, evaluations metrics, and optimizers as parameters of this object. It provides train and evaluates functions, which automates the GAN training and makes it convenient. On the other hand, its big disadvantage is, that it uses Tensorflow 1.x version. Some functions from Tensorflow 2.x are not available and some are deprecated in newer Tensorflow versions, which caused us compatibility problems.

In our case, Wasserstein loss function for protecting vanishing gradients and mode collapse effect was used. As evaluation metrics, inception score and Frechet inception distance were applied. In spite of these novel GAN techniques, we were not successful in getting the expected results.

In the other two attempts, the training was also highly unstable. In one epoch the discriminator was overpowering generator and in the next epoch the opposite. Metrics were oscillating and the networks didn't want to converge. We found out, that in contrast with classical classification or regression NN losses, the GAN losses are counterintuitive and in some cases, just evaluation shows the real performance of NN.

To solve these problems, stabilizing techniques [38] were used. GAN stabilization is more discussed in the following section.

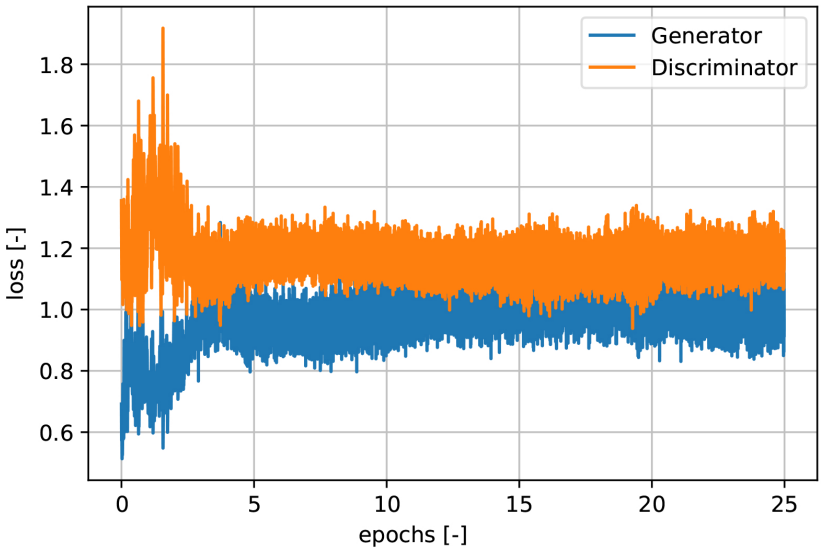


Figure 8.1: Example of stabilized GAN loss

8.4 Network architecture

The other attempt is based on a deep convolutional GAN for the MNIST dataset tutorial [19]. Its principal structure with layer types is depicted at figure 8.2. For input and output dimensions for each layer, please, refer to B.3 for generator and B.4 for the discriminator.

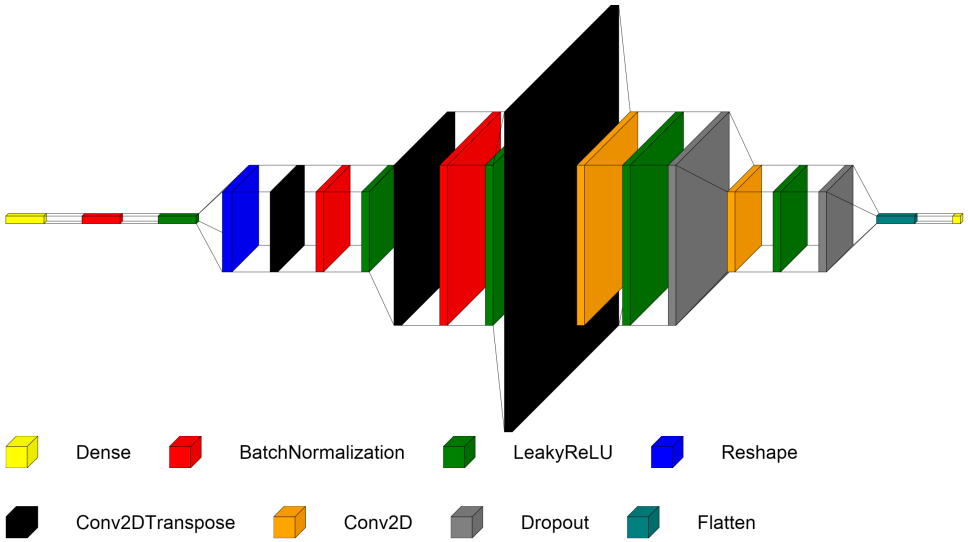


Figure 8.2: GAN model architecture

Our architecture is based on recommendations for training a stable GAN [38]. In the discriminator, the convolutional layers with stride replaced max-pooling layers for decreasing dimensions. Fully-connected layers are used only for input and output. A batch normalization layer was added after the first three layers of the generator. The batch normalization layer takes the input of the previous layer and normalizes it to have unit variance and zero mean, which stabilize the generator training process. As activation functions, leaky ReLU were used as they are improving the vanishing gradient problem.

The generally advised optimizer for GANs is Adam. It was used for both generator and discriminator with the same learning rate 5.10^{-5}. The model was trained with batch size 256 for 25 epochs. As the loss function, binary cross-entropy was used. For discriminator, it combines a loss on real and generated data and quantifies how well discriminator distinguishes them. For the generator, it quantifies how successful the generator is in tricking the discriminator for classifying the generated as real ones.

8.5 Evaluation

Training just on healthy data brings a big advantage of easiness and low cost of data collection. This approach can be also fast deployable on heterogeneous assets just using their operational data. On the other hand, there is not

yet a clear line which failure is more serious. The HI can be lower for tooth deformation than for the missing tooth in the gearbox. Intuitively, humans would say, that a missing tooth is a more serious failure. However, the vibration signals for tooth deformation could be higher. Then its data distribution is more distant from the healthy data than the missing tooth distribution is. For these reasons, we needed to clearly draw unique evaluation metrics for both generator and discriminator.

8.5.1 Discriminator evaluation

The discriminator output or so-called health indicator is in range <0;1>. Our goal was to distinguish all 4 faulty datasets from the healthy one. In other words, the HI for healthy is in higher ranges and has the least overlap with ranges of faulty datasets.

During the training, both models were saved for every epoch. Then we collected 768 decisions for each failure type in different conditions. One measure for choosing the best discriminator model was the maximum of sums of average failure decision distance from the average health decision distance. In other words, the further was the healthy decision from the failure ones, the better. The second metric was the sum of their variances. It means, that the decisions for the same failure type don't oscillate too much.

Datasets	Trained model	Collapsed model
no defect	0.587885	0.614583
gearbox tooth deformation	0.19399	0.375257
gearbox missing tooth	0.25219	0.369388
motor winding short	0.201465	0.47252
motor coil resistance increase	0.240406	0.440298

Table 8.1: Average health indicator

The collapsed model evaluation results are depicted at B.1 and B.2. The well-trained model decisions are depicted at 8.3 and 8.4. For both models, the average healthy value is approximately 0.6. In the case of a well-trained model, we can see an area around a value of 0.2 with low variance which belongs to the failure datasets. This area can be clearly distinguished from the healthy state range. In the case of the collapsed model, we see the big overlap and oscillation of decisions. Also, the average HI values are closer to the no defect state 8.1.

In the well-trained discriminator there are also a few decisions that overlap - HI of no defect is lower than HI of defect state. In the real application, more decisions would be taken and averaged to reach higher accuracy. One decision belongs to the 28 time steps of 28 features. In the time domain, it represents a $28ms$.

Another solution to improve would be the increase of time steps of the input data. However, this would lead to adding model layers and increasing the training and computation time. It also should be short enough, that the

signal doesn't possess a motion profile trend. In our case, we can reach higher accuracy with lower computational cost by the decisions averaging.

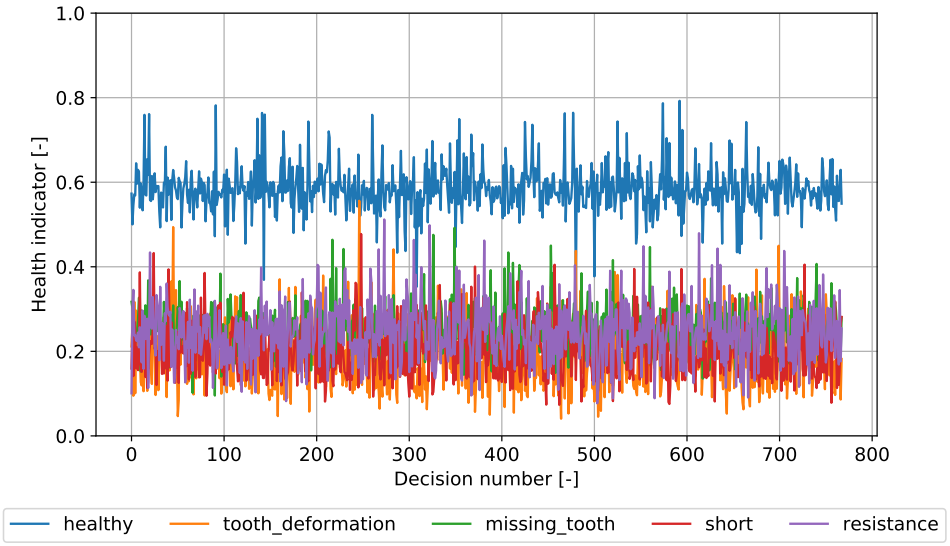


Figure 8.3: Decisions (HI) of trained discriminator

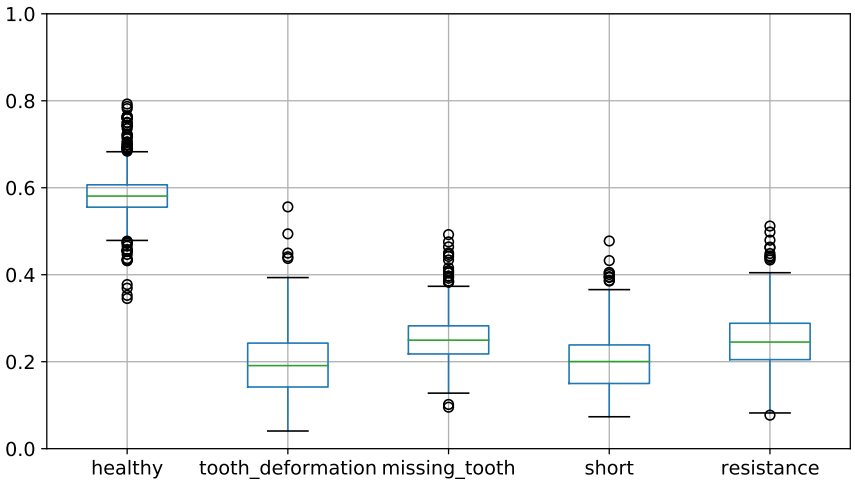


Figure 8.4: Boxplot of trained discriminator

8.5.2 Generator evaluation

The evaluation of the generator is more straightforward. Its main goal is to replicate the healthy data, which also means to learn the probability distributions of all features. The generator evaluation metric was chosen Wasserstein distance. It represents a distance between two probability distributions by computing the minimal cost of turning one distribution into another. The Wasserstein distance was computed between the healthy dataset and the same-sized generated dataset. It was executed for every saved generator

model. Based on this metric the best generator could be picked. In our case, the best metrics results were reached for the generator and discriminator duo saved in the same epoch.

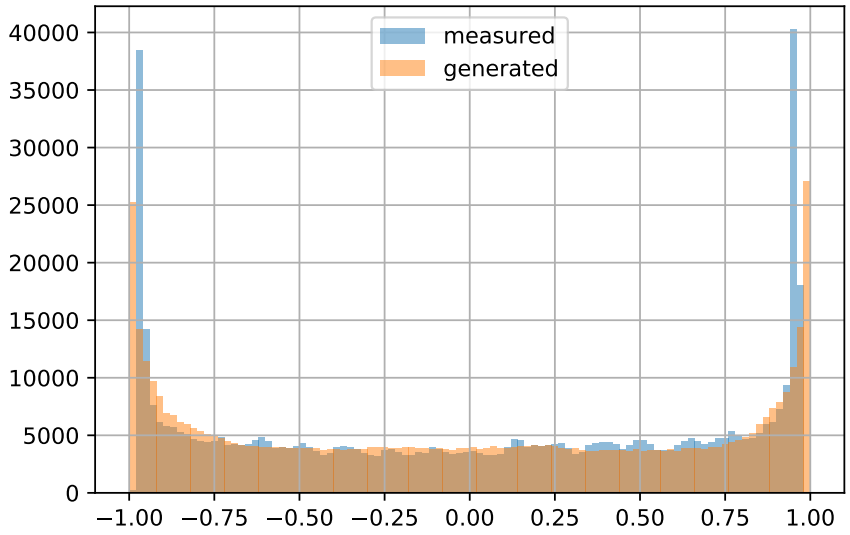


Figure 8.5: Histogram of measured and generated force values

A histogram of measured healthy force data compared with a histogram of generated force data can be seen in the figure 8.5. We can see that the generator was able to learn the data distribution of the force data.

Moreover, some features in the dataset have more difficult data distributions. For this case, we depicted the voltage mean feature histogram 8.6. This proves, that the generator is able to identify main peaks and learn to generalize its distribution.

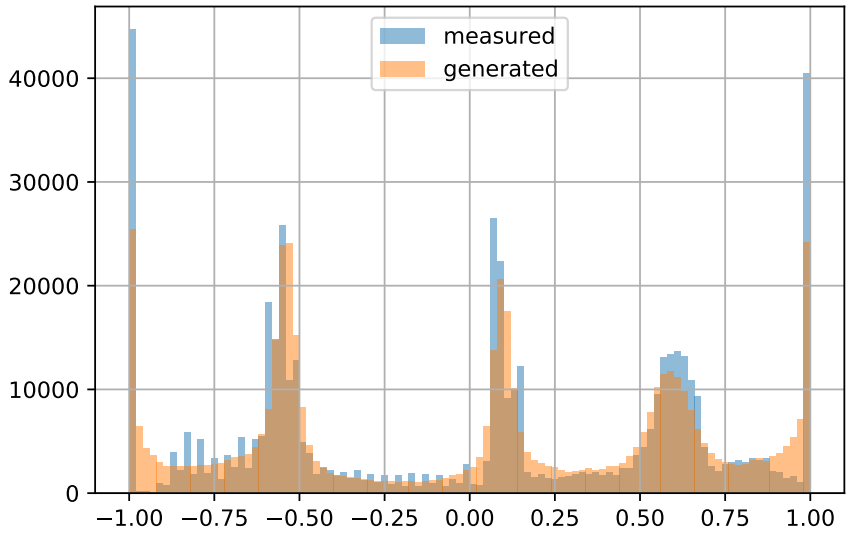


Figure 8.6: Histogram of measured and generated voltage mean feature values

To compare, the collapsed model was not able to learn even easier distributions. Depicted at force histogram 8.7

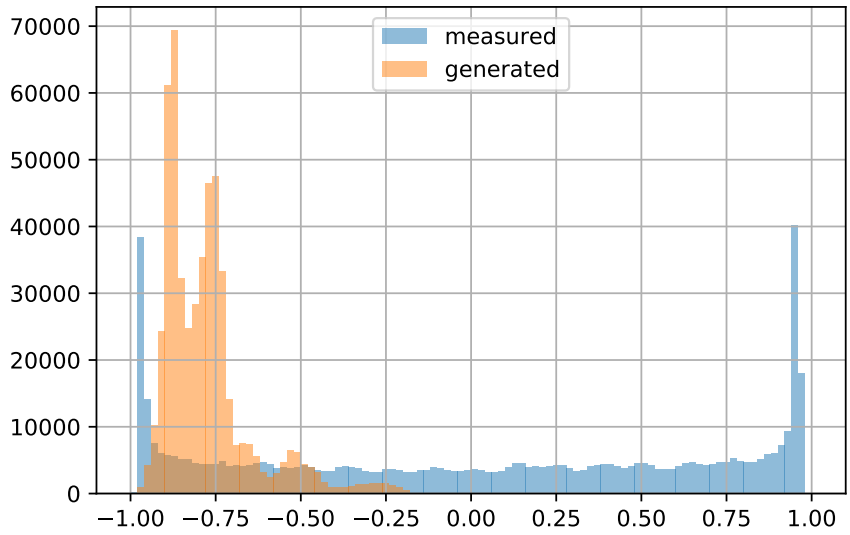


Figure 8.7: Histogram of measured and generated force values by collapsed model

Chapter 9

Supervised approach

For the supervised approach, we have chosen an LSTM architecture. It is a special type of recurrent neural network, which is designed for learning from time-series data. In this case, a health indicator isn't just a single float number, but a vector containing 4 float indicators. Each indicator stands for one measured failure mode and is in range <0;1>. The state of no defect is indicated by vector $(1, 1, 1, 1)$. If a value is decreased, it signalizes a specific failure 9.1. In our case, we expected just one value to decrease as we measured only datasets containing 1 failure. The state of gearbox missing tooth and motor winding short are the most fatal, so we assigned the value 0. The motor coil resistance increase and gearbox tooth deformation are also serious failures, but less crucial, so we set the indicator to 0.2 and 0.5 respectively. This is also for demonstrating, that our network is able to learn different ranges of degradation.

Meaning of indicator				
healthy/no defect	1	1	1	1
gearbox tooth deformation	0.5	1	1	1
gearbox missing tooth	1	0	1	1
motor winding short	1	1	0	1
motor coil resistance increase	1	1	1	0.2

Table 9.1: Health indicator / labels vector meaning with examples

9.1 Dataset

Our dataset is formed from 5 merged datasets for each failure type. Each failure dataset contains 12 measurements of duration 40 seconds for different load and motion profiles. Data were normalized into range <0;1> and labels were added. Then the dataset was split proportionally from each failure dataset. Major part - 60% for the training, 20% validation, and 20% for testing purposes. Data were also transformed to a sequence array with size 28 features times 28 time steps.

9.2 Network architecture

The core of our supervised architecture 9.1 were 2 LSTM layers followed by a fully-connected output layer with a size of HI (4x1 vector). To prevent overfitting, after both LSTM layers were added a dropout layer. The dropout ratio was set to 20%.

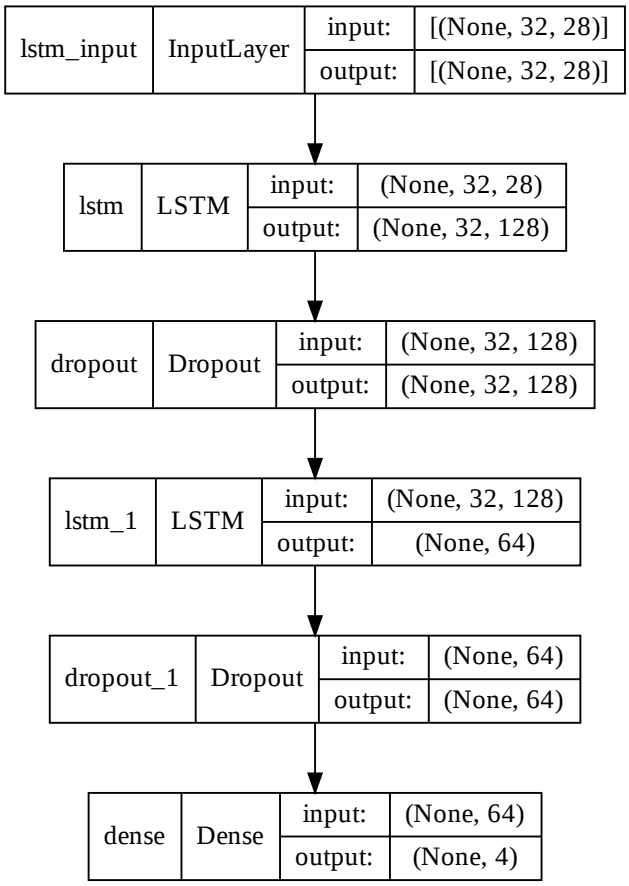


Figure 9.1: LSTM model (None stands for arbitrary batch size)

At first, we tried several learning configurations to find approximate hyper-parameters for a well-trained model. In order to find an optimal model, a grid search was used. All tuned parameters are in the table 9.2 totaling 108 combinations of model setups. The mean square error (MSE) was computed as a model loss. All of them were trained with Adam optimizer for 10 epochs using callback function saving the best performing model according to the minimal validation loss. The sequence length was fixed to 32.

Hyperparameter	values
batch size	$32, 64, 128, 256$
learning rate	$0.0001, 0.00005, 0.00001$
LSTM cells in layer 1	32, 64, 128
LSTM cells in layer 2	32, 64, 128

Table 9.2: Set of tuned parameters

In the table 9.3, we can see the top 5 configurations according to the training loss. The lowest loss was on a neural network with 128 LSTM cells in the first layer, 64 in the second one, learning rate 0.00005, and batch size 32. We can notice, that the lower batch sizes had better performance in general.

No.	batch size	learning rate	L1 cells	L2 cells	MSE loss
1	32	0.00005	128	64	$3.1776 \cdot 10^{-4}$
2	32	0.0001	64	64	$3.2322 \cdot 10^{-4}$
3	64	0.0001	128	128	$3.3786 \cdot 10^{-4}$
4	64	0.0001	32	128	$3.5314 \cdot 10^{-4}$
5	32	0.00005	128	32	$3.5874 \cdot 10^{-4}$

Table 9.3: Top 5 configurations based on grid search

Afterward, the second grid search was performed. The LSTM cells numbers were set according to the best results (L1 - 128, L2 - 64). Learning rate and batch sizes options were kept. The main reason to compute the second grid search was to find an ideal sequence length within options {32, 64, 128, 256, 512}. In this case, the results showed that bigger sequence lengths are performing worse in general. This can be explained by the motion profiles. A sequence length of 512 is having a data of 0.512 seconds of EMA operation. In this data, part of the motion profile can be seen. However, that is an undesired situation, as the motion profile doesn't possess degradation information. To conclude, we selected the best configuration according to the 9.3, where the sequence length was fixed to the lowest value 32.

In the figure 9.2, we can see the decrease of loss along with epochs. The validation loss is expected to be higher than the training loss, which applies to our best model loss. However, in some configurations, the first 2 epochs had the validation loss slightly lower. This can be explained by the dropout layers. During validation, the dropout layers are not enabled, which could lead to lower validation loss at the early stage of training.

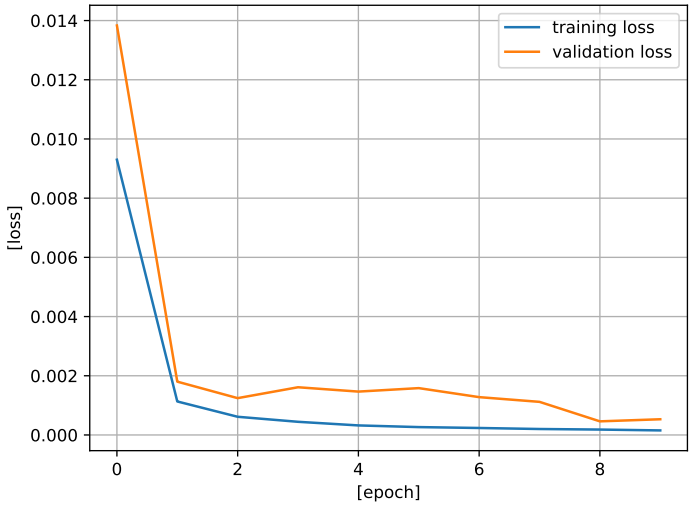


Figure 9.2: LSTM model loss

9.3 Evaluation

To evaluate and visualize, the trained model predicted labels for the testing dataset. Then they were compared with the true labels and plotted for the same graph 9.3. Label0 to label3 are creating final health indicator with size 1x4.

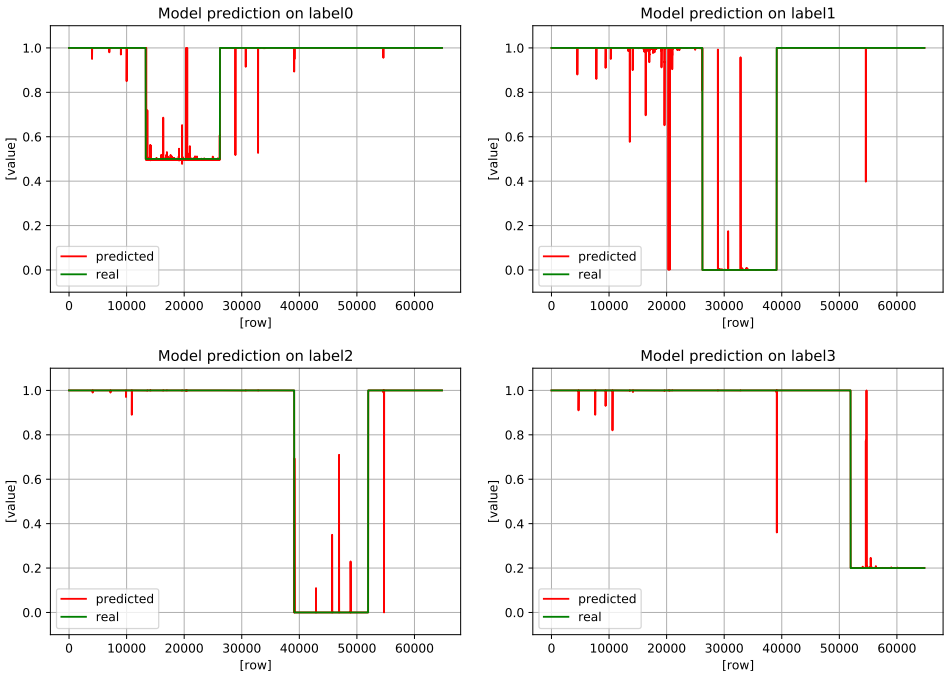


Figure 9.3: Evaluation of real and predicted health indicator

The testing dataset was merged in the same order as the order of rows in table 9.1. For example, the first part of the dataset is the healthy data. So the first fifth of all 4 indicators is expected to be 1. The last fifth of data stands for motor coil resistance increase. In this case, the 3 indicators are expected to be 1 and the fourth one to be 0.2. If the model was 100% correct, we would see just one line in each graph, as they would be identical. Now the red line stands for an improper value of the health indicator.

To compare, the supervised approach is more accurate and decisions oscillate less than the unsupervised one. Its health indicator conveniently shows us, which part of EMA is defected. On the other hand, it is necessary to collect measurements of failure modes, which increases the time and money cost. The supervised approach also doesn't cover all possible failures and some crucial ones may remain undetected.

Chapter 10

Reduced unsupervised approach

The unsupervised approach brought us fairy good results. However, its implementation in some environments would be difficult due to excessive external vibrations. In our case, if the EMA would be deployed to an aircraft with a mounted 3-axis accelerometer, it might work on the ground, but will not work in the air. Unexpected external vibrations caused by inconsistent air turbulence would bury vibration signals in the noise. The measurement of temperatures would be also challenging due to enormous external temperature changes caused by elevation gain.

This led us to think of measuring just electrical values - voltage and current consumption of EMA. These electrical units shouldn't be affected in such difficult conditions.

Again our goal was to build a GAN producing a health indicator in range <0;1>. Healthy conditions should be indicated by values close to 1. Worsening of EMA's health should result in lower values down to 0.

10.1 Training dataset

The same training dataset as in the unsupervised approach was used except for features. The raw voltage and current signal were sampled at 200kHz. To decrease dataset size, voltage and current were described by 6 features each ending up with a frequency of 1kHz. RMS, crest factor, mean, variance skewness, and kurtosis were computed for every 200 windows of the original signal. Other units (force, motion, vibrations, and temperatures), were dropped out of the dataset. Then the final dataset ended up having just 12 features. The dataset consists of 12 measurements for different load conditions and motion profiles. Each measurement duration is 40 seconds.

10.2 Testing dataset

The testing dataset also consists of 12 current and voltage features. It was merged from 5 scenarios - no defect, gearbox tooth defect, gearbox missing tooth, motor winding short, and motor coil resistance increase. Training and testing datasets were normalized to range <-1;1> and randomly shuffled.

10.3 Network architecture

The unsupervised GAN architecture needed to be adjusted. Due to a different number of features, each layer's dimensions were changed. The generator is generating an output of 28 time steps for 12 features. The discriminator inputs the generated output and creates a single float value health indicator. The number and types of layers remained unchanged. Detailed discriminator B.6 and generator B.5 NN architecture is depicted in the appendix.

The network was trained in different conditions with batch size 256 and binary cross-entropy loss. In this case, it was harder to reach convergence, so we didn't use identical learning rates for both networks. The generator was overpowering discriminator. To solve this problem, we increased the learning rate of the discriminator to 0.0001 and decreased the learning rate of the generator to 0.00009. This setup showed to work better. The training loss is depicted in figure 10.1. The best performing solution was in epoch 4, where the losses reached similar values.

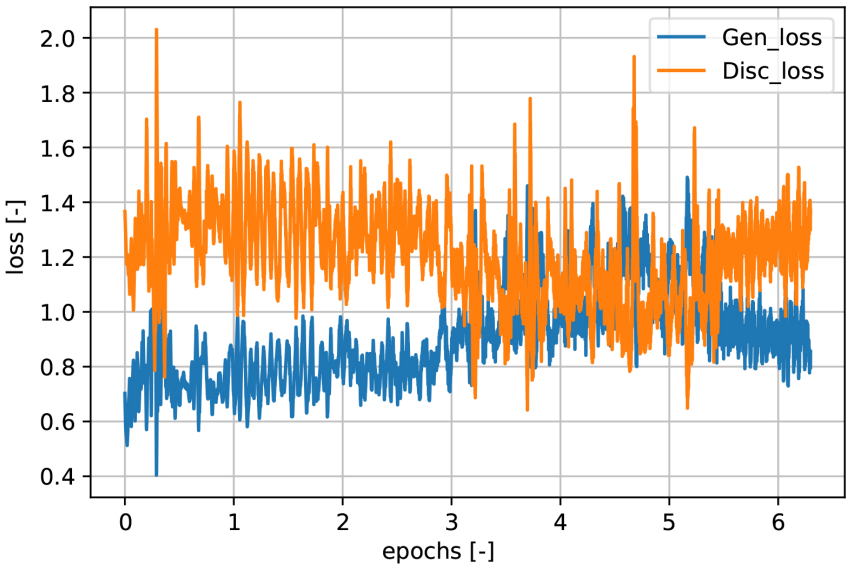


Figure 10.1: Example of RGAN loss

10.4 Evaluation

To be able to compare the GAN with the reduced GAN approach, we kept the evaluation metrics identical for both. In the case of reduced GAN, we expected better performance in electric failures. On the other hand, we were skeptical, if the network would be able to learn also non-electric failures and if yes, to which extent.

10.4.1 Generator evaluation

The performance of generators was measured by using Wasserstein distance. This distance represents the distance of two probability distributions and shows how well was generator able to learn the distribution of the EMA healthy operation.

Histograms for each feature were plotted to compare the distribution of measured and generated data. For illustration, two histograms of the best generator are attached - 10.2, 10.3. The current crest factor histogram 10.2 shows us a well-learned distribution. The distribution of the current mean feature 10.3, is sufficient enough to distinguish higher values from a healthy operation, but simplified. We can see, that the healthy operation values in range <-0.75;-0.50> are not covered. Unlike the GAN generator, reduced GAN generator output was more distant from the measured healthy data distribution.

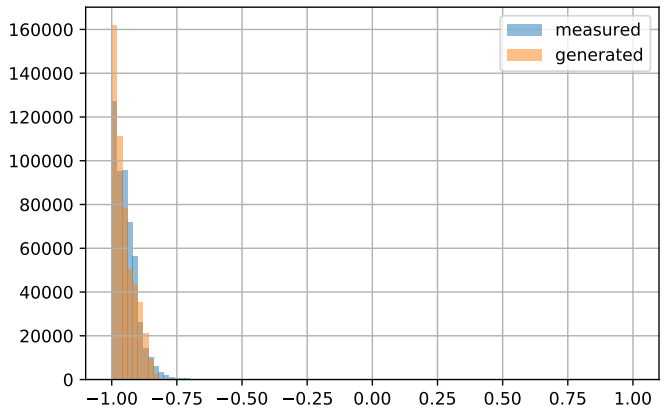


Figure 10.2: Histogram of measured and generated values for current crest factor

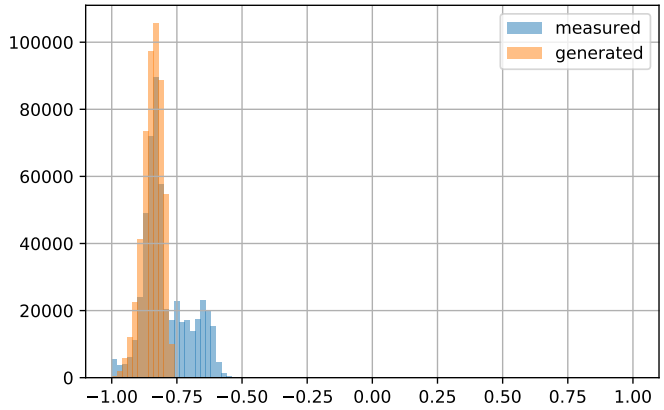


Figure 10.3: Histogram of measured and generated values for current mean

10.4.2 Discriminator evaluation

The health indicator is the direct output of the discriminator and is in range <0;1>. Higher values represent a better machinery health state. To evaluate we picked 768 decisions for each dataset in different conditions. The evaluation metric took into account the distance of failure data decisions from the no defect one and the variance of each mode decision. The best model was chosen based on the lowest sum of variances and the highest distance.

The average health indicator for each dataset is stored in the table 10.1. We can see a gap between no defect and failure data and as expected, the lowest values have the electrical failure datasets.

Dataset	Trained model
no defect	0.612726
gearbox tooth deformation	0.2377
gearbox missing tooth	0.2045
motor winding short	0.1211
motor coil resistance increase	0.1629

Table 10.1: Average health indicator

To compare with the GAN approach, the health indicator for no defect data has here much higher variance 10.4. Even though the average healthy data decision is 0.61, its values range from 0.38 to 0.91. The reduced GAN is less confident about the healthy state and to increase the accuracy, we should average more decisions.

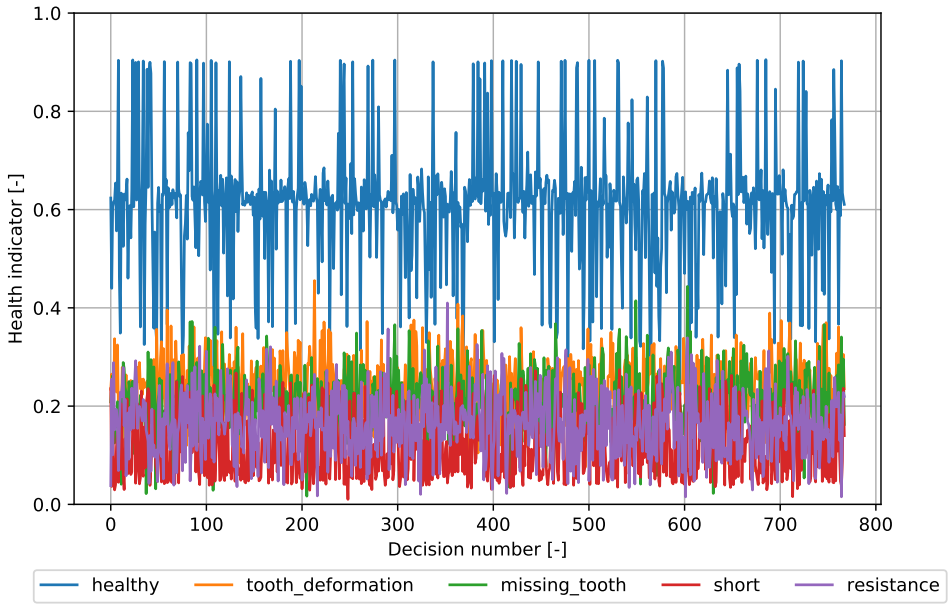


Figure 10.4: Decisions (HI) of trained reduced GAN discriminator

The motor short dataset has the lowest HI 10.5, which can be explained by the visualization of the data itself. Plotting measured data in various conditions showed us, that the motor short data has significant distinct current values in every measurement, which explains the lowest HI value.

The second-lowest average values were unsurprisingly reached by the motor coil resistance increase dataset. The changes in this data cannot be seen in every measurement setup, but in most of them are significant enough.

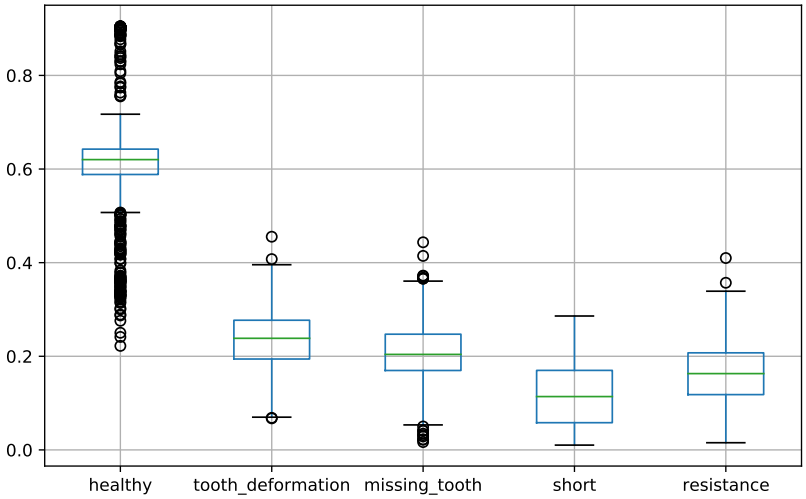


Figure 10.5: Boxplot of trained reduced GAN discriminator

The tooth deformation and missing tooth HI results surprised us. We assumed, that the reduced GAN without vibration data wouldn't be able to distinguish them so well. But as we can see in figure 10.5, their HI values are comparable with the electric failure ones. To understand this problem better, we visualized all 5 datasets for different load and motion profile conditions. Afterward, we realized, that the non-electric failures are also reflected in the electrical signals, which explains the well-performing HI for tooth deformation and missing tooth cases.

The figure 10.6 shows us the current graph for 5 failure modes in the same conditions - triangle motion profile and 150N load. As mentioned before, we can see the significant change of current signal for the motor winding short. On the other hand, the motor coil resistance increase is not so significant under these conditions. The main reason for choosing this case is to demonstrate the enormous change in the current signal for missing tooth in one direction of movement. And after a closer look, we can also see the signal change in the case of tooth defect. In contrast to healthy operation, current peaks are higher and last longer.

Each failure is more visible within different conditions. In this case, the missing tooth dataset current is enormously increased in one direction. Within another condition, it can be barely noticed by the human eye, which can also explain the variance of decisions - more significant signal change under some conditions, more confident HI classification, and vise versa.

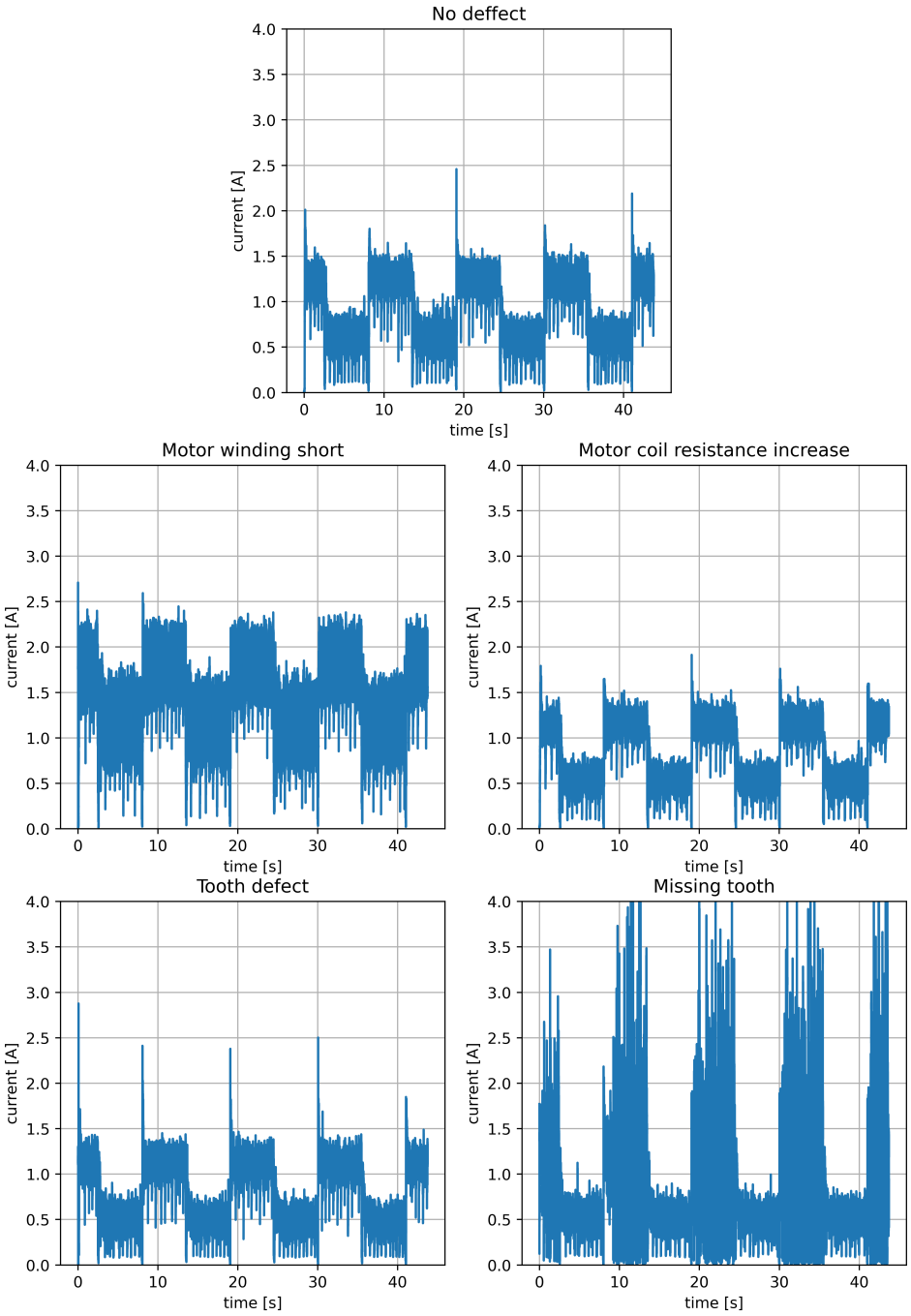


Figure 10.6: Current graph for load 150N and triangle motion profile

Chapter 11

Conclusion

The aim of the master thesis and prior master thesis project was to study deep digital twins and implement supervised and unsupervised deep digital twins for linear electromechanical actuator DSZY1 based on a measured dataset.

In the beginning, linear electromechanical actuators and their vulnerabilities were studied. Later an overview of possible digital twin approaches was introduced, focusing on part of data-drive approaches, especially artificial neural networks solutions. The raw measured dataset was preprocessed. To decrease the size and increase information gain, the sampling rate was decreased and descriptive features were extracted. As a part of the implementation, supervised LSTM-based, unsupervised GAN-based, and reduced unsupervised GAN-based solutions were proposed. In the unsupervised approaches, own customized evaluation metrics were introduced and all 3 solutions were tested. The advantage of the supervised approach is the ability of fault classification. It is not possible by unsupervised approaches, but they are trained just on healthy operational data, which brings a huge advantage on data acquisition. The reduced unsupervised approach is trained and tested just on motor voltage and current data, which solves the problem of difficult measuring conditions for vibrations and temperatures in an aircraft.

There still exists plenty of ways for improvement and deeper research. For example, in terms of unsupervised approach, finding a way of component-related fault classification would be an interesting topic for further investigation. Now, we possess only the overall health indicator, but a health indicator having the information of component degradation would have a great industry potential as the unsupervised approach can be trained just on operational data of assets.

We believe, that with our reduced unsupervised GAN approach, we contributed to a developing movement of AI fault diagnosis. To emphasize, our neural network is creating a health indicator trained just on healthy operation current and voltage data and creates a health indicator of overall state. This proves a high potential in this approach for further research and later industry deployment.

Appendix A

List of acronyms and symbols

Acronym/symbol	Meaning
ACE	Actuator Control Electronics
AI	Artificial Intelligence
ANN	Artificial Neural Network
BG	Bond Graph
CNN	Convolution Neural Network
DDT	Deep Digital Twin
DFT	Discrete Fourier Transform
EMA	Electro-Mechanical Actuator
FFT	Fast Fourier Transform
FD	Fault Detection
FID	Frechet Inception Distance
GAN	Generative Adversarial Network
HI	Health Indicator
LSTM	Long Short Term Memory
LCL	Lower Control Limit
MEA	More Electric Aircraft
MSE	Mean Square Error
NN	Neural Network
PDE	Power Drive Electronics
PCA	Principle Component Analysis
PHM	Prognostics and Health Management
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
UCL	Upper Control Limit

Appendix B

Figures

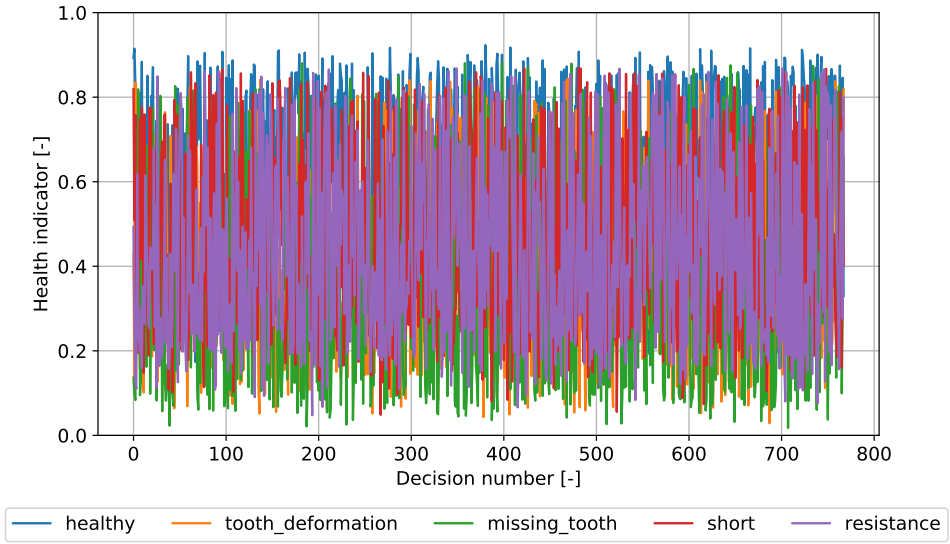


Figure B.1: Decisions (HI) of collapsed discriminator

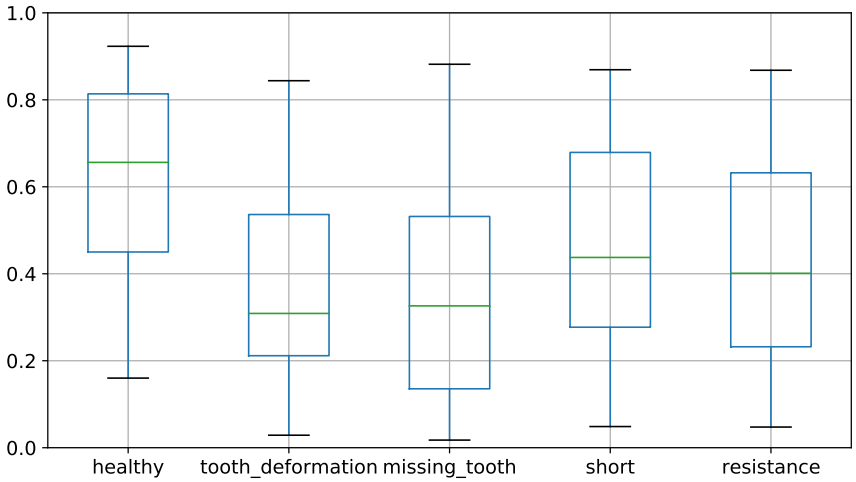


Figure B.2: Boxplot of collapsed discriminator

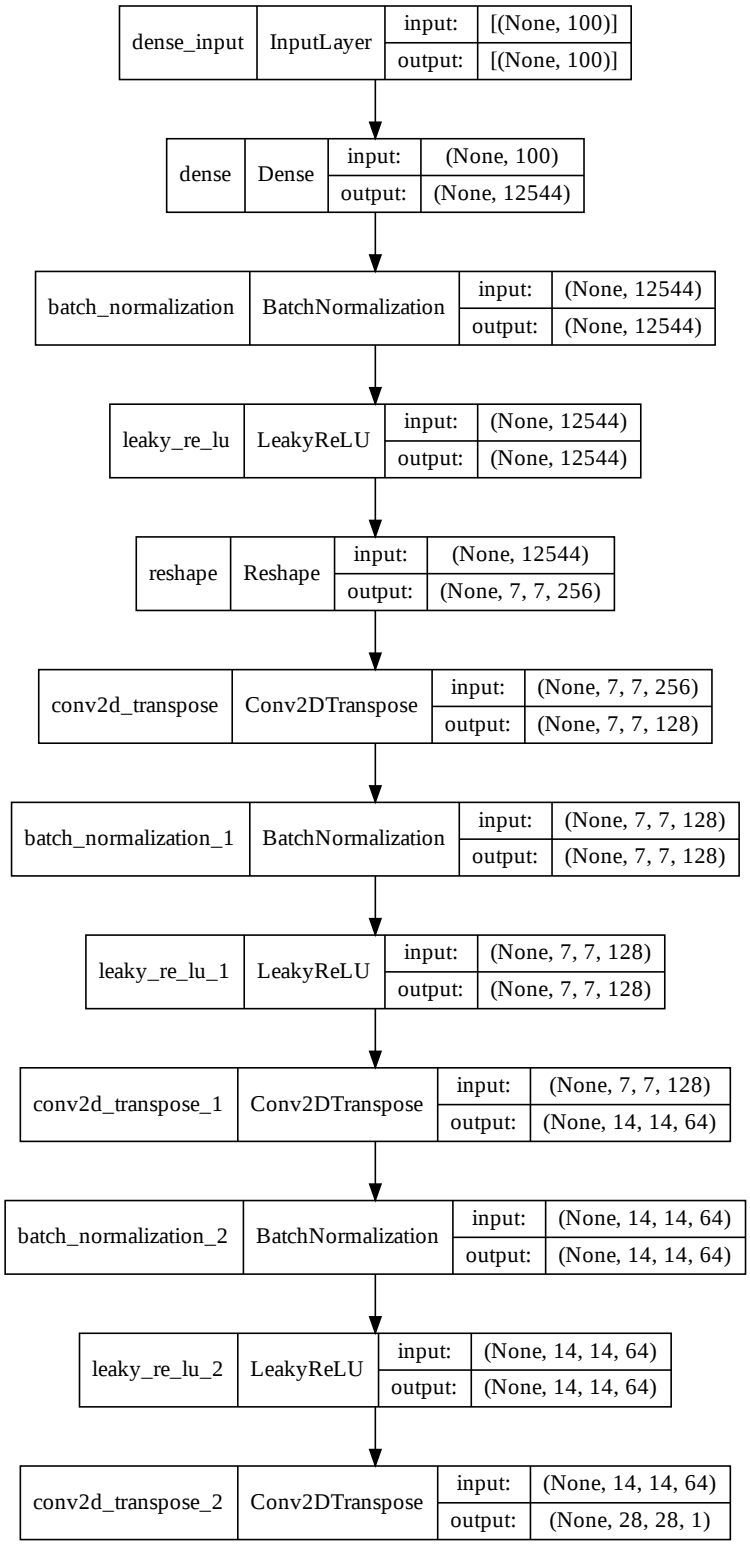


Figure B.3: Generator model (None stands for arbitrary batch size)

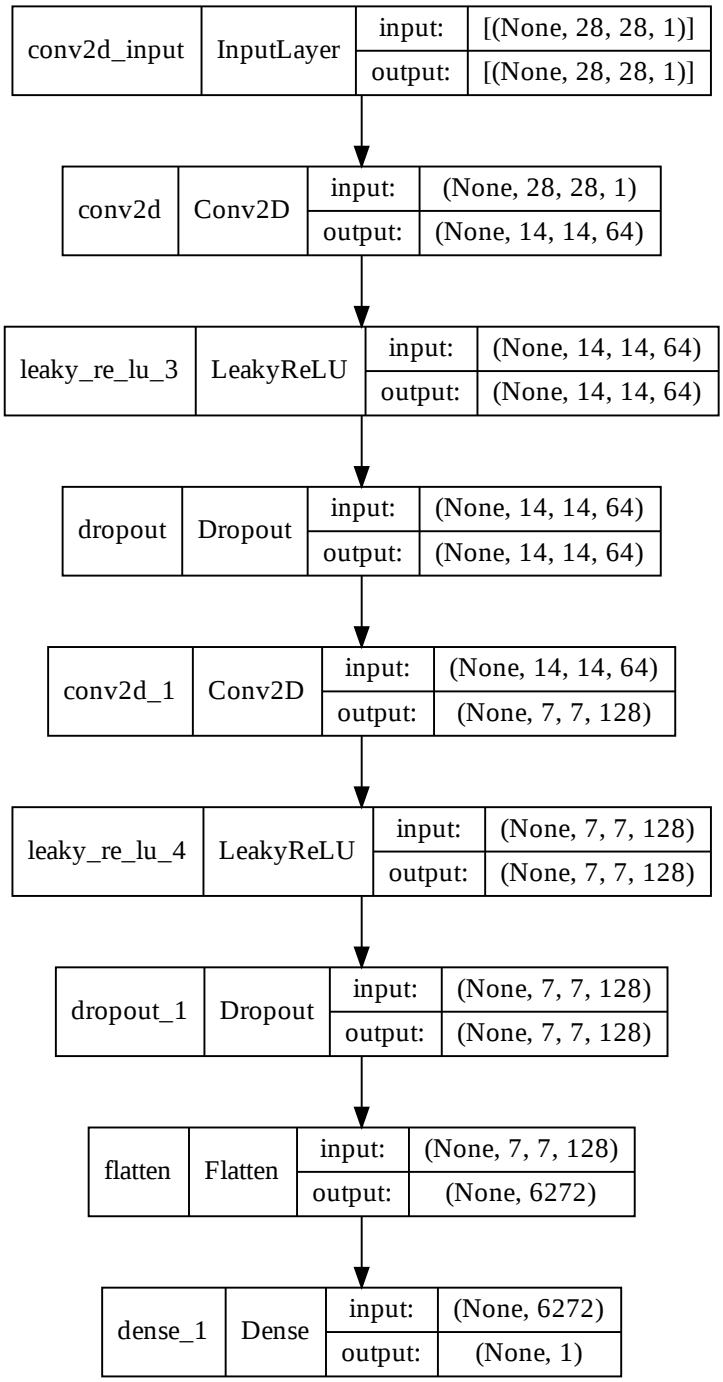


Figure B.4: Discriminator model (None stands for arbitrary batch size)

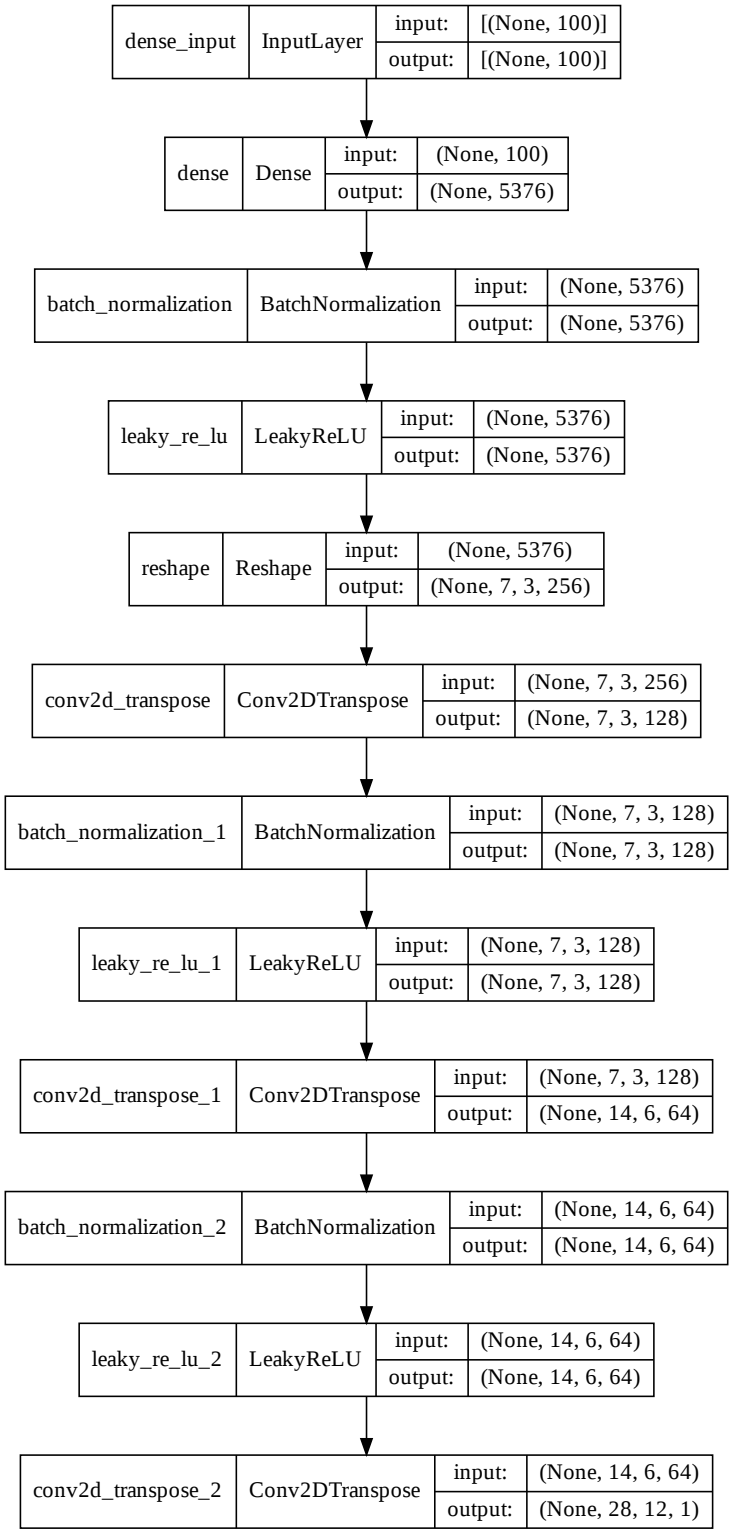


Figure B.5: Reduced generator model (None stands for arbitrary batch size)

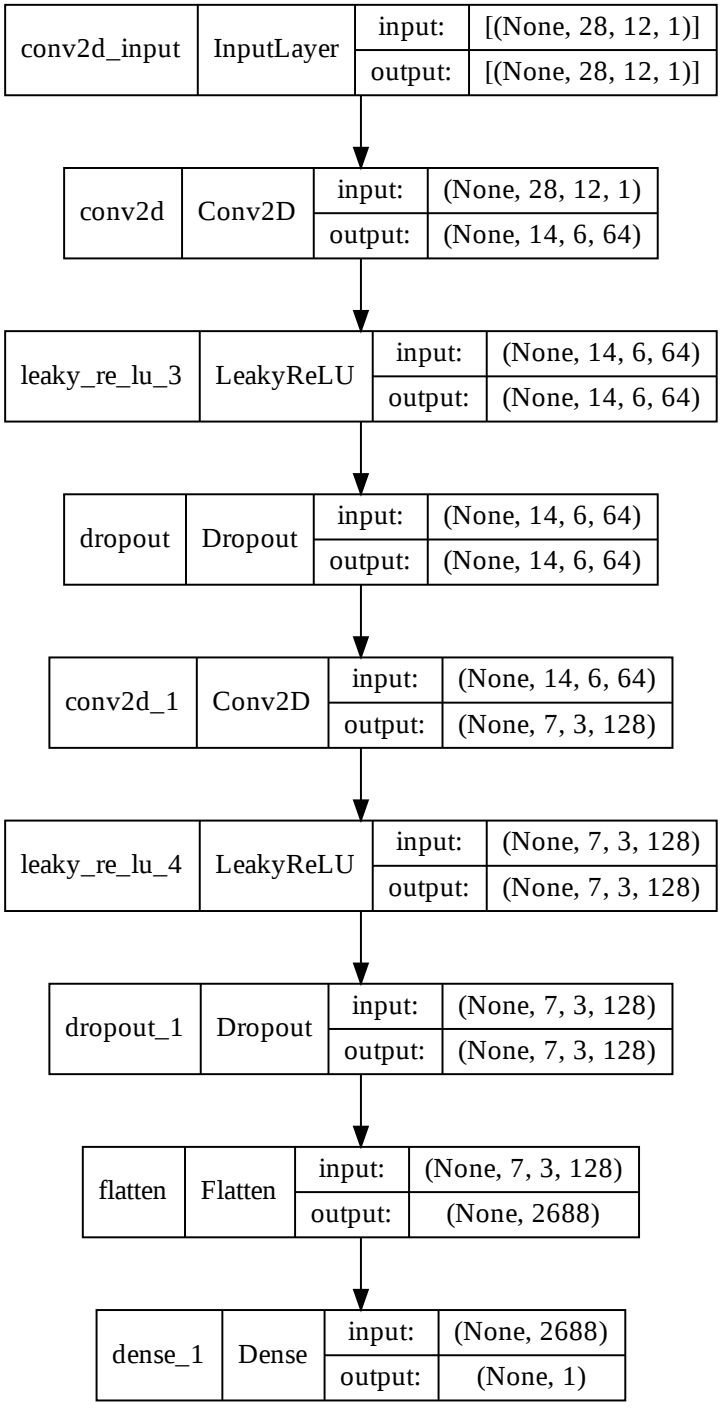


Figure B.6: Reduced discriminator model (None stands for arbitrary batch size)

Appendix C

Bibliography

[1] A.Bonnett and Ch.Young, *Explaining motor failure*, ECM, (2004).

[2] W. Booyse, D. N. Wilke, and S. Heyns, *Deep digital twins for detection, diagnostics and prognostics*, Mechanical Systems and Signal Processing, 140 (2020), p. 106612.

[3] F. Bre, J. Gimenez, and V. Fachinotti, *Prediction of wind pressure coefficients on building surfaces using artificial neural networks*, Energy and Buildings, 158 (2017).

[4] J. Brownlee, *Introduction to Generative Adversarial Networks*, Machine Learning Mastery, 2019. https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/.

[5] Colah, *Understanding LSTM networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[6] C.S.Byington and P.Stoelting, *A model-based approach to prognostics and health management for flight control actuators*, IEEE Aerospace Conference Proceedings, (2004).

[7] D.H.Hubel and T.N.Wiesel, *Repetive fields and functional architecture of monkey striate cortex*, (1967).

[8] Drive-System Europe Ltd., *Linear actuator DSZY1 datasheet*. https://www.drive-system.com/pdf/en/DC_Linearactuator_DSZY1_encoder.pdf.

[9] E.Balaban et al., *A diagnostic approach for electro-mechanical actuators in aerospace systems*, IEEE Aerospace conference, (2009).

[10] I. El-Thalji, *Dynamic modelling and fault analysis of wear evolution in rolling bearings*, PhD thesis, 05 2016.

[11] Entsoe, *Digital twin*. https://www.entsoe.eu/Technopedia/techsheets/digital-twin.

[12] J. Feng, X. Feng, J. Chen, X. Cao, X. Zhang, L. Jiao, and T. Yu, *Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification*, Remote Sensing, 12 (2020), p. 1149.

[13] Firgelli Automations, *Linear actuators*. https://www.firgelliauto.com/blogs/news/linear-actuators-101.

[14] J. Gertler, *Fault detection and isolation using parity relations*, Control Engineering Practice, 5 (1997), pp. 653–661.

[15] A. Gibson, *DC motors at high temperatures*, Drive Tech. https://drive.tech/en/stream-content/can-dc-motors-be-used-at-high-temperatures.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in neural information processing systems, 27 (2014).

[17] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

[18] Google, *Common GAN problems*. https://developers.google.com/machine-learning/gan/problems.

[19] ——, *Deep Convolutional Generative Adversarial Network in Tensorflow*. https://www.tensorflow.org/tutorials/generative/dcgan.

[20] ——, *GAN loss functions*. https://developers.google.com/machine-learning/gan/loss.

[21] ——, *TF-GAN tutorial*. https://colab.research.google.com/github/tensorflow/gan/blob/master/tensorflow_gan/examples/colab_notebooks/tfgan_tutorial.ipynb?utm_source=ss-gan&utm_campaign=colab-external&utm_medium=referral&utm_content=tfgan-intro.

[22] ——, *Time Series Forecasting*. https://www.tensorflow.org/tutorials/structured_data/time_series.

[23] S. Haag and R. Anderl, *Digital twin – proof of concept*, Manufacturing Letters, 15 (2018), pp. 64–66. Industry 4.0 and Smart Manufacturing.

[24] O. Hanuš and R. Šmíd, *Poruchy aktuátorú pro obecné letecké použití*, FEL ČVUT, 2019.

[25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, Advances in neural information processing systems, 30 (2017).

[26] J. Jordan, *Introduction to autoencoders*, 2018. https://www.jeremyjordan.me/autoencoders/.

[27] M. Kramer, *Autoassociative neural networks*, Computers Chemical Engineering, 16 (1992), pp. 313–328. Neutral network applications in chemical engineering.

[28] K.Tidriri et al., *Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges*, Annual Reviews in Control, (2016).

[29] M. Lebold, K. Mcclintic, R. Campbell, C. Byington, and K. Maynard, *Review of vibration analysis methods for gearbox diagnostics and prognostics*, (1985).

[30] D. S. Lee, J. M. Park, and P. A. Vanrolleghem, *Adaptive multiscale principal component analysis for on-line monitoring of a sequencing batch reactor*, Journal of Biotechnology, 116 (2005), pp. 195–210.

[31] Machine learning mastery, *Generative Adversarial Networks loss functions*. https://machinelearningmastery.com/generative-adversarial-network-loss-functions/.

[32] Matlab, *Digital twin*. https://www.mathworks.com/discovery/digital-twin.html.

[33] K. Medjaher and N. Zerhouni, *Hybrid prognostic method applied to mechatronic systems*, The International Journal of Advanced Manufacturing Technology, (2013).

[34] M.T.Amin et al., *Fault detection and pathway analysis using a dynamic bayesian network*, Chemical Engineering Science, (2019).

[35] Ohio Carbon Industries, *Carbon Brush Troubleshooting Guide*. https://www.ohiocarbon.com/upload/documents/oci_brush_troubleshooting_guide.pdf.

[36] OMNI.SCI, *Predictive maintenance*. https://www.omnisci.com/technical-glossary/predictive-maintenance.

[37] P.Němeček and E.Tomeh, *Vibrační diagnostika základních závad strojů*, Technická univerzita v Libereci. http://www.kvm.tul.cz/getFile/id:1851/Skripta%20Diagnostika%20z%C3%A1kladn%C3%ADch%20z%C3%A1vad%20N%C4%9B-To.pdf.

[38] A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, arXiv, (2015).

[39] R. B. Randall, *Vibration–based Condition Monitoring*, John Wiley Sons, Ltd, 2021.

[40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, *Improved techniques for training gans*, Advances in neural information processing systems, 29 (2016), pp. 2234–2242.

[41] SIM, *Bond Graphs*. https://www.20sim.com/webhelp/modeling_tutorial_bond_graphs_bondgraphs.php.

[42] T. Spinner et al., *An Intuitive Comparison of Autoencoders with Variational Autoencoders*, University of Konstanz. https://thilospinner.com/towards-an-interpretable-latent-space/.

[43] S.Schmidt et al., *Model identification using stochastic differential equation grey-box models in diabetes*, Journal of Diabetes Science and Technology, (2013).

[44] STMicroelectronics, *VNH7070AS datasheet*. https://www.st.com/en/automotive-analog-and-power/vnh7070as.html.

[45] I. Tabian, H. Fu, and Z. Sharif Khodaei, *A convolutional neural network for impact detection and characterization of complex composite structures*, Sensors, 19 (2019).

[46] T.Escobet and L. Travé-Massuyès, *Parameter estimation methods for fault detection and isolation*, (2001).

[47] Y.Cao et al., *Fault tree analysis of electro-mechanical actuators*, IEEE Aerospace conference, (2009).

[48] R. Šmíd, *B3M38DIT - stránky predmetu Diagnostika a testování*, ČVUT FEL. https://moodle.fel.cvut.cz/courses/B3M38DIT.