Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Diploma Thesis

Data-Driven Modelling of a Fixed Bed Regenerator using Machine Learning in Tensorflow

carried out for the purpose of obtaining the degree of Master of Science (MSc or Dipl.-Ing. or DI), submitted at TU Wien, Faculty of Mechanical and Industrial Engineering, by

Elisabeth Gütl

Mat.Nr.: 01014231

under the supervision of

Univ.Prof. Dipl.-Ing. Dr.techn. René Hofmann

Dipl.-Ing. Verena Halmschlager

Institute for Energy Systems and Thermodynamics

Signature

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Vienna, 21.10.2019

Signature

Acknowledgement

I would first like to thank my thesis advisors Univ.Prof. Dipl.-Ing. Dr.techn. René Hofmann and Dipl.-Ing. Verena Halmschlager. They allowed this paper to be my own work, but gave me helpful hints to push me into the right direction whenever I needed it. As well must I express my very profound gratitude to my parents, to my friends and boyfriend for providing me with their support and encouragement throughout my years of study until now. This accomplishment would not have been possible without them.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

The increasing digitalization influences our daily lifes and leads to a changing working world and society. The application areas of digital technologies are wide and not always directly evident. This thesis gives an example, how digitalization technologies can be applied in the energy sector. As a use case, a fixed bed regenerator, a sensible thermal energy storage, is used. On the one hand, a data-driven model of the storage is built by using Machine Learning in Tensorflow. On the other hand it is explained how the deployment of a Machine Learning model into production works. An introduction to the reference architecture model RAMI-4.0 is given, which describes the implementation of industry 4.0 methods. In order to create a data-driven model that predicts the loading and unloading cycles of the fixed bed regenerator, the modelling technique "Long Short-Term Memory" (LSTM), a kind of Artificial Neural Network is used. In total, three different LSTM models of the storage are built. Although the LSTM models manage to follow the main trend in the data, they predict the course of the true curves with considerable deviations. More data and different data-driven modelling approaches might be required to achieve more accurate predictions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Die zunehmende Digitalisierung wirkt sich auf unser tägliches Leben aus und führt zu einer sich verändernden Arbeitswelt und Gesellschaft. Die Anwendungsbereiche von Digitalisierungsmethoden sind breit und nicht immer direkt ersichtlich. Diese Arbeit stellt einen konkreten Anwendungsfall dar, wie Digitalisierung im Energiesektor einsetzt werden kann. Als Anwendungsfall dient ein Festbettregenerator, ein sensibler thermischer Energiespeicher. Auf der einen Seite wird ein datengetriebenes Modell des Energiespeichers mithilfe von maschinellem Lernen in Tensorflow erstellt. Auf der anderen Seite wird gezeigt, wie die Bereitstellung eines maschinellen Lernmodells in die Produktion funktioniert. Eine Einführung in das Referenzarchitekturmodell RAMI-4.0 wird gegeben, das die Implementierung von Industrie 4.0 Methoden beschreibt. Um ein datengetriebenes Modell zu erstellen, das die Lade- und Entladezyklen des Festbettregenerators vorhersagt, wird die Modellierungstechnik "Long Short-Term Memory" (LSTM) verwendet, eine Art von künstlich neuronalem Netzwerk. Insgesamt werden drei verschiedene LSTM-Modelle des Energiespeichers erstellt. Die LSTM-Modelle folgen zwar dem Haupttrend in den Daten, sagen jedoch den Verlauf der wahren Kurven mit erheblichen Abweichungen voraus. Möglicherweise sind mehr Daten und andere datengetriebene Modellierungsansätze erforderlich, um genauere Vorhersagen zu erzielen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Abbreviations

API	Application Programming Interface 12
AI	Artificial Intelligence 2
CPS	Cyber-Physical Systems 6
IDE	Integrated Development Environment 37
IoT	Internet of Things 5
ML	Machine Learning 2
OPC UA	Open Platform Communications Unified Architecture 11
PLC	Programmable Logic Controller 5
JSON	JavaScript Object Notation 27
RAMI-4.0	Reference Architecture Model Industry-4.0 3
REST	REpresentational State Transfer 27
RNN	Recurrent Neural Network 21
SOA	Service Oriented Architecture 11

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Symbols

Symbol	Unit	Description
Q	J	Amount of heat
m	kg	Mass of the storing material
c_p	J/(kgK)	Heat capacity of the storing material
T_2-T_1	K	Temperature difference
E_{Exergy}	J	Usable energy
T_S	K	Temperature of the storage
T_U	K	Ambient temperature
h(x)	-	Sum of weighted inputs
w_i	-	Weights
x_i	-	Input
x_{scaled}	-	Scaled value of a value x
x	-	Value that is going to be scaled
min(x)	-	Minimum value in the dataset
max(x)	-	Maximum value in the dataset

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Chapter 1

Introduction

1.1 Motivation

Digitalization is more and more becoming a part of industrial processes. To quote Peter Sondergaard from *Gartner*, an information technology research and advisory company:

> *Information is the oil of the 21st century, and analytics is the combustion engine.*
>
> (Peter Sondergaard)

The internal combustion engine was the major innovation in the 19th century and was further developed and mass produced in the 20th century. As for the 21st century, data will be one of the most valuable *resources*. Data is nowadays available in an unimaginable amount. Every sensor, every executed industrial process produces data. The digital economy is growing rapidly and has already surpassed the growth of more traditional branches. To be part of the digital transformation of economy and society promises profit and success. Companies who refuse to apply digital technologies and who cling on their traditional working environment will loose the connection to this new digital world.

Digitalization opens up whole new application areas. Predictive maintance, safety, digital twinning, value and assets maximization, to name a few. With the upcoming of cheaper

hardware and the further development of special approaches like Artificial Intelligence (AI), implementation of digital technology has also never been that easy. One of the keypoints of digitalization is the connection of machine and human to increase efficiency of working processes. Methods of a connection between machine and human are provided by industry 4.0.

In the world of digitalization everything is about data. But data is only valuable if used for analysis to gain information. Especially the applying of viable models to the data is one of the core elements of data analysis. Particularly the energy sector offers many areas in which the modelling of a certain problem can be useful, time and cost saving. In times of energy transition, the focus lies on intelligent interconnection of processes, known by the name smart grid, smart factory or smart city, to save energy that could be spared by intelligent switching processes. Another topic associated with the term energy transition, is the concept of using renewable energies for generating electricity. Renewable energies like wind or photovoltaic, go hand in hand with a volatile power supply because of the fluctuating energy production. A potential energy equalization can be achieved by an energy storage device that can provide a part of the base load at peak times.

Since energy storages not only play a role in energy transition, but are also used in other industrial applications, for instance the usage as air preheater in industrial processes, it is worthwhile to take a closer look at the data of an energy storage system and to model such an energy storage device to be able to make more precise statements about its behaviour. In general, there exist many kinds of modelling methods for data, which all have their advantages and disadvantages. Among them, data-driven models have gained importance and especially a technique of Machine Learning (ML), Artificial Neural Networks (ANNs), seems to deliver good results within a reasonable time frame and are able to recognize patterns and regularities in data.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1.2 Aim & Scope

The aim of this master thesis is to build a data-driven model of a sensible thermal energy storage device, namely a fixed bed regenerator (FBR), and to give an overview on how digital technologies can be applied in the energy sector. Common terms like digitization, industry 4.0 or data-driven modelling are explained. Another focus lies on the Reference Architecture Model Industry-4.0 (RAMI-4.0), to explain how industry 4.0 technologies can be more easily implemented in industrial processes. The measured load cycles of the FBR from the Institute for Energy Systems and Thermodynamics at the Technical University of Vienna are used for modelling the FBR. To create a data-driven model of this energy storage, an Artificial Neural Network is used. In this way, the applicability of this modelling approach for industrial applications is evaluated and positive and negative aspects can be discussed.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Chapter 2

Theoretical Background

In this chapter, first, commonly used terms are described which are further used in this thesis. An introduction to industry 4.0, digitalization and data is given, followed by a description of a standardized reference architecture for industry 4.0, RAMI-4.0.

2.1 Common Phrases

API "An Application Programming Interface (API) is a (hypothetical) contract between 2 softwares saying if the user software provides input in a pre-defined format, the later with extend its functionality and provide the outcome to the user software." [7]

AI "Artificial Intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning, reasoning and self-correction." [13]

Big Data "Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation." [8]

Cloud computing "The practice of using a network of remote servers hosted on the internet to store, manage and process data, rather than a local server or a personal computer." [13]

Digitalization "Digitalization is the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business." [8]

Digitization "The conversion of analogue to digital." [13]

Learning algorithm "A process or method used to extract patterns from data collection to identify and adapt appropriate solutions or applications for a device or system." [13]

ML "Machine learning (ML) is a branch of artificial intelligence that systematically applies algorithms to synthesize the underlying relationships among data and information." [19]

2.2 Industry 4.0, Data, Digitalization

The evolution of technology has changed the way we live our lives and shapes our modern world. But not long time ago, many technologies which we take nowadays for granted, did not exist. With the onset of the industrial revolution, technological progess was accelerating and is still in progress today. To illustrate how much the industrial revolution has characterized today's society, the stages of the industrial revolution are briefly explained. The industrial revolution can be divided into four different phases, as seen in fig. 2.1.

The first industrial revolution was characterized by the usage of steam power and the invention of the vapor combustion engine and ushered in the machine age. The second industrial revolution, also called the *Technological Revolution*, led to the principle of mass production. The third industrial revolution already takes a step into the direction of the fourth industrial revolution and is called the *digital revolution*. It is characterized by the development of Programmable Logic Controller (PLC) and the automation of production lines. The term *fourth industrial revolution* describes the connecting of machines and humans and their communication via a cloud-based platform, the Internet of Things (IoT) [11]. The fourth industrial revolution opens up many possibilities to increase the efficiency of processes and to save costs and can increase flexibility in the economic competition.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

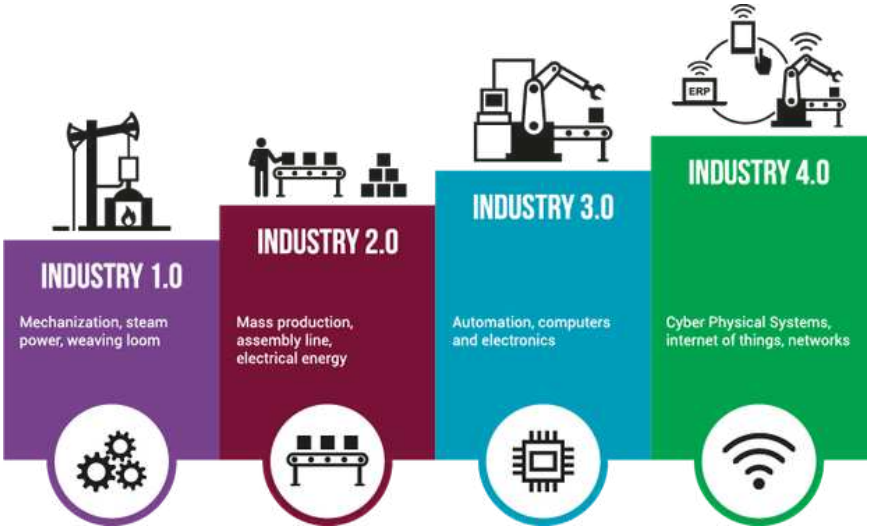

Figure 2.1: The four phases of the industrial revolution [9]

One of the core elements of the fourth industrial revolution is the use of digital technologies (digitalization) to digitize the production environment to a so called *smart factory*, that describes a production environment in which the factory organizes itself with low human intervention [25]. A smart factory is for instance characterized by a product communicating on its own with the manufacturing plant. Machines *know* when the next maintenance is due, due to sensors that pass on data to the cloud, for instance from malfunctioning. Robots work independently on a task at a working station and when finished, know via cloud-based platform which working station needs their help next.

Cyber-Physical Systems (CPS) play a central role in establishing a smart factory. CPS describes the connection of the physical world of machines and devices with the virtual world, the Cyberspace. This linkage is possible through the use of sensors and actuators. Sensors measure physical quantities and pass on these measurement data to a software. The required action is determined from the software and passed on to the actuators via the network [16]. An important point of a smart factory is the uniform communication of machines and devices and how methods of industry 4.0 can be integrated into an existing factory. An approach is provided by the three-dimensional Reference Architecture Model Industry-4.0 (RAMI-4.0).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.3 RAMI-4.0

The basis for a successful implementation of industry 4.0 concepts requires a standardized reference architecture. The RAMI-4.0 as a standardized reference architecture is used to ensure flexible usage of industry 4.0 and to make an easy implementation of components as part of an industrial process possible. The model was created by the founded platform industry 4.0, which is composed of the ZVEI, the VDMA and the Bitkom. The RAMI-4.0 model is a three-dimensional model with three axes. As showed in fig. 2.2, the three axes are: Layers, Life Cycle & Value Stream and Hierarchy Levels. Used items of the industry

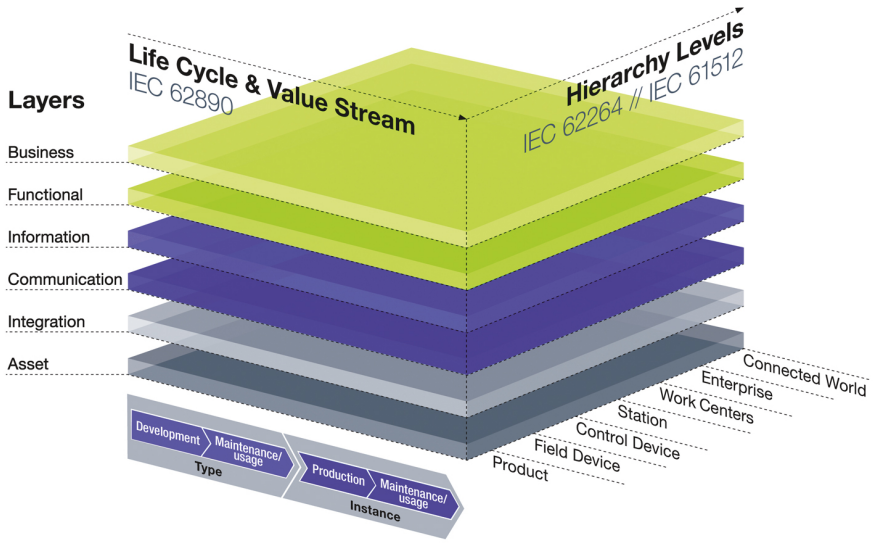

Figure 2.2: The three dimensional RAMI-4.0 [12]

4.0 process are called *assets*. These items do not need to be physical items, but they can also include software, patents and more [20]. RAMI-4.0 was among others introduced, because the former master-slave relationship was considered as outdated and unflexible. The term master-slave comes from the field of data transmission and explains the former way of transmission. One master regulates the sequence of data transmission from many slaves and sorts the sequence of transmission by importance. RAMI-4.0 enables a mutual communication without a parent master, where every asset as an participant of the process offers his services and can also utilize another assets offered services [33].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.3.1 Layers

The vertical layer axis consists of six layeres describing the properties of an asset and its functions within the connected IT-world. The layer axis exclusively specifies the frame for an integration of the plant into the connected world of services. Contrary to this, the Hierarchy Level on the horizontal axis describes the organization of the workflow within the plant. The layers of the vertical axis are described further in tab. 2.1.

Table 2.1: Six layers on the vertical axis of the RAMI-4.0 [33]

Layer	General description
Business	Business processes
Functional	The functions of an asset
Information	All the data of an communicating asset
Communication	How the asset communicates witihin the process
Integration	Connection between the physical asset and the digital world
Asset	The asset in the physical world

The layers of the vertical axis can be assigned to the physical or digital world. The lowest level of these layers, the asset, belongs to the physical world. The upper levels except the integration level are considered as part of the digital world. The integration layer represents the connection between physical and digital world. An example that would fit into the integration layer is an A/D converter. A measured physical value is converted into a digital signal and thus not exactly assignable [20].

2.3.2 Life Cycle & Value Stream

The Life Cycle & Value Stream, as seen in fig. 2.3, is dividied into *Type* and *Instance*. If the product belongs to a product line, the *Type* contains all the information about this product line. This includes the data and information about the whole product family. *Type* represents the Life Cycle and Value Stream of the product from the first idea until the prototype is built. *Instance* is specifically the Life Cycle and Value Stream data of the product when used [20].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

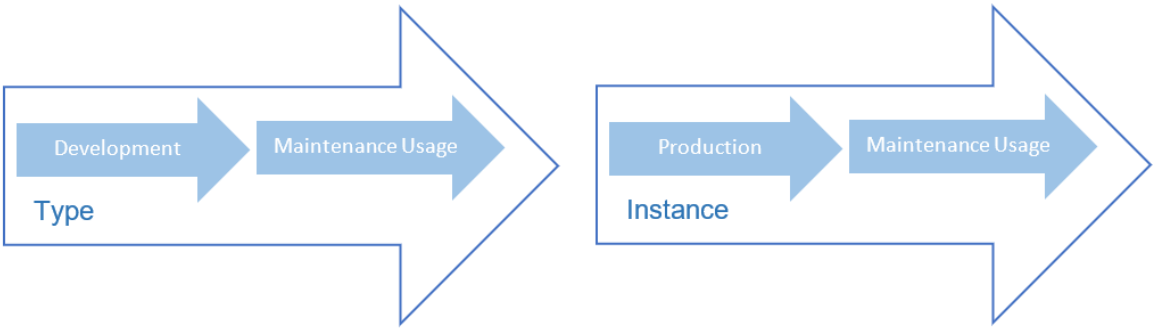

Figure 2.3: Life Cycle Value Stream, adapted from [20]

In reality, *Type* and *Instance* nearly run simultaneously but for a simplified representation, RAMI-4.0 lists it in a row one after another. The Life Cycle of *Type* collects all development data that includes design or simulation data. *Instance* contains information about maintenance and recycling as well as information about how and when it is used by customers. [33] Since *Type* and *Instance* in reality often run simultaneously, the Life Cycle and Value Stream Axis can be seen as one process which collects data from the first idea of a product until the product is recycled [20].

2.3.3 Hierarchy Level

On the second horizontal axis we find the Hierarchy Level. As already mentioned, the Hierarchy Level organizes the structure and workflow within the plant. It is closely related to the former automation pyramid, which describes the implementation of automation technology in a factory. It has been extended by three more levels: Product, Field Device and Connected World [20].

2.3.4 Administration Shell

The key principle of RAMI-4.0 is the so called administration shell, a term which was newly introduced by the working group of platform industry 4.0. Every asset in industry 4.0 has to provide a compatible object-oriented interface to ensure an easy implementation and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

standardized communication between all connected assets in the network. As shown in fig. 2.4, the asset is connected to the administration shell and provides all its information, but only the administration shell is connected to the network. That means that all the communication between assets and connected gadgets is done by administration shells [33].

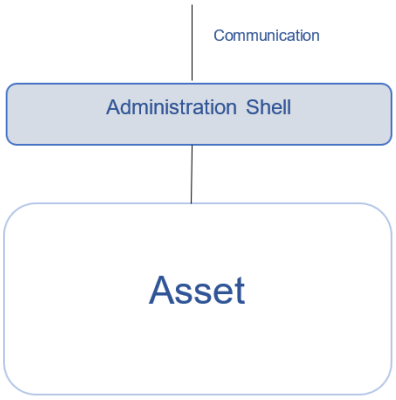

Figure 2.4: Structure of the Administration Shell, adapted from [20]

A physical item is characterized by its different attributes and features. That means in order to characterize an asset for information exchange in the network via administration shell, all information from the physical item needs to be converted into properties. This process in industry 4.0 is called property principle [20].

Administration shells can be also connected to a functional unit. To give an example: a robot can be seen as a functional unit with different components. It is possible to contact the administration shell of the robot, but it is also possible to contact the administration shell of one of the components, for example its axes [33]. It is also possible that an administration shell contains an asset with identical properties. To take up the example from the robot again: if we decide to use the property *material*, the property can be identical for many components of the robot [20].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.3.5 OPC UA

The aim of introducing RAMI-4.0 was to standardize implementation of industry 4.0. The focus of RAMI-4.0 is especially on common communication. In chapt. 2.3.1 the different kinds of layers have already been introduced. The level *Communication* is providing the exchange of data and information. A major obstacle to the introduction of Industry 4.0 is a common communication protocol that is also vendor independent. Open Platform Communications Unified Architecture (OPC UA) can provide a standardized and open source communication protocol. The protocoll specifies the rules how data is connected and transfered in the network. OPC UA uses a Service Oriented Architecture (SOA) for information exchange. The client sends a request and the server processes the request and sends the required information to the client [32].

2.3.6 Integration of RAMI 4.0

In the following, as practical example on how to use RAMI-4.0, the integration of a sensor is explained. RAMI-4.0 is a three dimensional model, but for reasons of clarity and comprehensibility a two dimensional model, as seen in fig. 2.5, is used without the Life Cycle & Value Stream axis.

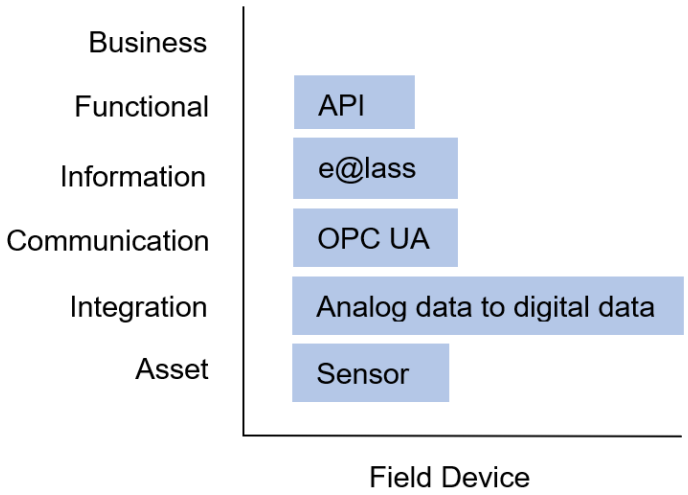

Figure 2.5: Possible integration of RAMI-4.0 explained with a sensor, adapted from [33]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The first step is to locate our sensor as part of the Hierarchy Level within the factory. One of these additional levels is called Field Device and sensors can be categorized to them. As shown in fig. 2.5 on the vertical axis are the *Layers*. The lowest of these layers is the asset. In this example the sensor is an asset. The Layer above is the integration layer. It converts the physical (analog) data into a digital signal. For an integration into industry 4.0 the sensor needs to have an administration shell as well as SOA for information exchange.

In the *Communication Layer* the communication standard OPC UA is used. Furthermore it needs to be classified in which data format the information from sensors is used. For classifying information usage in the Information Layer ecl@ss is usually used. ecl@ss is a data standard that allows an unambiguous description of products and services with iso compliant characteristics. In the *Funcional Layer* the access of sensor data is provided by using an Application Programming Interface (API). A sensor does not have any direct influence on the business layer therefore the business layer is empty [33].

2.4 Energy Storage Systems

The basic operation principle of an energy storage system is to provide energy when needed and to store energy when there is excess energy. In the real world energy provision rarely coincides with energy consumption. When energy demand is high, energy storage systems can cover the power peak. Another important application field of energy storage systems is the sector of renewable energies. Renewable energies come with a volatile, fluctuating energy production. Energy storage systems can provide a part of the base load if renewable energies can not produce the requiered amount of energy [24].

2.4.1 Classification of Energy Storage Systems

There are many ways how to classify energy storages. However, all of them can be split into three processes: to load, to store and to withdraw. They can be divided into primary energy storages and secundary energy storages. Primary energy storages can be only once loaded

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and unloaded, for example disposable batteries. Secondary energy storages are multiple loadable [34]. Energy storage systems are also classified by the physical way they store energy. It is possible to store energy in an electrical, mechanical, thermal or chemical way. Fig. 2.6 shows this way of classification for energy storage systems.

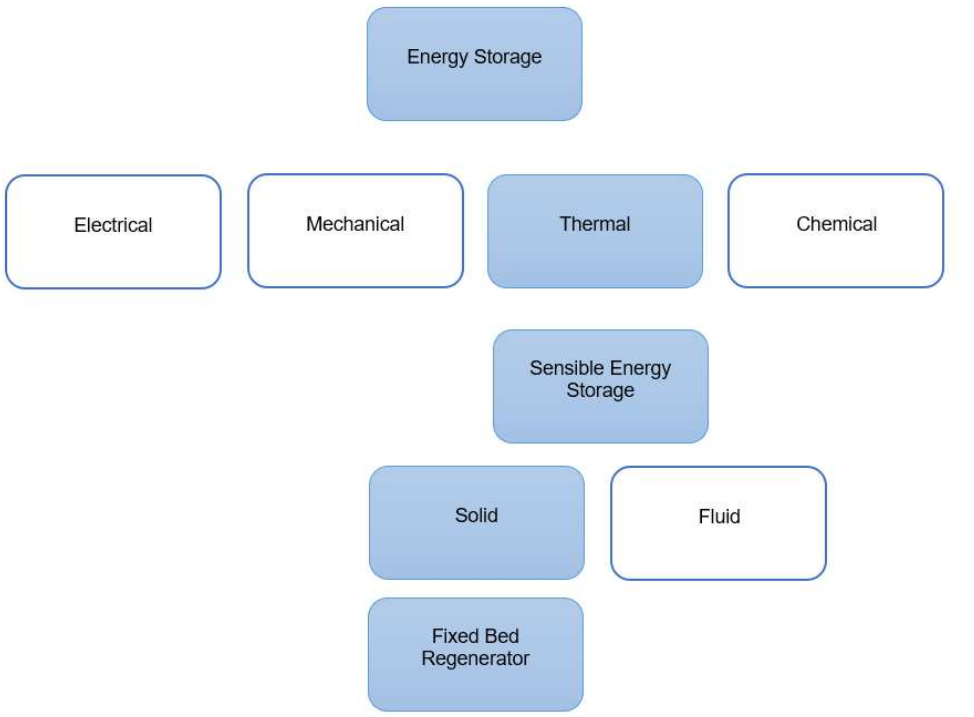

Figure 2.6: Classification of energy storage systems, adapted from [28]

In the overall classification of energy storage systems, a fixed bed regenerator is regarded as a sensible thermal energy storage with solid storage material [23]. Thus in the following, only thermal energy storage systems will be discussed in more detail.

2.4.2 Thermal Energy Storage (TES)

Thermal energy storage systems can be subclassified into latent, thermochemical and sensible heat storage, as shown in fig. 2.7. Latent thermal storages generate energy by phase transformation of their storage material. Phase transformation between solid-liquid or liquid-vaporous are possible but nowadays mostly solid-liquid phase transformation is used for latent thermal storages due to technical limits. Latent heat storages use so called Phase

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Change Materials (PCM) for this purpose. Loading and unloading happens when the PCM changes its current aggregate state. Thermochemical heat storages make use of reversible chemical reactions. An endothermic reaction absorbs enthalpy and loads the storage, on the contrary an exothermic reation unloads the storage and releases enthalpy [34].

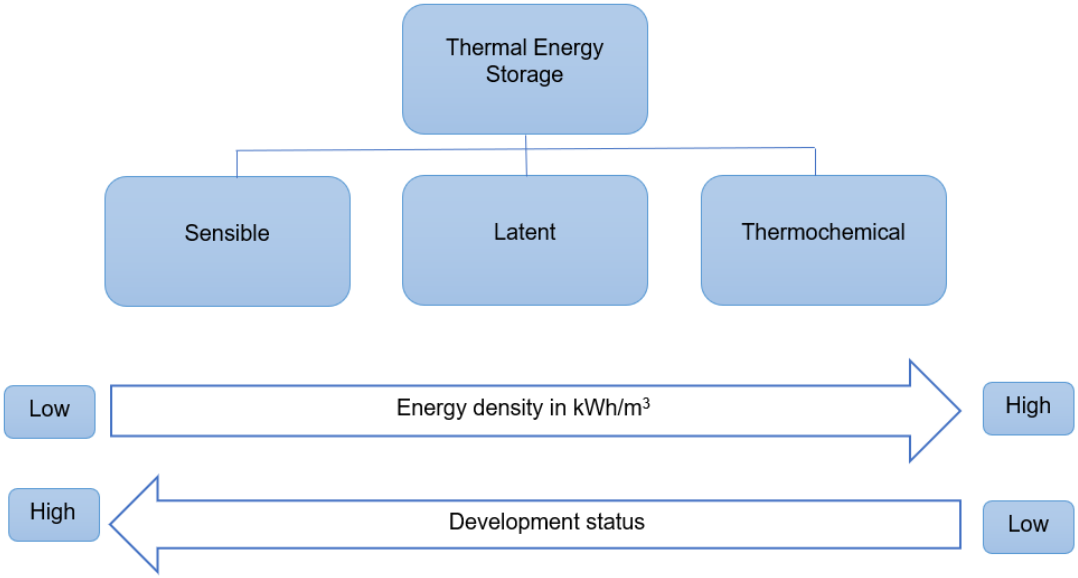

Figure 2.7: Classification of thermal storage systems, adapted from [34]

Sensible energy storages use temperature change for storing energy as heat. The stored amount of heat Q is equivalent to mass multiplied with heat capacity c_p of the used material and the temperature difference (T_2-T_1) of the storing medium before and after loading, see also equation 2.1.

$$Q = mc_p(T_2 - T_1) \tag{2.1}$$

Q : Amount of heat

m : Mass of the storing material

c_p : Heat capacity of the storing material

T_2-T_1 : Temperature difference

A sensible energy storage is restricted by the heat capacity of the used storing material as well as the maximum temperature difference. The higher the energy difference the higher is the usable energy. The higher the heat capacity of the storage material the less amount

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

of storing material mass is needed for the storage and the storage can be built smaller. The energy density of sensible energy storages is one of the lowest among thermal energy storages. This is due to the restricted energy density.

This leads to the definition of the term *usable energy*. The term *energy* stands for total energy and includes exergy and anergy. Exergy is the energy that can be changed to other energy forms without any loss. Anergy is the part of total energy that is considered as loss. It can not be further used or changed to other energy forms [34]. The loss is proportional to the surface of the storage but the stored heat is proportional to the volume of the storage medium. Because of that regenerators are build in a cylindric shape with small cross section in relation to the length of the cylinder [28]. Due to the first law of thermodynamics, the sum of exergy and anergy in a closed system is constant. Referring to the TES systems that means that the temperature difference of the storage medium before and after loading determines the amount of exergy that is changed into anergy. The exergy can be calculated according to equation 2.2.

$$E_{\text{Exergy}} = Q\frac{(T_{\text{S}} - T_{\text{U}})}{T_{\text{S}}} \tag{2.2}$$

E_{Exergy}: Usable energy

Q : Amount of heat

T_S : Temperature of the storage

T_U : Ambient temperature

Sensible energy storages can be further divided into liquid and solid sensible energy storages. As liquid storage materials mostly water ist used. Liquid sensible energy storages are able to transfer high power. Solid sensible energy storages on the contrary are able to store very high temperatures. For this purpose, storing mediums like concrete, sand, gravel and brick can be used. Besides the disadvantage of low energy densities, sensible energy storages are well developed and low in cost and therefore widely used nowadays [34].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.4.3 Fixed Bed Regenerator (FBR)

As already mentioned in chapt. 2.4, a FBR can be classified as a sensible thermal energy storage. It is best applicable for short and medium time storage application. A FBR is working with two working processes as shown in fig. 2.8. One is the process of loading and one is the process of unloading the stored energy. Hot air enters the FBR and is thus starting the process of loading with the hot air heating up the storage material. When energy is needed, the storage is unloaded: Cold air enters the FBR on the bottom and is heated by the storage material, respectively the stored energy inside the storage. In general there

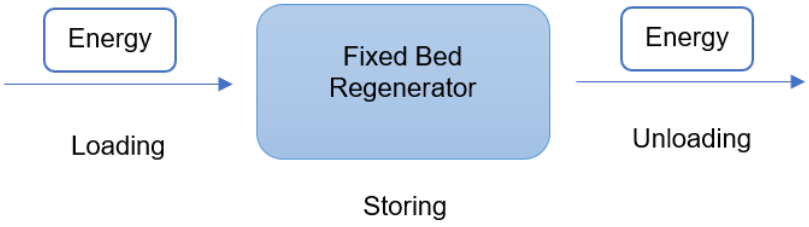

Figure 2.8: Basic operating principle of a fixed bed regenerator, adapted from [28]

are two types of regenerators which are distinguished, being fixed bed (not moving) and moving regenerator. In a FBR, not the storage medium neither the storage itself is moving. To guarantee a continous supply of energy, two regenerators must be parallel connected. In the case one regenerator is loading, the parallel connected regenerator is simultaneously unloading. In contrast, the moving regenerator system has two seperate chambers, in one chamber hot gas flows through in the other chamber, cold gas flows through. The regenerator is loaded and unloaded by moving from the hot gas side to the cold gas side.

It is also important to differentiate between the terms regenerator and recuperator. Both are heat exchangers but the working process is different. Regenerators are able to store energy, loading and unloading cycles are handled separately. In a recuperator, cold and hot gas flows through simultaneously. The two process streams are seperated by a wall which is also used for heat transfer from the hot fluid to the cold fluid [21].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.5 Data-Driven Modelling

The data gathered from industrial processes can be used for various tasks and applications. Data-driven modelling means the creation of models by using data in a bottom-up approach. In contrast, in a top-down approach, theoretical considerations are the basis of a model. In contrary to the top-down approach, that starts with an already chosen model and organizes data within that model, in a bottom-up approach the used data determines what model to choose [26]. Models that are solely based on data are called black box models. Fig. 2.9 shows how a black box model works. Input in the form of data enters the black box as stimulus

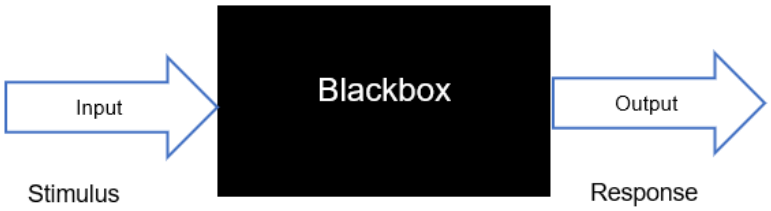

Figure 2.9: Structure of a black box model, adapted from [15]

from the left side. In the black box, data-driven modelling methods are applied which produce an output for further processing, also called response. The opposite of the black box model is the white box model. White box models only rely on physical correlations. A mix between white box and black box model is the grey box model. It combines data and knowledge to generate a meaningful output. Since data-driven modelling is based on data as input, the data needs to be of high quality and should be available in large quantities. Missing data can lead to an incomplete model and incorrect results [17].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.5.1 Machine Learning

One method of data-driven modelling is Machine Learning. Machine Learning (ML) is a partial area of AI. ML can be defined as:

Research in ML has been concerned with building computer programs able to construct new knowledge or to improve already possessed knowledge by using input information. [14]

There are many definitions of the term "learning". Learning is closely related to the term intelligence vice versa, to be intelligent contains the meaning to have the ability to learn. The general learning model provides a good approach to explain of which building blocks learning consists, see also fig. 2.10.

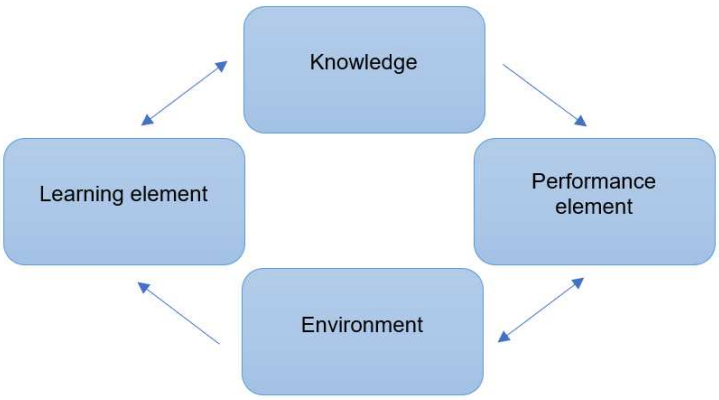

Figure 2.10: General learning model, adapted from [14]

Learning success of humans is usually measured by performance increase, that means how well they perform on specific tasks. The performance element uses its gained knowledge and applies it on the environment. The learning element extracts environmental impressions and processes them to generate knowledge [14]. ML algorithms do not have to be programmed explicitly to fulfill their desired output, they have the ability to generate accurate results by recognizing patterns and apply these learned patterns on similar tasks.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

So the working principle of ML is to get data as an input and to analyze this data to predict an output, while at the same time refreshing the output by feeding new input data.

ML can be classified into three different kind of learning techniques: Supervised, unsupervised and reinforcement learning.

Supervised Learning Supervised Learning uses only labeled data for training. That means, input and expected output of data is fed into the model and while training, the model tries to find a pattern on the labeled data. After training, the model can predict output on never seen data. If, for example, a model is fed with pictures of cats and dogs that are correctly tagged as cat or dog, the model can learn while training which pictures show dogs and which pictures show cats. After learning from tagged data, the model is able to predict from untagged data if a picture shows a cat or a dog.

Unsupervised Learning Unsupervised Learning means that the model does not know what the training data exactly shows since the data is unlabeled. These kind of models are for pattern recognizition and anomaly detection. The model tries to find a pattern behind the training data and to predict a specific output. An example for unsupervised learning would be a clustering analysis, where a company wants to cluster customer data by demographic affiliation to create targeted customer profiles.

Reinforcement Learning Reinforcement Learning is learning of a model through rewards. It is especially used in the video game sector. An example would be a virtual player in a game who is getting better and better through learning how the game works and through rewards that the player gets when playing well.

There are many different techniques of ML. Fig. 2.11 shows some techniques of supervised and unsupervised learning [18]. This work focuses on supervised learning with *Neural Networks*, which are further explained in the next section.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

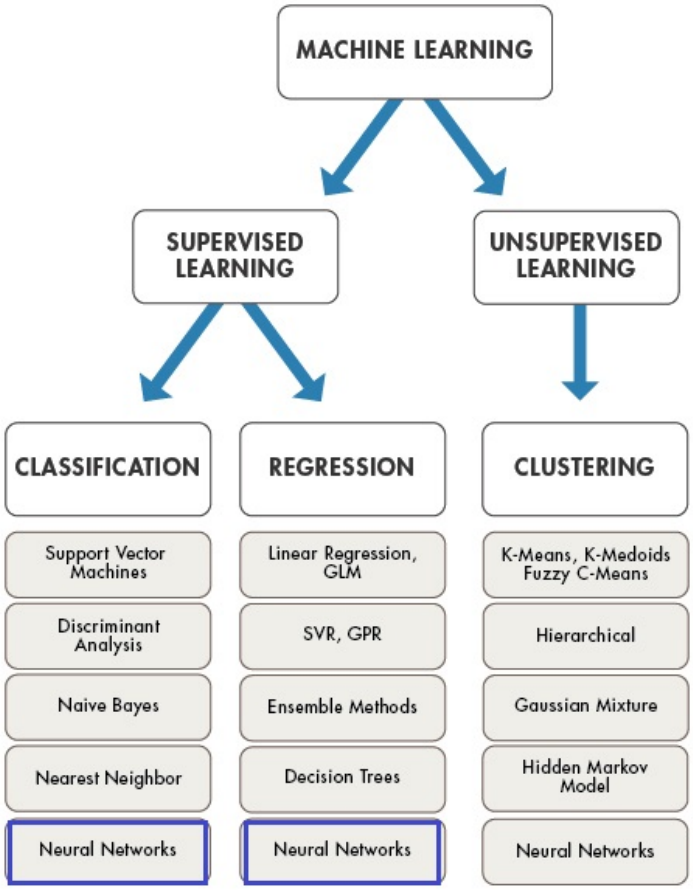

Figure 2.11: Techniques of ML [3]

2.5.2 Artificial Neural Network

Artificial Neural Network (ANN) are a replica of the human brain. To understand how an ANN works, it is necessary to take a closer look at what happens when the human brain processes information. Signal processing in the human brain happens in cells called *neurons*. Neurons are connected to each other with synapses. When a signal arrives at a neuron, an activation potential is released. If this activation potential exceeds a certain value (treshold) the information is passed on. ANNs rebuild this structure in a more simplified version [27]. Fig. 2.12 shows a possible simple structure of a feed forward ANN with four layers. It passes on information from an input layer to a first hidden layer, afterwards to a second hidden layer to process the information and in the end to an output layer. The output layer is the last layer and generates a prediction. These networks are called feed forward neural networks, because they do not contain any loops to feed back values. A feed forward ANN is

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

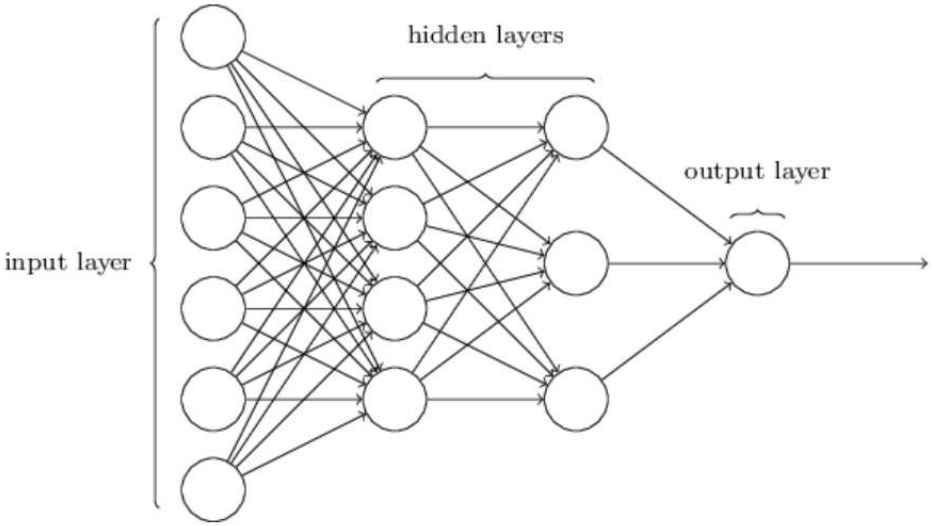

Figure 2.12: Work structure of a feed forward ANN [29]

the simplest form of an ANN. The type of ANNs that feed back values are called Recurrent Neural Network (RNN). These models approach the working process in the human brain more acccurate than feed forward neural networks do. For a better understanding of the behaviour of ANNs, the mathematical structure is being explained on a feed forward ANN. Fig. 2.13 shows the mathematical structure of one neuron in an ANN. The input to a single

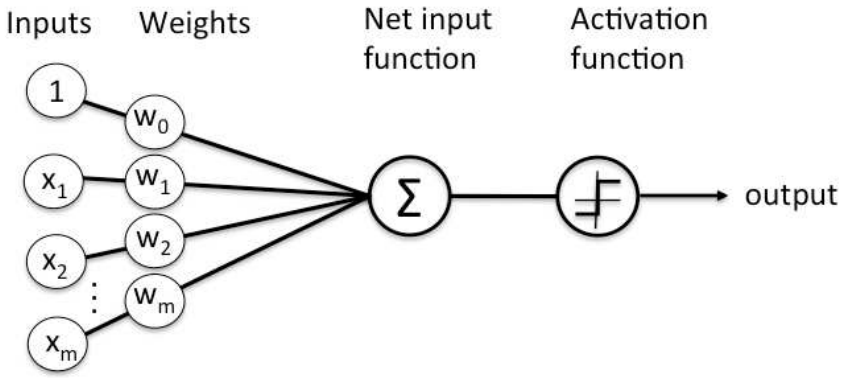

Figure 2.13: Mathematical structure of one neuron in an ANN [6]

neuron can either be features (variables of the dataset), if it is the input layer, or output from the previous layer, if it is a hidden layer or output layer. After applying weights to the input, the input enters the neuron as sum of weighted inputs *h(x)*. Weights can take on values between 0 and 1. The higher the value of a weight the more important an input is.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The sum of weighted inputs *h(x)* is calculated by equation 2.3. After entering the neuron, an activation function with a treshold is applied to *h(x)* [29].

$$h(x) = \sum_{i=0}^{n} w_i x_i \tag{2.3}$$

h(x): Sum of weighted inputs

w_i : Weights

x_i : Input

2.5.2.1 Activation Functions

As mentioned, there is a specific treshold which decides wether or not a value is passed on. This treshold, also called bias, is characterized by the chosen activation function [22]. Many different activation functions exist, but there are only a few who are commonly used. These are: *Sigmoid*, *Tanh* and *ReLU*.

Sigmoid As seen in fig. 2.14, the *Sigmoid* activation function is a S-shaped function. The convergence of the *Sigmoid* activation functions is slower in comparison to other activation functions and it also suffers from vanishing gradient problems, which will be explained in section 2.5.2.2. It can take on values between 0 and 1, calculated by equation 2.4.

$$y = \frac{1}{1 + e^{-x}} \tag{2.4}$$

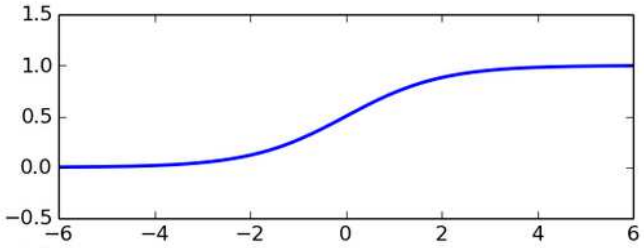

Figure 2.14: Shape of the Sigmoid activation function [10]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Tanh The *Tanh* activation function is also S-shaped, as seen in fig. 2.15, but it can take on values between -1 and 1. These values are calulated by equation 2.5. Negative input values into the activation function are also mapped as negative. Values that are zero are mapped as closely zero and positive values take on values between 0 and 1.

$$y = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.5}$$

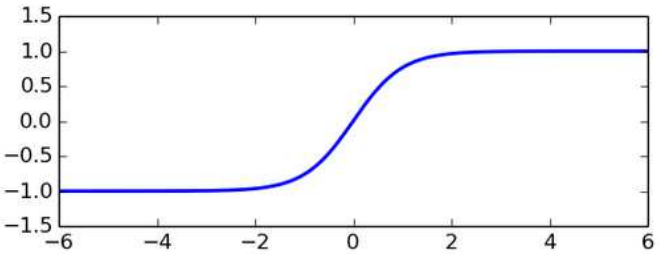

Figure 2.15: Shape of the Tanh activation function [10]

ReLU The Rectified Linear Activation Function, *ReLU*, is a non-linear activation function, which takes on the value zero when x is less than zero or negative. It takes on the value x, if x equals zero or takes on a positive value, see also equation 2.6. *ReLU* does not suffer from the vanishing gradient problem, as well does it converge faster than *Sigmoid* and *Tanh*. The shape of this activation function can be seen in fig. 2.16 [5].

$$y = max(0, x) \tag{2.6}$$

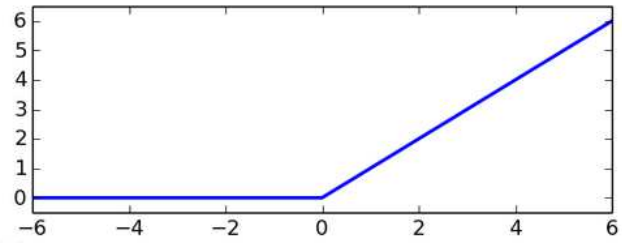

Figure 2.16: Shape of the ReLU activation function [10]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.5.2.2 Backpropagation

In former standard feed forward ANNs, values of a vector as an input have been fed through layers and output was produced, no weight adjustement was happening. But the purpose of an ANN is improvement through learning a certain pattern, that means adjusting weights is an important process to ensure a proper learning. For this reason backpropagation was introduced. Backpropagation means feeding back an error from the output layer to the input layer with the usage of a technique called *gradient descent*.

An error at the end of one training round results in the comparison of training and test data. The error signifies the difference of values between training and test data, so how well the ANN performed on training. When feeding back the error through the ANN the error is divided according to the weights of all neurons in a layer. A neuron with a high assigned weight will geht a higher error assigned than a neuron with a low weight. To newly adjust the weights in backpropagation the gradient descent helps to find the minimum of the error function [31].

There are two problems that can occur when backpropagating: exploring gradient problem and vanishing gradient problem. Exploring gradients are a problem of Deep Neural Networks and RNNs. Gradients become very large and therefore weights are updated to very high values. This results in an unstable network. A vanishing gradient means, the gradient becomes very small or even vanishes and training stops because weights are not updated [30]. For a better understanding of how a feed forward networks work and a more detailed explanation of the mathematic background behind neural networks, reference is made to the book of Tariw Rashid who provides a simple and understandable view on this topic [31].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.5.3 LSTM

Long short-term memory (LSTM) is a special type of Recurrent Neural Network (RNN). Both, LSTM and RNN, are good at learning sequences and have a feedback loop, which allows them to remember and to use information from previous time steps. RNNs have a indefinite memory, therefore, when predicting data with information far back in time, RNNs have difficulties producing a meaningful output, whereas LSTM has the capability to remember information far back in time. Thus LSTM does very well in learning sequences with long time lags [1]. Application area of LSTM is voice recognition, recognition of handwriting and especially forecasting of time series [36].

The upper part of fig. 2.17 shows the working structure of a RNN. The commonly used activation function of RNN is *Tanh*. An input value x_{t-1} is fed into the input layer, generating an output h_{t-1}. This output is being fed into the Neural Network in the next iteration, generating an output h_t. That means every output after an interation is passed on as input to the Neural Network in the next iteration step. The lower part of fig. 2.17 shows the

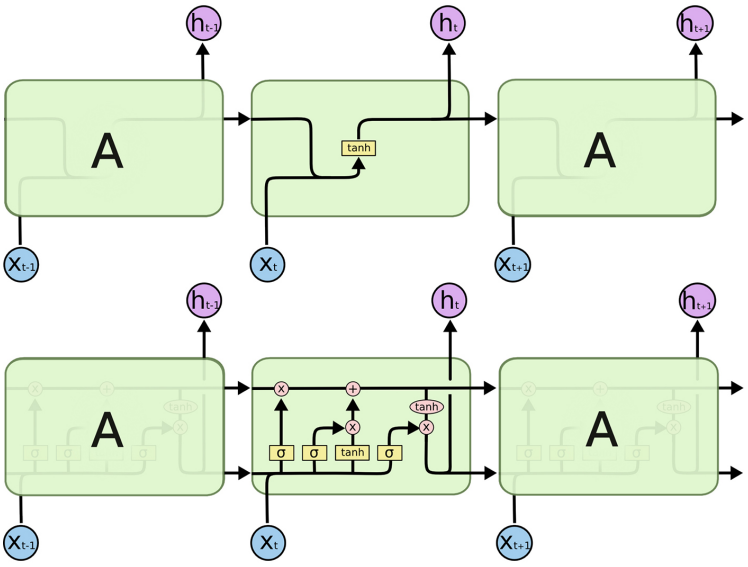

Figure 2.17: The upper part shows the working structure of a RNN, the lower part the structure of a LSTM [1]

working structure of a LSTM. LSTM does very well in remembering long-term dependen-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

cies. For a better understanding of what is going on in a LSTM cell, fig. 2.18 shows one cell with its three different gates [1]. The very top line in fig. 2.18 is called the cell state C.

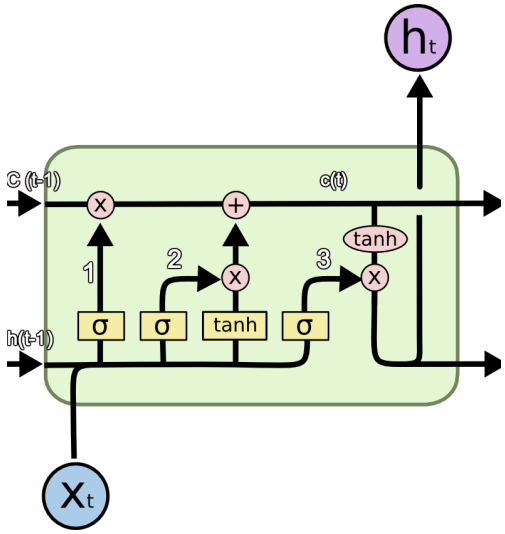

Figure 2.18: Closer look into once cell of a LSTM [1]

The cell state contains selective information of the past and enters on the left side as C_{t-1}. A LSTM cell consists of three gates called forget gate, input gate and output gate. Every of these gates has a specific task to fulfill. The forget gate 1 contains a Sigmoid function and is responsible to decide which information from the previous cells is being passed on to the top cell state and which information is being forgotten because it does not add any value to generate a meaningful output. It processes h_{t-1} and the input vector x_t. The next gate in the cell is the input gate 2. This gate consists of two different processes. The first is the updating of values which is achieved by combining the values of the input vector x_t with the previous output of the prior cell h_{t-1}. The tanh layer creates a vector with possible new values. These two processes are in the end combined to update the state of the cell. The last gate is the output gate 3. It decides which values of the cell state are passed on as input to the next cell. This is done by a tanh layer, which shapes the values between -1 and 1 and multiplies them with the former input which are passed via a Sigmoid layer. This process generates the output h_t [35].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.5.4 Deployment of Machine Learning Models

A lot of time and energy is invested in developing a model that performs well. While training a model and tuning hyperparameters is indeed a significant working process, to know how to deploy ML models into production is just as important. There are two common ways of making a code available in production: rewriting the code into the programming language used in production processes or turning a ML model into an API.

The way of rewriting the *Python* model into the programming language used in production processes, mainly *Javascript*, is a time consuming way and often inefficient since *Javascript* compared to *Python* does not have that good libraries for building a ML model. The other method is the deployment of a ML model with Flask as REpresentational State Transfer (REST) API. Flask is a web framework for Python, which allows to write web applications. This framework provides rules of interacting with a webserver. A web API is an API that is HTTP based and therefore hosted over a server. REST API is a type of API, setting rules and guidelines how to structure a web API. REST APIs are based on four HTTP requests which makes them easy to use: GET, PUT, POST and DELETE. For the deployment of ML models only the requests GET and POST are important to understand, so these two will be further explained. A GET request is used to retrieve information from a resource without modifying it. On the contrary, the request POST is used to send information to a server to create or update a resource.

Fig. 2.19 shows the process of a ML model deployment from the very beginning until the performance of the trained model on new data. In general, the deployment of a ML model with Flask can be split into two main processes: saving the trained model serialized and wrapping the model into an API by using Flask. The first step is to serialize a model into the dataformat JavaScript Object Notation (JSON), also called “pickle“ the model. The aim of serializing a model is to convert the trained model into a format, in this case JSON, within that the model can be stored or transmitted. This is done with the command *joblib* from scikit-learn. The model can be loaded back into the working space with the command *joblib.load* and will be then converted back into the former data format, also

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

called *Deserialization* [7].

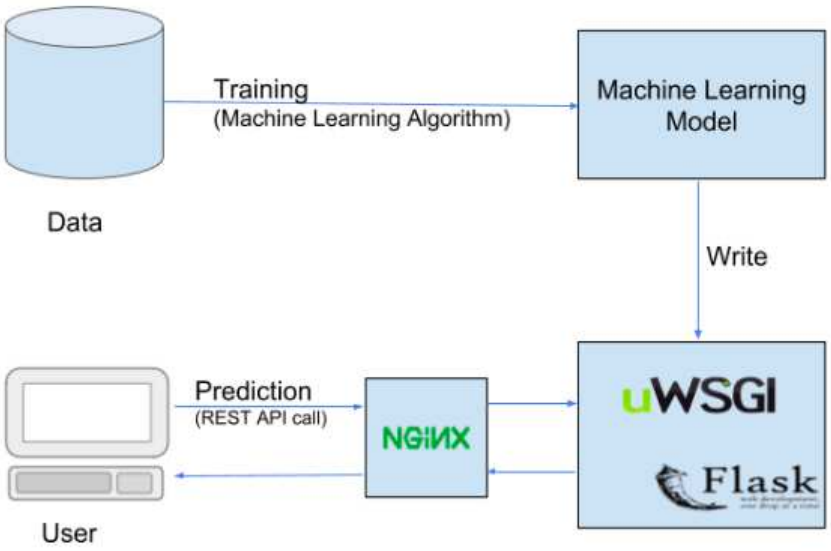

Figure 2.19: Deployment of a ML model with Flask as REST API [2]

The next step is the wrapping of the serialized model into an API to enable a client to use it to make predictions on new data. The wrapping of a ML model into an API needs to be done because many industrial applications use the programming language *Java* and a in *Python* written ML model can not be integrated into an application using an other programming language. The HTTP request GET receives new input data from the front end (User) which is then processed in the API wrapper to return a prediction as POST request, which is deserialized before outputting. *Nginx* is a web server but can be also used, as done here, as HTTP load balancer. A HTTP load balancer distributes HTTP requests to a number of servers [4].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Chapter 3

Experimental Setup & Data

In the following the FBR, an experimental pilot plant at the Institute for Energy Systems and Thermodynamics at the Vienna University of Technology, will be roughly explained to understand its charging behaviour. The sensors and measured quantities will be described and the resulting data is shown. All measurements were conducted in the course of a master thesis [28], which also provides more detailed information about the experimental setup. Fig. 3.1 shows the experimental plant without insulation and pipe holders.

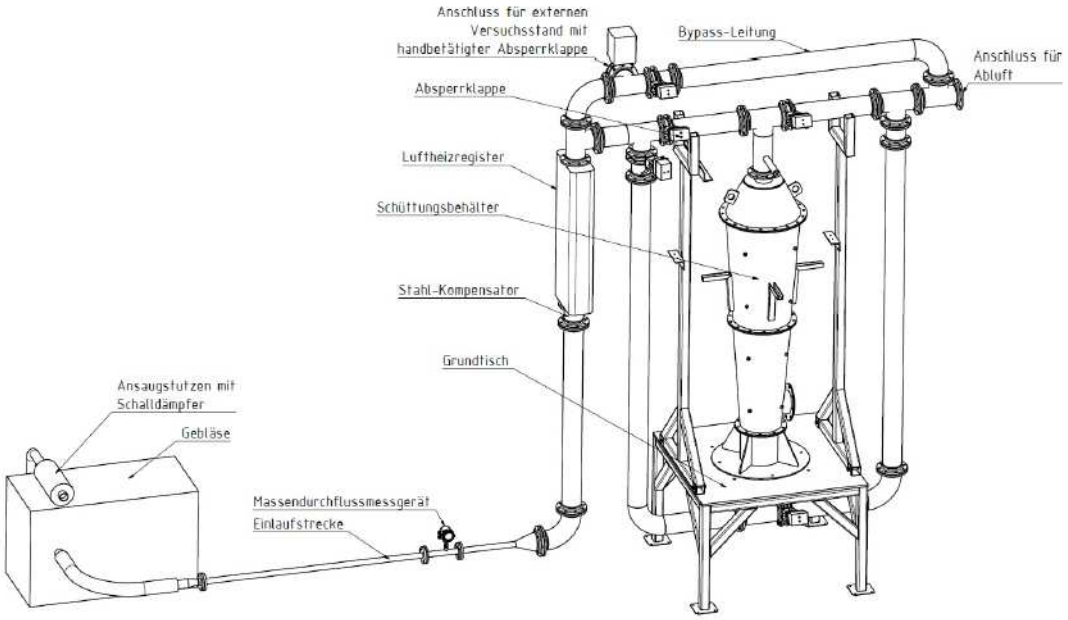

Figure 3.1: Sketch of the FBR test rig [28]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The packing of the FBR is gravel, thus gravel serves as storage medium and hot air provides energy in the form of heat. Gravel is a good storage medium for smaller regenerators since it is inexpensive and has good temperature characteristics [28]. Tab. 3 shows the material properties of gravel [21].

Density	2500 kg/m^3
Specific heat capacity	840 J/(kgK)
Thermal conductivity	1 W/(mK)

Table 3.1: Material properties of gravel [21]

3.1 Used Metrology

The FBR is equipped with different sensors, which are explained in the following.

3.1.1 Thermocouples

Thermocouples are used as temperature sensors in the FBR. In total, 16 thermocouples are built in. They measure the temperatur on four levels around the regenerator. Every level has four thermocouples to guarantee the correctness of the measurement around the radius of the cylinder. In fig. 3.2 these four levels are designated as “Messebene 1-4“. The four measured temperatures of each level are averaged to obtain an accurate estimation of the temperature at each level and result in the temperatures T1-T4.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

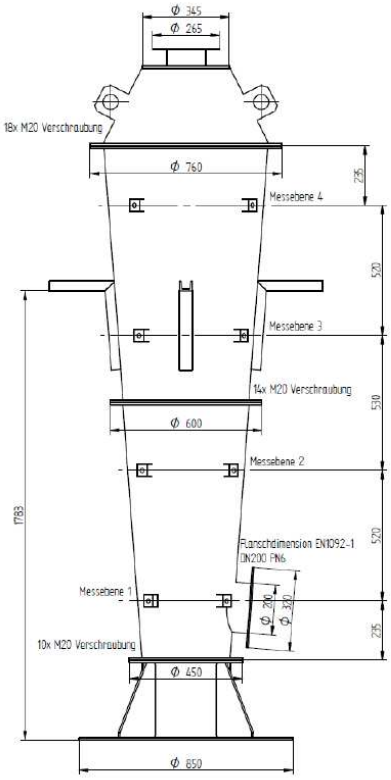

Figure 3.2: Thermocouples screwed in on four levels of the FBR [28]

3.1.2 Resistance Temperature Sensors

Two resistance temperature sensors, one at the top of the fixed bed regenerator (PT02) and one at the bottom of it (PT01), measure the entry and the exit temperature of the air flow. The measurement of PT02 is used to control the FBR and determines the switching between the states of loading and unloading.

3.1.3 Mass Flow Sensor

The mass flow is measured by two resistance temperature sensors type PT100.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.1.4 Pressure Sensor

On the top of the regenerator a relative pressure sensor is mounted to measure the prevailing pressure at different load cycles. To validate the values of the mass flow measured by PT100, differential pressure sensors are used additionally. One is mounted on the inlet and one on the oulet of the FBR [28].

3.2 Measurements

Figures 3.3 and 3.4 show how loading and unloading of the FBR works. For loading and unloading, the air flows from bottom to top of the reactor. Five butterfly valves are used to control the flow direction of the air.

When loading the FBR, hot air is entering the FBR from the bottom and heats the storage medium gravel in the FBR, on the top, the cooled down massflow is leaving. When the maximum temperature at the top of the FBR is reached, the FBR switches its operation mode from loading to unloading. Cold air is now entering the FBR from the bottom and takes up the heat from the gravel. This is called unloading. The total measured data consists of four measured series, M1, M2, M3 and M4, with different measurement settings. Tab. 3.2 shows the parameters and their definition and tab. 3.3 shows the values of these parameters.

Measurement series M2 is not discussed any further, since M2 has a varying mass flow and and considering varying mass flows would exceed the scope of this thesis. The data of measurement series M1 consists of four cycles, whereas the data of M3 and M4 was both measured with a cycle number of three. Technically, the data of measurement series M4 was measured with four cycles, but since M4 was directly measured after the last cycle of M3, the first cycle differs from the other three cycles and is not considered further [28].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

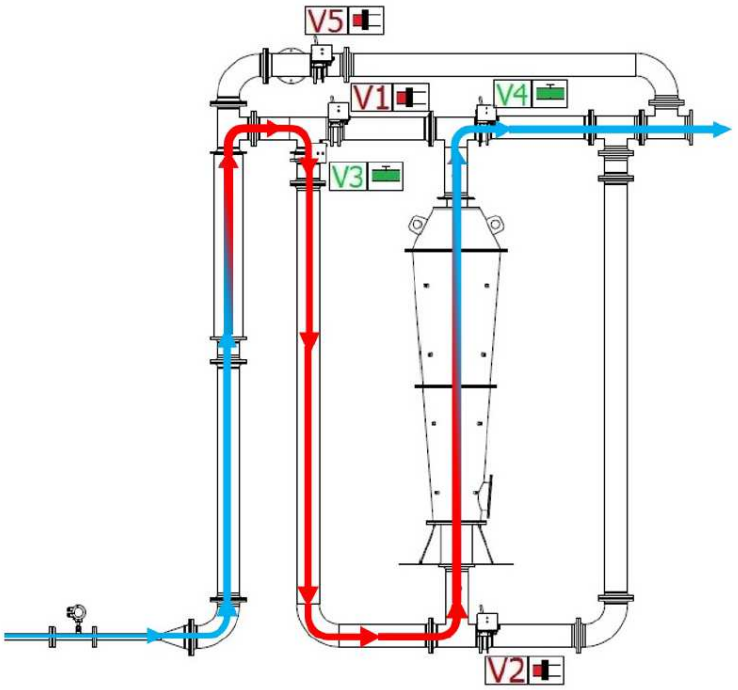

Figure 3.3: Air flow direction when loading the FBR [28]

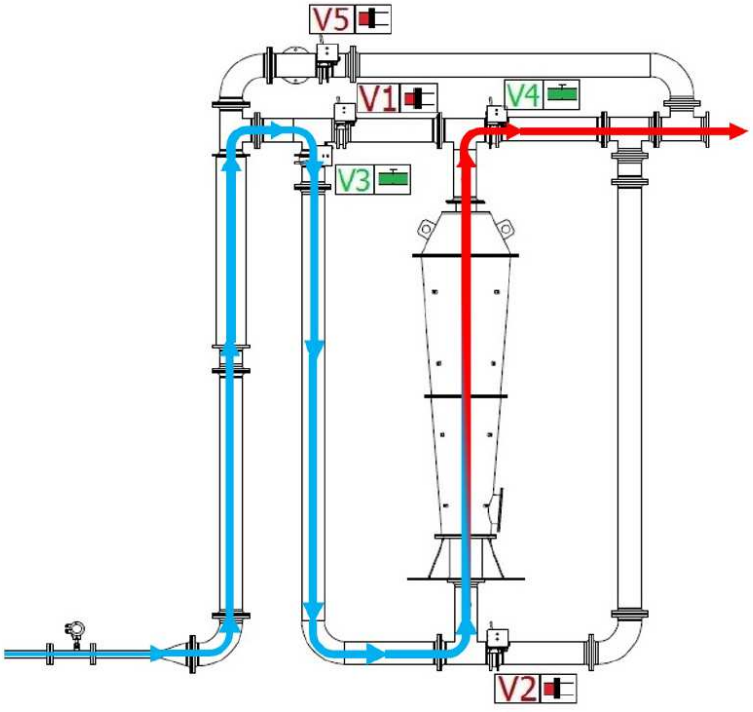

Figure 3.4: Air flow direction when unloading the FBR [28]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Parameter	**Description**
Cycles	One full cycle consists of unloading and loading
Mass flow	Constant mass flow during the measurement
T_loading	Inlet temperature of the air when loading the FBR
T_unloading	Inlet temperature of the air when unloading the FBR
T_up_loading	Temperature at the outlet until the storage is loading
T_up_unloading	Temperature at the outlet until the storage is unloading
Loading Time	Time passed until the storage is fully loaded
Unloading Time	Time passed until the storage is fully unloaded
Cycle duration	Total duration of one cycle

Table 3.2: Description of the experimental parameters [28]

Parameter	**M1**	**M3**	**M4**
Cycles	4	3	3
Mass flow in kg/h	146,4	150,0	150,0
T_loading in °C	310	310	310
T_unloading in °C	20	20	20
Tup_loading in °C	265	265	200
Tup_unloading in °C	50	200	150
Loading Time in h	11,7	15,7	5,2
Unloading Time in h	7,0	3,6	3,9
Cycle duration in h	18,7	19,3	9,0

Table 3.3: Parameters of the three measurement series [28]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Figures 3.5, 3.6 and 3.7, show the plotted temperature curves of T1-T4, PT01 and PT02, measured by thermocouples (T1-T4) and by resistance temperature sensors (PT01 and PT02) of measurement series M1, M3 and M4.

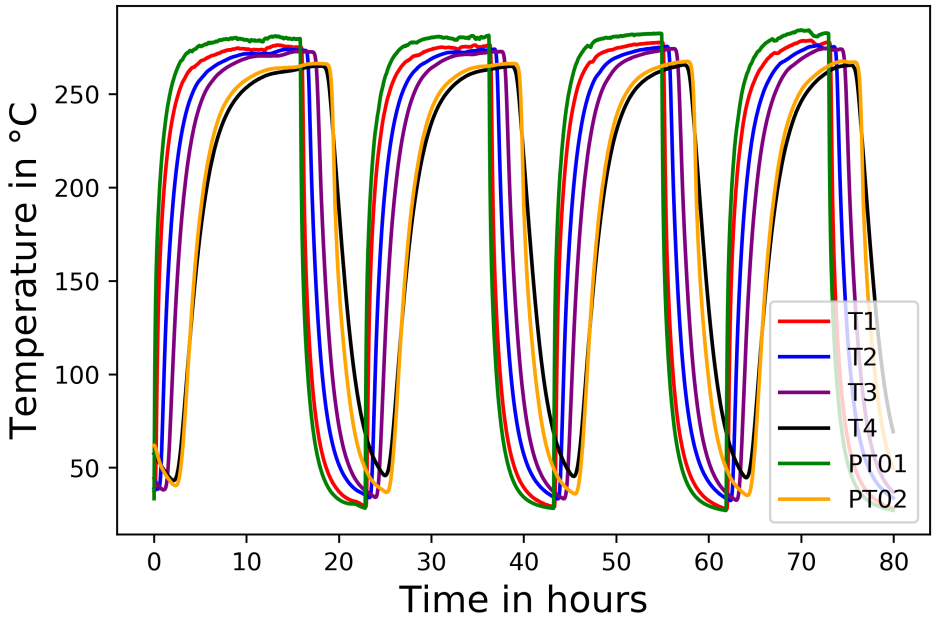

Figure 3.5: Plotted temperature curves of T1-T4, PT01 and PT02 of measurement series M1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

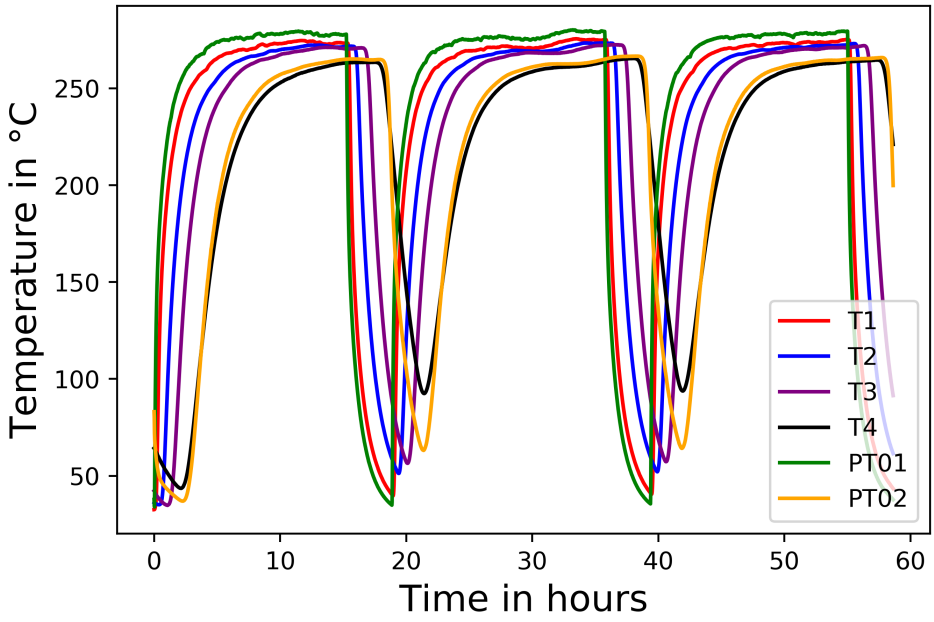

Figure 3.6: Plotted temperature curves of T1-T4, PT01 and PT02 of measurement series M3

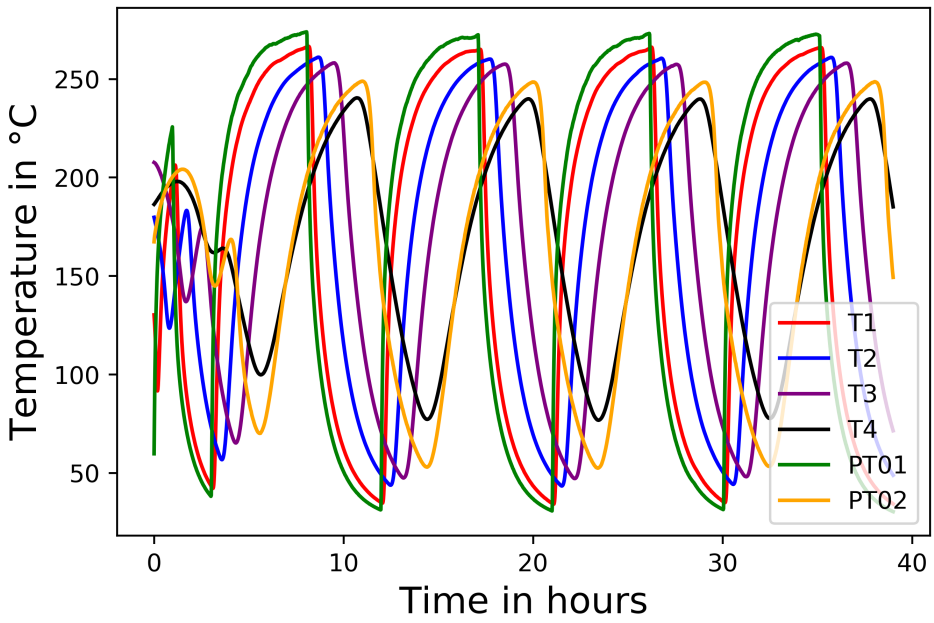

Figure 3.7: Plotted temperature curves of T1-T4, PT01 and PT02 of measurement series M4

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Chapter 4

Building of LSTM Models in Tensorflow

In the previous chapter it was discussed how and what data of the FBR was obtained. In this chapter the focus lies on data-driven modelling by using this data. The used data can be considered as time series, since all the used temperature data was measured over a constant time step. LSTM models perform well on time series data. Therefore, for generating a data-driven model, the modelling technique LSTM is used and is implemented in the open source framework *Tensorflow*. In the following sections, a documentation of the most important parts, when creating a LSTM model in *Tensorflow*, is being given. That also includes basics, in order that *Tensorflow* beginners can also understand it.

4.1 Development Environment & Libraries

The code is written in the programming language *Python* and uses the open source framework *TensorFlow*. *TensorFlow* is an open source programming library for AI and was developed by Google Brain. It is especially used on ML problems. *TensorFlow 1.10.0* and *Python 3.5* were used in this work. As Integrated Development Environment (IDE) for programming, *Spyder* is used. At the beginning of the code the needed packages are imported into the programming environment *Spyder*.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
```

In the following the used packages are further explained.

Pandas Pandas is a software library for *Python* and a tool for data manipulation and data analysis.

Scikit-learn Scikit-learn (short Sklearn), is a open source software library for ML. The programming language used for Scikit-learn is mainly *Python*.

Numpy Numpy is a package for the programming language *Python*. It allows a simple working with matrices, vectors and arrays.

Matplotlib Matplotlib is a programming library for visualizing *Python* scripts.

Keras Keras is a deep learning library, compatible with *TensorFlow* and *Theano*. *Theano* is an open source programming library, same as *TensorFlow*. *tf.keras* stands for the implementation of the Keras API into *TensorFlow*.

4.2 Loading of CSV Data

```
df=pd.read_csv('N-M1-Z1_N_TO265_TO50_M146_TLad310_V2.csv', delimiter =
    ';',skiprows=range(0,1),usecols=[1,2,3,4,5,6])
df=df.stack().str.replace(',','.').unstack()
df=df.astype(float)

```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
5 X=df.iloc[:,0].values
6 y=df.iloc[:,1:6].values
```

The data is imported from an Excel file that has been converted to CSV. CSV means comma separated values and its difference to an Excel file is, that CSV contains only the data, whereas Excel is a binary file, which besides the data, also contains formatting. For importing CSV data into *Spyder* a command called *pd.read_csv* from the pandas library is used. *skiprows* allows to skip a defined number of rows. In this example code, the first row is skipped, since the CSV file contains a header. *usecols* specifies the columns that are imported from the data. In this case the columns 1 to 6 are imported.

The next step is to replace commas in numbers in the CSV file to dots. This is done by the command in line 2. The values imported are *strings* and in the next command with *astype.float* converted to *float*. In the next step the columns are assigned to a specific variable. The command *df.iloc* seperates the desired rows and columns and assigns them to a new value. The first column is assigned to a variable called X. Columns 2-6 are assigned to a variable called y. The values in the square bracket specify the rows and columns. The first value in the bracket is a colon and means all rows of the data are passed to the new variable. The second value signifies the columns that are assigned to the new variable.

```
1 data=pd.Series(X)
2 data.plot()
3 plt.ylabel('Temperature')
4 plt.xlabel('Epochs')
5 plt.legend(['Name'], loc='lower right')
6 plt.savefig('Name.png')
7 plt.show()
```

To see if data is noisy or has other abnormalities, it can be plotted as series. For plotting data as series the pandas command *pd.Series* is used. It allows to convert the data format *float* into a series. *Matplotlib* is used to visualize it. With the command *plt.savefig* a figure in PNG format is saved.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.3 Scaling of the Data

```
1 scaler=MinMaxScaler()
2
3 X=numpy.array(X).reshape((len(X), 1))
4 X=scaler.fit_transform(X)
5 X=X[:,0]
6
7 y=scaler.fit_transform(y)
```

Scaling of the data is an important step when creating input for a LSTM. Usually values of data vary and lead to no proper results when feeding into a model. Scaling makes the data more even with fewer outliers. With the *MinMaxScaler* from the package *Scikit-learn*, the data is scaled in values between 0 and 1. This is done by equation 4.1. The *MinMaxScaler* expects two dimensional data (samples, features) as input for scaling. Therefore, in line 3, the one dimensional X was reshaped with *np.array* into two dimensional data.

$$x_{\text{scaled}} = \frac{x - min(x)}{max(x) - min(x)} \tag{4.1}$$

x_{scaled} :	Scaled value of a value x
x :	Value that is going to be scaled
$min(x)$:	Minimum value in the dataset
$max(x)$:	Maximum value in the dataset

4.4 Train/Test split

```
1 Xtrain, Xtest, ytrain, ytest=train_test_split (X,y, test_size=0.2,
    shuffle=False)
```

The data is now splitted into a training dataset and a test dataset. This is done by the command *train_test_split* from the package *scikit-learn*. The training dataset will be used

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

for training the model to recognize patterns in the data. This finding of patterns in data is also called fitting the model and results in minimizing the training error. Altough a trained model performs well on training data with a minimized error, in order to assure that the trained model really found a pattern in the data and is not just memorizing data, the model needs to perform also well on the test dataset. The prediction on the test dataset shows if the model is really fitting well on the whole dataset. The command *test_split* defines which percentage of the data is used for test data. *shuffle=False* means the whole data is not shuffled before splitting into train and test data. Since the used data is time-series data, the order of the data is crucial for a good prediction.

The splitting is done by using the two variables X and y. X is splitted into *Xtrain* and *Xtest*. *y* is split into *ytrain* and *ytest*.

```
1 Xtrain, Xval, ytrain, yval=train_test_split(Xtrain,ytrain, test_size=0.15,
      shuffle=False)
```

After splitting the data into training dataset and test dataset, a part of the training dataset is split off as a validation data set. The purpose of a validation dataset will be explained later.

4.5 Reshaping of the Data

```
1 time_step=1
2 features_input=1
3
4 Xtrain=Xtrain.reshape(Xtrain.shape[0],time_step, features_input)
5 Xtest=Xtest.reshape(Xtest.shape[0], time_step, features_input)
6
7 ytrain=ytrain.reshape(ytrain.shape[0], ytrain.shape[1])
8 ytest=ytest.reshape(ytest.shape[0], ytest.shape[1])
9
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
Xval=Xval.reshape(Xval.shape[0],time_step, features_input)
yval=yval.reshape(yval.shape[0],yval.shape[1])

Xall=X.reshape(X.shape[0],time_step, features_input)
yall=y.reshape(y.shape[0],y.shape[1])
```

The input to a LSTM layer must be three-dimensional. These three dimensions are: Samples, Time Steps and Features. The reshaping into a 3-D array is done by the command *reshape*. Samples signify the amount of data points in the dataset, in this case the number of rows of *Xtrain*. If the size of data is big, the prediction of a LSTM is not very accurate, that means the training dataset needs to be split into more samples. One sample contains a sequence of the dataset. Time steps mean the length of a sequence in one sample and has been chosen with 1. Features means the number of dimensions that the input has. If for example the input are features with Cartesian coordinates x,y and z and therefore 3-D, features would have the value 3.

4.6 Callback

```
callbacks=[tf.keras.callbacks.EarlyStopping(patience=10,min_delta=0,monitor='loss')]
```

The command *tf.keras.callbacks.EarlyStopping* defines when training should stop, by monitoring a measurable quantity, usually loss or validation loss. When reaching a defined minimum delta the training stops and is predicting an output. *patience=10* means the monitored quantity is not changing for 10 epochs and after 10 epochs with an unchanging value the training stops. *min_delta* signifies the minimum delta between two time steps that leads to stop the training when there is no improvement. *monitor* defines the quanitity which is monitored.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.7 LSTM

```
1 model=Sequential()
2
3 model.add(LSTM(5, input_shape=(None,features_input)))
4 model.add(Dense(2, activation='sigmoid'))
5
6 opt=tf.keras.optimizers.Adam(lr=0.01, decay=1e-6)
7 model.compile( optimizer=opt, loss='mse')
8
9 model.fit(Xtrain,ytrain, validation_data=(Xval,yval), epochs=150, batch_size=5,
    verbose=2, callbacks=callbacks)
```

The beginning of a LSTM model needs the command *Sequential*. The *Sequential* API from Keras allows to stack layers after another. Line 3 defines the input layer of the LSTM. The number after *LSTM* defines the number of neurons in the input layer, in this example code 5 input neurons have been chosen. The command *input_shape* specifies the shape of the input layer as Time Steps and Features. The Time Steps are set to *None*, that means the value of Time Steps can be set variably. This example code uses as activation function *sigmoid*. After the input layer another LSTM layer can be added if needed. The input shape does not need to be defined again if another layer is stacked. The command *Dense* is added after the last LSTM layer and changes the output shape. *Dense (2)* means the shape of the output has 2 features. If the *Dense* layer is not added, there is a value error that the shape is (None, 5), the number of batches and number of hidden neurons, instead of the expected shape (Time Steps, Features). As an optimizer *Adam* is used. The optimizer is responsible for the in chapt. 2.5.2 described gradient descent and is adjusting the weights according to the value of loss that is backpropagated. The optimizer also defines the learning rate (lr), the step size the gradient is using to minimize the loss. The decay signifies the learning rate decay after each update. *model.compile* is the compilation of which loss function is used and which optimizer. The loss function used for calculating the loss is the mean squared error *mse*.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In line 9 the model is fitted with *Xtrain* and *ytrain* and is trained on recognizing a pattern between the values of *Xtrain* and the values of *ytrain*. For cross validation, the command *validation_data* with the validation dataset *Xval* and *yval* is used. This validation dataset provides an unbiased evaluation of the training fit. The difference to the test data set is, that the validation dataset still allows tuning of hyperparameters, for example tuning of the learning rate or changing the number of neurons. *batch_size* defines the amount of training samples that are taken into consideration until the Neural Network is updated in its weights. *nb_epochs* signifies the number of iterations the whole training dataset is passed through. In this case the number of epochs is set to 150. The command *verbose* can take on the value 0,1 or 2. It defines how the training progress is being seen in the console (command line interpreter). 0 means none of the training progress is being shown, just the prediction after finishing training. 1 shows a progress bar. 2 shows the number of epochs and what epoch is currently running, as well does it show the training loss and validation loss and how many seconds have been needed to finish one epoch. *Callbacks* is calling the previously defined early stopping from *tf.keras.callbacks.EarlyStopping*.

```
1 model.save('lstm_model.h5')
```

This command safes the trained model as HDF5 file, a binary data format. It is now possible to apply this trained model on a new dataset with the command *load_model*. The architecture does not need to be defined again, since the training already happened before on a previous dataset. The new dataset needs to have a similar structure in data as the dataset used for training the model, only then will the model perform also well on the new dataset.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.8 Prediction

```
prediction=model.predict(Xtest)
```

After the model has been trained and hopefully learned to recognize a pattern in the data, this trained model is now applied on the test dataset. The model has so far only seen the training data, therefore is totally unbiased towards the test dataset. The prediction is done by the command *model.predict(Xtest)*. Based on the data of *Xtest* the model is now predicting an output *ytest*. This data is then compared with the true data of *ytest*.

```
prediction=model.predict(Xall)
```

For a better visualization the whole data X, reshaped as *Xall*, can be used in the command *model.predict* and is then compared with the true data of y, reshaped as *yall*.

```
prediction=scaler.inverse_transform(prediction)

yall=scaler.inverse_transform(yall)
```

The data predicted and also the values of *yall* are still scaled, that means they take on values between 0 and 1. To compare the results, the data needs to be unscaled (inversed). This is done by the command *scaler.inverse_transform*.

4.9 Visualization of the Data

```
plt.plot(prediction, color = 'green')
plt.plot(yall, color = 'black')
plt.xlabel('Timestep')
plt.ylabel('Temperature')
plt.legend(['Prediction', 'True'], loc='lower right')
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
6 plt.savefig('predictions.png')
7 plt.show()
```

The data is then visualized with the package *matplotlib*. Both curves can be plotted in one figure and thus differences in the predicted data, compared to the real data, can be seen.

```
1 plt.plot(model.history.history['loss'])
2 plt.plot(model.history.history['val_loss'])
3 plt.ylabel('loss')
4 plt.xlabel('epoch')
5 plt.legend(['train', 'test'], loc='upper right')
6 plt.savefig('Name_figure.png')
7 plt.show()
```

The plot saved in line 7 is the visualization of the course of (training) loss compared to the course of validation loss. The loss is calculated over the entire training dataset, the validation loss over the entire validation dataset. The course of loss and validation loss is an important indicator if the model has trained well and recognized a pattern in the data or was simply underfitting or overfitting. Underfitting means, the model did not recognize a pattern in the dataset at all, as a result, the validation loss is much higher than the loss. On the contrary, when a model overfits it is only memorizing data but not learning a pattern. In this case the validation loss is at first approaching the loss and is going up again after some epochs. A good fit is characterized by nearly the same value of loss and validation loss. The lower the loss and validation loss, the better the model.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 5

Results & Discussion

In this chapter, the results of a LSTM model predicting different output parameters by using PT01 as input parameter, are discussed. For this purpose LSTM models, using the data of measurement series M1, are created. All these models use the same structure of code, altough the hyperparameters and the architecture are varying and are chosen according to the best performance on the data. Since a LSTM model is only useful, if the trained model performs also properly on new datasets, it will be shown how these created models perform on the unbiased datasets of M3 and M4.

5.1 LSTM Models of the FBR

This section deals with the creation of different ANN models by using the modelling technique LSTM. These models are trained and tested by using the dataset of measurement series M1, in order to predict different output parameters. The predicted parameters are then compared with the true data. Afterwards it will be shown how the trained models perform on the new datasets of M3 and M4. For the sake of clarity, only the output parameter PT02 is predicted from new datasets, since PT02 shows the biggest time-shift to PT01 and because of that it is assumed, that it is the most difficult temperature curve to predict. For the creation of the models, only the temperatures PT01, PT02 and T1-T4 are used. In total three different models are built, tab. 5.1 gives an overview on the architecture

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and the parameters used for creating these models. Usually only the test dataset is used for prediction, but for visualization purpose the predictions of the whole data of PT02 are shown in the figures. To guarantee that the created models learned to recognize a pattern in the dataset and are not overfitted or underfitted, the loss and validation loss curves are as well visualized. If the curves of loss and validation loss converge until training stops, the model is called *a good fit*, that means the model recognized a pattern in the data. The last value of loss and validation loss indicates the difference (calculated error) between training and validation dataset when training stops.

	Model 1	**Model 2**	**Model 3**
Input Parameters	PT01	PT01	PT01
Output Parameters	PT02	T2, T3, T4, PT02	T1, T3, PT02
Shape Xtrain	(2877, 1, 1)	(2877, 1, 1)	(2877, 1, 1)
Shape Xtest	(1200, 1,1)	(1200, 1, 1)	(1200, 1, 1)
Shape ytrain	(2877, 1,)	(2877, 4)	(2877, 3)
Shape ytest	(1200, 1)	(1200, 4)	(1200, 3)
Shape Xval	(720, 1, 1)	(720, 1, 1)	(720, 1, 1)
Shape yval	(720, 1)	(720, 4)	(720, 3)
Test Size [%]	25	25	25
Validation Size [%]	20	20	20
Number of LSTM Layers	1	1	1
Number of Neurons in Layer 1	7	12	4
Activation Function	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adam	Adam	Adam
Learning Rate	0,01	0,01	0,01
Epochs	100	200	150
Batch Size	6	4	5

Table 5.1: Parameters and architecture of the different models

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2 Model 1

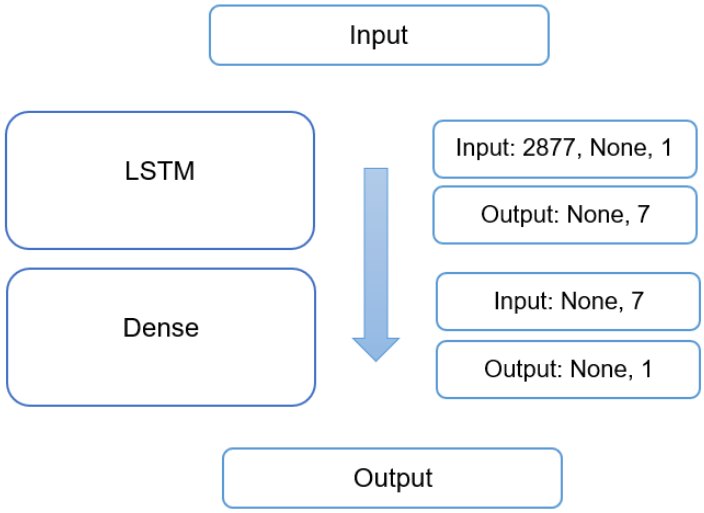

Figure 5.1: Architecture of model 1

Fig. 5.1 shows the architecture of model 1. It uses the temperature PT01 as input parameter and PT02 as output parameter. Altough the number of epochs has been chosen to 100, training stops after 58 epochs because of the early stopping command. Fig. 5.2 shows the prediction done by the trained model compared to the true data of PT02. When the state of

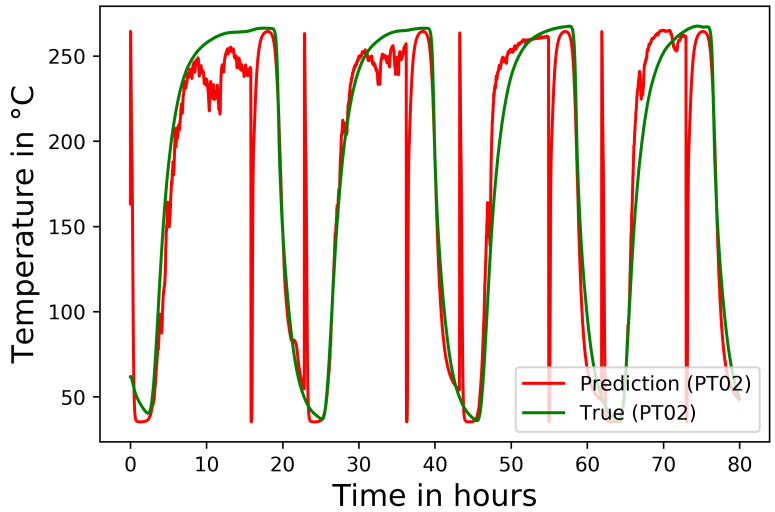

Figure 5.2: Predictions done by LSTM model 1 by using the data of the measurement series M1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

load changes from unloading to loading or the other way around from loading to unloading, the temperature curve takes a sudden turn which results in an unstable prediction. This can be seen after every abrupt turn in the course of the curve in fig. 5.2. Tab. 5.2 shows the last value of loss and validation loss when training stopped. In order to be able to make

Loss	**Validation Loss**
0,0259	0,0273

Table 5.2: Value of loss and validation loss of model 1

a statement about the learning behaviour of the model, the course of loss and validation loss must be considered. As seen in fig. 5.3, the LSTM model 1 is a good fit and the model was able to learn a pattern in the data as can be determined by the converging curves of loss and validation loss.

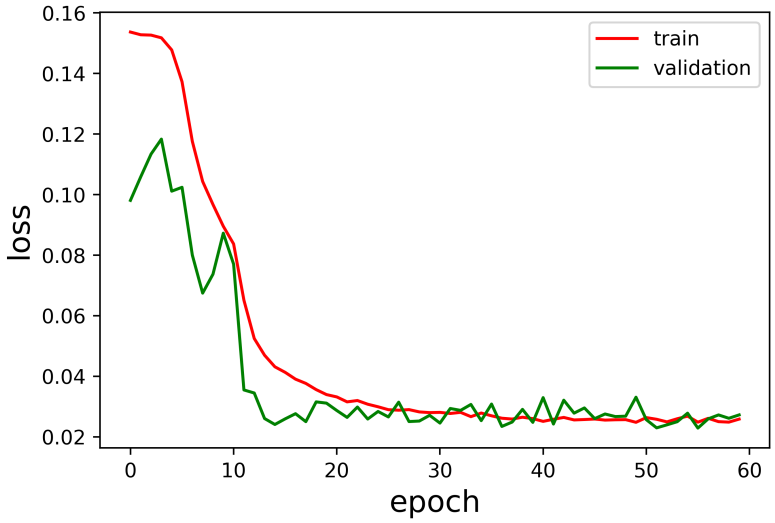

Figure 5.3: Course of loss and validation loss of model 1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Performance of Model 1

Fig. 5.4 shows the prediction of PT02 of dataset M3 done by the trained model 1. The trained model is able to predict the shape of the true PT02, but has its difficulties when predicting the values of an abrupt change in the curve. This results in a swing up or down, depending on the course of the curve.

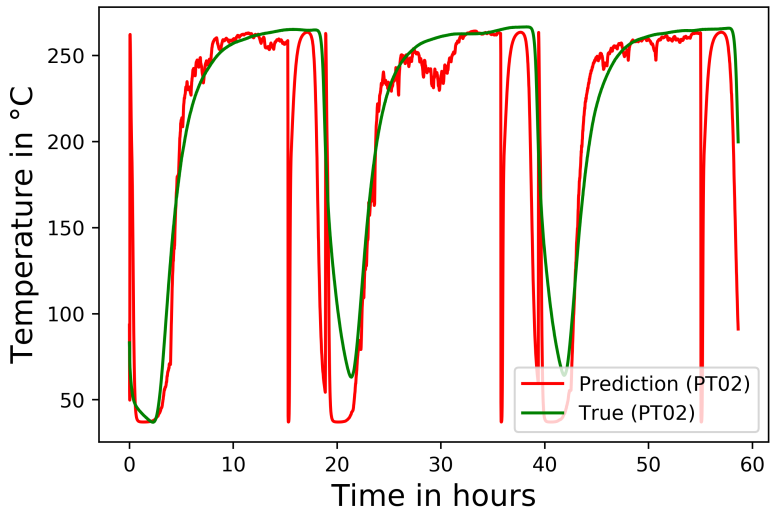

Figure 5.4: Performance of the trained model 1 to predict PT02 of dataset M3

In fig. 5.5, PT02 of dataset M4 is predicted. Overall, the course of the predicted PT02 curve follows the true course of the curve, but with a time-shift. Also, the shape of the prediction differs a lot from the true shape of PT02. Due to the fact that the temperature gradient of measurement series M4 is very large, it is assumed, that it is more difficult for the model to find a connection between the measurement series M1 and M4 than between M1 and M3. Because of the mentioned large temperature gradient of M4, the shape of the predicted curve has more similartities with M1 than with the true curve PT02 of M4.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

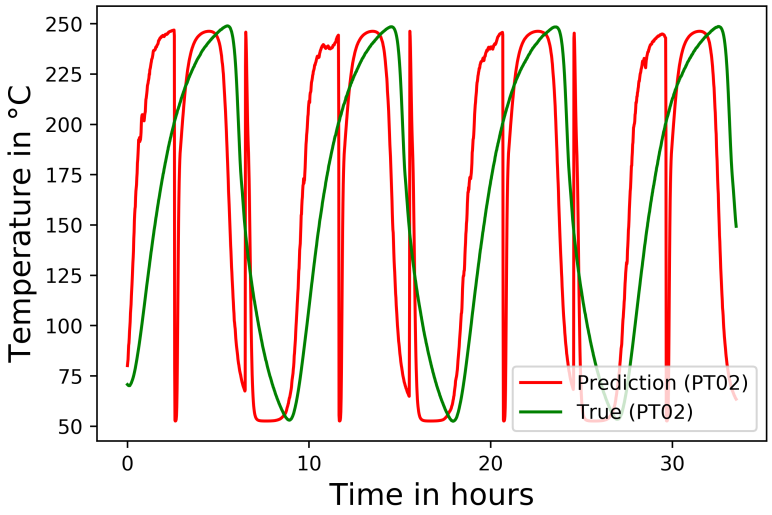

Figure 5.5: Performance of the trained model 1 to predict PT02 of dataset M4

5.3 Model 2

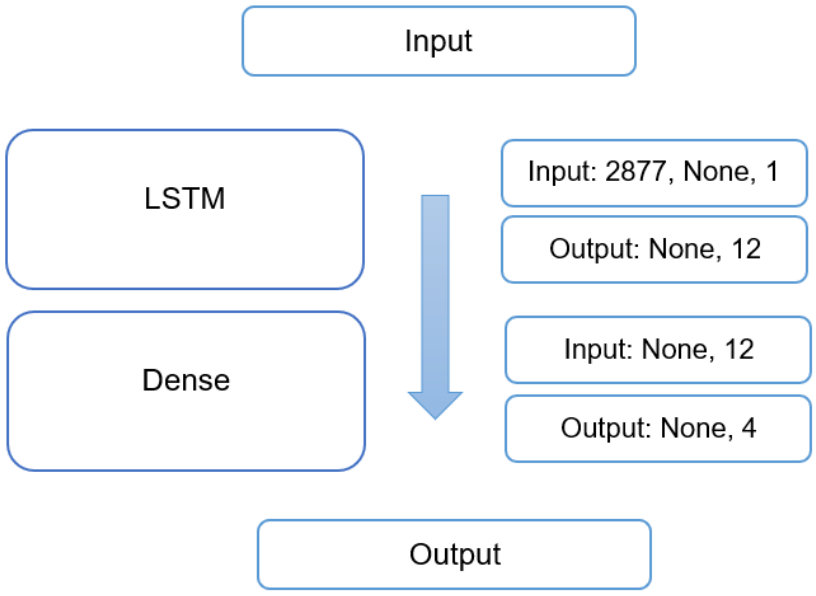

Figure 5.6: Architecture of model 2

In fig. 5.6 the architecture of the model can be seen. Model 2 uses PT01 as input parameter and T2, T3, T4 and PT02 as output parameters. In fig. 5.8 the predictions of the output parameters T1-T4 and PT02 done by LSTM model 2 are shown. In general, the predicted

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Loss	Validation Loss
0,0225	0,0282

Table 5.3: Value of loss and validation loss of model 2

curves and the true curves coincide, however, they show a deviation. The deviation between the true curve and the predicted curve is particularly noticeable in T4 and PT02. It is assumed that this is due to the fact, that T4 and PT02 are more time-delayed compared to PT01.

The course of loss and validation loss in fig. 5.7 shows the behaviour of the model which converges after 68 epochs when training stops with a loss of 0,0225 and a validation loss of 0,0282, see also tab. 5.3. The model can be considered as a good fit model, since loss and validation loss are converging.

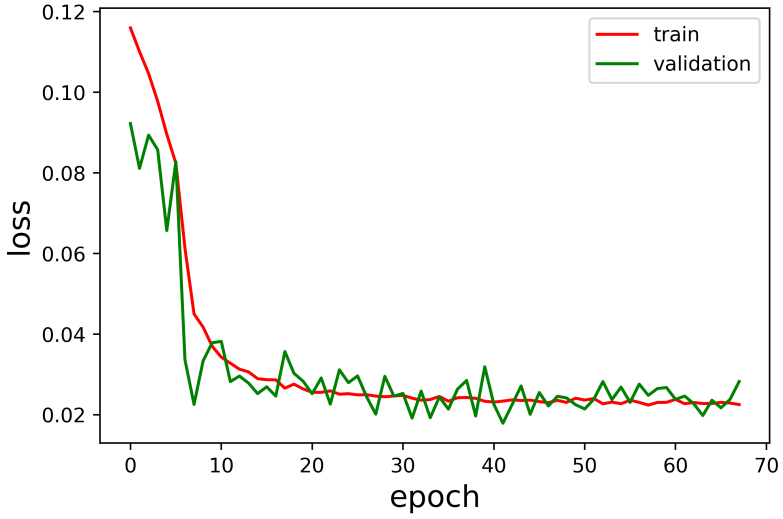

Figure 5.7: Course of loss and validation loss of model 2

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

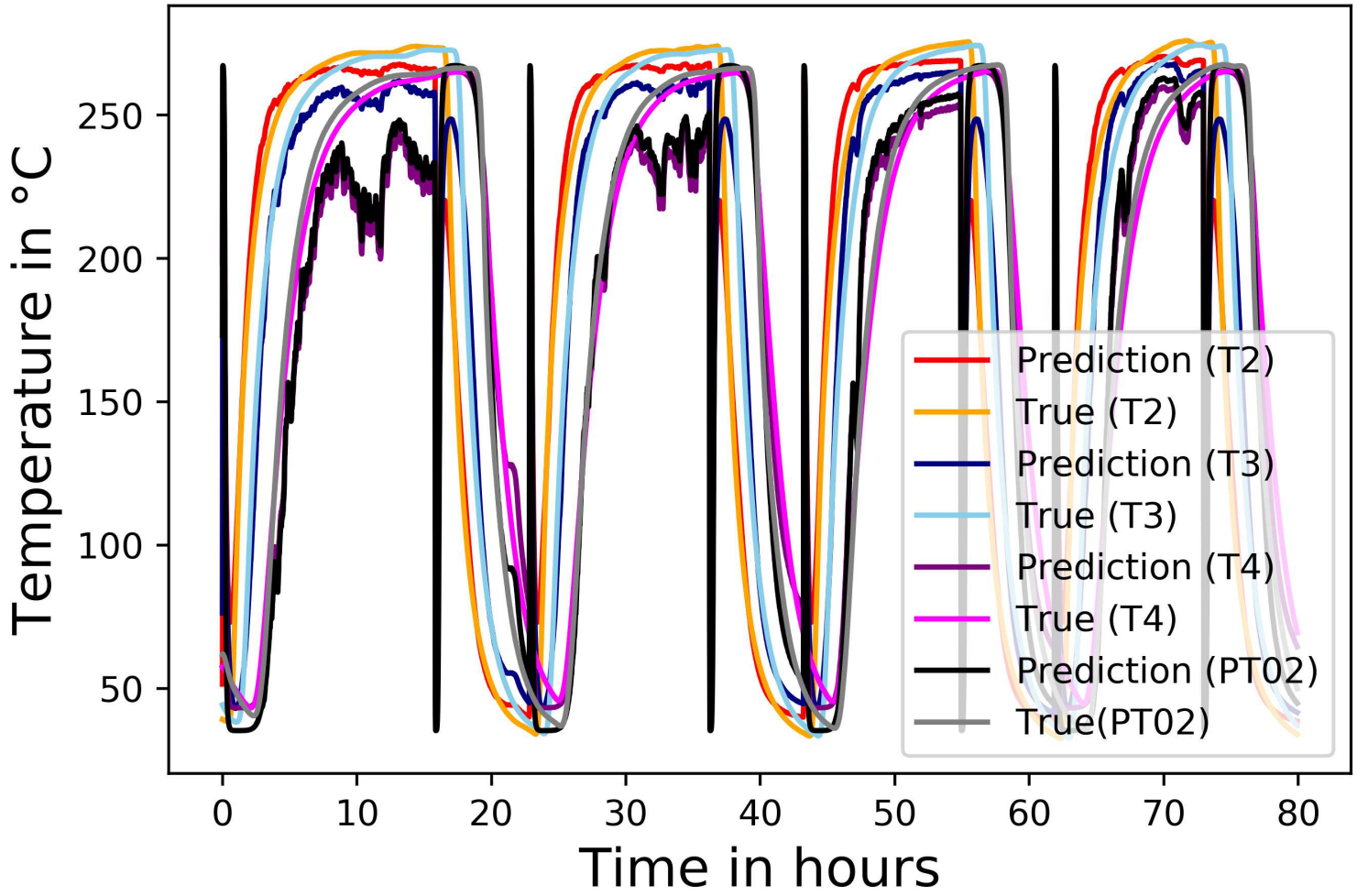

Figure 5.8: Predictions done by LSTM model 2 by using the data of the measurement series M1

Performance of Model 2

The prediction of model 2 of PT02 of the dataset M3 can be seen in fig. 5.9. The prediction follows the shape of the true data, but shows again an unstable behaviour everytime before the true curve takes an abrupt turn. Also, the model 2 is not able to predict the peak point on the right side of the loading curve, it remains slightly beneath. Fig. 5.10 shows the

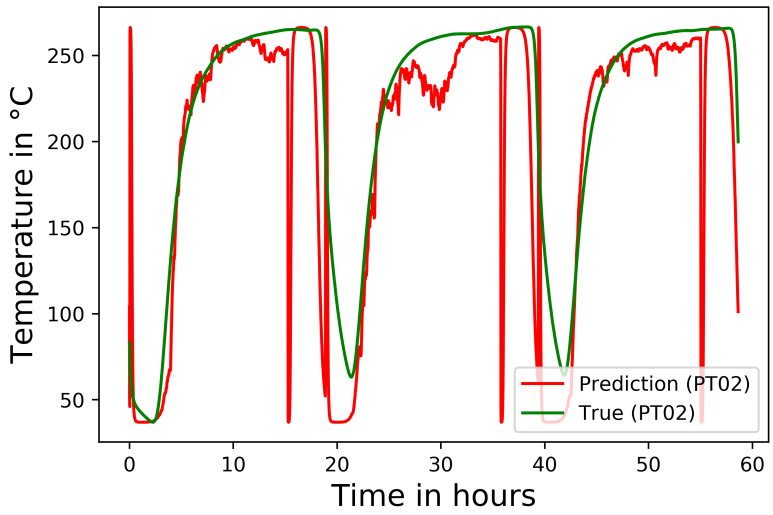

Figure 5.9: Performance of the trained model 2 to predict PT02 of dataset M3

prediction done by the trained model 2 on M4. The shape of the prediction again differs from the shape of the true curve PT02. It is presumed, that the model again struggles with the different shape of the curves in dataset M4 compared to M1 and M3.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

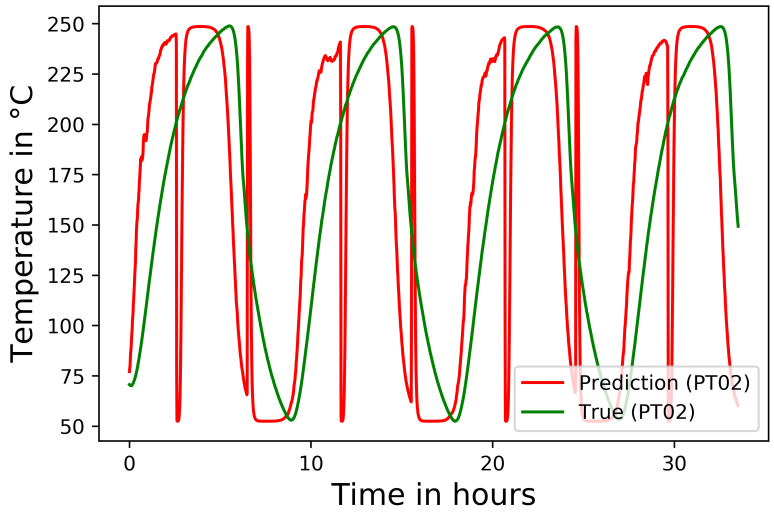

Figure 5.10: Performance of the trained model 2 to predict PT02 of dataset M4

5.4 Model 3

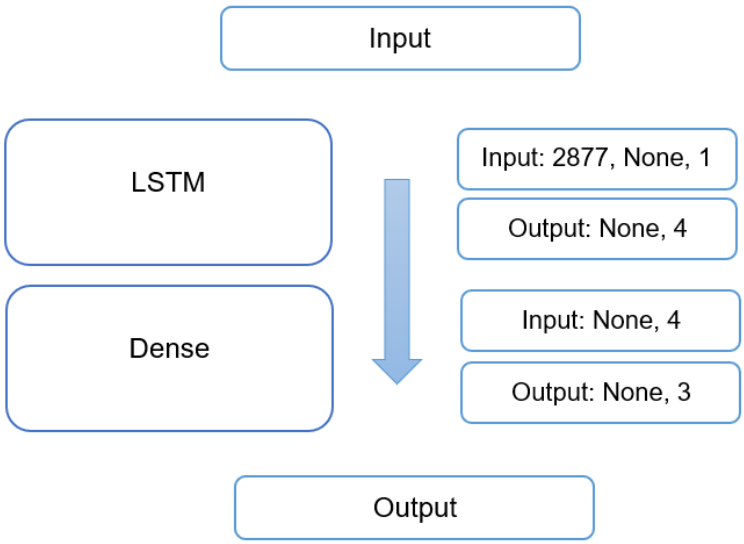

Figure 5.11: Architecture of model 3

Fig. 5.11 shows the architecture of model 3 with the input parameter PT01, whereas the output parameters are T1, T3 and PT02. Training stopped after 72 epochs with a loss of 0,0175 and a validation loss of 0,0199, as seen in tab. 5.4. Fig. 5.12 shows the predictions of T1, T3 and PT02 done by model 3 compared to the true data of T1,T3 and PT02 of measurement series M1. The model seems to find a connection between PT01 and T1, as

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

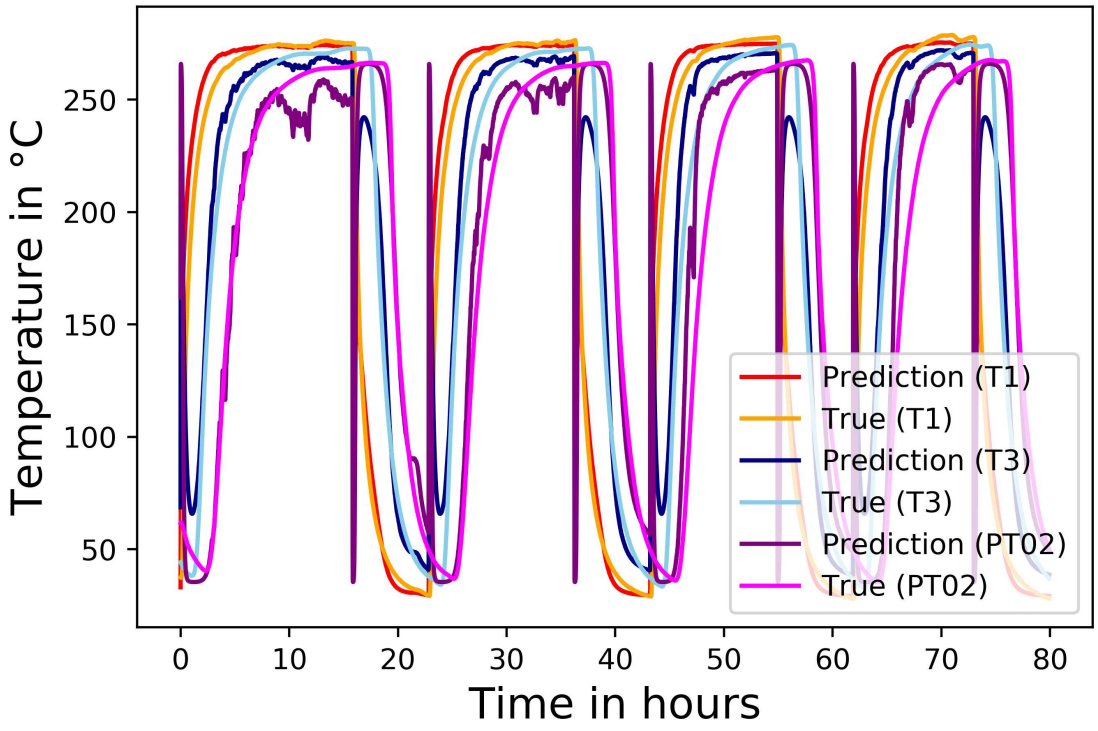

Figure 5.12: Predictions done by LSTM model 3 by using the data of the measurement series M1

the prediction of T1 is only slightly varying from the true T1, but does not manage to predict the other two curves, T3 and PT02, as accurately. The course of loss and validation

Loss	**Validation Loss**
0,0175	0,0199

Table 5.4: Value of loss and validation loss of model 3

loss in fig. 5.13 converges, thus the trained model 3 is a good fit.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

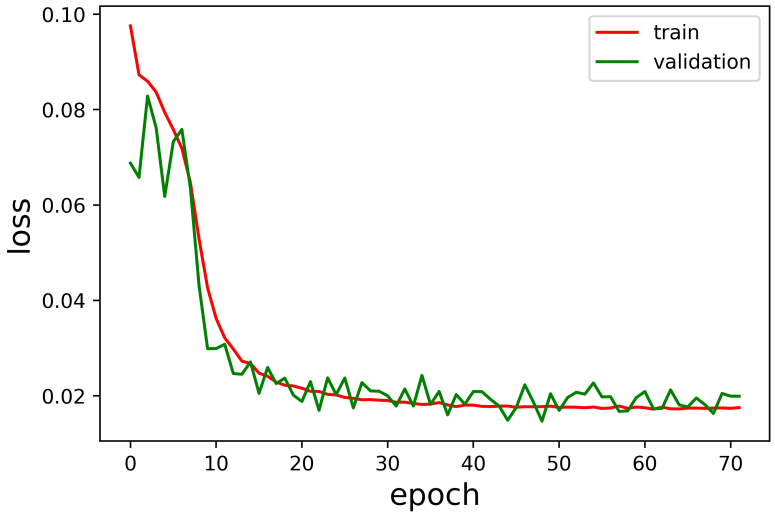

Figure 5.13: Course of loss and validation loss of Model 3

Performance of Model 3

Fig. 5.14 shows the predicted data PT02 of model 3 by using the dataset M3 versus the true PT02 of M3. Looking at the curve of the predicted PT02 in the operation mode loading, the predicted data varies more than in the operation mode unloading and is fluctuating especially when the curve reaches its peak. This fluctuation increases in the second cycle.

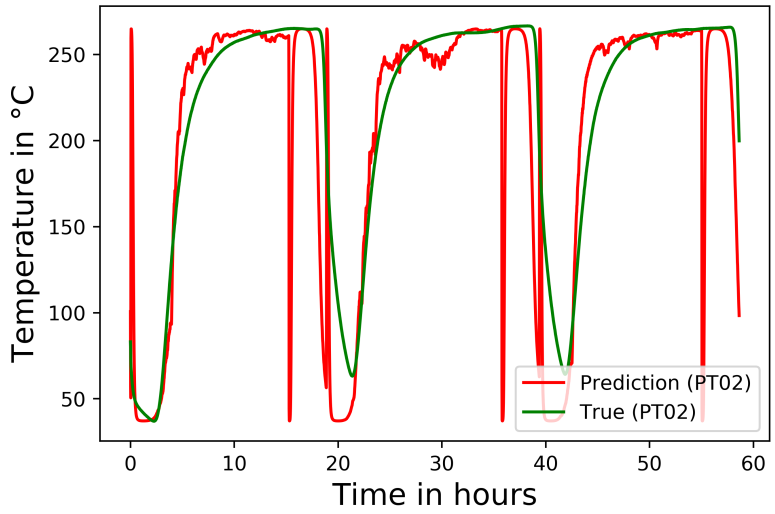

Figure 5.14: Performance of the trained model 3 to predict PT02 of dataset M3

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

This may be due to the small flattening in the second cycle in the true curve when loading. In fig. 5.15, the predicted data PT02, compared to the true data PT02 of M4 is shown. It can be seen, that the predicted curve PT02 does not coincide with the shape of the true curve of PT02 of M4. It seems that the large temperature gradient of M4 again causes difficulties for the model to recognize the true course of the curve of PT02 of M4.

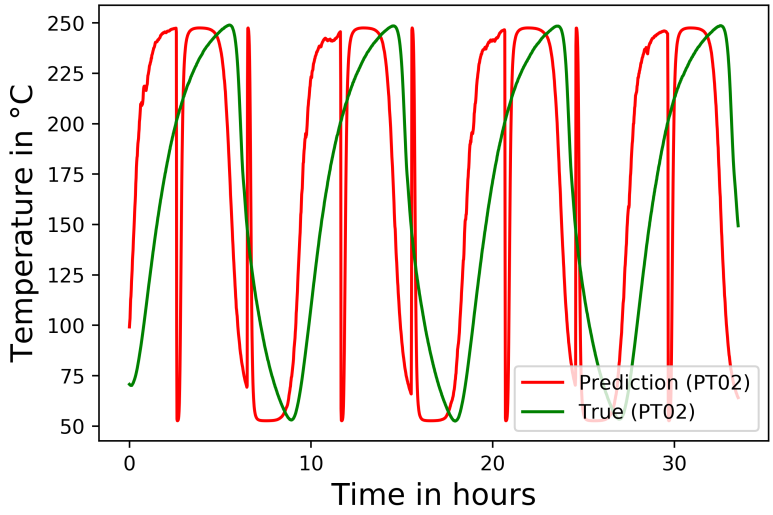

Figure 5.15: Performance of the trained model 3 to predict PT02 of dataset M4

5.5 Summary & Discussion

Data-driven modelling provides the possibility to gain information from data. Since this is a black box approach, the way this information is gained, is not always transparent and comprehensible. Especially when it comes to Neural Networks, it is often not directly evident how they came up with an output and how the learning of a pattern from input data was performed, since the architecture of these models is more complicated compared to other types of models. This means, however, that if the result is to be justified, as is the case, for example with the granting or refusal of a bank loan, a more transparent model should be selected. The advantages of Neural Networks are their ability to predict large amounts of data and to learn complex relationships. As well are they able to generalize and trained models can therefore predict on unseen data of similar shape. The above created LSTM

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

models have been all able to find a pattern in the data and to predict data that is similar to the true data. It is not clear, how the models found a connection between input and output data, due to the black box behaviour, therefore it is important to take a look at the curves of loss and validation loss. They are an indicator if the model learned a pattern (good fit) in the data or was only memorizing data (overfit), or even did not learn a pattern at all (underfit). The last value of loss and validation loss when training stops is an indicator for the error of training dataset and validation dataset compared to the true data and the lower this value is, the more accurate the model. The loss and validation loss curves of all three models show models that are a *good fit* and all finish training with a low value in loss and validation loss. The predictions done by the LSTM models are more accurate when predicting temperature curves that are only slightly time delayed to PT01. The model is still able to predict T4 and PT02, even tough T4 and PT02 show the largest time delay to PT01, but the predictions are less accurate and show fluctuating curves, especially when the operation mode is loading.

When the trained models predict on unseen data, all models are able to predict PT02 of measurement series M3 quite accurate. In all three models, the shape of the predicted curves are similar to the shape of the true curves, but all predictions show fluctuations when the operation mode is loading. All the trained models have difficulties to predict PT02 of measurement series M4. Since the model was only trained by using the dataset of M1, it may be difficult to predict the different curves of M4. This could be solved by training the model with different curves.

The created models perform similarly, none of them shows remarkable differences regarding accuracy. All models are able to predict chosen output paramters with a slight deviation, as well as to predict on unseen data with similar shape. The performance of the models may be improved by using more data, since Neural Networks peform well on large datasets. The dataset M1 consists of four cycles. If more data would be available, the model may be able to perform better on the data. Another problem could lie in the different number of cycles.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 6

Conclusion

In the course of this thesis, a data-driven model was created, that used the measured load cycles of a fixed bed regenerator, in order to create ANN models that are able to predict the temperatures in the reactor and on the outlet of the fixed bed regenerator, by solely using PT01 as input parameter. The load cycles of this energy storage have been measured over a constant time step, therefore the modelling technique LSTM was used for creating the models, since LSTM models perform well on time-series data. Also, an insight into the term industry 4.0 and other related terms was given and the reference architecture model, RAMI-4.0, described, which provides an uniform implementation of industry 4.0 methods. Another aspect of this thesis was the description of the deployment of Machine Learning models into production, by using Flask as REST API.

In total, three different models were created by using the measurement series M1. It could be shown, that altogether, the models have been able to find a pattern in the data. Although the predicted curves are slightly different, it can be seen that the predictions can still follow the true curves. However, the more time-delayed the curves were compared to PT01, the more the predictions varied. After training the models, they have been saved in order to apply them on new datasets. These new datasets vary in shape and length to M1. Both new datasets, M3 and M4, consist of three cycles, whereas the dataset of M1 consists of four cycles. M3 has a smaller temperature gradient than M1 and M4 has a larger temperature gradient compared to M1. Especially the large temperature gradient in dataset M4 seemed

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to make it difficult for the trained models to make accurate predictions on dataset M4 in all three models.

The prediction of the models on dataset M1, as well as the prediction done by the trained models on new datasets, all show an unstable course of the curve compared to the true data, when the course of the true curve takes an abrupt turn. This is the case in the data, when the energy storage changes the operation mode from loading to unloading or the other way around. This could be probably solved, by using more data. With the existence of more data, the model could more easily detect a pattern in the abrupt changes of the curves and make a more accurate prediction. Also, by using more data, it is assumed, that also the predictions of the trained models on new datasets can be more accurate and that a different curve shape and length has less influence.

All things considered, the LSTM models were able to predict data that is similar to the true data, but there are deviations. If the shape of the data varies a lot from the original data used for training, they have difficulties to make an accurate prediction. Overall it has to be said, that the LSTM model offers one variant of data-driven modelling.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Bibliography

[1] http://colah.github.io/posts/2015-08-understanding-lstms/ [website accessed on mar. 25th, 2019].

[2] https://cloudxlab.com/blog/deploying-machine-learning-model-in-production/[website accessed on apr. 29th, 2019].

[3] https://de.mathworks.com/help/stats/machine-learning-in-matlab.html [website accessed on mar. 23rd, 2019].

[4] https://medium.com/analytics-vidhya/how-to-deploy-simple-machine-learning-models-for-free-56cdccc62b8d[website accessed on apr. 29th, 2019].

[5] https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0 [website accessed on mar. 25rd, 2019].

[6] https://skymind.ai/wiki/neural-network [website accessed on apr. 11th, 2019].

[7] https://www.datacamp.com/community/tutorials/machine-learning-models-api-python [website accessed on apr. 26th, 2019].

[8] https://www.gartner.com/it-glossary/digitalization/ [website accessed on apr. 11th, 2019].

[9] https://www.netobjex.com/how-humans-are-empowering-digital-transformation-in-industry-4-0/ [website accessed on jun. 23rd, 2019].

[10] https://www.researchgate.net/figure/common-neural-network-activation-functions_fig7_305881131 [website accessed on apr. 11th, 2019].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[11] https://www.sentryo.net/the-4-industrial-revolutions/ [website accessed on jun. 23rd, 2019].

[12] Rami-4.0: https://www.dke.de/de/themen/industrie-4-0/rami-4-0 [website accessed on jan. 3rd, 2019].

[13] Digitalization and energy, 11/2017.

[14] *Methoden wissensbasierter Systeme: Grundlagen — Algorithmen — Anwendungen*. Vieweg, Wiesbaden, 3., erweiterte auflage edition, 2006.

[15] Konzept für die integration der mehrkörpersimulation in eine engineeringplattform, 2010.

[16] *Cyber-Physical Systems: Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion*, volume 11 of *acatech Position*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[17] *Mathematik im Fraunhofer-Institut: Problemgetrieben — Modellbezogen — Lösungsorientiert*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015 edition, 2015.

[18] Ethem Alpaydın. *Maschinelles Lernen*. Oldenbourg Verl., München, 2008.

[19] Mariette Awad and Rahul Khanna, editors. *Efficient learning machines: Theories, concepts, and applications for engineers and system Designers / Mariette Awad, Rahul Khanna*. The expert's voice in machine learning. Apress Open, New York, 2015.

[20] Udo Döbrich, Martin Hankel, Roland Heidel, Michael Hoffmeister, and DIN e.V. *Basiswissen RAMI 4.0 : Referenzarchitekturmodell und Industrie 4.0-Komponente Industrie 4.0*. Beuth Verlag, Berlin, GERMANY, 2017.

[21] Philipp Drochter. *Auslegung, Konstruktion und Errichtung eines Festbettregenerators*. Wien, 2016.

[22] Jörg Frochte. *Maschinelles Lernen : Grundlagen und Algorithmen in Python*. Hanser, München, 2018.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[23] Mukrimin Sevket Guney and Yalcin Tepe. Classification and assessment of energy storage systems. *Renewable and Sustainable Energy Reviews*, 75:1187–1197, 2017.

[24] Andreas Hauer, Stefan Hiebler, and Manfred Reuß. *Wärmespeicher*. BINE-Fachbuch. Fraunhofer IRB Verl., Stuttgart, 5., vollst. überarb. aufl. edition, 2013.

[25] Frank Herrmann. The smart factory and its risks. *Systems*, 6(4):38, 2018.

[26] R. Kitchin, T. P. Lauriault, and M. W. Wilson. *Understanding Spatial Media*. SAGE Publications, 2017.

[27] Rudolf. Kruse, Christian. Borgelt, Christian. Braune, Sanaz. Mostaghim, and Matthias. Steinbrecher. *Computational Intelligence*. Texts in Computer Science. Springer London : Imprint: Springer, London, 2nd ed. 2016 edition.

[28] Andreas Michalka. *Experimentelle Untersuchungen eines Festbettregenerators mit feinem Kies als Speichermaterial*. Wien, 2018.

[29] Michael A. Nielsen. Neural networks and deep learning, 2015.

[30] Razvan Pascanu, Tomas Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013*, 2012.

[31] Tariq Rashid and Frank Langenau. *Neuronale Netze selbst programmieren : ein verständlicher Einstieg mit Python*. Animals. O'Reilly, Heidelberg, 2017.

[32] Mario Schirnhofer. *Anforderungen und Funktionen eines I4.0 Spannmittels exemplarisch am Modell Aufspannturm*. Wien, 2018.

[33] Thomas Schulz, editor. *Industrie 4.0: Potenziale erkennen und umsetzen*. Vogel Business Media, Würzburg, 1. auflage edition, 2017.

[34] Michael Sterner and Ingo Stadler. *Energiespeicher*. Springer Vieweg, Berlin [u.a.].

[35] Ivan Vasilev. *Python deep learning*. Packt Publishing Ltd, Birmingham, second edition edition.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[36] Ramon Wartala. *Praxiseinstieg Deep Learning : mit Python, Caffe, TensorFlow und Spark eigene Deep-Learning-Anwendungen erstellen.* O'Reilly, Heidelberg, 1. auflage edition, 2018.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Figures

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Tables

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.