Application of Data Mining and Machine Learning Methods to Industrial Heat Treatment Processes for Hardness Prediction

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc., M.Eng. Yannick Lingelbach
geb. in Ostfildern

Tag der mündlichen Prüfung:	14. Dezember 2023
Hauptreferent:	Prof. Dr.-Ing. habil. Volker Schulze
Korreferent:	Prof. Dr. Ralf Mikut

Abstract

Industrial heat treatment processes are usually optimized for many years during series production, but the number of factors influencing hardness and the high measurement noise of end-of-line hardness testing make it increasingly difficult to further reduce costs or optimize the process. In this work, a data mining framework for batch processes was developed and applied to harness the rich data sources that fill up over time for two pilot use cases, namely bainitization of 20 000 and case hardening of 7 000 batches. All necessary data sources, preprocessing, cleansing, and feature extraction steps are outlined along with the corrections for drifts. A benchmark for the maximum achievable predictability was derived to assess the economic benefit of a use case at an early stage. The framework then applies step-by-step data mining techniques to quantitatively break down variance contributors such as material, production line, measurement device, batch and measurement position, as well as their interactions and dynamic behavior over time. Based on these factors, a set of feature subset selection, machine learning pipeline optimization, as well as training and evaluation approaches were explored in order to find the most robust prediction strategy for thermally treated components. For case hardening a custom solution, the hidden-state-pipeline was developed. Finally, an industry pilot shows how to implement these models in daily operations and transfer the process to other component types to reduce the costs of end of line tests.

Kurzfassung

Industrielle Wärmebehandlungsprozesse werden in der Serienproduktion üblicherweise über Jahre hinweg optimiert, aber die Vielzahl der Faktoren, die die Härte beeinflussen, und das hohe Messrauschen der End-of-Line-Härteprüfung machen es zunehmend schwieriger, die Kosten weiter zu senken oder den Prozess zu optimieren. In dieser Arbeit wurde ein Data-Mining-Framework für Batch-Prozesse entwickelt, um die umfangreichen Datenquellen, die sich im Laufe der Zeit ansammeln, für zwei Pilot-Anwendungsfälle nutzbar zu machen, nämlich das Bainitisieren von 20 000 und das Einsatzhärten von 7 000 Chargen. Alle notwendigen Datenquellen, Vorverarbeitungs-, Bereinigungs- und Feature-Extraktionsschritte werden zusammen mit den Korrekturen für Drifts dargestellt. Es wurde ein Benchmark für die maximal erreichbare Vorhersagbarkeit abgeleitet, um den wirtschaftlichen Nutzen eines Anwendungsfalls frühzeitig zu bewerten. Das Framework wendet dann schrittweise Data-Mining-Techniken an, um Varianzeinflüsse wie Material, Produktionslinie, Messgerät, Chargen und Messposition sowie deren dynamisches Verhalten über die Zeit quantitativ aufzuschlüsseln. Auf der Grundlage dieser Faktoren wurden eine Reihe von Featureauswahlverfahren, verschiedene Pipeline-Optimierungen für maschinelles Lernen sowie Trainings- und Bewertungsansätze untersucht, um die robusteste Vorhersagestrategie für thermisch behandelte Komponenten zu finden. Für das Einsatzhärten wurde eine maßgeschneiderte Lösung, die Hidden-State-Pipeline, entwickelt. Schließlich zeigt ein Industriepilot, wie diese Modelle im täglichen Betrieb implementiert und der Prozess auf andere Komponententypen übertragen werden kann, um die Kosten für die End-of-Line-Prüfung zu reduzieren.

Preface

This thesis was written during my employment at the Robert Bosch GmbH in cooperation with the Karlsruhe Institute of Technology (KIT). This section is a tribute to all the helping hands that made this work possible.

First and foremost, I'd like to thank my advisor Dr. Thomas Waldenmaier (Corporate Research), for his endless patience, unlimited support, guiding words, motivation, and open ear during frustrating times: "This is the way." Many thanks also to my supervisor Dr. László Hagymási (Powertrain solutions), who paved the way for access to data, material, and experts, connected the various business units and made my work visible throughout the company, as well as Sebastian Bannert for supporting my project. My gratitude also goes to Christian Derra and Toni Gazinkovski for their guidance and explanation of the equipment and sensors on the production side. Special thanks go to Johannes Fitz for the numerous insightful discussions and automation of my model with OpenShift. Last but not least, a big thank you to my organizing team of the PhD-exchange-community 'Machine Learning Methods', which I had the pleasure to co-found with Jakob Lindinger.

I would also like to express my deepest gratitude to my KIT supervisors. This work would not have been possible without Prof. Dr. Volker Schulze at the Institute for Applied Materials–Material Science and Engineering and his courage to offer an interdisciplinary Ph.D. in such a new field. I would also like to thank Prof. Dr. Ralf Mikut at the Institute for Automation and Applied Informatics for the many valuable comments and

insights on data mining and machine learning which significantly improved the analytic framework.

An unconventional credit goes to composer Worakls, whose music has steadily carried me through even the most intricate coding sections.

Finally, I most profoundly thank my family and friends. My mom, for delicious food, walks outdoors, and mood elevation during my more obnoxious moments. Finally, I express my deepest thanks to my lovely wife, Katharina, for serious proofreading, endless discussions about statistics and modeling approaches, as well as emotional support.

Eat your vegetables, floss your teeth, remember to say:
"It's difficult to quantitatively assess
the relative contribution of gens and environment
to a particular trade when they interact."

- ROBERT M. SAPOLSKY

Contents

List of Figures

List of Tables

Acronyms and Symbols

Acronyms

AUC	Area under the curve
CHD	Case hardening depth
CNN	Convolutional neural network
DNN	Deep neural network
El	Elongation
FN	False negatives
FMEA	Failure mode and effects analysis
FP	False positives
FPR	False positive rate
GA	Genetic algorithm
GB	Gradient boosting
GDOES	Glow-discharge optical emission spectroscopy
GRU	Gated recurrent unit
GWO	Grey wolf optimization
$\mathbf{H_P}$	Hollomon-Jaffe parameter

HV	Hardness in Vicker
IE	Impact energy/strength
IIR	Infinite impulse response
KDE	Kernel density estimation
KNN	K-nearest neighbors
LR	Linear regression
LSTM	Long-short term memory
MAE	Mean absolute error
ML	Machine learning
MSE	Mean squared error
NaN	Not a number
NN	Neural network
PCA	Principal component analysis
PDF	Probability density function
RMSE	Root mean squared error
RNN	Recurrent neural network
ROA	Reduction of area
SD	Standard deviation
SQL	Standard query language
SVM	Support vector machine
TN	True negatives
TP	True positives

TPR	True positive rate
TPOT	Tree-based pipeline optimization tool
TUS	Temperature uniformity survey
UTS	Ultimate tensile strength
YS	Yield strength

Abbreviations

a.k.a.	also known as
cf.	confer
chem. comp.	Chemical composition
meas. pos.	Measurement position
wt.	Weight

Indices

A	Austenitization
bl	Baseline
BS	Bainite start
C	Carbon
I	Isothermal
max	Maximum
min	Minimum
MS	Martensite start

sd	Standard deviation
T	Tempering
Q	Quenching

Operators and Symbols

Δ	Difference between two values of same unit
$x_{\mathbf{kurt}}$	Kurtosis of values in vector $\mathbf{x}$
n	Number of samples belonging to one set
r	Pearson correlation coefficient
R^2	Coefficient of determination
t	Time
T	Temperature
$x_{\mathbf{skew}}$	Skewness of values in vector $\mathbf{x}$
$x_{\mathbf{sd}}$	Standard deviation of values in vector $\mathbf{x}$
$\mathbf{x_{c,s}}$	Vector of measurements of channel c in segment s
y_i	The i^{th} observed value of the outcome, i^{th} label
$\hat{y}_i$	The prediction outcome of the i^{th} sample
$\bar{y}$	Mean of labels of n samples belonging to one set
$\sum_i^n$	The summation operator over the index i

Chemical Acronyms

C	Carbon

Cr	Chromium
C_2H_2	Ethyne, referred to as acetylene
Mn	Manganese
Mo	Molybdenum
Ni	Nickel
Si	Silicon

1 Introduction

Heat treatment of metals has been an established, widespread, and important processing step for centuries, with high significance for mechanical engineering, mobility concepts, consumer goods, and the economy in general. Parallel to the development of new materials and heat treatment technologies, new and innovative heat treatment processes are constantly being developed and further optimized in series production applications. These heat treatment processes not only modify the manufacturing properties of steel components for simple and cost-effective machining, but also allow tailored property specifications to achieve high functionality for a wide range of components with maximum stress resistance.

For economic reasons, usually the largest possible number of components are heat treated together in one batch. Thus, depending on the batch position, each component faces its own thermal and temporal sequence, which also leads to scattering in the local alloy composition of the components in thermochemical heat treatment processes. In total, this results in a wide variety of achievable results from component to component, from batch to batch, and from furnace to furnace. Therefore, usually at least one specimen is taken from each batch from a test position (determined by preliminary tests or experience) and subjected to quality assurance tests. Taking all influencing variables into account (with special emphasis on measurement noise), rather large tolerances of around $\pm$ 50 HV must be provided for hardness tests. To keep this spread relatively small and the process secure, its process variables critical for heat treatment – such as temperature, pressure, process gas composition – are identified (e.g., by

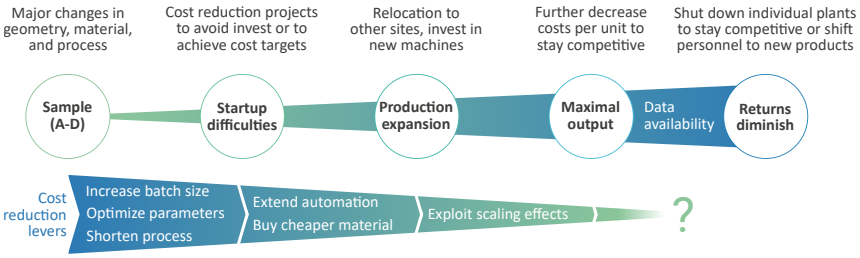


Figure 1.1: Schematic chronological stages of component production-live-cycle along with common cost reduction levers to increase profitability. The samples (A-D) are prototypes with increasing maturity level. A prototype at stage D has the required properties and is ready for production, [own representation]

means of a process failure mode and effects analysis (FMEA)) and monitored by appropriate sensor technology. This is in order to detect serious deviations from target variables already occurring during heat treatment and to issue appropriate alarms or messages to the operator. Alarms and quality test results, as well as all sensor signals, are stored as time series for traceability purposes in the field.

In order to reduce the cost of heat treatment per unit and subsequent testing, a plethora of methods are employed during production ramp-up and subsequent scale-up. Figure 1.1 outlines these stages chronologically with the corresponding levers for cost reduction from the first prototypical samples of a component, through expansion and maximal production, to diminishing yields. Initially, low hanging fruits like increased batch size (i.e., more units produced per heat treatment cycle) and optimization of parameters (e.g., higher or lower temperatures may allow shortening of the process or increase the quality) are reaped, costing little while having a sizable benefit. Further down the road, when startup difficulties are overcome and production roars, more steps can be automated (e.g., automatic instead of manual setup of batches[1]) and scaling effects exploited

[1] Batches can contain hundreds to several thousands of components.

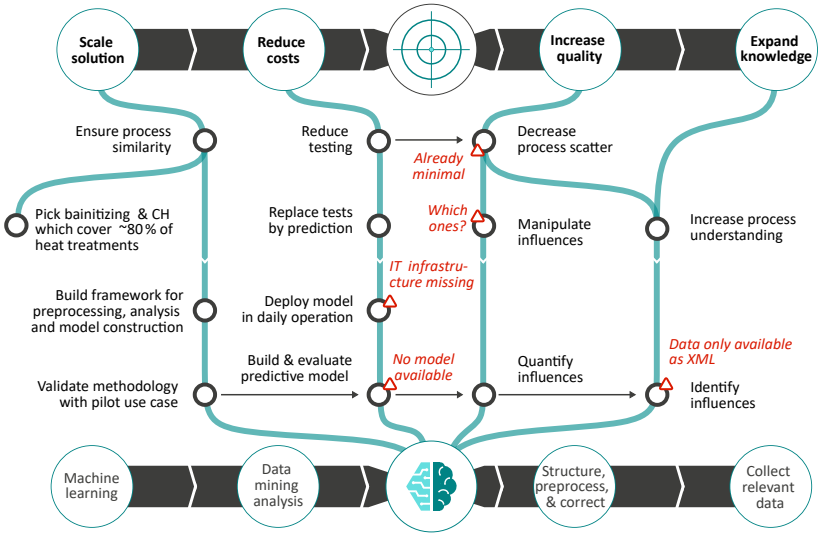


Figure 1.2: Goal of thesis and motivation to employ data mining and machine learning methods, [own representation]

(e.g., lower cost purchasing or improved machine utilization) to cut direct labor and material costs. Finally, towards maximal output, it becomes increasingly difficult to boost the profit margin as most levers are already exhausted. Further process optimization attempts may lead to less robust results, while quantification and manipulation of the influences of preceding process steps (up to steel production) are difficult to attain.

Fortunately, as the Industry 4.0 wave sweeps across manufacturing companies, countless projects are being set up to store and connect data from the manufacturing processes over the years, most of which are piling up untouched and unused in folders or databases. This thesis is one of these projects, that actually follows through. Digging into that goldmine of production data to identify and utilize its potential for further cost reduction and enhanced product quality is its objective. Figure 1.2 maps the milestones to be accomplished on the journey towards these goals, discussed below from top to bottom and from right to left.

Everything begins, as it must, with the expansion of knowledge by increasing process understanding. For this purpose, process data must be evaluated, which is currently only available as an XML file for each heat treated batch. Detecting influences requires extracting, structuring, and processing the data from these files. To increase component quality or optimize the reliability of assurance measures, the variance around the target properties (which should already be minimal after years of optimization) has to be further reduced. This presupposes that responsible influences are quantitatively known and can be manipulated. A goal with greater leverage is cost reduction through reduced testing, which will be central to this dissertation. Industrially available standard test methods, such as destructive hardness testing, are subject to considerable scatter and incur costs in the form of specimen preparation, test equipment, and laboratory personnel, but could be replaced by a predictive model deployed in daily operations. This in turn presuppose that influences are known and an IT-infrastructure as well as a model for deployment are available. Data mining and machine learning (ML) methods will be applied to build such models and perform the necessary analyses to quantify the influences, after aggregating all relevant data sources and transforming their data into a usable, structured format.

Finally, a cost reduction strategy that works can be scaled to similar processes[2]. Such scaling assumes that a validated framework for data preprocessing, analysis, and model building for heat treatment processes is in place. Hence, this thesis will develop and use such a data mining framework shown in Figure 1.3 as a structural approach for batch processes in particular and heat treatment in general to attain the aforementioned goals.

After motivating and explaining the structure of this thesis in the introduction, Chapter 2 outlines the state of the art for heat treatment and

[2] The two use cases bainitizing and case hardening (CH) already cover approx. 80 % of the heat treatments that are usually applied on an industrial scale [13].

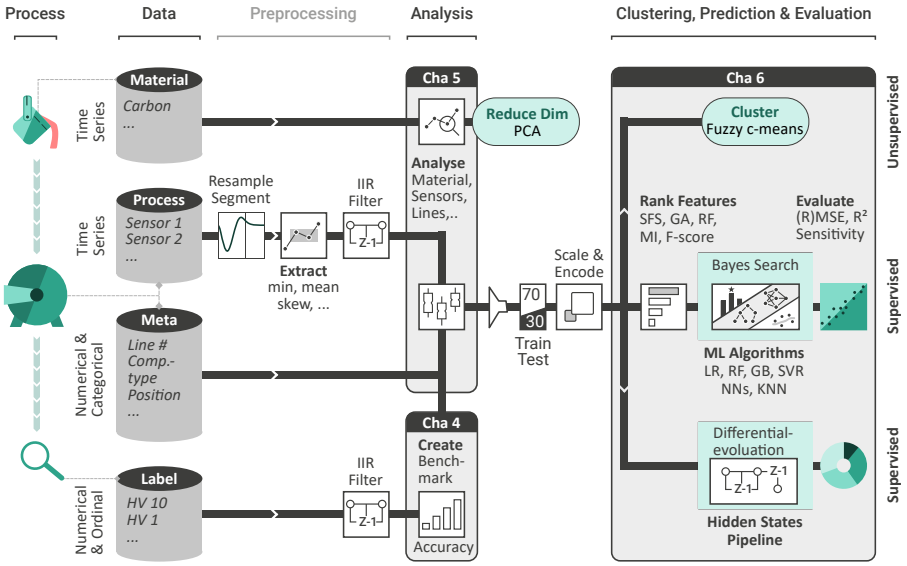


Figure 1.3: Detailed framework for data mining and machine learning in heat treatment processes (Cha: Chapter, Meta: Metadata)

data mining, as well as their combination. Chapter 3 details all materials and data (e.g., production lines, sensors, and test procedures) shown in the left part of Figure 1.3 and methods (e.g., preprocessing, feature extraction, analysis, and ML algorithms), mostly to the right, used for analysis in subsequent chapters. To understand the properties of the quality measures (labels) we seek to predict later on, Chapter 4 sheds light on their distribution, behavior over time, and influence of batch position. It also quantifies the error inherent in the measurement procedure and utilizes the findings to create a prediction benchmark (i.e., achievable prediction error). In Chapter 5 individual effects of material, process and metadata are further quantified. The knowledge gained is then used in Chapter 6 to:

1. create robust prediction or forecasting pipelines for the hardness of a component after heat treatment,

2. explain most of the variance in the hardness distribution, and
3. propose a cost reduction strategy that recommends how many test pieces can safely be replaced by a prediction.

Actual deployment is described in detail in Chapter 7 alongside the IT infrastructure for day-to-day operation at the Bosch production plant in Stuttgart to validate the framework. Finally, Chapter 8 summarizes the findings and provides an outlook for further research.

2 State of the Art

This chapter provides a general introduction to the two central fields of research. To warm up, Section 2.1 introduces heat treatment of metallic components, followed by data driven methods in Section 2.2 focusing on data mining aided by machine learning. An application of the latter to the former is given in Section 2.3. Finally, Section 2.4 poses the open research questions that will be addressed in this thesis.

2.1 Heat Treatment

Steel components are manufactured by many different process steps such as melting, forging, milling, turning, hardening, and grinding. In general, steel in its 'soft state' is easier to process and causes less wear on machine components. Conversely, for their subsequent practical use, steel components often must resist surface abrasion, mechanical stress, and high strain or pressure. Thus, they seldom simultaneously meet the requirements they are supposed to have during manufacturing and subsequent utilization. Many components are, therefore, hardened as one of the last steps in their production chain in order to increase the mean time to failure and prevent early breakdown. Hardening is achieved by a change in the atomic microstructure via heat treatment resulting in desired material properties optimally suiting the application [39]. DIN 4885 [100] describes this heat treatment process as:

> "A Series of Operations in the course of which a solid ferrous product is totally or partially exposed to thermal cycles to

> bring about a change in its properties and/or structure. The chemical composition of the ferrous product may possibly be modified during these operations."

Depending on the desired properties, numerous procedures are available for hardening. Among other parameters, they differ in heat supply (e.g., by furnace, induction, or laser), the quenching media (e.g., oil, salt, or gas), the addition of further alloying elements (e.g., carbon or nitrogen), and temperature profile. Some procedures target the whole cross-section of a component, others only aim to harden the surface layer [40,90]. Figure 2.1 depicts the two heat treatment procedures relevant for this thesis: the first being a bainitic treatment and the second being a case hardening by low pressure carburization with high pressure gas quenching, subsequent deep freezing, and tempering. Both treatments are explained in the following, along with their specific applications.

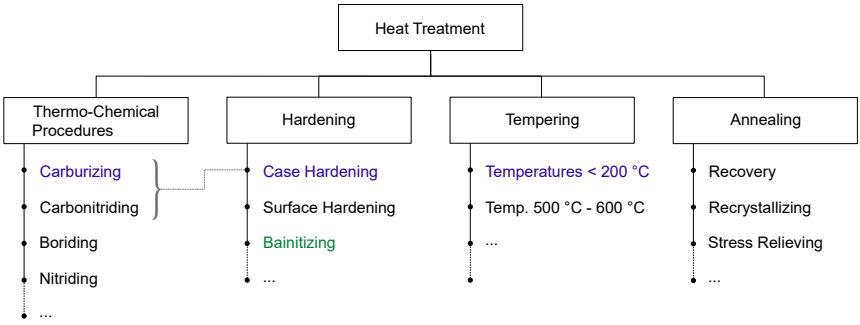


Figure 2.1: Overview of heat treatment procedures based on [88]

2.1.1 Bainitizing

Components of a diesel injection pump must withstand the stress of fast changing high pressure cycles. To achieve high durability and hardness throughout the whole cross-section of the components (e.g., cylinder heads)

with little distortion, the steel can be transformed into bainite[1]. A typical temperature profile of a bainitic heat treatment is drawn in Figure 2.2. According to [88, 100], bainitization can be divided into three process steps: *Austenitization* (heating to austenitizing temperature T_{A} and holding for a sufficiently long period Δt_{A}), *quenching* (cooling at a rate fast enough to avoid the formation of ferrite or pearlite to a temperature T_{Ia} above the martensite start temperature T_{MS}), and *isothermal transformation* (partial or total transformation of the austenite to bainite). Quenching includes the option for (1) a martensitic nucleation (quenching below T_{MS} for a short time, to obtain first martensite needles), while the subsequent transformation is performed either (2*a*) at T_{Ia} (one-stage bainitizing) or (2*b*) with a heating to T_{Ib} after Δt_{Ib1} (two-stage bainitizing).

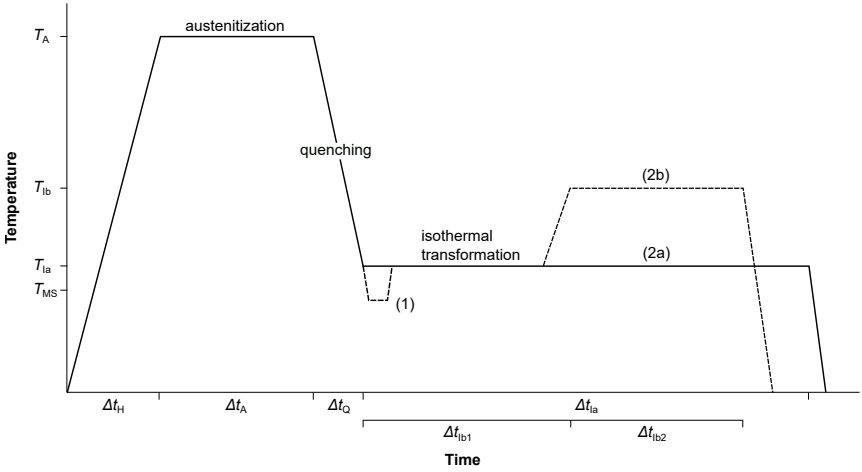


Figure 2.2: Schema of a time-temperature profile for bainitizing [26]

[1] Bainite DIN 10052 [103]: "Metastable constituent formed by the decomposition of austenite in a temperature interval between the temperature at which pearlite forms and that at which martensite starts to appear. It consists of supersaturated ferrite in which carbon has been finely precipitated in the form of carbide."

Austenitization

Austenitization consists of heating components for a time Δt_H until the whole component has reached temperature T_A and a subsequent holding for a time Δt_A. As the component's core temperature lags behind its surface, the heating process is divided into heat up and equalization[2]. It serves the purpose of obtaining a desired microstructure defined by a characteristic distribution of chemical elements (homogeneity), size of grain as well as number and size of carbides. These factors determine the conversion kinetics of all possible phase transformations during the subsequent quenching and isothermal transformation.

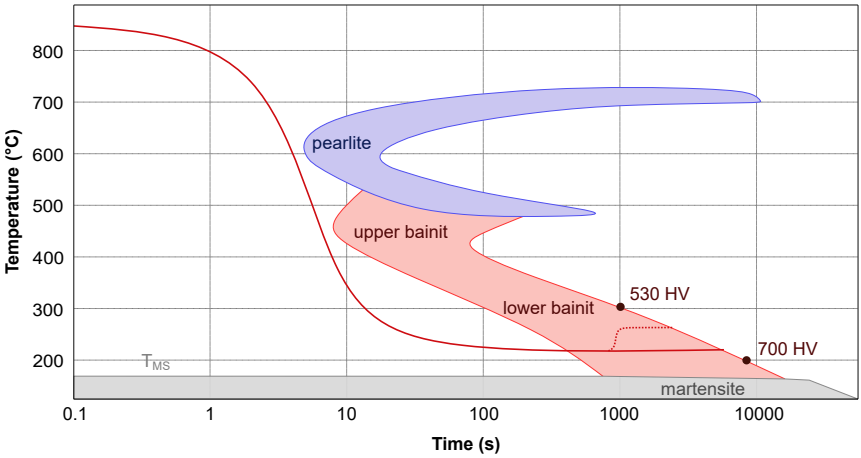


Figure 2.3: Schematic time-temperature-transformation diagram with isothermal bainitic phase transformation of 10 mm round rod made of 100Cr6 [121]

[2] Period during which core temperature converges to surface temperature, whereby the latter is already stable.

Quenching

Quenching is the most time critical step, since the rate of cooling, among other things depending on the alloy composition and the austenitizing conditions, must be fast enough to avoid the formation of pearlite or ferrite as indicated by Figure 2.3. It depicts a continuous-cooling transformation diagram for the steel 100Cr6 which alongside its eponymous application in roller bearings, is also used widely in the automotive industry, especially for diesel injection systems, because high carbon and chromium content ensure high strength and hardenability [18]. When the final temperature T_{Ia} is reached after a time Δt_{Q}, the quenching is completed. The closer the final temperature T_{Ia} lies above T_{MS}, the longer the time necessary until complete transformation to bainite, but also the more desirable the resulting microstructure of bainite, as it is more fine-grained [143].

Isothermal transformation

During isothermal soaking the transformation to a bainitic microstructure is completed. Depending on steel and austenitization conditions, the phase transformation may take a long time, therefore, it would be beneficial to shorten this process. By increasing the soaking temperature after a time Δt_{Ib1} to T_{Ib}, the transformation is accelerated with the effect of significantly higher fatigue resistance at the cost of reduced compressive strength. In return, the time for complete transformation can be reduced by 75 % (i.e., $\Delta t_{\mathrm{Ib1}} + \Delta t_{\mathrm{Ib2}} = \frac{1}{4}\Delta t_{\mathrm{Ia}}$) [34].

A sensitivity analysis of the heat treatment parameters on the resulting hardness is given in Table 2.1.

Table 2.1: Important process parameters for bainitic treatment by [26]

Parameter (↑)	**Hardness**	**Influence**
T_{A}, Δt_{A}	↑	Increased amount of dissolved carbon in austenite leads to stronger lattice distortion
T_{Ia}, T_{Ib}	↓	Increased carbon diffusion leads to relaxation of lattice distortion
Δt_{Ia}, $\Delta t_{\mathrm{Ib1,2}}$	↑	Increase in the volume fraction of bainite formed at low temperature (given, that no martensite was formed)

2.1.2 Case hardening

Diesel nozzle bodies must withstand an injection pressure of up to 2700 bar necessitating high strength of their surface. Case hardening can achieve such a requirement by initially carburizing or carbonitriding components (i.e., adding carbon and/or nitrogen at austenitizing temperature) and subsequently hardening them to form martensite. Typically, an industrialized heat treatment procedure of this type consists of case hardening with subsequent deep freezing (optional) and tempering (obligatory), which is why they are subsumed under the headline case hardening. Figure 2.4 delineates a time-temperature profile of the process at hand [100]. Next to easy machinability before heat treatment, this procedure allows to combine high surface strength with relatively high core strength. Especially for work pieces of geometrically adverse design (e.g., notches or bore intersections), this treatment enhances the locally endurable load. As a result, such work pieces can be subjected to higher stresses, particularly in the case of cyclic loads as often is the case in engines.

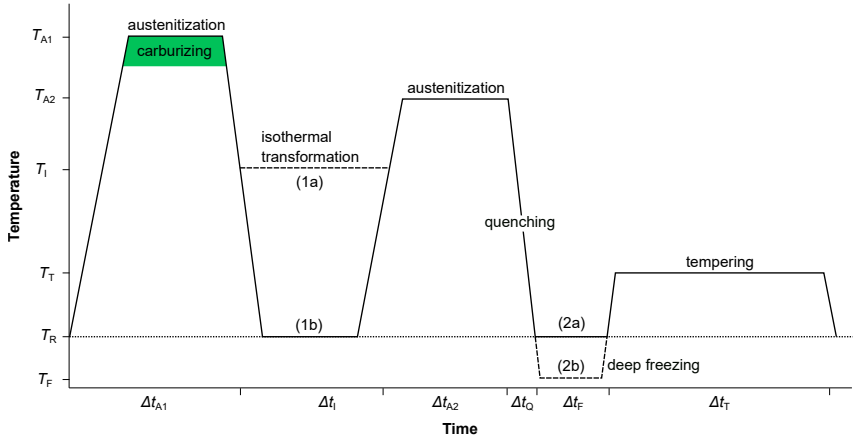


Figure 2.4: Schema of a time-temperature profile for case hardening with optional isothermal transformation and deep freezing followed by tempering [100]

Low pressure carburizing

To reduce the ingress of atmospheric oxygen, the process is carried out in a vacuum furnace well below atmospheric pressure, referred to as low pressure carburizing. Surface oxidation of components is most undesirable, as the resulting oxide formation leads to work piece failure. In the first step, after furnace evacuation, components are heated to austenitization temperature T_{A1}, which subsequently leads to phase transformation of the initial ferrite and pearlite into a fully austenitic microstructure. The austenitic microstructure allows a fast inclusion of carbon atoms into the work piece surfaces. During carburization, the components are offered a carbon donor (e.g., acetylene C_2H_2, exemplified by the green patch in Figure 2.4) to increase their hardenability and the maximum achievable hardness of the near-surface layer. The carbon enriched layer might have a thickness of only a couple of tenth up to several millimeters, depending on the requirements of further work piece processing and its later application. The sequence of pulses and pauses of the carburizing gas must be carefully engineered to achieve the desired degree of carburization, while

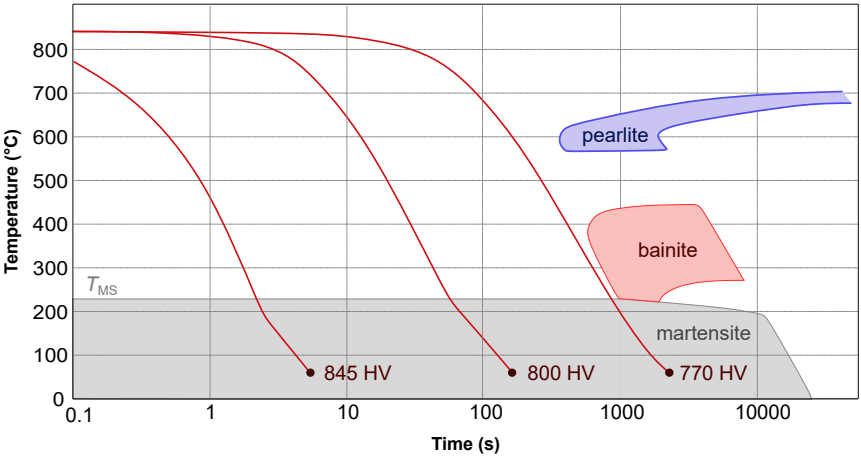


Figure 2.5: Schematic continuous time-temperature-transformation diagram, continuous martensitic phase transformation of 18CrNi8 after 15 min austenitization, carburized to C=0.56 % [121]

preventing too intensive carbide and soot formation. As carbides are considered critical for fatigue strength of the work piece, the soot, resulting from oversaturation of carbon in the atmosphere, might damage parts of the furnace, e.g., the heating system.

After carburization, components can either be brought to (1a) a temperature T_{I} at which isothermal transformation to pearlite occurs, or to (1b) the room temperature T_{R}. If (1b) involves a quenching, it is called direct quench hardening even if the temperature is lowered before quenching. A slow cooling to T_{R} with a subsequent further austenitization and quenching is called single quench-hardening. (1a) serves the grain refinement and leads to a more desirable martensitic structure after quenching. As indicated in Figure 2.5, the cooling rate during quenching must be high enough to reach the critical temperature T_{MS} required for martensite formation without the introduction of pearlitic or bainitic structures. If the desired amount of martensite was formed after quenching and cooling to

room temperature T_{R}, components can be tempered directly afterwards (2a). Otherwise, a sub-zero treatment is necessary (2b) [41].

Deep freezing

A certain amount of retained austenite is left in the carbon enriched surface-near layer which is dependent on the chemical composition after carburizing, as well as austenitization temperature, and duration. Further transformation from austenite to martensite after quenching only occurs at temperatures below T_{R}. Deep freezing components at T_{F} for a time Δt_{F} leads to the desired martensite-to-austenite ratio for a given application [6].

Tempering

Quenching and, if necessary, deep freezing result in a martensitic microstructure with extreme tension, which yields its ultimate hardness. Unfortunately, it is then also more susceptibility to cracking and fracturing. For this reason, the steel is tempered (i.e., brought to temperature T_{T} below 200 °C for Δt_T) in order to reduce the tensions and, thereby, gaining the required toughness at the expense of some hardness. The effect for varying tempering temperatures and duration is formalized by the Hollomon–Jaffe parameter (HP), given in Equation (2.1), where T_{T} is in Kelvin and Δt_T in hours. The constant C is dependent on the material used and often set to 20 for carbon-manganese and low-alloy steels. It is not critical in correlating the interdependence of tempering temperature and time [16, 64].

$$H_p = \frac{T_{\mathrm{T}}}{1000}(C + log(\Delta t_T)) \tag{2.1}$$

A sensitivity analysis of the heat treatment parameters on the resulting hardness is given in Table 2.2.

Table 2.2: Important process parameters for case hardening [79, 87]

Parameter (↑)	**Hardness**	**Influence**
Carburization	↑	Provided that carbon is in solid solution, increasing the carbon content up to 0.8 wt.-% leads to higher hardness as long as austenite is fully converted to martensite during quenching; if the martensite formation is not completed the resulting hardness might be reduced with increasing carbon content due to more retained austenite; for carbon contents higher than 0.8 wt.-% carbide formation has to be considered; for fully martensitic microstructure the hardness is increased further
T_A	↑	Increased amount of dissolved carbon in austenite leads to stronger lattice distortion and lesser carbides which were formed during carburizing
Δt_Q	↓	Less lattice distortion and possible formation of softer ferrite, pearlite, and bainite depending on the local carbon content
T_F	↓	Increased austenite to martensite ratio
T_T, Δt_T	↓	Increased carbon precipitations leads to relaxation of lattice distortion

2.1.3 Quality evaluation

For many heat treatment processes the objective is to improve and refine final material properties of a given workpiece. As properties like fatigue strength or wear resistance are seldom measured directly for quality assurance, the material property hardness is typically chosen as indirect criterion to evaluate a heat treatment's result [43].

The achievable degree of hardness depends mainly on a material's chemical composition. An alloy's hardenability is the "capacity of a steel to

give rise to martensitic and/or bainitic transformations" [100]. This capacity must be further differentiated in maximum achievable hardness and depth of hardening. While the first describes the maximal hardness that is achievable under optimal conditions moderated by the carbon content of the steel or alloy in question, the latter describes the hardness profile along the longitudinal section. The measurement of these properties is described below.

Hardness

Hardness measurements can be carried out on surfaces or on microsections of a workpiece with a reasonable effort and are widely used in industry. Since the hardness of case hardened steels declines with increasing distance from the surface, hardness measurements are used to characterize the penetration depth of the carbon enriched layer known as case hardening depth (CHD). That is, the distance from the surface at which the hardness falls below a defined value. It is determined by subsequent indentions perpendicular to the surface until the hardness in question is reached.

The precision of the measurement procedure[3] is limited by a number of factors. (1) The force with which the indenter is pressed into the specimen is allowed to deviate 1 % from the norm, (2) a single indenter can be used

[3] Under the assumption that hardness is proportional to the load necessary to produce a constant sized impression, Smith and Sandland developed the Vickers hardness measurement method [130] whereby a pyramid shaped indenter, usually a diamond, is pressed into the test specimen by a precisely controlled test force. This force is maintained for a specific dwell time, normally 10 to 15 seconds, and can range from some gram to several kilogram which is indicated as number behind the unit (e.g., HV 10 implies a force resulting from 10 kp with $1\,\mathrm{kp} = g_\mathrm{N} \cdot 1\,\mathrm{kg}$). As samples get harder, the test force must be increased for accurate measurement. The indenter is removed after completion of dwell time leaving a square shaped indent on the surface of the sample. As indentation pyramids are of a precisely defined shape, the Vickers hardness number can be derived as a function of the test load divided by the surface area of the indent. This area is determined by averaging the optical measurement of its diagonals.

Table 2.3: Left, maximum permissible span r_{rel} for n indents on hardness comparison plates > 250 HV. Right, example for 700 HV based on DIN EN ISO 46507-3 [101]

Number of inprints	5 10 15 20 25 r_{rel} **in %** for > 250 HV	5 10 15 20 25 e.g., r_{rel} **in HV** for 700 HV
HV 0.2 - $<$ HV 5	4.0 5.2 6.0 6.4 6.8	28.0 36.4 42.0 44.8 47.6
HV 5 - HV 100	2.0 2.6 3.0 3.2 3.4	14.0 18.2 21.0 22.4 23.8

over 30,000 times and is prone to abrasion. (3) The optical evaluation of the diagonals depends strongly on surface quality and incident light. According to DIN 6507 [101] repeated hardness measurements taken on a hardness comparison plate to evaluate equipment accuracy are allowed to scatter in a defined range r_{rel} depending on number and force of indents as well as measured hardness, shown in Table 2.3. For example, 25 indents of HV 10 on a 700 HV hardness comparison plate may lie in a range r_{rel} of 3.4 % which is equivalent to 23.8 HV. A complete analysis of reproducibility for hardness measurements is given in [65].

Surface carbon content

The carbon content in the near-surface layer of a component can be determined via glow-discharge optical emission spectroscopy (GDOES), a method for the quantitative analysis of metals and other non-metallic solids. Argon ions gradually ablate the layers of the metallic sample used as a cathode in a direct current plasma. Photons are emitted by blasted out atoms diffusing into the plasma. As the excited waves have characteristic wavelengths which are recorded by means of a downstream spectrometer, the number of atoms from each element can be quantified. Measurements are sensitive to ambient temperature (i.e., fluctuations of $\pm 0.1\,°C$ inside

the chamber lead to erroneous results) and exhibit a scatter of $\pm 0.02\,\mathrm{wt\,\%}$ for carbon [56].

2.2 Data Mining and Machine Learning

Aptly described by Maimon and Rokach, *Data Mining* is the process of gaining a valid, comprehensive, and novel understanding of data. This form of knowledge discovery recognizes patterns in the available information through automatic exploratory data analysis and inference statistics [91]. Figure 2.6 depicts the commonly used data mining framework from which the structure of this chapter is adopted. The first Section 2.2.1 elaborates on advanced visualization techniques and statistical methods for scientific knowledge discovery. Hereafter, the theoretical foundation is laid for the typical data modeling pipeline, broken down in Section 2.2.2 preprocessing of data, Section 2.2.3 selection of useful data parts, and the process of self-learning pattern recognition, referred to as 2.2.4 *Machine Learning* (ML)[4]. Section 2.2.5 *Evaluation* builds the capstone of the knowledge discovery endeavor. Although the framework in Figure 2.6 might be read in a linear fashion, the mining process usually needs many iterations, starting with smaller circles (e.g., exploring data first and reformulating the problem or collecting additional data) to larger circles (e.g., altering the process based on important selected features), whereby the interplay is indicated by the two-way arrows.

Industry 4.0 [63] in this work only plays a role insofar as the tools and methods used presuppose that the production step of heat treatment has already been digitized. Further digitization and communication along the value chain is desirable and may improve data collection and analysis, but is not the subject of this thesis.

[4] A comprehensive explanation of all models will be given in this chapter, although some models might be mentioned in earlier sections.

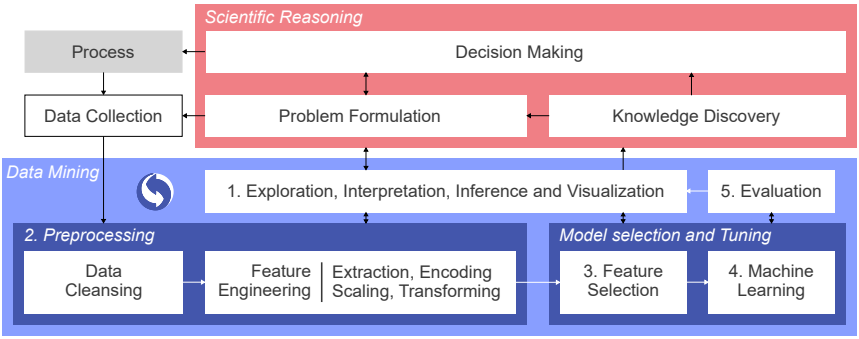


Figure 2.6: Design process for knowledge discovery using a data mining framework, based on [46, 93, 107]

2.2.1 Visual analytics and statistic

For the synthesis and representation of information, the literature suggests a multidisciplinary approach that draws on domain knowledge of the datas' origin (i.e., heat treatment and material science), analytical reasoning (i.e., statistical methods and models), and visual representations techniques (i.e., visual analytics [139]). These techniques seek to improve information transparency, accelerate analytic discourse, and rapid model evaluation, correction, and improvement [73]. To gain a comprehensive understanding of the data and be able to formulate new hypotheses, in a first step, the data sets are explored mostly by visual means to reveal simple statistical measures. After formulating a collection of hypotheses such as the difference between feature distributions or model performances, inference statistics may be used to test them.

Exploratory data analysis

Exploring data sets to understand distributions, chronological behavior, and generate hypotheses makes use of various statistical measures and visualization techniques. As our eyes are the only broadband connection

to our brain for the time being, graphical depictions can speed up transparency and, thereby, comprehension as described in [138]. In the following, common practice exploration techniques for time series, distributions, and relationships between variables are detailed.

Time series may usually be depicted pointwise over time. For noisy sets a *rolling window* (a.k.a moving average[5]) serves as smoothing function by calculating the mean of all values in the window ±d days of the actual date. For larger windows (i.e., larger d), longer trends become visible while shorter fluctuations are lost. Further, *autocorrelations* may exhibit seasonality by correlating time series with a shifted version of itself. If the correlation for a certain lag (e.g., minutes, days, month) is significantly higher than the others, this may indicate a repetitive or time-dependent nature. Lastly, Fourier transform, spectrograms, and wavelets may serve for the study of high periodic signals.

Distribution of data can be quantified by many mathematical parametric distribution types. However, numbers alone seldom give an intuitive sense of a data set's spread and many empirical distributions do not match one of the common types, which is why the following graphical facilitators are used [57].

Histograms represent the number of samples in chosen bins by height. Choice of bin width and start may lead to different or even misleading visual representations. Although smaller bins generally produce more truthful representations, the overall distribution might get lost.

Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function (PDF) used as an alternative to histograms to generate a smooth curve of what is likely to be the true distribution

[5] If the weights are the same and add up to 1, then the terms rolling window, moving average (MA), and finite impulse response (FIR) filter may be used interchangeably.

[109]. This is especially useful when plotting multiple distributions whose underlying values are physically continuous[6] (e.g., temperature) but are only measured discretely (e.g., [..., 2 °C, 3 °C, 4 °C, ...]). The above techniques work well for the individual examination of one or a small number of distributions, as variance and skew as well as multimodality (i.e., PDF has more than one local maximum or peak) are easily detectable.

Box and *Violin Plots* are used to compare distributions to each other where the first explicitly depicts distribution percentages while the latter is still able to detect multimodalities.

Relationships between variables most commonly are shown by *scatter plots* revealing correlations and, possibly, clusters of a third variable if color coded. Frequent combinations include (input, input), (input, output), and (output, prediction).

For large data sets *contour plots* are used[7]. They combine relationship and distribution by outlining the estimated 2-dimensional density function of the two variables, drawing contours at levels of same density and increasing darkness of shade with increased density.

(Clustered) Heat maps show the magnitude of a value in relation to the complete set that contains the tuples of two sets at x- and y-axis. These visual cues easily reveal how phenomena are clustered as well as extreme values. Often the correlation coefficient between features is plotted as heat map to reveal highly correlated features [148].

For the sake of completeness, it is mentioned that the above visualizations should not be regarded as a causal link. Even between highly correlated variables, if a thoroughly tested physical law cannot plausibly explain it. This is especially important when explaining features later. Neither should

[6] A consideration of quantum mechanics is not necessary here.

[7] Too many points in one scatter plot obscures the distribution of these point, because it is impossible to tell which points overlap and which do not.

a significant difference between values or scores be assumed if no confidence intervals are provided [3]. The following section introduces the methods for such claims.

Inference statistics

In order to test the multiple hypotheses that have been formulated during data exploration, statistical methods are used, that correct for the family-wise error[8]. Correcting this error is particularly important for machine learning, as many features and models are compared, easily leading to spurious correlations and claims of seemingly different scores that are in fact not justified. These tests allow to discard a certain null hypothesis H_0 (e.g., furnace 1 and 2 produce equal results), in favor of the alternative hypothesis, when the probability (p-value) of the observation's occurrence, under the assumption that H_0 is true, falls below a significance level α. The estimation of p can either be derived from a post-hoc test or the confidence intervals around the statistic in question.

Post-hoc multi comparison tests automatically integrate the correction for the family-wise error in their calculations. Depending on the test, the distributions to be compared must satisfy certain requirements like comparable sample size or homogeneity of variance. Two examples are introduced below.

Tukey's HSD test is an honestly significant difference (HSD) multiple comparison procedure that calculates all pairwise t-tests between the groups and corrects for the family-wise error rate by determining the significance level α through the studentized range distribution. It requires equal samples size and variance [141].

[8] That is, a correction for the increased probability of making one or more false discoveries (type I errors) when performing multiple hypotheses tests.

Scheffé test is also a multiple comparison procedure that is robust against imbalanced sets, as it uses all possible contrasts[9] among the factor level means, resulting in a lower test power.

Confidence intervals ($c\,\%$ CI) of a given statistical property, e.g., the mean μ, indicate that if an experiment was repeated sufficiently often and a $c\,\%$ CI calculated for μ in each trial, this CI would in $c\,\%$ of experiments include the true μ [24]. They may also be used as an alternative to some significance tests [112]. They allow transitioning from a null hypothesis testing framework, where only a dichotomous outcome (i.e., rejecting or keeping H_0) is possible, to a more quantifiable approach that might show how large the difference between populations is based on CI's of their means. When for example, the two 95 % CI do not overlap, then the significance level is $p < .01$ where sample sizes should be greater than 10, and the error margins do not differ by a factor of more than 2 [25]. Using sufficiently large CI amounts to a Bonferroni correction $p_i \leq \frac{\alpha}{m}$ for the family-wise error, where m is the number of hypotheses and p_i the corresponding p-values. By this means, box plots with confidence intervals may serve as a first impression of whether two distributions are significantly different, thereby partly integrating statistical inference methods in the visualization. The estimation of a CI assumes knowledge about the underlying distribution[10]. If this is either not the case or too expensive to test for, the below method may be used.

Bootstrapping is a computationally intensive statistical resampling method used as a non-parametric way of estimating CIs of statistical values in unknown distributions. It is beneficial for empirical distributions not belonging to the common parametric ones: Given that an empirical distribution contains n values, first, we resample n values from that

[9] A contrast is a linear combination of variables whose coefficients add up to zero.
[10] This is usually difficult to obtain for large ML-sets.

distribution with replacement, known as bootstrap sample. Second, we calculate the descriptive statistic (e.g., median, mean) we want a CI for. Third, we repeat 1. and 2. between 250 and 10.000 times, depending on the methods used for CI estimation and the narrowness of the CI [42]. Lastly, we use the collection of obtained values to estimate the CI. Besides the quickly calculable percentile method, there are more precise methods, including normal approximation, (accelerated) bias-corrected method, and the approximate bootstrap confidence method [60].

2.2.2 Preprocessing

According to a survey presented in Forbes, data scientists spend nearly 80 % of their time with data preparation [114]. The following sections dive into this preparation process, divided into a short cleansing section and a more extensive feature engineering part. Much of the success in machine learning is the successful engineering of features, which a model can readily understand. Moreover, groups of models react with varying sensitivity to the degree to which data has been transformed beforehand. While, for example, tree-based models are more robust to unscaled and irrelevant features, others, like neural networks or linear regressors, are not [80] (p.27).

Data cleansing

The usability of data is dependent on its quality and purpose of use. To enhance usability, factors that could compromise consistency, accuracy, timeliness, believability, and completeness need to be examined and, if possible, counteracted [57] (p. 84). Real-world data sets often come with a multitude of defects which makes data cleansing an indispensable preprocessing step (e.g., hardness measurements lack accuracy resulting in inconsistencies of repeated measurements, data might not be available in digital format, or lack a unique identifier). Moreover, missing values need

to be imputed, often done by replacing the gap with the mean or median of all samples. For time series, a search for data blocks with similar properties may be performed that are copied and then pasted into gaps [146]. Further steps include the deletion of outliers, correction of misspelled fields, and removal of duplicates. Many of the steps presuppose a certain amount of domain knowledge and their necessity is dependent on the rigorousness of the data collection process.

Feature engineering

The success of a machine learning algorithm depends mainly on what kind of data it is presented with. A feature is a numeric representation of information inherent in the raw data. Formulating beneficial features concerning the task, the data, and the model at hand, utilizing extraction, encoding, scaling, and transformation is called feature engineering which practitioners spend a majority of their time on[11]. Naturally, higher-quality inputs result in better, faster, and more easily trainable models [151]. Consequently, the first point in the data preparation checklist by Guyon and Elisseeff reads: "Do you have domain knowledge? If yes, construct a better set of 'ad hoc' features" [55], pointing out the field-specific nature of the engineering process, resulting in a relatively small amount of literature about the topic, as generalization across domains is difficult. Furthermore, ML methods differ in their ability to learn specific types of features (counting, differences, polynomial, etc.), which is why it is important, how the features are presented to an algorithm [62]. The upcoming paragraphs will describe commonly available data types derived from those, what kinds of features can be extracted, and which further transformations might be helpful.

[11] "At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily, the most important factor is the features used. Learning is easy if you have many independent features that each correlate well with the class." [33]

Data types of the raw data generally fall into categorical[12] and numerical[13] data and determine largely how and what kind of features are being engineered. Categorical data represents an unordered characteristic such as component type or machine number and needs to be converted into a numerical representation by encoding. Numeric types may be continuous or discrete and usually need to undergo scaling, which is potentially followed by further transformations. In rare cases binning is used to form groups from numeric data (e.g., age buckets $[10\,\mathrm{y} < 20\,\mathrm{y} < ...]$). Time series (i.e., numerical data with timestamps) usually exhibit a high degree of autocorrelation, resulting in a need for tailored feature extraction to compress the information in the many redundant data points.

Feature encoding of categorical variables is necessary as ML algorithms usually only accept numerical values. In case a category is already represented by a number, some algorithms (esp. for regression) assume that some order existed between categories (e.g., category 4 is higher or better than 3) [35]. If this is not the case or undesirable a *one-hot encoding* or *binary encoding* can be used. The first, a.k.a. dummy encoding, is a group of bits with value (0) among which only a single high bit (1) exists and where the length represents the number of groups. Its position in the group represents the category. Binary encoding assigns ascending integers to the categories starting from zero and converts them to their binary representation. This can be advisable for high-dimensional categories to avoid an explosion of the feature space (i.e., one new column per additional category). However, performance may suffer, which might be counteracted by feature hashing (i.e., using the binary value resulting from applying a hashing function to the feature categories) [123]. Table 2.4 shows the encoding of four production line numbers, in which case an

[12] a.k.a. nominal
[13] ordinal, interval, and ratio scales

Table 2.4: One-hot and binary encoding of different categories: Four production lines and days of the week. Arrays are used in their transposed form (vertical) in a feature vector later

category	line 1	line 2	line 3	line 4	Monday	Tuesday	...
one-hot	[1 0 0 0]	[0 1 0 0]	[0 0 1 0]	[0 0 0 1]	[1 0 ... 0]	[0 1 ... 0]	
binary	[0 0]	[0 1]	[1 0]	[1 1]	[0 0 0]	[0 0 1]	

input vector would be appended by four additional rows, each representing one line number.

Feature extraction from time series is essential as most ML algorithms prefer non-redundant already featurized inputs[14], with as few dimensions as possible for easier detection of relevant signals. Despite the rapid growth of *deep learning* algorithms which specialize in automatic extraction of specific patterns (e.g., CNN for pattern recognition in pictures or EEG data), the former statement generally holds because these algorithms need vast amounts of data to recognize the patterns that otherwise would be handed to an algorithm in already featurized form. Consequently, data should pass through noise reduction, resampling, and subsequent feature extraction for better results. This mainly applies to sensor signals, but occasionally also concerns a series of labels[15] explained in the following two paragraphs.

[14] ML algorithms that do not specialize in time series have difficulty using a complete (unprocessed) time series array as input because it contains too much redundant information. Values in the array are auto-correlated, which is true for almost all sensor signals.

[15] If a label's fluctuations or trends over time can not be explained by the features the past labels might be used as features. Turning these past labels into features often involves some form of averaging, that is, noise reduction.

Noise reduction with digital filters is done to clean a sensor signal and/or extract longterm fluctuations of a label by feeding the series $x[.]$ to a low-pass filter characterized by its impulse response, given in Equation (2.2). For finite impulse response (FIR) filters, this amounts to a weighted moving average[16] with all $a_k = 0$, which means FIR filters do not use their past output $y[.]$ as feedback, rendering them always stable. Infinite impulse response (IIR) filters additionally use their past output $y[.]$ as input[17], which usually allows for a smaller filter order M and N and more complex design, but also may render the filter unstable due to the feedback (for linear filters, sufficient stability criteria exist that are easy to check) [125]. The coefficients a_k and b_k are calculated during filter design which, for low-pass filters, includes a cutoff frequency ω_0 above which frequencies are attenuated.

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k] - \sum_{k=1}^{N} a_k \, y[n-k] \tag{2.2}$$

Feature extraction from signals is dependent on the application (e.g., temperature, velocity, or neurophysiological biosignals). It might be enough to extract basic statistics like mean, median, minimum, maximum, standard deviation, skew, and kurtosis. Other time series might need special segmentation, filtering, Fourier transformation, or extraction of permutation entropy to provide meaningful features to the algorithm. A comprehensive list along with the extraction process is provided by [7, 22]. In order to extract meaningful features while maintaining physical explainability, the use of domain knowledge can be crucial when deciding on which statistical values to extract from which part of a time series.

[16] This is similar to a rolling window, where every element is weighted individually.

[17] This feedback works as a memory. In this way, the filter remembers its current state and does not have to recalculate it from past inputs. It can thus be much shorter than a comparable FIR filter.

Feature scaling for numerical features changes the range of a data set to a defined interval, where the unit of a feature (e.g., meters vs. inch vs. light-years) may affect how a model makes use of it. The first reason is that floats[18] are more precise for small numbers. The second reason being the differences in the dimensions of the weights (i.e., internal parameters of some ML algorithms) learned by the algorithm, which results in over- or underemphasis of features and problems in learning, due to so-called exploding[19] gradients. Transforming the data to a standard range by normalization[20] helps to mitigate such issues, in particular for NN-like and clustering algorithms [57](p.114).

Min-max normalization is one of the simplest data transformations shifting the original distribution to an interval between [0, 1] or [-1, 1]. A formula for the interval [0, 1] is given in Equation (2.3) where x_{min} and x_{max} are the minimum and maximum of the vector $\mathbf{x}$ that holds the same feature of every sample [71]. The inclusion of extreme values makes it susceptible to outliers that are better handled by the next technique.

$$x_{min-max} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.3}$$

Robust normalization is similar to min-max normalization but uses the median instead of the mean to center the data and the range between the first q_{25} and third q_{75} quartile for normalization, in order to be more robust against outliers. Depending on the distribution of the to-be-normalized data, other quantiles may be used as well.

[18] Is a type of representation format of a number in a computer.
[19] Extremely large gradients may cause gradients to become unstable which may prevent convergence.
[20] Normalization and standardization may be used interchangeably in data processing.

$$x_{robust} = \frac{x - x_{median}}{x_{q75} - x_{q25}} \tag{2.4}$$

Z-score normalization, also known as standardization, is a transformation that maps the data to a distribution with a mean of zero and unit-variance using Equation (2.5), where $\bar{x}$ is the mean and σ the standard deviation of $\mathbf{x}$. It is widely used for NNs, even between layers, known as batch normalization [86].

$$x_{z-score} = \frac{x - \bar{x}}{\sigma} \tag{2.5}$$

Feature transformation arose from the need to transform inputs of linear regression models to be able to capture nonlinear relationships between in- and output. As the ability to learn specific nonlinear relationships varies between models, features may be transformed by the following Equations [62]: Logarithms and power functions (2.6), differences, and ratios (2.7), polynomials (2.8), counts and rational differences (2.9). Ideally, the modeler knows the physical relationship between in- and output.

$$y = \log(x), \qquad y = x^2 \tag{2.6}$$

$$y = x_1 - x_2, \qquad y = \frac{x_1}{x_2} \tag{2.7}$$

$$y = 2 + 3x + 4x^2, \qquad y = \frac{1}{2 + 3x + 4x^2} \tag{2.8}$$

$$y = \sum_{i=1}^{n} 1 \text{ if } x_i > t \text{ else } 0, \qquad y = \frac{x_1 - x_2}{x_3 - x_4} \tag{2.9}$$

2.2.3 Feature selection and predictor importance

This section deals with the general problem of finding a subset of features that preserves the necessary learning information while the joint entropy between the individual features is minimized (i.e., redundancy elimination). Too few informative features restrict the model's ability to perform its ultimate task. Too many or irrelevant features result in more expensive and tricky to train models [151]. Following Occam's razor (i.e., using the simplest explanation for a given hypothesis), predictive models usually exhibit less overfitting, work faster, and have more explanatory power when given only a useful (i.e., relevant and redundancy-free) subset of features [122]. While selecting only useful features may lead to the exclusion of redundant but relevant features, conversely, a selection of only relevant[21] features may be suboptimal for the modeling purpose due to the introduction of redundancies [55].

Three categories of subset selection methods may be distinguished, namely, *filters*, *wrappers*, and *embedded methods*, each with their own measure of usefulness and relevancy [69,122]. More elaborate techniques like wrappers and embedded methods may significantly improve predictor performance compared to simpler filters. Especially in domains with more voluminous feature sets, the curse of dimensionality needs to be counteracted at risk of overfitting. An automatic feature construction aided by domain knowledge can also improve performance and yields a more compact feature set [55]. Neither of the techniques can address the problem of causal inference between feature and label, which needs to be solved by the domain experts at some point in order to build reliable models.

[21] While there exist several mathematical definitions for relevancy, this work shall use the term in an intuitive sense, meaning having an impact on the target variable. For an extended discussion of relevancy, see [76].

Filters

A feature relevance score is determined only by intrinsic properties between feature and label. Features with low scores are then removed and the remaining subset is used for prediction. In contrast to other methods, filter techniques solve the feature subset selection problem independent of the ML algorithm, making computation much cheaper and allowing for easier scalability to high-dimensional data sets. The scores only have to be calculated once and any desired algorithm uses the chosen features. Unfortunately, features are only evaluated individually. Thus, the filters ignore feature dependencies, which may worsen prediction performance. Also, the interaction with a particular algorithm is not taken into account [122]. Three of the more frequently used filter methods are explained below; a comprehensive review can be found in [51].

F-ratio is generally used as a measure for the goodness of fit when comparing different models[22]. Equation (2.10) [49] describes the F-ratio, where R^2 is the coefficient of determination, k is the number of predictors, and N being the number of observations. The F-ration can also be thought of as the fraction between variance explained by the model and unexplained variance. A p-value can be calculated from F indicating whether a predictor significantly increases the prediction or not (i.e., has a high correlation with the target).

$$F = \frac{(N - k - 1)R^2}{k(1 - R^2)} \tag{2.10}$$

Mutual information (MI) or information gain (IG) measure the reduction in uncertainty (or the "information gained") for X by quantifying

[22] While the term adjusted R^2 is more commonly used for regression, the F-ratio is widely used in classification.

the information bits that can be obtained from observing Y. Equation (2.11) gives an approximation for the mutual information $I(X,Y)$, where the probability p for points falling into various bins[23] i, j is approximated[24] by the number of points in that bin. Thereby, it can capture nonlinear relationships between X and Y [78].

$$I(X,Y) \approx I_{\text{binned}}(X,Y) = \sum_{ij} p(i,j) \log \frac{p(i,j)}{p_x(i)p_y(j)} \tag{2.11}$$

(R)Relief originally was designed as a feature selection algorithm for a two-class problem by calculating the feature quality based on its ability to distinguish nearby instances of both classes. A randomly selected point searches for the nearest neighbors from the same class and a different class. Then, the so-called relief estimate W[A] is determined for all features and their weights are updated [75]. This procedure was adapted to regression problems by [120], hence the (R) in front of Relief.

Wrappers

Wrappers use ML algorithms as a scoring function with a selection procedure 'wrapped' around the model, that is, a subset of features is evaluated by training and testing a model. This way, interactions between model and feature are included in the search. Also, important dependencies between features can be considered, which comes with a higher risk of overfitting. The main drawback is the high computational cost of the procedure. The search for an optimal subset may be divided in two classes. 1) Sequential or deterministic selection, where features are stepwise added to or removed

[23] Supports of X and Y have been partitioned into bins of finite size beforehand.
[24] E.g., $p_x(i) \approx n_x(i)/N$, where $n_x(i)$ is the number of points in bin i of X and N the total number of points.

from a subset. This may be computationally infeasible for the exponentially growing number of subsets for high dimensions.
2) Heuristic or randomized searches can better traverse large feature spaces guided by an optimization procedure and might be less prone to local optima. [19, 122]

Sequential selection in its simplest form *sequential forward selection* (SFS) grows a set of features by starting with an empty set and then adding the feature that yields the best performance when added to the set. In every step, thus, all remaining features are evaluated with the current subset until prediction performance decreases or the required number of features is included. In this naive form, not all feature dependencies might be taken into account[25]. *Sequential backward selection* (SBS) already contains all dependencies since the algorithm starts with a set of all features and stepwise removes the feature whose exclusion leads to the smallest decrease in predictor performance which is a highly computationally intensive procedure. *Sequential floating forward selection* (SFFS) is more flexible than the naive SFS as it can add and remove features but might also overfit the data stronger and is distinctly more computationally intensive [19, 119].

Genetic algorithms (GA) belong to the heuristic methods and are a subtyp of evolutionary algorithms (EA). While the latter use real numbered values as encoding, GAs are limited to integer or even binary encoding. Both simulate the survival of population members (i.e., possible solutions to the optimization problem) over many generations based on their fitness (i.e., score of the evaluation metric to be optimized). Population members differ by their genetic code (DNA), which is an array with zeros and ones in its simplest form (i.e., GAs). If a member represents a

[25] Although SFS sometimes misses hidden relationships, making a very early selected feature redundant, it works much better than its reputation and is not particularly computationally expensive, especially for small feature numbers.

feature subset, then the subset of included features are marked with ones while all remaining features have a zero. The evolution steps are listed below:

1. **Creation of the initial population** is performed by initializing the DNA of each population member at random (e.g., an array with zeros and ones) with population size being a hyperparameter.
2. **Fitness evaluation** [26] uses a given evaluation metric (e.g., prediction error of ML algorithm) to determine the fitness score of each population member, where members with very low fitness may be eliminated or die in a so-called tournament with other members. In addition, a small portion of the best individuals from the last generation (without changes) can also be transferred to the next generation, called elitism selection. Other methods include the roulette wheel, rank, and steady-state selection.
3. **Cross-over** occurs between pairs of the current population drawn at random, with fitter members having a higher probability of being chosen proportional to their rank. During mating of each pair, two new offspring emerge by crossing over the DNA of their parents, that is exchanging sequences of their arrays at randomly selected segments.
4. **Mutation** flips bits of the DNA at random. Although this occurs with rather low probability, some improvements may be obtained which can lead out of a dead end like a local optimum. The mutated offspring then forms a new population generation.

Steps 2, 3, and 4 are repeated until some termination criterion is reached (e.g., no fitness improvement, a maximal number of generations, or calculation time). These steps are at the heart of every GA, but implementations may vary between applications [84].

[26] Sometimes is also listed as 4^{th} step.

Embedded methods

Embedded methods integrate the feature selection into the training process, making them work more efficiently than wrappers. By optimizing a two-part objective function consisting of a prediction metric and a penalty for large amounts of features (i.e., avoiding the usage of too many irrelevant variables), they may reach a solution faster as the need for retraining all subset disappears. Examples of such algorithms are decision trees and ensembles thereof (e.g., random forest), as well as Lasso and Ridge regression using $l1$ and $l2$ regularization for their weights, respectively [19, 55, 122].

2.2.4 Machine learning algorithms

Machine Learning (ML) refers to the development and application of algorithms or statistical models with the ability to automatically improve their internal parameters based on experience (i.e., data) [94], in order to perform a specific task (e.g., make predictions or decisions) effectively without using explicit instructions, but relying on patterns and inference instead [10]. Depending on the nature of the task, different categories can be distinguished, the most common being (un)-supervised and reinforcement learning [150], see Figure 2.7:

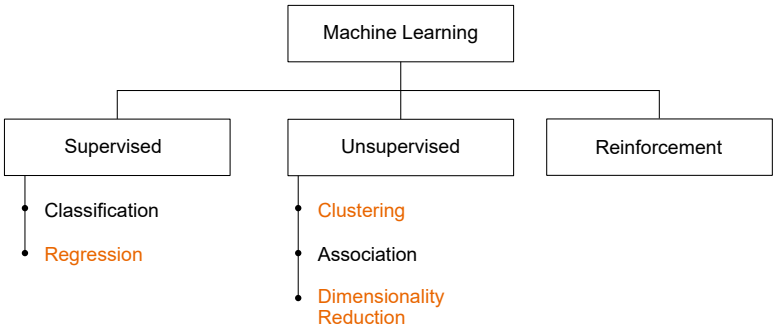


Figure 2.7: Types of machine learning

Supervised learning refers to automated decision-making by generalizing from examples (i.e., ability to make correct predictions for unseen data by independently learning from a training set) [96]. The algorithm learns to map inputs to desired outputs and generate accurate predictions [80]. The problem is termed classification if the predicted label is a categorical value (e.g., pictures of a microstructure characterized as good, neutral, or bad), whereas regression is the prediction of continuous values (e.g., hardness measurement, CHD).

Unsupervised learning is applied to data sets without pre-existing labels in order to automatically cluster the data (i.e., divide by similarity), reduce its dimensionality with minimal information loss, find associations (i.e., identify sequences), or detect outlier by recognizing previously unseen patterns [124].

Reinforcement learning happens during the interaction of an algorithm, called agent, with an environment that can be modified through inputs, called actions. The environment provides the agent with some state variables during the interaction as well as a reward at the end of a session. The agent tries to maximize this reward over the course of many sessions by optimizing its actions based on the provided state variables (e.g., finding the optimal process parameters to gain a desired hardness, given that a simulation of the process exists).

Supervised regression models

Many algorithms now subsumed under the term machine learning were introduced in the last century, when the available computing power did not allow mass deployment on large data sets. Today, research is progressing rapidly, designing new architectures and introducing modifications to the body of algorithms. Almost all of which are available in a version for

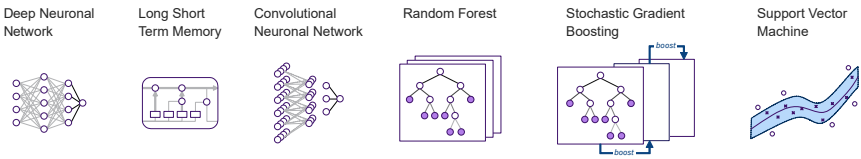


Figure 2.8: Machine learning algorithms for supervised regression, [own representation]

classification and regression. As the focus of this study lies on continuous labels, explanatory emphasis is on regression. The most popular ML methods shown in Figure 2.8 are explained below.

Linear models

Linear regression (LR) optimizes the weights w of the linear function $y = w_0 + \sum_{i=1}^{N} w_i x_i + \varepsilon$, where y is the label, x_i the features, N the number of features, and ε a random error. If the optimization of the weights includes a penalty for large weights, we talk about **Lasso regression** ($l1$ regularization) and **Ridge regression** ($l2$ regularization) [97].

Sparse kernel methods

Support vector machine (SVM) finds a (hyper-) plane that maximizes the gap between two classes in feature space. The features are often transformed into a higher dimension via a kernel to account for nonlinear relationships. Since the points closest to the dividing plane are the most relevant, they are called support vectors, hence the name. Its correspondence for regression problems is the *support vector regressor* (SVR) developed by [36].

Neural networks

(Deep) neural network ((D)NN) uses multiple layers of nodes[27], often fully[28] connected by weights, which are being optimized by backpropagation. The prefixed term *deep* implies that more than a certain amount[29] of hidden layers of nodes is used. More layers allow for learning more complex relationships but generally need more data and are more prone to overfitting.

Convolutional neural network (CNN) is a class of deep neural networks that is most commonly used to classify images. Due to its 2D input structure that allows for extraction of features based on spatial coherence (e.g., edges and patches), it is particularly advantageous. The prefixed term *convolution* refers to the mathematical operation by the same name that works as a filter on the image to detect patterns. CNNs, thus, have the ability to learn filters that extract features' characteristics for an image class and massively reduce the number of free parameters due to the coupling structure compared to a DNN with only fully connected layers [85].

[27] In the standard multilayer perceptron, every single node, also referred to as neuron, contains a function similar to LR plus an output function (mostly sigmoid or ReLu). Other functions (e.g., radial basis function) exist but are rarely used.

[28] Each node of one layer is connected with each node of the following layer. However, there exist many different architectures optimized for a particular use case with fewer connections (e.g., CNNs).

[29] There is an ongoing discussion about how many hidden layers are necessary to warrant the term *deep* learning.

Recurrent neural network (RNN) is, in contrast to the feed forward[30] NNs mentioned above, equipped with feedback connections. It is, therefore, especially good at processing sequences of data (e.g., speech recognition or text). Two advanced architectures that work with this technique are particularly prominent; namely the *gated recurrent units* (GRU) [20] and the *long short-term memory* (LSTM) [54], which make use of gates that decide which inputs and outputs to keep and forget.

Ensembles of regression trees

Random forest (RF) constructs a multitude of uncorrelated decision trees (so-called weak learners) at training time, outputting the mean prediction of the individual trees. The term *random* refers to the element of chance involved in making decisions when creating the trees. Random forests better correct for decision trees' tendency of overfitting to their training set [14].

Gradient boosting (GB) iteratively constructs the model in a stage-wise fashion from an ensemble of weak prediction models, typically decision trees being trained and pruned on examples that have been filtered by previously trained trees. It generalizes them by allowing optimization of an arbitrary differentiable loss function [37].

Instance-based learning

K-Nearest neighbors (KNN) finds the k samples that are closest in feature space to the sample that is being predicted. The prediction is then

[30] "Normal" NNs and CNNs are feed forwards because information is always processed in one direction (i.e., the direction of output) and is never stored or fed back to the same or previous neurons or nodes.

a weighted average of the labels of these k samples. As it memorizes every instance of the training set, computation time increases drastically with high-dimensional data sets.

Unsupervised clustering and dimensionality reduction

Cluster analysis finds substructures in unexplored data by assigning data points to clusters in such a way that items (usually consisting of multiple data points) attributed to the same cluster are as similar as possible, while other items are most dissimilar. Similarity measures include distance, connectivity, and intensity.

Fuzzy c-means (FCM) introduced by Bezdek in 1981 [8] is an extension of k-means. Unlike its predecessor, one data point can belong to multiple clusters during the clustering process as each point possesses weights that indicate the affiliation to each cluster. First, a predefined number of random cluster centers (centroids) are determined, then, each point is assigned to the closest centroids. Third, the squared distances between centroids and belonging points are calculated and summed for each cluster. Lastly, the centroids are moved to minimize these sums. Steps 2, 3, and 4 are repeated until a stable minimum of sums is reached. This method has the following advantage: the initial set of random centroids does not influence the final clusters as much as k-means. In addition, the creation of a noise cluster allows the detection of outliers that are not close enough to one of the other clusters [104].

Principal Component Analysis (PCA) approximates a set of (statistical) variables by finding their most relevant linear combinations (i.e., principal components) [70].

(Variational) autoencoder (VAE) is a form of nonlinear PCA that uses a NN architecture to learn a compressed representation of a data set to extract the relevant features. The encoder is a shrinking NN to the dimension z, while the decoder is its mirror image that tries to reproduce the original input from the information given in z [77]. The variational autoencoder forces the distribution of z to be Gaussian by adding an approximation of the Kullback-Leibler-Divergence between z and a standard normal distribution to the loss function of the network.

2.2.5 Model evaluation

Two types of evaluation can be distinguished, which may use the same metric, but not necessarily do so. The first concerns the optimization during each training pass and is calculated by an algorithm's internal loss function. It often is continuously differentiable and the basis for parameters' adjustment to make better predictions. The second determines the goodness-of-fit between measured and predicted values, particularly for the test set.

Evaluation metrics

While ML packages usually come with a fixed set of loss functions that can not readily be altered, as the internal optimization depends on it, several different final scores can easily be calculated. These scores are coupled with the prediction task they are seeking to evaluate. The most prominent metrics for regression and classification are explained in the following.

Regression metrics

Mean squared error (MSE) between labels y_i and corresponding predictions $\hat{y}_i$, given in Equation (2.12), is used as loss function for most

algorithms and often is the evaluation criteria for adjustment of the learning parameters [80]. The root mean squared error (RMSE) might be more readily interpreted since it has the same dimensionality as the label it is calculated from (e.g., if the original unit of measure of the label is Vickers, then the RMSE is also in Vickers). However, the RMSE does not lend itself to universal comparability between models because knowledge about the label and its distribution is necessary to determine the actual predictive power. It is therefore often compared to a dummy regression, a form of intelligent guessing. The RMSE can be minimized by using the mean of the labels of the training data $\bar{y}$ as a prediction for all samples. The resulting RMSE_{bl} in Equation (2.14) gives a baseline on what an algorithm has to achieve at least to make valuable predictions.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.12}$$

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{2.13}$$

$$\text{RMSE}_{bl} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.14}$$

Coefficient of determination (R^2) can be derived from the MSE and gives a comparison between the actual and baseline performance of the model. Negative values indicate below baseline results, that is worse than guessing $\bar{y}$, while values close to 1 attest minimal error between observed and predicted outcomes. It can also be interpreted as the proportion of the

variance in the target explained by the features. While different computational definitions exist for the R^2 score, the one given below is preferable as it generalizes well to problems outside linear regression.

$$R^2 = 1 - \frac{\text{MSE}_{model}}{\text{MSE}_{bl}} \tag{2.15}$$

Classification metrics

Confusion matrix of a two-class prediction evaluation shows the four decisive values: true positives/negatives (TP)/(TN) (number of correctly classified) and false positives/negatives (FP)/(FN) (number of misclassified) labels, in a two-by-two matrix. The most common ratios between these values are accuracy, precision, true positive rate (TPR), and false positive rate (FPR). Accuracy measures the percentage of correctly classified labels $\frac{TP+TN}{N}$, where N is the sample size, while precision $\frac{TP}{TP+FP}$ shows what percentage of the labels predicted as class 1 (i.e., positive class) were correct. The TPR $\frac{TP}{TP+FN}$, also called sensitivity or recall, conversely measures what percentage of the true class 1 labels was actually found. In ML, it is sometimes called the probability of detection. The FPR $\frac{FP}{FP+TN}$, also known as the probability of false alarm or fall-out rate, indicates what percentage of class 2 labels was wrongly classified as belonging to class 1 [45].

Receiver operating characteristic can be used to find an optimal threshold to decide whether a prediction belongs to class 1 (star) or class 2 (circle), see Figure 2.9. Regression problems can be converted to classification problems using a class separator that assigns the true values to either class circle ($<$ separator) or class star. In order to determine the classifier's discriminatory ability, a ROC curve plots the TPR against the FPR at various threshold settings. By variation of the threshold, a suitable trade-off between sensitivity and specificity can be made for the

problem at hand (e.g., more emphasis on finding all positives requires a higher sensitivity while more emphasis on avoiding FP requires a better specificity) [45].

Area under the curve (AUC) is calculated from a ROC and gives an indication of how well the classifier performs, where an AUC ≈ 0.5 is equal to guessing while an AUC close to 1.0 indicates optimal discriminatory ability.

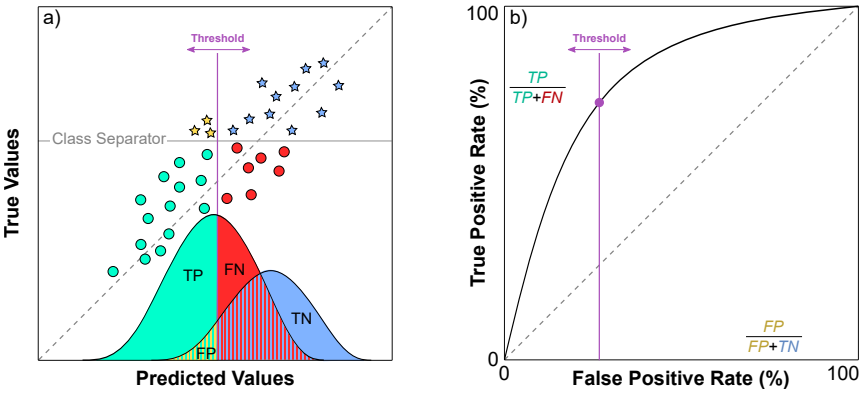


Figure 2.9: a) Scatter plot of true and predicted values with a classification in circles (< class separator) and stars b) ROC curve showing the trade-off between true positive (TPR) and false positive rate (FPR), based on [80]

Resampling methods

The risk of overfitting rises along with nowadays more complex models that have the ability to learn a given input-to-label mapping virtually by heart, such that labels can be predicted close to perfection. To avoid a model's loss of generalizability, the training and subsequent evaluation must be performed in such a fashion that overfitting can be detected, at best during training but at the latest during final evaluation. A number of

methods are available to split data, as well as train and evaluate a model to approximate its generalization ability. Often, the choice depends on the number of samples available, their distribution, and how the modeler wants to 'spend' these data points [80, 95].

Train-dev-test split is the most commonly used and least expensive split in training, development (a.k.a. validation), and test (a.k.a. evaluation) set by a ratio of approximately 7:1:2, respectively. After (and in some cases during) training with the training set, the development set is used to detect overfitting. With these two sets, different models are built and tuned. Only in the last stage, when it comes to model selection, the test set is evaluated to get an unbiased view of the models' predictions. Unfortunately, the available amount of data not always allows spending 30 % of the data purely on the final evaluation. Further, sets might contain disproportionate amounts of easy or difficult to learn samples, such that different splits certainly lead to different results.

(Stratified) k-fold cross-validation (CV) splits the data (randomly) into k equally sized sets. The model is then trained k times on $k-1$ of the sets, while the remaining set (called held-out set) is used for evaluation. This way k development sets can be used for evaluation, with typical numbers of k between 3 and 10. In extreme cases k takes the number of samples called *leave-one-out cross-validation* (LOOCV). *Repeated* CV performs the k-fold CV process several times to get a more truthful distribution of the evaluation score. *Stratified* splitting is a non-random form of partitioning such that each set 'strata' approximately contains the same distribution of labels and inputs to make them more comparable.

Monte Carlo Simulation in general performs repeated evaluation of a function, process, or experiment with parameters randomly drawn from a given distribution. It can be used for model evaluation by randomly

splitting the data into training and evaluation set numerous times (e.g., 10 to 100 iterations) and then summarizing the evaluation scores, which amounts to repeated k-fold CV for a large number of iterations.

Bootstrapping already discussed in Section 2.2.1 may also be used for model evaluation by repeatedly drawing samples with replacement[31] to build the training set and subsequent evaluation with the out-of-bag samples.

Hyperparameter optimization

The number of hyperparameters (e.g., learning rate, number of layers in a NN, or splits in an RF) may vary greatly between model families. Consequently, model tuning becomes an optimization problem of its own if the hyperparameter space is high dimensional.

Grid search exhaustively searches through all possible combinations of hyperparameter settings in a grid. It may provide the most comprehensive overview of the effects of a set of hyperparameters but is only feasible for very small dimensions because it evaluates the objective function[32] at every point in the grid, instead of directly optimizing for convergence to a minimum.

Random search evaluates the objective function at randomly chosen points in the hyperparameter grid. This way, much fewer evaluations need to be performed while still coming close to an optimum.

[31] Some samples may be represented multiple times in the training set.
[32] Used to calculate the model performance for a particular set of hyperparameters.

Genetic algorithms as described in Section 2.2.3 take randomly parameterized models as generation zero and evolve better models by crossover and mutation between well performing models [83, 106]

Factors affecting model performance

Typically, a data scientist works backward through the ML pipeline to find factors that might worsen predictability when model performance does not live up to the expectancy. Starting from optimization of hyperparameter and identification of ineffective, erroneous, or missing features, over the improvement of feature scaling and transformation to counteracting a potentially large class imbalance [23]. On top of these factors, discretization of continuous values (e.g., resolution of the sensors) and measurement noise in either the predictors or the label (e.g., the influence of background lighting when measuring a hardness imprint) put strict boundaries on the achievable predictability [80]. Lastly, a "Type III" error may occur, that is, answering the wrong question. For example, the ultimate test for a cylinder head is the amount of pulsed pressure it can withstand. A hardness measurement may give some indication of how sturdy the component is, but it does not answer the question of longevity in the field [74].

2.3 Application of supervised ML to Heat Treatment and Material Science

As the wave of enthusiasm for machine learning sweeps through industry and research, material science and heat treatment communities are evolving to explore the predictive capabilities of these methods. Some work published so far is listed in Table 2.5 providing a detailed non-exhaustive overview regarding the application of ML methods to heat treatment, sorted primarily by target and type of treatment. Correspondingly, this section is split into the prediction of mechanical and component properties.

2.3.1 Prediction of mechanical properties

Unsurprisingly, the chemical composition is the number-one predictor used in many models, followed by a variety of heat treatment parameters. They were used to predict austenite, bainite, and martensite start temperature [5,53,115], the volume fraction of bainite in low carbon steels [127], or the bainite plate thickness [126]. The latter also using Gibb's free energy, austenite strength, and carbon concentration as features. The hardness of bainitically hardened steels was investigated by [113, 128] on the basis of mass fractions of the alloying elements as well as heat treatment parameters, confirming the high relevance of manganese for bainite kinetics also found by Bhadeshia [9].

Because hardness depends considerably on chemical composition, it was also featured in an early model of Vermeulen to predict complete Jominy hardness profiles [142], to optimize for a desired hardness while changing input parameters [140] and to predict hardness and impact strength [44]. While the former two also made use of heat treatment parameters, the latter included tensile properties. Hardness was also predicted for the low-alloy steel C45E based on electromagnetic hysteresis loops, fed to particle swarm optimization, [152] and based on tempering time and duration [135].

Chemical composition and heat treatment parameters also predict mechanical properties of steels in different applications such as tensile strength, impact toughness, or hot-ductility and -strength as shown by Sterjovski et al. [132]. Focusing on prediction of tensile strength of martensitic hardened low-alloy Cr-Mo steels [137] noticed strong influence of confounded data (e.g., outliers), and introduced a data cleansing strategy to improve prediction accuracy. A model with similar in- and outputs was used by Reddy [117,118] in a genetic algorithm (GA) to design optimal composition of medium carbon steels.

Images pose a special challenge because input features must first be extracted from detectable patterns. DeCost et al., therefore, applied CNNs

to segment and identify microstructures and used an SVM to predict the annealing schedule that led to these microstructures [28, 29].

In order to predict the hardness of laser-hardened samples, temperature curves were determined with the aid of thermal models from which effective carbon diffusion times and cooling times were derived [105]. In a similar approach, Lambiase [82] used a priori calculated time-temperature profile by aid of the one-dimensional heat conduction equation as input.

Surface hardening by induction was also investigated with NNs. Stich used motor speed and component temperature to predict the maximum hardness increase [133], while Nguyen provided a model as input for prediction of quality and process control [98].

NNs were also applied to welding, rolling, and squeeze casting, respectively, to predict the Charpy toughness from chemical composition and interpass temperature [108], ultimate tensile strength (UTS) and yield strength (YS) from chemical composition and rolling parameters [129] as well as hardness, impact energy (IE), and UTS from melt and die temperature [2].

2.3.2 Prediction of component properties

Publications are likewise available for the prediction of component behavior. ML methods were most commonly used for prediction of crack propagation [27, 32, 48, 58] and lifetime under cyclic loading [4, 52, 68, 72, 131].

Fatigue crack propagation behavior has been predicted mainly from stress intensity. Sample size ranged from $N = 12807$ for Ni-Ti-Al alloys [32], over 60 [58] down to 8 samples of cast iron [27]. Prediction of crack length in stainless steels based on chemical composition and welding parameters was made by [48].

Research has also been done on the prediction of various lifetime parameters under cyclic loading. Based on 30,000 simulated parts of SAPH 440

multi-axially stressed bodies, the most critical parameters were identified by analysis of the dynamic behavior and then fed to a NN to predict the fatigue life. 5 % of the generated training data were sufficient for optimal training outcome [72]. Creep rupture strength of X10CrMoVNb9 was predicted based on chemical composition and various material properties after heat treatment [52]. Using only 5 samples, Solon-Alverez tried to predict the rolling contact fatigue [131], while Jin was more successful with 110 samples and an input of chemical composition, heat treatment, and contact stress [68].

An approach to predict Wöhler lines for a number of low-alloy steels was presented by Artymiak [4]. This was carried out based on data of steel life predictions in the low cycle fatigue (LCF) and high cycle fatigue (HCF) range, whereby only mechanical material and stress parameters, as well as the type of load and the notch factor in fatigue, were included as input variables. The results were compared with synthetic Wöhler lines, which were calculated with the help of literature references. The results prove the fundamental applicability of NNs to material fatigue applications. However, the linking of heat treatment parameters to resulting service properties has not yet been carried out by Artymiak.

Without reference to any heat treatment procedure, a prediction of the probability density functions for the fatigue life of the case hardening steel 20NiCrMo2 was made on basis of the applied step stress conditions and mechanical properties [50]. Up to 20,000 simulated samples were used to predict the damage from fatigue loading parameters [38]. Finally, the prediction of tribological performance of highly alloyed steel based on material and applied pressure should be mentioned [17].

Samples		Steel	Treatment	Input	Model	Target	Ref
lit	788	-	austenite	heating rate, chem. comp.	Gaussian Process	$T_{A,onset}$ (530-921 °C) $T_{A,complete}$ (650-1060 °C)	[5]
lit	2277	-	martensite	chem. comp.	NN,RF, GB,AdB,..	T_{MS} (200-800 °C)	[115]
lit	247	-	bainite	chem. comp.	NN	T_{BS} (250-700 °C)	[53]
lit	300	-	bainit	Gibbs free energy, T_I, C%, strength austenite	NN	plate thickness (25-330 mm)	[126]
lit	437	-	bainite	chem. comp.	NN	volume fraction (0-0.6)	[127]
lit	220	-	bainite	chem. comp., T_A,T_I,Δt_I	NN	hardness (315-760 HV)	[128]
exp	96	-	bainite	Mo%, Cu%, Mn%, Ni% T_I,Δt_I	NN	hardness (370-520 HV 10)	[113]
db	4000	-	hardened	chem. comp., T_A	NN	hardness (Jominy) (20-65 HRC)	[142]
lit	3532	-	hardened	chem. comp., T_A, cooling rate	NN	hardness (200-650 HV)	[140]
exp	104	-	hardened	chem. comp., YS, UTS, El	NN	hardness (192-224 HV 10) IE (222-353 J)	[44]
exp	5	1.1191 C45E4	surface-hardened	hysteresis loops	NN, LR	hardness (200-700 HV)	[152]
exp	33	1.1191 C45E	hardened, tempered	T_T, Δt_T	NN	hardness drop (2-26 HRC)	[135]
sim	75, 86, 221	-	hardened, welded, casted	chem. comp., treatment parameters	NN	hardness (160-500 HV), TS_{hot} (40-170 MPa), IE (10-170 J), ROA (9-90 %)	[132]
lit, db	600	-	hardened	chem. comp., heat treatment parameters	NN + GA	UTS (600-1600 MPa), $Stress_{Proof}$ (400-1400 MPa), ROA (25-75%),IE(25-150 J),El(10-30)	[137]
lit	140	low alloy, medium C	hardened	chem. comp., Mn/S ratio, cooling rate, T_T)	NN + GA	UTS (700-1300 MPa, YS), (550-1200 MPa), El(13- 30 %),ROA (30-65 %), IE (15-95 J)	[117, 118]
lit	24, 24	ultra high C	hardened	image	CNN	segment in image (of microstructure, particle segmentation)	[29]
lit	cr(600),ultra cr(195)	high C	hardened	images micrographs (cr(600) = 2400, cr(195) = 80), cooling method	CNN+ SVM	microstructures (prealite, bainite,. . .), $T_{annealing}$, $\Delta t_{annealing}$	[28]
	cr(3) = 213	1.2344 X40CrMoV5	laser	cross-sectional temperature distribution by 3-D thermal simulation	CNN+ cGAN	hardness distribution	[105]
exp	4 i(20)	K340	laser	T_{max}, cooling time	NN	hardness (220-700 HV)	[82]
	45	-	induction	motor speed, part temperature	NN	hardness (87-90 HR15N)	[133]
sim	1200	AH32	induction	model	NN	induction line, deformation	[98]
db	5973	-	welded	chem. comp., T,Δt	NN	Charpy toughness (0-356 J)	[108]

Samples	Steel	Treatment	Input	Model	Target	Ref
lit, 1892 db	-	rolling	chem. comp., rolling parameters	NN	UTS (420-650 MPa) YS (250-550 MPa)	[129]
exp 8	Al-alloy 2219	squeeze casting	T_{melt}, T_{die}	KNN+ GWO	hardness (78-94HB), IE (3-5.7 J), UTS (60-250 MPa)	[2]
12807	Ni-Ti-Al alloys	-	mechanical properties, test-specimen characteristics, stress-intensity range and test-frequency	NN	fatigue crack growth rate ($10^{-8} - 10^{-2}$ mm/cycle)	[32]
60	-	-	stress intensity ranges (DK)	NN	fatigue crack growth rate ($10^{-7} - 10^{-1}$ m/cycle)	[58]
exp 8	cast iron	-	ΔK, stress intensity factor amplitude	NN	fatigue crack propagation ($10^{-10} - 10^{-6}$ m/cycle) over (3-50 MPa$\sqrt{m}$)	[27]
lit, 487 db	-	-	chem. comp., samples thickness, welding parameters	NN, SVM	total crack length (0-20 mm)	[48]
sim 29951	SAPH 440	defor-mation	material properties, fatigue life parameters	NN	fatigue life (10^2-10^{15} blocks)	[72]
db 1396	1.4903 X10CrMoVNb9	-	chem. comp., UTS, Stress,El, ROA, heat treat. temps./dur.	NN	$\Delta t_{Rupture}$ (10^2-10^4h) creep rupture strength (60-110 MPa)	[52]
exp 5	hypereutectoid pearlitic		-	NN	rolling contact fatigue $6 \cdot 10^4$-$18 \cdot 10^4$ cycles	[131]
exp 110	chromium alloyed	hardened, tempered	chem. comp., heat treatment parameters, contact stress	NN	contact fatigue life 0-$3 \cdot 10^6$ cycles	[68]
1000	(cast) steel	-	UTS, YA; notch factor, surface roughness, type of loading	NN	S-N curve: stress-amplitude (180-420 MPa) over (10^3-10^6 cycles)	[4]
232	1.6523 21NiCrMo2	step-stress	specimen characteristics, stress-intensity, UTS, YS, El, ROA, breaking strength, ...	NN	probability density for stress	[50]
sim 10^2-$2 \cdot 10^4$		-	fatigue loading (material properties, spectral characteristics)	NN	damage	[38]
exp 216	highly alloyed	nitro-carburizing	material, bulk hardness, rotational speed, applied pressure	NN	tribological performance	[17]

Table 2.5: Meta analysis of ML applied to material science and heat treatment. The following abbreviations are used to enhance readability.
Samples: (lit) literature, (db) database, (exp) experiment, (sim) simulated
Input & Target: (chem. comp.) chemical composition, (UTS) ultimate tensile strength, (YS) yield strength, (ROA) reduction of area, (EL) elongation, (IE) impact energy/strength)
Model: (NN) neural network, (RF) random forests, (GB) gradient boosting, (adB) ada boost, (LR) linear regression, (KNN) K-nearest neighbor, (cGAN) conditional generative adversarial network, (SVM) support vector machine, (GA) genetic algorithms, (GWO) Grey wolf optimization

2.4 Open Questions

There has been rapid progress in material science and heat treatment concerning the use of NNs. Bhadeshia claims, however, that it may be rather hasty to assert that the method is established. A number of serious drawbacks result from the often incomplete research and publication of models [9]. While the literature already covers a wide variety of input-output prediction pairs as far as heat treatment and material science are concerned, there is an unexplored territory regarding the application of differing ML methods to large real-world data sets focusing on high accuracy in a small parameter range.

- Neural Networks as one of the most prominent ML methods are able to make use of highly nonlinear relationships and are used heavily throughout literature and industry due to their performance capabilities, flexibility, and popularity. However, a vast body of ML methods like random forests or gradient boosting trees seems to be fairly underrepresented, which leaves a big gap as to which models are best suited for a particular prediction problem in heat treatment.
- As data is at the heart of every ML prediction, an adequate number of samples is necessary to sufficiently cover the many nonlinear maps between input and output space. However, many of the presented papers work with limited resources of only a few hundred samples leaving the question open of whether (1) the actual nonlinear relationships could be captured or (2) the problem at hand did not contain a mapping that would have needed a NN.
- The reported prediction accuracy for most of the work was astonishingly high. Classically, this can be achieved by expanding the input

and output space until the relationship can easily be detected. Indeed, a relatively wide range of measurements regarding the target variables can be observed quite frequently. There appears to be a lack of attempted predictions for much narrower ranges with much less variability in the input, raising whether a prediction accuracy that suffices industry standard is achievable and what amount and kind of input data would be necessary.

- Predictions from real-world applications are inherently more complex than literature-based models, as many more influences and mistakes happen between data generation and model ingestion. Moreover, the effort of obtaining industry data is more costly and, yet, its acquisition can lead to valuable insights into hidden relationships and dependencies. Given that industrial hardening processes often contain company secrets, their data is seldom available to public, academic research leaving the question of how and what can be learned from these kinds of data sets.

- Machine learning algorithms mostly work well with a fixed set of uncorrelated input features, which, fortunately, are often contained in literature and databases (e.g., chemical composition). Complete time series of processes and necessary feature extraction are performed very seldom. Thus, it is not known yet which feature extraction process concerning heat treatment yields the best results, nor which resulting features can point process developers in the direction of improvement.

- Application of data mining and machine learning requires a considerable upfront investment, for example, data digitization, data scientists, and IT infrastructure. The economic benefits of applying these methods to heat treatment have not yet been reported, which raises the question of whether these methods can achieve significant cost reductions or quality improvements.

3 Materials and Methods

In order to provide a transparent basis from which the proposed framework can be emulated to similar processes, this chapter lifts the lid on the materials and methods that were created and used for the analyses and predictions of the later chapters. Section 3.1 introduces the process chains, detailing the industrial production lines as well as sensor, meta and quality data collected from bainitizing Section 3.1.1 and case hardening Section 3.1.2. These data are at the heart of the mining process outlined in Section 3.2 including data set structure, preprocessing, feature extraction, and filtering techniques. Subsequently, we examine the architecture of our general ML pipeline (3.3) and a custom hidden state pipeline (3.4) that generate predictions or forecasts from the processed data. Section 3.5 finally describes the implementation with Python.

3.1 Process Chain and Data Collection

This section briefly explains the available data pools along the process chain from material composition, over metadata and sensor signals of the heat treatment process to the assessment of the component quality that shall ultimately be predicted.

The components subjected to heat treatment are parts of the common rail direct fuel injection system, including a high-pressure pump, producing the desired injection pressure and an injector, dispensing fuel in the desired amount and frequency into the vehicle's engine. Both are connected via

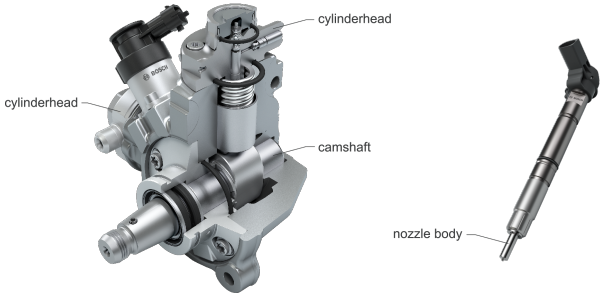


Figure 3.1: CP4 high-pressure pump (left) and CRI3 piezo injector (right) of a common-rail system [12]

rails. Figure 3.1 shows a CP4 high-pressure pump, consisting of two high-pressure elements, each integrated into a housing with its own camshaft, moving to generate the required high pressure. On the opposite, the pump piston is held by the cylinder head through which injection takes place. The component of interest regarding the injector is the nozzle body which partly extends into the engine's cylinder. As components differ in material and functional requirements, they are subjected to carefully designed heat treatment procedures explained in the following.

3.1.1 Bainitizing

Material

In this study, the CP4 cylinder head is the principal object of investigation for the heat treatment process bainitizing. All the different types of CP4 cylinder heads are made from the bearing steel 100Cr6 (1.3505, AISI 52100) and are processed as well as quality tested in the same way, which is why they can be analyzed together. Hot-rolled bars of 100Cr6 in Ovako's specification 803Q are used for production, as they have a particularly high purity grade in terms of size and distribution of non-metallic inclusions as

Table 3.1: Chemical composition of the bearing steel 100Cr6 in wt.-% with upper and lower limits as contracted with the supplier. The mean was calculated from three melting certificates.

	Cr	**C**	**Mn**	**Si**	**Ni**	**Cu**	**Al**	**Mo**	**P**	**S**
Upper limit	1.60	1.00	0.40	0.40	0.250	0.250	0.055	0.100	0.020	0.002
Mean	1.47	0.96	0.31	0.24	0.084	0.081	0.031	0.027	0.008	0.001
Lower limit	1.40	0.92	0.20	0.15	-	-	0.020	-	-	-

well as minimal variations in chemical composition. The chemical composition of the raw material, produced by ingot casting, is given in Table 3.1 providing upper- and lower limits as well as an average of measurements taken from three different batches.

Production line

Cylinder heads are bainitized in salt bath lines, henceforth often only referred to as lines, for large-scale industrialized batch processes whereby several hundred of the components are combined in multiple layers to form a batch. Eight comparable IPSEN salt bath lines of the chamber furnace type TQA-4(5) with subsequent low-temperature circulating air furnaces are used for the investigated production. They are shown schematically in Figure 3.2 which also delineates the furnace chamber and salt bath temperature curves measured in the process steps over time. Component temperatures are not measured in daily operation but only during routinely performed temperature uniformity surveys. Below can be found a detailed description of the two-step bainitization process scheduled and controlled by DEMIG's Prosys.

After automatically being pushed into the furnace chamber, the batch is heated to austenitizing temperature in the natural gas-driven furnace,

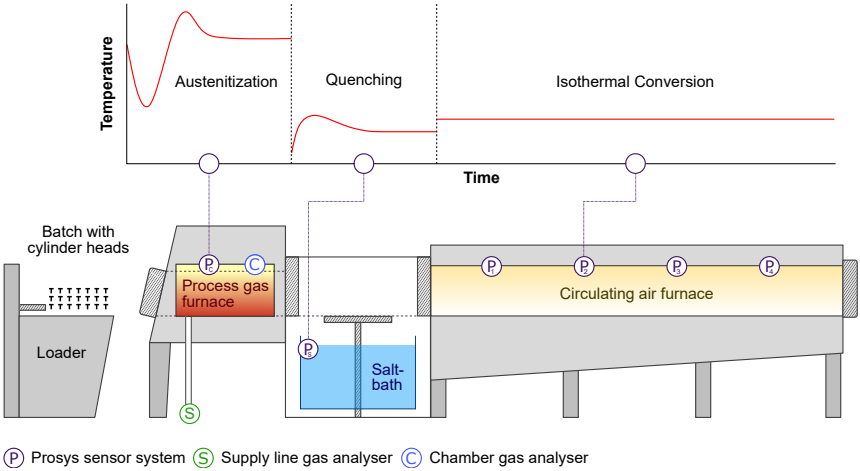


Figure 3.2: Diagram of process gas furnace with integrated salt bath and low-temperature air circulation furnace. Three sensor systems are depicted: Prosys sensor system (purple) as part of the feedback control system with sensors in the complete line, supply line gas analyzer (green) to measure gas mixture that goes into the furnace, and chamber gas analyzer (blue)

where it remains for a defined period. Upon completing the required soaking, the front and middle doors are raised just enough to allow a batch to be pushed into the next chamber and quickly lowered into the salt bath for quenching, beginning the bainitic transformation. Consequently, the salt bath temperature rises as components are much hotter than the salt and the bainitic transformation is an exothermal process, requiring external cooling of the bath to achieve a steady temperature. The slight temperature overshoot in the salt bath depends on the feedback controller settings. Finally, batches are lifted and moved through a circulating air furnace in which the components are kept at an isothermal temperature above salt bath level until the desired degree of transformation is reached.

Metadata

Each measured cylinder head carries, in addition to its quality measurements, relevant meta-information listed in Table 3.2, which is any information that is not a time series from a sensor:

- The batch position points to the exact location in the batch,
- the line indicates which salt bath line was used for heat treatment,
- date and time stem from the point when the batch is pushed into the furnace,
- alarm type and duration are recorded from the system, and
- component type defines the cylinder head's geometry.

The possible influences of these data points are studied in the chapters indicated in the eponymous column. To avoid misunderstandings it is already mentioned here that not the date itself is used as direct input for the IIR filter, but the intervals between successive batches have to be determined. These intervals are important because a longer interval indicates that the last input value used to update the IIR filter may not be a good indication of the current state.

Sensor signals

In order to acquire data, two different types of sensor systems are used, indicated in Figure 3.2 by circled letters: Prosys (P) and supply line gas analyzer (S). Whilst the two systems are described in the following paragraphs, a list of the most important sensor signals of all systems are given in Table 3.3.

[1] Indicates which kind of components are treated in the furnace prior to the actual batch. Some components use up more of the carbon contained in the atmosphere than others. This might have an influence for the following batches.

Table 3.2: Metadata and features derived for bainitization

Name	Type	Example	Feature	Chapter
Batch position	categorical	9	-	4.2.2
Line	categorical	26	one-hot line	5.2.2
Date	ordinal	24.04.2019	IIR	5.2.3
Time production start	ordinal	12:34:56	-	
Alarm type	categorical	door defect	duration	5.2.3
Alarm duration	numerical	300 seconds		
Component type	categorical	slim line	-	-
Previous components[1]	categorical	cylinder head	-	-

The Prosys system (P) offered by DEMIG is installed in each of the ten production lines providing feedback for its control system. Sensors are mounted throughout the entire line while data is stored on a central server to guarantee traceability of possible complications (e.g., evaluating the heat treatment process of components later failing in the field). Data is recorded and stored for the time a batch is in a particular production step, leaving the state of empty chambers unrecorded. Records can be exported as XML files after process completion, including a unique tracking number of the batch as well as timestamps that allow for synchronization with foreign system events.

One of these systems is the gas analyzer (S) of the pipeline, which supplies gas to all furnaces. Analyzing the gas mixture in the supply pipeline is important as its composition fluctuates over time, e.g., winter to summer, due to the fact that the provider must guarantee only a minimum calorific value, not a defined gas composition [30].

Table 3.3: Selection of sensor signals recorded during bainitizing

Sensor	Measure	Unit	Sensor	Measure	Unit
P_C	Furnace temperature	$°C$	S	Methane CH_4	%
P_C	Oxygen O_2	mV	S	Ethane C_2H_6	%
P_C	Calculated carbon level	$\%C$	S	Propane C_3H_8	%
P_C	Mass flow src	l/h	S	Carbon dioxide CO_2	%
P_S	Salt bath temperature	$°C$			
P_1	Heating power left	%			
P_1	Heating power right	%			

Table 3.4: Bainite: Quality data of cylinder head after bainitization

Label	Type	Example	Scaling	Chapter
Surface hardness	numerical	700 HV	robust	4.2.1
Core hardness	numerical	680 HV	robust	4.2.1

Quality Assessment

Various inspection characteristics must be fulfilled to release a batch for further handling. That is, process limits must not have been exceeded, and hardness measurements, as well as microstructure analysis, must indicate successful heat treatment. For further quality assurance, periodic sampling is performed to determine the carbon content. The last and, for economic reasons, least frequently performed evaluation is a pulse test, mimicking the real-life operation.

A minimum of one[2] cylinder head of every batch is taken from a defined test position to determine hardness and microstructure. Surface and core hardness are assessed by averaging three HV 10 indents. The former is on a partially ground part of the surface, the latter on a microsection of the cylinder head. After cutting, embedding, grinding, and polishing, hardness measurements are executed automatically, consisting of indentation and optical measurement. While the averaged core and surface hardness values are stored in a database, the six original values are only noted in the paper version of the batch document.

Figure 3.3 schematically illustrates a longitudinal cut through a cylinder head with three indents for core hardness measurement. Specialized staff uses microscopes to compare the microsection to reference images to determine microstructural properties like needle length, internal oxidation, and carbide formation. The resulting classifications are also entered into the database.

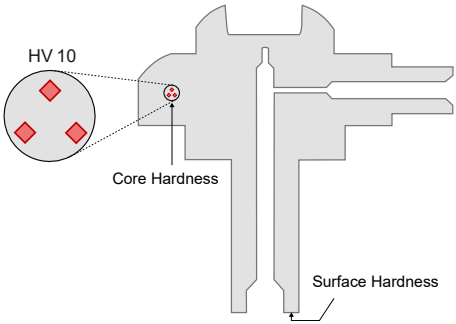


Figure 3.3: Schematic longitudinal cut of cylinder head with exemplary check positions

Samples chosen for GDOES are analyzed by a GDA650 from Spectruma GmbH. It is equipped with a high-resolution CCD-optic with a focal spot

[2] For quality assurance purposes, more than one cylinder head is tested if the heat treatment process does not remain within the defined limits.

diameter of 2.5 mm. As the ablation of the sample's layers begins on its contaminated surface, measurements can only be relied upon from a depth of approximately 2 µm, depending on the roughness of the surface. The carbon profile is created into a depth of about 25 µm.

3.1.2 Case hardening

Material

Injection nozzle bodies are manufactured from the case hardening steel 18CrNi8 (1.5920) supplied by the Stahl Judenburg GmbH in a certified degree of purity. In particular, the high proportion of nickel ensures a desired degree of hardenability. An inspection of the steel composition can be found in Table 3.5 along with upper and lower bound according to BOSCH order specifications.

Table 3.5: Chemical composition of the case hardening steel 18CrNi8 in wt.-% along with upper and lower bound according to BOSCH order specification. The symbol "-" indicates that no upper and/or lower bound is given

	Cr	**Ni**	**C**	**Mn**	**Si**	**Cu**	**Al**	**Mo**	**P**	**S**
Upper limit	2.100	2.150	0.220	0.640	0.300	-	0.040	0.15	0.035	0.035
Mean	2.006	2.075	0.176	0.567	0.155	0.002	0.031	0.006	0.014	0.021
Lower limit	1.700	1.750	0.130	0.360	-	-	0.015	-	-	-

Production line

Nozzle bodies are processed in batches of several thousand pieces, not necessarily of exactly the same type. They pass through a vacuum furnace, a deep freezer, and a tempering furnace, as depicted in Figure 3.4, all

governed by the same controlling system Prosys by DEMIG. For the investigated production the following equipment is available: Three comparable IPSEN VUTK-524 vacuum heat treatment furnaces, two deep freezers (one Linde LKS 1.0 and one CES of type CTC-LIN-900x7000x1200-S-FL), and three IVA tempering furnaces of type RH 966 RVE. The route on which the batches pass through the stations is variable (e.g., after furnace #1, batches can be deep cooled in either the Linde or CES freezer).

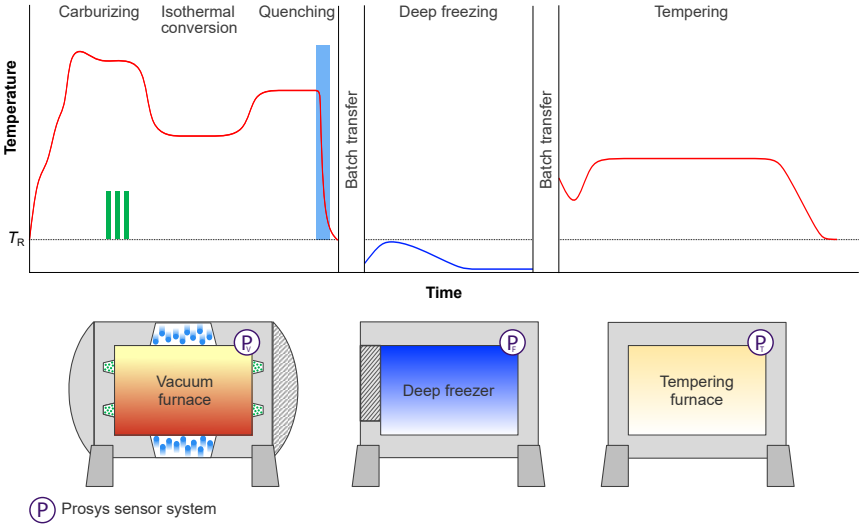


Figure 3.4: Diagram of vacuum furnace, deep freezer, and tempering furnace with respective temperature profiles. Green bars indicate carburization by acetylene injection, the blue bar quenching by nitrogen, [own representation]

Vacuum furnaces are loaded by forklifts, the door is closed and evacuation begins. After sufficient pressure reduction, the chamber is flushed with nitrogen for initial convection. The subsequent evacuation step at elevated temperature is followed by radiant heating in several steps to austenitization temperature, giving pieces in the middle of the batch time to catch up with the temperature of their colleagues closer to the heating rod. At

T_{A1} the furnace is flooded with hydrogen to reduce possibly existing oxides on the surface of the component, which is pumped out shortly before the carburization phase to avoid dilution of the carbon donor with hydrogen [89]. Austenitization takes place at a slightly higher temperature than subsequent carburization by pulsed injection of acetylene (systematic name: ethyne), indicated by the green bars in Figure 3.4. The nozzle bodies' desired case hardening depth and the required surface near microstructure, e.g., retained austenite or carbide formation, depend on the number and length of acetylene pulses and the diffusion times at the end and intermediate to the acetylene pulses. To control the surface layer properties (i.e., the carbon profile), the exact concentration of carbon achieved in the diffusion stage at a certain depth must be carefully calibrated [81].

To reach the isothermal conversion stage, the temperature is lowered by mild quenching via nitrogen injection. Soaking at a temperature for pearlitic phase transformation, which supports refinement of grains, and subsequent heating to austenitization temperature, which now is considerably lower for the hardening step, takes several hours. The final quenching below T_{MS} is achieved by high-pressure injection of nitrogen alternating between top and bottom, indicated by the blue bar and balls. When a temperature close to T_R is reached, the door can be opened and the batch transferred via forklift.

Freezers are constantly kept at a temperature well below T_R. The batch is, thus, placed in an already cooled chamber – constantly controlled by liquid nitrogen – to reach a sufficiently cold component temperature, allowing the transformation of the retained austenite into martensite to the desired degree.

Tempering the freshly deep-frozen components is the final heat treatment step and takes place in one of the three tempering furnaces that are already heated to a temperature below 200 °C. After a considerable drop in furnace temperature due to the cold batch, T_T is recovered by strong counter

Table 3.6: CH: Metadata and features derived

Name	Type	Example	Feature	Chapter
Batch position	categorical	c1	one-hot position	4.3.2
Date	ordinal	24.04.2019	IIR	4.3.2
Time production start	ordinal	12:34:56	-	-
Vacuum furnace	categorical	1	one-hot vacuum	5.3.2
Freezer	categorical	2	one-hot freezer	5.3.2
Tempering furnace	categorical	3	one-hot tempering	5.3.2
Component family	categorical	F00VW	one-hot family	5.3.3
Alarm type	categorical	door defect	duration	5.3.3
Alarm duration	numerical	300 seconds		

heating and kept for several hours to gain the desired component properties in terms of strength and toughness.

Metadata

Table 3.6 provides an overview of the meta-information available for each tested component of a batch of nozzle bodies. While the lines for bainitization are cohesive heat treatment lines, a nozzle body can take different routes through vacuum furnace, freezer, and tempering furnace. The remaining information is similar to the bainitizing data but exhibits different behavior, as will be shown in the upcoming chapters.

Sensor signals

All sensors are linked to the central control system Prosys that appoints a unique identifier (Prosys ID) to each batch to allow traceability between

Table 3.7: Selection of sensor signals recorded during case hardening, deep freezing and tempering

Sensor	Measure	Unit	Sensor	Measure	Unit
P_V	Temperature furnace	°C	P_F	Temperature freezer	°C
P_V	Pressure Barocel	mbar	P_T	Temperature furnace	°C
P_V	Pressure	mbar	P_T	Temp. heating rod	°C
P_V	Acetylene C_2H_2	l/h	P_T	Pressure	mbar
P_V	Leakage rate	mbar $\frac{m^2}{s}$			

production steps. A selection of relevant signals is listed in Table 3.7 while all systems include multiple temperature sensors as a backup in case of malfunction. Signals are recorded and stored only for the process-time along with the respective Prosys ID as XML files. Due to a dissimilar design of freezer #1 and #2 location and number of mounted temperature sensors differs considerably, leading to slightly varying signals.

Quality Assessment

Two nozzle bodies are sampled from every batch with alternating positions (i.e., components from positions 1 and 2 for batch i, components from positions 3 and 4 for batch i+1). Every sampled nozzle body is longitudinally cut as depicted in Figure 3.5, embedded, ground, and polished. A DuraScan from Struers then performs hardness evaluation with HV 1 on various positions and distances from the surface, an excerpt of which is given in Table 3.8, depending on the geometry of the nozzle body type.

Combined average scores were calculated for some measurement position pairs with similar distances to the surface to mitigate measurement errors. We subtract the mean $\bar{x}_{m_i}$ of the two measurement position ($m1$ and $m2$)

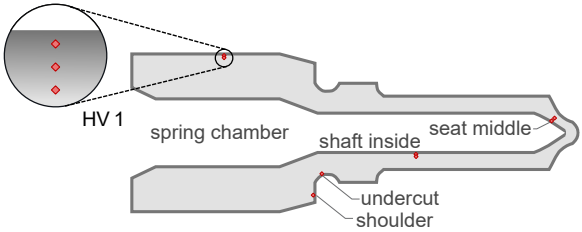


Figure 3.5: Schematic longitudinal cut of nozzle body with an exemplary check positions for hardness measurements

Table 3.8: Quality data after case hardening on specific positions of the nozzle body

Position	Distance	Combined	Position	Distance	Combined
Undercut	0.1 mm		Seat middle	0.4 mm	
Shoulder	0.1 mm	} Score 0.1	Shaft inside	0.4 mm	} Score 0.4
Seat middle	0.1 mm		Shaft inside	0.7 mm	
			Core		} Score 0.7
			CHD (550)		

distributions $\mathbf{x}_{m_i}$, then add the values in each array[3], and divide by two, see Equation (3.1).

$$\mathbf{x}_{Score} = \frac{(\mathbf{x}_{m1} - \bar{x}_{m1}) + (\mathbf{x}_{m2} - \bar{x}_{m2})}{2} \tag{3.1}$$

[3] An array contains the labels of all samples as a column vector.

3.2 Data Mining

3.2.1 Terminology

As the art of data analysis and predictive modeling is practiced in many different scientific domains, which all contributed to the field in their own language, a variety of terms are being used synonymously. In contrast, other terms have miscellaneous meanings across domains. Based on [80] this thesis makes use of the following terminology:

- *Data point* refers to a single value or instance of a measurement.
- *Features* or *predictors* are explanatory variables, that is, a measurable property that ideally represents an independent and informative characteristic of an observed phenomenon, on basis of which predictions are being made. A feature is either calculated from many data points (e.g., mean or maximum temperature) or a single data point (e.g., production line, day of the week)
- *Input* or *sample* refers to the set of data points (including sensor-, metadata, and derived features) that belong to the execution of a heat treatment on the same batch of components. The total number of samples is equal to the number of heat treatments for which all necessary data points (including the outcome) were recorded and stored.
- *Outcome*, *target*, or *label* refer to the value that is being predicted. In this work mostly a measured hardness.
- *Batch* in the context of
  - heat treatment refers to the group of components that were combined in a multilayer structure and processed together,
  - ML refers to the subset of samples that are processed by an algorithm in one optimization pass.

- *Training set* is the subset of all samples that is used to develop a model or pipeline and optimize its parameters.
- *Development set* or dev set is used during and/or after each training to evaluate the generalization capacity of the trained model and detect overfitting. The model does not learn parameters from this set.
- *Test set* is used only for the purpose of evaluation of the final, already optimized model.
- *ML method* or *algorithm* refers to the set of instructions implemented in a software package that calculate particular outputs from given inputs.
- *Model* is a trained instance of an ML method, that is, its parameters have been optimized during training.
- *Model training* refers to the process of a model learning/optimizing its internal parameters to make better predictions on the training set. Often model training is repeated on different data sub sets and hyperparameter settings.

3.2.2 Data sets

Multiple data sets of varying sizes are used throughout the thesis in order to be able to test multiple hypotheses regarding explainability of measurement and process noise as well as resulting quality scatter. Table 3.9 indicates the respective number of measurements per label[4], how many batches were heat-treated, and the number of specimens taken from each batch, as well as kind of measurement. The sets in row 1 are historic data, that is, the three original hardness measurements where retrieved manually from the quality inspection files from 565 of the heat treatments. That

[4] The number of labels depends on the use case. For bainitization, they can be found in Table 3.4, for case hardening in Table 3.8.

means, these values contain the full measurement error (incl. specimen preparation, different equipment, etc.) that will later affect the labels to be predicted. The sets in rows 2 and 3 were created by conducting 100 indents on a hardness comparison plate. The respective steel manufacturer provided the material composition set as a CSV file. Labels from the remaining sets stem from respective quality databases at Bosch. Sensor and meta information was extracted from XML files produced by the DEMIG system for each batch and then stored in an SQLite database on the author's local machine for easier access. For all machine learning and optimization procedures, we used the training set that roughly accounts for the first 70 % of all samples, i.e., data from before 01.01.2020 for the bainite use case and before 01.01.2017 for the case hardening use case.

Table 3.9: Data sets for analysis. # Label: number of measurements per label, # Batches: number of batches heat treated, Sens: (✓) if data from sensors is available for these labels, #Spec.: number of specimens taken from each batch., Meas: type of measurement procedure where 3·HV (three indents in same area of measurement position), B: bainitizing, C: case hardening, Section: reference to corresponding Section in this work, Analysis: objective of the consideration. Note: The first line contains data from daily quality control of cylinder heads, while lines 2 and 3 were created using a hardness comparison plate.

# Label	# Batches	Sens	# Spec.	Meas		Section	Analysis
3·565			1	3·HV 10	B	4.2.4	Measurement error
100			1	HV 10	B	4.2.4	
100			1	HV 1	C	4.3.4	
900	100		9	HV 10	B	4.2.2	Position and benchmark
8600	4300		2	HV 1	C	4.3.2	
160			1	wt.-%	B	5.2.1	Material composition
370			1	wt.-%	C	5.3.1	
21800	21800	✓	1	HV 10	B	5.2, 6.2	Meta and process feature for ML
11500	6900	✓	>=1	HV 1	C	5.3, 6.3	

3.2.3 Resampling and segmentation of time series

Resampling

While the sensors continuously deliver an output signal, measurement points are saved approximately every 60 seconds most of the time. That is, during time-critical events like quenching, the frequency is raised. In order to be able to write all time series in one table with shared, equidistant timestamps, they must be resampled and cut to equal length, which is necessary for multiple reasons.

Operating on a complete table with synchronized, equidistant time series instead of working on individual series makes their segmentation, extraction of features, and plotting much faster, easier, and more reliable in terms of comparability between time series. Additionally, most ML algorithms using complete times series as input need to be provided with the same size input, although some RNNs can work with variable length.

As measurement points are not always taken with exactly the same interval and some batches spend more time in the furnace than others, the data must first be brought in a processable format. To account for interval variability, the time series t_m with measurements x_meas is resampled with a fixed period $\Delta\, t_\mathrm{r}$ =60 s by linear interpolation resulting in x_resamp at t_{r_i} shown in Figure 3.6. During quenching $\Delta\, t_\mathrm{r}$ is lowered to appropriately capture the cooling dynamic. This fact needs to be considered when extracting features from the time series later (e.g., the temperature mean should only be calculated from time series with equidistant measurement points in order not to overweight or underweight individual temperature measurements and to preserve comparability between data sets). The respective time series cuts can be found in Section 5.2.4.

The interpolation stops before the last measurement point cutting of the reminder between $t_{\mathrm{m},max} - t_{\mathrm{r},max} > 0$. The true duration is stored separately and used as feature as was explained above. The vector size is

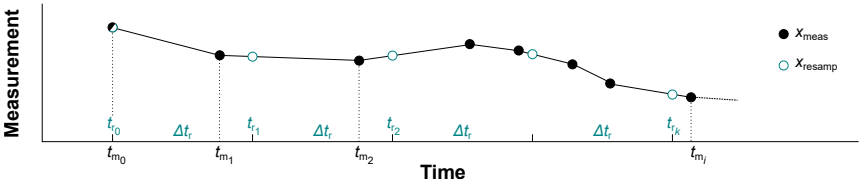


Figure 3.6: Resampling of times series using linear interpolation between measured values

now $n_r = \frac{\hat{t}_{\mathrm{r},max}}{T_\mathrm{r}}$ where $\hat{t}_{\mathrm{r},max}$ is the duration of the longest sample. For samples shorter than the maximum length the vector is filled with NaNs for feature extraction and zeros in case it is used directly as input[5].

Segmentation

In order to extract useful information from the heat treatment process, its different phases (e.g., austenitization, quenching) must be located in the time series. A time series always consists of a multi-column vector whose first column contains timestamps while the remaining columns contain the sensor measurements. The latter columns are called channels. As long as these phases are provided in one of the channels by the control system, the segmentation is straightforward. Unfortunately, this is not the case for our setups. Luckily, the bainitization process can be cut at fixed intervals, as only the last segment of each station (i.e., process gas furnace, salt bath,...) has a variable length and the remaining intervals are fixed[6]. A detailed examination of the segments is provided in Section 5.2.4.

Partitioning the case hardening process is more demanding since its operating plan contains, unlike the previous procedure, *wait-until* blocks that cause the heating process to dwell at a specific temperature until the gap

[5] ML packages generally do not accept inputs containing NaN values.
[6] The bainitizing program has no variable time blocks in the process gas furnace and salt bath.

between reference and the actual temperature has closed sufficiently. As a consequence, heat treatments do not have the same length (Δt_{A1} can vary significantly from batch to batch)[7] and process sections (e.g., acetylene pulses for carburizing or quenching to T_I) are, therefore, not to be found after the same timespan after the start. This means that cutting the time series at predefined intervals would lead to bins with unequal content. Consequently, intervals for cutting are either found by specifically targeting a particular location in the time series with multiple if-statements (first method) or based on jumps in reference values (second method), see Figure 3.7.

Working with the first method assumes that it is already known through domain knowledge which sections and metrics in a channel are of interest. Then, these sections can be targeted by specific requests (e.g., finding the first austenitization phase using the conditions: $t{>}t_{\mathrm{s}}$, $t{<}t_{\mathrm{s}+1}$, $T{<}T_U$, $T{>}T_L$, with fictitiously t_{s} =30 min, $t_{\mathrm{s}+1}$=120 min, T_U=800 °C, and T_L =790 °C, where T can be either the measured or target temperature). Used are then only values, that are part of this region. In general, this method is preferred because it is easier to implement and finds the better features, since only data of specific regions of interest is used to extract features (e.g., mean or skew).

The second method segments the complete process. To detect changes in reference channels, the difference quotient $\varphi(x_i, x_{i+1}) = \frac{x_{t_{i+1}} - x_{t_i}}{t_{i+1} - t_i}$ between successive points is taken. As the jumps are vastly different in magnitude, especially between different channels (e.g., a temperature change of 50 K vs. a pressure change of 2000 mbar), thresholds x_{th} have to be calibrated for each channel and often amended with additional conditions, like temperature or time ranges, to correctly identify a segmentation point. As can be seen in Figure 3.7b), the first segmentation point (i.e., $s_1 = 1$ start of heat treatment) is indicated by three points of discontinuity, one in $\varphi(T)$ and two in $\varphi(P)$, while start of segment 4 has only one. Segmentation

[7] See Figure 2.4 for the complete process.

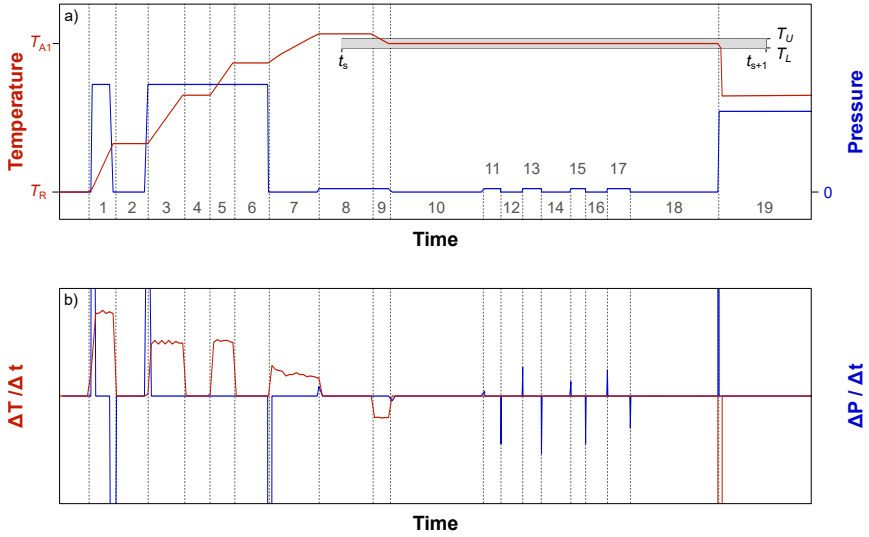


Figure 3.7: Principle segmentation methods for vacuum furnace process (e.g., case hardening): a) reference temperature and pressure during heat up and austenitization, where the gray area marks a region of special interest derived by domain knowledge for extraction of mean austenitization temperature b) respective difference quotient as basis for segmentation without applying detailed domain knowledge

points are first determined per channel, that is every discontinuity larger than a threshold $\varphi(\varphi(.)) > x_{\text{th}}$ is marked as possible cutting point t_c. Then further necessary conditions are tested for (e.g., is the value before or after the t_c equal to zero. With this, we sort out values that are part of a ramp, which is the case for segment 1, 3, 5, etc.) to find or eliminate cutting points. In a last step, cutting points are combined to a single point if they lie together close enough. In sum, over 30 segments are created in this way for the vacuum furnace, deep freezer, and tempering furnace process. The data of each sample (i.e., process data of one batch) is processed to find the respective segments. To determine whether the correct segments have been found, each segment's characteristic values (e.g., mean and length) are calculated. If for a sample either not the correct number of segments were found or a certain amount of the characteristic values of

Table 3.10: Statistical values as features extracted from resampled sensor signal $\mathbf{x}_{c,s}$ of channel c in segment s

Statistical feature	Notation	Statistical feature	Notation
Mean value	$\overline{x}_{c,s}$	Standard deviation	$x_{c,s,sd}$
Median value	$x_{c,s,med}$	Skew	$x_{c,s,skew}$
Maximum value	$x_{c,s,max}$	Kurtosis	$x_{c,s,kurt}$
Minimum value	$x_{c,s,min}$	Segment duration	$\Delta t_{c,s}$

the segments lies outside a range of 3 standard deviations from the mode of the characteristic distribution from all samples, then the sample is excluded. After complete segmentation, all segments are resampled to an appropriate frequency depending on the dynamic of each segment.

Although this method works for over 99 % of the samples which stem from a period of about 6 years, significant changes in the heat treatment procedure intervals would likely need either a fairly complicated adaption of the segmentation determination or be incompatible with the current segmentation. Previous data could not be used anymore, or a different segmentation would be needed to be used for all samples. If the segmentation and resampling procedure is successful, we can now extract statistical indicators from the respective sections.

3.2.4 Process feature extraction

Given that segments s have already been defined as spanning from t_s to t_{s+1}, then vector $\mathbf{x}_{c,s}$ contains the resampled measurements in s, of channel c belonging either to a particular sensor or other to information from the process controller. From this vector the features given in Table 3.10 are calculated and stored in a feature vector $\mathbf{x}_{f(c,s)}$. Figure 3.8 illustrates this process exemplifying the statistical properties of the distribution formed

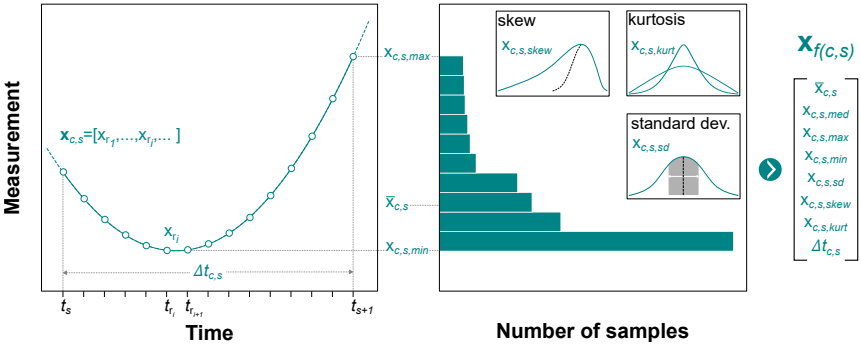


Figure 3.8: Feature extraction from resampled measurements $\mathbf{x}_{c,s}$ of channel c in segment s

by the values in $\mathbf{x}_{c,s}$. In addition, further process specific features are calculated from these statistical values (e.g., Hollomon–Jaffe parameter from the temperature channel using $\bar{T}$ and Δt in the respective segment of the tempering furnace). Feature vectors $\mathbf{x}_{f(c,s)}$ are calculated domain knowledge specific only for the interesting segment and channel combinations (e.g., the mass flow of acetylene is irrelevant during quenching or temperatures far below austenitization). Feature vectors of interest are stacked as shown in Equation 3.2 and are appended by process-specific features like the Hollomon-Jaffe parameter H_P. The matrix $X_{input}^{process}$ then should hold all potentially relevant information about the process each sample went through. For clarity, sample identifiers (1^{st} and n^{th}) are not used in the matrix H_P, but the sample is written above the respective column.

Table 3.11: Examples for duration features derived from alarm durations

feature	$alarm_1$	$alarm_2$	$alarm_i$	...
duration	0	0.1	0.95	...

$$X_{input}^{process} = \overbrace{\begin{bmatrix} \overline{x}_{1,1} \\ x_{1,1,med} \\ x_{1,1,max} \\ x_{1,1,min} \\ x_{1,1,sd} \\ x_{1,1,skew} \\ x_{1,1,kurt} \\ \Delta t_{1,1} \\ \vdots \\ \mathbf{x}_{f(c,s)} \\ \vdots \\ H_P \end{bmatrix}}^{1^{st}sample} \cdots \overbrace{\begin{bmatrix} \overline{x}_{1,1} \\ x_{1,1,med} \\ x_{1,1,max} \\ x_{1,1,min} \\ x_{1,1,sd} \\ x_{1,1,skew} \\ x_{1,1,kurt} \\ \Delta t_{1,1} \\ \vdots \\ \mathbf{x}_{f(c,s)} \\ \vdots \\ H_P \end{bmatrix}}^{n^{th}sample} \quad \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \text{features extracted from } 1^{st} \text{ segment and } 1^{st} \text{ channel (e.g., } T_{\text{furance}} \text{ when batch is loaded into furnace)} \\ \\ \text{features extracted from channel c in } s^{th} \text{ segment} \end{matrix} \tag{3.2}$$

Each sample is then further appended by the one-hot encoded meta information from Tables 3.2 and 3.6, respectively. Alarms are included as features as shown in Table 3.11. If an alarm did not occur for a particular sample it is assigned value 0 (e.g., $alarm_1$), else the scaled alarm duration is used (e.g., in the particular example in Tables 3.2 $alarm_2$ occurred. It had a duration of 10 % of the longest $alarm_2$ that occurred of all batches).

3.2.5 Filtering

Ordinarily, filtering is applied to time series of noisy sensor signals, but fortunately, neither temperature nor pressure sensors are affected. Hardness measurements, by contrast, are strongly affected. As will be shown

in later chapters, the mean hardness of consecutive batches is not stationary for longer periods but fluctuates or drifts significantly. To follow a hardness trend of consecutive hardness measurements over weeks, these measurements, shown in gray and green in Figure 3.9, are fed to an IIR filter. The black line shows the closest approximation to the true hardness trend realized by a noncausal[8] filter with no phase delay. Our goal is to follow this black line as closely as possible, however we are limited by the information available (i.e. 1. we cannot look into the future, 2. we do not measure all batches), elaborated on below. The small dots show the hardness of a sample in green (if it was actually measured) and gray (if it would have been measured) from a particular batch at the time it was made, which is why they are not equidistant[9].

If the black trend was to be evaluated over a longer period of time a noncausal filter solution (or central rolling window) would be optimal. However, for prediction purposes, we are interested in the immediate development of the trend but can only look as far as the subsequent measurement. What is more, cost reduction dictates that only every n^{th} batch (e.g., for $n = 4$ at $t_0, t_4, t_8, t_{3n}, t_{4n}$) is tested for hardness, leaving us with fewer values to estimate the trend. As results are needed within a fixed time frame after production, only a causal[10] filter comes into consideration, introducing a phase delay which in Figure 3.9 adds up to around the time between three consecutive batches. How close the prediction may come to the black line of the noncausal filter depends strongly on the size of this fixed time frame.

In the following example, the goal is to predict the batch at t_7. Specimens from each 4^{th} batch are tested and testing takes Δt_m. Predictions and filter outputs in Figure 3.9 are drawn at the point in time (e.g., t_4, t_7) for

8 Uses values from the "future". That means it can only be applied in hindsight if the measurements are already available (no real-time filtering).

9 The time between individual batches produced on the same line may vary considerably. The same line may also produce different component types.

10 Only values from past and present can be used.

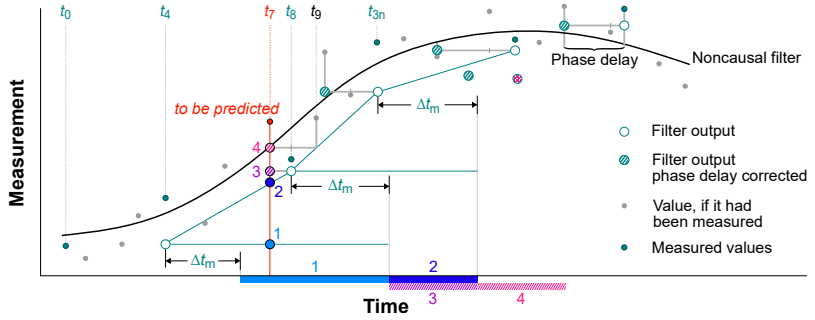


Figure 3.9: Four predictions for the batch at t_7 based on a filter applied to intermittently tested batches dependent on the available information of past and future measurements. The colored bars on the time line indicate the time frame for which a certain prediction (1-4) is achievable. Better predictions need more information and are, thus, only available later. White diagonal stripes indicate a phase delay correction. The noncausal filter (black line) applied to hardness measurements of consecutive batches is the benchmark of best possible approximation

which they are approximations, not at the point in time at which they were generated. Doing so, the following scenarios with chronological, increasing information availability are likely:

1. Uncorrected: Only the measurement of the last test batch at t_4 is available and the last output of the filter is used as prediction.

2. Interpolation: The measurement from the batch at t_8 is available and an interpolation between the last two filter outputs is used as prediction.

3. Phase corrected: Same circumstances as before, but a phase delay correction is "applied" by using the prediction of the batch two periods into the future at t_9. In this case the prediction for t_9 is the last filter output at t_8.

4. Interpolation and phase corrected: The time frame is so long that we can wait for the measurement at t_{3n}. With this information we can use an interpolation for t_9 as the best prediction for t_7.

The example shows that the phase delay can be mitigated by shifting the filter results (or interpolations in between) k steps backward. If k = -2 the filter output at t_9 is used as prediction for t_7 (in the past of t_9), or in other words, if we want to have a prediction for t_7 we must wait for the prediction at t_9 to correct for the phase delay. Shifting the filter results k steps forward[11] ($k > 0$) simulates the information delay when a prediction has to be made without the measurements of the previous k batches available (e.g., specimen preparation is delayed during the night shift). It can also be interpreted as forecasting the hardness of the batch k steps into the future from the last measured batch.

To simulate this process of intermittent testing, a digital low-pass Butterworth filter was implemented. For filter design the scipy function `signal.iirfilter(`N,ω`)` was used, which returns the filter coefficients b and a [144]. Its parameters (i.e., order N and cutoff frequency ω) were optimized by Dual Annealing [149] (`scipy.optimize.dual_annealing`). The results are presented in Section 6.2.1. Optimization resulted in order $N = 1$ for over 99 % of cases. A first-order Butterworth filter then takes the simple form of Equation (3.3), where y_i is the filter state or output and x_i a measurement (e.g., hardness) for time/batch i. The parameter a can be interpreted as the percentage of the memorized value y_{n-1} (i.e., the previous filter output) that is used for the next prediction while $b = 1 - a$ is the proportion of the new measurement used to update the previous state y_{n-1}.

$$y_n = \frac{b}{2}(x_n + x_{n-1}) + a\, y_{n-1}, \quad \text{with } b = 1 - a; a, b \in (0, 1) \tag{3.3}$$

Figure 3.10 depicts the coefficients a and b at different ω. Accordingly, a first-order filter with $\omega = 0.03$ uses 91 % of its memorized previous state and updates it with 9 % of the averaged last and current measurements.

[11] E.g., if $k = 1$ the filter output at t_1 is used for t_2 (that is in the future of t_1).

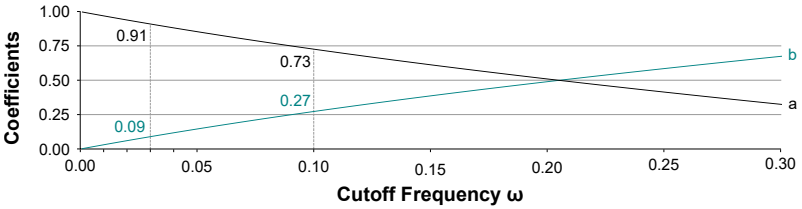


Figure 3.10: Relationship between filter coefficients a, b, and cutoff frequency ω of first-order Butterworth filter

The optimization was restrained from using $\omega > 0.5$, which would lead to very fast filters that overshoot the target for a step response.

Filters are often not applied to a complete series of measurements (e.g., from 2018 to 2021) but to chronologically coherent subseries, where two measurements are not further away than 10 days. Otherwise, the series is cut. Filters are initialized with the mean of the first three measurements of that subseries. Lastly, filters are often not applied to all measurement points at once but only to every second or n^{th} measurement. To still make use of the complete training set, the filter is then applied twice or n times to every other not used measurement, see Table 3.12. Parameters are then optimized by dual annealing based on the joined RMSE.

Table 3.12: Two rounds of filter application to every second measurement

Round	**Measurements→**	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	update filter with	x_1		x_3		x_5		x_7		x_9	
	predict		x_2		x_4		x_6		x_8		x_{10}
2	update filter with		x_2		x_4		x_6		x_8		x_{10}
	predict			x_3		x_5		x_7		x_9	

3.2.6 Data analysis and visualization

Because this work is strongly supported by visualizations, the following guidelines shall help the reader to understand the depictions easily. All figures were created or post-processed using Inkscape [67]. For easier recognizability, temperature differences in figures are given in $\Delta°C$ in accordance with DIN 1301-1, not in Kelvin.

Colors, if not stated otherwise, usually carry the following meaning: blue for surface (or surface near) hardness measurements, green for core (or surface distant) ones; violet for prediction related results; yellow for measurement error; rainbow for different salt bath lines, shades of red, blue, and orange for case hardening stations.

Box plots, if not stated otherwise, show 90 % of the data ranging from the 5^{th} to the 95^{th} quantile, with the box containing 50 % ranging from the 25^{th} to the 75^{th} quantile being divided by the median (shown in orange). Notches around the median usually indicate the 99.9 % confidence intervals (CI), determined by bootstrapping with 10.000 iterations. Outliers are omitted to avoid cluttering the depiction and reduce window size for better focus on the difference of distributions.

Histograms, if not stated otherwise, are centered around their respective mean $\bar{x}$ or median x_{med} and usually depict hardness distributions. That is, from each value in the distribution $\mathbf{x}$ their mean $\tilde{\mathbf{x}} = \mathbf{x} - \bar{x}$ or median $\tilde{\mathbf{x}}_{med} = \mathbf{x} - x_{med}$ are subtracted and a histogram of the shifted distribution $\tilde{\mathbf{x}}$ is plotted. Consequently, negative values in the graph are softer than the mean or median and positive values are harder.

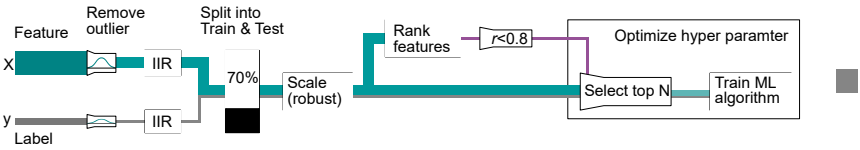


Figure 3.11: General machine learning pipeline

3.3 General Machine Learning Pipeline

The following sections introduce the ML pipeline used in Section 6.2.2 depicted in Figure 3.11. The IIR filters are those introduced in Section 3.2.5. Training data makes up the first 70 % of all samples (before 01.01.2020 for bainite and before 01.01.2017 for case hardening). Transformation to similar variance is done with the robust scaler from scikit-learn. The remaining pipeline steps are detailed from left to right below.

3.3.1 Outlier removal and drift correction

Before the actual train and test sets are built, outliers need to be removed from the feature as well as the label set because the IIR filters are sensitive to those extreme values and would falsely correct successive values. The distribution of the individual features and labels over all samples are analyzed and samples removed whose values lie further than 4 standard deviations away from the median. The remaining samples are drift corrected by an IIR filter of order N=1, ω_{feature}=0.01 for features and ω_{label}=0.018 for the labels. Only each n^{th} measurement may be used for the label correction, where correction means that the filter output is subtracted from the original values to remove the drift or fluctuation. In case predictions of real hardness are of interest (e.g., serial production), the filter output is later added back to the prediction of the pipelines. An evaluation that only considers the predictability due to the process variation works without this addition. Finally, after the chronological train-test split, a robust scaler is used on the features.

3.3.2 Feature ranking

To provide the machine learning algorithm with the most informative features, the following algorithms were used to rank them by predictive importance: Sequential feature forward selection (SFS) with linear regression, genetic algorithms (GA) with linear regression, manual selection with domain knowledge, feature importance attribute by random forest (RF), mutual information criteria (MI), and F-score. The packages used can be found in Table 3.13. The RF was trained 80 times and the mean of the `feature_importances_` attribute used to rank the features. The genetic algorithm usually does not provide a ranking but returns a set of important features. Hence, the algorithm was deployed several times with increasing size of the feature sets to return (i.e., `max_features` in [5, 10, 15, ..., 60]). These sets were then concatenated by adding those features from the respective larger set that were not yet in the list of ranked features. Each method generates a list of features sorted by their importance from most to least predictive. Features of this list are then successively correlated with features of a lower rank, where the latter may be removed if $r > .8$. The pipeline then uses this final list to sort the feature matrix accordingly.

3.3.3 Pipeline optimization

Pipelines were created by the following sklearn function:
`make_pipeline( SelectPercentile(percentile), RobustScaler(), <ML_method>(*args))`, where `percentile` is an argument that determines how many top N features are handed to the scaler and then to the ML methods which may differ considerably in their complexity and number of tunable hyperparameters. While NNs have many such parameters (e.g., number of layers and their size, learning rate, activation function, etc.) LRs possess none.

[12] is an ANN

Table 3.13: Hyperparameter settings (=) and/or their range (:) for optimization of algorithms. The Type column indicates the usage of the algorithm, where O is the optimization of: OFe = features, OFH = hidden states pipeline, OFi = filter parameters, OP = pipeline, and Su = supervised learning, Un = unsupervised learning. A reference to the packages is given in Table 3.14. The StackingRegressor, in this case, trains three NNs (i.e., MLPRegressor) in parallel and then averages their output. It might be though of as an ensemble of NNs

Type	Algorithm	Package	Hyperparameter (range)
OFe	Genetic SelectionCV	sklearn-genetic	estimator = LinearRegression, cv = 5, max_features = [5, 10, . . . , 60], n_population = 1000, n_generations = 100, n_gen_no_change = 20
OFe	RandomForest-Regressor	scikit-learn	bootstrap = True, min_samples_leaf = 12, min_samples_split = 10, n_estimators = 100
OFe	f_regression	scikit-learn	k = 'all'
OFe	mutual_info_regression	scikit-learn	k = 'all'
OH, OFi	differential-evolution	scipy	polish=True
OH	dual_annealing	scipy	
OP	BayesSearchCV	skopt	cv = 5, n_iter = 100
OP	TPOTRegressor	tpot	generations = 1000, population_size = 500, cv = 5
Su	LinearRegression	scikit-learn	
Su	MLPRegressor[12]	scikit-learn	early_stopping = True, learning_rate = adaptive, hidden_layer_sizes: Int(1,100), alpha: Real(1e-4, 0.5, 'log-uniform'), beta_1,2: Real(0.5, 0.9999)
Su	StackingRegressor	scikit-learn	estimators = [MLPRegressor, MLPRegressor, MLPRegressor]
Su	RandomForest-Regressor	scikit-learn	n_estimators: Int(10, 300), max_depth: Int(2, 15), min_samples_split: Int(2, 30), min_samples_leaf: Int(1, 30), min_impurity_decrease: Real(1e-5, 0.999, 'log-uniform')
Su	GradientBoosting-Regressor	scikit-learn	learning_rate: Real(1e-4, 0.2, 'log-uniform'), n_estimators: Int(10, 300), min_samples_split: Int(2, 30), min_samples_leaf: Int(1, 30), max_depth: Int(2, 15)
Su	SupportVector-Regressor	scikit-learn	C: Real(1e-6, 1e3, 'log-uniform'), tol: Real(1e-6, 0.99, 'log-uniform'), epsilon: Real(1e-6, 0.99, 'log-uniform')
Su	KNeighbors-Regressor	scikit-learn	n_neighbors: Int(2,100), leaf_size: Int(2,100), p: Real(1,2)
Un	PCA	scikit-learn	n_components = 2
Un	FCM	fuzzy c-means	n_clusters = [5; 300]

With growing dimensions of the hyperparameter space, tuning, and preventing overfitting of a model become increasingly difficult. Thus, a Bayesian search algorithm with 5-fold cross-validation from the skopt package is used to optimize the hyperparameters of the pipeline constituents (i.e., `percentile` and `args` of the respective ML method), the results of which can be found in Appendix Table A.1. In addition to the pipeline described above, one further pipeline was created and optimized by the TPOT regressor. The respective optimization algorithms as well as the ML methods and the parameters optimized can be found in Table 3.13. It also lists the unsupervised algorithms implemented in this work. For all parameters that are not listed the *default* of the respective function was used.

3.4 Custom Hidden States Pipeline

3.4.1 Modeling approach

As will be discovered in the upcoming Sections 4.3.2 and 5.3.2, the long-term behavior of different stations in the case hardening process as well as the influence of the batch position of the specimens and their component type can vary considerably. Because general-purpose ML pipelines can not properly describe such behavior, a hidden states model is introduced that estimates the influences and current states of the contributing factors from the measured hardness of the specimens.

Influences are collected in the model below, see Equation (3.4). It rests on the on the unconfirmed assumption that the final hardness y of a given case hardened component at a point in time t_i can be calculated as the sum of: x_{Base} influences prior to case hardening (e.g., material composition or annealing of raw material), x_R sum of the contribution of the individual stations (i.e., the route taken), Δ the static offset caused by batch position and component type, and ϵ noise including the measurement error.

$$y[t_i] = x_{Base}[t_i] + \underbrace{x_{Vacuum}[t_i] + x_{Freez}[t_i] + x_{Temp}[t_i]}_{x_R[t_i]} + \underbrace{\Delta_{Pos} + \Delta_{Comp}}_{\Delta} + \epsilon \tag{3.4}$$

For simplicity and because no other interactions are known, a simple additive model was chosen. In order to make predictions with such a model, the dynamic states x_{Base}, x_R as well as the offsets in Δ need to be estimated. Fortunately, we will see that the process parameters in industrialized operation move within a very narrow window and the hidden states[13] x are highly autocorrelated, making them predestined for a filter application. In the following, a first-order IIR filter algorithm is introduced that makes a prediction about the hardness y that is expected from a given specific combination of route, component, and position based on its hidden internal states. It then updates these hidden states based on successive hardness measurements of batches incorporating route, component type, and batch information. Before diving into the algorithm itself, the subsequent paragraphs outline its individual components, with vectors $\mathbf{v}$ in bold and subscripts denoting affiliation: route (R), vacuum furnace (V), deep freezer (F), tempering furnace (T), batch position (P), and component type (C). A superscript T indicates a transposed vector (i.e., row to column or vice versa).

For every hardness measurement y_{meas} we have additional information encoded in feature vectors $\mathbf{f}_R, \mathbf{f}_P$, and $\mathbf{f}_C$. For our particular case, the first three entries of $\mathbf{f}_R$ hold the vacuum furnace encoding, the following two the deep freezer, and the last three which tempering furnace was used (e.g., $\mathbf{f}_R = [1, 0, 0, 1, 0, 1, 0, 0]$ would activate all the first stations). Congruently, the hidden states vector holds the 8 current states of the several stations,

[13] Hidden, because they can not be measured directly but must be estimated from available measurements.

that is $\mathbf{x}_R = [x_{V1}, x_{V2}, \ldots, x_{T3}]$, which represent the hardness contribution of each station at a specific point in time. The base state x_{Base} gets its own variable.

The final ingredient are the variables to be optimized, each falling into one of two categories: 1) filter coefficients a_{Base}, b_{Base}, $\mathbf{a}_R$, and $\mathbf{b}_R$, 2) offsets $\mathbf{c}_P$ and $\mathbf{c}_C$. Coefficients are used analogous[14] to the notation for first-order IIR filters or ARMA models as shown exemplary[15] in Equation (3.5), where $\mathbf{a}_R = [a_V, a_V, a_V, a_F, a_F, a_T, a_T, a_T]$, $\mathbf{b}_R$ analogous, and $\odot$ denotes element wise multiplication. That means, that stations of the same type share the same filter coefficients (e.g., all vacuum furnaces are updated with a_V and b_V).

$$\begin{aligned} x_{Base}[n] &= a_{Base} x_{Base}[n-1] + b_{Base} y \\ \mathbf{x}_R^T[n] &= \mathbf{a}_R^T \odot \mathbf{x}_R^T[n-1] + \mathbf{b}_R^T y \end{aligned} \tag{3.5}$$

3.4.2 Model execution

The key idea to this filter pipeline is that only the offsets and states that affect the current measurement are used for prediction and update. Optimization, however, is done for all elements at once. If, for example, a harder component would, by chance, take a specific route more often, it would be impossible to find out whether the route or component contributed to the increased hardness when only looking at the final results. By including all information in a single model, such differences can be distilled out. The following paragraphs explain the individual steps to be executed.

[14] Cf. Section 2.2.2. Formally x is the signal measured and y the filter output. In our case x is the hidden state and y the measured hardness, to be consistent with ML notation.

[15] These equations demonstrate the general method and are not the exact equations later used in the algorithm. Those are explained in Section 3.4.2 below.

Activate respective offsets according to the current feature vector. Depending on which of the k batch positions and which of the j component types the measured specimen was taken from, the respective offsets are written to Δ_{Pos} and Δ_{Comp}, where $(\cdot)$ is the scalar product:

$$\begin{aligned} \Delta_{Pos} &= \mathbf{c}_P \cdot \mathbf{f}_P^T, & \mathbf{c}_P &= [c_{p1}, \ldots, c_{pk}], & (e.g., \mathbf{f}_P &= [0, 1, \ldots, 0]) \\ \Delta_{Comp} &= \mathbf{c}_C \cdot \mathbf{f}_C^T, & \mathbf{c}_C &= [c_{c1}, \ldots, c_{cj}], & (e.g., \mathbf{f}_C &= [1, 0, \ldots, 0]) \end{aligned}$$

Predict hardness from last states. Based on the model (3.4) given above, the filter predicts the hardness $\hat{y}_n$ for a given combination of route, component type, and position based on the last states $\mathbf{x}[n-1]$.

$$\hat{y}[n] = x_{Base}[n-1] + \mathbf{x}_R[n-1] \cdot \mathbf{f}_R^T + \Delta_{Pos} + \Delta_{Comp} \tag{3.6}$$

Update states with new measurement information. First, we correct the measurement y_{meas} by subtracting the estimated offsets Δ_{Pos} and Δ_{Comp} because we want to make the states update independent from component type and batch position.

$$y_{cor}[n] = y_{meas}[n] - \Delta_{Pos} - \Delta_{Comp} \tag{3.7}$$

The update of x_{Base} is calculated from Equation (3.8). Additionally, we enforce that this base state truly follows the complete amplitude of hardness drifts by setting the filter gain $a_{Base} + b_{Base} = 1$. Otherwise, the optimization algorithm might become unstable or attribute[16] parts of the overall fluctuation to $\mathbf{x}_R$ which is supposed only to carry the offset between the base and individual stations and not parts of the overall fluctuation.

[16] by setting $a_{Base} + b_{Base} < 1$

$$x_{Base}[n] = a_{Base} x_{Base}[n-1] + \underbrace{b_{Base}}_{=1-a_{Base}} \; y_{cor}[n] \tag{3.8}$$

Accordingly, Equation (3.9) calculates the delta between measurement and current base state, since the remaining hidden states in $\mathbf{x}_R$ are only updated with this difference Δy.

$$\Delta y = y_{cor}[n] - x_{Base}[n] \tag{3.9}$$

Finally, Equation (3.10) updates those hidden states that contributed to the measurement (i.e., the route taken by the batch from which the test specimen was obtained). The remaining states stay the same. Since only one measurement is used to update all the states the last term Δy in Equation (3.10) is a scalar.

$$\mathbf{x}_R^T[n] = \underbrace{(\vec{1} - \mathbf{f}_R^T) \odot \mathbf{x}_R^T[n-1]}_{\substack{\text{Keep unaffected states} \\ \text{the same,}}} + \underbrace{\mathbf{a}_R^T \odot \mathbf{x}_R^T[n-1] \odot \mathbf{f}_R^T}_{\substack{\text{use fraction of the} \\ \text{old states and..}}} + \underbrace{(\mathbf{b}_R^T \odot \mathbf{f}_R^T)\Delta y}_{\substack{\text{..update with fraction} \\ \text{of new measurement}}} \tag{3.10}$$

Stable optimization of the filter can be ensured by restricting all filter coefficients to be $\in (0, 1)$ and $\mathbf{a}_R + \mathbf{b}_R \preceq 1$. This filter is now applied to the series of hardness measurements and supplied with information about each measurement's route, component, and position. A differential evolution algorithm then optimizes the coefficients and offsets to minimize the MSE between $\mathbf{y}_{meas}$ and $\hat{\mathbf{y}}$, where $x_{Base}[0]$ is initialized with the mean of the first three measurements of $\mathbf{y}_{meas}$.

3.5 Implementation with Python

The complete framework (i.e., data conversion, databases, analysis, machine learning, etc.) was implemented using the Python programming language in the Anaconda ecosystem. Packages that provide specific functionalities are listed in Table 3.14. The choice for Python is based on the following reasoning: R is slower and has fewer ML-related packages. Matlab is not open source. C++, C#, and Java might execute code faster but take longer to implement (i.e., rapid prototyping). Python is the best choice since our use cases are not time-critical (i.e., fast real-time execution necessary).

[17] Is part of the standard Python library.
[18] Uses the DEAP package, that implements the actual genetic algorithm, for feature subset selection.

Table 3.14: Packages used for storing, preprocessing, and plotting of data as well as supervised (Su) and unsupervised (Un) machine learning

Package	**Version**	**Applikation**	**Citation**
conda	4.7.12	Package management	
python	3.7.8	-	
numpy	1.19.1	-	[59]
pandas	1.1.2	-	[116]
xml.dom.minidom	-[17]	Parse XML files	
sqlite	3.33.0	Database	
matplotlib	3.3.2	Plotting	[66]
seaborn	0.11.0	Plotting	[145]
tsfresh	0.17.0	Extract feature	[22]
statsmodels	0.12.2	Post hoc tests	[111]
fuzzy c-means	0.0.6	Un-ML	[31]
sklearn-genetic[18]	0.3.0	GA feature ranking	[15]
deap	1.3.1	GA	[47]
scikit-learn	0.23.2	Su/Un-ML, scale, encode	[110]
scikit-optimize (incl. skopt)	0.8.1	Optimization (Bayes search)	[61]
scipy	1.5.2	Optimization and filter	[144]
keras	2.3.1	Su-ML	[21]
tensorflow	1.14.0	Su-ML	[1]
tpot	0.11.5	Su-ML, optimization	[147]

4 Label Analysis

4.1 Introduction

Making good predictions usually presupposes a good understanding of the target in question, factors biasing the target, as well as the quality of the process by which the target was quantified. Often, this quantification itself is prone to scattering. Therefore, to explain the overall variation in the distribution of a label, this chapter analyzes the various sources of scatter and bias individually. It starts with an overview of the meas. pos. on the cylinder head in Section 4.2.1 and nozzle body in Section 4.3.1, followed by positional effects in the batch, due to slightly different temperatures and gas mixtures, for bainitizing Section 4.2.2 and case hardening Section 4.3.2, uncovering their differences and dependencies. Subsequently, an upper limit for predictability is derived from these dependencies Section 4.2.3 and 4.3.3. This benchmark already points out how effectively ML methods can be expected to learn from the given measurements and which limitations are set by irreducible measurement noise[1], which is explored in Section 4.2.4 for HV 10 and Section 4.3.4 for HV 1.

A list of noise generators includes, but is not limited to, the following:

– Diamond abrasion reducing edge sharpness of the indent.

[1] See [80] (p. 524) if the label is affected by significant measurement noise, the irreducible error increases in severity. The R^2 then has a lower upper bound due to this error.

- Resolution of the measurement optic limiting the precision of indent edge detection.
- Variation in specimen preparation leading to different surface conditions or a shift in indent position.
- Recalibration of measurement devices entailing an offset between measurements of different devices and/or periods.
- Carbide formation causes the surface to be harder in some places than others.

Although this list may appear exaggerated to the reader, the influence of these factors on determining hardness and, therefore, its predictability can not be understated. When considering measurement results, correlations, and predictions, one should keep in mind that hardness measurement does not equal hardness measurement, as will be shown in the following sections. The figures in these sections generally adhere to the following color schema: Blue for surface (or surface near) hardness measurements, green for core (or surface distant) ones. Additional colors are explained in the respective legends.

4.2 Bainitizing

4.2.1 Measurements on the cylinder heads

For evaluation of the bainitization process, mainly two measurements on the cylinder head are relevant, namely, core and surface hardness, as described in Section 3.1.1, taken from a fixed position in every batch. This section examines the distribution around their means, shown in Figure 4.1, along with the labels' development over time. Both histograms show a spread around 60 HV. However, while the core hardness is distributed symmetrically, with 50 % of labels in a window of 11 HV around the median,

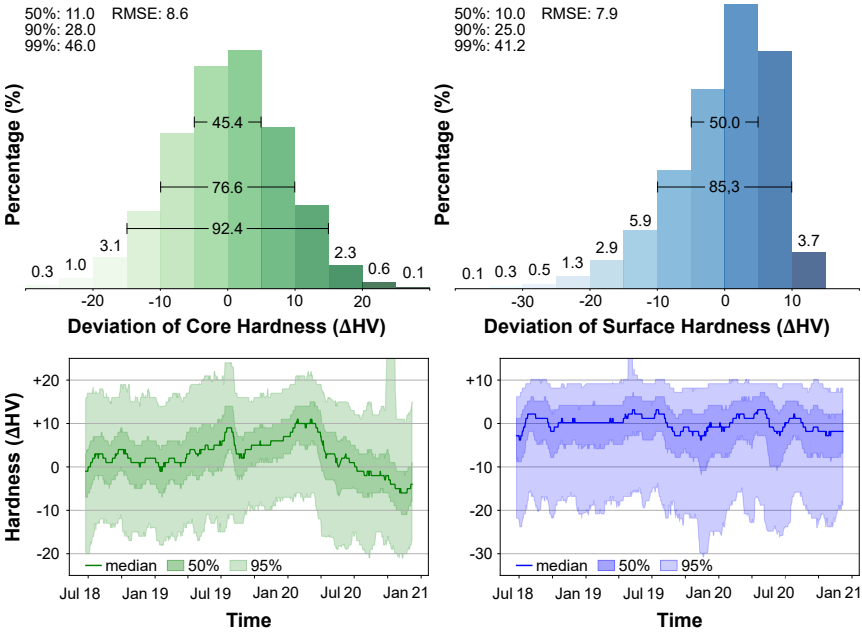


Figure 4.1: Above) distribution of core and surface hardness around their mean (=0), for explanation see Chapter 3.2.6. Below) median, 50 %, and 95 % boundaries of the labels in a rolling window over 25 days

the surface hardness is left skewed[2]. The maximal achievable hardness[3] explains this skewness leading to a sharp drop on the right side.

This representation can also be interpreted as error distribution using the mean of all values as a predictor. In this case, the prediction error would already be smaller than ±10 HV for 76.6 % of measured core hardness values. The root mean squared error (RMSE) between prediction (in this case mean) and measured value may be used as a good first indication

[2] Mean is to the left of the median, with a tail to the left.

[3] Given the amount of carbon in the material, the process gas in the furnace, and the fixed heat treatment parameters, it is physically impossible to realize greater hardness at the surface.

for a prediction's accuracy. However, to fully understand the behavior of a predictor, it is always necessary to assess the whole error distribution as predicting labels at the edges of these distributions (i.e., outliers that are too hard or soft) might prove challenging due to the vast imbalance between outliers and satisfying samples.

Regarding the behavior over time, the median of the core hardness is subject to fluctuations greater than 12 HV, making up roughly 20 % of the total spread. Further investigations suggest that chemical composition[4], as determined by the steel supplier, does affect achievable hardenability elaborated on in the upcoming Chapter 5. The frequency-of-use of the lines could not predict the drift, which does not exclude its possible influence but, at least, diminishes the probability. The measuring device might be ruled out since firstly, the daily check on a hardness reference plate does not correlate with the drift, and secondly, this behavior needed to be similar for core and surface. But, a correlation between surface and core hardness could not be found. In contrast to the factors mentioned above (that hardly seem to be impactful), the position of a test specimen in the batch produces a consistent bias. While all labels above were taken from the standard position, the following section investigates the whole batch.

4.2.2 Position in the batch

A component's position in a batch can significantly influence the heat treatment result as heating behavior, and quenching characteristics are location-dependent (e.g., components closer to the heating elements reach the target temperature faster). Thus, regularly testing multiple pieces at specific batch locations is necessary to ensure that all components in one batch meet the requirements that allow a release to the customer. It also ensures that predictions made for and learned from one position generalize

[4] 100Cr6 contains minor variations in its material composition for the period in question.

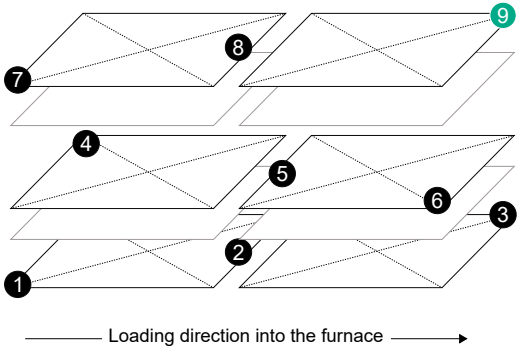


Figure 4.2: Batch positions of test specimens with number 9 being the standard specimen that is always tested

well to the complete batch. Hence, this section explores the consistent bias in hardness due to exposure to slightly different physical influences on each position.

Figure 4.2 shows the specific position of each specimen tested during an extensive routine check. From 100 batches of such nine-piece measurements, the distribution of surface and core hardness is shown in Figure 4.3. To the right, we see the mean values of each position with their respective 95 % confidence intervals (CIs) determined by bootstrapping with 4000 iterations. As seen from the box plots, a certain discrepancy exists between different positions for the mean values and variances.

The confidence intervals suggest a significant difference between test specimens' distributions, with the ones for the surface being wider due to more substantial measurement noise. For the surface, it is thus harder to achieve the same significance levels for differences between positions as for the core. Tukey's HSD test backs up this claim, as shown in Figure 4.4 by the red asterisks. Further properties of this figure are assessed in the next Section 4.2.3. These results imply that, for machine learning purposes, it is important to distinguish predictions for different positions in the batch.

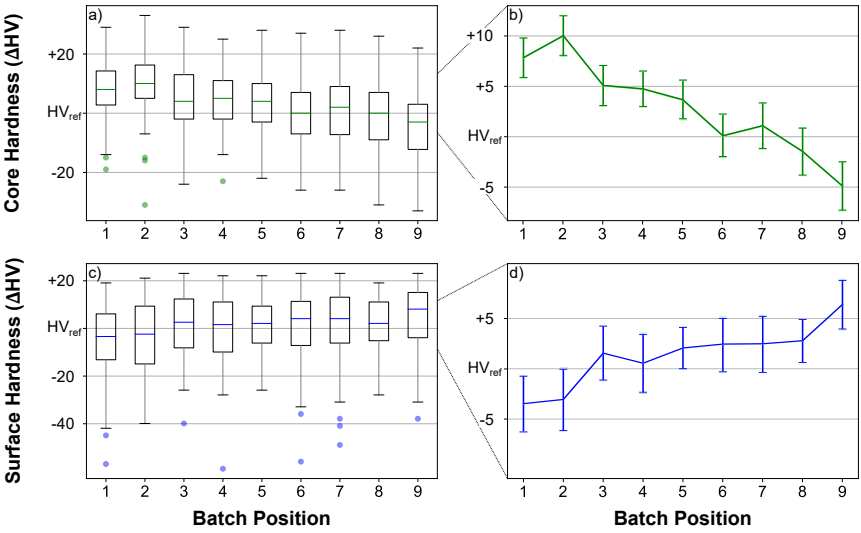


Figure 4.3: Box plot of hardness measurements for a) core and b) surface, given as difference to the respective HV_{ref}, c) and d) show their mean values' $\pm$ 95 % confidence intervals per batch position, taken from 100 batches

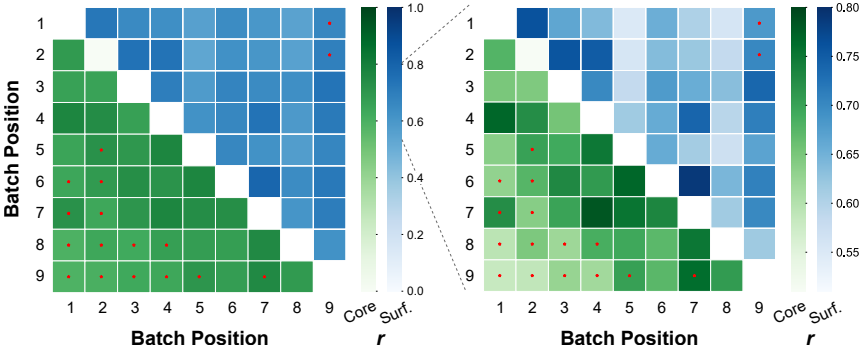
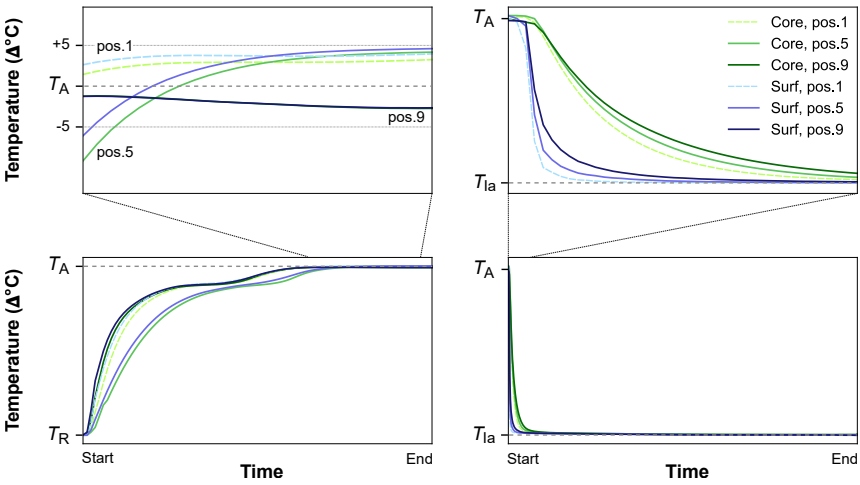


Figure 4.4: Correlation of hardness between nine test specimen positions for core and surface. Indication of significant difference between distribution means of two positions by (∗) calculated by Tukey's HSD test with $p < .01$



(a) Process furnace **(b)** Salt bath

Figure 4.5: Temperature uniformity survey for (a) process gas furnace and (b) salt bath from 35 batches. Shown is the heating from room temperature T_R to target austenitization temperature T_A and subsequent quenching to first transformation temperature T_{Ia}. The upper left graph is a zoom in on the austenitization phase while the upper right zooms in on the quenching

To understand why the hardness differs between batch layers and why surface and core hardness show inverse behavior, it is helpful to have a look at the temperature uniformity surveys (TUS). Thermocouples take the actual component temperature of positions 1, 5, and 9 on their respective surface and core, the latter by drilling into the component. Figure 4.5 shows the average temperature per position of 35 TUSs. Position 1 and 5 (bottom to middle layers of the batch) are associated with a higher temperature T_A during austenitization for core and surface. For quenching, the reverse is true. Position 9 (upper layer) is not quenched as harshly (i.e., encounters higher T_{Ia}) as lower layers since it is the last to enter the salt bath, with a clear distinction visible between core and surface temperature.

The high core hardness of position 1, thus, can be explained by the greater T_A, as more carbon is solved and fewer carbides remain, while a faster

quenching rate allows the formation of finer-grained bainite. At the component's surface, two effects counteract each other. While a higher T_A likewise facilitates carbide dissolution, it also might reduce the relative amount of carbon in the atmosphere and, thereby, lessen its capacity as a carbon donor. As expected, a GDOES analysis (Figure 4.6) shows significant differences between surface[5] and core carbon content with increasing distance from the measured surface. Surprisingly, core carbon content also seems to differ between position 1 (pos. 1) and the others. A potential explanation might be the higher T_A than pos. 9 and longer holding period than pos. 5. Both factors possibly allow carbon to diffuse deeper into the surface of the test component at pos. 1, thereby adding to the core hardness of the component. Other positions of the same cluster do not differ significantly. Consequently, pos. 9 contains the same average amount of carbon in the surface as other positions but resolves less of it due to a lower T_A which leads to a slightly higher bainite start temperature. Although a faster quenching would (for complete transformation) lead to a harder bainitic structure, the slightly slower quenching of pos. 9 allows for an earlier start of the transformation process as well as a shorter overall transformation Δt_{Ib}. Thus, upper layer components have a higher bainite to austenite ratio after the first transformation phase. With bainite formed at the first transformation temperature being harder than the bainite formed at the second transformation temperature, the final resulting surface hardness is higher. Now, that the differences between positions are established, the section below elaborates on commonalities.

[5] The lesser degree of carbon concentration at the very surface (i.e., 3 µm) is most likely due to decarburizing oxygen that enters the chamber when batches are pushed to the salt bath, having just enough time to steal away a few carbon atoms from the surface.

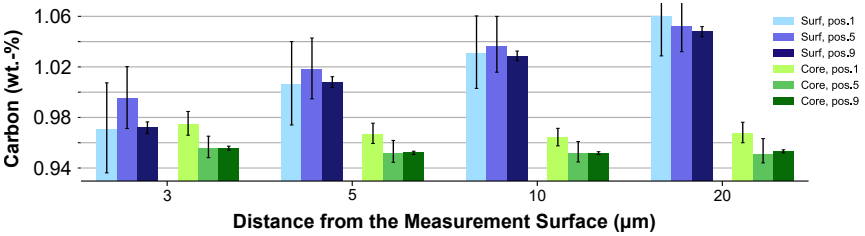


Figure 4.6: Mean carbon content for three batch positions of core and surface in increasing depths, with 99.9 % CI. Number of tested pieces per position: pos. 1 = 355, pos. 5 = 289, pos. 9 = 4958

4.2.3 Prediction benchmark from batch positions

To avoid fruitless continuation of optimization regarding the ML models developed later, this section establishes a benchmark that already provides an upper bound for the best expectable prediction accuracy. After coming close to this benchmark, any further attempts to improve the models' accuracy (e.g., bigger model, better features or subset thereof, hyperparameter tuning) are destined to fail. The benchmark is based on the relationship between batch positions. Figure 4.4 from the previous section, for example, indicates the Pearson correlation coefficient r between the nine positions by darkness of color. As might be expected, generally, the distance between two positions seems to decrease their correlation slightly. Under the assumption that two spatially close components of the same batch should show the same hardening effect, the correlations appear to be moderate. A more detailed picture is given in the Appendix A.1. To estimate the precision with which the hardness of one position can be predicted by the hardness of another position from the same batch, linear regression was used[6]. Figure 4.7 provides the resulting distribution of the RMSE as box plots including their medians as well as their 95 % CI based on a 4000-fold

[6] Related models (e.g., Huber, Lasso, Ridge, ...) lead to similar results even when using warping or quantile transformation to adjust for the skewness of the surface hardness.

bootstrapping. The same analysis using the R^2 score can be found in the Appendix A.2.

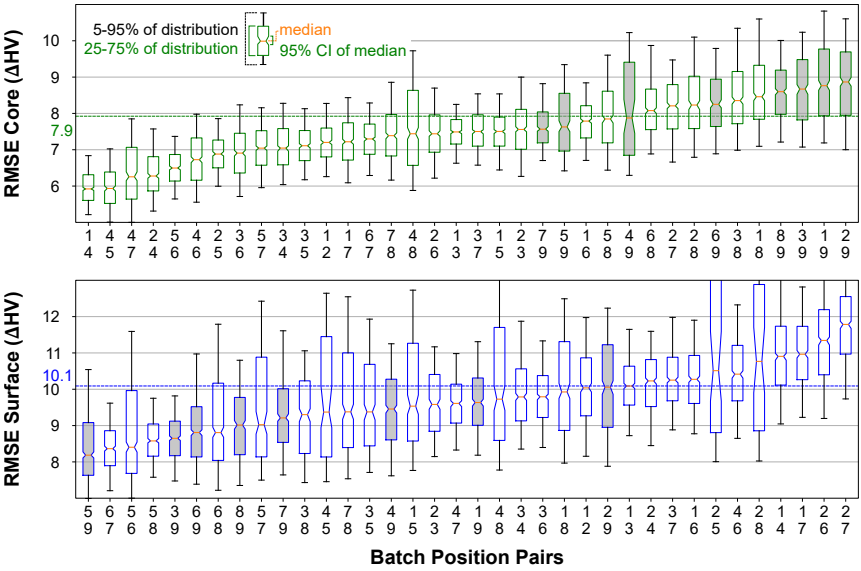


Figure 4.7: RMSE distribution by 4000-fold bootstrapping for linear regression of each position pair. Whiskers indicate the 5 and 95 percentiles of the distribution, boxes the 2nd and 3rd quartile, notches the 95 % CI of the median

This analysis suggests a significant difference in the predictability between position pairs. The Scheffe test for pairwise comparison in Appendix A.3 supports this claim. The wide distribution range between whiskers is due to the small number of 100 data points, containing some outliers that strongly influence the RMSE, depending on how often they have been drawn in a particular bootstrap. Nevertheless, some positions clearly exhibit more similar behavior for the resulting core hardness than others, which might partly be explainable by spatial proximity (e.g., 4 and 7 are close with high correlation while 9 and 1 (as well as 9 and 2) are far away with low correlation). Besides the lower correlation of

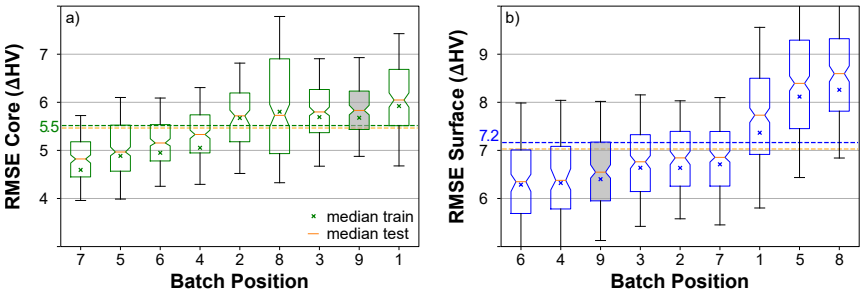


Figure 4.8: RMSE of a) core and b) surface hardness when using 8 positions as ground truth and the remaining position to predict the former by linear regression; distributions show RMSE of the test sets from Monte Carlo sampling 1000 times with train-test split of 70:30. The cross shows the median of the training set prediction

opposed position pairs between the lowest and highest level, other location patterns (e.g., front-back, left-right) are not easily distinguishable. Hence, further attempts to explain better predictability between specific pairs are omitted not to be fooled by randomness. However, from the means (i.e., $\text{RMSE}_{\text{Core}}$=7.9 HV and $\text{RMSE}_{\text{Surface}}$=10.1 HV) it is immediately clear these predictions can not suffice as a benchmark because, first, their values are partly worse than using the mean of the complete distribution as a predictor (conf. Section 4.2.1), and second, the measurement error of two parts now affects the accuracy.

A much better benchmark can be established by using the mean of 8 positions as ground truth and predicting this value with the 9^{th} position[7]. Figure 4.8 shows that there still exist significant differences between the predictive capabilities of each position to predict the whole batch[8], but the overall RMSE is much lower (i.e., $\text{RMSE}_{\text{Core}}$=5.5 HV and $\text{RMSE}_{\text{Surface}}$=7.2 HV) compared to the piecewise predictions. They are used as an approximation for the achievable predictability between test

[7] All positions have been centered around zero by subtracting their respective means, to account for the offset between positions.

[8] The mean of the remaining 8 positions.

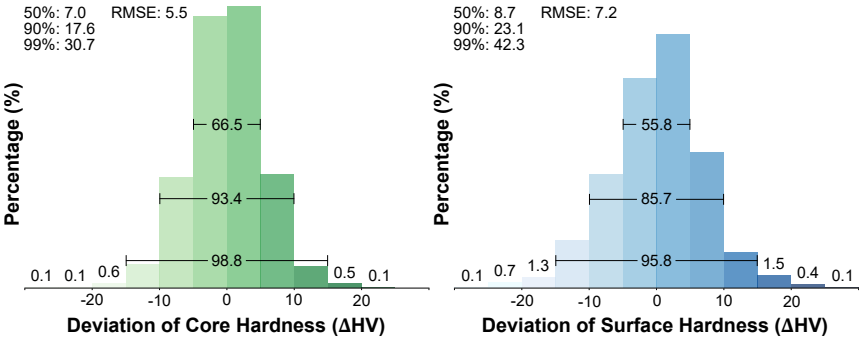


Figure 4.9: Benchmark error distribution

specimens when drawing a test specimen randomly from one position and generalizing its hardness measurement to the complete batch, resulting in an $\bar{R}^2_{\text{core}}$ of 0.65 and $\bar{R}^2_{\text{surface}}$ of 0.58.

Interestingly, there is no clear pattern in terms of generalizability from one position to the complete batch. Positions 4 and 6 distinctly show below-average errors (dashed line) for surface and core hardness, which would make them the best candidates for regular inspection in terms of generalizability to the complete batch[9]. Especially pos. 9 seems to show one of the worst generalizations for core and best for surface hardness predictions, rendering an explanation based on position somewhat implausible.

To sum up, Figure 4.9 gives the desired benchmark in the form of the accumulated test set prediction errors from all positions and shall serve as a yardstick for the ML predictions from process parameters. It assumes that predicting a component's hardness based on the measured hardness of all remaining components is the closest approximation to predictions from process parameters achievable, including the noise in the process,

[9] This might not be the preferred testing strategy which usually seeks to test the worst position in order to safeguard the complete batch.

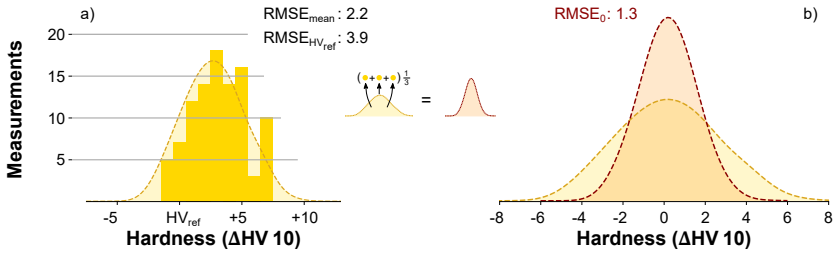


Figure 4.10: a) Histogram of 100 HV 10 indents on a standardized hardness comparison plate with $HV_{ref} = 692$ HV. The dashed line estimates the PDF using Gaussian kernels. b) yellow: PDF from a), red: PDF from repeatedly drawing 3 values and taking the mean to simulate three indents

preparation, and measurement. The following section estimates the contribution of this measurement noise to the overall variance of the hardness distribution.

4.2.4 Measurement error

While specimen preparation (e.g., cutting, embedding, polishing) has an influence on the measured outcome, here, the focus lies on the irreducible, not negligible noise from indention force and measurement of diagonals. It was evaluated by 100 HV 10 indents on a standardized comparison plate with a nominal hardness of 692 HV the result of which can be found in Figure 4.10 a). Although the PDF looks like a normal distribution, both the Shapiro-Wilk and Anderson-Darling test, reject this hypothesis. To simulate the testing practice of taking the mean of three indents (cf. Section 3.1.1), a Monte Carlo simulation was used resulting in Figure 4.10 b), illustrating the accuracy gain of repeated indents[10].

While this analysis shows that, in principle, some of the labels' inaccuracy can be attributed to the measurement procedure itself, it does not account

[10] PourAsiabi and colleagues [113], for example, used the mean value of 8 indents to improve the accuracy of their labels.

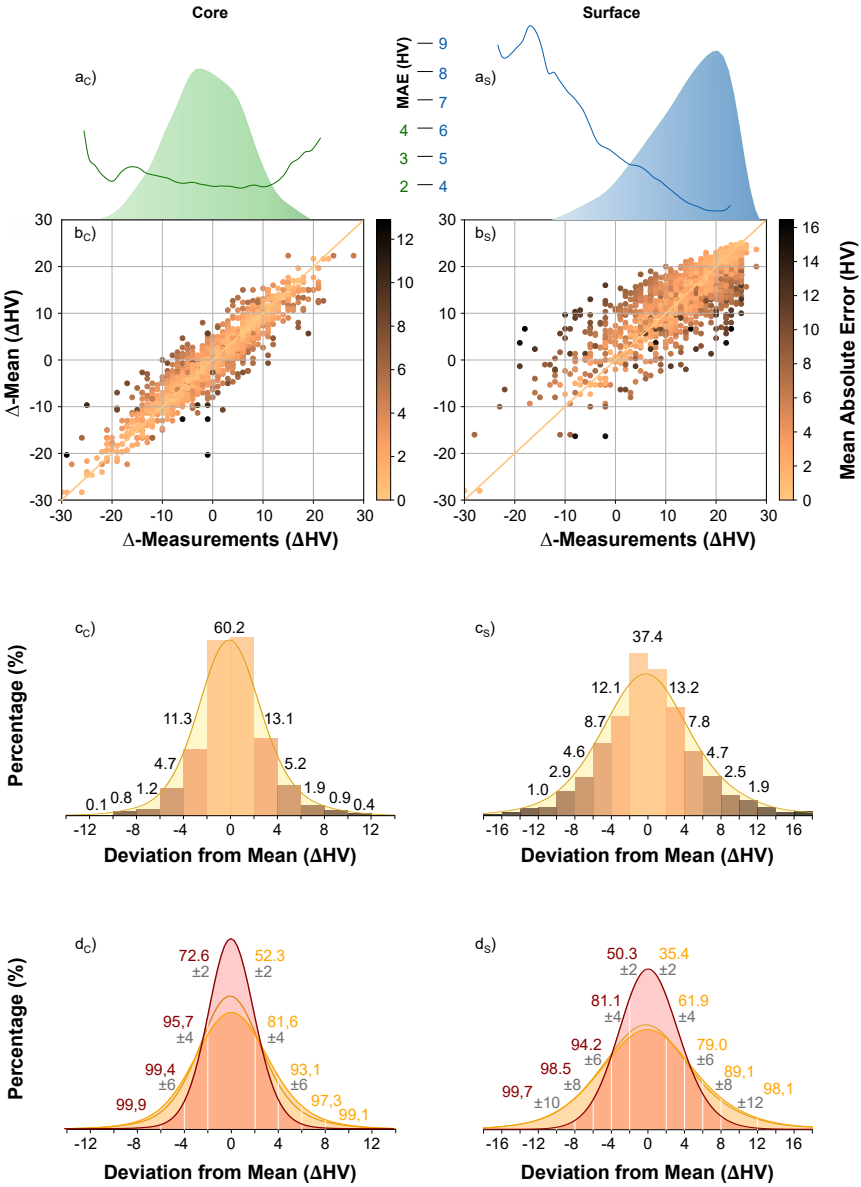


Figure 4.11: a) PDF of mean of three indents along with their MAE from the mean. b) scatter plot of means (y-axis) over their respective three indents (x-axis). Darkness of color signals greater MAE. c) histogram of the indents' deviation from their mean with color correspondence to b). d) estimated error distribution of one indent in orange and the mean of three indents in red

for additional contributors. To get a more complete picture, the original three hardness values of 565 batches were retrieved. Figure 4.11 a shows that their estimated PDF matches the distribution of the complete data set, compare Figure 4.1, indicating that the samples are representative of the overall distribution. Figure 4.11 b) then plots the three measurements (x-axis) over their mean (y-axis), where the darkness of color indicates the mean absolute error (MAE) from their respective mean. The upper figure also delineates the averaged MAE of the three indents over their actual hardness. It clearly shows that the further the mean of three indents is away from the distribution center, the larger their MAE. Consequently, data points at the distribution edge are more prone to measurement error as the underlying indents have a greater MAE. Vice versa, a greater MAE of the three indents more likely pushes their mean to the distribution edge. This phenomenon is also shown in Appendix A.4 by a Monte Carlo simulation.

The deviation from the three indents' mean is also shown as a histogram in Figure 4.11 c) which is wider than the distribution of the hardness comparison plate, suggesting that further scatter is introduced in routine testing. The MAE is much greater for the surface measurements c_S) likely due to the lower preparation effort and uneven dispersion of carbon in the surface.

As the mean of the three indents is close to but not precisely the true hardness, this error, which was determined in Figure 4.10 b), must be added to the MAE of the distribution in c). The result is shown in orange in Figure 4.10 d). It represents the closest estimate of the measurement error distribution of one indention from which now again three values are repeatedly drawn and averaged to simulate the standard procedure. It gives rise to the red distribution as the best estimate for the true error (or closest estimate) from 3 measurements.

In order to be able to estimate the R^2 loss due to measurement noise, the true hardness distribution (i.e., the distribution of the true hardness without measurement noise) must be recovered. Figure 4.12 a maps the

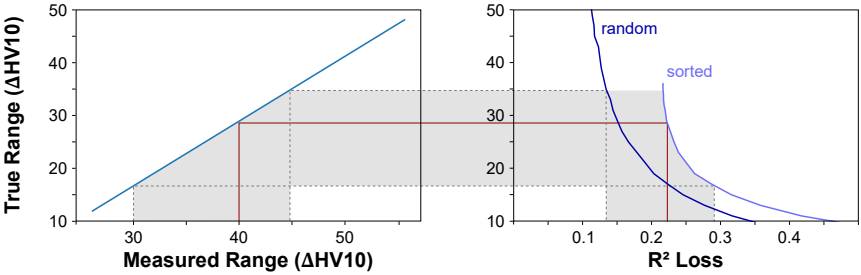


Figure 4.12: a) Map of measured distribution range to estimated true distribution range without measurement noise. b) Loss in R^2 score when applying measurement noise to the true distribution. The wider the underlying distribution, the smaller the influence of a constant measurement error. The gray area indicates the most probable range of distribution range (and resulting loss) inferred from the distributions in Figure 4.1

range (delta between the minimum and maximum) of this recovered true distribution to the measured distribution by adding[11] the red measurement noise from Figure 4.11 d) to the recovered true distribution.

If there was no measurement noise, an R^2 of linear regression between true and measured distribution would be equal to 1. That is, the R^2 loss would be zero, because those distributions would be the same. However, to calculate the loss of R^2 due to measurement noise, it is important to know how large the measurement noise is compared to the range of the distribution (cf. signal-to-noise ratio). In the following, this dependence between R^2 loss, measurement error, and distribution range is explained.

Linear regression between values of the true and measured distribution results in an R^2 score which is strongly dependent on the range (delta

[11] Adding the red error distribution to the true distribution can either happen randomly (dark blue) or sorted (light blue), Figure 4.11. Sorted means that the n values are drawn from each distribution at random, but then both arrays are sorted and added. This is done because the MAE rises with distance from the distribution mean, as shown in Figure 4.11, and so, the biggest negative and positive errors from the error distribution are added to the smallest and highest values from the true distribution. As mentioned above, values at the edges are at the edges because they most likely experienced a higher measurement error.

between the minimum and maximum) of the true and measured distribution as shown in Figure 4.12. Because the measurement error is constant, it leads to a stronger R^2 loss for narrower distributions[12]. At a range of approximately 40 HV, the core hardness distribution is subject to a loss of 0.22 (indicated by the red line) due to measurement errors, which means that the hardness prediction by measurement reaches a benchmark of R^2 = 0.78. This R^2 provides an upper limit for any prediction based on the 3-fold 10 HV measurement procedure used above. The prediction of core hardness from process parameters must lie below this benchmark and most likely is below $\bar{R}^2_{\text{core}}$ of 0.48 derived from multiple position testing in Section 4.2.3.

In sum, about 25 % to 50 % of the overall variance of the labels is explained by the measurement procedure with another 20 % due to drifts in core hardness. In the following, the same investigation is performed for measurements after case hardening. Further predictors for variance are explored in Chapter 5.

4.3 Case Hardening

In contrast to the cylinder head, both, more positions on a nozzle body, referred to as measurement positions (meas. pos., Section 4.3.1), as well as more test specimen from a single batch, referred to as batch positions (Section 4.3.2), are evaluated (cf. Section 4.3.3). Further, hardness on a nozzle body is measured by a single HV 1 indent elaborated on in Section 4.3.4.

[12] An error of 2 HV has a greater impact among values between 0 and 10 HV (20 % error) than 0 and 100 HV (5 % error).

4.3.1 Measurements on the nozzle body

To understand the data available for selected meas. pos. on the specimen, Figure 4.13 depicts their hardness distributions, ordered by increasing mean hardness from left to right and top to bottom. Distributions are composed of pieces from four batch positions, explained in detail in the upcoming Section 4.3.2. All labels in the shown histograms were corrected by their batch position medians with their offset to the overall median to avoid a variance broadening in the shown distribution.

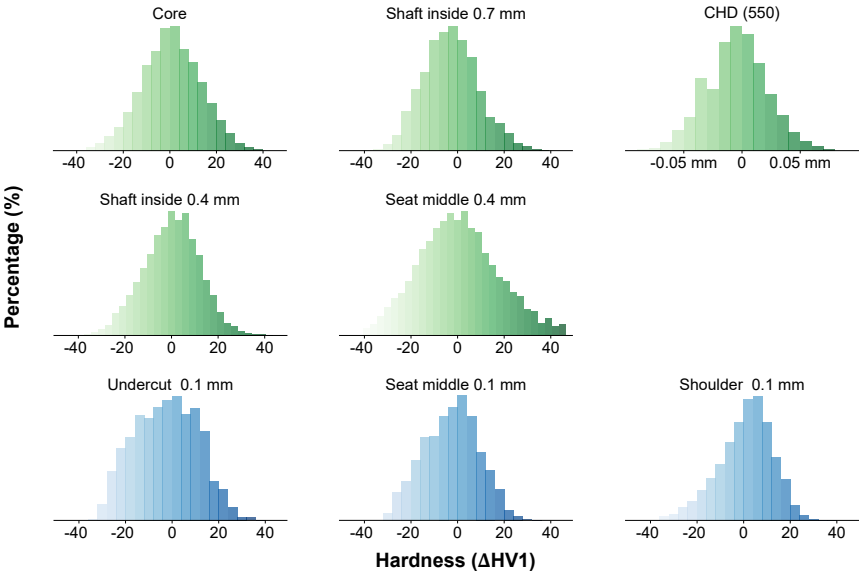


Figure 4.13: Hardness distribution of meas. pos. on the nozzle body with increasing mean hardness from left to right and top to bottom. Green indicate meas. pos. >= 0.4 mm away from the surface, blue = 0.1 mm

Most meas. pos. seem to exhibit a comparable distribution in terms of shape and variance, with notable exceptions for undercut, shoulder, and seat middle 0.4 mm. Greater variance in the undercut measurements might be due to the concave geometry of the nozzle body at this position, see

Figure 4.14. It exhibits less surface for carbon uptake, probably because it is shielded from the acetylene flow by the way the rag (that holds the nozzle bodies) is constructed, which also explains why this position is the least hard from the group measured at 0.1 mm. Similar to the surface of the cylinder head, the shoulder has a left skewed[13] distribution. It indicates that the distribution is close to the achievable hardness, which is also congruent with it being the hardest measurement point. The increased variance of seat middle at 0.4 mm might be due to several factors.

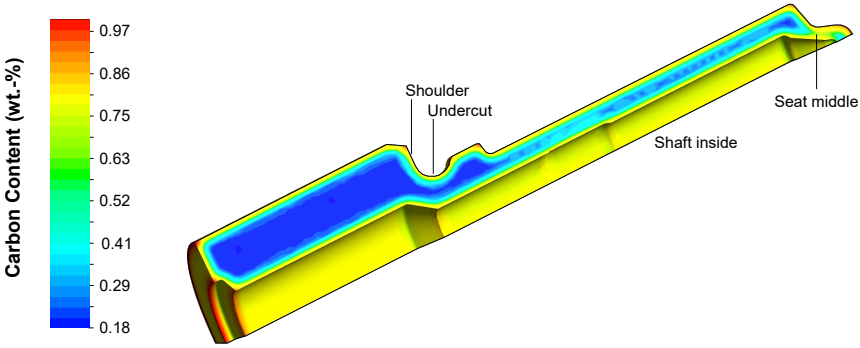


Figure 4.14: Carbon content of a nozzle body after the carburization step simulated with ANSYS CFX 18.2

First, a slight deviation from cutting the nozzle body in its very center during specimen preparation results in a significant displacement of the meas. pos., as the drill hole inside the nozzle body is relatively thin. Second, the geometry of the nozzle tip has a unique design for different customers, resulting in varied drilling holes and carbon diffusion parameters. Lastly, the thin geometry at the seat middle, see Figure 4.14, allows for carbon to also diffuse from the outside of the nozzle. It has a much higher surface-to-volume ratio and penetrates as far as to reach the carbon diffusing from the inside. It is expected that measurement points with

[13] Mean is to the left of the median, with a tail to the left.

similar carbon diffusion properties exhibit comparable hardening effects, elaborated on in the following.

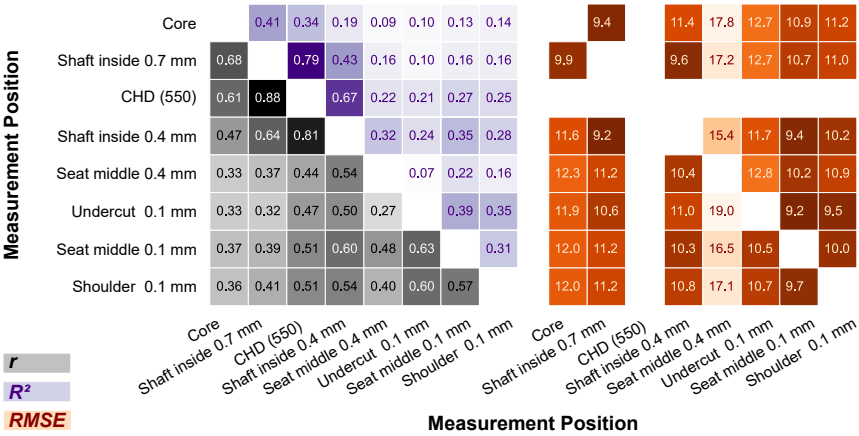


Figure 4.15: Correlation coefficient r between two measurement positions in grey. R^2 score (purple) and RMSE (red) from linear regression when predicting the hardness of one meas. pos. from another

While surface and core hardness of the cylinder head were independent of each other, the multiple measurement points of the nozzle body display clear correlations, shown in Figure 4.15. Mutual distance from the surface, expectedly, serves as a good predictor for a position pairs' r in most cases. Also, shaft inside and CHD exhibit a strong correlation, as the latter is estimated from the former, while seat middle at 0.4 mm generally shows decreased predictability, most likely due to the particular carburization behavior mentioned above. Figure 4.15 additionally points towards the nonlinear relationship between r and R^2. In this case $R^2 \neq r^2$ because R^2 was calculated from a prediction using linear regression, that is, fitting a linear regression to the data points of two measurement positions and then predicting the first position from that second and vise versa. Figure 4.15 additionally shows the RMSE from these predictions, where the prediction is always made from y-axis to x-axis. The RMSE clearly is unsymmetrical, that is, predicting meas. pos A (e.g., Seat middle 0.4 mm) from meas. pos

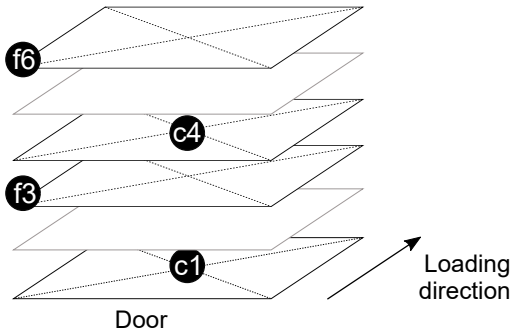


Figure 4.16: Batch position of the 4 test specimens in a case hardening batch

B (e.g., Shaft inside 0.4 mm) leads to quite different results then predicting B from A.

The $r = 0.81$ between shaft inside 0.4 mm and CHD (550) only achieves an $R^2 = 0.68$, while the $r = 0.64$ to shaft inside 0.7 mm already drops to an $R^2 = 0.43$. This seems like an astonishingly loose connection between two positions that are only 0.3 mm away from each other on the same test specimen and points towards a strong potential influence of a hardness measurement error. Although a not to be underestimated portion of scatter will be attributable to the measurement procedure, these findings already hint at the difficulties involved in precisely learning the hardness from process parameters, conducted in Chapter 6.

4.3.2 Position in the batch

After comparing meas. pos. on the same specimen, we now turn to the behavior between different batch positions. Test specimens are regularly

sampled from two alternating position pairs, including a center c1-c4 and front f3-f6 pair, shown in Figure 4.16, where numbers indicate the layer[14].

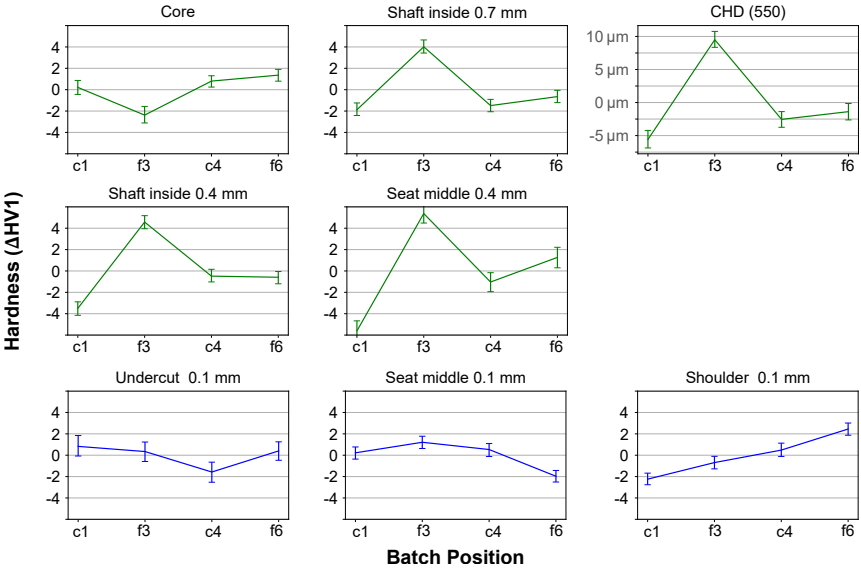


Figure 4.17: Mean values per measurement and batch position with 95 % CI, taken from 3000 batches

Since two measurement specimens are taken from every nozzle body batch, many more data points are available for different batch positions compared to the cylinder head, resulting in narrower confidence intervals for the mean of every batch position, shown in Figure 4.17. Once more, a batch position-dependent bias can be observed, especially for f3, which is significantly harder than the other green meas. pos., except for the core. Especially the already well-known meas. pos. seat middle 0.4 mm creates a substantial difference of 10 HV between c1 and f3. Since only the core

[14] Bottom most layer has number 1, as it is the first layer that is filled with nozzle bodies.

shows a decreased hardness for f3, the current hypothesis for these differences is the carbon donator distribution behavior in the vacuum furnace. Acetylene is injected through nozzles in the door and the three walls, see Appendix A.5. Under the assumption that spatial closeness to the injection increases the amount of time and gas a batch position is exhibited to the carbon donor, f3 should receive the most opportunity to absorb carbon. Although f6 also sits close to the injectors, it likely receives less acetylene since the injectors are mounted behind the heating bar[15], while c1 and c4 sit deep inside the batch. This small head start of f3 during every injection phase could be enough to diffuse significantly more carbon into deeper parts of the component, leading to an increased hardness for this batch position, except for the core. Since carbon saturation occurs at the components' surface (blue positions) during each acetylene supply phase for all batch positions, no significant difference in hardness is found there.

Surprisingly, the temperatures at the various positions seem to have a lesser impact on the resulting hardness than the carbon. The only meas. pos. readily explained by the temperature uniformity surveys in Figure 4.18 for the vacuum furnace and Figure 4.19 for deep freezing and tempering, is the core, which is least affected by carburization. Position f3, in this case, has the lowest hardness in accordance with the significantly lower quenching rate of 140 s compared to approx. 88 s for all other positions. Besides the lower austenitization temperature at c1, which might result in a slight loss of hardness, the remaining temperature conditions, except for quenching, are roughly equal for the positions or at least seem not to affect the resulting hardness consistently. Neither final tempering nor deep freezing temperature warrants a hypothetical claim to have an effect on hardness since the temperature spread between the positions in question is too small.

[15] The bar blocks the way for free gas flow and heats the injected gas significantly.

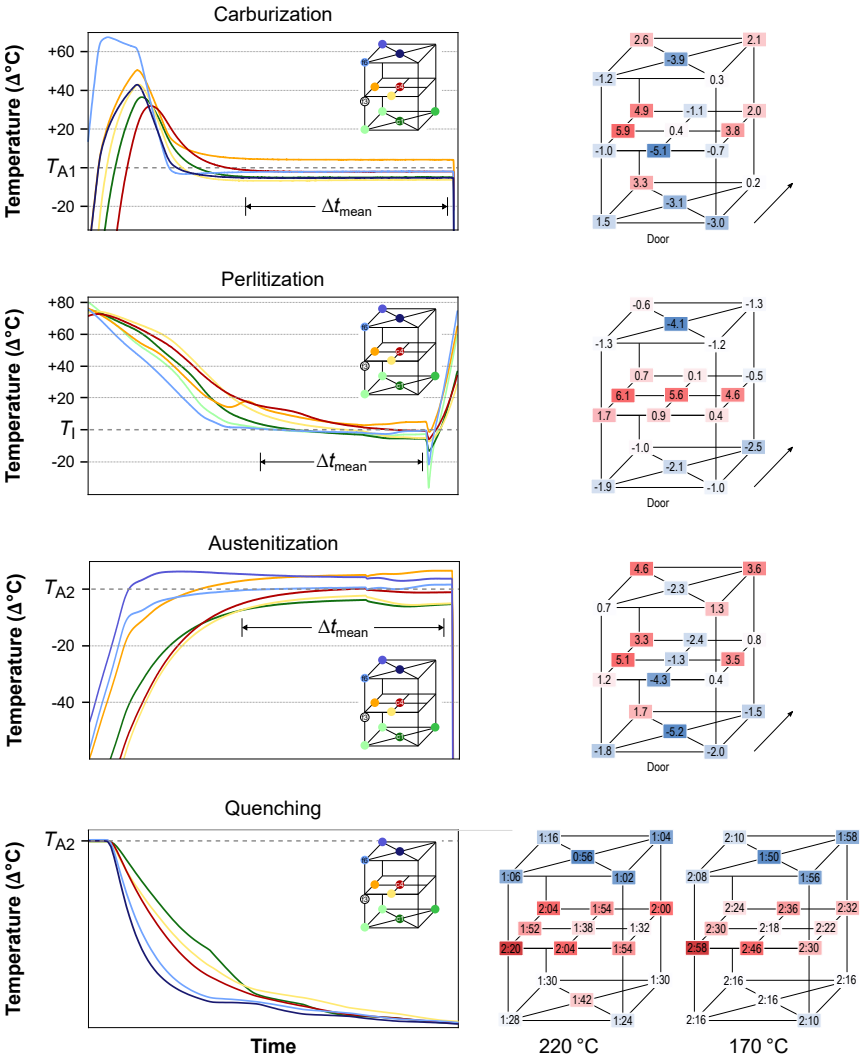


Figure 4.18: Temperature uniformity surveys of vacuum furnace. Values in the grid (right) represent mean deviation from the respective mean (over Δt_{mean}) carburization T_{A1}, perlitization T_I, and austenitization T_{A2} temperature (in °C). For quenching, the time (in m:ss) to reach 220 °C and 170 °C is given

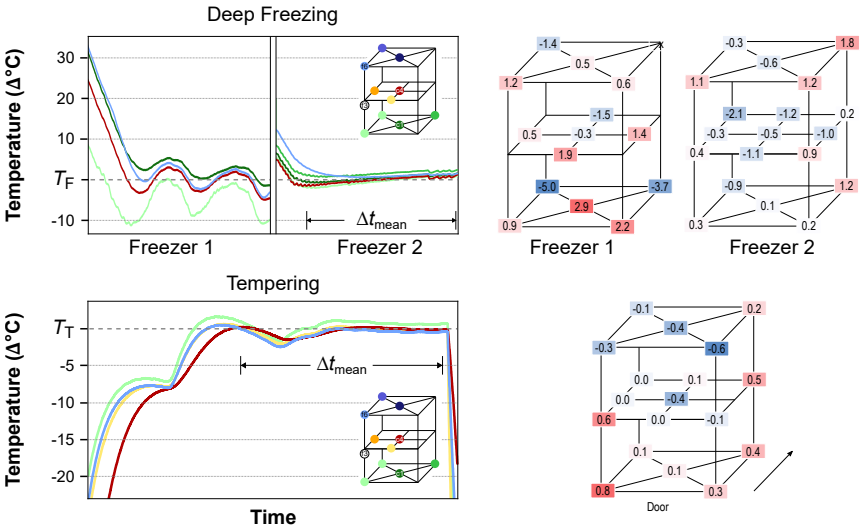


Figure 4.19: Temperature uniformity survey for two deep freezers and the three tempering furnaces. Values in the grid (right) represent the mean deviation from mean (over Δt_{mean}) freezing T_{F} and tempering T_{T} temperature (in °C) during soaking

In summary, carburization behavior and quenching rate seem to be more impactful for a batch position's hardness, at least for surface distant ones (green), than the various soaking temperatures. Near-surface differences in meas. pos. (blue) per batch position evade an immediate interpretation. Furthermore, it can be inferred from the TUS that the intra-batch temperatures scatter much stronger than the inter-batch ones, prompting the implication that nozzle bodies within the same batch might have greater variance than nozzle bodies from the same positions of consecutive batches. This variation in the measurements position behavior over time is discussed in the following.

Behavior over time

To uncover seasonal and temporal behavior of quality indicators, Figure 4.20 shows a smoothed trend of meas. pos.[16]. The curves have been shifted horizontally such that the depiction in one graph is possible, which means that the y-axis does not show the true hardness in HV but can be used to infer the fluctuation in ΔHV. The following three types of fluctuation may be distinguished: first, sharp drops or ascents (↑ 2014, ↓ mid 2015, ↓ mid 2018), second, cyclic oscillation (from mid 2015 until 2017), and last, small, fast scatter throughout the graph which is presumably primarily attributable to measurement noise.

Drops in hardness for strongly carburized positions in 2015 are most likely due to a rise of the initial[17] tempering temperature as a countermeasure for a strong undershoot of the target temperature. It would also explain why shaft inside 0.7 and core are not affected, since, for one, the thermal energy takes a long time to reach deeper layers[18] and, for another, much less distorted martensite by less carbon, which could be tempered, was formed in the first place. Changing the diamond on the indenter and recalibrating the measurement device led to a sharp drop during 2018. Figure 4.20 a also reveals a sudden decrease in standard deviation (SD) after the diamond change 2018 (i.e., $SD \approx 7\,\mathrm{HV}$ (before), $SD \approx 5\,\mathrm{HV}$ (after)) recognizable by the much closer blue horizontal line patterns. They emerge because the measurement optic is not able to dissolve more precisely. Combined with internal rounding, this leads to discretization with a step size larger than a single Vicker, leading to the line-shaped patterns in the depiction. Such maintenance and recalibration events demand that not all labels be

[16] The graph includes measurements from all four test positions in the batch, which was corrected for by shifting all values to the common mean.
[17] While entering the furnace.
[18] A change in the outside temperature affects the inside of a nozzle body much later and to a lesser degree.

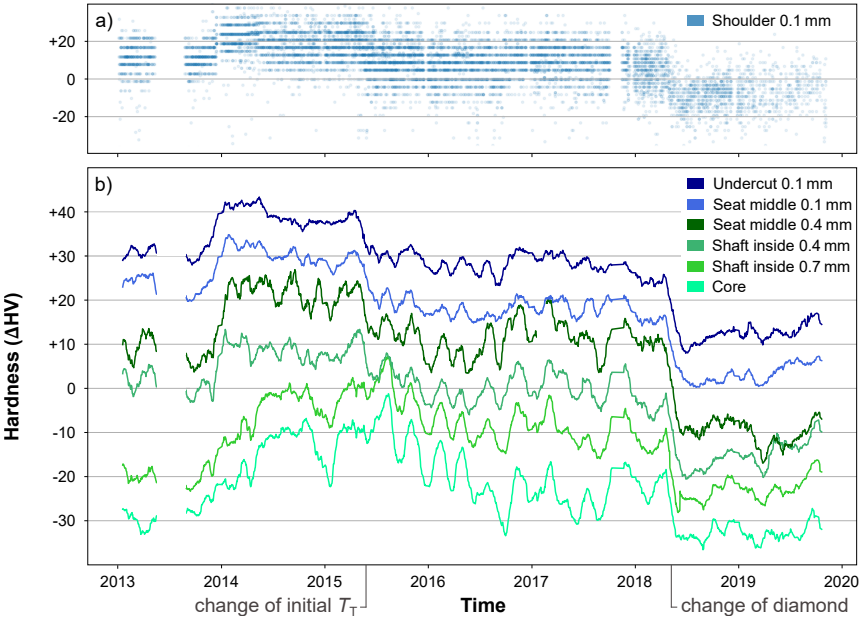


Figure 4.20: a) Single measurements on shoulder 0.1 mm, b) mean hardness of various meas. pos. over time, smoothed by a rolling window of 25 days

treated equally and must be corrected period-wise if used in a single data set.

Locating plausible hypotheses for type 2 cyclic (e.g., 2016) fluctuations is more challenging, as no single event can be held accountable. The supply chain is proposed as main factor as steel composition, extrusion production, and annealing may experience changes over time. A comparison between blue (i.e., near-surface, carburized meas. pos.) and green (i.e., surface remote positions) lines, especially in the period around 2016, speaks for the steel composition hypothesis, as the blue graphs show much less fluctuation due to the carburization while the hardenability of the green graphs, ceteris paribus, is dependent solely on steel composition. Chapter 5.3.1 will further substantiate this hypothesis. Plant maintenance can

most likely be ruled out as inspections occur at different times for each furnace, and the fluctuation does not change much when looking at machines individually.

Overall, the measurement points move remarkably uniformly. The parallel motion is reflected in the correlation described earlier, where the sharp drop in 2018 indeed adds to that effect. Thus, part of the correlation does not stem from physically similar behavior between positions but a recalibration of the measurement device. The following section seeks to quantify this relationship between batch and meas. pos. in more detail.

4.3.3 Prediction benchmark from batch positions

Similar to the cylinder head, LR was used repeatedly to predict the two positions in each position pair c1,4 and f3,6 from each other in a 1000-fold bootstrapping, the result of which can be found in Figure 4.21.

While the variances of R^2 distributions are much narrower because over 3000 batches were evaluated, their means are also widespread. Generally, the correlation should decrease for greater hardness, as can be seen in Figure 4.21, from left to right, because measurement precision wears off (cf., upcoming chapter). The three outliers are explained as follows: The core hardness exhibits worse predictability than most other positions because it does not enjoy the mitigating effects of carburization. Forming the lower end, seat middle 0.4 mm prominently sits around $R^2 = 0.19$, while the undercut averages at $R^2 = 0.50$ marking the upper end. Low predictability of the former is readily explained by the great variance in degree of carburization[19] as well as low repeatability[20] in preparation and indention. It is more difficult to explain the high R^2 of the latter. Its distribution (cf.,

[19] Carbon diffuses from inside and outside.

[20] The very small, round geometry makes it more difficult to cut precisely in the middle and, thereby, often leads to an offset of the indent.

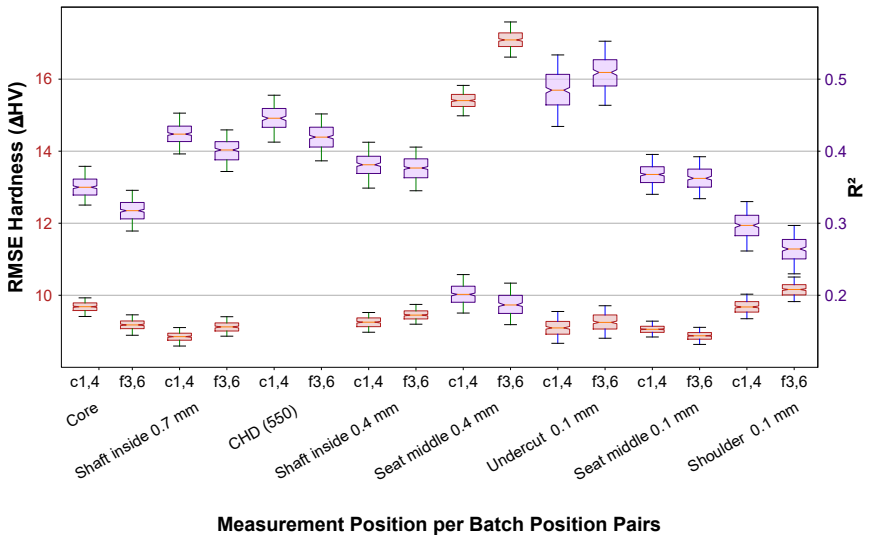


Figure 4.21: Estimated R^2 score distribution (purple, right y-axis) for nozzle bodies by 1000-fold bootstrapping for each position pair. Respective RMSE distribution in red with y-axis on the left. Whiskers indicate the 5 and 95 percentiles of the distribution, boxes the 2nd and 3rd quartile, notches the 95 % CI of the median

4.13) has a negative kurtosis[21] leading to a more uniform distribution with more values at the edges. Such a property can enhance the R^2 because the total sum of squares becomes larger.

In most cases, the centerpieces can be predicted slightly but significantly better from each other than the two front test specimens. Although it is clearly visible that predictability depends rather on measured position on the specimens than position pair of test specimens in the batch. Consequently, a benchmark for predictability was inferred for each meas. pos. individually as the mean R^2 score of each meas. pos. These benchmarks are strongly dependent on the precision with which hardness is determined. An assessment of HV 1 measurement precision is given below.

[21] It is more round (less peaky) than a normal distribution.

4.3.4 Measurement error

The diagonals of an HV 1 indent on a hardness comparison plate of about 780 HV are about 48.7 µm long. According to Formula 4.1, subtracting only 0.1 µm leads to a decrease of 3 HV indicating, that a precise hardness measurement necessitates a sharp diamond and a high focus measurement optic [102].

$$HV = c\,\frac{F}{d^2} \quad \text{, with } c = 0.1891 \text{ and } F = 9.806\,65\,\text{N} \tag{4.1}$$

By repeating the measurement of the same diagonals (d_1,d_2) of one HV 1 indent 50 times, an estimate of the measurement's optical precision can be gained. The smallest 8 % of values measured on a hardness comparison plate of 780 HV were $\bar{d}_{min} = 48.6\,\mu\text{m} \mathrel{\widehat{=}} 776\,\text{HV}$ and the largest 8 % $\bar{d}_{max} = 48.9\,\mu\text{m} \mathrel{\widehat{=}} 785\,\text{HV}$. Thus, the optic alone introduces a spread of around 9 HV with a resolution not allowing to resolve single HVs but only about 3 HV for this hardness.

To investigate the error between measurements, one hundred HV 1 indents were made on a standardized hardness comparison plate of 710 HV. The yellow histogram in Figure 4.22 a shows the result of these measurements, which approximately fit a triangular distribution. Under the assumption that the true hardness distribution (red) after heat treatment also has a triangular shape, Figure 4.22 b) shows the resulting distribution (orange) when applying the measurement error to the original distribution.

This result closely resembles the distribution shapes of Figure 4.13. On this basis, the R^2-loss resulting from the measurement error can be estimated. Figure 4.22 c) maps the range of the measured distribution (x-axis) to the estimated true distribution range (y-axis) when subtracting the measurement error. Figure 4.22 d) then shows the reduction of R^2 in dependency of the range of the distribution the error was applied on. Obviously, the

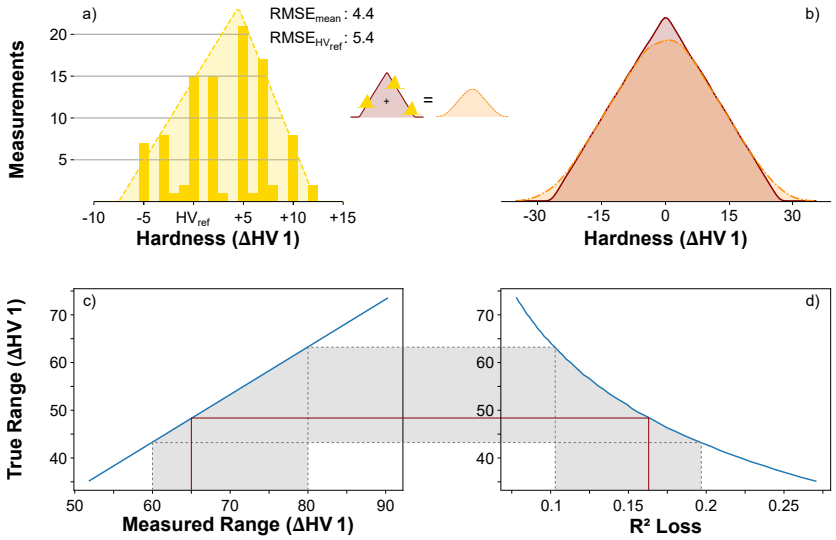


Figure 4.22: a) Histogram of 100 HV 1 indents on a hardness comparison plate of 710 HV with estimated PDF (dashed line). b) red: assumed true hardness distribution, orange: PDF after adding the yellow measurement noise to the red distribution. c) Map of measured distribution range to estimated true distribution range without measurement noise. d) Loss in R^2 score when applying measurement noise to the true distribution

larger the range of the true distribution, the more insignificant the influence of the same measurement error and, hence, the smaller the loss of R^2. As most measured distributions in Figure 4.13 have a range of about 65 HV, the red line indicates that on sole evaluation of the measurement error the best prediction can never exceed $R^2 = 1 - 0.16 = 0.84$. After correction for the recalibration that artificially widens the distribution around 15 HV, the influence becomes much larger and drops the achievable R^2 to 0.70. The difference between this irreducible error and the benchmarks derived earlier points out the significance of precise specimen preparation.

4.4 Discussion

For label analysis in general, as we have seen, it is not enough to know the distribution of the variable under examination but also necessary to investigate its dynamic behavior over time. Not only does it reveal drifts and fluctuations, but it can also indicate discontinuities in the way measurements were obtained, whether it is due to a diamond change, a new measurement device, a change of supplier, or a changed quality of the supplied. This dynamic portion is indispensable for a holistic understanding of the partial variations that make up the label distribution. In addition, a gap-free data collection, including a log of all changes made with regards to recording and process, is imperative to enable proper post hoc explanations. Such information can then be used, for example, to explain the relationship between the influence of local temperature differences on the consistent hardness offsets between batch positions. These discrepancies are to be expected from most kinds of batch processes.

Several sets of labeled data should be collected, if possible, to capture the variance contributions due to measurement errors (including sample preparation, instrument, and measurement optics) and derive a prediction benchmark that cannot be exceeded based on the process used to obtain the label. This approach allows a very early assessment of the achievable predictability and thus which economic benefit can be maximally achieved. Such data sets may also point out how the trustworthiness of hardness measurement drops with increasing distance from the expected values. In addition, it is strongly recommended to regularly check the measuring devices (especially for hardness) to obtain comparable and usable data.

Finally, it was shown that it is not sufficient to rely upon one statistical measure to quantify prediction capacity. Mean values, RMSE, and R^2 with respective confidence intervals are good indicators but might not tell the full truth (e.g., mean of unevenly split categories) of a specific research question.

In sum, a data scientist analyzing the labels belonging to a new heat treatment process is advised to do the following: Analyze the dynamic label behavior, including an explanation of drifts, discontinuities, and changes in variance over time by use of rolling windows or other filters. Assess the influence of various batch positions to understand the hardness distribution in different locations as well as derive a benchmark for their predictability. Evaluate the accuracy of the measurement procedure (incl. indentation, optical resolution, and specimen preparation) to find the limitations of possible predictions and the meaningfulness of single measurements.

As always, the interpretation of the diagrams should be made with great caution, which means that in the case of minor deviations or contradictory results, premature conclusions should be avoided, as unmeasured and/or unknown influences may well distort results. The remaining known and measured influences, including different furnaces, routes, process parameters, and alarms, are explored in the next chapter.

5 Feature Analysis

5.1 Introduction

While some of the variances in the labels could already be attributed to the measurement error, this chapter seeks to explain as much of the remaining variance as possible by analyzing the various properties of the two heat treatment processes. These, in turn, may later serve as features for the ML algorithm if a sufficient physical explanation for their suitability as predictors could be established. Expert knowledge thus plays a role in feature selection that should not be underestimated.

Changes in chemical composition are uncovered in the first Section *Material* for the 100Cr6 (5.2.1) and 18CrNi8 (5.3.1), explaining much of the larger long-term fluctuations in the label. As not all production lines are built equal, Section 5.2.2 takes a closer look at the individual lines for bainitization, while 5.3.2 examines the different stations of the case hardening process as well as the routes a nozzle body batch can take through these stations. Further, meta feature analyses of alarms and component types are provided in 5.2.3 and 5.3.3, respectively. In each of the final sections, we address the features extracted from the sensor signals in order to show how much variance is actually caused by variations in the process of bainitization 5.2.4 and case hardening 5.3.4 itself. Moreover, we encounter the difficulty of selecting good features for the right reasons, since temporal changes in one feature may cancel out the effect of another or result in spurious correlations.

5.2 Bainitizing

Before the results of the feature examination are discussed, the following paragraphs set the scope detailing which analysis will and will not be included. Previous chapters examined test specimens from all batch positions. But since position 9 is the only component tested after every produced batch (i.e., standard specimen) this chapter will, without loss of generality, focus on these specimens, as much more data is available and predictions will be made for this test position, exclusively.

Although the following investigations focus on the core hardness of the cylinder head as a methodical demonstration, the analytical approach for the surface hardness is generally the same. However, the latter exhibits three undesirable characteristics that make analysis more difficult. First, the measurement error is increased at the surface due to the inferior preparation and higher hardness (smaller indent). Second, the surface is exposed to a fluctuating furnace atmosphere, the composition of which currently is not measured precisely but is known to change the surface carbon content of the cylinder heads and, thereby, influences its hardness. Third, the controlled gas flow in the furnace atmosphere also differs for the various components produced in a particular line. These components themselves absorb different amounts of carbon and, thus, may leave behind different amounts of carbon in the furnace. While the hypothesis that a cylinder head batch is decarburized if it follows a batch of components with a lower enrichment gas flow process must currently be rejected based on the data available, future research with more sensors may investigate such questions in more depth. Lastly, geometric variations between the cylinder head types are only of small magnitude and have no significant influence on the hardness at the measurement positions, which is why no deep dive in this matter is provided. However, the following section focuses on material composition rather than geometry.

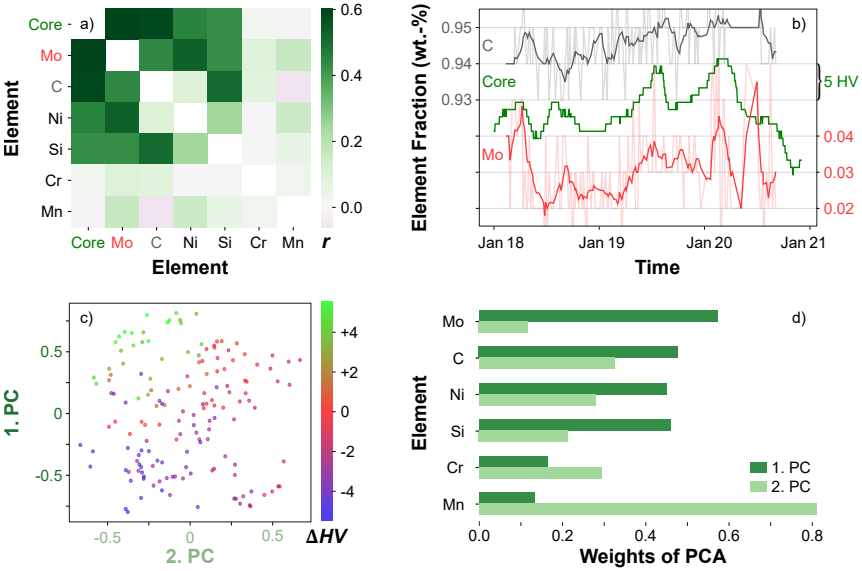


Figure 5.1: a) Correlation between weight percentages of elements in 100Cr6 and core hardness, b) share of carbon (C) and molybdenum (Mo) in 100Cr6 along with core hardness over time, c) scatter plot of 1^{st} and 2^{nd} principal component (PC) of chemical composition with color indicating respective hardness, d) weights of 1^{st} and 2^{nd} PC with explained variance of 38 % and 18 %, respectively

5.2.1 Material

As mentioned in Section 4.2.1, a sufficient change in the chemical composition of the bearing steel 100Cr6 might influence its hardenability and cause significant fluctuations. This section investigates these changes and their consequences for the bainitization of the cylinder head's core hardness, as well as predictability of hardness from the chemical composition.

A steel melt usually is obtained from a mixture of particular ores and scrap metal, each containing specific proportions of chemical elements (hereafter referred to only as elements). Naturally, their shares in consecutive steel

melts are, except for manganese (Mn), all positively[1] correlated, as shown in Figure 5.1 a), containing a selection of elements affecting hardenability. Molybdenum (Mo) and carbon (C) exhibit a high r suggesting a particularly strong positive influence on resulting hardness, also visible in their partial parallel movement to the core hardness in Figure 5.1 b). In accordance with the literature [99], both are, in fact, generally hardness enhancing. Evidently, they are not the only factors, as indicated by the Mo spike in mid 2020 which is not accompanied by a corresponding peak in core hardness. Chromium (Cr), for example, is known to increase hardenability as well but only has a small r. Yet, it must not be concluded that it has no influence in general, but only that its weight fraction is kept relatively constant over time, as required by high steel quality specifications.

To investigate whether element concentration indeed may be associated with core hardness variation, Figure 5.1 d) shows a PCA with two principal components (PC). While the weights of the 1^{st} PC closely resemble the correlation coefficients to the core hardness, the 2^{nd} PC mainly contains the uncorrelated Mn. As can be seen in the scatter plot 5.1 c), the PCA manages quite well to separate steel batches resulting in greater hardness, colored in green, from the less hard, colored blue (without any knowledge of the target). Clusters mainly form due to horizontal separation (1^{st} PC), although the 2^{nd} also helps to carve out some of the red points of medium hardness. Based on these observations, the fluctuation in core hardness might be partially attributable to changes in chemical composition. To gather further evidence for such a claim, this fluctuation shall be predicted by the elements using ML methods.

The ML pipeline used for predictions was optimized by TPOT, consisting of polynomial feature transformation, an AdaBoost regressor, and Least Angle Regression, detailed in Section 3.3.3. As can be inferred from Figure 5.2, the way in which the data is split into training and test sets makes a

[1] The addition of one element to the melt by scrap metal commonly involves the addition of other elements with a proportional share.

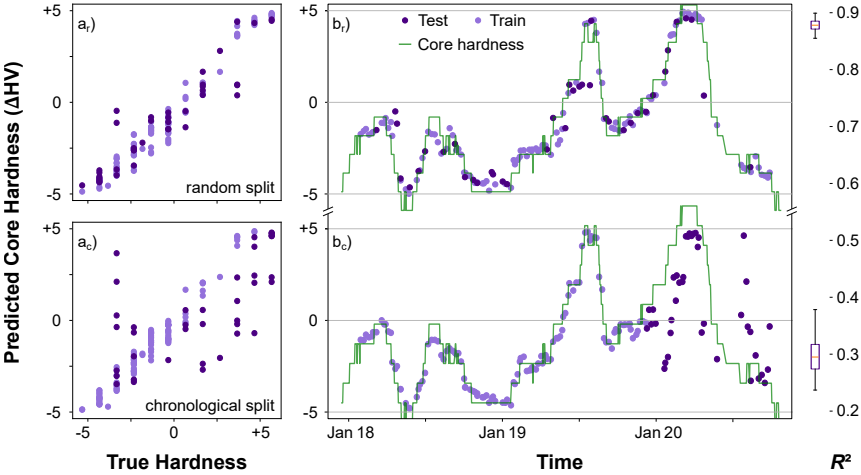


Figure 5.2: Prediction of median core hardness by splitting test and train randomly a_r), b_r) and chronologically a_c), b_c): a) scatter plot of predicted to true hardness; b) same predictions over time along with median core hardness; right) box plot of R^2 score of the respective test sets from 200 trainings

profound difference. While predicting randomly sampled training points is satisfactory, future periods from which no training data is given to the algorithm can only be predicted with much less accuracy, also indicated by the R^2 box plots. Results from this chronological split reveal the impracticality of an ML approach for the precise forecast of the drift from alloy composition because the remaining drift factors are not yet known. Moreover, it shows that the confidence in the predictions of an ML pipeline should depend on the proximity of the train-test split to the real-world use case to prevent bias in the results due to information leakage[2]. Generally, it is difficult for regression models to extrapolate if the true underlying system structure can not be derived from the data or is even physically unknown (i.e., no scientific model exists for the phenomenon). If physical

[2] In case of a random split, information about "future" leaks into the training set.

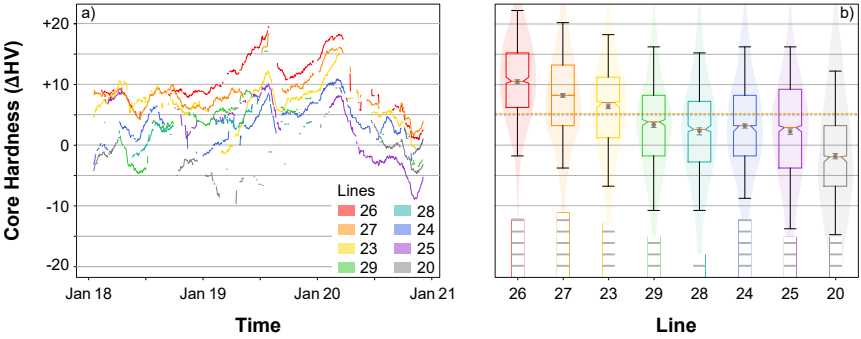


Figure 5.3: a) mean core hardness over time per line smoothed by a centered rolling window of 31 days (±15 d). Lines are only[3]shown for production days. b) box plot of core hardness with median and its 99 % CI. Dashed lines show median (orange) and mean (gray). Columns at the bottom show number of batches produced per line

properties of the system are known the design of hybrid models can advance the extrapolation capabilities [11]. The final prediction model must, nevertheless, get information about the drift to make meaningful predictions. Because the chemical composition does not suffice to do so, Section 6.2.1 will introduce filters to track these drifts. The following sections will analyze further factors leading to differences in core hardness over time, starting with the production lines.

5.2.2 Production line

Although the data was produced by lines similar in construction, slight variations are expected due to, e.g., sensor placement, isolation from environmental influences, or maintenance cycles that impact the heat treatment process. To capture varied output originating from such line dissimilarities, their individual produced mean core hardness over time as well as a box plot are shown in Figure 5.3. A similar analysis for the surface can be found in the Appendix A.6.

The following three observations are immediately apparent: First, lines do consistently produce cylinder heads with significantly different hardness but mostly possess similar variance. Second, the lines experience a collective drift, moving up and down in parallel. Third, the number of cylinder head batches processed during the shown period differs visibly - see the columns at the bottom of Figure 5.3 b). Reasons for the offset between lines include varying age, slight variation in process settings, overhaul, and divergence in construction. While the influence of process settings will be assessed in Section 5.2.4, the individual characteristic of each line may not be separable in more detail. As the offset between lines can not be predicted and is not constant over time, fluctuations must be tracked individually. It also means that a desired reduction in test parts and a replacement by prediction can only go so far, as the fluctuation is still recoverable from the remaining test parts. Periodic fluctuations will be discussed in the next section alongside additional (meta) information that might prove valuable.

5.2.3 Metadata

Seasonality

Frequent changes in production circumstances like outside temperature fluctuation over year and day or decreased capacity utilization of lines on weekends might lead to cyclic hardness variation. Figure 5.4 a) shows an autocorrelation analysis where an array of the mean core hardness of each day is correlated with an array where these values were shifted by d days (lag in days) with the turquoise shaded area indicating insignificant correlation. The representation is limited to one month since the analysis yields little evidence for repeated behavior over longer (i.e., months or

[3] A rolling window would also provide values for a day with no production using the days before and/or behind the current day. In this case, values are set to NaN.

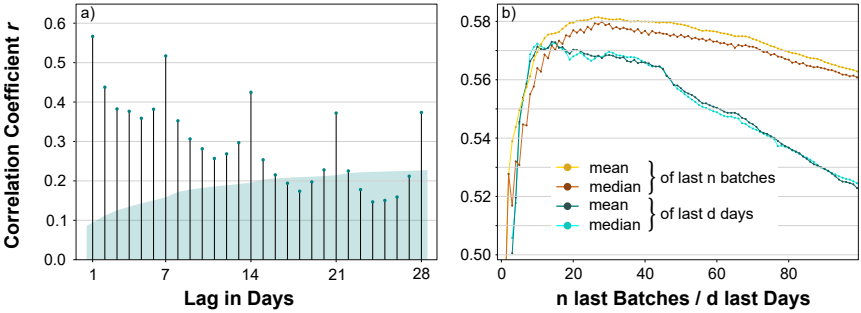


Figure 5.4: a) autocorrelation of core hardness per day with the same array of mean hardness values shifted by d days (lag in days), b) correlation of today's hardness with mean hardness of last n batches or d days (implemented with a rolling window)

years) or shorter (i.e., hours) periods. The hypothesis that summer-winter and day-night outside temperature fluctuations alter production outcome could be rejected on this account. For a lag of one day, r is equal to .57, implying that the average mean hardness of today may be a good estimate for the hardness of tomorrow (or yesterday). This claim is supported by Figure 5.4 b) that shows the correlation between today's hardness and the last n batches (d days respectively). The mean hardness of the last 20 batches produced on one line gives a good prediction about the upcoming batches.

Figure 5.4 a) also shows weekly spikes, which likely stem from batches produced on Saturdays which exhibit a significant drop in hardness, see Figure 5.5 a). Production hardness does not change over the course of a day (with a very slight exception for 15 o'clock batches), see b). A further investigation shall reveal the reason for this behavior as it can not be assumed that a Saturday (or the afternoon) in and of itself is causing a diminished core hardness rather than potential temporary irregularities, which are usually captured by alarms.

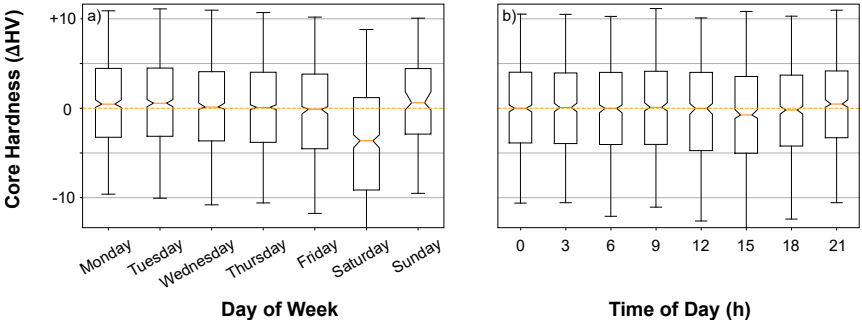


Figure 5.5: Box plot of core hardness from batches with production start on given: a) day of the week and b) time of the day

Alarms

To use alarms as features, a causal relationship must exist between a triggered alarm and the cause for this alarm leading to a change in output quality (i.e., core hardness). This fact is spelled out explicitly since the mere comparison of average hardness between batches produced, including a specific alarm and those without, would lead to false inferences. To make this point less abstract, two examples shall be given: If alarm A would be more frequently exhibited by a line that generally produces softer parts (e.g., line 20), then, in overall comparison, batches produced including A would exhibit a lower than average hardness. If the cause for alarm A is not also causing a lower hardness, a misleading feature would be introduced. This can be circumvented by either a) looking at the alarms for each furnace individually but, thereby, losing explanatory power as the data is split up and reduced, or by b) correcting the hardness values for each line such that the median of each line is equal to the overall median. The second example concerns the hardness fluctuation over time. If alarm A coincides with a time interval of greater hardness (e.g., March 2020), maybe because of a defect relief valve, then, even after the previous corrective measures, this alarm would be associated with a greater average hardness even though the two are not causally related. To reduce the

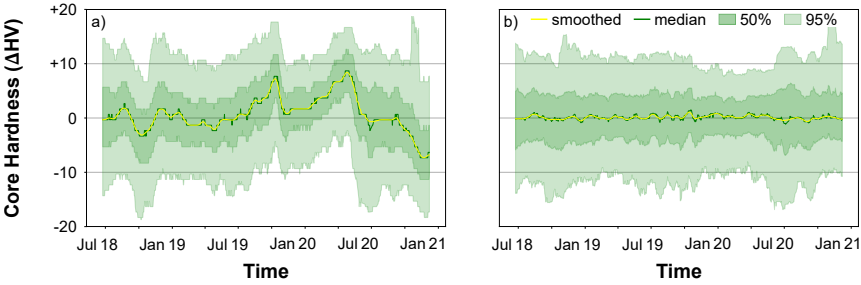


Figure 5.6: Core hardness after a) correction for furnace offset and subsequent b) time drift correction using the yellow line from a)

probability for this attribution error, all hardness values can be corrected by the joint fluctuation that occurs for all lines over time.

In sum, the hardness values of each line are corrected by their offset to the common mean (correcting for line difference) resulting in Figure 5.6 a); then the joint fluctuation (yellow line) is subtracted from each hardness value (correcting for trends), resulting in 5.6 b). After this correction, alarm data can more safely be examined for all furnaces at once. Unfortunately, this correction nullifies alarms that actually occur for all furnaces at once and lead to a changed hardness (e.g., alarms concerning the overall process gas supply system). Such events must be examined before, although the probability for this kind of event is quite minimal.

The complete set of alarm types lies well beyond 200, which is why a selection of alarms is analyzed here to showcase their properties. To include a particular alarm A as a feature, it must occur often enough to make statistically reliable assertions. Furthermore, its occurrence must be part of a causal chain leading to significantly increased or decreased labels, as compared to the baseline, examples of which are shown in Figure 5.7.

In the following, alarms are explained from left to right, sorted by associated median core hardness, along with their occurrence rate and Scheffe's

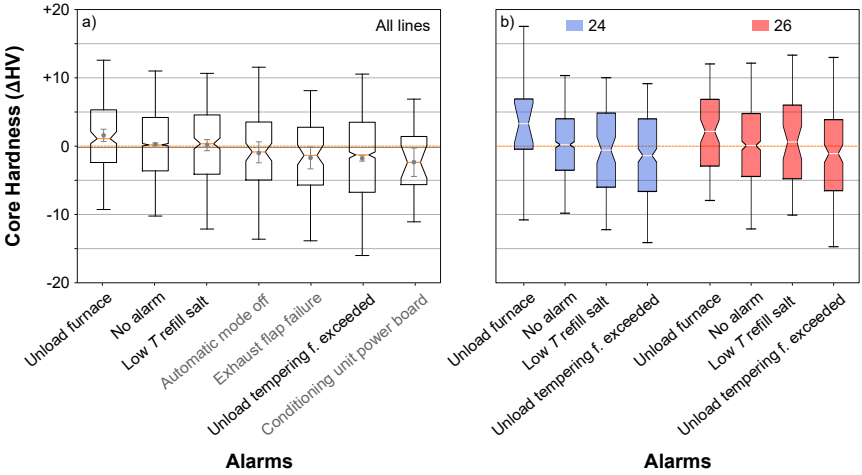


Figure 5.7: Box plot showing core hardness of batches for which a certain alarm occurred along with overall median (orange dashed) and mean (gray dashed) for a) all lines and b) lines 24 and 26

p-value for pairwise comparison[4] to the baseline (no alarm). The box plot suggests that $A_{\text{Unload furnace}}$ (3 %, $p = .003$.) leads to increased core hardness which might be explained by prolonged austenitization, as the batch is not pushed through the furnace to the salt bath in due time. A batch with no alarms tends to have, on average, only a slightly higher hardness than one with at least one alarm, indicating that the cause for most alarms does not affect production outcome in the short term. Neither the temperature of the container from which salt is refilled, $A_{\text{Low T refill}}$ (4 %, $p = .99$), nor switching off the automatic mode, $A_{\text{Automatic mode off}}$ (1 %, $p < .15$), seem to have detrimental effects on the batch. The latter gives some credit to the machine operators who seem to act fast enough with a manual override when a problem occurs.

Process interference by $A_{\text{Exhaust flap failure}}$ (1 %, $p = .008$) seems very unlikely from a physical perspective and the observed significant hardness

[4] A significant difference between alarm A and no alarm is assumed for $p < .01$

loss might well be a false positive. The most frequent alarm $A_{\text{Unload tempering exceeded}}(20\,\%, < .001)$, indicating that the batch has been in the tempering furnace longer than desired, explains the aforementioned hardness drop on Saturdays. This elongated tempering furnace dwell time of a batch will be elaborated on in the upcoming section. Diminished hardness occurring with $A_{\text{Conditioning unit power board}}(0.5\,\%, < .07)$ (i.e., the cooling unit of the power control board is malfunctioning) is most likely due to electrical signal artifacts produced by the overheated control board which might entail any number of problems. As this alarm was only found in line 29, it might be necessary to examine every produced batch when such an alarm is triggered.

Figure 5.7 b) confirms that the offset between furnaces was removed successfully[5]. It also exemplifies that, although it is possible to analyze lines individually, fewer data generally lead to larger confidence intervals which might make detection of significant differences per line and alarm (e.g., $A_{\text{Unload tempering exceeded}}$ for line 26) difficult.

In conclusion, the decision of which alarms to include as features is foremost based on the analysis above. If no significant influence is observed, the feature is most likely not included. If an alarm shows significance but can be explained by a process feature, the latter is given precedence because it more precisely pinpoints the problem. Such process features are examined in the next section.

5.2.4 Sensor signals

Intuitively, differences in hardness between batches are predominantly attributed to a change in the heat treatment procedure, which is captured by

[5] Otherwise, all values of furnace 26 would have been much higher.

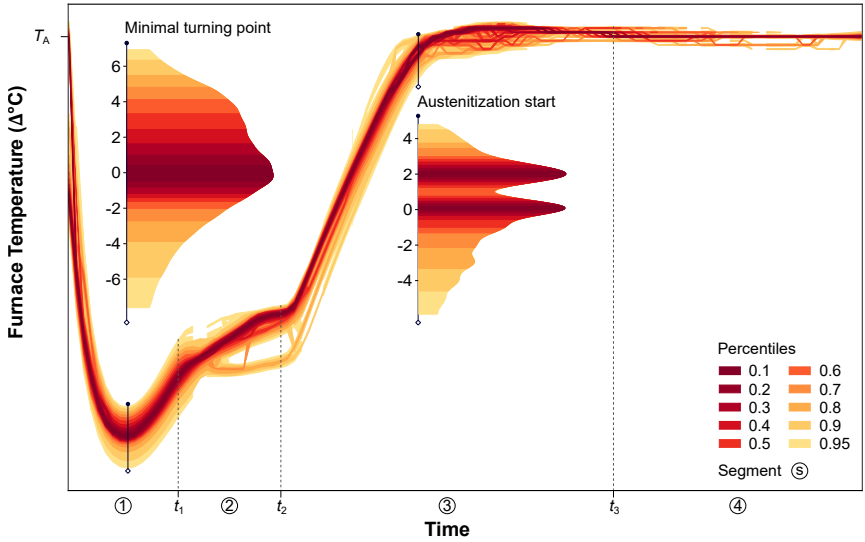


Figure 5.8: Kernel density estimation of furnace temperature curves at each minute from $\sim$ 20,000 batches along with segmentation for feature extraction from all lines

sensors throughout the line. This section investigates how large these process deviations are for each section and how they affect the resulting core hardness, beginning with a segmentation of the heat treatment procedure.

Process sections

The segmentation intervals are based on the process stages, ideally, such that all significant values for a prediction are extracted but not more (i.e., using too many segments). The convection furnace for tempering is not included as temperatures across all segments and lines are similar. Chosen segments for process gas furnace and salt bath are shown in Figures 5.8 and 5.9, respectively.

These plots contain a condensed version of all temperature curves in the process gas furnace and salt bath in the form of a kernel density estimation at each resampled point in time. Temperatures are measured with a precision of 1 K resulting in a discretization reflected in the plots, especially for the salt bath (and more attenuated for austenitization), by the individual dark bands. Both the interpolation step in resampling and the kernel density estimation lead to a deviation from pure integer values and, thereby, more closely approximate the true distribution, which has a smooth shape. The darkness of color indicates the percentiles closest to the modal values (i.e., extreme values or most dense regions)[6].

In comparison to the temperature uniformity surveys for different batch positions in Section 4.2.2 from 35 batches, the spread of these temperature curves from approximately 20,000 batches from all lines seems to be quite small (e.g., $\Delta 16\,\mathrm{K}$ for the minimal turning point of all measured process gas furnace temperatures and $\Delta 9\,\mathrm{K}$ for the 90^{th} percentile shortly before austenitization). Critical process stages like austenitization and quenching in the salt bath have even smaller windows indicating that intra-batch temperatures deviate more strongly than temperatures measured at a fixed point between batches. This serial process appears to be quite robust and stable, with such minor temperature deviations hardly resulting in huge hardness differences.

To determine whether learning from these minimal variety temperatures is reasonable, we take a closer look at different hardness buckets and their associated salt bath temperature. Figure 5.10 sorts batches into four bins based on their core hardness and then plots the 60^{th} percentile temperature band of each bin in their respective color. As expected, greater hardness is associated with a slightly lower soaking temperature T_{Ia} for

[6] The intervals for the percentiles have been calculated by using the estimated PDF. Individual percentiles were found by shifting up and down the PDF to find its zeros until the integral between the zeros of the CDF was equal to the percentile. Of course, multi-modal PDFs have multiple zeros (e.g., 6 zeros for 3 modes in Figure 5.9).

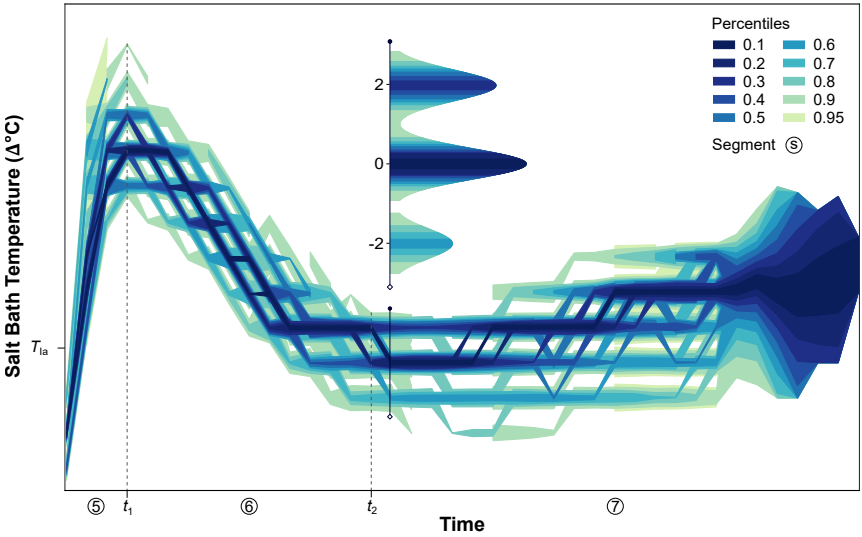


Figure 5.9: Kernel density estimation of salt bath temperature curves at each minute from $\sim$ 20,000 batches along with segmentation for feature extraction from all lines

isothermal conversion at t_2. Thus, including a temperature-related feature from segment 7 should be of predictive value even if the variations seem to be minimal. Alternatively, the time from immersion in the salt bath to reaching the peak temperature (approximately at t_1) and the undershoot (approximately at t_2) can also serve as valuable indicators.

Process features

To include all important characteristic properties of the heat treatment process into the prediction, 156 features from sensors throughout the lines have been extracted, as described in Section 3.2.4. As demonstration, this section assesses three selected features: mean salt bath temperature $\overline{T}_{\text{salt bath},7}$ in section 7 , maximal mass flow of the enrichment gas $\text{Mass flow}_{\text{furnace},3,\max}$ in section 3, and temperature skew in the furnace

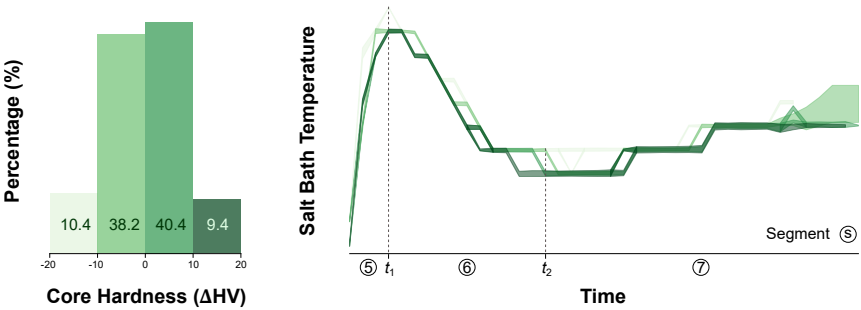


Figure 5.10: Left: histogram of core hardness with 4 bins, right: respective salt bath temperature bands (60^{th} percentile)

$T_{\text{furnace,3,skew}}$ also in section 3. Analyzed are their dependency on line, development over time, and correlation to associated hardness (i.e., label to be predicted).

Feature $\overline{T}_{\text{salt bath,7}}$ is depicted over the course of a two and a half year period for every line in Figure 5.11 a). Hardly any significant changes can be noted over time, except for line 23 that undergoes a drop of 3 K in 2019. More significant are the differences between lines (e.g., 2.5 K between the means of line 20 and 26), as shown in the box plots b) that contain the distribution of all $\overline{T}_{\text{salt bath,7}}$ per line. Notably, this feature seems to be quite stable for each line with very little variance such that the boxes (25^{th} to 75^{th} percentile) are often barely overlapping. The minimal correlation of -.04 with the core hardness, shown in c), hardly allows any conclusion to be drawn. Nonetheless, the accumulation of chronologically stable differences between the lines might be a reason for the hardness offset between the lines, which itself are due to a multitude of possible reasons: different temperature settings for each line, temperature sensor bias (e.g., orientation (correct T at wrong position) or calibration (correct position but wrong T)), or unmeasured influences like paneling.

To examine correlations between features that appear to be mostly stable over time for each line and a label that is susceptible to measurement error,

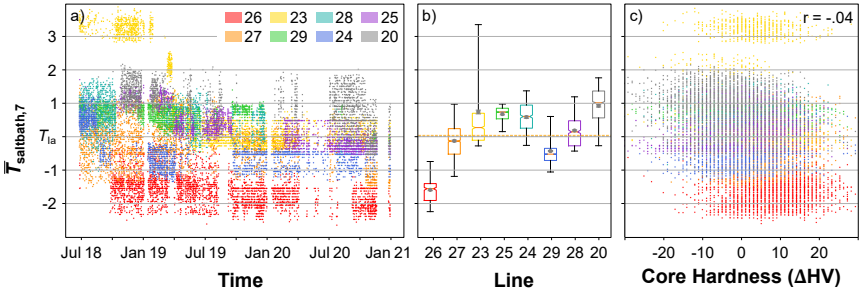


Figure 5.11: a) $\overline{T}_{\text{salt bath},7}$ of individual batches per line over time, b) box plot per line of $\overline{T}_{\text{salt bath},7}$ with 99 % CI, c) scatter plot of $\overline{T}_{\text{salt bath},7}$ and core hardness for each batch

we take the mean of each month for both feature and label. In this way, we can alleviate the measurement error and might capture changes in core hardness that are explained by slow changes in the mean of a feature in a particular line.

As shown in Figure 5.12 a), the correlation between $\overline{T}_{\text{salt bath},7}$ and core hardness now becomes r = -.24. It does still barely support a salt bath influence hypothesis. Although the points of one color (i.e., same line) do lie closer to the imagined negative correlation line[7], it seems unreasonable to attribute a specific change in core hardness to a change in $\overline{T}_{\text{salt bath},7}$.

The maximal enrichment gas flow $\text{Mass flow}_{\text{furnace},3,\text{max}}$ shortly before austenitization is a paragon of spurious correlation with core hardness (r = -.46), since there exists no causal link between the mass flow and core hardness of the component. It just so happens that line 26 (red), which produces the parts with the highest core hardness, also has the lowest mass flow. Line 27 (orange) follows at some distance, while the remaining lines cluster in the lower right corner. It is essential to be aware of such associations (or the lack thereof) since ML algorithms do not learn cause-effect relationships but use inputs that they can map to a specific

[7] upper left to lower right

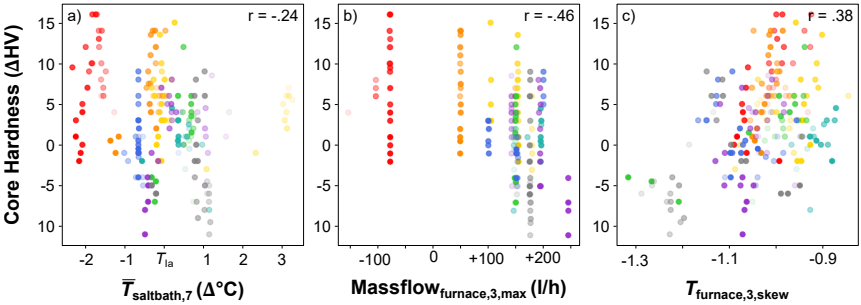


Figure 5.12: For each line, one-month-windows have been cut and the respective mean of a) $\overline{T}_{\text{salt bath},7}$, b) Mass flow$_{\text{furnace},3,\text{max}}$, c) $T_{\text{furnace},3,\text{skew}}$ correlated with the mean core hardness of that window. More transparency indicates that the values lie further in the past. Colors indicate line association, see Figure 5.11

output. If a model learns the correlation in (d) and is used for predictions, it would likely give incorrect results in time. Especially if the mass flow of line 26 is increased, resulting in a decreased core hardness prediction but having no actual effect on core hardness. Incidentally, the reverse is true for the surface hardness with an $r = .6$. Here, more gas can deliver more carbon resulting in greater surface hardness. Thus, caution must be exercised when correlating features with labels, always considering other influential factors (measurable and immeasurable).

As a last example we investigate c) $T_{\text{furnace},3,\text{skew}}$ which is a feature of true predictive power. Higher skew means that T_{A} is reached faster, in turn leading to a longer austenitization time which leads to greater core hardness as suggested by the positive correlation of $r = .38$. Compared to minimum or maximum temperatures, in general, features related to dwell time are more promising indicators of hardness. Especially, the length between the last[8] segments (i.e., 4, 7, and convection furnace[9]) of each process step varies substantially between batches. These differences in

[8] All other segments are fixed in length for bainitizing.
[9] It comprises one large segment of its own.

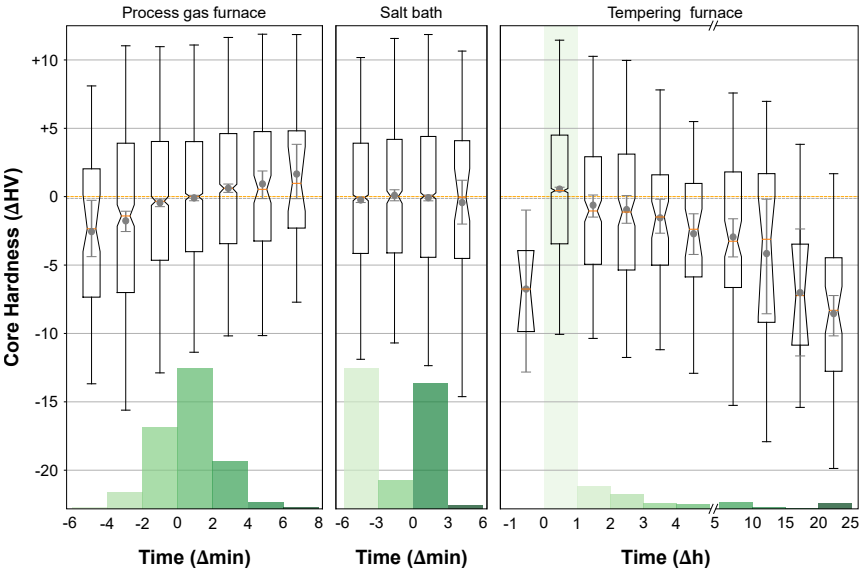


Figure 5.13: Core hardness over differences in dwell time between batches in process gas furnace, salt bath, and tempering furnace, along with histogram of respective bin size in green

dwell time evoke a change in hardness with clear physical explanation, as shown in the next section.

Dwell time

Although the heat treatment procedure is conducted by a well-controlled system, the process step durations (i.e., dwell time in furnace, salt bath, and tempering furnace) are subject to fluctuation. This elongated or shortened austenitization or soaking time has an effect on resulting hardness, depicted in Figure 5.13.

Remaining in the process gas furnace longer[10] increases time spent for austenitization which is known, under otherwise identical conditions, to increase the resulting hardness. No apparent effect is found for the salt bath. The dwell time was reduced about 5 min in the middle of 2019, see Appendix A.7, resulting in an uneven distribution of bins but not in a drop or increase in hardness. Surprisingly, an increased soaking time in the tempering furnace significantly reduces hardness. Most likely, the transformation to bainite leading to increased hardness at some point tips into a kind of tempering that softens the bainitic structure again. From this inspection, the lower hardness coinciding with alarm $A_{\text{Unload tempering exceeded}}$ is readily understood. It turns out that batches produced late Saturday stay in the tempering furnace until they are unloaded early Monday morning, rendering Saturday batches lower in hardness due to increased soaking time that effectively tempers the cylinder heads.

5.3 Case Hardening

Generally, the analytical approach for case hardening features is similar in structure to bainitization, with exceptions detailed in the following. As discussed in the previous chapter, the four batch positions from which two test specimens are sampled alternatingly cause a constant bias. In order to be able to use labels from all positions, they have been corrected by their respective median. Further, the features are primarily analyzed with respect to the scores (i.e., Score 0.1/0.4/0.7), where surface near measurement positions (meas. pos.) are more important because carburization has a more significant influence here and the result is more critical for the nozzle bodies' expected mean time to failure. The last Section 5.3.4 will also give a justification for the suitability of these scores.

[10] Could also partly be due to a longer time necessary to reach T_{A}.

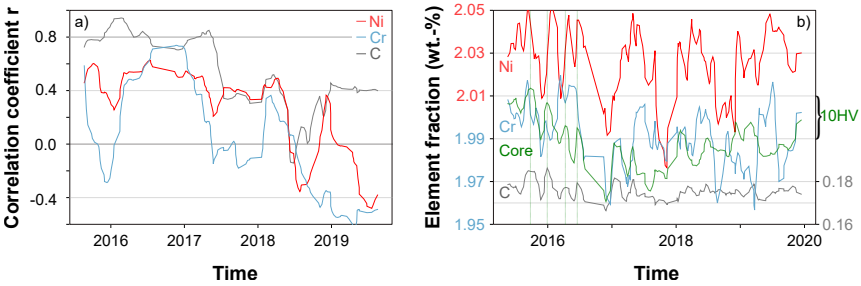


Figure 5.14: a) Correlation coefficient r calculated between core hardness and weight fraction of elements with a rolling window[11] (50 days), b) weight fraction of carbon (C), nickel (Ni), and chromium (Cr) in 18CrNi8 along with core hardness over time smoothed with a rolling window (5 days) with triangular shape

5.3.1 Material

Variations in chemical composition of the case hardening steel 18CrNi8 were proposed to cause fluctuations in the core hardness in Section 4.3.2 based on Figure 4.20. In the following, this hypothesis is investigated by partial correlation and regression analysis.

Unlike the 100Cr6, correlations between core hardness and weight fractions of specific elements in the 18CrNi8 are time-dependent for the period under consideration[12]. Thus, instead of a correlation heat map, Figure 5.14 a) shows the correlation coefficients of the usual suspects C, Cr, and Ni with core hardness over time along with the time series they were derived from shown in Figure 5.14 b). The congruent shape of their curves suggests a strong influence of C on core hardness during 2016/17. This finding is reflected in a high r during that period. It is also observable for Ni, although to a smaller degree. In fact, a similar synchronous movement is displayed by Mn and Si for 2016, see Appendix A.8. Such coincidental synchronous

[11] r is calculated for all value pairs in the corresponding window.
[12] A time dependency might also very likely be found for the 100Cr6 if observed over a more extended period.

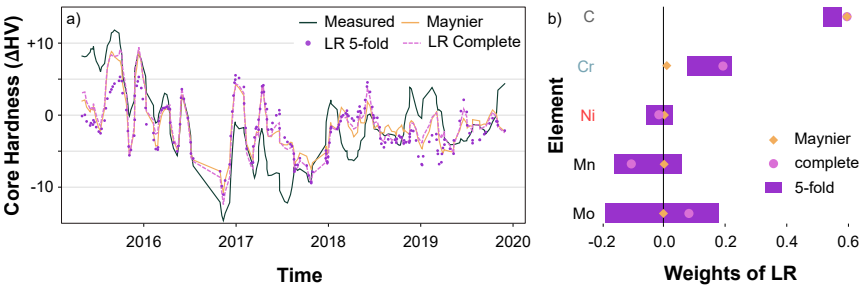


Figure 5.15: a) Prediction from LR after learning the complete core hardness and after learning from 4 years and predicting the remaining year (5-fold) based on the elements in b). b) Coefficient in Mayniers model (orange route), weights learned for the complete data set of core hardness (pink dot). Dark purple bars show minimum and maximum of weights learned during the 5-fold training

changes of elements likely intensify oscillation of core hardness. An awareness of such interdependence might prevent an investigation of fluctuations in the wrong hypothesis space. However, as correlations change over time, chemical composition's exact cause and effect relations are more challenging to quantify. They are mitigated by several unmeasured effects[13] from any other process step (e.g., annealing or carbide formation).

Figure 5.15 a) demonstrates such a quantification attempt via linear regression[14], based on the elements in 5.15 b). A complete regression fitted to the whole curve serves as a benchmark for achievable predictability, simulating complete knowledge availability. The 5-fold regression learns from 4 years and then predicts the respective fifth year. Neither of the curves predicts the one large down and upward trend that the green curve (Measured) displays (i.e., starting at nearly +10 HV before 2016, dropping to -10 HV in 2017 and recovering to 0 HV in 2020), which presumably is not due to chemical composition but could be a drift of a measurement

[13] A partially negative r like for Cr might also result from a simultaneous increase of Cr and decrease of C, thereby dropping hardenability although Cr was rising.

[14] It may be assumed, that the small changes in chemical composition under consideration are locally linear.

device. In contrast, in almost all cases, the model accurately predicts the direction of a short up- or downward trend but regularly misestimates the magnitude of the amplitude. From the weights, it may also be inferred that C is the only reliable predictor, with Cr aiding a little. None of the remaining elements contributes consistent weights and should, therefore, not be used for prediction as their influence is too small or nonlinear to be captured.

In sum, the relatively small fluctuations of C seem to affect hardenability (respectively hardness) more deeply and consistently than either Cr or Ni. While an accurate prediction seems out of reach, at least a trend direction might be forecastable in some cases. Finally, the fluctuations in 2016/17 can confidently be attributed to the material composition. More fluctuations are to come in the next section.

5.3.2 Production line

In contrast to the bainitization in a single line, nozzle body batches are processed in three different stations (cf., 3.1.2). This section investigates individual furnaces, freezers, and the effect of routes taken through these stations. To evaluate the difference between stations, Figure 5.16 plots the drift corrected[15] mean hardness of batches produced with a given station over time along with their overall CIs. The following three observations can be made.

1) Stations mostly, but not always, behave consistently for different scores (i.e., if vacuum furnace 1 has the highest mean hardness for Score 0.1, it most likely also does so for Score 0.4 and 0.7) with a notable exception for the deep freezers. 2) In contrast, scores are affected differently by the various stations. Scores 0.4 and 0.7 vary strongly for the vacuum furnace,

[15] The long-term drifts (e.g., Section 5.2.1 and 4.3.2 were approximated by a first-order IIR filter and, then, subtracted from the respective measurement score, to make differences in stations visible.

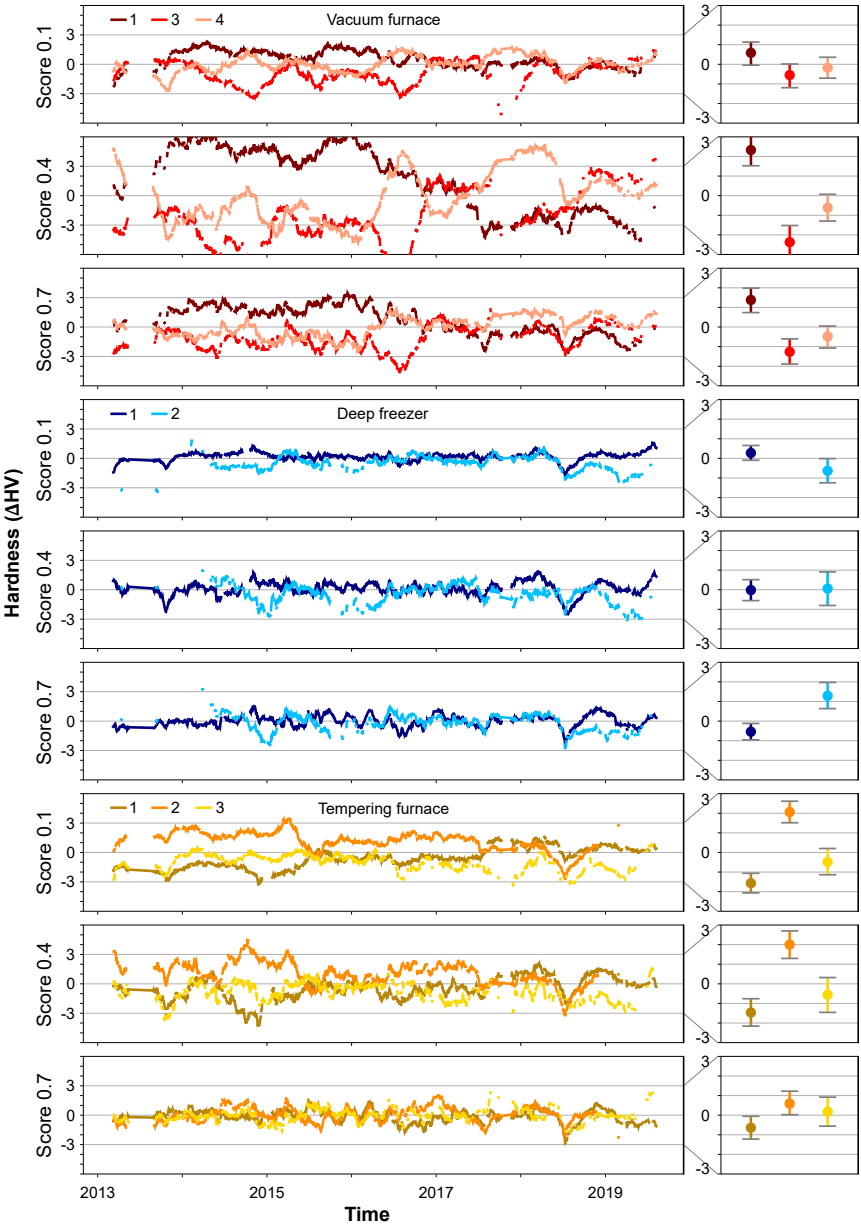


Figure 5.16: Left) hardness scores smoothed by rolling window (100 days) of batches associated with a particular vacuum furnace, deep freezer, or tempering furnace. Scores were initially corrected by their common drift calculated by a first-order IIR filter. Right) mean and 99.9 % CI of stations

while tempering affects 0.1 and 0.4 more strongly. These observations support a temperature-based hypothesis where the temperature influence is mitigated by carburization for the surface-near Score 0.1 but has a much more substantial impact during tempering on these same surface near layers. These claims will be further strengthened in the upcoming process feature section. 3a) Variance between stations seems to have decreased over the years, presumably as production was optimized stepwise. 3b) Although overall variance was reduced, there are still significant differences between stations that change over time (e.g., Score 0.4 in the vacuum furnace). While vacuum furnace 1 during 2015 produced at a level being around 7 HV higher than furnace 2, in 2018, the reverse was true. This example underlines the importance of time-depended inspections since an analysis of the CIs might have led to the wrong conclusion, that 1 always produces harder than 2. Thus, passing a feature such as a station to an ML algorithm can lead to undesirable predictions, especially when the training and test sets are separated in time[16]. Although such individual deltas might seem relatively small, they might add up to noticeable differences when taking specific routes through the stations, see Figure 5.17. It shows how the combination of different vacuum and tempering furnaces leads to divergent mean hardness over time. For Score 0.4, this divergence averages at 10 HV underlining the greater effect caused by adding up seemingly small differences in stations. Careful planing of routes could help to prevent undesirable deviations from the target hardness. Balancing the effects of individual vacuum furnaces by changing the parameters of freezing or tempering furnaces for a particular combination seems too complex and would probably not yield the desired result. Instead, the furnace parameters should be optimized to approach a common mean behavior (i.e., all vacuum furnaces, freezer, etc. behave in the same way).

[16] They should be chronologically separated to prevent data leakage from the present to the future.

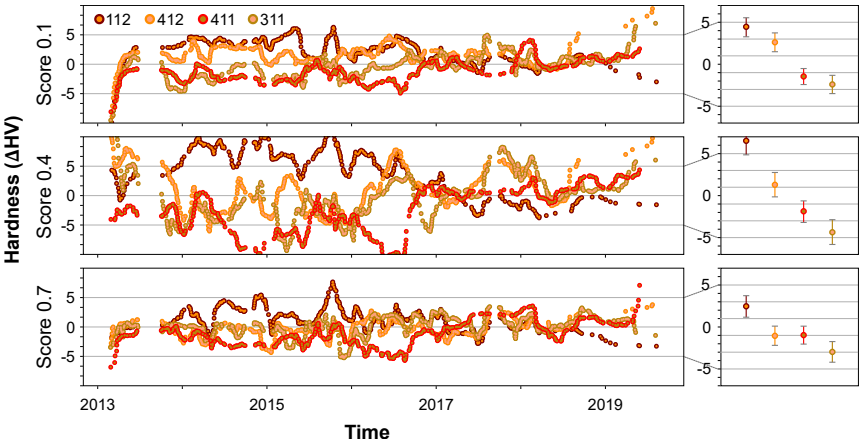


Figure 5.17: Left) hardness scores smoothed by rolling window (100 days) of batches associated with a particular route (e.g., 112 = vacuum 1, freezer 1, tempering 2) Scores were initially corrected by their common drift calculated by a first-order IIR filter. Right) mean and 99.9 % CI of route

It will be difficult to determine which part of the fluctuation can be attributed to process parameter differences and which are due to maintenance or other unmeasured external circumstances. In the latter case, it might also be necessary to track the hardness level of each station individually for each score to identify actual adverse routes and detect stations' divergence. To understand the hardness variance, in addition to the difference between stations, it is also important to know what type of component was heat treated, as explained below.

5.3.3 Metadata

Components

Batches contain nozzle bodies of different component families (e.g., X2, X4) that are further divided into types (e.g., 6, 8), each with slight geometrical variations in particular at the nozzle itself (cf., Section 3.1.2). These,

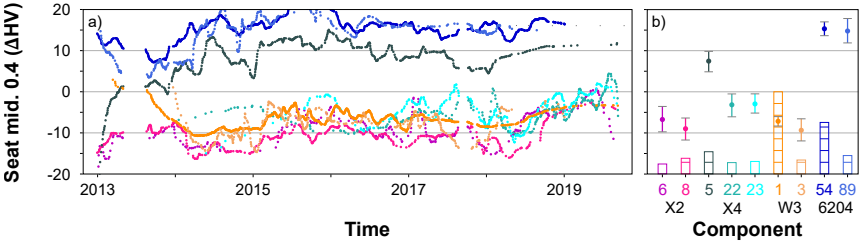


Figure 5.18: a) Hardness of Seat middle 0.4 mm over time. It was firstly corrected by the common drift using a slow IIR filter and secondly smoothed for each component type by an individual IIR filter with an ω proportional[19] to the amount of components produced indicated by the bars in b), which also shows the means and their 99.9 % CI

in turn, slightly influence the absorption behavior of heat and carbon. A strong influence of such geometrical differences manifests in hardness of the Seat middle 0.4 mm, shown in Figure 5.18, most likely due to carburization behavior as the walls are thinnest here (other meas. pos. are not significantly different). Components of family 6204 are consistently 20 HV harder over time[17] than the remaining families, which means that most of the spread of the particularly wide Seat middle 0.4 mm distribution can be attributed to component family differences. No difference is observed between the components of the same family, with a notable exception for X4: type 5 is harder than type 22 and 23, which is likely due to the fact that the latter two are made of a different grade of steel (i.e., electroslag remelting (ESU)[18]). Thus, whether a component type or family serves as a valuable predictor and whether types should be combined into a single feature must be judged on a case-by-case basis but need not be tracked over time. The last meta-information that may serve as a feature is the alarms, briefly discussed below.

[17] Thus, it is not necessary to track differences between families over time. Also, there would be no physical explanation to do so, because the geometry does not change over time.

[18] Elektroschlacke-Umschmelzverfahren

Alarms

The author was not able to derive helpful information from the occurrence of an alarm and link it to a particular hardness measure of a batch. Although some alarms are correlated with reduced or increased quality measures, these effects vanish when correcting for the collective trend meaning that they most likely co-occurred with a time span when the overall hardness was higher or lower. Second, significant alarms usually stem from mechanical failure of the production line. Since there are very few moving parts in the case hardening stations (as compared to the bainitization line), barely any alarms of this type occur. Third, no causal link could be established between such alarms and indicated deviation of the quality measure. In the words of Prof. Mikut: "There is not much to be found in a competent process."

Discussion

Despite the fact that some interesting effects were found in the data presented above, claims of causal effects should be made tentatively. Fluctuations over time, uneven distribution, or unmeasured influences on a particular feature, be it station, component, route, alarm, or measurement position, can lead to spurious correlations and imply significant differences where none exist. It could be, for example, that more components of family A took route 1 and more of family B route 3. It is now difficult to attribute a deviation from the mean to either the component family or the route. The same goes for shifts due to material composition, possibly resulting in a hardness drop of a certain component type. It leaves open the question of which of both caused the drop or even if a third factor was involved. On that score, although potentially valuable, these analyses

[19] The ω must be adjusted for each component type because each has a different "sampling rate". Otherwise, components produced less frequently would be smoothed more than if the same filter were applied to a more common component.

must also be taken with a sufficient amount of skepticism and must not replace well-controlled experiments. With these words of caution in mind, we now turn to the influences of process fluctuations as measured by the sensor signals.

5.3.4 Sensor signals

The information of the sensor signals is analyzed in featurized form, the extraction process of which is described in Section 3.2.4, which already contains the heat treatment's key values in condensed form. As the number of extracted features may be sizable, this section first investigates which features are actually of predictive value not to feed useless or redundant information to the algorithm later and better understand which differences between the process actually lead to dissimilar results over time. Second, it is useful to confirm that the additional labels created (i.e., Score 0.1/0.4/0.7) are predictable by the same input features as the meas. pos. they were derived from (i.e., are influenced by the same physical phenomena during the process).

Feature over time and label correlation

A first impression of a feature's usefulness can be gained by its correlation coefficient with the respective label, calculated for every extracted feature and label combination. Among the features with the highest $|r|$ are those shown in Figure 5.19. It contains the feature values and their correlation with the labels over time as well as a scatter plot of feature and label, respectively.

Neither of the features remains continuous over time, nor are they similar for the stations they were measured in. Vacuum furnace 3, for example, was not able to generate the same maximal quenching pressure leading to a higher temperature 25 s after quenching before maintenance of the

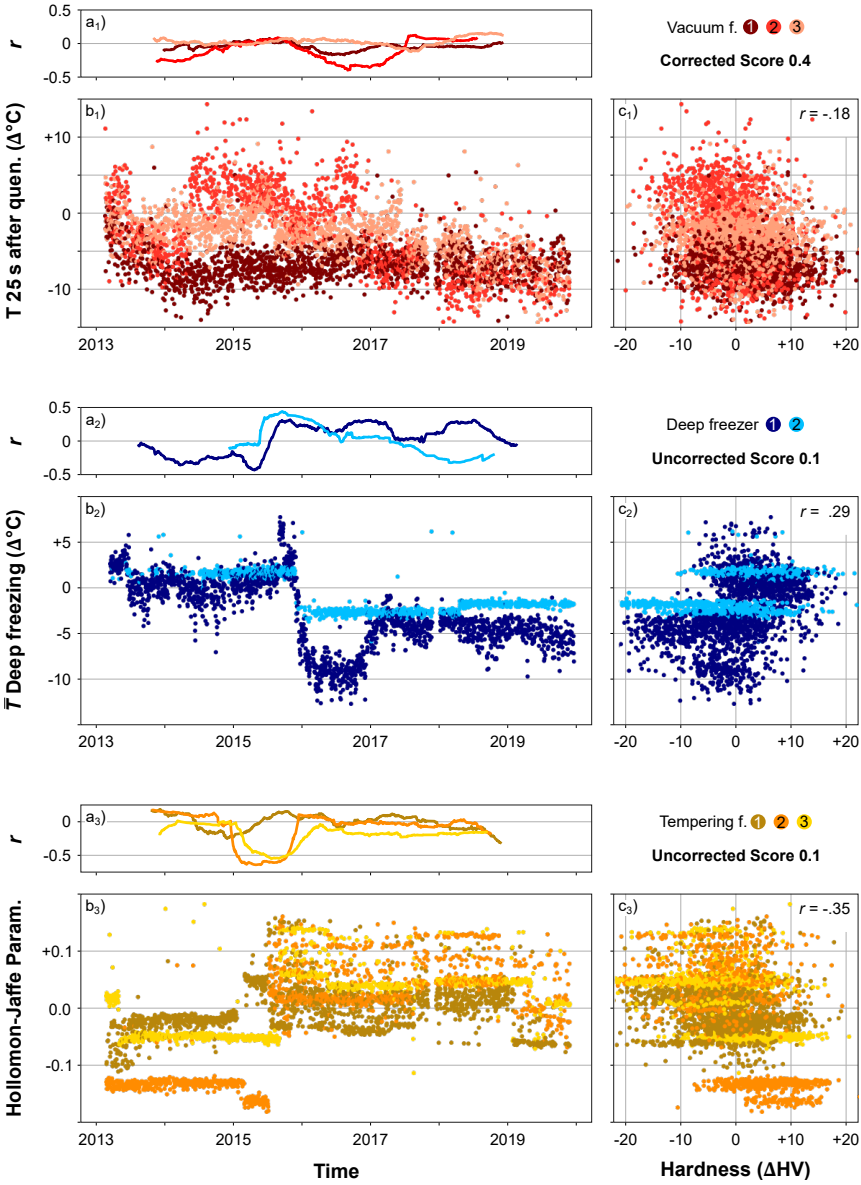


Figure 5.19: a) Rolling window correlation (300 days) of label with feature per station, b) feature over time per station, c) scatter plot of feature and label. The values of the label Score 0.4 were time drift corrected by subtracting their long-term fluctuations determined by an IIR filter

system at the beginning of 2017. During this period, the rolling r drops remarkably, indicating the expected negative correlation, where a lower temperature is associated with a greater hardness of Score 0.4, also inferable from the scatter plot. When a single line is stationary (e.g., after 2018), the rolling r fluctuates around zero for all stations, suggesting that the process is so stable (with respect to this feature) and/or the measurement error is so high that inference of hardness influence seems almost impossible, with one exception for the freezer. Although the mean cooling temperature of deep freezer 2 is rock stable, its rolling r decreases continuously. Further, the scatter correlation claims a positive r, which would contradict the expected behavior, where lower freezing temperatures lead to more transformation of the remaining austenite leading to higher hardness. This phenomenon (i.e., a drift of rolling r and positive scatter r) occurs because the label Score 0.1 was not shift corrected. As was shown in Figure 4.20, the hardness drops during 2018 (diamond change) and 2015 (rise of initial T_T) while, coincidentally, the freezing temperature was also lowered in 2016. Omitting such a shift correction, thus, may lead to the appearance of an erroneous correlation. On the other hand, shift correction may also erase effects like the one visible for the Hollomon–Jaffe parameter (H_P), cf. Section 2.1.2. Due to the temperature increase between batches of the tempering furnace in 2015, nozzle bodies experience a slightly stronger tempering effect resulting in a higher H_P. A lower H_P naturally results in a harder martensitic surface, especially exhibited by tempering furnace 2, which incidentally was the one that generally produced harder components (cf. Figure 5.16). Again, this correlation effect is only visible during the large discontinuity as indicated by the deflection of the rolling r at these points as well as the shifted hardness values in the scatter plot.

In sum, larger feature discontinuities for a single station (e.g., vacuum furnace 3) or all stations (e.g., tempering furnaces) show visible effects on hardness while stationary fluctuations are not noticeably reflected in the rolling r. Some feature-label correlations are only visible when shift

correcting the label; others are not or exhibit erroneous relations. The complete tracking of each station (proposed above) will provide a feature that accounts for larger changes in each line. In turn, the ML algorithm might not be able to learn something from these changes, raising how to optimally present features to the ML algorithm during training. This matter will be analyzed in the upcoming chapter.

Correlation between features

Redundant information between features as well as their integrity are shown in Figure 5.20 by the correlation between selected features of the deep freezer, vacuum furnace, and the shift corrected label Score 0.1. In line with expectation, mean and minimal freezing temperature are prominently positively correlated as well as all features belonging to the tempering furnace, where the time spent at tempering temperature has the most substantial influence on the H_P. Since the H_P (calculated from Δt_T and $\overline{T}_\mathrm{T}$) is the best predictor for hardness among the three, it should be the one included in the feature set while dropping the other two as redundant. The same would be valid for the deep freezer (i.e., only keeping one) if the ML algorithm can actually use one of them. This might be difficult due to their low correlation, which now is negative as expected due to the shift correction, as compared to Figure 5.19. In- and exclusion of other features will, thus, be determined by a feature selection algorithm.

Surprisingly, correlations between features of freezer and tempering furnace are negative suggesting that lower deep freezing leads to higher tempering, which in fact is not the case. Deep freezer temperatures were indeed lowered at the time when initial T_T was risen, resulting in a negative correlation, but no causal connection could be found (i.e., the temperature of freezer and tempering are independent).

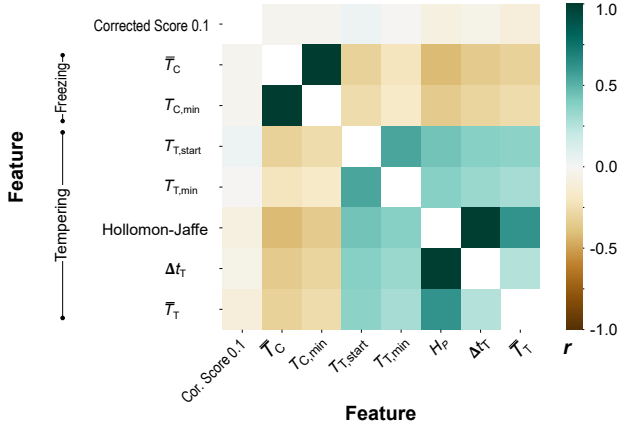


Figure 5.20: Heat map of correlation coefficient r between features

Conformity between meas. pos. regarding feature

To evaluate whether the same features are predictive of different labels, we turn to Figure 5.21. For each label-feature combination, the correlation coefficient r was calculated. Now each of the 8 meas. pos. has features with which it is more or less correlated. Figure 5.21 then shows whether the same feature, represented by one dot, is highly correlated with two different labels. If a high correlation between the feature rankings of two labels exist (e.g., $r = 0.99$ between features of Shoulder 0.1 mm and Seat middle 0.1 mm), it can be expected that the same specific physical conditions influence those labels in the process. Between most meas. pos. (with exception to Undercut 0.1 mm) this is true, expectedly more so in clusters of similar depth (i.e., 0.1 mm vs. 0.4 mm vs deeper). Thus, the scores derived previously are meaningful aggregations of meas. pos. and input features do not have to be optimized individually for all labels but these groups of label scores. Most features of vacuum furnace and deep freezer are of little use indicated by the peak around $r = 0$ of the distributions shown along the diagonal of Figure 5.21. This is also due to the much higher number of features extracted from the vacuum furnaces, containing

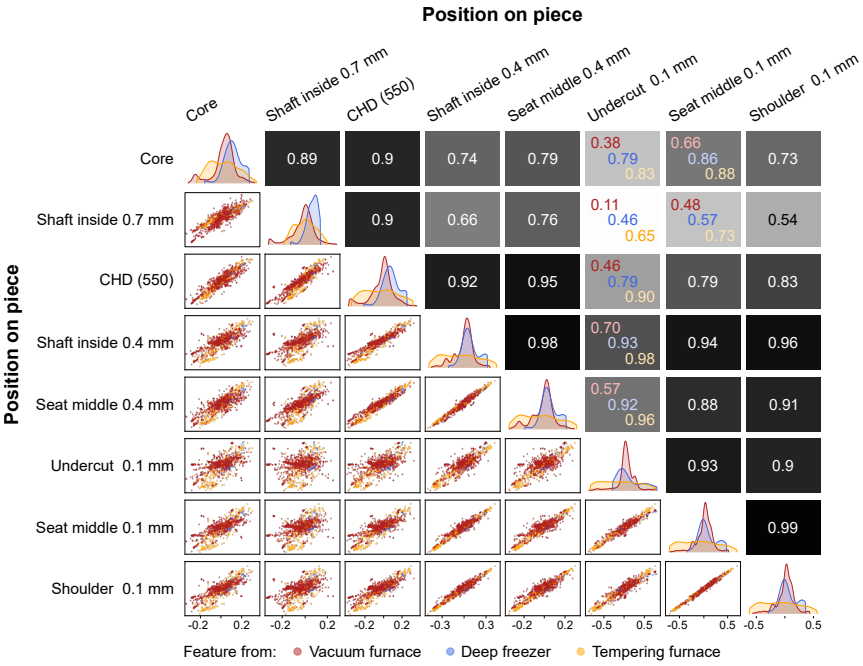


Figure 5.21: Each dot represents one extracted feature from vacuum furnace (red), deep freezer (blue) or tempering furnace (orange). The scatter plot shows the correlation coefficients r of that feature with the respective labels (i.e., position on piece - lines and columns). The heat map in turn gives the correlation coefficient r of the scatter plot. If the individual r between vacuum-, tempering furnace, and deep freezer are greater than 0.2 then their individual values are shown

many more segments and channels, many of which are useless byproducts. Conversely, features of the tempering furnace are more often found at the edges of the distribution and scatter plots, indicating a higher r and, therefore, more significant influence.

5.4 Discussion

Generally, even seemingly small percentage changes in material composition (especially carbon concentration) can lead to severe deviations in measured hardness, especially for measuring positions and processes where little or no carburization occurs. Both the models obtained from empirical data and those known from the literature support this claim. However, the resulting hardness variance usually cannot be predicted directly from the proportion of the respective elements since too many other influences are at work. For example, the line or route through different furnaces and freezers may result in an offset. That is, a heat treatment program applied to similar batches in two seemingly identical furnaces can lead to consistently different results. Moreover, the high stability of industrial processes makes the influence analysis of the heat treatment itself laborious since the differences between lines are more significant than between successive batches of the same line. Part of the problem may be that measurements of absolute temperature or absolute pressure are not accurate. While the internal feedback controller achieves a minimal error between measured and reference temperature, the measurement may be off by a few degrees from the true temperature—a problem to be expected in any heat treatment process. Consequently, long-term changes (e.g., change of salt bath temperature) are more indicative than minor variations in the same furnace. Interestingly, regarding hardness prediction, the alarms do not provide any additional information beyond the process measurements (i.e., at least as long as the operator performs his job well)[20]. In contrast, an influence of the measuring procedure and the measuring position on the hardness result can very well be recognized, especially when the geometry

[20] There are cases when alarms may be necessary if the operators do not work cleanly. For instance, the salt bath loses salt over time. If it is not refilled in time, the top layer of the batch is no longer completely covered with salt. These parts will then not be quenched correctly. The salt bath temperature does not capture this mishap. To prevent such ill-fortune from happening, there is an alarm that is triggered when the salt bath level is too low.

of a component is slightly changed at a critical position (e.g., nozzle tip of a nozzle body or tooth of a gear).

Higher-level domain knowledge is essential for a variance breakdown of heat treatment processes in general since only meaningful and not all possible features should be extracted from the process in order to avoid spurious correlations and possibly resulting unfavorable measures as a consequence. It is to be expected that the influences examined above will be found in most heat treatment processes, and it may be difficult to disentangle their intertwined nature.

6 Machine Learning

6.1 Introduction

To put the proposed data mining framework to the test, we investigate how much of the variance studied in the previous chapters can actually be learned and predicted by our machine learning and hidden states pipeline. The results determine which cost reduction strategy (e.g., reduced testing) is applicable. First, optimal fluctuation tracking by filters and their ability to forecast hardness of upcoming batches under various information restrictions is discussed in Sections 6.2.1 and 6.3.1. For bainitization, we proceed with 6.2.2 to investigate the predictive power of process features, explain what the models learned and what the optimal learning strategy is. In 6.2.3 we access the difficulties arising from process outliers. For case hardening, we focus on a complete percental breakdown of variance contributors in 6.3.2.

6.2 Bainitizing

6.2.1 Forecasting and label tracking

To account for unknown, not measured, and immeasurable influences (e.g., material composition, modifications through maintenance, etc.) that present changing influences over time, the hardness level of each line needs to be tracked continuously, as was hinted at in Section 5.2.2.

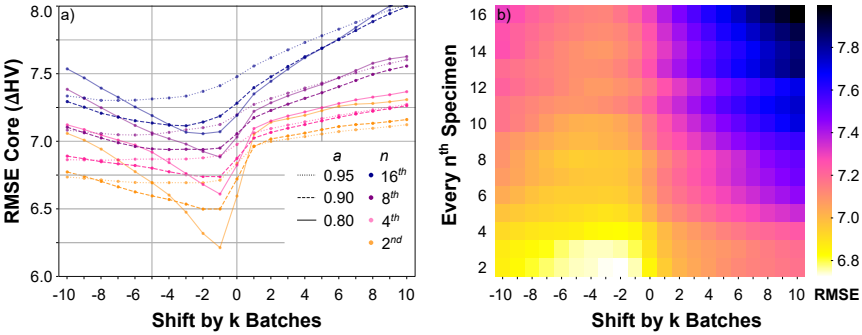


Figure 6.1: Average RMSE between label and output of causal first-order IIR filter applied to individual lines for a) different a using every 2^{nd}, 4^{th}, 8^{th}, and 16^{th} batch as filter input, b) for every n^{th} batch as filter input to predict the remaining batches with $a = 0.9$. Predictions were shifted by k batches before RMSE calculation to simulate information delay (k>0) or advance (k<0)

This section elaborates on the optimal filter parameters to track the mean produced core hardness of each line. Since the goal is to reduce test specimens, in a first step, we investigate how the fraction of tested to predicted pieces influences the RMSE between label and filter output. Of course, the ultimate intent is to sample as little as possible (only every n^{th} sample) while still achieving acceptable filter performance.

As described in Section 3.2.5, we use a causal first-order Butterworth IIR filter. Because parameter optimization led to $N = 1$ in over 99 % of cases, the filter could be reduced to a function of only one parameter, that is, the retaining percentage[1] a. Figure 6.1 a) shows the RMSE between label and filter output for different a, number of test specimens used as filter input (i.e., 1 out of n, or every n^{th})[2], and time-related measurement information availability (i.e., k>0 forecasting hardness k batches into the future, k<0 phase delay correction possible). Three conclusions may be drawn from

[1] $y_n = a\, y_{n-1} + \frac{1-a}{2}(x_n + x_{n-1})$

[2] The higher n, the more costs can be saved by not having to test n-1 pieces.

this figure: first, a retaining percentage a of around 0.9 seems to strike a good balance between forecasting and phase delay correction. A faster filter (smaller a) may perform better when phase delay correction is possible (i.e., measurements from batches produced after the current batch are available)[3] but is worse for forecasting. Second, forecasting (k>0) accuracy is worse than phase delay correction (k<0). Consequently, if the measurement of test specimens takes a long time and a prediction is needed rapidly[4], the prediction will suffer. Third, using fewer test specimens as input also leads to worse performance. Thus, for prediction purposes, it would be important to wait for the information that can correct the phase delay (optimal correction lies at $k \approx$ -1, that is, waiting for the next batch), where testing less than every 8^{th} batch seems unreasonable. With these preconditions, it should be possible to follow the trend closely enough.

Figure 6.1 b) shows prediction accuracy when using $a = 0.9$ and every n^{th} batch as input. If fewer batches are tested, they need to be shifted slightly further back for optimal phase correction. On the one hand, testing fewer batches might take less time (although probably mitigated by the decreased staff situation due to the reduced testing). On the other hand, the time between tested batches increases, stretching the amount of time until information about "future" batches is available for phase delay correction and, thereby, possibly delaying the release of a batch for further processing. In practice, therefore, the consequences of information delay must be factored into the trade-off between accuracy and cost reduction when deciding how many batches to test ultimately. Moreover, this decision may be different for various lines.

The following optimization looks at lines individually and assumes that every second batch is tested and every other batch predicted. Figure 6.2 a)

[3] This is possible because a prediction is not required immediately after production of a batch. Thus, the test result of a batch produced after the batch for which a prediction is to be made can be used to correct the filter phase delay.

[4] In some cases, the components are supposed to be further processed as fast as possible to prevent a pileup of stock.

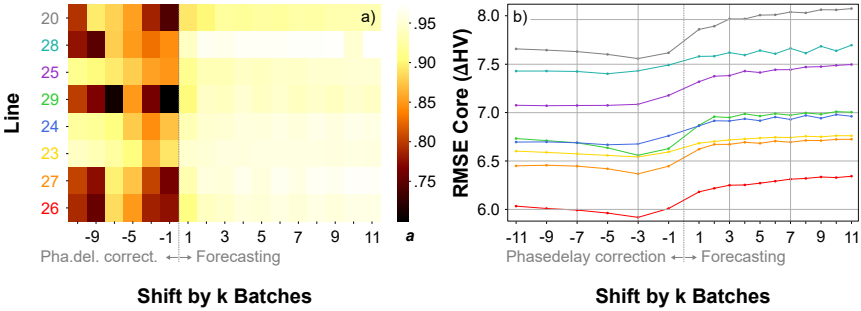


Figure 6.2: Causal first-order IIR filter applied to every 2^{nd} batch to predict every other batch. Predictions were shifted by k batches. a) Optimal retaining percentage a for each salt bath line and shift of k batches, b) $RMSE$ when predicting every second unmeasured batch with the optimal a for each line and different shifts k. Color indicates the line number displayed to the left of a)

shows a heatmap of optimal retaining percentages a per k revealing that, although a varies, in general, the sensitivity of a seems very small (i.e., a big change in a leads to a small change in RMSE)[5]. In the case of forecasting, filters prefer an increased a (corresponding to slower updating), as the best prediction for the future is the current mean production hardness. A small a would give too much weight to the most recent measurements and, thus, most likely deviate from the best estimate for the current mean, though could be helpful to capture small trends as seen in the phase delay corrected outputs[6].

In b), the corresponding RMSE is shown per line. Interestingly, prediction accuracy between different lines is significantly different because the

[5] This is important since the design process of a robust IIR filter is thereby drastically simplified. All lines could use the same filter, independent of k and n.

[6] Cf. Table 3.12: For the optimization, only uneven numbers of phase delay shifts k are considered because every second batch is given to the filter in order to predict every second+k^{th} batch. If k were even, then the filter would have to predict the batch it was given (e.g., $k = 2$) two steps before and try to optimize for that by a high order. However, it should only predict the uneven batches it has not used to update its state.

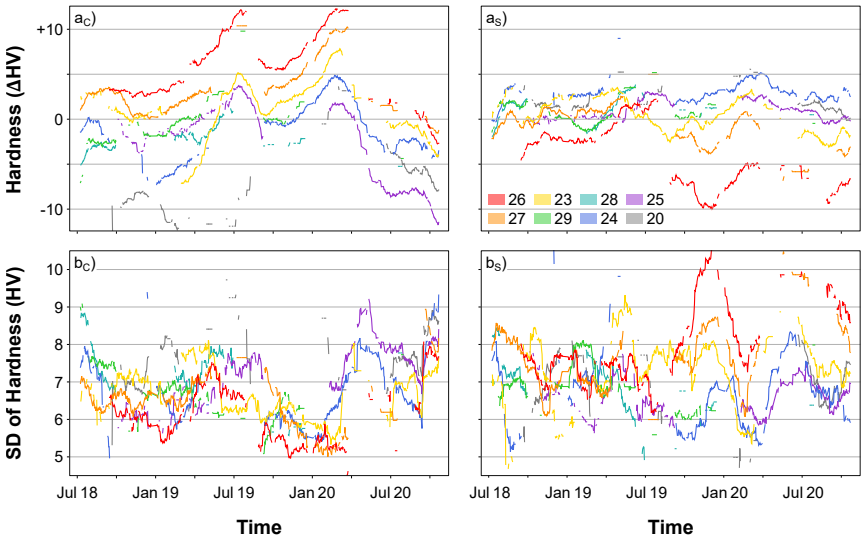


Figure 6.3: Rolling window (50 days) using a) mean and b) standard deviation (SD) of core (C) and surface (S) hardness per line

stability (i.e., the hardness variance produced) is dependent on line age, maintenance, and charging. Larger hardness variations per line (i.e. quality instability per product) could be due to the higher number of different products produced on the same line.

The standard deviation (i.e., produced quality scatter) changes unusually uniformly over time for the different lines, as can be seen in Figure 6.3. Consequently, the RMSE between filter output and label fluctuates to the same extent[7]. The long-term hardness fluctuation does not quite seem to explain the changes in variance. It would have been expected that periods of large changes in hardness would result in greater variance, but this seems to be true only occasionally (e.g., core hardness middle of 2020,

[7] The formula for RMSE and SD is similar ($\sqrt{\frac{1}{n}\sum_i^n (y_i - \mu)^2}$) with the only difference that in the former case μ is the filter output y_n and in the latter μ is the mean over the samples inside the rolling window used (i.e., dummy predictor output).

surface hardness end of 2019). The uniformity of the variance movement suggests that external factors affecting all lines cause these production instabilities (e.g., the ramp down during the corona crisis, starting in Q1 2020, most likely led to worse production conditions). It also means that a prediction outcome must always be compared to the actual variance for the predicted period to assess whether it is valuable and compare it to other periods (i.e., train test splits). The following section attempts to predict the part of this variance that may be explained by the features collected in the previous chapters using data from before 2020 for training and optimization of different pipelines and the remaining 30 % (after 2019) as the test set.

6.2.2 Prediction from features

IIR correction and scaling

As we have seen in the previous sections, for each line, both the labels and the process features fluctuate around individual quasi-stationary points that are subject to change over time. Therefore, the following correction and learning strategies are proposed in order to achieve optimal predictions, see Table 6.1. Training of the ML model might be done with all lines together (i.e., one training set containing data from all lines) or for each line individually (i.e., the number of training sets is equal to the number of lines). The latter might be necessary if the physical line properties differ significantly (e.g., higher temperature leads to lower hardness in one line but higher hardness in another line). The hardness drift is corrected for all conditions by the IIR filter proposed above, which means that, first, the ML model is only learning to predict the deviation from the corrected hardness and, second, the impact of long-term feature fluctuations are already included in this correction. As a consequence, feature fluctuations should be corrected by a filter as well, such that a deviation from

the current feature-mean can be learned as a deviation from the current hardness-mean, which is reflected in the strategy *Xy-all*.

All four strategies were applied to the pipelines explained in the upcoming sections. The test set results are the R^2 score distributions shown in the box plots of Figure 6.4, which confirms that *Xy-all* is the best strategy for feature and label corrections. Accordingly, results in the following paragraphs are based on the *Xy-all* approach.

Strategy	Data correction	Training
Xy-all	X and y corrected	train with all lines
y-all	only y corrected	train with all lines
Xy-indiv.	X and y corrected	train lines individually
y-indiv.	only y corrected	train lines individually

Table 6.1: Four strategies for data correction and training

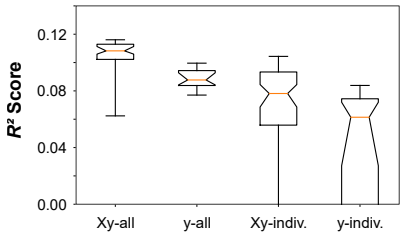


Figure 6.4: Distribution of R^2-scores from 48 optimized pipelines for each strategy

Feature selection and correlation

Different ranking methods were used to provide the ML algorithm with an optimal feature subset to sort the features according to their predictive power regarding the core hardness. Table 6.2 provides the top 11 features calculated by each method, where the manual selection was made with knowledge of the results of the other algorithms by domain expertise. The percental contribution of each additional feature to an overall prediction score calculated by linear regression[8] (LR) is provided in Figure 6.5 a).

[8] Is equivalent to the results of each successive round of SFS.

Table 6.2: Top 11 features as ranked by the feature selection algorithms: Sequential feature forward selection (SFS) with linear regression, genetic algorithms (GA) with linear regression, manual selection, feature importance by random forest (RF), mutual information criteria (MI), and F-score. Color code indicates affinity to line segment: process gas furnace, salt bath, isothermal convection furnace

Rank	SFS	GA	Manual	RF	MI	F-score
1	t_{isotherm}	t_{isotherm}	t_{isotherm}	t_{isotherm}	t_{isotherm}	t_{isotherm}
2	$\overline{T}_{\text{furn,4}}$	$\overline{T}_{\text{furn,4}}$	$\overline{T}_{\text{furn,4}}$	$A_{\text{unload iso}}$	$A_{\text{unload iso}}$	$A_{\text{unload iso}}$
3	$T_{\text{furn,1,med}}$	$T_{\text{furn,1,med}}$	$T_{\text{furn,1,med}}$	$\overline{T}_{\text{furn,2}}$	$\overline{T}_{\text{furn,2}}$	$\overline{T}_{\text{furn,4}}$
4	$T_{\text{furn,2,med}}$	$\overline{T}_{\text{furn,2}}$	$\overline{T}_{\text{furn,2}}$	$\overline{T}_{\text{furn,4}}$	$T_{\text{furn,3,min}}$	$T_{\text{furn,2,min}}$
5	$\overline{T}_{\text{salt,6}}$	$T_{\text{salt,1,med}}$	$\overline{T}_{\text{salt,6}}$	$T_{\text{furn,2,min}}$	t_{furn}	$\overline{T}_{\text{furn,2}}$
6	t_{furn}	$T_{\text{furn,2,med}}$	t_{furn}	$T_{\text{furn,2,med}}$	$T_{\text{furn,3,med}}$	$A_{\text{defect door}}$
7	$\text{Part}_{\text{type A}}$	$\text{Part}_{\text{type A}}$		$T_{\text{furn,1,med}}$	$T_{\text{furn,2,sd}}$	$T_{\text{furn,1,med}}$
8	$A_{\text{flame missing}}$	$A_{\text{flame missing}}$		$T_{\text{furn,4,min}}$	t_{salt}	$T_{\text{furn,2,med}}$
9	$T_{\text{salt,6,sd}}$	$T_{\text{salt,6,sd}}$		$T_{\text{furn,2,max}}$	$T_{\text{salt,7,min}}$	$T_{\text{furn,1,min}}$
10	$T_{\text{salt,5,min}}$	$T_{\text{salt,5,min}}$		t_{furn}	$T_{\text{furn,4,max}}$	$T_{\text{furn,4,max}}$
11	$T_{\text{salt,5,max}}$	$\overline{T}_{\text{salt,6}}$		$T_{\text{furn,4,max}}$	$T_{\text{salt,7,sd}}$	$T_{\text{furn,2,sd}}$

Undoubtedly, the time spent in the convection furnace $t_{\text{isothermal}}$ has the strongest predictive power (due to the previously discussed tempering effect), accounting already for about 75 % of the total score. The second place is occupied by the mean austenitization temperature $\overline{T}_{\text{furn,4}}$ closely followed by the median starting temperature $T_{\text{furn,1,med}}$, jointly contributing another 15 %. With a salt bath feature in fifth place, the SFS, GA, and manual set already contain information of all line segments in the first hand full of features, where any additional information seems to be only of marginal importance. Further, the top features are remarkably uncorrelated, as can be seen in Figure 6.5 b), indicating a sound feature ranking.

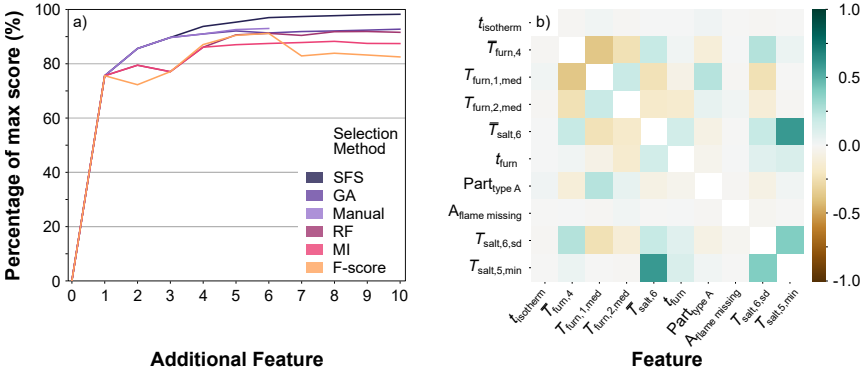


Figure 6.5: a) Performance development of LR when adding the next best additional feature from each subset selection method b) correlation between features of SFS set

To achieve optimal predictions, the hyperparameters of eight plus one different ML pipelines were optimized on the training set by 5-fold cross-validation, as specified in Section 3.3.3. Eight of the pipelines, composed of a robust scaler, percentile selection, and ML algorithm, were optimized by Bayesian search (see Appendix Table A.1), while the ninth was created by tree-based genetic programming using TPOT. Each of the eight pipelines was optimized for every feature subset presented above, leading to 48 combinations plus the TPOT pipeline optimized only on the SFS set due to the expensive computational resources. Figure 6.6 provides four types of information for each pipeline: a) R^2-scores during optimization, b) number of optimal features, as well as R^2 scores for c) training and d) test set. The following paragraphs first discuss the properties of the feature sets, move on to the number of features selected, and finally elaborate on the different algorithms.

SFS, GA, and manual selection (referred to as the purple set) require far fewer rounds of Bayesian search to reach an optimum than RF, MI, and F-score (referred to as the pink set) for most pipelines. This fact is also reflected in the number of features chosen from each set. While the number of features selected from the purple set, averaging around 15, is quite

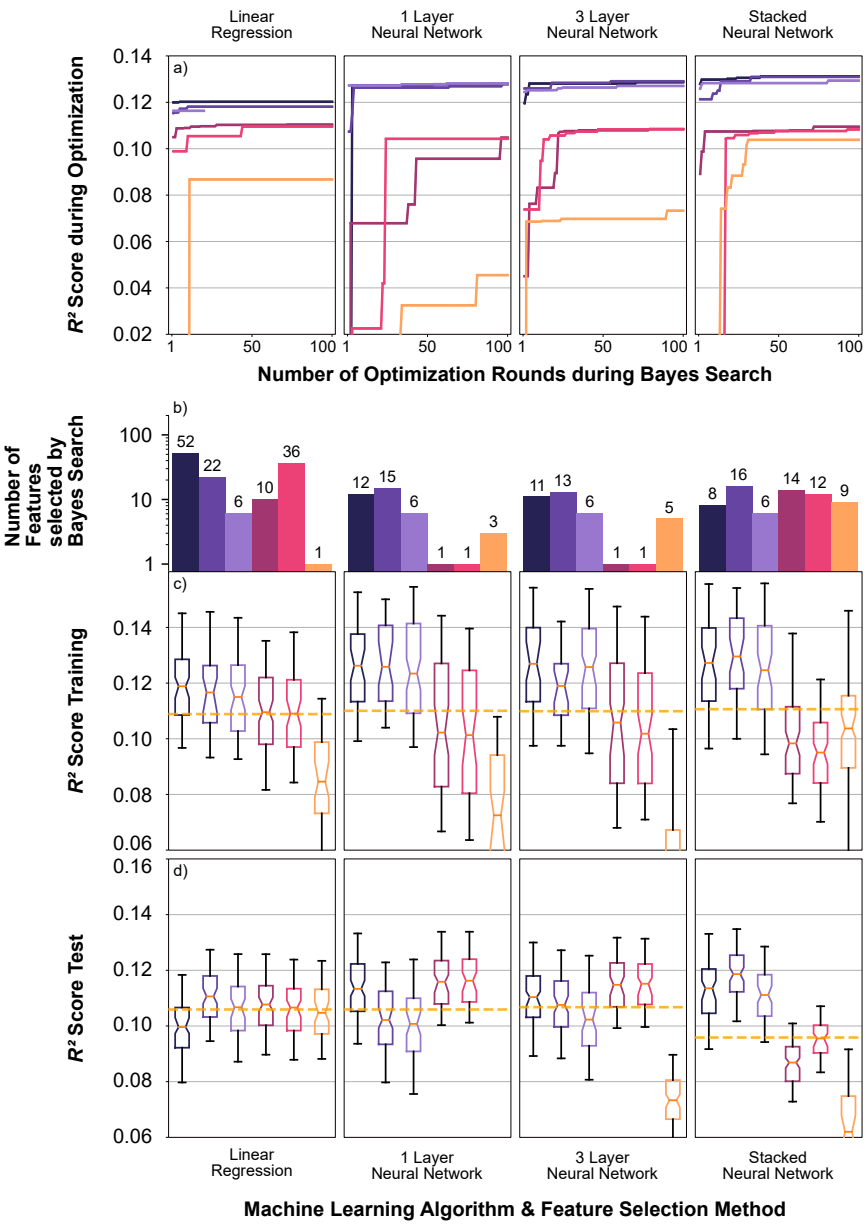
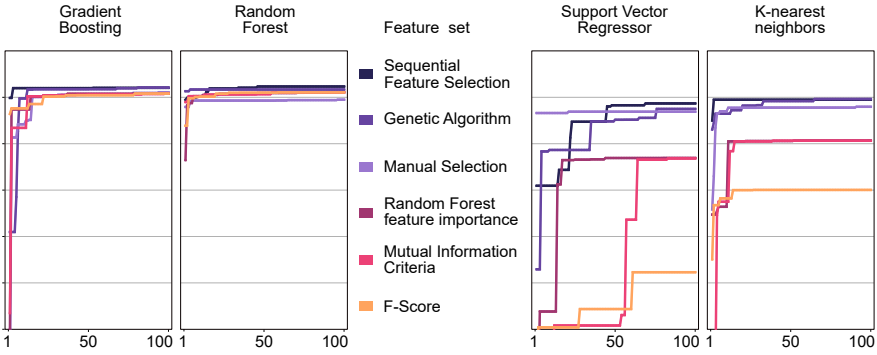
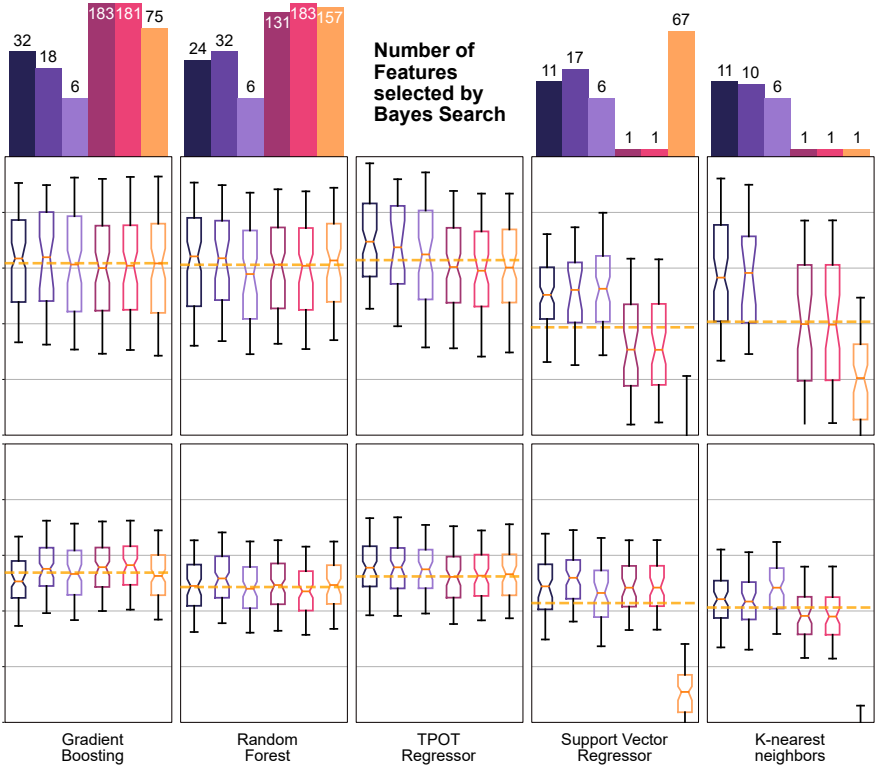


Figure 6.6: a) R^2 Score per optimization step of each ML pipeline per feature set by Bayes search, b) final number of top features selected by Bayes search, c) R^2 on training set by 5-fold CV and 50-fold bootstrap of each fold, d) same as c) but learning on training set and predicting on test set



Number of Optimization Rounds during Bayes Search



Machine Learning Algorithm & Feature Selection Method

similar for the various pipelines, two different strategies emerge for the pink set. One could either take many features and then evaluate importance (embedded methods) or only use the most important feature(s), as can be seen for NNs, SVR, and KNN, to avoid redundant information. It seems that algorithms try to compensate for the suboptimal features in the pink set by adjusting their hyperparameters but in most cases can not catch up (especially salient for the F-score) with the purple set, except for the tree-based ensemble methods that possess an internal evaluation method to evaluate feature importance. Consequently, the purple set is associated with better scores on the training set than the pink one, with particularly bad results for the F-score. It may thus be inferred that a good set of features facilitates pipeline optimization and robustness during training. Unfortunately, in this case, it does not imply better generalization to the test set, which might well be a particular problem with this entire data set, as the overall R^2 scores are very low and strongly dependent on the ML algorithm used.

The implemented ML methods can be loosely divided into three and a half categories which are discussed below from left to right: 1a) LR, 1b) NNs (i.e., 1-layer NN, 3-layer NN, and stacked NN), 2) ensemble trees (i.e., GB, RF, TPOT), and 3) other (i.e., SVR, KNN).
Compared to its nonlinear bigger brothers, LR performs astonishingly well during training and subsequent generalization to the test set, possibly due to its inability to overfit the data. The comparison also suggests that the relation between features and label is primarily of linear nature, and more complex models like the NNs can barely make use of their ability to discover and learn non-linearities. While NNs may perform slightly better on the purple training set, they are much more prone to overfitting, as seen from the corresponding test set. In fact, the opposite is true as well. Underfitting the pink training set (comprised of only one feature) may lead to better results in the test set. Overall, the stacked NN retains the best generalization score, most likely due to its ensemble nature.

The popularity of ensemble trees is readily understood when considering the minimal amount of hyperparameter optimization required to achieve excellent training and test results (with slight overfitting for the RF), regardless of which feature set they run through. Due to their embedded feature selection, they smoothly handle the pink set, although they choose to use fewer features when given a better-ranked set. However, using trees for regression comes at the cost of accepting unsmooth prediction boundaries, as we will see in the upcoming section.

The remaining methods generally exhibit the same behavior for the pink set as the NNs did and have a slightly worse performance. They do not or barely overfit the purple training set. Excuses for this shortfall for the SVR may be its concentration on points further away from the mean that it uses as support vectors but are most likely affected by stronger measurement noise[9]. The KNN simply is not constructed for regression tasks in the first place.

Sensitivity analysis

For deeper comprehension of the relationships that the model has actually learned on the training set, a sensitivity analysis is performed by holding all but one input constant (i.e., median of the respective feature distribution) and varying this feature from its smallest to its largest occurring value[10]. Because the resulting inputs are partially artificial, the feature space is additionally clustered using fuzzy c-means. The centroids of these 300 clusters can then be used as prototypical inputs to the model to gain further understanding of the true sensitivity of a particular feature variation. In the left column of Figure 6.7, 95 % of the feature distribution lies within the dark blue line, while the right column contains only these 95 %.

9 These points are less useful

10 Based on the assumption that the model output is close to a linear combination of its inputs around the median.

Additionally, the shaded areas show where 50 % (darker blue) and 100 % (lighter blue) of the learned input-output mappings lie when retraining the model 111 times by sliding a window of 1000 samples chronologically over the input of the training set. Learning from training and test set results in the dashed line. Using Gradient Boosting for prediction is indicated in orange.

Undoubtedly, the feature t_{isotherm} (i.e., deviation from reference duration of isothermal conversion in the convection furnace) exhibits the most considerable variance explaining more than 10 HV of the hardness deviation from the median with two particular outliers to the left and right. The prediction of hardness loss with increasing dwell time in the convection furnace seems justified, though the extrapolation for the converse is certainly not valid (i.e., shortening $t_{\mathrm{isothermal}}$ leads to increasing hardness), underscoring the dilemma of trustworthiness in ML models. Since the algorithm has never seen (or at least not seen enough) batches that were in the convection oven for too short a time, it cannot learn the second bainite conversion behavior properly and, therefore, extrapolates incorrectly. Consequently, the confidence in a particular prediction depends on which part of the feature space the input originates. Boundary regions with few data points may need to be made impermissible for prediction (this statement naturally generalizes to the remaining features as well).

While the relationship above was physically explainable, the remaining features have a narrow variance and contribute only single Vickers to the overall prediction. Thus, the following explanatory attempts are merely hypotheses that need to be validated by further research. Three of the

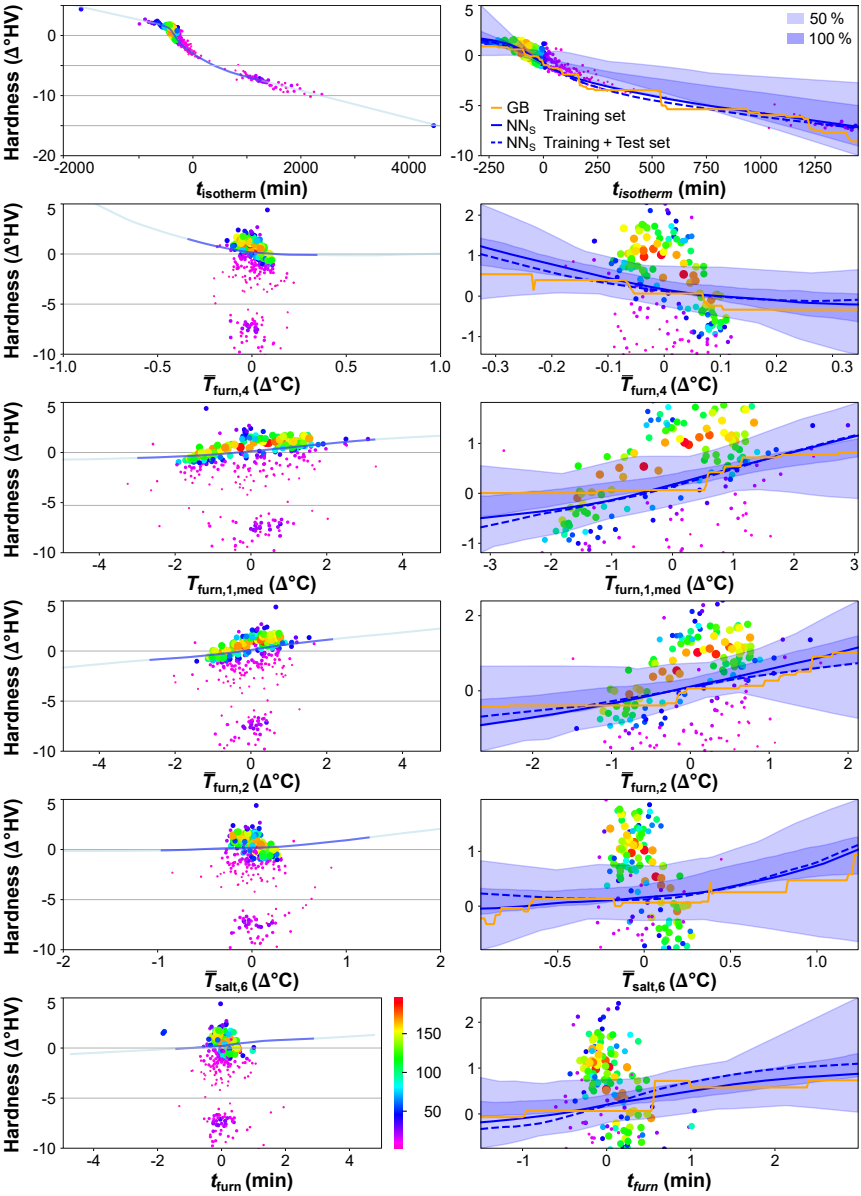


Figure 6.7: Left: sensitivity analysis of stacked NN varying one feature and holding the remaining constant (median). Dark blue line indicates 95 % of the features distribution. Cluster size is indicated by size of the circles and color. Right: stacked NN (blue), GB (orange)

four furnace features (i.e., $T_{\text{furn,1,med}}$, $\overline{T}_{\text{furn,2}}$, and t_{furn}) support the hypothesis that a longer austenitizing period leads to increased core hardness[11]. More carbides are resolved, resulting in higher carbon content in solid solution, which increases the overall hardenability, given that the microstructure had enough time for homogenization. Conversely, reducing temperature $\overline{T}_{\text{furn,4}}$ should decrease carbide solution, which, according to the model, has the effect of also increasing hardness. It may be speculated that slightly longer austenitization allows for a more homogeneous microstructure, whereas higher temperature at median time only leads to locally increased carbide solution, which might prolong the transformation start to bainite. Less austenite would then locally be transformed to the harder stage-1 bainite, leading to slightly more stage-2 bainite. This line of reasoning would be congruent with the salt bath feature $\overline{T}_{\text{salt,6}}$, where higher salt bath temperatures lead to slightly increased hardness by shortening the bainite transformation start point of stage-1, although the prototypical cluster inputs seem not to follow the sensitivity lines.

We can conclude that some of the core hardness variance can be reasonably predicted by a physically congruent ML model, with one feature doing most of the work. Yet, the low variance of most features indicates either i) that the process does indeed not generate more variance (i.e., it stems from prior processes, for instance), ii) that unmeasured process properties affect the resulting hardness, and/or iii) that the measurement error does not allow for better learning and prediction.

From Figure 6.7 it may also be inferred that different algorithm families (e.g., NN_S and GB), learn more or less physically reasonable feature-label mappings. While NNs generally learn smooth functions[12], as would be expected locally for most physical relations, the tree family learns buckets

[11] A higher temperature in sections 1 or 2 indicates that the final austenitizing temperature can be reached more quickly, thereby prolonging austenitization. In this case, a better feature would be the time spent over austenitization onset above 780 °C [26] S.55

[12] Is dependent on the activation function.

that do not appear physically meaningful for either inter- or extrapolation, as shown by the orange line. The right column of this figure also shows how the mapping may change over time, see 50 % and 100 % areas. Although the mapping direction is almost always similar, the gradient can vary greatly due to feature correlation, real physical changes, or initialization of model training, among other factors. Whether it is necessary to incorporate these temporal changes in the model training over time and to forget old data is discussed in the next section.

Rolling prediction

To understand the temporal behavior of the data-model interaction, Figure 6.8 illustrates three different training-prediction scenarios by comparing their RMSE with the SD of the measured hardness, used as a baseline:

1. Blue scenario: rolling-retrain-from-start ($\text{roll}_{\text{start}}$) mimics retraining the model every 500 batches with all previously collected data and predicts the upcoming 500 batches Then it is retrained again. The complete process was done three times.
2. Red scenario: rolling-retrain-window ($\text{roll}_{\text{window}}$) uses only the last 1500 batches to predict the upcoming 500, also done three times.
3. Purple scenario: ($\text{train}_{\text{once}}$) uses all data before the start of 2021 and then predicts 2021 without retraining.

The RMSE of the predictions is highly dependent on the current SD. As we have seen during sensitivity analysis, models learn slightly different feature-label mappings over time, but depending on the training initialization, they may also learn different mappings on the same data set. In some cases (e.g., Q4 2018, Q2 2020), it might be slightly more beneficial to forget the old data and relearn from newer data because some dependencies indeed changed (i.e., red curves mostly below blue curves in these cases). However, for more extended periods, it seems advantageous to use the entire data history for training (i.e., blue below red). Once the model has seen enough

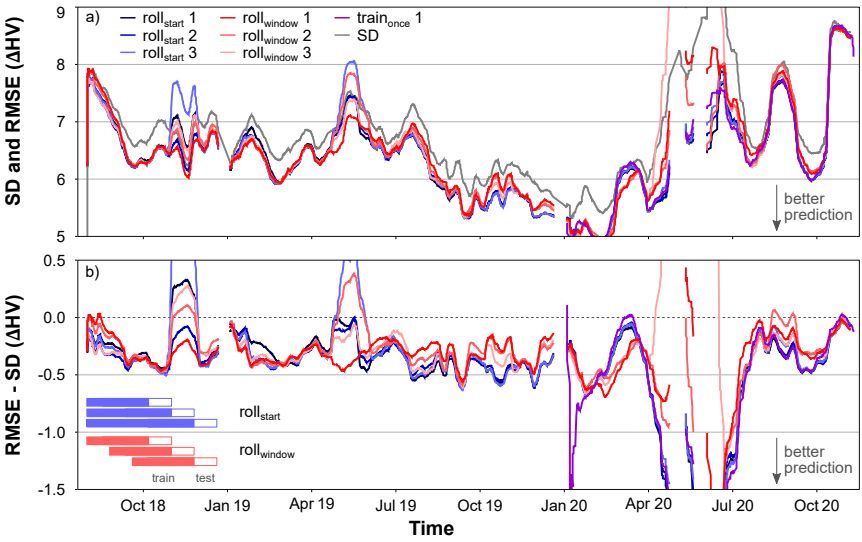


Figure 6.8: a) Standard deviation (SD) of core hardness and RMSE of hardness prediction in a rolling window of 30 days, b) difference between RMSE and SD

data, it seems there is nothing to be gained from training on more data (i.e., purple is very similar to blue. It matches the observation in the sensitivity analysis Figure 6.7 of minimal difference between training and training + test set). Regarding the implementation of such a pipeline in an actual use case, retraining seems unnecessary if a particular data threshold is exceeded, which is only valid if no process changes are anticipated in the future.

The better than usual prediction performance (i.e., comparison of blue and purple to SD) between April and July 2020 can be explained by the ramp down due to the corona crisis, which led to short-time work and lower line utilization. More batches, therefore, exceeded the usual dwell time in the convection furnace and experienced a tempering effect (explaining the higher SD) that can be predicted particularly well by the ML model.

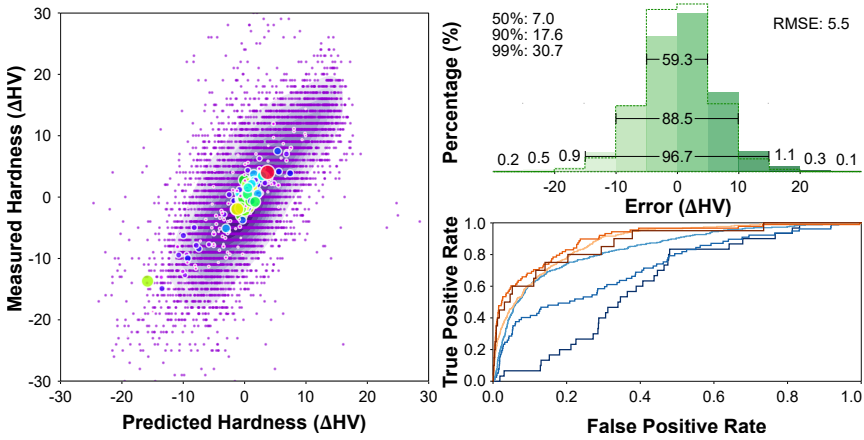


Figure 6.9: Left: scatter plot of predicted versus measured test set hardness. Right: resulting error distribution compared to derived benchmark and ROC curve for various thresholds (i.e., deviation from the distribution mean) with respective areas under the curve (AUC)

Error distribution and ROC curve

Our final evaluation reveals what can and can not be expected from the pipeline optimized above. Figure 6.9 shows a test set evaluation including a scatter plot between predicted and measured hardness, their error distribution compared to the benchmark derived in Section 4.2.3, and a ROC analysis, which are discussed in that exact order below. Scatter plot and error distribution show that a solid 88 % of predictions lie in a band of ±10 HV around the main diagonal, which in and of itself is a satisfying result. Further, when averaging predictions belonging to one cluster (introduced during sensitivity analysis), the maximal error from the diagonal drops to ±12 HV suggesting that a majority of outliers are due to measurement error or unknown influences. The closeness between the error distribution of prediction and benchmark also indicates that further optimization might be intricate.

Although these results may justifiably be called decent, assertions regarding their implication should be made with the utmost caution. This is because the exciting action takes place at the edges of the distribution where predictability plummets (e.g., for benchmark 99 % of the error lie within ∼30 HV, for prediction it is ∼42 HV). Ultimately, the algorithm is supposed to predict outlieres, that is, solve the classification problem between good head treatments and bad ones (i.e., hardness of components is to soft (below a given threshold in shades of red) or to high (above a given treshhold in shades of blue), see the left of Figure 6.9). ROC curves for the harder distribution part point in a problematic direction: the further the threshold (i.e., values beyond the threshold are defined as outliers) is from the overall mean, the more the AUC decreases. A risk assessment via the ROC curve suggests that, depending on the threshold value between 20 % and 60 % of test specimen results with measurements beyond that threshold would not be classified as outliers[13]. In sum, this means that while most predictions are quite good, those that should indicate outliers do not accomplish this task consistently, including the impossibility of knowing whether these outliers are due to measurement artifacts or truly deviate in core hardness due to unknown influences. The last section shall shed some light on at least those outliers due to process deviations.

6.2.3 Clustering and anomaly detection

As we have seen above, accurately predicting hardness values is rather difficult, and the imbalance of good to near-tolerance parts, as well as measurement error, makes finding parts that are out of specification an even more daunting endeavor. This section approaches the task by unsupervised clustering of process data using fuzzy c means. Outliers may

[13] When accepting a 10 % false positive rate and ignoring the uppermost threshold with very view samples.

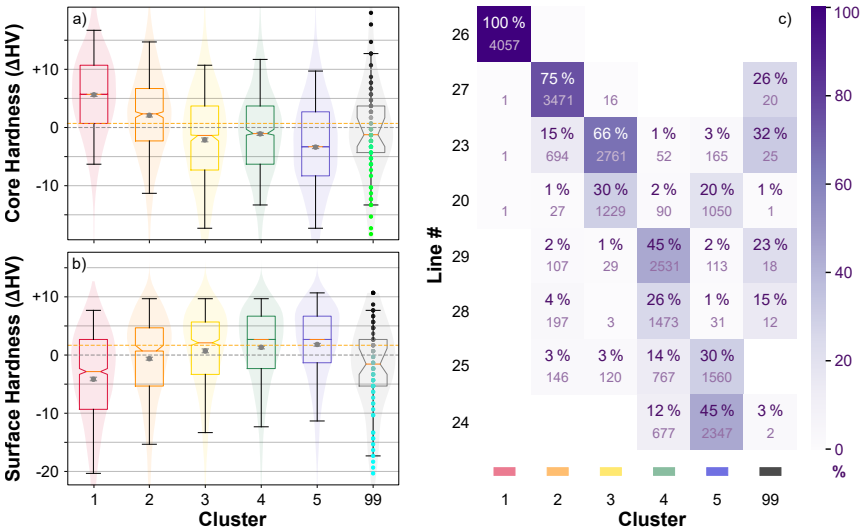


Figure 6.10: Box plot of a) core and b) surface hardness for each cluster found by fuzzy c-means. Colored points in cluster 99 represent the hardness of the 78 individual samples that belong to this noise cluster, c) heat map that shows the percental composition of each cluster by line along with the absolute number of samples in braces below

then be identified as those samples that are too far away from any cluster centers.

The clustering is applied to the resampled time series of temperature, mass flow, and C-level in the furnace as well as temperature in the salt bath. Five groups were determined by increasing the number of clusters incrementally until the algorithm sorts significantly fewer samples in the marginal cluster. Samples whose highest probability of belonging to a cluster was less than $p < .24$ were sorted into outlier[14] cluster 99. The box plots in Figure 6.10 display the core and surface hardness distributions along with line affiliation as heat map of every cluster. Two observations

[14] The value of p was chosen based on the distribution of all p values, where values smaller $p < .24$ visually displayed an outlier set.

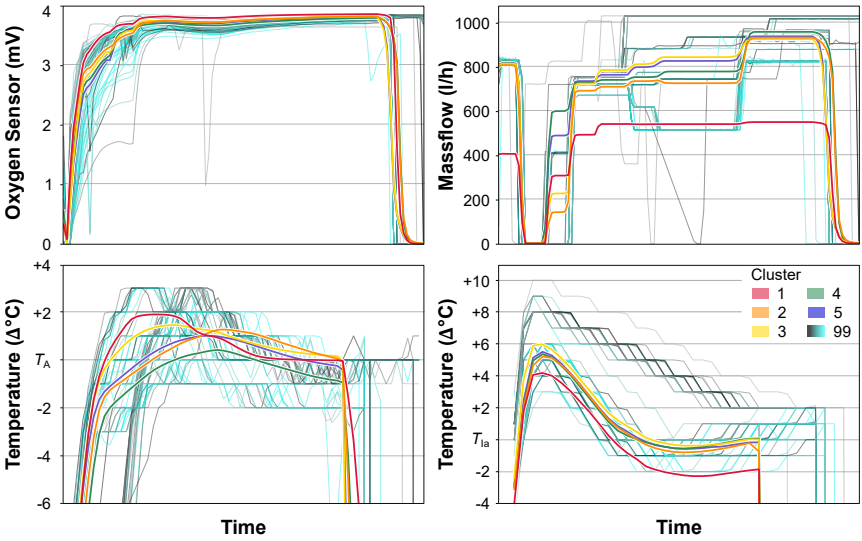


Figure 6.11: Centroids of the 5 clusters found by using the inputs temperature, mass flow, and C-level in the furnace as well as temperature in the salt bath

immediately catch the eye. First, unexpectedly, the distribution of the outlier cluster is similar in variance and even closer to the overall average than the five main clusters. Because this cluster contains samples that went through a process maximally dissimilar to the common procedure, it would have been expected to have much broader variance[15] with hardness values much closer to the tolerance limit. Based on this observation, the process robustness may be considered remarkably high, which is also evidence for the impracticality of finding outliers based on the process data assessed to date. Second, clusters are strongly associated with particular lines, as indicated by corresponding colors. While cluster 3 (yellow) consists of 96 % line 20 and 23, cluster 1 (red) even entirely consists of line 26 samples. Thus, the box plots mainly show the joint hardness distribution of two lines, respectively, which are very similar in their process behavior (conf.

[15] Or even show a bimodal distribution with peaks at "good" and "bad".

Section 5.2.2). Furthermore, it can be clearly seen once again that the process variation between lines is much higher than within individual lines over time. In this case, a precise hardness prediction within a single line is more difficult or rather the prediction of the deviation from the current mean hardness of a line. In order to evaluate whether hardness differences between clusters may still be attributable to process dissimilarities, their centroids are shown in Figures 6.11 and 6.12 along with the outliers in cluster 99, suggesting that non-arbitrary differences between lines exist.

The hardness distribution of core and surface may be explainable by focusing on the five main centroids. Comparing the yellow and red clusters, the latter shows the lowest quenching temperature, which may lead to greater hardness in the core but also slows down transformation to bainite[16] at the surface, thus, the lowest hardness there. This resembles the effect found for different test piece positions in Section 4.2.2, where pieces lower in the batch are quenched faster and become harder in the core and less hard at the surface. The lower enrichment gas flow in the red cluster presumably also contributes to its decreased surface hardness. Although it also measures the highest C-level, the difference between carbon content per cluster is much smaller than mass flow.

While the outliers support the mass flow hypothesis (i.e., darker outliers are harder and have an increased mass flow), they strongly contradict the quenching hypothesis. Higher salt bath temperatures are associated with both higher core and surface hardness samples from cluster 99. Nevertheless, it should be safe to assume that the findings above are valid despite the outlier behavior because these samples are anomalies and few in number. In summary, although it is relatively easy to find these process-related outliers, predicting their behavior in terms of hardness is not feasible at this time, reinforcing the previously proposed guideline that predictions may only be made for a batch whose process parameters fall within a

[16] Although lower quenching temperatures can lead to greater hardness at the surface, this is only the case for a complete transformation to bainite at that temperature.

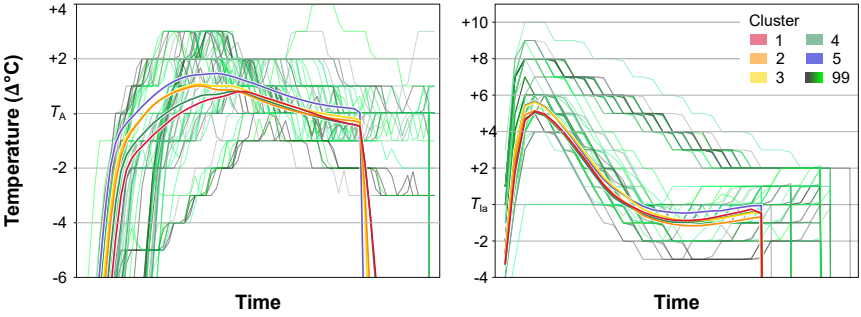


Figure 6.12: Centroids of the 5 clusters found by using temperature in the furnace and in the salt bath

defined confidence window. Our second use case poses a quite different challenge regarding process influences, as will be discussed below.

6.3 Case Hardening

6.3.1 Forecasting and label tracking

In the following, the generalizability of the modeling approach from the previous section to the case hardening use case is investigated. Analogous to the bainitization use case, data were divided into training and test set (i.e., training data from before 2017 (70 %), test data thereafter), then feature and label correction, selection, and optimization of several models was performed. Among the most important were always features related to quenching and categorical features such as station or component type. When given the opportunity, selection algorithms rarely selected features that were corrected by a filter. While training scores ranged from R^2=0.1 to 0.2, the test set did not even hit the dummy regression mark of zero, clearly exposing the non-generalizability of the models to newer data. As seen in the last chapter, stations change their behavior over time which the standard ML algorithms could not capture. Tracking fluctuations of

influences on the case-hardening process necessitates a more elaborate approach since batches may take different routes, components geometry has an impact on measured hardness, and test specimens come from different batch positions. However, hardness differences due to component type or batch position may be assumed to be static (i.e., barely change over time).

Model Selection

Hence, a custom model was created (detailed in Section 3.4) that accounts for the dynamic behavior of the stations by tracking their hidden states and correcting for component and position offsets with respective variables. Interpretability of such model parameters is then much better and optimization time reasonable. At this point, the interested reader may already speculate about the possible use of RNNs known for their ability to handle dynamic time series, which were also tested and could in no way match the performance of well-calibrated filters with additional offsets accounting for categorical influence. Accordingly, this chapter will elaborate on the hidden states pipeline's optimization and evaluation, ending on a surprising note concerning the station's influence.

Optimization

Close to the supposed use case, only every second measurement was used as input during training of the model to then predict its respective successor, as was specified in Table 3.12. Therefore, it is a forecasting model that uses its actual state to predict the hardness outcome when a specific component at a specific batch location takes a particular route through the stations before undergoing heat treatment.

Optimization of the custom hidden states pipeline was performed on the training set in the following way:

Table 6.3: Optimal parameters for the custom hidden states pipeline, introduced in Section 3.4. Optimization was performed via differential evolution and dual annealing(*) on training data. Sea: Seat middle, Sho: Shoulder, Sha: Shaft inside. V: vacuum furnace, F: deep freezer, T: tempering furnace, p_i and c_j are in ΔHV. The color is intended to aid the reader in recognizing the different magnitude of values within each parameter group (i.e., a, b, p and c), where darker color generally indicates smaller or negative values

		Filter Parameter							Positions				Components								
	RMSE	a_{Base}	a_V	a_F	a_T	b_V	b_F	b_T	p_1	p_3	p_4	p_6	c_{A1}	c_{A2}	c_{B1}	c_{B2}	c_{C1}	c_{C2}	c_{D1}	c_{D2}	c_{D3}
Score 0.1	6.47*	0.95	0.99	0.45	0.99	0.01	0.04	0.01	-0.9	0.2	0.5	0.2	0.7	-1.3	-0.3	-0.5	-0.3	-0.4	-0.4	2.0	1.3
	6.47	0.95	0.99	0.43	0.99	0.01	0.04	0.01	-0.9	0.1	0.5	0.2	0.7	-1.2	-0.3	-0.5	-0.3	-0.4	-0.5	1.9	0.7
	6.57	0.98	0.24	0.56	0.94	0.05	0.02	0.03	-0.9	0.6	0.1	0.3	1.2	-1.0	-0.4	0.4	-0.2	-1.1	0.7	1.6	0.8
	6.58	0.96	0.48	0.26	0.98	0.07	0.04	0.01	-1.1	0.0	0.9	0.2	0.2	-1.9	-0.6	-1.4	1.1	-0.2	-0.5	5.6	-1.4
Sea1	8.18	0.97	0.82	0.38	0.92	0.03	0.04	0.04	0.1	0.8	1.2	-2.1	1.7	-1.7	-0.6	-1.4	1.8	0.2	1.9	4.8	4.1
Sho1	8.72	0.96	1.00	0.18	1.00	0.00	0.01	0.00	-2.2	-0.8	0.4	2.5	-0.1	-0.2	-0.3	-0.4	-0.3	-1.1	-2.1	4.6	-0.6
Score 0.4	9.01*	0.95	0.96	0.39	0.34	0.04	0.05	0.05	-4.0	5.0	-0.8	-0.2	8.7	-5.5	-2.8	-3.4	-2.9	-3.4	5.9	-1.0	-1.6
	9.02	0.94	0.98	0.28	0.27	0.02	0.05	0.05	-4.0	5.0	-0.8	-0.2	8.7	-5.5	-2.8	-3.4	-2.9	-3.4	5.9	-1.0	-1.6
	9.07	0.94	0.98	0.29	0.79	0.02	0.08	0.02	-4.1	5.0	-1.0	0.1	8.3	-5.8	-2.7	-4.1	-4.1	-3.5	4.1	-2.3	-1.0
	9.07	0.97	0.95	0.41	0.35	0.04	0.06	0.07	-3.4	5.1	-1.3	-0.5	8.6	-4.9	-2.6	-4.3	-2.6	-3.3	8.2	-1.4	-4.1
Sea4	11.73	0.96	0.97	0.42	0.39	0.03	0.01	0.05	-5.2	5.0	-0.7	1.0	18.2	-9.1	-6.4	-6.5	-6.0	-9.9	9.9	-1.0	-2.4
Sha4	9.80	1.00	0.90	0.58	0.58	0.07	0.08	0.05	-3.5	4.5	-0.6	-0.4	-1.1	-2.1	-0.5	-0.8	0.5	1.0	1.5	1.9	0.9
Score 0.7	7.80*	0.87	0.99	0.18	0.04	0.01	0.04	0.02	-0.9	1.1	-0.4	0.3	-0.4	-0.3	2.6	3.2	-0.5	0.2	-0.2	0.3	-0.8
	7.80	0.87	0.99	0.17	0.04	0.01	0.04	0.02	-0.9	1.1	-0.4	0.3	-0.4	-0.3	2.6	3.2	-0.5	0.2	-0.3	0.2	-0.8
	7.83	0.88	0.99	0.30	0.18	0.01	0.06	0.01	-0.8	1.1	-0.4	0.1	-0.4	2.0	2.5	2.6	0.2	-0.3	-0.7	0.8	-2.8
	7.92	0.91	0.51	0.15	0.53	0.10	0.06	0.01	-1.5	1.8	-0.2	-0.1	-0.2	-1.1	2.4	3.5	-2.5	-1.6	0.2	0.0	0.0
Sha7	8.60	0.90	0.98	0.44	0.00	0.02	0.03	0.04	-2.2	4.2	-1.7	-0.3	-1.3	-1.4	5.2	5.7	-0.6	-0.1	-0.4	0.0	-0.8
Core	9.62	0.90	0.56	0.18	0.15	0.02	0.04	0.00	0.3	-2.0	0.8	0.8	0.5	0.7	0.0	0.8	-0.3	0.6	-0.1	-0.1	-0.9

- Row 1, 7, 13: Optimization of Score (0.1, 0.4, 0.7) via dual annealing
- Row 2-4, 8-10, 14-16: Optimization of Score (0.1, 0.4, 0.7) via differential evolution. That is, a total of three times per score
- Row 5+6: Optimization of Seat middle 0.1 and Shoulder 0.1 via differential evolution
- Row 11+12: Optimization of Seat middle 0.4 and Shaft 0.4 via differential evolution
- Row 17+18: Optimization of Shaft 0.7 and Core via differential evolution

The resulting parameterization is shown in Table 6.3 which is examined in the following from left to right. Generally, dual annealing takes longer[17] but is always as good or better as the best differential evolution result, where optimal parameters for the best results (i.e., first two lines of each score) are almost equal, despite the very different optimization approaches. Interestingly, the model performance often barely changes for parameter deviations from that optimum (e.g., a_V=0.24 for Score 0.1), hinting at the slow sensitivity of some features explored in the upcoming section. Tracking the base variation (e.g., influences by the material composition or processes previous to heat treatment) is best done with a slow filter (i.e., high a_{Base}), as expected, where Score 0.7 exhibits the fastest version, likely because material composition leads to the most substantial fluctuations in that depth. Also, decreased hardness in that depth may lead to less distorted measurements such that the filter does not have to rely on memorized values as much to smooth out errors. Filter parameters[18] of the stations give a mixed picture. The influence of the vacuum furnace seems ambiguous (i.e., mostly high with some outliers) while tracking the freezer seems hardly worthwhile. Surface near measurements are affected significantly by the tempering furnace behavior, while deeper layers largely remain unaffected.

Position and component parameters reveal one major drawback of using combined scores. While the scores consistently have smaller RMSEs than the measurement positions they were calculated from (mostly due to mitigation of measurement error), they lose the ability to account for the partially very different behavior of the underlying measurement position. For example, component type affects hardness very differently for seat

[17] Strongly dependent on the setting of the optimization algorithm, including the maximal number of iterations, initial conditions, and convergence criteria.

[18] As a reminder: the base state is updated with a_{Base} and $b = 1 - a_{Base}$, while the sum of coefficients for each station type may be < 1, thereby indicating the relative importance of a specific station type (i.e., when $a_i + b_i$ is small, relative contribution of station type i is small).

middle 0.4 $c_{A1} = 18.2\,\Delta\text{HV}$ and shaft inside 0.4 $c_{A1} = -1.1\,\Delta\text{HV}$. Expectedly, individual geometries in particular influence Score 0.4, mainly because of the shape of the nozzle tips, as shown in the last chapter. To see how these parameters generalize to unseen data, the following section evaluates training and test set.

Evaluation

Results of training and test predictions using the custom hidden states pipeline are shown in Figure 6.13. Again, combined scores have lower RMSEs for all depths then the original measurement positions, where overall lower RMSE is generally coupled with a higher R^2, as expected. Contrary to intuition, test results often show better performance (i.e., higher R^2 and lower RMSE) than their corresponding training sets. This behavior is readily explained by the chronological train-test split where the test set includes the diamond change (leading to less scatter) and recalibration of the measurement devices (leading to a large change in mean hardness)[19], cf. Section 4.3.2.

For a better interpretation of the model performance, we compare it with the results from measuring two components in one batch and then predicting one component from the other, cf. Section 4.3.3, shown in the figure as triangles, pointing in the direction of the better performance. Overall, model predictions are as good or better than the triangle benchmark which means, that knowing the approximated immediate past mean hardness of a meas. pos. might be better than the measurement of one test specimen

[19] The spread between measurement values is artificially enlarged by this recalibration. Small process or component type errors do now have a smaller share in the overall distribution. Because filters can track this change, they reach a higher R^2 score because what they can measure (fluctuation) now has a bigger contribution to the hardness variance.

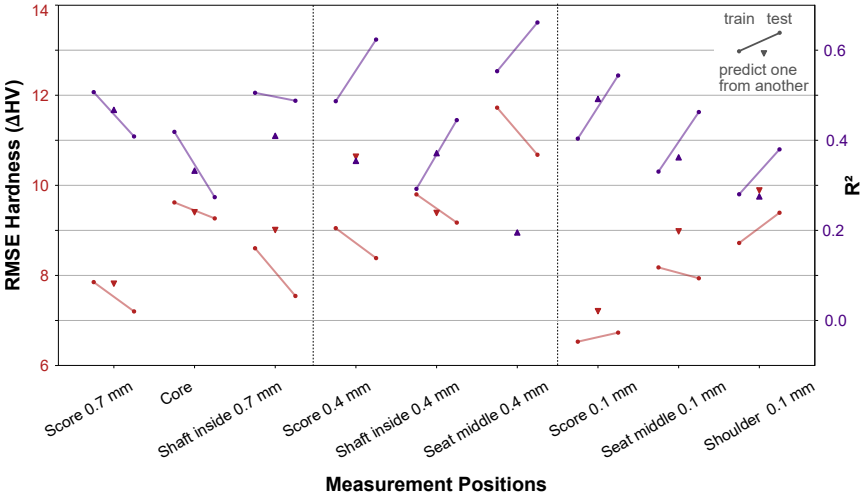


Figure 6.13: Prediction results (i.e., RMSE in red with y-axis on the left and R^2 score in purple on the right y-axis) for training and test set for different measurements position as well as associated scores

of a batch to make a prediction about the hardness of the second test specimen[20]. Admittedly, it is not an entirely fair comparison because the LR was given no information about the component type, which is particularly apparent for depths of 0.4 mm, where model predictions are much better than measurement predictions and components type has a significant influence. It also explains why the model on average exhibits a higher R^2 score for 0.4 mm, because it can make good use of the component type information, as compared to 0.1 and 0.7, which is also apparent in the behavior over time shown in the next section.

[20] The reason for this is that the large drifts over time can be tracked by the filter and used for prediction. The LR, on the other hand, needs to deal with all of the measurement errors and has no information about the local temporal state of a meas. pos. Such findings further worsen the trustworthiness of a single HV 1 measurement.

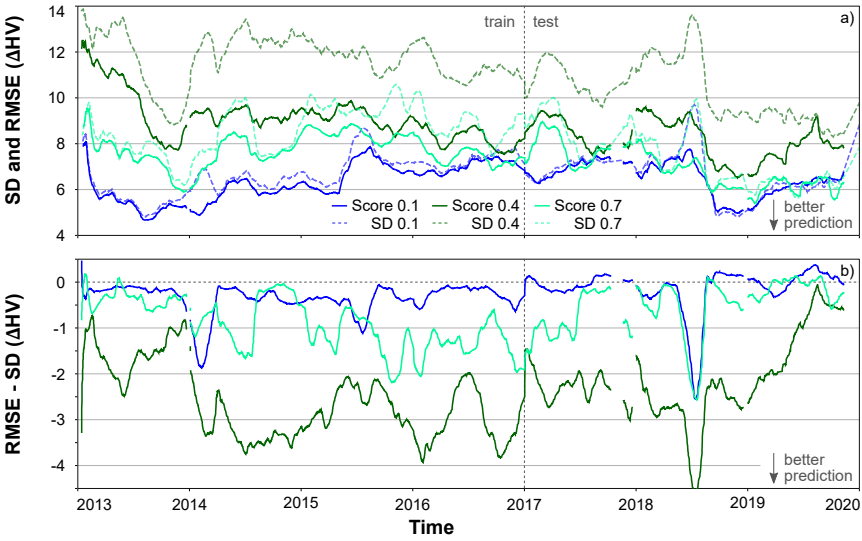


Figure 6.14: a) Standard deviation (SD) of scores and RMSE of hardness predictions in a rolling window of 100 days, b) difference between RMSE and SD

Rolling prediction

A comparison of the models' RMSE with the SD of the labels over time reveals what can actually be predicted apart from the base variation. Figure 6.14 shows the dynamic behavior of the measurements' scatter over time with no particular correlation between labels but a clear drop in 2018 (i.e., diamond change), explaining the lower RMSE of the test sets from above. It turns out that for Score 0.1, there is nothing much to be predicted, most likely because the carburization levels the playing field (i.e., very low influence of material composition), the SD is already low, and the measurement positions have the greatest hardness, increasing the probability of high measurement error. On the other hand, for Score 0.4, component type and position give huge leverage for better predictions. It also starts from a much worse SD, leaving plenty of room for improvement.

Seemingly, models generalize well from training to test, although the end of 2019 foments some skepticism, with strongly degrading performance. While the influence of the component type can be assumed to be stationary for some time, we also saw in Section 5.3.3 that the hardness drifts slightly over time, likely explaining the decreased performance in 2019. If components react differently to changes in one of the stations, this offset has to be relearned, or the component type be made a hidden state in the model[21]. Generally, it seems like sound advice to continuously access performance when running models for a prolonged period of time (e.g., over a year) and recalibrate parameters in case of degradation. Lastly, performance loss may also be experienced when further reducing the number of test parts or predicting too far into the future, as shown in the next paragraph.

Reducing parts

Figure 6.15 assesses the information availability influence on model performance. Two trends are immediately visible. Predicting further into the future worsens performance, while some amount of phase delay correction (i.e., knowing the hardness of a successor batch before predicting the previous one) increased the R^2 score. Performance also deteriorates for higher n when only testing each n^{th} component or batch. However, the loss seems to be relatively moderate even after quartering the testing efforts[22] which might, therefore, be a reasonable cost reduction strategy. Moreover, even after reducing the tests to each eighth sample for the combined score (e.g., score 0.7) is better than using every second sample with only the individual measurement positions. Still, caution should be exercised when using such a method (i.e., skipping test specimens) because components of different

[21] In fact, this seems unreasonable and may not even improve the performance of the model due to the large number of different component types and the then irregular updating of these states.

[22] Which would be equivalent to testing one nozzle body of every second batch.

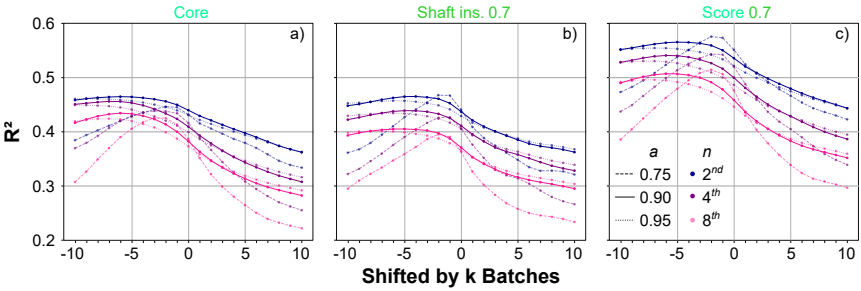


Figure 6.15: Causal first-order IIR filter applied to every 2^{nd} measurement piece. Predictions were shifted by k batches. R^2 score for chosen meas. pos. with different a and use of every 2^{nd}, 4^{th}, and 8^{th} test specimen

types may then be tested at different batch locations that may not be generalizable to other types. Lastly, faster filters (a=0.75) exhibit a higher vulnerability to improper phase delay correction, and the recommended action derived therefrom might be to use a slower filter at the cost of a slight performance loss, as was suggested by the optimization algorithms.

6.3.2 Analysis of variance

Finally, this section breaks down the variance into contributing factors and their relative importance to the deviation from the mean of the label distribution for each score, see Figure 6.16. The bars show the total contribution in HV, that is, how much of the error can be better predicted than using a dummy regression (predicting the mean of the label distribution). Pie charts show the relative contribution to the variance with the predictable R^2 score in the center. The R^2 itself is not coherent (i.e., $R^2(A) + R^2(B) \neq R^2(A\&B)$), which means that contributions may be overlapping and should be seen as rough estimates[23].

[23] Shares were determined by calculating the R^2 score after using all but one specific feature and by only using that specific feature.

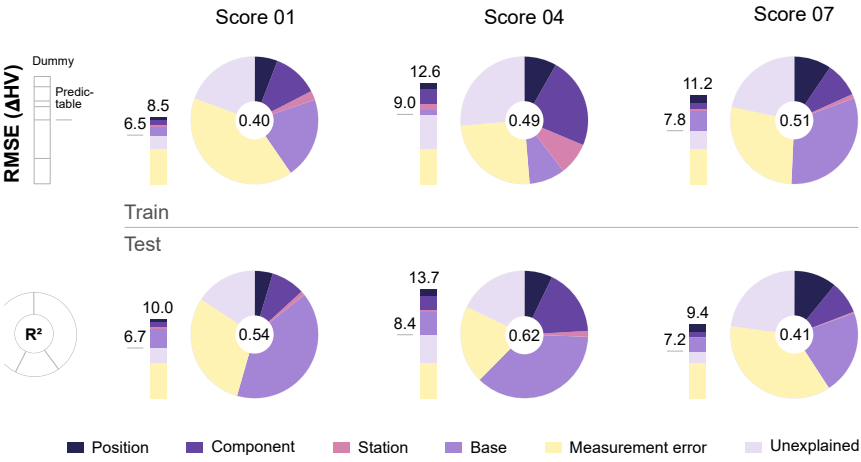


Figure 6.16: Contribution of various sources to variance of label distribution

The measurement error (yellow) with an RMSE of 4.5 is taken from Section 4.3.4 and gives a lower bound of the measurement and specimen preparation influence. It may be assumed to be comparable for the different scores and, therefore, has the same size in every bar, although its relative contribution changes significantly depending on the size of the remaining influences. Station (i.e., the heat treatment process itself) has the smallest predictive power explaining why the RMSE scores in Table 6.3 are insensitive to changes in filter parameters of the freezer, vacuum, and tempering furnace. After all these analyses, it turns out that the process is so stable that label variation does stem from everything but heat treatment. For this reason, the use of a complicated hidden states model, besides being interesting, is not necessary to obtain accurate predictions. Batch

position and component type are better predictors. They have more influence than process variations between batches themselves[24]. In most cases, however, knowing the baseline fluctuation provides the most explanatory power for label variation, although its relative contribution might vary quite strongly over time, as can be seen between training and test set for Score 0.4. In sum, most of the variance can be explained, with about half being predictable by the model. The last chapter presents the respective recommendations and conclusions that may be drawn from these analyses, as well as their applicability in daily work.

6.4 Discussion

While it is relatively easy to track the average hardness state influenced by the line or furnace and the alloy composition over time using an IIR filter, in industrialized stable heat treatment processes, it is generally difficult or even impossible to predict the hardness changes caused by the small random variations in heat treatment parameters from day to day. If a prediction (or forecast) is to be made for the process, a few well-corrected features are the best choice, with a chronological train-test split being mandatory to detect overfitting. Predictive performance must then always be reported over time to capture differences between train and test set, changes in measurement (or process) variance affecting the RMSE, and changes in long-term fluctuation affecting the R^2 value (a better score not always implies a more accurate prediction). Regarding the potential for improvement in hardness prediction, either there are too few well-calibrated, high-resolution sensors mounted in the line to detect these changes, or, more likely, the process itself really does not contribute significantly to

[24] It goes without saying that not position and component type per se are responsible for the differences, but differences in local temperature, process gas composition, and local quenching intensity. In this respect, the local process is responsible but can only be represented here by position and type. If the local data were available for each process, other predictions would certainly be possible.

hardness variance (with the exception of increased convection oven dwell time on weekends). More meaningful are the component type, batch position, and the number of samples used to update the filter, which means that hardness testing can never be replaced entirely by predictions since too many factors are involved that influence the hardness result. Furthermore, hardness tests not only indicate successful heat treatment but can also reveal irregularities in the preceding process chain from which no data are (or even can be) available, such as the accidental mixing up of material. Ultimately, the measurement error determines the upper bound on prediction accuracy, as it is itself one of the most significant contributors to the overall variance.

Machine learning models must be trained with caution as their tendency to overfit the data may lead to erroneous predictions for future unseen data. As counteraction, initially, simple models are to be trained with a chronological train-test split and few, well-selected, physically understood features. Although ensemble methods (e.g., random forest or boosting tree) may lead to slightly better predictions, training a simple NN and a linear regression model may be the better choice. The NN is preferable to the LR only when its performance is significantly better in the test set. Compared to ensemble methods, NN and LR have better interpretability and learn smoother functions with respect to the true physical system properties. Predictions that extrapolate from the learned distribution, particularly for such nonlinear processes as heat treatment, should be averted. In short, machine learning is a sharp sword to be employed carefully.

These findings beg the question of whether an economically viable cost reduction strategy can be derived that trades-off the savings from reduced testing with the consequences of producing out-of-spec components multiplied by the probability of not detecting them (due to reduced testing). Empirical risk assessment is quite problematic because none of the labels in the data set were out of specification (in terms of hardness), and the measurement error is higher for near-tolerance values (i.e., it is difficult

to know if a measured hardness is really out of specification or only very poorly measured). The answer to this question depends, as so often, on the individual risk aversion of the production manager and the quality department. The results suggest that halving the testing effort is well worth it in our specific use cases. The methods presented above may not find the outliers, but then there are also no outliers to be found. What the methods can and should be used for is to track the individual features and labels. As a result, critical trends are easy to identify, deviations from the norm are immediately visible[25], and the influence of line, type, and batch position can be handily taken into account when evaluating a particular measurement result. In order to reap the fruits of the preceding endeavors, some final hurdle must be overcome, that is, deploying the methodology into daily production.

[25] When a bad part is found, it is relatively easy to determine the likelihood of whether a defect occurred during heat treatment or whether the problem lies elsewhere.

7 Deployment

An industrial machine learning project may generally distinguish between three phases: development, deployment, and operation, see Figure 7.1. While the development cycle, a.k.a. CRISP-DM (CRoss-Industry Standard Process for DM) [107] was the primary focus of this thesis up to this point, addressing the business case, data collection, analysis, and understanding, as well as model development and validation, the latter two are elaborated on in this chapter. They are concerned with model integration into day-to-day operations, requiring an IT infrastructure that enables

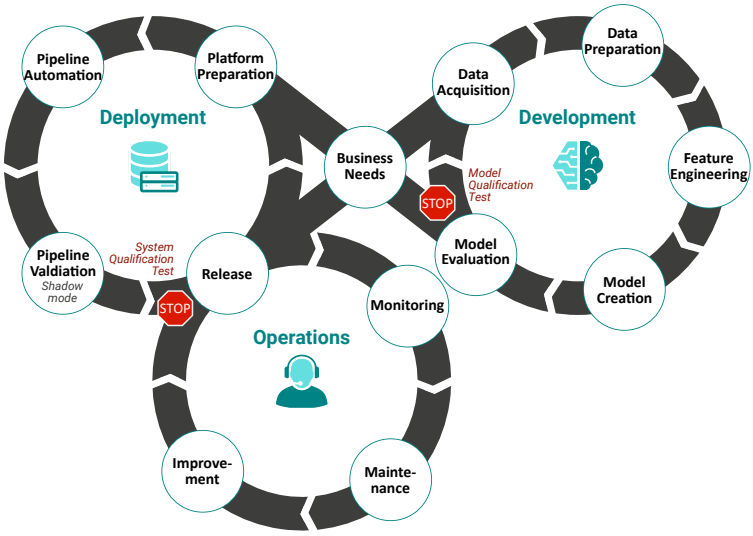


Figure 7.1: Schematic development-, deployment-, and operations-cycle as used in the current project, based on [107, 136]

easy access, monitoring, error logging, maintenance, and scalability. These three stages are often referred to as MLOps when the primary goal is to use machine learning models.

Often enough, specifically, the integration efforts are underestimated because necessary resources such as clean data, knowledge of deployment platforms, databases, source code management, and versioning, as well as continuous data flow, are rarely covered by the data scientist who developed the model, and should be consummated by an information or computer scientist [92].

For this thesis the system was fully implemented (including platform preparation and pipeline automation) up to the Pipeline validation in shadow mode. That is, the system can be monitored already and runs on the production system, but release to replace test specimens has not yet been granted. The following sections dive into the deployment, explaining which steps were taken to implement the models developed in previous chapters into daily business.

Platform preparation lies at the heart of a successful deployment for continued stable operation. The IT infrastructure used for this project is explained in more detail below and shown in Figure 7.2. Choosing an integrated development environment (IDE) certainly is a matter of personal preference. For a tight budget Spyder seems better suited for the data sciences task due to its easy access to variables and layout proximity to Matlab and R Studio, PyCharm and Visual Studio might be the better choice for deployment since debugging and including a source code repository is easier. If affordable PyCharm Professional seems to be the best choice. This IDE is also used more often by computer scientists. For versioning and traceability of code changes, any major Git-based repository is sufficient (most companies have a subscription to the professional version of the major brands).

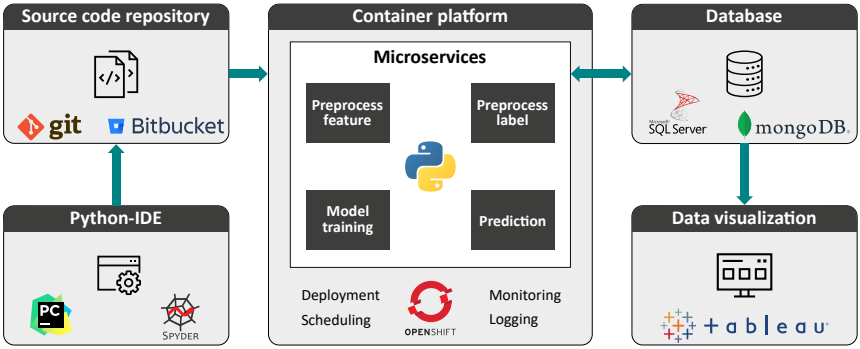


Figure 7.2: Deployment and operations framework

Choosing a deployment environment (e.g., local machine, virtual machine, or a container platform — in our use case *OpenShift* by Red Hat) requires a little more thought as it depends on the available resources project scope and requirements for the future. While deployment on a local machine is relatively easy to accomplish, maintenance, stable operation, and scalability are horrendous (e.g., system updates, accessibility for programmers, and hardware stability are just a few of many caveats), thus this choice is only suggested as a stopgap solution for a temporary rapid prototype or when resources are scarce. Virtual machines (VMs) and container platforms alleviate most of these pain points, with their own pros and cons: VMs are a well-understood industry standard that emulate an entire computing environment and provide more security through this encapsulation. Unfortunately, they take longer to boot, backup, or migrate between platforms. In addition, their images typically consume gigabytes (i.e., a physical server can support fewer VMs than containers). Containers are more lightweight (i.e., image in the megabyte range), spin up in milliseconds, and require fewer IT resources to deploy, run, and manage. On the other hand, all containers must run on the same operating system, are somewhat less secure, and operate in an evolving ecosystem due to the novelty of the technology.

Database selection should be application-dependent but is commonly dictated by available resources. For both use cases, a non-SQL database (i.e., Mongo DB) was used for bainitization and an SQL DB (i.e., Microsoft SQL Server) for case hardening. While deleting entries, creating new collections (e.g., using Robo 3T), and dumping data in any format into a Mongo DB is fairly easy, its speed (i.e., retrieving and filtering data) seems comparably slow. In contrast, an SQL DB enforces certain data types and structures, enhancing their integrity and considerably speeding up retrieval. That means more thought must be put into how to store the data into an SQL DB, but this may save a lot of time later on. In addition, SQL is widely known and well documented, making it easier to put the query burden on the database than loading more data and then filtering with Python commands. Both DBs allow to either store the complete model as a binary file or their weights. However, due to higher speed, data type enforcement, and tabular structure, the SQL version is recommended, as it often also makes visualization much easier, as most such tools (e.g., tableau) expect tabular data or even have their own SQL interface. For this thesis, 4 tables (resp. collections in the MongoDB) where created: Label, Feature, Prediction and model.

The Label table holds one measurement value per row along with measurement type, unit, the component type tested, time stamp, and unique ID that links to a specific batch. In addition, it holds the filter output calculated[1] from previous labels of the same component type and salt bath line (or furnace) as an indication of the current hardness state. Although it increases the table size, it is essential to have each measurement in a single row because data can be processed and filtered much easier. Moreover, new types of components and measurements can easily be entered into the table, making the generalizability of the concept to new components, measurements, or furnaces much easier.

[1] These values are calculated by microservices explained in the upcoming section.

Generally, the same would be valid for the Feature table, but putting each feature in an individual row would overload the database. Consequently, the number of features to be stored in the DB must be known beforehand for the SQL Database. Here, MongoDB can play out its full advantage since new features can be added to a collection at any point. However, this functionality should not be overused, as dissimilar structures between categories (e.g., components, furnaces) lead to more complicated, error-prone processing.

The Prediction table is similar in structure to the Label table, with one prediction in each row. Additionally, it holds the information with which model the prediction was made.

The Model table holds the models used for prediction, along with a timestamp of its last training, a timestamp of expiration (i.e., if retraining is necessary every six months, it is reflected in this timestamp), as well as the features used for the model and the training scores.

Pipeline automation involves several tasks. Splitting the tasks at hand (i.e., preprocessing, model training, and prediction) into microservices (i.e., independent applications) and running them on the OpenShift platform provides a most resource-efficient and stable deployment. It already integrates monitoring and logging as well as scheduling (i.e., each microservice - except for model training - is executed every 15 min), see Figure 7.3. This makes the maintenance of individual modules much more effortless and enhances overall system stability. In order to find out which data to process next (each heat treated batch has its unique identifier), each microservice possesses its own table or collection[2] in a database and compares the entries it has already made to its designated table with any new data available. This way, even if a service breaks down, it can easily be reloaded and take of from the last new ID.

[2] In a non-SQL database a collection is the equivalent of a table in an SQL DB.

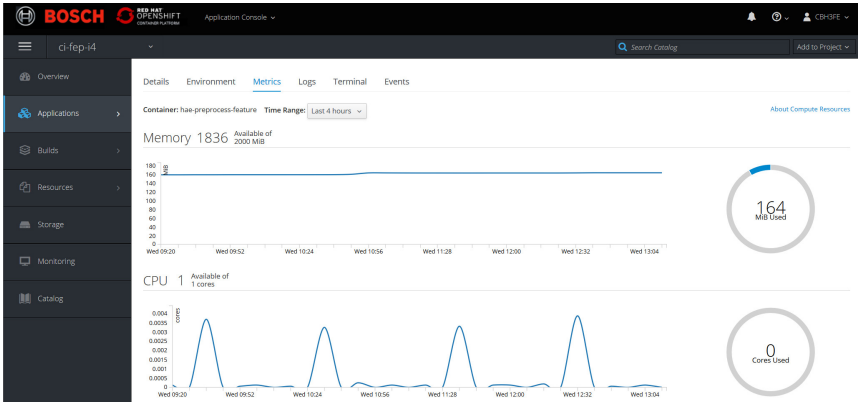


Figure 7.3: Screenshot of OpenShift console used as deployment and operations framework

In contrast to the data mining process, where samples can be handled in batches, daily business automation requires that IDs be handled individually, resulting in rewriting the code to overcome the following challenges. Knowledge about distributions or distinct values that previously was inferred from the data must now be stored beforehand and then retrieved (e.g., detecting outlier). Failures during the processing of a particular ID (e.g., missing entry or new categorical value) must be caught by try-except statements, and their ID still be entered into the database to acknowledge that the processing of this ID failed and to prevent the application from trying to reprocess the same ID again and again. IIR filters need the last N values they calculated. Thus, either these values need to be stored away (e.g., into a states table) and then retrieved during prediction, or an FIR filter approximates it, and only the last measured values need to be retrieved. Both strategies worked fine.

Finally, microservices are ideally written generically enough to handle different types of components that are heat treated similarly. For our specific use cases, only a few parameters (i.e., window size and temperature conditions) are component-specific, such that the feature extraction task for

solenoid valve, camshaft, and roller shoe bainitization can be handled by the same task as for the cylinder head with the given parameter changes. Of course, an individual model must be trained for each label-component combination, but since all data have the same format, the same microservice can be used for all combinations, with the same being true for the predictions.

By setting the system up using this structured modularity, it is easy to retrain models (i.e., by triggering the resp. microservice) and add new components or salt bath lines. For a new component the respective feature extractions parameters have probably to be adjusted (e.g., points in time for quenching, maximal or minmal tempetures, etc.). This problem can also be solved by using ontologies between similar heat treatment processes [134]. As soon as a sufficient amount of data is available in the data base for either one, the respective microservice will pick up the additional component or line and start making predictions for it.

A completely new heat treatment technique (e.g., carbonitriding) would require more implementation effort since feature extraction could be wildly different from the bainitizing procedure. The overall concept would stay the same, but a good amount of analysis would be required before deployment. To ensure that these services and pipelines are functioning properly, the entire framework must be audited, as described below.

Pipeline validation and release are the final milestones to be achieved before operations cash in the promised benefits. Therefore, three requirements must be fulfilled to pass the system qualification test: Successful completion of shadow mode, comprehensive Failure Mode and Effects Analysis (FMEA), and training of and restriction to personnel allowed to interact with the system.

During shadow or test mode, the system is running as if used for daily operations, but its output is still continuously compared to the physical measurements of the label. It may not only reveal modeling mistakes but

any data processing difficulties (i.e., database connection, system speed, empty- or wrong database entries) that may break the system. Malicious database entries can also be made deliberately to test the system's reaction and validate or restrict the manual insertion of data upon entry into the database (e.g., drop-down menus for categorical entries, check for number format and range if a numeric value is expected).

An FMEA's goal is to avoid defects from the outset instead of discovering and correcting them later. For this purpose, possible causes of defects, severity, and probability of their occurrence, as well as the probability of detecting them, should be identified and evaluated already in the development phase, along with respective countermeasures.

Often the most significant source of trouble sits between the keyboard and the back of the chair. Therefore, it is essential to train all persons who interact with the system and have a small circle of administrators who can make changes to any subparts. Adjustments to the system structure should only be made after consultation with all users (i.e., defined responsibilities). Non-compliance causes maintenance costs to skyrocket.

Monitoring and maintenance are essential tasks for continuous daily operation. This is true for the system status itself (i.e., are the services running as scheduled, cf. Figure 7.3) as well as for the output the system generates (i.e., extracted features and predictions for the individual components, cf. Figure 7.4). Such monitoring provides the ability to continuously assess how much discrepancy exists between measured and predicted values in order to respond to trends promptly, Figure 7.4 (a). Color-coded confidence (based on the number of available past data points and proximity to the main cluster) of the prediction can also aid interpretation. Figure 7.4 (b) shows all the models for different components, batch and measurement positions, as well as the R^2 score they achieve on the test set and the number of samples available for training. With the exception of one component (with negative R^2 value, colored red), the models show some predictive power, with the most informative legacy feature[3] being the dwell time in the convection furnace. Although predictive performance and the number of samples are not directly correlated, too small a number of samples impairs predictive performance. Expectedly, the prediction of the surface hardness is much more error-prone and may not be used as a reliable source for true hardness, see Figure 7.4 (c).

[3] The features used are displayed in the column LstFeat and have been calculated by SFS.

(a) Prediction (circles) vs. measurements (lines) for the solenoid valve

(b) Models used for different components along with R^2 score

(c) Scatter plot: prediction vs. measurement for solenoid valve

Figure 7.4: Screenshot of tableau dashboards to monitor system status

8 Summary

Heat treatment of batches is a rather intricate process, the result of which is influenced by many parameters, starting from the material composition of the components and preceding process steps, through the position of the batch, the temperatures, and the gas mixture, to the measuring method. Over the life cycle of a component being produced, these steps are optimized in terms of stability and cost efficiency, starting with the low-hanging fruits such as increased batch size, moving on to shortening process step duration until only small improvements with minor economic benefits can be realized. During this period of diminishing marginal improvements, the datasets continue to grow, paving the way to discover previously undetected relationships through data mining. These methods allow to quantitatively estimate the relative contribution of each influencing factor to the final heat treatment result (with the focus on hardness in this work), thus highlighting the area with the highest potential for improvement and cost reduction effects. Raising profit margins by replacing destructive end-of-line testing with machine learning predictions is the primary focus here.

From two use cases, bainitizing of cylinder heads (100Cr6, 20 000 batches) and case hardening (CH) of nozzle bodies (18CrNi8, 7 000 batches), data were collected and merged, including steel manufacturers' material composition, meta- and sensor data from the processes, hardness measurements of components, and hardness comparison plates. Preparation and analysis

were performed according to the framework for batch heat treatment developed in this work, consisting of data preprocessing (merging and cleaning sources, extracting features, and correcting drifts), label and feature analysis, as well as machine learning.

Label analysis reveals that highly optimized industrialized heat treatment achieves a quite narrow quality window for both processes (bainitizing : ± 35 HV, SD = 8.6 HV, CH: ± 40 HV, SD = 9-15 HV), with almost no hardness values out of tolerance. A significant portion of variance is caused by measurement noise (RMSE to hardness comparison plate reference: 2.2-3.9 HV for bainitizing (HV 10), 4.4-5.5 HV for CH (HV 1)) which is further amplified by test specimen preparation. These additional effects were quantified by testing multiple specimens from the same batch and using one to predict the mean value of the remaining positions. We can see that the current measurement and processing procedure alone limits the predictive capacity to a maximum score of 0.5 - 0.7 resulting in a predictability benchmark that is batch and measurement position-dependent (bainitizing: $\mathrm{RMSE}_{\mathrm{core}}$ = 5-6 HV, $\mathrm{RMSE}_{\mathrm{Surface}}$ = 6-8 HV, CH: RMSE: 9-10 HV). The offset between different batch positions (up to: 15 HV (bainitizing), 10 HV(CH)) could be largely explained by temperature uniformity studies for bainitization but could hardly be explained for CH. Measurement positions on the same component for bainitization were uncorrelated, while those of equal depth for CH were predictable from each other with $R^2 = 0.3 - 0.42$. To mitigate the imprecise HV 1 hardness test, measurement positions of the same depth were combined to the scores 0.1, 0.4, and 0.7 mm. Lastly, the large drifts over time account for another 10-20 HV sometimes attributable to changes in measurement procedure (like recalibration or diamond changes of hardness testers) and sometimes to the features elaborated on below.

Feature analysis shows that most of the long-term hardness fluctuations can be readily explained by the material composition, where even slight

carbon fluctuations ($\pm$ 0.01 wt.-%) in the raw material lead to significant hardness drifts. Both data-driven machine learning approaches and established physical models (e.g., Maynier) arrive at comparable weights for the individual elements and can predict the direction of hardness change as the material composition varies, but often misestimate the amplitude because alloy composition is not the only factor contributing to fluctuations. That is, reacting to future material changes with, for example, modified quenching pressure is possible but certainly not easy. Changes in process parameter settings (e.g., tempering furnace temperature between batches) also explain some of the hardness jumps over time with correlation coefficients up to $r = .3 - .4$. These correlations must be examined with utmost caution and sound domain expertise to avoid falling for spurious correlations. Measured temperatures and pressures generally have very limited predictive power. The extracted process features from these measurements show minimal variation (e.g., $\pm 2\,^{\circ}\mathrm{C}$ during austenitization for both bainitizing and CH) and are barely correlated with hardness $r < .05$. This is true even after correcting for all other known influences (long-term fluctuations, offsets, etc.). Effectively, the process or feedback controller does a marvelous job where the difference between successive batches on the same furnace or line is much smaller than between different lines. Consequently, depending on the station(s) used for production at the same time, hardness difference caused by route, line, or component may be up to 7 HV (bainitizing) and 20 HV(CH). Alarms either do not provide any predictive information regarding hardness or hint at process deficiencies that are easily visible from process parameters, like a prolonged dwell time of the second bainitization stage over weekends. Employing this holistic influence quantification through data mining (incl. time, material, process, batch position, plant, measuring equipment, etc.), statistically robust, generalizable statements can be made about the source of variance, and recommendations can be derived as to where intervention would be most beneficial in order to reduce the variance and further optimize the process. The prediction of this hardness variance using machine learning methods is summarized below.

Machine learning based on process parameters produces quite different results for the two use cases, except for fluctuations. Fortunately, these can reliably be tracked with a first-order Butterworth filter that uses weights around $b = 0.1$ from newly measured values to update its state. It loses about 0.7 RMSE points when the number of support points used to track trends and predict the next values is repeatedly halved.

For bainitization label drifts must be corrected for each individual line (which also differ in the hardness variance produced), as well as their features, which should be calculated based on domain knowledge about the process. Generally, sequential feature forward selection and genetic algorithms then deliver the most predictive uncorrelated subsets. After comprehensive pipeline optimizations with Bayes search, small, robust ML models that can learn minor nonlinearities are best suited for predictions as they barely overfit and still capture the relevant relationships. Although tree-based ensembles generally perform well, a sensitivity analysis also revealed that they can only learn buckets and may not be adept at extrapolation. This analysis also shows that prediction from minimal variance features does not explain a lot of final hardness variance attributable to the process variation itself ($R^2 = 0.11$). Unsupervised analysis with fuzzy c-means mainly found clusters representing single lines and demonstrates that it is almost impossible to detect hardness outliers based on the heat treatment parameters themselves. This is also reflected in a ROC analysis, which can reach an AUC of up to 0.9 for some thresholds, but also shows that between 20-60 % of the outliers would not be detected. These thresholds do not represent the true tolerance limits, as there were almost no true outliers to begin with.

For case hardening the classical ML approach does not work because the process variations are too small and the measurement noise too large. Instead, predicting the hardness of subsequent batches can be accomplished with a hidden state pipeline that considers material variation by tracking

routes through various stations and the position of the test specimens in the batch and its component family. This forecast is as good as testing two specimens from the same batch and then predicting one from the other. The resulting RMSE and R^2 depend strongly on measurement position. Further, a rolling prediction over time shows that parameters seem not time-invariant and should be relearned each year. This pipeline may explain up to 80 % of the variance, with an individual breakdown showing that the largest scatter is caused by measurement error, followed by general fluctuations mainly due to alloy changes. Interestingly, the station seems to have almost no influence here.

Deployment in daily operation was done on an OpenShift platform and works for bainitization of various component categories. A cost reduction strategy (e.g., halving the number of test specimens) may then be implemented based on the known factors contributing to the variance by monitoring all features and label drifts, making predictions from the analyzed features, excluding those batches with properties that are too far from the main cluster, and hedging against bad predictions by adding an additional safety band around the prediction that must not exceed the limits for the measurements.

To summarize, the proposed framework shows how to collect and preprocess data, derive a benchmark for maximum achievable predictability, break down the variance into its contributing factors, and apply state-of-the-art machine learning methods to predict the hardness of heat treated batches. Furthermore, it is shown how to launch a use case in daily operation and transfer the findings to another component. Although many relations could be discovered by data mining and explained by material science, the data-driven models are unusable for extrapolation and the development of new heat treatment processes. Hybrid models, which have inherited their internal structure from physical models and use machine learning methods only for complicated relationships, have the potential to

further increase process understanding and open up a whole new avenue for process development.

A Appendix

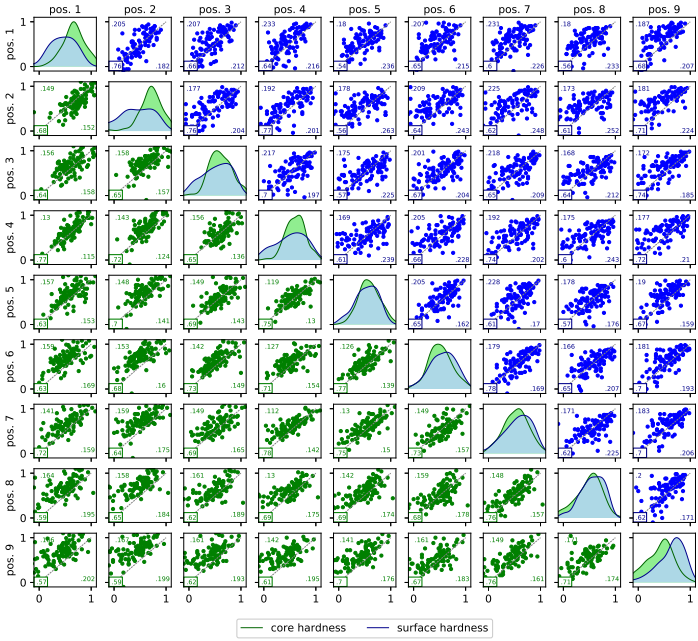


Figure A.1: Bainite: Scatterplot between batch positions regarding core and surface hardness. Upper left corner indicates the mean square error e for the normalized data, while the lower right corner contains the correlation coefficient r. Each point depicts the hardness measurement of two test specimens' positions from the same batch with the hardness of one position on the x- and the other position on the y-axis. The deviation from identity (i.e., same hardness), indicated by the grey diagonal line, is due to several reasons including, position bias, measurement errors, different hardness and hardenability of the blanks and process noise (e.g. convection in the furnace and salt bath). The lower left corner of each subplot contains the correlation coefficient

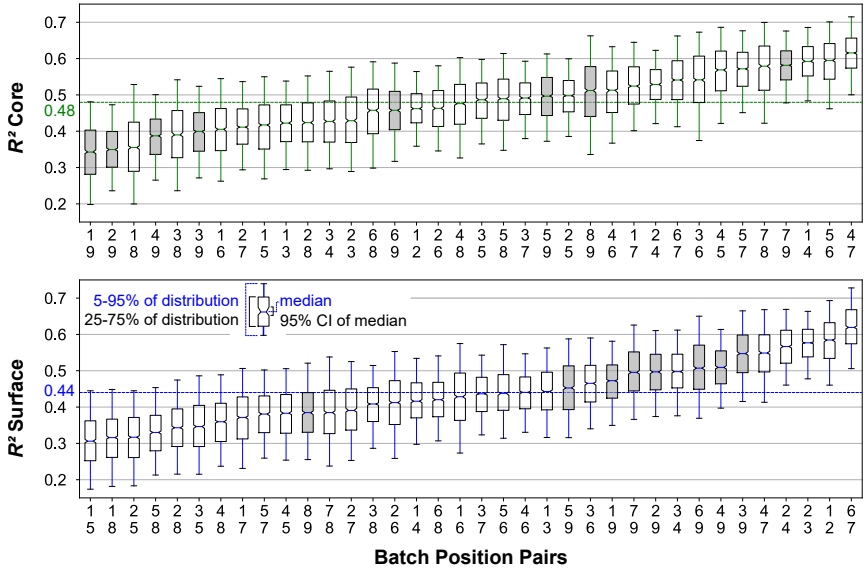
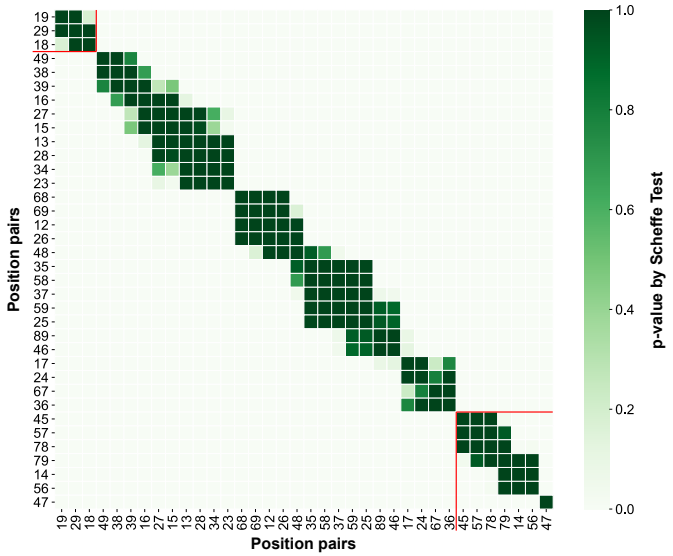
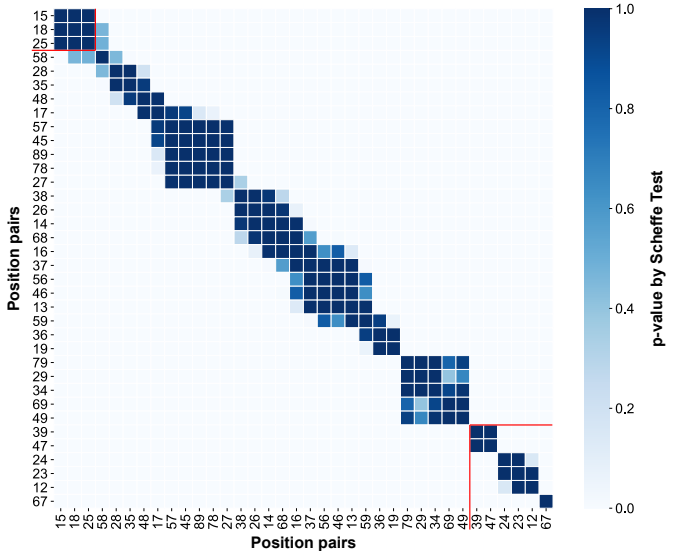


Figure A.2: Bainite: Estimated R^2 score distribution by 4000-fold bootstrapping for each position pair. Whiskers indicate the 5^{th} and 95^{th} percentiles of the distribution, boxes the 2^{nd} and 3^{rd} quartile, notches the 95 % confidence interval of the median



(a) Core hardness



(b) Surface hardness

Figure A.3: Bainite: Scheffetest for pairwise comparison of all position pairs showing their significant difference

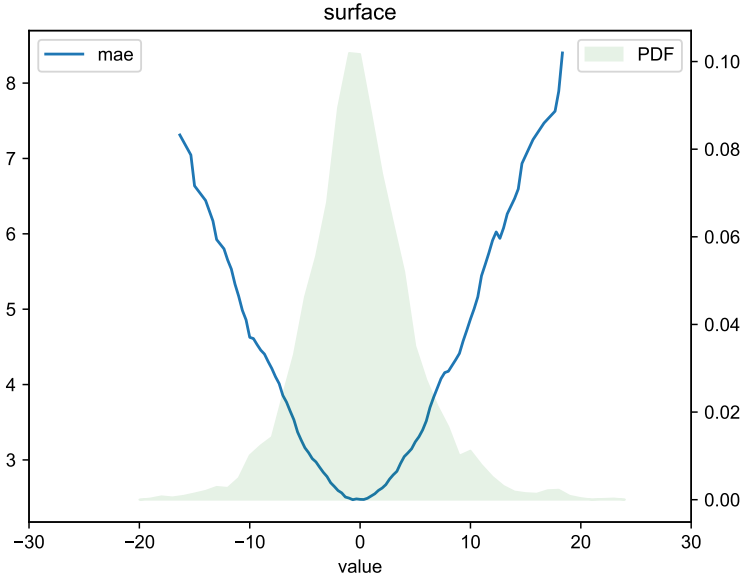


Figure A.4: Bainite: Histogram of mean absolute error in HV between the three imprints and their mean. The measurement of the surface hardness shows a standard deviation of $\sigma_s = 5$ HV, while the spread in core is smaller with $\sigma_c = 3$ HV

Figure A.5: CH: Picture of IPSEN vacuum furnace used for case hardening nozzle bodies. Black rods are used for heating and white nozzles inject the acetylene

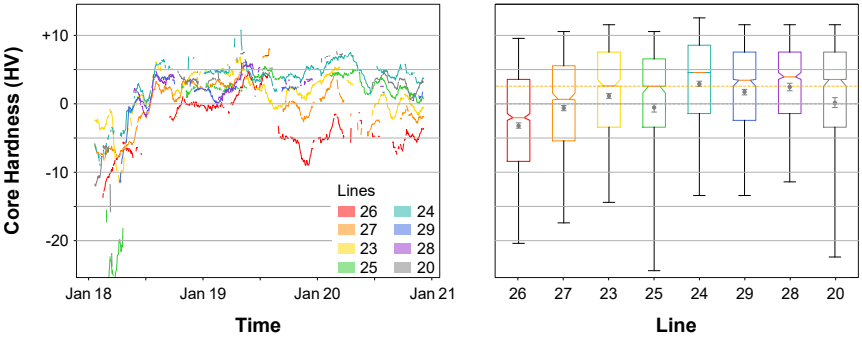


Figure A.6: Left: Mean surface hardness over time per furnace smoothed by a centered rolling window of 31 days (±15 d). Lines are only shown for production (A rolling window would also provide values for a day with no production using the days before and or behind the current day. In this case values are set to NaN). Right: boxplot of surface hardness with median and its 99.9% CI

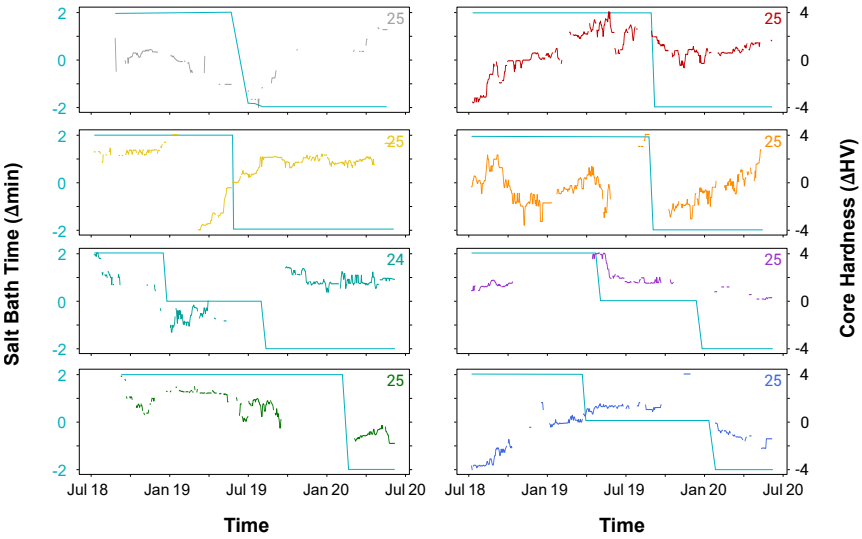


Figure A.7: Bainite: Rolling window over dwell time and core hardness per line for salt bath. No effect can be found for reduced salt bath dwell time

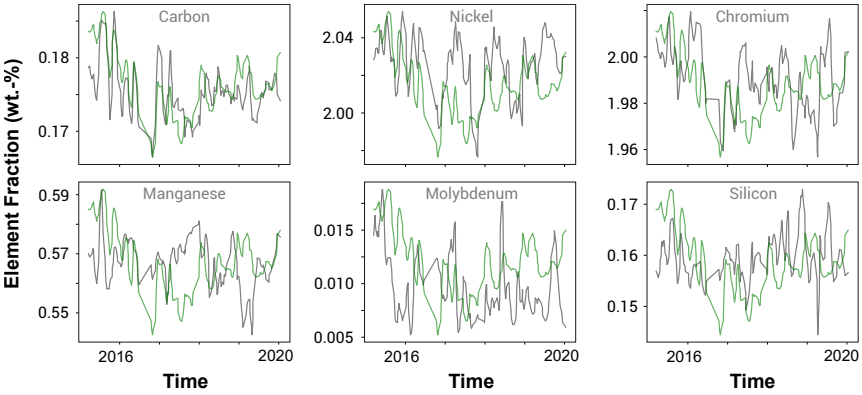


Figure A.8: CH: Weight fraction of elements in 18CrNi8 along with core hardness (in green) over time smoothed with a rolling window of size 5 with triangular shape

Table A.1: Bainite: Optimization results of Bayes search of different pipelines and feature subset selections

model name	select by	cv5 mean	cv5 median	percentile	n feature					
LR	SFS	0.12	0.120	87	52					
LR	GeneticAlgo	0.118	0.114	37	22					
LR	manual 2	0.116	0.121	100	6					
LR	RandomFor	0.11	0.113	6	10					
LR	Mutual Inform	0.11	0.111	20	36					
LR	F score	0.087	0.084	1	1					
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**alpha**	**beta 1**	**beta 2**	**layer size**	
NN single	SFS	0.127	0.125	20	12	0.005	0.500	1	100	
NN single	GeneticAlgo	0.127	0.133	26	15	0.006	0.878	0.876	86	
NN single	manual 2	0.127	0.133	100	6	0	0.500	0.751	100	
NN single	RandomFor	0.108	0.108	1	1	0	0.753	1	100	
NN single	Mutual Inform	0.102	0.099	1	1	0.062	0.500	0.500	100	
NN single	F score	0.068	0.072	2	3	0.5	0.85	0.799	4	
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**layer 1 size**	**layer 2 size**	**layer 3 size**		
NN 3 layer	SFS	0.13	0.135	19	11	100	2	45		
NN 3 layer	GeneticAlgo	0.121	0.122	23	13	21	100	62		
NN 3 layer	manual 2	0.126	0.132	100	6	100	30	42		
NN 3 layer	RandomFor	0.105	0.107	1	1	100	45	100		
NN 3 layer	Mutual Inform	0.105	0.110	1	1	100	100	100		
NN 3 layer	F score	0.051	0.048	3	5	83	2	72		
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**layer size**	**layer size**	**layer size**		
NN stacked	SFS	0.127	0.133	14	8	100	1	100		
NN stacked	GeneticAlgo	0.132	0.136	28	16	83	27	72		
NN stacked	manual 2	0.126	0.130	100	6	3	67	23		
NN stacked	RandomFor	0.108	0.115	8	14	1	27	45		
NN stacked	Mutual Inform	0.099	0.092	7	12	99	10	100		
NN stacked	F score	0.072	0.103	5	9	30	94	80		
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**learning rate**	**max depth**	**min samp. leaf**	**min samp. split**	**n estim.**
GB	SFS	0.124	0.133	54	32	0.015	4	30	26	254
GB	GeneticAlgo	0.124	0.133	30	18	0.200	2	1	30	78
GB	manual 2	0.122	0.126	100	6	0.065	2	1	30	207
GB	RandomFor	0.121	0.125	100	183	0.042	2	1	2	300
GB	Mutual Inform	0.122	0.127	99	181	0.038	2	30	15	244
GB	F score	0.122	0.128	41	75	0.082	2	16	21	86
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**min impur. decr.**	**max depth**	**min samp. leaf**	**min samp. split**	**n estim.**
RF	SFS	0.124	0.135	40	24	0.029	8	12	2	200
RF	GeneticAlgo	0.123	0.133	54	32	0.043	9	12	16	274
RF	manual 2	0.119	0.125	100	6	0.063	14	1	30	300
RF	RandomFor	0.121	0.129	72	131	0.013	7	17	30	191
RF	Mutual Inform	0.121	0.130	100	183	0.001	11	7	2	300
RF	F score	0.122	0.131	86	157	0.001	8	22	2	300
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**C**	**epsilon**	**tol**		
SVR	SFS	0.116	0.116	19	11	0.068	0.084	0		
SVR	GeneticAlgo	0.114	0.113	17	10	0.019	0.9	0.002		
SVR	manual 2	0.114	0.118	100	6	0.017	0.016	0		
SVR	RandomFor	0.094	0.099	1	1	0.008	0.404	0		
SVR	Mutual Inform	0.094	0.098	1	1	0.003	0.9	0		
SVR	F score	-5.735	-0.954	1	1	0.001	0.088	0.001		
model name	**select by**	**cv5 mean**	**cv5 median**	**percentile**	**n feature**	**leaf size**	**n neighbors**	**p**		
KNN	SFS	0.119	0.117	19	11	35	92	1.561		
KNN	GeneticAlgo	0.119	0.114	29	17	100	100	2.000		
KNN	manual 2	0.116	0.118	100	6	100	100	1.663		
KNN	RandomFor	0.101	0.102	1	1	87	96	1.043		
KNN	Mutual Inform	0.101	0.101	1	1	2	97	1.048		
KNN	F score	0.08	0.081	37	67	62	100	1.000		

Bibliography

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv preprint:1603.04467* (2016).

[2] Adithiyaa, T., Chandramohan, D., and Sathish, T. Optimal prediction of process parameters by GWO-KNN in stirring-squeeze casting of AA2219 reinforced metal matrix composites. *Materials Today: Proceedings 21* (2020), 1000–1007.

[3] Altman, N., and Krzywinski, M. Association, correlation and causation. *Nature methods 12*, 10 (2015), 899–900.

[4] Artymiak, P., Bukowski, L., Feliks, J., Narberhaus, S., and Zenner, H. Determination of S-N curves with the application of artificial neural networks. *Fatigue & Fracture of Engineering Materials & Structures 22*, 8 (1999), 723–728.

[5] Bailer-Jones, C., Bhadeshia, H., and MacKay, D. Gaussian process modelling of austenite formation in steel. *Materials Science and Technology 15*, 3 (1999), 287–294.

[6] Baldissera, P., and Delprete, C. Deep cryogenic treatment: a bibliographic review. *The Open Mechanical Engineering Journal 2*, 1 (2008), 1–11.

[7] Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., and Gamboa, H.

TSFEL: Time Series Feature Extraction Library. *SoftwareX 11* (2020), 100456.

[8] Bezdek, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced Applications in Pattern Recognition. Springer US, Boston, MA, 1981.

[9] Bhadeshia, H. K. D. H., Dimitriu, R. C., Forsik, S., Pak, J. H., and Ryu, J. H. Performance of neural networks in materials science. *Materials Science and Technology 25*, 4 (2009), 504–510.

[10] Bishop, C. M. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

[11] Böhland, M., Doneit, W., Gröll, L., Mikut, R., and Reischl, M. Automated design process for hybrid regression modeling with a one-class SVM. *at - Automatisierungstechnik 67*, 10 (2019), 843–852.

[12] BOSCH. Corporate Media Gallery, 2018.

[13] BOSCH. Wärmebehandlungsseminar, 2018.

[14] Breiman, L. Random Forests. *Machine Learning 45*, 1 (2001), 5–32.

[15] Calzolari, M. sklearn-genetic, 2020.

[16] Canale, L. C., Yao, X., Gu, J., and Totten, G. E. A historical overview of steel tempering parameters. *International Journal of Microstructure and Materials Properties 3*, 4/5 (2008), 474.

[17] Cavaleri, L., Asteris, P. G., Psyllaki, P. P., Douvika, M. G., Skentou, A. D., and Vaxevanidis, N. M. Prediction of Surface Treatment Effects on the Tribological Performance of Tool Steels Using Artificial Neural Networks. *Applied Sciences 9*, 14 (2019), 2788.

[18] Chae, J.-Y. *Application of Lower Bainite Microstructure for Bearing Steels and Acceleration of Transformation Kinetics*. PhD thesis, Pohang University of Science and Technology, 2012.

[19] Chandrashekar, G., and Sahin, F. A survey on feature selection methods. *Computers & Electrical Engineering 40*, 1 (2014), 16–28.

[20] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.

[21] Chollet, F., et al. Keras, 2015.

[22] Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A. W. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing 307* (2018), 72–77.

[23] Collins-Thompson, K. Applied Machine Learning in Python: University of Michigan: Module 3: Evaluation, 15.05.2019.

[24] Cumming, G. *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Multivariate applications series. Routledge, New York, 2012.

[25] Cumming, G., and Finch, S. Inference by eye: confidence intervals and how to read pictures of data. *The American psychologist 60*, 2 (2005), 170–180.

[26] Dabrock, E. *Experimentelle und modellbasierte Untersuchung der Wirkzusammenhänge zwischen Prozessparametern und mikrostrukturellen sowie mechanischen Eigenschaften am Beispiel des trockenen Bainitisierens von 100Cr6*. Dissertation, Technischen Universität Kaiserslautern, Kaiserslautern, 2019.

[27] D'Agostino, L., de Santis, A., Di Cocco, V., Iacoviello, D., and Iacoviello, F. Fatigue crack propagation in Ductile Cast Irons:

an Artificial Neural Networks based model. *Procedia Structural Integrity 3* (2017), 291–298.

[28] DeCost, B. L., Francis, T., and Holm, E. A. Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures. *Acta Materialia 133* (2017), 30–40.

[29] DeCost, B. L., Lei, B., Francis, T., and Holm, E. A. High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel. *Microscopy and Microanalysis 25*, 1 (2019), 21–29.

[30] Deutscher Verein des Gas- und Wasserfaches. G 260 (A): 2013-03: Gasbeschaffenheit, March 2013.

[31] Dias, M. L. D. Fuzzy-c-means: An implementation of Fuzzy C-means clustering algorithm, v1.6.0, url: https://git.io/fuzzy-c-means, [Software], 2019.

[32] Dimitriu, R. C., and Bhadeshia, H. Fatigue crack growth rate model for metallic alloys. *Materials & Design 31*, 4 (2010), 2134–2139.

[33] Domingos, P. A few useful things to know about machine learning. *Communications of the ACM 55*, 10 (2012), 78–87.

[34] Dong, J., Vetters, H., Hoffmann, F., Bomas, H., Hirsch, T., Kohlmann, R., and Zoch, H.-W. Gefüge und mechanische Eigenschaften von Wälzlagerstählen nach verkürzten Wärmebehandlungen in der unteren Bainitstufe∗. *HTM Härtereitechnische Mitteilungen 61*, 3 (2006), 128–135.

[35] Draper, N. R., and Smith, H. "Dummy" Variables. In *Applied Regression Analysis*, N. R. Draper and H. Smith, Eds., Wiley Series in Probability and Statistics. Wiley, 1998, pp. 299–325.

[36] Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. Support vector regression machines. In *Advances in Neural Information Processing Systems* (1997), pp. 155–161.

[37] Drucker, H., and Corinna Cortes. Boosting Decision Trees. *Advances in neural information processing systems* (1996), 479–485.

[38] Durodola, J. F., Li, N., Ramachandra, S., and Thite, A. N. A pattern recognition artificial neural network method for random fatigue loading life prediction. *International Journal of Fatigue 99* (2017), 55–67.

[39] Eckstein, H.-J. *Technologie der Wärmebehandlung von Stahl: mit 86 Tabellen*. Dt. Verlag für Grundstoffindustrie, 1987.

[40] Edenhofer, B. An overview of advances in atmosphere and vacuum heat treatment. *Heat Treatment of Metals(UK) 26*, 1 (1999), 1–5.

[41] Edenhofer, B., Joritz, D., Rink, M., and Voges, K. 13 - Carburizing of steels. In *Thermochemical Surface Engineering of Steels*, E. J. Mittemeijer and M. A. J. Somers, Eds., Woodhead Publishing series in metals and surface engineering. Woodhead Publishing, Cambridge, 2015, pp. 485–553.

[42] Efron, B. Better Bootstrap Confidence Intervals. *Journal of the American Statistical Association 82*, 397 (1987), 171.

[43] Ermert, V., Horsch, A., Klein, D., Kohlmann, R., Mahlig, R., and Rentrop, B. Qualitätssicherung in der Wärmebehandlung*. *HTM Journal of Heat Treatment and Materials 63*, 4 (2008), 195–200.

[44] Faizabadi, M. J., Khalaj, G., Pouraliakbar, H., and Jandaghi, M. R. Predictions of toughness and hardness by using chemical composition and tensile properties in microalloyed line pipe steels. *Neural Computing and Applications 25*, 7-8 (2014), 1993–1999.

[45] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters 27*, 8 (2006), 861–874.

[46] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. From Data Mining to Knowledge Discovery in Databases. *AI Magazine 17*, 3 (1996).

[47] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research 13* (2012), 2171–2175.

[48] Feng, S., Zhou, H., and Dong, H. Using deep neural network with small dataset to predict material defects. *Materials & Design 162* (2019), 300–310.

[49] Field, A. P. *Discovering statistics using IBM SPSS statistics: And sex and drugs and rock 'n' roll / Andy Field*, 4th ed. ed. SAGE, London, 2013.

[50] Figueira Pujol, J. C., and Andrade Pinto, J. M. A neural network approach to fatigue life prediction. *International Journal of Fatigue 33*, 3 (2011), 313–322.

[51] Forman, G. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research 3*, Mar (2003), 1289–1305.

[52] Frolova, O., Roos, E., Maile, K., and Müller, W. Representation of the heat specific creep rupture behaviour of 9% Cr steels using neural networks. *Transactions on Machine Learning and Data Mining 4*, 1 (2011), 1–16.

[53] Garcia-Mateo, C., Sourmail, T., Caballero, F. G., Capdevila, C., and García de Andrés, C. New approach for the bainite start temperature calculation in steels. *Materials Science and Technology 21*, 8 (2005), 934–940.

[54] Gers, F. A. Learning to forget: continual prediction with LSTM. In *9th International Conference on Artificial Neural Networks: ICANN '99* (1999), IEE, pp. 850–855.

[55] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (2003), 1157–1182.

[56] Hagymási, L. *Modellierung der Stoffübertragung beim Niederdruckcarbonitrieren mit Ammoniak und Acetylen.* Dissertation, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2016.

[57] Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*, 3 ed. The Morgan Kaufmann series in data management systems. Elsevier, Amsterdam, 2012.

[58] Haque, M. E., and Sudhakar, K. V. ANN based prediction model for fatigue crack growth in DP steel. *Fracture of Engineering Materials and Structures 24*, 1 (2001), 63–68.

[59] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature 585*, 7825 (2020), 357–362.

[60] Haukoos, J. S., and Lewis, R. J. Advanced statistics: bootstrapping confidence intervals for statistics with "difficult" distributions. *Academic emergency medicine : official journal of the Society for Academic Emergency Medicine 12*, 4 (2005), 360–365.

[61] Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., and Shcherbatyi, I. Scikit-optimize, 2020.

[62] Heaton, J. An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016* (3/30/2016 - 4/3/2016), IEEE, pp. 1–6.

[63] Hermann, M., Pentek, T., and Otto, B. Design Principles for Industrie 4.0 Scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)* (2016), IEEE, pp. 3928–3937.

[64] Hollomon, J. H., and Jaffe, L. D. Time-temperature relation in tempering steel. *Transactions of the American Institute of Mining and Metallurgical Engineers 162* (1945), 223–249.

[65] Horsch, A. Reproduzierbarkeit der Härtetiefenbestimmung CHD - NHD -SHD. In *Härtereikrei Mittlerer Neckar*. https://arnold-horsch.de/files/vortrag2017_reproduzierbarkeit.pdf, 2017.

[66] Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering 9*, 3 (2007), 90–95.

[67] Inkscape-Development-Core-Team. Inkscape Project, v1.1, url:https://inkscape.org, [Software], 2007.

[68] Jin, H., Wu, S., and Peng, Y. Prediction of Contact Fatigue Life of Alloy Cast Steel Rolls Using Back-Propagation Neural Network. *Journal of Materials Engineering and Performance 22*, 12 (2013), 3631–3638.

[69] John, G. H., Kohavi, R., and Pfleger, K. Irrelevant Features and the Subset Selection Problem. In *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 121–129.

[70] Jolliffe, I. T. Principal Components in Regression Analysis. In *Principal Component Analysis*, I. T. Jolliffe, Ed., Springer Series in Statistics. Springer New York, New York, NY, 1986, pp. 129–155.

[71] Juszczak, P., Tax, D., and Duin, R. P. W. Feature scaling in support vector data description. In *Proc. asci* (2002), pp. 95–102.

[72] Kang, J., Choi, B., Kim, J., Kim, K., and Lee, H. Neural network application in fatigue damage analysis under multiaxial random loadings. *International Journal of Fatigue 28*, 2 (2006), 132–140.

[73] Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., and Melançon, G. Visual Analytics: Definition, Process, and Challenges. In *Information Visualization*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds., vol. 4950 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 154–175.

[74] Kimball, A. W. Errors of the Third Kind in Statistical Consulting. *Journal of the American Statistical Association 52*, 278 (1957), 133–142.

[75] Kira, K., and Rendell, L. A. A Practical Approach to Feature Selection. In *Machine learning*, D. E. Sleeman and P. E. Edwards, Eds. Morgan Kaufmann, San Mateo, CA, 1992, pp. 249–256.

[76] Kohavi, R., and John, G. H. Wrappers for feature subset selection. *Artificial intelligence 97*, 1-2 (1997), 273–324.

[77] Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal 37*, 2 (1991), 233–243.

[78] Kraskov, A., Stögbauer, H., and Grassberger, P. Estimating mutual information. *Physical Review E 69*, 6 Pt 2 (2004), 066138.

[79] Krauss, G. Martensite in steel: strength and structure. *Materials Science and Engineering: A 273-275* (1999), 40–57.

[80] Kuhn, M., and Johnson, K. *Applied Predictive Modeling*. Springer, New York, 2013.

[81] Kula, P., Dybowski, K., Wolowiec, E., and Pietrasik, R. "Boost-diffusion" vacuum carburising – Process optimisation. *Vacuum 99* (2014), 175–179.

[82] Lambiase, F., Di Ilio, A. M., and Paoletti, A. Prediction of Laser Hardening by Means of Neural Network. *Procedia CIRP 12* (2013), 181–186.

[83] Le, T. T., Fu, W., and Moore, J. H. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics (Oxford, England) 36*, 1 (2020), 250–256.

[84] Leardi, R. Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection. *Journal of Chemometrics 8*, 1 (1994), 65–79.

[85] LeCun, Y., Bengio, Y., et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks 3361*, 10 (1995), 1995.

[86] Li, B., Wu, F., Lim, S.-N., Belongie, S., and Weinberger, K. Q. On Feature Normalization and Data Augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 12383–12392.

[87] Liedtke, D. Über den Zusammenhang zwischen dem Kohlenstoffgehalt in Stählen und der Härte des Martensits. *Materialwissenschaft und Werkstofftechnik 34*, 1 (2003), 86–92.

[88] Liedtke, D., Hoferer, M., Illgner, K. H., Pirzl, N., and Stiele, H. *Wärmebehandlung von Eisenwerkstoffen*, 10 ed., vol. 349 of *Kontakt & Studium*. expert verlag, Renningen, 2017.

[89] Lippmann, N. Patent: Verfahren zum Einsatzhärten von Bauteilen aus Warmarbeitsstählen mittels Unterdruckaufkohlung, 2004.

[90] Liscic, B., Tensi, H. M., Canale, L. C., and Totten, G. E. *Quenching Theory and Technology*. CRC Press, 2010.

[91] Maimon, O., and Rokach, L. *Data Mining and Knowledge Discovery Handbook*. Springer US, Boston, MA, 2010.

[92] Makinen, S., Skogstrom, H., Laaksonen, E., and Mikkonen, T. Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? In *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)* (2021), IEEE, pp. 109–112.

[93] Mikut, R., Reischl, M., Burmeister, O., and Loose, T. Data mining in medical time series. *Biomedizinische Technik. Biomedical engineering 51*, 5-6 (2006), 288–293.

[94] Mitchell, T. M. *Machine Learning.* McGraw-Hill series in computer science. McGraw-Hill, New York and London, 1997.

[95] Molinaro, A. M., Simon, R., and Pfeiffer, R. M. Prediction error estimation: a comparison of resampling methods. *Bioinformatics 21*, 15 (2005), 3301–3307.

[96] Müller, A. C., and Guido, S. *Introduction to machine learning with Python: A guide for data scientists.* O'Reilly, Beijing, 2017.

[97] Nadaraya, E. A. On Estimating Regression. *Theory of Probability & Its Applications 9*, 1 (1964), 141–142.

[98] Nguyen, T.-T., Yang, Y.-S., Kim, K.-S., and Hyun, C.-M. Prediction of heating-line paths in induction heating process using the artificial neural network. *International Journal of Precision Engineering and Manufacturing 12*, 1 (2011), 105–113.

[99] Norm. DIN EN 10083-3: Steels for quenching and tempering, Beuth, Berlin, January 2007.

[100] Norm. DIN EN ISO 4885: Ferrous materials – Heat treatments – Vocabulary, Beuth, Berlin, July 2018.

[101] Norm. DIN EN ISO 6507-3: Metallic materials - Vickers hardness test, Beuth, Berlin, July 2018.

[102] Norm. DIN EN ISO 6507: Metallic materials - Vickers hardness test, Beuth, Berlin, July 2018.

[103] Norm. EN 10052: Vocabulary of heat treatment terms for ferrous products, Beuth, Berlin, October 1993.

[104] Nwadiugwu, M. C. Gene-Based Clustering Algorithms: Comparison Between Denclue, Fuzzy-C, and BIRCH. *Bioinformatics and biology insights 14* (2020), 1–6.

[105] Oh, S., and Ki, H. Deep learning model for predicting hardness distribution in laser heat treatment of AISI H13 tool steel. *Applied Thermal Engineering 153* (2019), 583–595.

[106] Olson, Randal S. and Bartley, Nathan and Urbanowicz, Ryan J. and Moore, Jason H., Ed. *Evaluation of a Tree-Based Pipeline Optimization Tool for Automating Data Science* (New York, New York, USA, 2016), Association for Computing Machinery.

[107] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. CRISP-DM 1.0: Step-by-step data mining guide. *SPSS inc 9* (2000).

[108] Pak, J., Jang, J., Bhadeshia, H. K. D. H., and Karlsson, L. Optimization of Neural Network for Charpy Toughness of Steel Welds. *Materials and Manufacturing Processes 24*, 1 (2008), 16–21.

[109] Parzen, E. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics 33*, 3 (1962), 1065–1076.

[110] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[111] Perktold, J., Seabold, S., and Taylor, J. statsmodels, 2019.

[112] Pezzullo, J. The Relationship between Confidence Intervals and Significance Testing, 2020.

[113] PourAsiabi, H., PourAsiabi, H., AmirZadeh, Z., and BabaZadeh, M. Development a multi-layer perceptron artificial neural network model to estimate the Vickers hardness of Mn–Ni–Cu–Mo austempered ductile iron. *Materials & Design 35* (2012), 782–789.

[114] Press, G. Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says. *Forbes* (23.3.2016).

[115] Rahaman, M., Mu, W., Odqvist, J., and Hedström, P. Machine Learning to Predict the Martensite Start Temperature in Steels. *Metallurgical and Materials Transactions A 50*, 5 (2019), 2081–2091.

[116] Reback, J., McKinney, W., Jbrockmendel, van den Bossche, J., Augspurger, T., Cloud, P., Hawkins, S., young, G., Sinhrks, Roeschke, M., Klein, A., Terji Petersen, Tratner, J., She, C., Ayd, W., Naveh, S., Patrick, Garcia, M., Schendel, J., Hayden, A., Saxton, D., Jancauskas, V., Gorelli, M., Shadrach, R., McMaster, A., Battiston, P., Skipper Seabold, Kaiqi Dong, Chris-B1, and H-Vetinari. Pandas 1.2.4, 2021.

[117] Reddy, N. S., Krishnaiah, J., Hong, S.-G., and Lee, J. S. Modeling medium carbon steels by using artificial neural networks. *Materials Science and Engineering: A 508*, 1–2 (2009), 93–105.

[118] Reddy, N. S., Krishnaiah, J., Young, H. B., and Lee, J. S. Design of medium carbon steels by computational intelligence techniques. *Computational Materials Science 101* (2015), 120–126.

[119] Reunanen, J. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research 3* (2003), 1371–1382.

[120] Robnik-Šikonja, M., and Kononenko, I. An adaptation of Relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, vol. 5. pp. 296–304.

[121] Rose, A., and Hougardy, H. *Atlas zur Wärmebehandlung der Stähle*, vol. 2. Stahleisen mbH, Düsseldorf, 1972.

[122] Saeys, Y., Inza, I., and Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics 23*, 19 (2007), 2507–2517.

[123] Seger, C. *An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.* Degree Project, Royal Institute Of Technology, Stockholm, Sweden, 2018.

[124] Sejnowski, T. J., and Hinton, G. E. *Unsupervised learning: Foundations of neural computation.* Computational neuroscience. MIT Press, Cambridge, Mass. and London, 1999.

[125] Semmlow, J. L., and Griffel, B. *Biosignal and medical image processing*, third edition ed. CRC Press Taylor & Francis Group CRC Press is an imprint of the Taylor & Francis Group an Informa business, Boca Raton, 2014.

[126] Shah, M., and Das, S. K. An Artificial Neural Network Model to Predict the Bainite Plate Thickness of Nanostructured Bainitic Steels Using an Efficient Network-Learning Algorithm. *Journal of Materials Engineering and Performance 27*, 11 (2018), 5845–5855.

[127] Sidhu, G., Bhole, S. D., Chen, D. L., and Essadiqi, E. Determination of volume fraction of bainite in low carbon steels using artificial neural networks. *Computational Materials Science 50*, 12 (2011), 3377–3384.

[128] Sidhu, G., Bhole, S. D., Chen, D. L., and Essadiqi, E. Development and experimental validation of a neural network model for prediction and analysis of the strength of bainitic steels. *Materials & Design 41* (2012), 99–107.

[129] Singh, S. B. Neural network analysis of steel plate processing. *Ironmaking and Steelmaking 25*, 5 (1998), 355–365.

[130] Smith, R. L., and Sandly, G. E. An Accurate Method of Determining the Hardness of Metals, with Particular Reference to Those of a High Degree of Hardness. *Proceedings of the Institution of Mechanical Engineers 102*, 1 (1922), 623–641.

[131] Solano-Alvarez, W., Peet, M. J., Pickering, E. J., Jaiswal, J., Bevan, A., and Bhadeshia, H. Synchrotron and neural network analysis of the influence of composition and heat treatment on the rolling contact fatigue of hypereutectoid pearlitic steels. *Materials Science and Engineering: A 707* (2017), 259–269.

[132] Sterjovski, Z., Nolan, D., Carpenter, K. R., Dunne, D. P., and Norrish, J. Artificial neural networks for modelling the mechanical properties of steels in various applications. *Journal of Materials Processing Technology 170*, 3 (2005), 536–544.

[133] Stich, T. J., Spoerre, J. K., and Velasco, T. The application of artificial neural networks to monitoring and control of an induction hardening process. *Journal of Industrial Technology 16*, 1 (2000), 1–11.

[134] Svetashova, Y., Zhou, B., Pychynski, T., Schmidt, S., Sure-Vetter, Y., Mikut, R., and Kharlamov, E. Ontology-Enhanced Machine Learning: A Bosch Use Case of Welding Quality Monitoring. In *The Semantic Web – ISWC 2020*, J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, Eds., vol. 12507 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2020, pp. 531–550.

[135] Taghizadeh, S., Safarian, A., Jalali, S., and Salimiasl, A. Developing a model for hardness prediction in water-quenched and tempered AISI 1045 steel through an artificial neural network. *Materials & Design 51* (2013), 530–535.

[136] Tamburri, D. A. Sustainable MLOps: Trends and Challenges. In *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (01.09.2020 - 04.09.2020), IEEE, pp. 17–23.

[137] Tenner, J. *Optimisation of the heat treatment of steel using neural networks*. PhD thesis, University of Sheffield, 2000.

[138] Thomas, J. J., and Cook, K. A. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, California, 2005.

[139] Thomas, J. J., and Cook, K. A. A visual analytics agenda. *IEEE computer graphics and applications 26*, 1 (2006), 10–13.

[140] Trzaska, J., and Dobrzański, L. A. Application of neural networks for designing the chemical composition of steel with the assumed hardness after cooling from the austenitising temperature. *Journal of Materials Processing Technology 164-165* (2005), 1637–1643.

[141] Tukey, J. W. Comparing Individual Means in the Analysis of Variance. *Biometrics 5*, 2 (1949), 99–114.

[142] Vermeulen, W. G., van der Wolk, P. J., de Weijer, A. P., and van der Zwaag, S. Prediction of jominy hardness profiles of steels using artificial neural networks. *Journal of Materials Engineering and Performance 5*, 1 (1996), 57–63.

[143] Vetters, H., Dong, J., Bomas, H., Hoffmann, F., and Zoch, H.-W. Microstructure and fatigue strength of the roller-bearing steel 100Cr6 (SAE 52100) after two-step bainitisation and combined

bainitic–martensitic heat treatment. *International Journal of Materials Research 97*, 10 (2006), 1432–1440.

[144] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., and van Mulbregt, P. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods 17*, 3 (2020), 261–272.

[145] Waskom, M. Seaborn: statistical data visualization. *Journal of Open Source Software 6*, 60 (2021), 3021.

[146] Weber, M., Turowski, M., Cakmak, H. K., Mikut, R., Kuhnapfel, U., and Hagenmeyer, V. Data-Driven Copy-Paste Imputation for Energy Time Series. *IEEE Transactions on Smart Grid 12*, 6 (2021), 5409–5419.

[147] Weixuan Fu, Olson, R., Nathan, Grishma Jena, PGijsbers, Augspurger, T., Romano, J., PRONOjit Saha, Sahil Shah, Raschka, S., Sohnam, DanKoretsky, Kadarakos, Jaimecclin, Bartdp1, Bradway, G., Ortiz, J., Jorijn Jacko Smit, Jan-Hendrik Menke, Ficek, M., Akshay Varik, Chaves, A., Myatt, J., Ted, Badaracco, A. G., Kastner, C., Crypto Jerônimo, Hristo, Rocklin, M., and Carnevale, R. EpistasisLab/tpot, 2020.

[148] Wilkinson, L., and Friendly, M. The History of the Cluster Heat Map. *The American Statistician 63*, 2 (2009), 179–184.

[149] Xiang, Y., Sun, D., Fan, W., and Gong, X. Generalized simulated annealing algorithm and its application to the Thomson model. *Physics Letters A 233*, 3 (1997), 216–220.

[150] Zhang, Y. *New Advances in Machine Learning*. InTech, 2010.

[151] Zheng, A., and Casari, A. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Oreilly & Associates Inc and divibib GmbH, Wiesbaden, 2018.

[152] Zhu, Z., Sun, G., He, C., and Liu, A. An Intelligent Method for Prediction of Surface-Hardness in Surface-Hardened Steel Rod. In *4th International Conference on Control and Robotics Engineering (ICCRE)* (2019), pp. 122–126.

List of Publications

Journal articles

[1] Lingelbach, Y.; Hagymási, L.; Waldenmaier, T.; Schulze, V. Prediction of Hardness after Industrialized Bainitization of 100Cr6 based on Process Parameters by Application of Machine Learning Methods. *HTM Journal of Heat Treatment and Materials 75*, 4 (2020), 212–224. doi:10.3139/105.110415.

[2] Lingelbach, Y.; Waldenmaier, T.; Hagymási, L.; Mikut, R; Schulze, V. Material Matters: Predicting the Core Hardness Variance in Industrialized Case Hardening of 18CrNi8. *Materials Science and Engineering Technology 53* , 5 (2020), 576–589. doi: 10.1002/mawe.202100249.

Conference contributions

[1] Lingelbach, Y. Data based approaches for intelligent process control in heat treatment. In *Heat Treatment Congress, Köln* (2019)

[2] Lingelbach, Y. Data are AI's best friend: how to apply machine learning to heat treatment. In *European Conference on Heat Treatment* (2020)