



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart

Dissertation (Dr.-Ing.)

Neuronale Netzwerke für transiente Vorgänge im Antriebsstrang

Neural Networks for transient Operation on Powertrains

Von der Fakultät 7 für Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Valerian Cimniak, M.Sc. (cand. Dr.-Ing.)
aus Kirchheim (Teck)

Hauptberichter: Prof. Dr.-Ing. Michael Bargende (i.R.)

Mitberichter: Prof. Dr.-techn. Bernhard Geringer

Mitberichter: Prof. Dr.-Ing. André Casal Kulzer

Tag der mündlichen Prüfung: 06.07.2023

Institut für Fahrzeugtechnik Stuttgart
Juli 2023



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



Vorwort

Diese Arbeit entstand im Zuge meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Fahrzeugtechnik der Universität Stuttgart (IFS) und darauffolgend am Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS).

Mein persönlicher Dank gilt Prof. Dr.-Ing. Michael Bargende und Prof. Dr.-Ing. André C. Kulzer für die wissenschaftliche Betreuung und Ermöglichung meiner Anstellung am Institut. Dies war in Zeiten der Coronapandemie, in denen viele Firmen Kurzarbeit anmeldeten und Promotionsvorhaben ablehnten, alles andere als selbstverständlich.

Ebenso möchte ich Dr. Ing. Michael Grill und Dr. Ing. Dominik Rether für die außerordentlich kompetente und umfassende Betreuung sowie für die gute Zusammenarbeit und das offene Ohr, welches Sie in der Abteilung verankerten, danken. Der fachliche Austausch hat maßgeblich zum Erfolg dieser Arbeit beigetragen.

Weiter gilt Dipl. Ing. Martin Brutsche und Dr. Ing. Markus Wenig von WinGD ein großes Dankeschön für die finanzielle Ermöglichung meiner Anstellung und fachliche Begleitung im Bereich Datenanalyse und Simulation.

Ebenfalls möchte ich mich bei allen Mitarbeitern in AP3 für die gemeinsame Zeit bedanken. Ob im Büro, auf dem Flur oder in der Kaffeeküche; es ließ sich stets hervorragend über die aktuellen Tagesgeschehen philosophieren. Besonders unsere KI-Runde montagnachmittags werde ich noch länger im Gedächtnis behalten.

Zuletzt geht mein Dank an meine Eltern, die mir den Rücken freigehalten haben und eine ungestörte Verwirklichung meiner Promotion mittrugen.



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



Abstract

Future drive systems are becoming increasingly complex. For example, the hybridization of internal combustion engines opens up extensive possibilities for a split in the powertrain, which, in combination with the design of a combustion process adapted to it, contains numerous tuning parameters. In addition, alternative fuels are essential for achieving CO₂ neutrality in the mobility sector, which imposes further challenges as the combustion properties differ compared to conventional gasoline or diesel fuels.

The use of computer simulations for the calculation and representation of the relationships of complex physical systems, such as for the design of modern powertrains, has become increasingly important in recent decades ([4], p.11). In this context, attempts are made at an early stage in the development process to make statements about the behavior of specific components or entire component groups with the aid of suitable simulation environments. This primarily saves costs, since molds and tools often have to be manufactured in single batches for the production of prototypes. Accordingly, new parts are designed, produced and tested for each change in the test setup. If pre-development is therefore supported by computer simulations, adjustments of the engineer can be implemented quickly and cost-efficiently to the simulation model and their effects or potential can be effectively estimated, as well as a necessary test bench operation for further investigations can be justified.

In contrast to CFD- (Continuous Fluid Dynamics) or FEM- (Finite Element Method) simulations, 0- and 1D-simulations have proven to be a suitable approach for the physical calculation of processes that do not have to be spatially resolved at all points ([41], p.3). While calculating so, the overall system is divided into one or more zones and the state variables are averaged over them. This leads to a significant reduction of the necessary computational effort with equally high accuracies.

Nevertheless, even a quasi-dimensional calculation of mechanical or thermodynamic systems usually requires more time than the underlying processes lasts in reality. For example, the simulation of five million operating cycles of a gasoline engine with a H₂ combustion process takes several days (see chapter 4, p.70) and is not yet suitable for real-time modeling. Neural networks, on the other hand, are generally capable of making very fast predictions about the state of a system due to their ability to approximate mathematical equations via vector operations (see chapter 4, p.70).



The basics for this procedure are old. Mathematically, the processes of neuronal activity are already described in 1943 by Walter Pitts and Warren McCulloch in [73], whose further developments are taken up in chapter 2.1. In [3], Alan Turing described 1950 for the first time the properties of a machine, which are preconditions for being called intelligent and derived from this the theses of his Turing test, which is still used today. In 1958, Frank Rosenblatt developed the first supercomputer at IBM, which was able to recognize individual digits on an image with 400 pixels [20]. This weighed several tons and was about the size of a complete office room. It was only around the turn of the millennium, with the rapidly increasing power density of computers by the use of semiconductor elements, that further significant successes in the field of artificial intelligence could be recorded. For example, the Google research team in [18] was the first to defeat a chess computer based on brute-force algorithms, respectively the calculation of all possible moves, using AI. For a long time, the Chinese game “Go” was considered invincible by computers, since its large board allows more moves than could be calculated by current supercomputers. In [17] and [48], the same research team presents AI-based algorithms that easily beat the world's best Go players and master other, even more complex, world-class games. Further, AI advances on image and speech recognition have become ubiquitous for the daily use of smartphones today.

In the simulation of powertrains, this shows that neural networks can now be trained with the help of a manageable computing infrastructure and are capable of achieving comparable or even higher accuracies than the physics-based simulation of real systems at a greatly reduced prediction speed.

Basically, a distinction is made between supervised and unsupervised learning methods when training neural networks. While unsupervised learning algorithms are well suited for e.g. learning control strategies [55] or other tasks, where no defined initial variables are given, the strength of supervised learning methods lies in finding correlations and regularities on discrete distributions. Furthermore, for supervised learning problems a distinction is made between classification and regression problems. Classification means, input variables are divided into different classes based on previous observations and this principle finds a wide range of application in image recognition [32]. On the other hand, regression attempts to find regularities in data sets, thereby finding suitable relationships between given input and unknown output variables. Algorithms of supervised learning in the field of regression are therefore used to modeling the cyclic and transient processes in the powertrain for the chapters 4 and 5 of this thesis. Chapter 6 deals with a regression analysis using unsupervised learning methods.



In contrast to image recognition, where mainly convolutional networks (CNN) are used, other network structures are suitable for regression analysis. If the processes to be mapped are not or only periodically time-dependent, continuous networks (Feed Forward Networks) can be used, since in these structures no information from previous observations is retained. If time information is necessary to describe transient systems, recurrent neural networks are used, as for example in the solution of the Traveling-Salesman Problem [23]. A further development of these network architectures are LSTM cells (Long Short-Term Memory) by the introduction of a long- and short-term memory by Jürgen Schmidhuber and Sepp Hochreiter in 1997 in [61]. There, with the help of changeable gates, it is achieved that time information can be stored over a longer duration. These, together with transformer networks based on the principle of self-learning attention [78], represent state of the art algorithms for modeling transient processes.

The idea of using adaptive algorithms to map physical relationships is an old one, as already pointed out. For example, neural networks can be trained to estimate the parameters of differential equations [51]. As early as 1998, in [56] is already investigated how to replace subtasks of engine management, such as shifting the ignition timing to control the focus of the mass fuel burned position, with feed-forward networks. However, since test bench data is used there, the data basis, and thus the generalization of the networks, is comparatively rather small.

New at the Institute of Automotive Engineering in Stuttgart is the combination of conventional 0D-1D simulation software for powertrain development (cf. chapters 2.3 and 2.4) with machine learning algorithms. On the one hand, this has the advantage of greatly accelerating simulations with little loss of accuracy ([15], p.1). On the other hand, wide DOEs (Design of Experiments) can be carried out with the help of virtual test carriers and thus very large amounts of data can be generated effectively.

In the research area about the use of neural networks for powertrain development, the number of publications is still rather limited ([63], [2], [49], [4]), although the contributions are steadily increasing. For steady-state operating points, it is shown in ([15], p.16) that neural networks achieve promising results in modeling averaged individual combustion variables over a working cycle during the high-pressure part of the compressions. For example, in a data set with over one million operating points, a deviation of the predicted indexed mean pressure using neural networks of less than 1.87% compared to the simulation is obtained there. Further, it was shown in ([16], p. 8) that a reasonable interconnection of subnetworks results in a significant improvement of the prediction quality for complex combustion chamber variables, such as the location of the peak pressure. Based on this, the modeling of cyclic and crank angle resolved combustion chamber variables is investigated in this thesis, as well as the transfer of transient overall systems into neural structures is addressed.



On the detection of irregularities in measured signals of complete mechatronic systems, the company Airbus has attracted greater attention in the community of data scientists with its ISS project, for the monitoring of functionally relevant measured variables, in [8]. However, the fundamentals of anomaly detection using auto-encoders and LSTM cells are older (cf. [52], [53]). There, it is already described how irregularities in different measured signals are detected with the help of a transient capable encoder-decoder architecture and the calculation of an error parameter.

The aim of this work is to contribute on the consideration of neural networks for the acceleration of various simulation software in the field of powertrain development. Furthermore, advantages, challenges and limitations in dealing with so-called black-box models will be presented, where it is difficult to understand how the algorithms arrive at a result in detail. In addition, the topic of anomaly detection using auto-encoders, which are special network architectures for the representation of complex data sets, will be addressed and, based on the previous investigations, a methodology will be presented, which allows the detection of irregularities in real sensor signals using measurement data and LSTM cells. In all approaches special attention is paid to the greatest possible generality of the models.

The first chapter deals with the calculation of crank angle resolved characteristics of an H₂ commercial vehicle engine in the 0D/01 simulation software “Zylindermodul” and a subsequent surrogate modeling of the underlying relationships in the combustion chamber during the high-pressure part of the compression. Necessary considerations for the boundary conditions at intake closes and characteristics of the combustion are manipulated by various distribution functions. Finally, the mapping accuracy of the generated neural networks is investigated in detail.

In the second chapter, transient driving profiles of a large and slow-running 2-stroke engine are calculated using the simulation software “GT-Suite”. In contrast to the crank angle resolved modeling, additional information flows in the time dimension are necessary, in order to be able to map e.g. thermal inertial behavior in the drivetrain system with the aid of neural networks. Therefore, special attention is paid to the methods of data acquisition in order to achieve the greatest possible generalization of the models later, respectively to cover as many occurring driving situations as possible.



In the final chapter, measured sensor signals are used to build a neural network model that is capable of detecting irregularities in the drivetrain of a large and slow-running 2-stroke engine during operation. Different approaches are investigated to process real-world measurement data, which, in contrast to simulation and test bench data, can be subject to errors and unequally distribution. In addition, the model created will be applied to measurement data from previous years for further investigations, and the anomalies found thereby will be evaluated and analyzed.

Basically, it could be shown that neural networks are suitable to accelerate complex simulations of mechanical and thermodynamic systems in the field of powertrain development many times over. With help of the presented methods for data preparation and compilation as well as the appropriate network architecture, the mapping accuracy can reach similar accuracies as the physics-based simulation. Continuous networks (cf. chapter 2.1.6) are suitable for modeling periodic and cyclic processes, where the time axis is mapped and unfolded at the output via the neurons of the last layer.

In the case of the predictions based on the example of various crank angle-resolved parameters in the combustion chamber during the high-pressure part of the compression of a premixed and spark-ignited H₂ combustion process, the model validity at 500-fold prediction acceleration is in the range of 99.81 %, where the deviations compared with the simulation are less than 4 % (cf. section 4.3).

During the stress test of the neural networks from chapter 4, where the boundary conditions at inlet closes are randomly initialized five million times, it has been found out that the mapping of the processes in the combustion chamber at a very short combustion time in combination with high burn rates, due to the high laminar flame speed in the stoichiometric area when using H₂ as fuel gas, requires a finer resolution output layer. Here, a time axis resolved in 1°KW is not sufficient to represent the high gradients between two neurons and additional neurons in the output layer are proposed to model the processes in, for example, 0.1 °KW steps. However, this is accompanied by increased computational costs in the training as well as for the predictions and it should be weighed up to what extending the mapping of such limiting phenomena using the neural networks is necessary and reasonable before re-modeling. Further improvements by revising the dataset are rather excluded, since the pressure curves of burns with extreme burning rates are modeled satisfactorily despite their rare occurrence.



Even when modeling time-dependent processes, where a flow of information about previous events is crucial for predicting current time steps, high mapping accuracies of less than 2% difference compared to simulation could be achieved while strong simulation acceleration (Tabelle 5-6 - Tabelle 5-10). In contrast to the crank angle-resolved modeling, in which each combustion progression is assigned its own model, several characteristic variables of a drive train are predicted simultaneously here.

An interesting aspect of training the time-dependent LSTM cells is, that the learning rate, time-window and -overlap have a significant influence on the prediction accuracy of the transient effects. If they are set too high (or low), sometimes only binary minimum solutions can be obtained for the mapping of the transient processes. Due to the complex data preparation and compilation, in which every possible occurring power range is evenly distributed, the created neural networks are able to learn general system behavior and do not reproduce predefined operating profiles. This can be seen in the evaluation of randomly generated and compiled driving profiles, in which the engine behavior is also reproduced over a longer period of time with little deviation from the simulation.

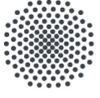
A possible approach for further research is to work out whether the gates of the LSTM cells, which regulate the time-dependent information flow, at some point generate overloads or irregularities in the mapping of transient effects. For this, year-long and contiguous profiles could be predicted and the behavior of the cell state can be studied in more detail. Further, fine tuning, so called transfer-learning, with low learning rate applied to increase the model accuracy should be further investigated. If this can be implemented online during model operation, neural network models can learn aging processes or wear and tear in the powertrain and thus continuously adapt their output to the system state. Likewise, in this work only a few characteristic engine variables have been investigated as an example so far. More complex network architectures with additional input, intermediate and output variables are necessary for the transfer of a complete engine model, including air path, combustion chamber and charge exchange.

The detection of irregularities with the help of special encoder-decoder architectures is successfully applied to several output variables simultaneously in different subsystems of a real powertrain. One challenge lies in dealing with the real measurement data, which contains sensor errors compared to simulations or freely available and already processed data sets. On the other hand, real driving profiles do not necessarily correspond to suitable training data for model building and these must first be compiled using individual and presented methods.



With the introduction of a compression ratio, which represents the relationship between the parameters at the latent vector and the output variables, different models of different subsystems can be compared with each other depending on their complexity. During the evaluation of the individual anomaly detections, in which each subsystem is assigned its own model, it was possible to find and analyze irregularities and, in some cases, real issues of damage. For example, anomalies in the piston movement as well as rotations at the piston ring can be detected with the model of the combustion chamber wall temperatures. Hydraulic systems can also be monitored by comparing the set and actual pressures and the pump control via the control unit of the model. A major advantage here is that higher-dimensional considerations often exceed human comprehension. For three or four measured variables, multidimensional rules can be defined within the measured values may move in relation to each other.

When monitoring more complex subsystems or entire drive trains in which many measured variables interact with each other, the probability of overlooking or not having considered individual events with human perception is high. Here AI-based monitoring can help to detect even small or hard-to-detect irregularities. However, for a production-ready implementation of the created neural networks from chapter 6, further considerations are necessary for the calculation of the anomaly score. For example, alarms cannot be raised for every event, if the anomaly value is above the threshold. Integral considerations are conceivable here to work out the severity of an anomaly. Also, certain evaluations still require clarifications with the engineers responsible for the components in order to further analyze the reconstruction behavior of the models. Although care is taken in the modeling to compile all operating ranges in the training in an evenly distributed manner, it is still conceivable, that anomalies are also output in the training due to data points, that are rarely available or operating states that are not to be monitored. For example, a reverse run of the drive train is not yet considered in the model, and it can be assumed, that the anomaly evaluation or the model must be adapted accordingly for such a use case.



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



Kurzfassung

Zukünftige Antriebssysteme werden immer komplexer. Beispielsweise ergeben sich durch die Hybridisierung von Verbrennungsmotoren umfangreiche Möglichkeiten für die Leistungsaufteilung im Triebstrang, welche in Kombination mit der Auslegung eines darauf angepassten Brennverfahrens zahlreiche Abstimmparameter enthält. Zudem sind alternative Kraftstoffe für das Erreichen einer CO₂-Neutralität im Mobilitätsbereich unerlässlich, was weitere Herausforderungen mit sich bringt, da sich die Verbrennungseigenschaften im Vergleich zu herkömmlichen Benzin- oder Dieselmotoren unterscheiden.

Computermodelle können zum einen dabei helfen die Kosten während der Vorentwicklung zu reduzieren, indem sie notwendige Messkampagnen am Prüfstand eingrenzen, zum anderen lassen sich damit Serienprodukte überwachen und auf korrekte Funktionsweise überprüfen. In dieser Arbeit werden Methoden für die Generierung und Zusammenstellung großer Datenmengen von unterschiedlichen Antriebssystemen und -komponenten untersucht, sowie datenbasierte und echtzeitfähige Modelle, basierend auf künstlichen neuronalen Netzwerken, für die genannten Einsatzzwecke aufgebaut und analysiert.

Das erste Kapitel beschäftigt sich mit der Berechnung von kurbelwinkelaufgelösten Verläufen eines H₂-Nutzfahrzeugmotors in der OD/01-Simulationssoftware „Zylindermodul“, sowie einer darauf folgenden Ersatzmodellierung der zugrundeliegenden Zusammenhänge im Brennraum während dem Hochdruckteil der Verdichtung. Dabei werden notwendige Überlegungen für die Randbedingungen bei „Einlass schließt“ getroffen und Kenngrößen der Verbrennung mit Hilfe von Verteilungsfunktionen beeinflusst. Abschließend wird die Abbildungsgenauigkeit der generierten neuronalen Netzwerke weitreichend untersucht.

Im zweiten Kapitel werden transiente Fahrprofile eines langsam laufenden zwei-Takt Großmotors mit Hilfe der Simulationssoftware „GT-Suite“ berechnet. Anders als bei der kurbelwinkelaufgelösten Modellierung sind dabei zusätzliche Informationsflüsse in der Zeitdimension notwendig, um z.B. thermisches Trägheitsverhalten im Antriebssystem mit Hilfe von neuronalen Netzwerken abbilden zu können. Weiter wird ein besonderes Augenmerk auf die Methoden der Datenerfassung gelegt, um später eine größtmögliche Generalisierung der erstellten Modelle zu erreichen bzw. möglichst viele auftretende Fahrsituationen abzudecken.



Im letzten Kapitel wird mit Hilfe von gemessenen Sensorsignalen aus realem Fahrbetrieb ein neuronales Netzmodell aufgebaut, welches in der Lage ist, Unregelmäßigkeiten im Antriebsstrang eines langsam laufenden zwei-Takt Großmotors zu detektieren. Dabei werden verschiedene Ansätze untersucht, reale Messdaten aufzubereiten, welche, im Gegensatz zu Simulations- und Prüfstandsdaten, mit Fehlern behaftet und einseitig verteilt sein können. Außerdem werden die erstellten Modelle für weiterführende Untersuchungen auf Messdaten der vergangenen Jahre angewendet und die dabei gefundenen Anomalien ausgewertet und analysiert.

Grundsätzlich konnte gezeigt werden, dass die Modellierung mechanischer und thermodynamischer Systeme mit Hilfe von neuronalen Netzwerken erstaunliche Genauigkeiten erzielen kann, wenn dafür geeignete Methoden bei der Datenvorbereitung angewendet werden. Hervorzuheben ist dabei die geringe Vorhersagedauer, deren Größenordnung um mehrere Zehnerpotenzen geringer als die der entsprechenden physikbasierten Berechnung ist.

Bei den kurbelwinkelaufgelösten Modellierungen liegt der Gültigkeitsbereich, in dem es zu keiner merklicher Abweichung zwischen Berechnung und Vorhersage der neuronalen Netzwerke kommt, bei 99.81 %. Eine potentiell mögliche Simulationsbeschleunigung liegt dabei um das 500-fache. Die Abbildung von kurzen Brenndauern in Kombination mit hohen Brennraten, welche bei der Verwendung von H_2 als Kraftstoff, bedingt durch die hohe laminare Flammgeschwindigkeit im Stöchiometrischen auftreten kann, stellt dabei eine Herausforderung dar. Durch eine Erhöhung der Modellkomplexität im Zusammenspiel mit steigendem Rechenaufwand lassen sich diese starken Gradienten dennoch abbilden.

Die Modellierung der transienten Motorgrößen mit Hilfe von neuronalen Netzwerken liefert aufgrund der gleichmäßigen Datenverteilung in der Simulation zufriedenstellende Ergebnisse von unter zwei Prozent Abweichung für alle ausgewerteten Simulationen bei ebenfalls starker Rechenbeschleunigung. Exemplarisch werden hier drei charakteristische Motorgrößen für die Methodenentwicklung betrachtet, welche teilweise stark transientes Verhalten aufweisen. Die Überführung eines vollständigen Motorsimulationsmodells, aufgebaut mit Luftpfad, Brennraum und Ladungswechsel, in ein neuronales Netzwerk erfordert allerdings zusätzliche Ein- und Ausgangsgrößen sowie eine komplexere Modellarchitektur, kann aber prinzipiell nach derselben Vorgehensweise erfolgen.



Die umfangreichste Netzarchitektur dieser Arbeit, welche für die Fehlererkennung bei transienten Messdaten angewendet wird, ist durch eine spezielle Encoder-Decoder Struktur sowie eine leistungsabhängige Verteilung und Gewichtung der Sensorsignale in der Lage, Systemzustände zu erkennen, welche nicht der Norm entsprechen. Für alle untersuchten Subsysteme konnten dabei Auffälligkeiten gefunden und analysiert werden, die im Betrieb zu Problemen führen können. Ebenfalls wird bei regulärem Betrieb keine Anomalie von den Modellen ausgegeben. Dabei besteht der Vorteil gegenüber Grenzwertschranken oder menschlicher Überwachung grundsätzlich darin, dass die Signale vom Modell in ihrer Wechselwirkung zueinander betrachtet werden, bei der die menschliche Auffassung aufgrund der hohen Komplexität im Mehrdimensionalen oftmals an ihre Grenzen stößt.



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



Abkürzungen

AGR	Abgasrückführung
ADAM	Adaptive Momentum Estimation
allg.	Allgemein
A/D	Analog / Digital
bzw.	Beziehungsweise
ca.	Circa
engl.	Englisch
ff.	Für folgende
ggf.	Gegebenenfalls
i.A.	In Anlehnung an
i.d.R.	In der Regel
KNN	K-Nearest Neighbor
KI	Künstliche Intelligenz
KW	Kurbelwinkel
LSTM	Long Short-Term Memory
emax	Maximaler Fehler
MAE	Mean Absolute Error
MSE	Mean Squared Error
NFZ	Nutzfahrzeug
ZOT	Oberer Totpunkt der Zündung
Pred	Predecessor
pmi	Pressure Mean Index



ReLu	Rectified Linear Unit
SeLu	Scaled Exponential Linear Unit
Succ	Successor
SMOTE	Synthetic Minority Oversampling Technique
TanH	Tangens Hyperbolicus
vgl.	Vergleich
WinGD	Winterthur Gas & Diesel
z.B.	Zum Beispiel



Formelzeichen und Variablen

MAE	\emptyset -Absoluter Fehler	(2.1.4)
MSE	\emptyset -Quadratischer Fehler	(2.1.5)
T	Abtastintervall	(2.2.4), (2.2.5), (2.2.6), (2.2.5), (2.2.6)
σ	Aktivierungsfunktion	(2.1.2), (2.1.3), (2.1.26), (2.1.28)
A	Anomaliegrad	(6.1)
v	Ausgabeneuron	(2.1.12), (2.1.13), (2.1.14)
o	Ausgang	(2.1.12), (2.1.13), (2.1.14), (2.1.26), (2.1.28)
y	Ausgang	(2.1.2), (2.1.3), (2.1.4), (2.1.5), (2.1.6), (2.1.7), (2.1.8), (2.1.9), (2.1.10), (2.1.11), (2.1.22), (2.1.23), (2.1.24), (2.1.25), (2.1.26), (2.1.27), (2.1.28), (2.2.1), (2.2.2)
B	Bandbreite	(2.2.6)
θ	Bias	(2.1.1), (2.1.3), (2.1.15), (2.1.16), (2.1.27), (2.1.28)
Δ	Differenz	(2.1.6), (2.1.18)
n	Drehzahl	
x	Eingang	(2.1.1), (2.1.2), (2.1.3), (2.1.7), (2.1.8), (2.1.9), (2.1.10), (2.1.11), (2.1.22), (2.1.23), (2.1.24), (2.1.25), (2.1.26), (2.1.27), (2.1.28), (2.2.1), (2.2.2), (2.2.4), (2.2.5), (2.2.6), (2.2.8)
ζ	Eingang	(2.1.12)
e	Eulerzahl	(2.1.11), (2.2.6)
e	Fehler	(2.1.12), (2.1.13), (2.1.18)
F	Fehlerfunktion	(2.1.8)
f	Forget Gate	(2.1.26), (2.1.27)
f	Frequenz	(2.2.6)
f	Funktion	(2.1.23), (2.1.24), (2.1.25), (2.2.1), (2.2.2)
w	Gewichtung	(2.1.1), (2.1.3), (2.1.15), (2.1.16), (2.1.17), (2.1.18), (2.1.22), (2.1.26), (2.1.27), (2.1.28)
∇	Gradient	(2.1.18), (2.1.19)



δ	Gradient	(2.1.13), (2.1.14), (2.1.16), (2.1.17), (2.1.18)
θ	Grenzwert	(6.1)
H	Huber-Fehler	(2.1.6)
i	Input Gate	(2.1.27)
K	Kombinationen	(2.2.10)
$X(j\omega)$	Komplexer Frequenzgang	(2.2.6)
K	Kompressionsrate	(6.2)
π	Kreiszahl	(2.2.6)
L	Leistung	(2.2.9), (5.1)
l	Leistungsschritt	(2.2.9)
l	Lernmuster	(2.1.12), (2.1.13), (2.1.14), (2.1.15), (2.1.16), (2.1.17), (2.1.18)
η, n	Lernrate	(2.1.18), (2.1.21)
μ	Mittelwert	(2.2.3)
m	Momentum (1. Ordnung)	(2.1.19), (2.1.20), (2.1.21)
v	Momentum (2. Ordnung)	(2.1.19), (2.1.20), (2.1.21)
s	Nachfolgeneuron	(2.1.13), (2.1.14), (2.1.15), (2.1.16), (2.1.17), (2.1.18)
$succ$	Nachfolger	(2.1.13), (2.1.14), (2.1.17), (2.1.18)
\ln	Natürlicher Logarithmus	(2.1.11)
out	Netzausgabe	(2.1.12), (2.1.13), (2.1.14), (2.1.16), (2.1.17), (2.1.18)
in	Netzeingabe	(2.1.15), (2.1.18)
net	Netzeingabe	(2.1.13), (2.1.14), (2.1.15), (2.1.16), (2.1.17), (2.1.18)
z	Neuronenaktivität	(2.1.1), (2.1.2), (2.1.22)
U	Neuronenmenge	(2.1.12), (2.1.13), (2.1.14)
∂	Partielle Ableitung	(2.1.9), (2.1.13), (2.1.16), (2.1.17)
P	Polynom	(2.2.2)
T	Schrittweite	(2.2.8)
δ	Schwellenwert	(2.1.6), (2.2.4), (2.2.5)



ε	Stabilisierungskonstante	(2.1.21)
σ	Standardabweichung	(2.2.3)
b	Steigung	(2.1.7), (2.1.8), (2.1.9), (2.1.10), (2.1.11)
γ	Streckung	(2.1.11)
∞	Unendlich	(2.2.4), (2.2.5), (2.2.6)
a	Verschiebung	(2.1.7), (2.1.8), (2.1.9), (2.1.10), (2.1.11)
u	Verstecktes Neuron	(2.1.13), (2.1.14), (2.1.16), (2.1.17), (2.1.18)
$pred$	Vorgänger	(2.1.15), (2.1.16)
p	Vorgängerneuron	(2.1.15), (2.1.16)
t	Zeit	(2.1.23), (2.1.24), (2.1.25), (2.2.4), (2.2.5)
Z	Zeitfenster	(2.2.8)
C	Zellstatus	(2.1.27)
β	Zerfallsrate	(2.1.19), (2.1.20)
z	z-Score	(2.2.3)



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



Inhaltsverzeichnis

Vorwort.....	I
Abstract.....	III
Kurzfassung	XI
Abkürzungen.....	XV
Formelzeichen und Variablen.....	XVII
Inhaltsverzeichnis.....	1
Abbildungsverzeichnis.....	5
Tabellenverzeichnis.....	9
Hauptteil.....	11
1 Einleitung.....	11
2 Grundlagen.....	15
2.1 Maschinelles Lernen	15
2.1.1 Begriffsabgrenzung	15
2.1.2 Zusammenhänge an einem Neuron.....	18
2.1.3 Aktivierungsfunktionen.....	19
2.1.4 Fehlerfunktionen.....	20
2.1.5 Optimierungsverfahren	22
2.1.6 Netzarchitekturen	30
2.2 Datenaufbereitung	38
2.2.1 Interpolation.....	38
2.2.2 Veränderung der Abtastrate.....	41
2.2.3 Erstellen konsistenter Zeitreihen.....	43
2.2.4 Leistungsklassifikation.....	44
2.2.5 Datenaugmentation	45
2.3 Zylindermodul	47
2.4 GT-Suite	48
3 Stand der Technik	49
4 Modellierung zyklischer Vorgänge	51
4.1 Datengenerierung	51



4.1.1	Drehzahl.....	53
4.1.2	Luftmasse.....	54
4.1.3	Lambda	55
4.1.4	AGR-Rate.....	56
4.1.5	Starttemperatur.....	56
4.1.6	Zündzeitpunkt.....	57
4.1.7	Bauteiltemperaturen	59
4.2	Modellierung I	59
4.2.1	Datenaufteilung	60
4.2.2	Netzarchitektur	60
4.2.3	Druckverlauf I-I	61
4.2.4	Druckverlauf I-I.II	64
4.2.5	Druckgradient I-II	67
4.2.6	Brennverlauf I-III	68
4.2.7	Wandwärmeverlauf I-IV	71
4.3	Belastungstest	73
5	Modellierung transienter Vorgänge.....	81
5.1	Datengenerierung	81
5.1.1	Propellerkurve	82
5.1.2	Zulässige Betriebsbereiche	83
5.1.3	Simulierte Leistungsprofile.....	84
5.1.4	Datenextraktion	86
5.2	Modellierung II	87
5.2.1	Datenaufteilung	87
5.2.2	Netzarchitektur	88
5.2.3	Zeitabhängigkeit	89
5.2.4	Modellierungsschema.....	90
5.2.5	Leistungsänderung 25-100%	91
5.2.6	Leistungsänderung 25-100%	92
5.2.7	Leistungsänderung 5-25%.....	93
5.2.8	Waveloading.....	94



5.2.9	Vollständiges Leistungsprofil	95
5.2.10	Transferlearning II-III	96
6	Detektion von Unregelmäßigkeiten	99
6.1	Messaufzeichnungen	99
6.1.1	Datenextraktion	100
6.1.2	Datengewichtung.....	101
6.2	Subsysteme	102
6.2.1	Kolbenbewegung.....	103
6.2.2	Verbrennung.....	103
6.2.3	Einspritzung.....	103
6.2.4	Hydraulik	103
6.3	Modellierung III	104
6.3.1	Datenaufteilung	104
6.3.2	Netzarchitektur	104
6.3.3	Rekonstruktionsfehler	105
6.3.4	Verschlüsselungsgrad	106
6.3.5	Kolbenwandtemperaturen III-I.....	106
6.3.6	Verbrennung III-II.....	114
6.3.7	Einspritzung III-III.....	116
6.3.8	Hydraulik III-IV.....	121
	Diskussion.....	125
	Literaturverzeichnis	129
	Anhang.....	135
7	Parametrierung Zylindermodul.....	135
8	Modellübersicht	143
8.1	Druckverlauf I-I.....	143
8.2	Druckgradientenverlauf I-II	143
8.3	Brennverlauf I-III	143
8.4	Wandwärmeverlauf I-IV.....	144
8.5	Vollmotormodell II-II	144
8.6	Kolbenwandtemperaturen (AD) III-I.....	144



8.7	Verbrennung (AD) III-II.....	145
8.8	Einspritzung (AD) III-III.....	145
8.9	Hydraulik (AD) III-IV	146
9	Zusätzliche Darstellungen.....	147



Abbildungsverzeichnis

Abbildung 2-1: Begriffsabgrenzung von Schlagwörtern des maschinellen Lernens (i.A. [31], S.4)	17
Abbildung 2-2: Aufbau einer Zelle im menschlichen Gehirn ([71], S.10)	17
Abbildung 2-3: Mathematische Betrachtung an einem Neuron (i.A. [66], S.62)	18
Abbildung 2-4: Verschiedene Aktivierungsfunktionen	20
Abbildung 2-5: Huber Fehlerfunktion für $\delta = 1$	22
Abbildung 2-6: Regressionsgerade einer Geschwindigkeitsmessung	23
Abbildung 2-7: Logistische Regressionskurve für Infektionszahlen in einer Pandemie	25
Abbildung 2-8: Kontinuierliches Neuronales Netz mit zwei Zwischenschichten	30
Abbildung 2-9: Rückgekoppeltes Netz zur Berechnung von expliziten Differentialgleichungen (i.A. [71], S. 148)	32
Abbildung 2-10: Schematischer Aufbau einer LSTM Zelle mit verschiedenen Schleusenfunktionen (i.A. [11])	33
Abbildung 2-11: Nebeneinanderschaltung von LSTM-Zellen zur Zeitreihenmodellierung	35
Abbildung 2-12: Schichtung von LSTM-Zellen I (return_state=True, return_sequences=False)	36
Abbildung 2-13: Schichtung von LSTM-Zellen II (return_state=False, return_sequences=True)	37
Abbildung 2-14: Schichtung von LSTM-Zellen III (return_state=True, return_sequences=True)	37
Abbildung 2-15: Temperaturverlauf mit Überlastungen [38]	39
Abbildung 2-16: Korrigierter Temperaturverlauf ohne Überlastungen	40
Abbildung 2-17: z-Score für einen Temperaturverlauf mit Überlastung	41
Abbildung 2-18: Klassifiziertes einstündiges Zeitfenster der Leistungsklasse (20, 60)	45
Abbildung 2-19: Klassifizierte einstündige Zeitfenster der Klasse (80, 40) mit interpolierter Sequenz 4*	46
Abbildung 2-20: Klassifiziertes einstündiges Zeitfenster der Klasse (100, 90) mit überlagerten Sequenzen 2*, 3*, 4*	47
Abbildung 4-1: Verteilung des indizierten Mitteldrucks in der Simulation	54



Abbildung 4-2: Laminare Flammengeschwindigkeit für verschiedene Kraftstoffe nach [25].....	55
Abbildung 4-3: Unterschiede zwischen der Vorgabe der Starttemperatur oder des Startdrucks in der Simulation.....	56
Abbildung 4-4: Normalverteilung des Zündzeitpunktes für $\mu=-10$ und verschiedene σ	58
Abbildung 4-5: Verteilung der Schwerpunktlage bei uniformer und normaler Verteilung des Zündzeitpunkts	58
Abbildung 4-6: Kolbentemperaturen bei Vollast für NFZ-Motoren nach [45]	59
Abbildung 4-7: Netzlayout für die kurbelwinkelaufgelöste Modellierung I.....	61
Abbildung 4-8: Auswertung der Druckverläufe für verschiedene Füllungen und Randbedingungen nach Tabelle 4-6 mit Parametrierung I-I	63
Abbildung 4-9: Auswertung der Druckverläufe für verschiedene Füllungen und Randbedingungen nach Tabelle 4-6 mit Parametrierung I-I.II.....	66
Abbildung 4-10: Auswertung der Druckgradienten aus Abbildung 4-9	68
Abbildung 4-11: Auswertung der Brennverläufe für die Parametervariation nach Tabelle 4-13 mit Modellierung I-III	70
Abbildung 4-12: Auswertung der Wandwärmeverläufe für die Parametervariation nach Tabelle 4-13 mit Modellierung I-IV	72
Abbildung 4-13: Größte maximale Fehler während des Belastungstests für die modellierten Druckverläufe I-I.II.....	75
Abbildung 4-14: Größte maximale Fehler während des Belastungstests für die modellierten Druckgradienten I-II.....	76
Abbildung 4-15: Größte maximale Fehler während des Belastungstests für die modellierten Brennverläufe I-III	77
Abbildung 4-16: Größte maximale Fehler während des Belastungstests für die modellierten Wandwärmeverläufe I-IV.....	78
Abbildung 5-1: Exemplarische Propellerkurve für den 5X72DF im FPP-Betrieb (i.A. [74], S.2-7)	83
Abbildung 5-2: Zeitliche Grenzwerte für den 5X72DF bei einer Änderung der Leistung im FPP-Betrieb nach (i.A. [74],S. 5-9)	84
Abbildung 5-3: Auszug aus den generierten Leistungsprofilen entsprechend Tabelle 5-2.....	85
Abbildung 5-4: Auszug aus den Daten für die Simulation des Waveloadings bei unterschiedlicher Leistung, Amplitude und Frequenz	86



Abbildung 5-5: Auszug der Extraktion von verschiedenen Fahrmanöver aus den simulierten Leistungsprofilen	87
Abbildung 5-6: Netzlayout für die zeitaufgelöste Modellierung II.....	88
Abbildung 5-7: Transientes Verhalten der Abgastemperatur in der Simulation	90
Abbildung 5-8: Modellierungsschema für die zeitaufgelöste Modellierung II	91
Abbildung 5-9: Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-I	92
Abbildung 5-10: Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-II	93
Abbildung 5-11: Modellierung einer transienten Leistungsänderung von 5-25% mit Parametrierung II-II	94
Abbildung 5-12: Modellierung von Waveloading bei 50% Leistung mit Parametrierung II-II.....	95
Abbildung 5-13: Modellierung eines vollständigen Fahrprofils mit Parametrierung II-II	96
Abbildung 5-14: Modellierung eines vollständigen Fahrprofils mit Parametrierung II-III	97
Abbildung 6-1: Reales Fahrprofil eines Schiffes mit 12X92DF Antriebsstrang.....	101
Abbildung 6-2: Gewichtung der Trainingsdaten für die Leistungsklassifikation entsprechend Tabelle 2-1.....	102
Abbildung 6-3: Netzlayout für die Fehlererkennung III	105
Abbildung 6-4: Auswertung der Fehlerkennung III-I am Fahrprofil 1 von Zylinder 8	108
Abbildung 6-5: Auswertung der Fehlerkennung III-I am Fahrprofil 1 von Zylinder 7	109
Abbildung 6-6: Auswertung der Fehlerkennung III-I am Fahrprofil 2 von Zylinder 1	110
Abbildung 6-7: Auswertung der Fehlerkennung III-I am Fahrprofil 2 von Zylinder 12	111
Abbildung 6-8: Detailauswertung der Fehlerkennung III-I am Fahrprofil 2 von Zylinder 12.....	112
Abbildung 6-9: Schematische Ansicht des Kolbenkontakts von Zylinder 12 während dem Fahrprofil 2	112
Abbildung 6-10: Detailauswertung der Fehlerkennung III-I am Fahrprofil 3 von Zylinder 1.....	113
Abbildung 6-11: Detailauswertung der Fehlerkennung III-II am Fahrprofil 4 von Zylinder 7.....	115



Abbildung 6-12: Detailauswertung der Fehlerkennung III-II am Fahrprofil 4 von Zylinder 8.....	116
Abbildung 6-13: Detailauswertung 1 der Fehlerkennung III-III am Fahrprofil 1 (Februar-März).....	117
Abbildung 6-14: Detailauswertung 1 der Fehlerkennung III-III am Fahrprofil 1 (Februar-März).....	119
Abbildung 6-15: Auswertung der Fehlerkennung III-III am Fahrprofil 1	120
Abbildung 6-16: Auswertung der Fehlerkennung III-IV-A am Fahrprofil 4	122
Abbildung 6-17: Auswertung der Fehlerkennung III-IV-B am Fahrprofil 4	123
Abbildung 9-1: Auswertung ohne Anomalien der Fehlerkennung III-III	147
Abbildung 9-2: Auswertung ohne Anomalien der Fehlerkennung III-IV-A	148
Abbildung 9-3: Auswertung ohne Anomalien der Fehlerkennung III-IV-B	148



Tabellenverzeichnis

Tabelle 2-1: Parametrierung des ADAM-Optimierungsverfahrens	29
Tabelle 4-1: Geometrische Randbedingungen des H ₂ -Motormodells	52
Tabelle 4-2: Variablen der Kennfeldraasterung des H ₂ -Motormodells	53
Tabelle 4-3: Aufteilung des Ausgangsdatensatzes für die Modellierung I-I	60
Tabelle 4-4: Netzparameter für die kurbelwinkelaufgelöste Modellierung I-I	61
Tabelle 4-5: Gesamtfehler für die Modellierung des Druckverlaufs I-I	62
Tabelle 4-6: Konstante Parameter für die Auswertung der Zylinderdruckverläufe	63
Tabelle 4-7: MAE und maximaler Fehler bei der Modellierung des Druckverlaufs für verschiedene Füllungen I-I	64
Tabelle 4-8: Änderungen der Trainingsarchitektur für die Modellierung I-I.II	65
Tabelle 4-9: MAE und maximaler Fehler bei der Modellierung des Druckverlaufs für verschiedene Füllungen I-I.II	66
Tabelle 4-10: Gesamtfehler für die Modellierung der Druckgradienten I-II	68
Tabelle 4-11: Gesamtfehler für die Modellierung des Brennverlaufs I-III	68
Tabelle 4-12: Gesamtfehler für die Modellierung des Wandwärmeverlaufs I-III	69
Tabelle 4-13: Parametervariation für die Auswertung der Brenn- und Wandwärmeverläufe	69
Tabelle 4-14: MAE und maximaler Fehler nach Modellierung I-III des Brennverlaufs für verschiedene indizierte Mitteldrücke	71
Tabelle 4-15: MAE und maximaler Fehler nach Modellierung I-IV des Wandwärmeverlaufs für verschiedene indizierte Mitteldrücke	73
Tabelle 4-16: Maximal zulässige Abweichungen und Überschreitungen während dem Belastungstest der neuronalen Netzmodelle	74
Tabelle 5-1: Geometrische Randbedingungen und Kennwerte eines 5X72DF-Antriebsstrangs der Firma WinGD ([74], S.1-1)	82
Tabelle 5-2: Zeitliche Grenzwerte für den 5X72DF bei einer Änderung der Leistung im FPP-Betrieb nach ([74],S. 5-9)	84
Tabelle 5-3: Aufteilung des Ausgangsdatensatzes für die Modellierung II	88
Tabelle 5-4: Netzparameter für die zeitaufgelöste Modellierung II	89
Tabelle 5-5: Durchschnittliche und maximale Fehler bei der Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-I	92



Tabelle 5-6: Durchschnittliche und maximale Fehler bei der Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-II	93
Tabelle 5-7: Durchschnittliche und maximale Fehler bei der Modellierung einer transienten Leistungsänderung von 5-25% mit Parametrierung II-II	94
Tabelle 5-8: Durchschnittliche und maximale Fehler während der Modellierung von Waveloading bei 50% Leistung mit Parametrierung II-II	95
Tabelle 5-9: Durchschnittliche und maximale Fehler bei der Modellierung eines vollständigen Fahrprofils mit Parametrierung II-II	96
Tabelle 5-10: Durchschnittliche und maximale Fehler bei der Modellierung eines vollständigen Fahrprofils mit Parametrierung II-III	97
Tabelle 6-1: Geometrische Randbedingungen und Kennwerte eines 12X92DF-Antriebsstrangs der Firma WinGD ([75], S.1-1)	100
Tabelle 6-2: Ausschnitt der Lastklassifikation aus einem Fahrprofil entsprechend Abbildung 6-1	101
Tabelle 6-3: Aufteilung des Ausgangsdatensatzes für die Modellierung III	104
Tabelle 6-4: Netzparameter für die Fehlererkennung III-I	107
Tabelle 6-5: Gewichtung der Kolbenwandtemperaturen für die Fehlererkennung III-I	107



Hauptteil

1 Einleitung

Die Anwendung von Computersimulationen für die Berechnung und Darstellung der Zusammenhänge komplexer physikalischer Systeme, wie z.B. bei der Auslegung von modernen Antriebsstränge, hat in den vergangenen Jahrzehnten stark an Bedeutung gewonnen ([4], S.11). Dabei wird im Entwicklungsprozess frühzeitig versucht, Aussagen über das Verhalten bestimmter Bauteile oder ganzer Bauteilgruppen mit Hilfe von geeigneten Simulationsmodellen zu treffen. Das spart in erster Linie Kosten, da für die Fertigung von Prototypen häufig Formen und Werkzeuge in Einzelerie hergestellt werden müssen. Für jede Änderung im Versuchsaufbau werden dementsprechend neue Teile konstruiert, produziert und getestet. Wird die Vorentwicklung dabei simulativ unterstützt, lassen sich Anpassungen des Ingenieurs (m/w/d) schnell und kosteneffizient im Simulationsmodell umsetzen und deren Auswirkungen bzw. Potential effektiv abschätzen sowie ein notwendiger Prüfstandsbetrieb für weiterführende Untersuchungen rechtfertigen.

Bei der physikalischen Berechnung von Vorgängen, die nicht an allen Stellen räumlich aufgelöst sein müssen, hat sich im Gegensatz zur CFD- (engl. Continuous Fluid Dynamics) oder FEM- (Finite Elemente Methode) Simulation eine 0- und 1D-Simulation als geeigneter Berechnungsansatz herausgestellt ([41], S.3). Dabei wird das Gesamtsystem in eine oder mehrere Zonen unterteilt und die Zustandsgrößen darüber gemittelt berechnet. Dies führt bei gleicherweise hohen Genauigkeiten zu einer deutlichen Reduktion des notwendigen Rechenaufwands.

Dennoch benötigt auch eine quasidimensionale Berechnung von mechanischen oder thermodynamischen Systemen teilweise mehr Zeit, als die zugrundeliegenden Vorgänge in Realität andauern. Beispielweise benötigt die Simulation von zwei Millionen Arbeitsspielen eines Ottomotors mit H₂-Brennverfahren mehrere Tage (vgl. Kapitel 4, S.74) und eignet sich bislang nicht für echtzeitfähige Modellierungen. Neuronale Netzwerke sind dagegen, aufgrund ihrer Eigenschaft mathematische Gleichungen über Vektoroperation anzunähern, generell in der Lage, sehr schnelle Vorhersagen über die Zustände eines Systems treffen zu können (vgl. Kapitel 4, S.74).

Die Grundlagen dafür sind alt. Mathematisch sind die Vorgänge von neuronaler Aktivität bereits 1943 von Walter Pitts und Waren McCulloch in [73] beschrieben, deren Weiterführungen in Kapitel 2.1 aufgegriffen werden. In [3] beschreibt Alan Turing 1950 erstmals die Eigenschaften einer Maschine, welche Voraussetzungen dafür sind, als intelligent bezeichnet zu werden und leitet daraus die Thesen seines bis heute angewendeten Turing-Tests ab.



1958 entwickelte Frank Rosenblatt bei IBM den ersten Supercomputer, welcher in der Lage war, einzelne Ziffern auf einem Bild mit 400 Pixeln zu erkennen [20]. Dieser wog mehrere Tonnen und hatte in etwa die Größe eines kompletten Büroraumes. Erst mit der stark zunehmenden Leistungsdichte von Computern durch die Verwendung von Halbleiterelementen, konnten um die Jahrtausendwende weitere maßgebliche Erfolge im Bereich der künstlichen Intelligenz (im Folgenden KI) verzeichnet werden. Beispielsweise konnte das Forschungsteam von Google in [18] erstmals einen Schachcomputer, welcher auf Brute-Force Algorithmen bzw. dem Berechnen aller möglichen Spielzüge beruht, mit KI besiegen. Lange Zeit galt dabei das chinesische Spiel „Go“ von Computern als unbezwingbar, da es aufgrund seines großen Spielbretts mehr Spielzüge ermöglicht, als mit aktuellen Supercomputern berechenbar sind. In [17] und [48] stellt dasselbe Forschungsteam KI-basierte Algorithmen vor, welche die weltbesten Go-Spieler mühelos schlagen und weitere, noch komplexere Spiele auf Weltklassenniveau meistern. Ferner sind die Fortschritte durch KI in der Bild- und Spracherkennung bei der Verwendung von Smartphones heutzutage allgegenwärtig geworden.

Für die Simulation bedeutet das, neuronale Netzwerke lassen sich nun zum einen mit Hilfe einer überschaubaren Recheninfrastruktur trainieren, und sind weiter in der Lage, bei stark reduzierter Vorhersagegeschwindigkeit vergleichbare oder sogar höhere Genauigkeiten wie die physikbasierte Simulation realer Systeme zu erreichen.

Grundsätzlich wird beim Training von neuronalen Netzwerken zwischen überwachten und unüberwachten Lernverfahren unterschieden. Während sich unüberwachte Lernalgorithmen z.B. gut für das Erlernen von Regelungsstrategien [55] oder anderen Aufgaben eignen, bei denen keine definierten Ausgangsgrößen vorgegeben werden, liegt die Stärke der überwachten Lernverfahren darin, Zusammenhänge und Gesetzmäßigkeiten in diskreten Verteilungen zu finden. Weiter wird bei Fragestellungen des überwachten Lernens zwischen Problemen der Klassifikation und Regression unterschieden. Klassifikation bedeutet, Eingangsgrößen werden basierend auf vorangegangenen Beobachtungen in unterschiedliche Klassen eingeteilt, und dieses Prinzip findet in der Bilderkennung einen großen Anwendungsbereich [32]. Bei der Regression wird dagegen versucht, Gesetzmäßigkeiten in Datensätzen zu finden, um dadurch passende Beziehungen zwischen gegebenen Eingangs- und unbekanntem Ausgangsgrößen zu finden. Algorithmen des überwachten Lernens im Bereich der Regression werden daher für die Modellierung von zyklischen und transienten Vorgängen im Antriebsstrang bei den Kapiteln 4 und 5 dieser Arbeit verwendet. Kapitel 6 beschäftigt sich mit Hilfe von unüberwachten Lernmethoden mit der Regressionsanalyse.



Im Gegensatz zur Bilderkennung, bei der hauptsächlich faltende Netzwerke (engl. Convolutional Networks - CNN) eingesetzt werden, eignen sich für die Regressionsanalyse andere Netzstrukturen. Sind die dafür abzubildenden Vorgänge nicht oder nur periodisch zeitabhängig, lassen sich kontinuierliche Netze (engl. Feed Forward Networks) verwenden, da in diesen Strukturen keine Informationen von vorangegangenen Beobachtungen behalten bleiben. Sind Zeitinformationen zur Beschreibung von transienten Systemen notwendig, werden rückgekoppelte Netzwerke (eng. Recurrent Neural Networks) verwendet, wie z.B. bei der Lösung des Traveling-Salesman Problems [23]. Eine Weiterentwicklung dieser Netzarchitekturen sind LSTM-Zellen (engl. Long Short-Term Memory) durch die Einführung eines Lang- und Kurzzeitgedächtnisses von Jürgen Schmidhuber und Sepp Hochreiter 1997 in [61]. Dort wird mit Hilfe von veränderbaren Schleusen (engl. Gates) erzielt, dass Zeitinformationen über längere Dauer gespeichert werden können. Diese stellen zusammen mit Transformer-Netzwerken, welche auf dem Prinzip der selbstlernenden Aufmerksamkeit (engl. Self-Attention) [78] beruhen, den Stand der Technik für die Modellierung von transienten Vorgängen dar.

Mit dieser Arbeit soll ein Beitrag dazu geleistet werden, neuronale Netzwerke für die Beschleunigung von diverser Simulationssoftware im Bereich der Antriebsstrangentwicklung in Betracht zu ziehen. Weiter sollen Vorteile, Herausforderungen und Grenzen im Umgang mit sogenannten Black-Box Modellen dargestellt werden, bei denen nur schwer nachvollziehbar ist, wie die Algorithmen im Detail zu einem Ergebnis kommen. Zudem wird das Thema der Anomalieerkennung mit Hilfe von Auto-Encodern, welche spezielle Netzarchitekturen für die Repräsentation komplexer Datenmengen darstellen, aufgegriffen und, basierend auf den vorangegangenen Untersuchungen, eine Methodik vorgestellt, welche es ermöglicht, mit Hilfe von Messdaten und LSTM-Zellen Unregelmäßigkeiten in echten Sensorsignalen aufzufinden. Dabei wird bei allen Ansätzen ein besonderes Augenmerk auf eine größtmögliche Allgemeingültigkeit, bzw. Generalisierung gelegt.



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



2 Grundlagen

2.1 Maschinelles Lernen

Künstliche Intelligenz, maschinelles Lernen und neuronale Netzwerke sind seit einigen Jahren häufig verwendete Schlagwörter im wissenschaftlichen Umfeld. Im folgenden Kapitel werden diese Begriffe kurz erläutert und voneinander abgegrenzt. Zusätzlich wird auf die wesentlichen, in dieser Arbeit verwendeten Algorithmen eingegangen und die mathematischen Zusammenhänge, welche beim Anpassen der Gewichte eines neuronalen Netzwerks an verschiedene Problemstellungen notwendig sind, dargelegt.

2.1.1 Begriffsabgrenzung

In der Entwicklung des industriellen Zeitalters von handgefertigten Transportmitteln, wie z.B. dem ersten Benz Patent-Motorwagen, hin zu vollautomatischen Produktionsstraßen für Elektrofahrzeuge, sind die dafür verwendeten Maschinen immer eigenständiger geworden. Während der Mensch früher elementarer Bestandteil im Prozessablauf war, muss er heute in der Regel nur noch in überwachender Funktion tätig sein oder Bewegungen ausführen, welche noch zu komplex für eine Automatisierung sind.

Diese Selbstständigkeit geht mit der Eigenschaft einher, dass Maschinen zunehmend in einer Art und Weise agieren, die umgangssprachlich als intelligent bezeichnet wird. Künstliche Intelligenz dient dementsprechend als Überbegriff (vgl. Abbildung 2-1) für eine maschinelle Anpassungsfähigkeit während der Interaktion von Maschinen in einer sich verändernden Umgebung ([31], S.2). Gestützt wird künstliche Intelligenz unter anderem durch das Zusammenspiel vieler Sensoren, welche auf Umgebungseinflüsse und -änderungen reagieren und den Programmablauf entsprechend darauf anpassen.

Diese Intelligenz wird als künstlich bezeichnet, weil sie vorgedacht und einprogrammiert ist ([31], S.2). Als Beispiel dafür kann eine Ampelschaltung an der Hauptverkehrsachse einer industriestarken Großstadt mit hohem Pendleranteil herangezogen werden. Im Normalbetrieb werden dort Haupt- und Nebenachse stets im selben Intervall geschaltet. Während der Zeitspannen des Berufsverkehrs kann es für die innerstädtische Gesamtverkehrslage allerdings vorteilhaft sein, diese Schaltzeiten anzupassen. Dafür wird mit Hilfe von Kameras und Induktionsschleifen das Verkehrsaufkommen gemessen und die Schaltzeiten der Hauptverkehrsachse entsprechend verlängert, um einen Rückstau ins Stadtzentrum zu vermeiden. Sind die Pendler am Ziel angekommen, werden die Nebenachsen wieder normal bedient, um die Mobilität innerhalb der Stadt zu gewährleisten.



Diese Anpassung findet, basierend auf Messdaten, unabhängig vom Zeitpunkt oder Datum statt, was umgangssprachlich als „intelligent“ bezeichnet wird, da z.B. an Feiertagen keine Veränderung der Ampelschaltung aufgrund des fehlenden Berufsverkehrsflusses notwendig ist. Der Verkehr wird also stets sinnvoll und effizient bzw. intelligent geleitet, und es entstehen keine unnötigen Wartezeiten.

Eine Untergruppe der künstlichen Intelligenz bildet das maschinelle Lernen (vgl. Abbildung 2-1). Dabei wird der Fähigkeit von Maschinen Rechnung getragen, selbstständig und aufgrund von Beobachtungen eine Art eigenes Handlungsvermögen zu erlangen ([31], S.3). Die Vorgabe aller möglichen Kombinationen von Ein- und Ausgangsgrößen ist dabei nicht zwingend erforderlich, und es besteht eine gewisse Freiheit in der Umsetzung des Programmcodes. Agiert die Maschine unter bisher unbekanntem Umgebungseinflüssen, kann durch einen vorangegangenen Lernvorgang mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass die getroffene Entscheidung trotz neuer Eingangsparameter zu einem zufriedenstellenden Ergebnis führt.

Im Beispiel der Ampelschaltung kann das maschinelle Lernen z.B. durch eine Umkehrung des vorangegangenen Szenarios verdeutlicht werden. In Folge einer Großveranstaltung am Stadtrand steigt der Verkehr auf der Nebenachse stark an. Über die Hauptachse hat die, nun durch Algorithmen des maschinellen Lernens aufgerüstete, Ampelschaltung bereits gelernt, dass sich eine Verlängerung der Schaltzeiten bei hohem Verkehrsaufkommen vorteilhaft auf den Verkehrsfluss auswirkt. In Folge dessen ist die Schaltung in der Lage, das gelernte Verhalten durch Beobachtungen an der Hauptachse eigenständig auf die Nebenachse zu übertragen. Dafür wäre innerhalb festgelegter Ablaufstrukturen im Gegensatz dazu eine manuelle Anpassung durch den Entwickler (m/w/d) notwendig.

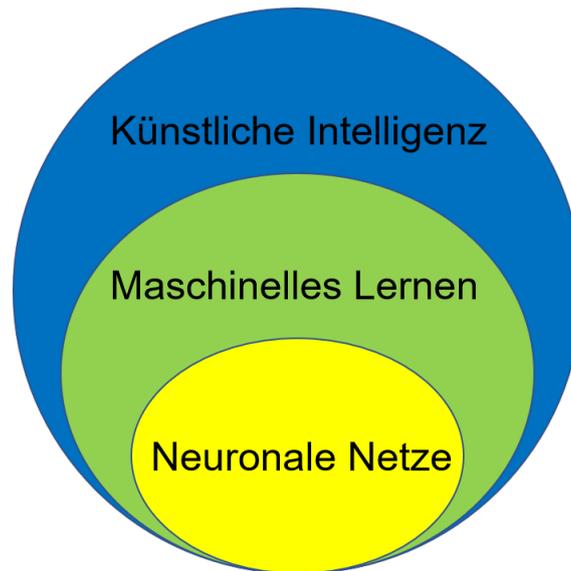


Abbildung 2-1: Begriffsabgrenzung von Schlagwörtern des maschinellen Lernens (i.A. [31], S.4)

Ein weit verbreiteter Bestandteil für die Umsetzung einer maschinellen Lernfähigkeit sind neuronale Netzstrukturen. Diese bilden daher eine Untergruppe des maschinellen Lernens (vgl. Abbildung 2-1) und stellen lernfähige Algorithmen dar, welche dem Aufbau des menschlichen Gehirns nachempfunden sind ([31], S.4). In Abbildung 2-2 ist der Aufbau einer Zelle im menschlichen Gehirn abgebildet. Ausgehend vom Zellkern werden elektrische Impulse über die Axone weitergeleitet und zuletzt an den Synapsen auf andere Zellen übertragen ([71], S.10).

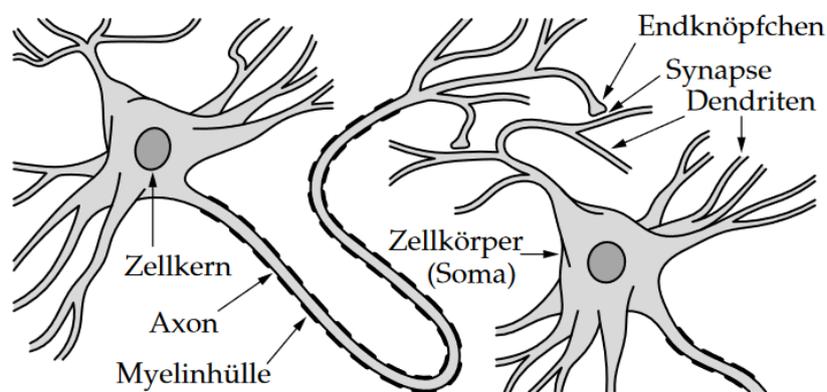


Abbildung 2-2: Aufbau einer Zelle im menschlichen Gehirn ([71], S.10)

In den folgenden Kapiteln wird, ausgehend von einem einzelnen Neuron erläutert, wie dieser Informationsaustausch grundsätzlich am Computer nachgebildet werden kann.



Zudem werden verschiedene Netzstrukturen für die Problemstellungen dieser Arbeit vorgestellt, und es wird auf ausgewählte mathematische Zusammenhänge beim Anpassen der Gewichtungen, bzw. dem Trainingsvorgang eingegangen.

2.1.2 Zusammenhänge an einem Neuron

Der schematische Aufbau einer einzelnen menschlichen Gehirnzelle lässt sich stark vereinfacht durch Vektoren und mathematische Operationen an einem Knotenpunkt beschreiben. In Abbildung 2-3 ist dafür ein Neuron mit drei Eingängen x_1, x_2, x_3 und einem Ausgang y exemplarisch dargestellt (i.A. [66], S.62).

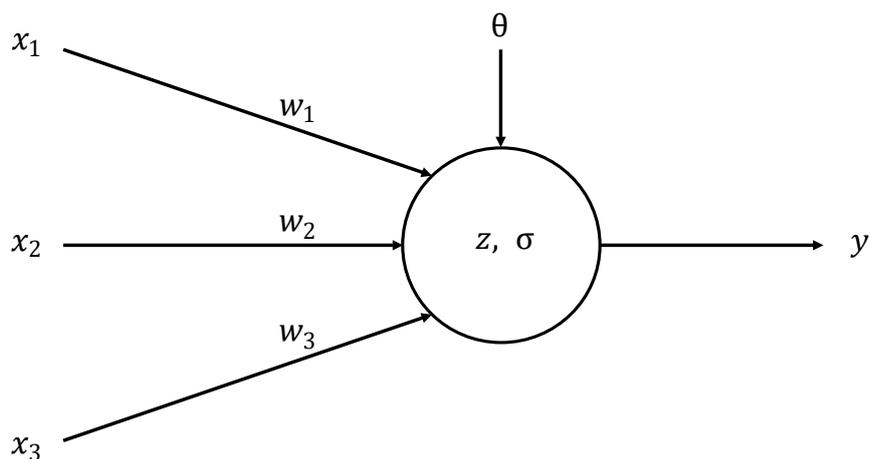


Abbildung 2-3: Mathematische Betrachtung an einem Neuron (i.A. [66], S.62)

Mit Hilfe der Gewichtungen w_1, w_2, w_3 können den Eingängen unterschiedliche Wichtigkeiten für die Bildung der Ausgangsgröße y zugemessen werden. Die Neuronenaktivität z stellt die Verrechnung dieser Größen mit einem Bias θ dar, welcher eine zusätzliche Stellgröße für die Bildung des Ausgangs darstellt, und berechnet sich nach Gleichung (2.1.1) wie folgt ([66], S.62-63).

$$z = \theta + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 \quad (2.1.1)$$

Der Wert für die Ausgabe y wird anschließend mit Hilfe der Aktivierungsfunktion σ gebildet, auf deren Funktion im nächsten Kapitel genauer eingegangen wird.

$$y = \sigma(z) \quad (2.1.2)$$



Allgemein und in vektorieller Schreibweise formuliert, kann die Übertragung von Informationen an einem einzelnen Neuron nach Gleichung (2.1.3) folgendermaßen beschrieben werden.

$$\vec{y} = \sigma(\theta + \vec{x} \times \vec{w}) \quad (2.1.3)$$

Dadurch lässt sich die Berechnungsvorschrift auf beliebig viele Ein- und Ausgangsgrößen erweitern.

2.1.3 Aktivierungsfunktionen

Die Aktivierungsfunktion σ (vgl. Abbildung 2-3 und Gleichung (2.1.3)) ist den Schwellenwerten im menschlichen Gehirn nachempfunden, unterhalb denen keine Übertragung von Informationen stattfindet, bzw. das nachgeschaltete Neuron gehemmt wird. Da bis heute kein Optimierungsverfahren gefunden ist, welches mit sprunghaften Aktivierungsfunktionen arbeiten kann ([71], S.32), wird die Aktivität eines Neurons mit Hilfe von mathematischen Funktionen verändert oder beschränkt ([71], S.13).

In Abbildung 2-4 sind exemplarisch einige, für das Training von neuronalen Netzstrukturen häufig verwendete, Aktivierungsfunktionen abgebildet. Die Sigmoid Funktion und der Tangens hyperbolicus haben die Eigenschaft, Neuronaktivitäten zu begrenzen. Je nach Funktion, können die Wertebereiche für den Ausgang nur $\in (0, 1)$ bzw. $\in (-1, 1)$ betragen. Im Gegensatz dazu verstärkt die Exponentialfunktion die Aktivität eines Neurons im positiven Bereich stark.

Da die Optimierungsverfahren in diesem Kapitel für das Anpassen der Gewichtungen an eine bestimmte Problemstellung mit Hilfe eines Gradientenabstiegsverfahrens erfolgen ([71], S.59), ist die lineare Aktivierungsfunktion dafür nur bedingt geeignet. Die ReLu-Funktion (engl. Rectified Linear Unit) ist eine Weiterentwicklung der Linearfunktion und macht deren Ableitung veränderlich. An der Stelle $x = 0$ ist die ReLu-Funktion unstetig, was in der Praxis mit der Annahme $x'(0) = 0$ gelöst wird und kein Problem bei den Gradientenberechnungen verursacht, da jede Aktivierungsfunktion nur eine Riemann-Integrierbarkeit¹ voraussetzt und keine Stetigkeit ([58], S.2 und [71], S.51). Außerdem ist ReLu-Funktion in der Lage, negative Aktivierungen zu nullen, was einer Neuronenabschaltung gleichkommt.

¹ Eine Funktion darf endlich vielen Unstetigkeiten mit endlich großen Wertebereichen besitzen, um als Riemann-integrierbar zu gelten ([39], S. 51)



Die SeLu-Funktion (engl. Scaled Exponential Linear Unit) ist ein Versuch, Aktivitäten global zu normalisieren. Sie ist so gewählt, dass alle Aktivierungen in einem Netzwerk im Durchschnitt null ergeben und eine Standardabweichung von eins besitzen ([26], S.2). Dies wirkt sich nach ([26], S.9) vorteilhaft auf die Optimierung von Netzstrukturen mit vielen Schichten und Neuronen aus.

Die Wahl der Aktivierungsfunktion hängt stark von der verwendeten Netzstruktur und Problemstellung ab, es kann daher keine grundsätzliche Empfehlung getroffen werden. Während begrenzende Aktivierungsfunktionen aufgrund der komplexen Ableitung aufwändiger zu berechnen sind, haben teillineare oder verstärkende Aktivierungsfunktionen den Nachteil, explodierende Gewichtungen, bzw. Gradienten hervorrufen zu können ([24], S.2).

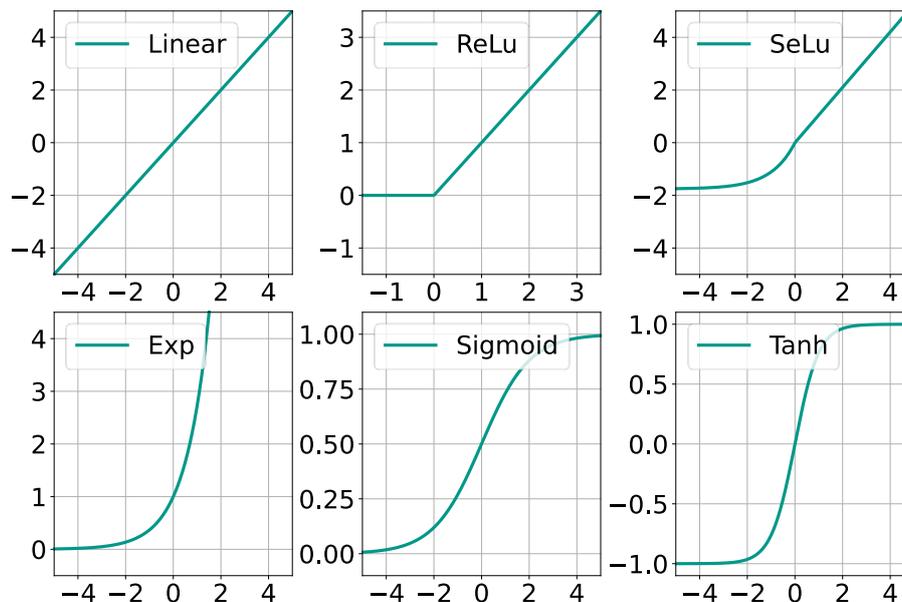


Abbildung 2-4: Verschiedene Aktivierungsfunktionen

2.1.4 Fehlerfunktionen

Fehlerfunktionen werden grundsätzlich dafür verwendet, Abweichungen zwischen einer, als gegeben betrachteten Ausgangsgröße y_{Ist} und einer anderen Vergleichsgröße y_{Soll} gleicher Einheit zu berechnen. Im Bereich der Simulation und des maschinellen Lernens können dadurch z.B. Fehler zwischen Messdaten und Modellvorhersagen eines verwendeten Algorithmus bestimmt werden. Die Berechnung des durchschnittlichen absoluten und quadratischen Fehlers wird dabei als selbstverständlich betrachtet, nur der Vollständigkeit wegen und daher ohne Quellen angegeben.



Eine einfache und häufig verwendete Fehlerfunktion ist der durchschnittliche absolute Fehler (engl. Mean Absolute Error - MAE) nach Gleichung (2.1.4), bei dem jeder Abweichung ihr Betrag zugeordnet wird und daraus anschließend der Mittelwert gebildet wird.

$$MAE = \sum_{n=0}^i \frac{|y_{Ist} - y_{Soll}|}{i} \quad (2.1.4)$$

Um hohe Abweichungen stärker zu gewichten, wird der durchschnittliche quadratische Fehler (engl. Mean Squared Error - MSE) nach Gleichung (2.1.5) verwendet. Dabei werden die Abweichungen zuerst quadriert, bevor daraus anschließend ein Mittelwert gebildet wird.

$$MSE = \sum_{n=0}^i \frac{(y_{Ist} - y_{Soll})^2}{i} \quad (2.1.5)$$

Durch die Fehlerfunktion nach Huber werden Fehler, die einen bestimmten Schwellenwert δ unterschreiten, quadratisch gewichtet ([50], S.74-75). Die darüber liegenden werden linear und absolut gewichtet. Dementsprechend wird bei der Fehlerberechnung vor der Mittelung eine Fallunterscheidung nach Gleichung (2.1.16) getroffen, wie in Abbildung 2-5 für $\delta = 1$ veranschaulicht ist. Das hat den Vorteil, dass Fehler, welche bereits unter 1 liegen, durch die quadratische Gewichtung noch geringer ausfallen und somit ein robusteres Fehlerbild bei der Gesamtfehlerberechnung entsteht ([50], S.73).

$$H_{\delta}(\Delta y) = \begin{cases} \frac{1}{2} \Delta y^2, & |\Delta y| \leq \delta; \quad \Delta y = y_{Ist} - y_{Soll} \\ \delta * (|\Delta y| - \frac{1}{2} * \delta) & \end{cases} \quad (2.1.6)$$

$$MH = \sum_{n=0}^i \frac{H_{\delta}(\Delta y)}{i}$$

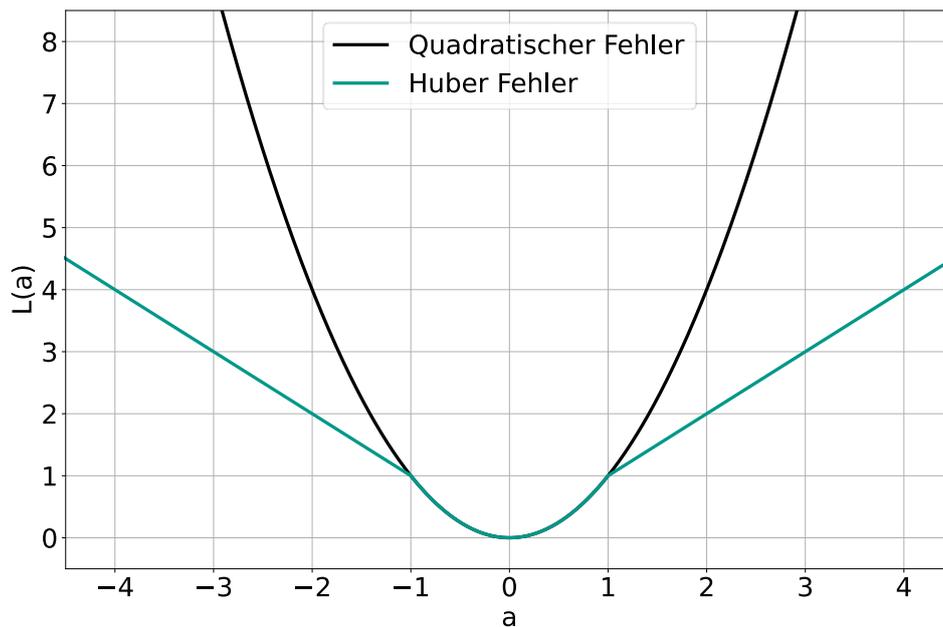


Abbildung 2-5: Huber Fehlerfunktion für $\delta = 1$

2.1.5 Optimierungsverfahren

Das Prinzip der Regression ist alt. Für eine gegebene Menge an Teilpunkten, wie sie z.B. bei Geschwindigkeitsmessungen während einer Testfahrt entstehen können (vgl. Abbildung 2-6), soll näherungsweise eine Gesetzmäßigkeit gefunden werden (i.A. [71], S.479-481, Beweisführung nach [27]).

Die Messpunkte lassen erahnen, dass sich eine lineare Regressionsgerade eignet, um eine allgemeine Beschreibung zwischen der Zeit und der gemessenen Geschwindigkeit im betrachteten Zeitfenster herzustellen. Daher wird grundsätzlich von folgendem linearen Zusammenhang nach Gleichung (2.1.7) ausgegangen, bei der a für die y -Verschiebung und b für die Steigung steht.

$$y = a + bx \quad (2.1.7)$$

Mit Hilfe einer Minimierung der Fehlerquadrate können die Parameter a und b so gewählt werden, dass alle Punkt bestmöglich durch eine Ausgleichsgerade (grün) beschrieben werden. Die Wahl der Quadratsumme als Fehlerfunktion hat den Vorteil, dass die Fehlerfunktion überall stetig differenzierbar ist, im Gegensatz zum absoluten Fehler, der um die Null eine Unstetigkeit aufweist ([71], S.480).

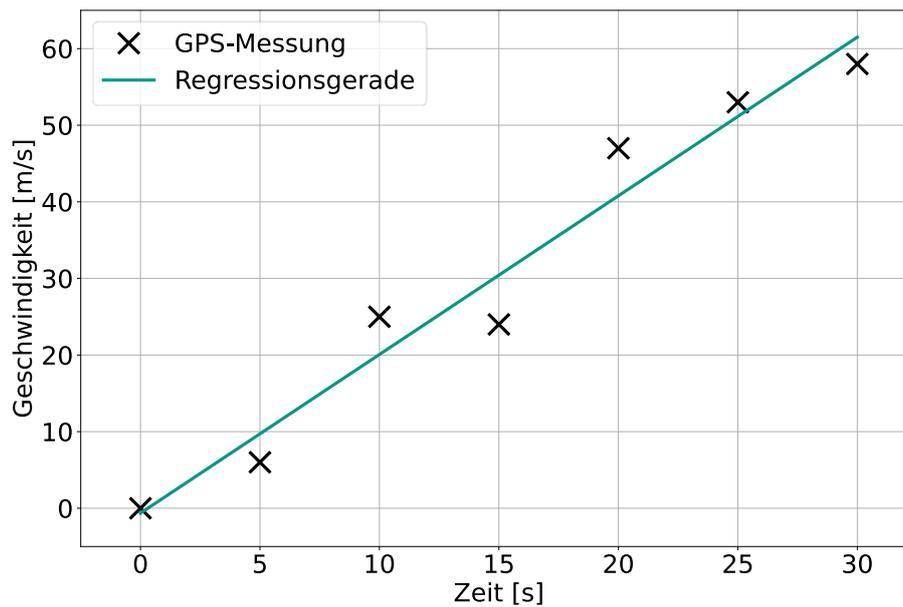


Abbildung 2-6: Regressionsgerade einer Geschwindigkeitsmessung

Die Quadratsumme der Fehler berechnet sich zusammengesetzt aus den Einzelfehlern nach Gleichung (2.1.8) folgendermaßen.

$$F(a, b) = \sum_{i=0}^n (a + bx_i - y_i)^2 \quad (2.1.8)$$

Eine notwendige Bedingung, damit die Fehlerfunktion $F(a, b)$ überhaupt ein Minimum besitzt, ist das Entfallen der partiellen Ableitungen nach den Parametern a und b (vgl. Gleichung (2.1.9) und [71], S.480).

$$\begin{aligned} \frac{\partial F}{\partial a} &= \sum_{i=0}^n 2(a + bx_i - y_i) = 0 \\ \frac{\partial F}{\partial b} &= \sum_{i=0}^n 2(a + bx_i - y_i)x_i = 0 \end{aligned} \quad (2.1.9)$$

Daraus lässt sich durch Umformen folgendes Gleichungssystem (2.1.10) erstellen, welches auch als Normalgleichungssystem bezeichnet wird ([71], S.480).



$$\begin{aligned}na + \left(\sum_{i=0}^n x_i\right)b &= \sum_{i=0}^n y_i \\ \left(\sum_{i=0}^n x_i\right)a + \left(\sum_{i=0}^n x_i^2\right)b &= \sum_{i=0}^n x_i y_i\end{aligned}\tag{2.1.10}$$

Dieses Gleichungssystem besitzt eine eindeutige Lösung (Beweisführung in [27]), welche auch als Regressionsgerade bezeichnet wird [27]. Numerisch und für die Messpunkte entsprechend Abbildung 2-6 am Computer berechnet, besitzt dieses Gleichungssystem die Lösung $a = 1.250$ und $b = 2.079$. Physikalisch interpretiert bedeutet das, die Fahrzeugmessungen werden näherungsweise bei einer konstanten Beschleunigung von ca. $2 \frac{m}{s^2}$ durchgeführt und mit einer Geschwindigkeit von ca. $1.2 \frac{m}{s}$ gestartet.

Dieses Verfahren lässt sich durch eine beliebige Erweiterung der Polynome auch auf mehrdimensionale Problemstellungen anpassen [27]. Im Zusammenhang mit dem Training von neuronalen Netzwerken liefert die multilineare Regression eine Lösung für ein Neuron mit beliebig vielen Ein- und Ausgängen und linearer Aktivierungsfunktion.

Diese Methodik kann nun um die Sigmoid-Aktivierungsfunktion erweitert werden, was auch als logistische Regression bezeichnet wird ([71], S.53). Dafür wird die Sigmoid-Funktion, welche z.B. bei der Beschreibung von Infektionszahlen während einer Pandemie verwendet wird (vgl. Abbildung 2-7), mit Hilfe einer Logarithmierung in Gleichung (2.1.11) linearisiert ([71], S.58). Da die Sigmoid-Funktion nur zwischen null und eins definiert ist, wird zusätzlich die Streckung Y eingeführt.

$$\begin{aligned}y &= \frac{Y}{1 + e^{a+bx}} \\ \frac{1}{y} &= \frac{1 + e^{a+bx}}{Y} \\ \frac{Y - y}{y} &= e^{a+bx} \\ \ln\left(\frac{Y - y}{y}\right) &= a + bx\end{aligned}\tag{2.1.11}$$



Durch die Logarithmierung können die Messdaten entsprechend der linken Seite umgerechnet und wieder mit Hilfe eines Normalgleichungssystems an die Problemstellung angepasst werden. Wird nun ein weiteres Mal die Methodik der kleinsten Fehlerquadrate angewandt, ergibt sich numerisch berechnet für $a = 0.02354$ und $b = -1.0433$. Aus der Kurvensättigung folgt $Y = 80000$.

Die Gewichtungen für das Erlernen der Infektionszahlen an einem Neuron mit Sigmoid Aktivierung und einem Ein- und Ausgang sind dementsprechend Bias $\theta = a$ und Gewicht $w = b$ ([71], S.58). Zusätzlich wird ein Multiplikator $f(y) = Yy$ für den Offset zur sigmoiden Normfunktion benötigt.

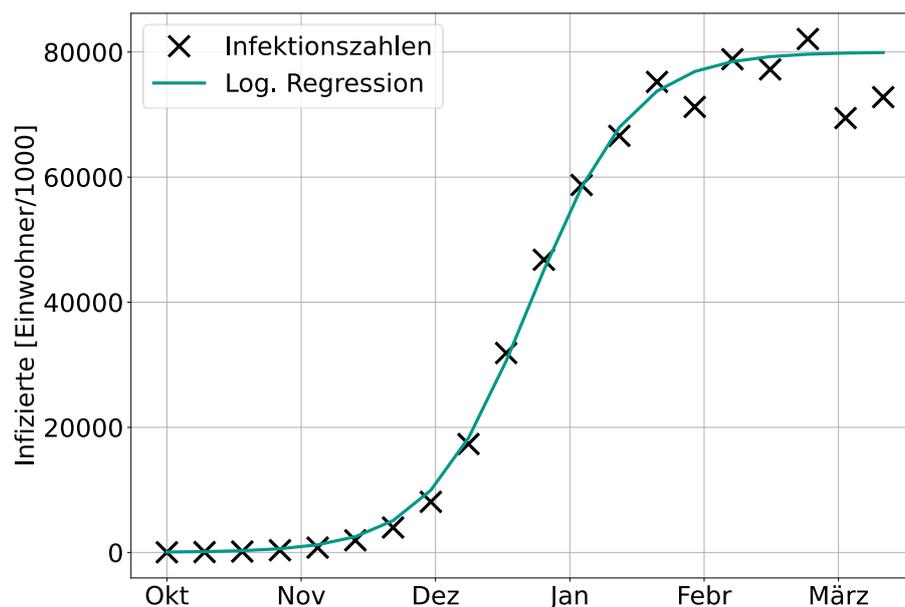


Abbildung 2-7: Logistische Regressionskurve für Infektionszahlen in einer Pandemie

Diese Vorgehensweise lässt sich ebenfalls, ähnlich zur multilinenen Regression, auf mehrere Ein- und Ausgänge anwenden ([71], S.58).

Da die Berechnung der Fehlerquadratsumme und deren Minimum auf die Ausgangsschicht beschränkt ist ([71], S.58), eignet sich dieses explizite Verfahren nicht für Betrachtungen an neuronalen Netzstrukturen mit mehreren Schichten.

Für das Optimieren von mehrschichtigen Perzeptren sind Überlegungen zu einer schrittweisen Annäherung der Fehlerfunktion auf eine Problemstellung mit anschließender Fehlerrückübertragung an die vorangegangenen Schichten, notwendig. Im Prinzip werden die Gewichte und Biase dafür erst zufällig initialisiert und anschließend nach und nach verändert, bis sich eine zufriedenstellende Ausgabe einstellt, weshalb dieses Vorgehen auch als „Methode des rekursiven Gradientenabstiegs“ bezeichnet wird ([71], S.59).



Über die Ableitung der Fehlerfunktionen lässt sich während der Anpassung der Parameter die Richtung bestimmen, in welche die Gewichtungen angepasst werden müssen. Werden die Gewichte dabei stetig entgegen des Gradienten bewegt, lässt sich dadurch zumindest ein lokales Optimum finden ([71], S.59).

Durch die Differenzierbarkeit der Aktivierungsfunktion lässt sich die Fehlerfunktion nach allen Schichten partiell und rekursiv ableiten, und der Fehler wird rückwärts durch das Netz zurückgeführt ([71], S.59). Die Gewichte werden dadurch auch in anderen, nicht Ausgangsschichten, Stück für Stück angepasst.

Im Folgenden soll eine allgemeine Rekursionsformel nach ([71], S.59-64) für ein Neuron u in einer der versteckten Schichten k mit der Neuronenmenge U_k eines Perzeptrons mit r Schichten, $u \in U_k, 0 < k < r - 1$, bezogen auf ein einzelnes Lernmuster, $l = (\vec{\zeta}^{(l)}, \vec{\delta}^{(l)})$ hergeleitet werden. Seine Vorgängerneuronen sind $pred(u) = \{p_1, \dots, p_n\} \subseteq U_{k-1}$, der zugehörige Gewichtsvektor ist $\vec{w}_u = (-\theta_u, w_{up_1}, \dots, w_{up_n})$ mit dem Bias θ_u .

Die Ausgabe $out_v^{(l)}$ eines Ausgabeneurons v wird lediglich indirekt von der Netzeingabe $net_u^{(l)}$ des betrachteten Neurons u durch seine Nachfolgeneuronen $succ(u) = \{s \in U \mid (u, s) \in C\} = \{s_1, \dots, s_m\} \subseteq U_{k+1}$ über deren Netzeingabe $net_s^{(l)}$ beeinflusst.

Der Fehler $e^{(l)}$ ergibt sich wieder aus der Fehlerquadratsumme nach Gleichung (2.1.12), diesmal über alle Ausgabeneuronen U_{out} .

$$e^{(l)} = \sum_{v \in U_{out}} e^{(l)} = \sum_{v \in U_{out}} \left(o_v^{(l)} - out_v^{(l)} \right)^2 \quad (2.1.12)$$

Durch Anwendung der Kettenregel ergibt sich für die partielle Ableitung der Fehlerfunktion nach der Netzeingabe Gleichung (2.1.13).

$$\frac{\partial e^{(l)}}{\partial net_u^{(l)}} = \delta u^{(l)} = \sum_{v \in U_{out}} \sum_{s \in succ(u)} \left(o_v^{(l)} - out_v^{(l)} \right) \frac{\partial out_v^{(l)}}{\partial net_s^{(l)}} \frac{\partial net_s^{(l)}}{\partial net_u^{(l)}} \quad (2.1.13)$$

Beide Summen sind endlich, daraus ergibt sich Gleichung (2.1.14).



$$\begin{aligned}\delta u^{(l)} &= \sum_{s \in \text{succ}(u)} \left(\sum_{v \in U_{\text{out}}} (o_v^{(l)} - \text{out}_v^{(l)}) \frac{\partial \text{out}_v^{(l)}}{\partial \text{net}_s^{(l)}} \right) \frac{\partial \text{net}_s^{(l)}}{\partial \text{net}_u^{(l)}} = \\ &= \sum_{s \in \text{succ}(u)} \delta s^{(l)} \frac{\partial \text{net}_s^{(l)}}{\partial \text{net}_u^{(l)}}\end{aligned}\quad (2.1.14)$$

Für die partielle Ableitung der Netzeingabe wird die Netzeingabe zuerst nach allgemeiner Definition und unter Berücksichtigung der vorangegangenen Ausgaben $\text{out}_p^{(l)}$ entsprechend Gleichung (2.1.15) aufgestellt.

$$et_s^{(l)} = \vec{w}_s \vec{i}n_s^{(l)} = \left(\sum_{p \in \text{pred}(s)} w_{sp} \text{out}_p^{(l)} \right) - \theta_s \quad (2.1.15)$$

Ein Element von $\vec{i}n_s^{(l)}$ ist die Ausgabe $\text{out}_u^{(l)}$ des Neurons u , weil $\text{out}_{p_s}^{(l)} = \text{out}_u^{(l)}$ ist. Dementsprechend hängen $\text{net}_s^{(l)}$ und $\text{net}_u^{(l)}$ nur über $\text{out}_u^{(l)}$ zusammen, woraus Gleichung (2.1.16) folgt.

$$\frac{\partial \text{net}_s^{(l)}}{\partial \text{net}_u^{(l)}} = \left(\sum_{p \in \text{pred}(s)} w_{sp} \frac{\partial \text{out}_p^{(l)}}{\partial \text{net}_u^{(l)}} \right) - \frac{\partial \theta_s}{\partial \text{net}_u^{(l)}} = w_{su} \frac{\partial \text{out}_u^{(l)}}{\partial \text{net}_u^{(l)}} \quad (2.1.16)$$

Da die Summe nur für $p = u$ benötigt wird, verschwinden einige Terme, und es ergibt sich Gleichung (2.1.17).

$$\delta u^{(l)} = \left(\sum_{s \in \text{succ}(u)} \delta s^{(l)} w_{su} \right) \frac{\partial \text{out}_u^{(l)}}{\partial \text{net}_u^{(l)}} \quad (2.1.17)$$

Dadurch ergibt sich eine schrittweise Rekursionsformel für die Berechnung der partiellen Ableitungen der Neuronen in den versteckten Schichten.



Der Term

$$\sum_{s \in \text{succ}(u)} \partial s^{(l)} w_{su}$$

beschreibt die Differenz $o_u^{(l)} - \text{out}_u^{(l)}$ zwischen optimaler und tatsächlicher Ausgabe des Neurons u für das Lernmuster l .

Diese Differenz kann auch als Fehlerwert bezeichnet werden, und mit deren Hilfe lassen sich die Fehlerwerte rekursiv von der Ausgangsschicht auf die vorangegangenen Schichten zurückrechnen, was auch als Fehlerrückübertragung (engl. Error Backpropagation) bezeichnet wird ([71], S.64).

Mit der Einführung einer Lernrate η , welche ein Maß für die größtmöglich zugelassene Anpassung während einem Lernvorgang ist, ergibt sich Gleichung (2.1.18) für die Änderung der Gewichtung an einem betrachteten Neuron.

$$\begin{aligned} \Delta \vec{w}_u^{(l)} &= -\frac{\eta}{2} \vec{v}_{\vec{w}_u} e^{(l)} = \eta \delta u^{(l)} i \vec{n}_u^{(l)} = \\ &= \eta \left(\sum_{s \in \text{succ}(u)} \partial s^{(l)} w_{su} \right) \frac{\partial \text{out}_u^{(l)}}{\partial \text{net}_u^{(l)}} i \vec{n}_u^{(l)} \end{aligned} \quad (2.1.18)$$

Um Rechenzeit zu sparen, findet die Anpassung der Gewichtungen häufig über mehrere Lernmuster statt. Dafür wird die Änderung $\Delta \vec{w}_u^{(l_n)}$ über alle Lernmuster l_1 bis l_n aufsummiert und die Anpassung der Gewichte erst anschließend durchgeführt. Je nach Lernrate kann es zum Versagen des beschriebenen Verfahrens kommen, z.B. wenn viele Lernmuster in Kombination mit einer hohen Lernrate die Gewichte anfangs zu stark von der Problemstellung entfernen. Deshalb empfiehlt es sich an die vorgestellte und iterative Berechnungsvorschrift zuerst konservativ, mit wenigen Lernmustern und kleinen Lernraten, bei einem entsprechenden Optimierungsproblem heranzugehen.

Bei zu kleinen Lernraten kann es jedoch passieren, dass die Anpassung der Gewichte nicht ausreichend stark erfolgt und die Näherungslösung auch nach vielen Iterationsschritten kein Optimum findet.

Daher wurde in weiterführender Forschung das Optimierungsverfahren ADAM (engl. Adaptive Momentum Estimation) entwickelt. Dieses Verfahren beruht ebenfalls auf der Methode des stochastischen Gradientenabstiegs. Die Gewichtungen werden dabei allerdings proportional zu ihren Gradienten geändert. Das Verfahren besitzt also eine an den Lernfortschritt anpassungsfähige Lernrate ([19], S.3).



Dabei werden für jedes Lernmuster l die Momente m_t und v_t rekursiv mit Hilfe des Gradienten ∇_t nach Gleichung (2.1.19) berechnet.

$$\begin{aligned} m_k &= \beta_1 m_{k-1} + (1 - \beta_1) \nabla_k, & m_0 &= 0, & k &\in [l] \\ v_k &= \beta_2 v_{k-1} + (1 - \beta_2) \nabla_k^2, & v_0 &= 0, & k &\in [l] \end{aligned} \quad (2.1.19)$$

Da die Momente bei den ersten Iterationen gegen Null konvergierten ([19], S.3), werden die korrigierten Momente \hat{m}_k und \hat{v}_k nach Gleichung (2.1.20) eingeführt.

$$\begin{aligned} \hat{m}_k &= \frac{m_k}{1 - \beta_1^k} \\ \hat{v}_k &= \frac{v_k}{1 - \beta_2^k} \end{aligned} \quad (2.1.20)$$

Die Lernrate berechnet sich anschließend nach Vorgabe einer größtmöglich zulässigen Lernrate η entsprechend Gleichung (2.1.21).

$$n_k = \frac{\eta}{\sqrt{\hat{v}_k} + \varepsilon} \hat{m}_k \quad (2.1.21)$$

Der Wert ε liegt im Bereich $\varepsilon \in [10^{-8}, 10^{-6}]$ und garantiert die numerische Stabilität des Verfahrens ([19], S.2), da sich die Lernrate gegen Ende eines Trainingsvorgangs bei sehr kleinen Momenten trotzdem weiter verringern kann.

Die empfohlenen Parameter bei der Verwendung des ADAM-Optimierungsverfahrens sind nach ([19], S.2) in Tabelle 2-1 gelistet.

Parameter:	Wert:
β_1	0.9
β_2	0.999
ε	10^{-7}

Tabelle 2-1: Parametrierung des ADAM-Optimierungsverfahrens



2.1.6 Netzarchitekturen

Ausgehend von den Betrachtungen in Kapitel 2.1.2, können Neuronen auf verschiedene Art und Weise miteinander verknüpft werden. Grundsätzlich wird bei der Regression zwischen kontinuierlichen und rückgekoppelten Netzstrukturen unterschieden ([66], S. 79).

Kontinuierliche Netzstrukturen, wie exemplarisch in Abbildung 2-8 dargestellt, können Informationen nur vom Eingang zum Ausgang weitergeben und lassen sich deshalb vergleichsweise einfach berechnen. Zwischen Ein- und Ausgangsschicht können beliebig viele Zwischenschichten liegen, die Berechnungsvorschriften ändern sich dadurch nicht. Für mehrdimensionale und stationäre Optimierungsprobleme lassen sich gute Ergebnisse in Kombination mit einer geringen Rechendauer erzielen, weshalb kontinuierliche Netze den Einstieg für Näherungslösungen mit Hilfe von neuronalen Netzen und deren Verständnis in der Regression bilden.

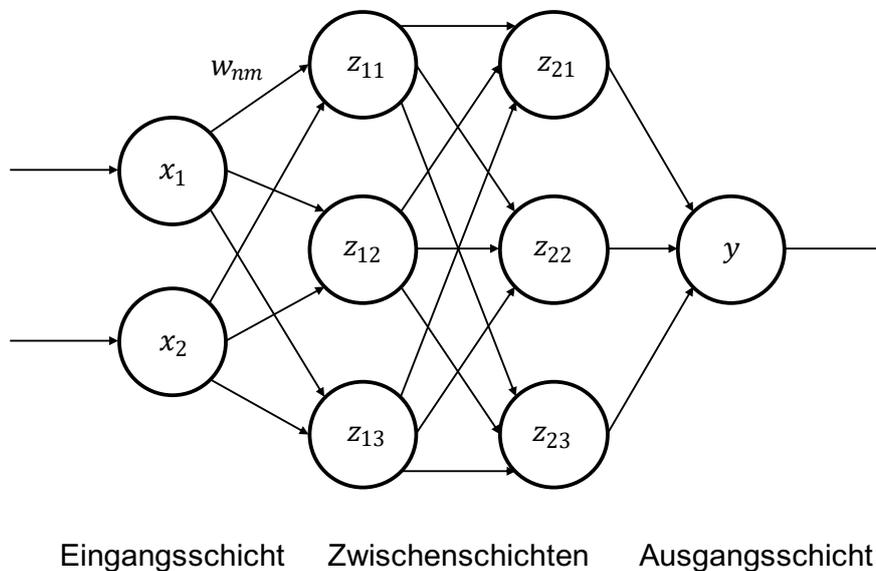


Abbildung 2-8: Kontinuierliches Neuronales Netz mit zwei Zwischenschichten

In Abbildung 2-8 ist kontinuierliches neuronales Netzwerk mit zwei Zwischenschichten abgebildet. Mit Hilfe von Vektormultiplikationen wird der Ausgangswert y über die Gewichtungen w berechnet. Die Eingänge x hängen demnach mathematisch über Gleichung (2.1.22), analog zu Gleichung (2.1.3), folgendermaßen mit dem Ausgang ohne Aktivierungsfunktion zusammen.

$$y = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \times \begin{pmatrix} w_{11}z_{11} \\ w_{12}z_{12} \\ w_{13}z_{13} \end{pmatrix} \times \begin{pmatrix} w_{21}z_{21} \\ w_{22}z_{22} \\ w_{23}z_{23} \end{pmatrix} \quad (2.1.22)$$



Für Problemstellungen, bei denen vorangegangene Werte für die Berechnung des Ausgangs notwendig sind, wie zum Beispiel bei der rekursiven Berechnung von Differentialgleichung, werden rückgekoppelte Netzstrukturen verwendet ([71], S.146-148).

Demnach lässt sich eine beliebige explizite Differentialgleichung durch ein rückgekoppeltes neuronales Netzwerk abbilden. Dafür wird eine gegebene explizite Differentialgleichung n-ter Ordnung nach Gleichung (2.1.23)

$$\begin{aligned}x^{(n)} &= f(t, x, \dot{x}, \ddot{x}, \dots, x^{n-1}) \\y_1 &= \dot{x}, y_2 = \ddot{x}, \dots, y_{n-1} = x^{n-1}\end{aligned}\tag{2.1.23}$$

unter der Einführung von $n - 1$ Ausgangsgrößen y in folgendes Gleichungssystem nach Gleichung (2.1.24), bestehend aus n gekoppelten Differentialgleichungen erster Ordnung, überführt.

$$\begin{aligned}\dot{x} &= y_1 \\ \dot{y} &= y_2 \\ &\vdots \\ \dot{y}_{n-2} &= y_{n-1} \\ \dot{y}_{n-1} &= f(t, x, y_1, y_2, \dots, y_{n-1})\end{aligned}\tag{2.1.24}$$

Wird der Differentialquotient durch einen Differenzenquotient ersetzt, ergeben sich die n Rekursionsformeln zur Beschreibung des Differentialgleichungssystems nach Gleichung (2.1.25).

$$\begin{aligned}x(t_i) &= x(t_{i-1}) + \Delta t * y_1(t_{i-1}) \\ y_1(t_i) &= y_1(t_{i-1}) + \Delta t * y_2(t_{i-1}) \\ &\vdots \\ y_{n-2}(t_i) &= y_{n-2}(t_{i-1}) + \Delta t * y_{n-3}(t_{i-1}) \\ y_{n-1}(t_i) &= y_{n-1}(t_{i-1}) + f(t_{i-1}, x(t_{i-1}), y(t_{i-1}), \dots, y_{n-1}(t_{i-1}))\end{aligned}\tag{2.1.25}$$



Jede dieser Gleichungen lässt sich nun über ein Neuron beschreiben, welches die Größen auf der linken Seite mit Hilfe der rechten Seite berechnet. Wird für die Neuronen weiterhin die Größe z verwendet und für die Gewichtungen $w = \Delta t z$ eingeführt, kann dieser Zusammenhang nach Abbildung 2-9 exemplarisch veranschaulicht werden (i.A. [71], S. 148).

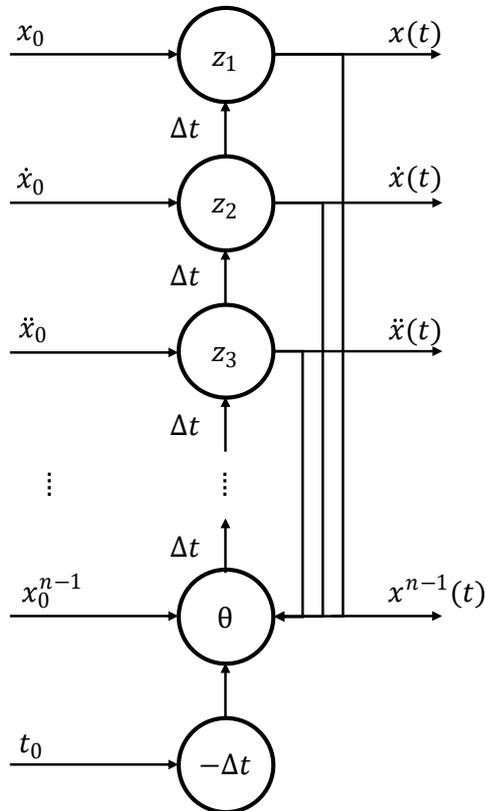


Abbildung 2-9: Rückgekoppeltes Netz zur Berechnung von expliziten Differentialgleichungen (i.A. [71], S. 148)

Für die Modellierung von transienten Vorgängen ist es wichtig, Informationen über längere Zeiträume zu speichern. Rekurrente Netzarchitekturen sind theoretisch in der Lage, mehrere vorangegangene Zeitschritte über weitere Rückkopplungen zu berücksichtigen. Dadurch verzögert sich die Fehlerrückführung aber zunehmend, und die Berechnung des Gesamtsystems wird verlangsamt ([61], S.1).

Eine Weiterentwicklung der rekurrenten Netzstrukturen stellen LSTM-Zellen (engl. Long Short-Term Memory) dar ([61], S.1). Anstatt über einzelne und rückgekoppelte Neuronen wird in Zellstrukturen mit veränderbaren Informationsflüssen (engl. Gates) versucht, eine Art Kurz- und Langzeitgedächtnis zu implementieren, was zu einem konstanteren und schnelleren Trainingserfolg als bei rekurrenten Netzwerken führt ([61], S.1).

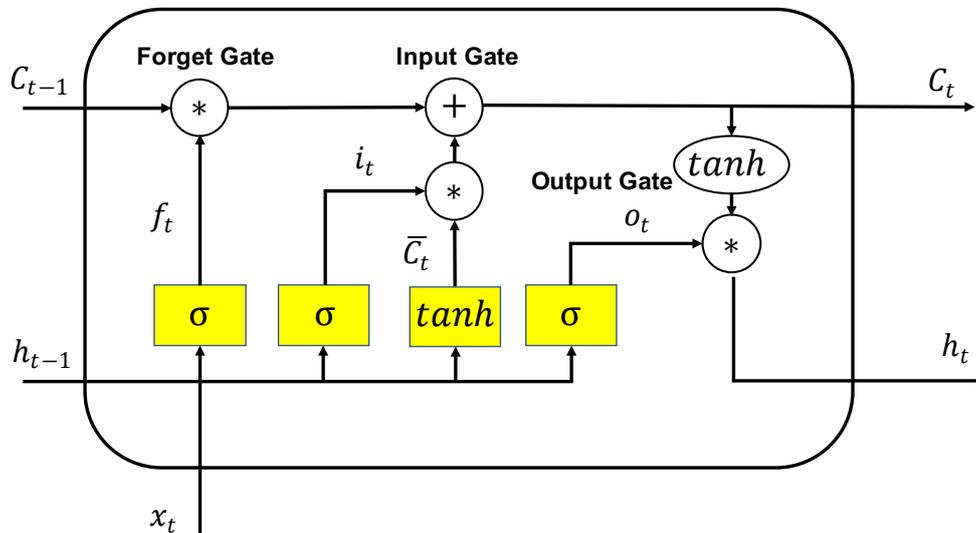


Abbildung 2-10: Schematischer Aufbau einer LSTM Zelle mit verschiedenen Schleusenfunktionen (i.A. [11])

In Abbildung 2-10 ist der schematische Aufbau einer LSTM Zelle mit ihren verschiedenen Schleusen dargestellt (Bild i.A. [11] und Gleichungen nach [61]). Informationen mehrerer vorangegangener Zeitschritte können dabei über den Zellstatus C (engl. Cell State) gewichtet und weitergeleitet werden, welcher überwiegend für die Ausbildung eines Langzeitgedächtnisses verantwortlich ist. Der Zellausgang h (engl. Hidden State) gibt Aufschluss über den aktuellen Zeitschritt und stellt den Informationsfluss für das Kurzzeitgedächtnis bereit.

Zuerst wird über die Gewichtungen w_f und b_f , den Bias θ_f (nicht in Abbildung 2-10 enthalten), dem vorangegangenen Zellausgang h_{t-1} und aktuellen Eingang x_t berechnet, wie viel der bisherigen Langzeitinformationen vorangegangener Zeitschritte C_{t-1} behalten oder nicht behalten, bzw. vergessen werden soll. Dieses Gate wird daher auch als (engl.) Forget Gate bezeichnet. $f_t = 1$ bedeutet dabei vollständiges Beibehalten, $f_t = 0$ vollständiges Verwerfen der Informationen und wird über die Multiplikation mit einer Sigmoid-Funktion $\sigma \in (0,1)$ angenähert. Gleichung (2.1.26) beschreibt die mathematischen Zusammenhänge an dieser ersten Schleuse.



$$f_t = \sigma(w_f \times [h_{t-1}, x_t] + \theta_f), \quad f_t \in (0, 1) \quad (2.1.26)$$

Anschließend werden am (engl.) Input Gate neue Informationen aus dem vorangegangenen und aktuellen Zeitschritt zum bisherigen Zellstatus addiert. Dafür werden ein potentieller neuer Zellstatus \bar{C}_t und der aktuelle Eingang i_t mit Hilfe der Gewichtungen $w_c, \theta_c, w_i, \theta_i$ (nicht in Abbildung 2-10 enthalten), dem vorangegangenen Zellausgang h_{t-1} und dem aktuellen Eingang x_t gebildet. Die Aktivierungsfunktion \tanh dient dabei einer Regulierung des Langzeitinformationsflusses und beugt vor, dass einzelne Zellstatus sehr große oder kleine Werte annehmen (engl. Overflow), da $\tanh \in (-1, 1)$. Abschließend kann der aktuelle Zellstatus C nach Gleichung (2.1.27) berechnet werden.

$$\begin{aligned} \bar{C}_t &= \tanh(w_c \times [h_{t-1}, x_t] + \theta_c) \\ i_t &= \sigma(w_i \times [h_{t-1}, x_t] + \theta_i) \\ C_t &= C_{t-1} * f_t + i_t * \bar{C}_t \end{aligned} \quad (2.1.27)$$

Zuletzt wird der neue Zellausgang h_t am (engl.) Output Gate nach Gleichung (2.1.28) gebildet. Dafür wird über die Gewichtungen w_o, θ_o (nicht in Abbildung 2-10 enthalten), dem vorangegangenen Zellausgang h_{t-1} und dem aktuellen Eingang x_t erst ein potentieller neuer Ausgang o_t berechnet, welcher dann abschließend, zusammen mit dem aktuellen Zellstatus, den neuen Zellausgang ergibt.

$$\begin{aligned} o_t &= \sigma(w_o \times [h_{t-1}, x_t] + \theta_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (2.1.28)$$

Auch hier wird mit Hilfe eine \tanh -Funktion versucht, den Kurzzeitinformationsfluss zu regulieren und zu verhindern, dass einzelne Zellausgänge sehr große oder kleine Werte annehmen.

Über diese anpassbaren Gewichtungen und Schleusen sind LSTM-Zellen in der Lage, Informationen vorangegangener Lernmuster für das Lösen einer aktuellen Problemstellung zu speichern und wiederzuverwenden (vgl. [61], S.6).



Bei der Verschaltung von mehrschichtigen neuronalen Netzwerken mit Hilfe von LSTM-Zellen gibt es mehrere Möglichkeiten. Zum einen werden bei der Betrachtung von Zeitreihen mehrere Zellen nebeneinandergeschaltet, sodass für jeden Zeitschritt t eine Zelle aktiv ist, wie in Abbildung 2-11 veranschaulicht. Über den Zellausgang h_t und den Zellstatus C_t werden Informationen vom ersten zum letzten Zeitschritt geleitet.

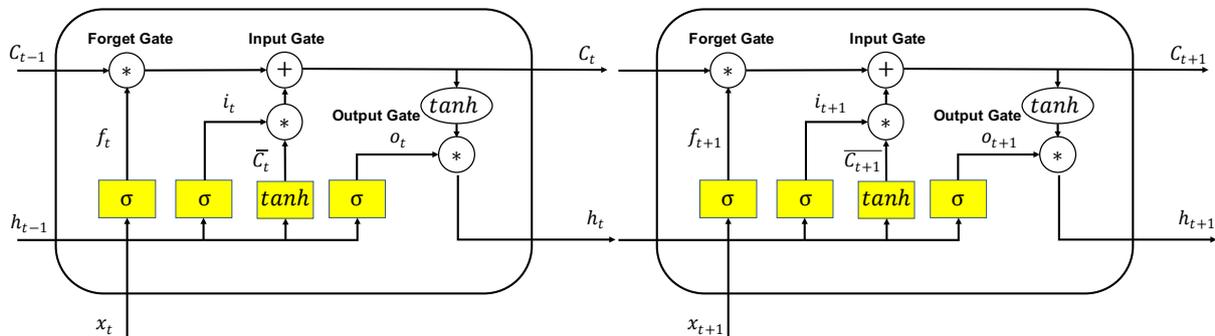


Abbildung 2-11: Nebeneinanderschaltung von LSTM-Zellen zur Zeitreihenmodellierung

Weiter können eine Schicht und eine Zelle hintereinandergeschaltet werden, sodass dazwischen nur der letzte Zellstatus C_{t+n} und der letzte Ausgang h_{t+n} weitergegeben werden (Abbildung 2-12 - `return_state=True`, `return_sequences=False`)² und die neue Zelle damit, gemeinsam mit einem neuen Eingang x_k , initialisiert wird. Das bedeutet konkret: In der oberen Schicht liegt keine Zeitreihe mehr vor. Durch die Weiterleitung des Zellstatus und -ausgangs werden allerdings Informationen über die vergangenen Zeitschritte aus der vorigen Schicht übertragen und die Netzstruktur ist dadurch an der Ausgangsschicht in der Lage, basierend auf der Eingangszeitreihe, Werte eines zusammenhängenden nächsten Zeitschritts vorherzusagen. Dies kann z.B. nützlich sein, wenn ein neuer Wert basierend auf einem Zeitverlauf und einem Startwert vorhergesagt werden soll.

² `return_state` und `return_sequences` beziehen sich auf die Einstellungsbenennung in Tensorflow 2.7

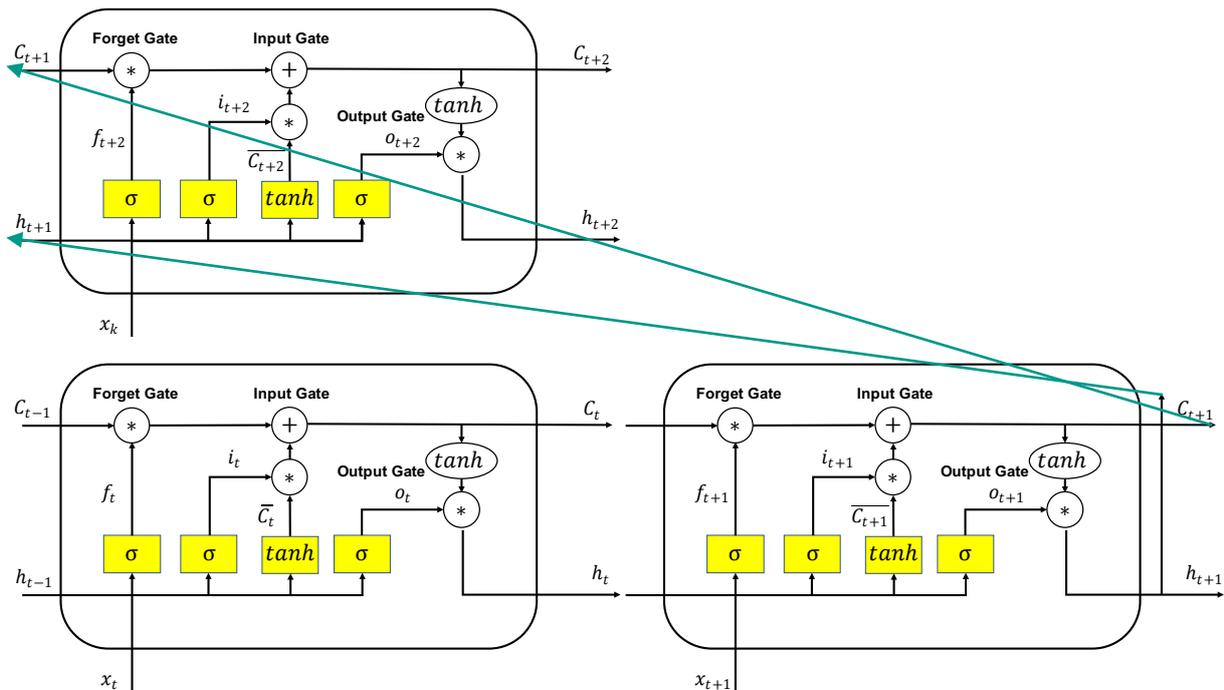


Abbildung 2-12: Schichtung von LSTM-Zellen I (return_state=True, return_sequences=False)

Zum anderen lassen sich zwei Schichten so hintereinanderschalten, dass alle Ausgänge der kompletten Zeitreihe $h_0 - h_n$ von Schicht zu Schicht übertragen werden (vgl. Abbildung 2-13). In der letzten Schicht liegt dann ebenfalls wieder eine Zeitreihe vor (return_state=False, return_sequences=True). Dieser Aufbau ist sinnvoll, wenn mehrere LSTM-Schichten hintereinandergeschaltet und dabei z.B. ganze Zeitreihen am Ausgang abgebildet werden sollen.

Zuletzt ist auch die Weitergabe von Zeitreihen in Kombination mit der Initialisierung über Zellstatus und Netzausgang implementierbar (return_sequences=True, return_state=True). Dieser Aufbau ist bei vielschichtigen LSTM-Zellen nützlich, wenn z.B. weitere Schichten vor den Netzaufbau aus Abbildung 2-12 geschaltet, Zeitreihen weitergegeben und dabei ein Langzeitgedächtnis am Ausgang gebildet werden sollen.

Der Zellstatus und -ausgang können manuell zurückgesetzt oder initialisiert werden ([30], S.2), wenn beim Training z.B. längere Sprünge auf der Zeitachse auftreten oder Datensätze voneinander unabhängige Zeitabschnitte besitzen.

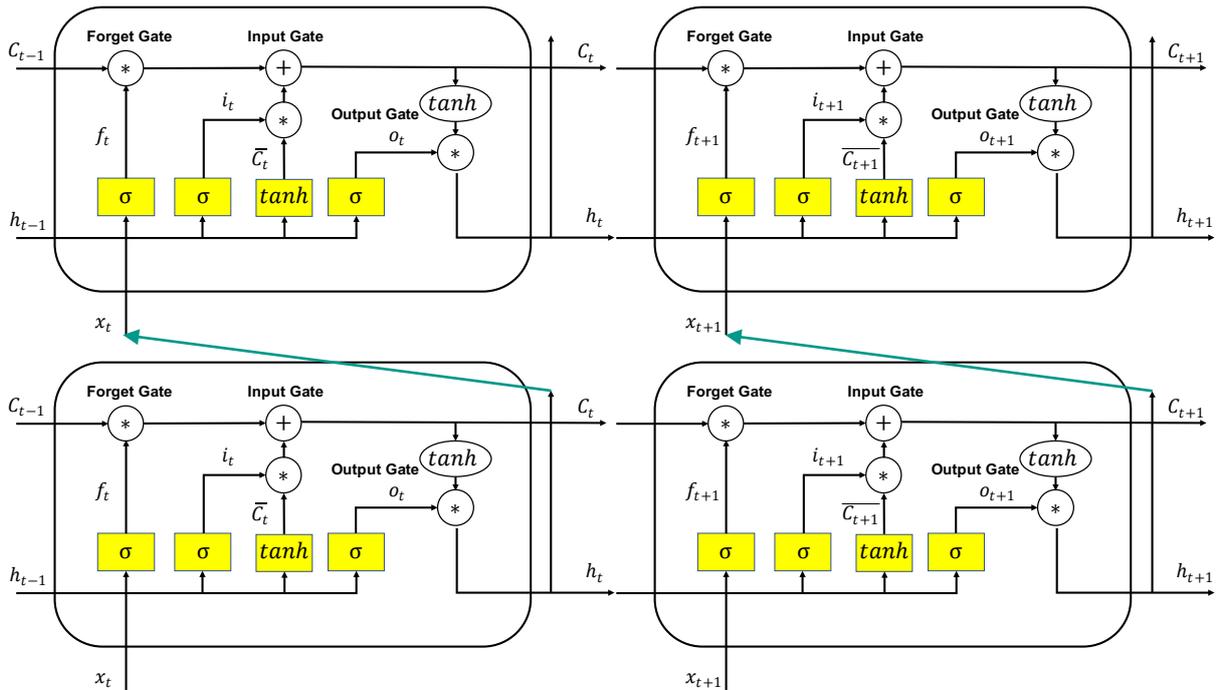


Abbildung 2-13: Schichtung von LSTM-Zellen II (return_state=False, return_sequences=True)

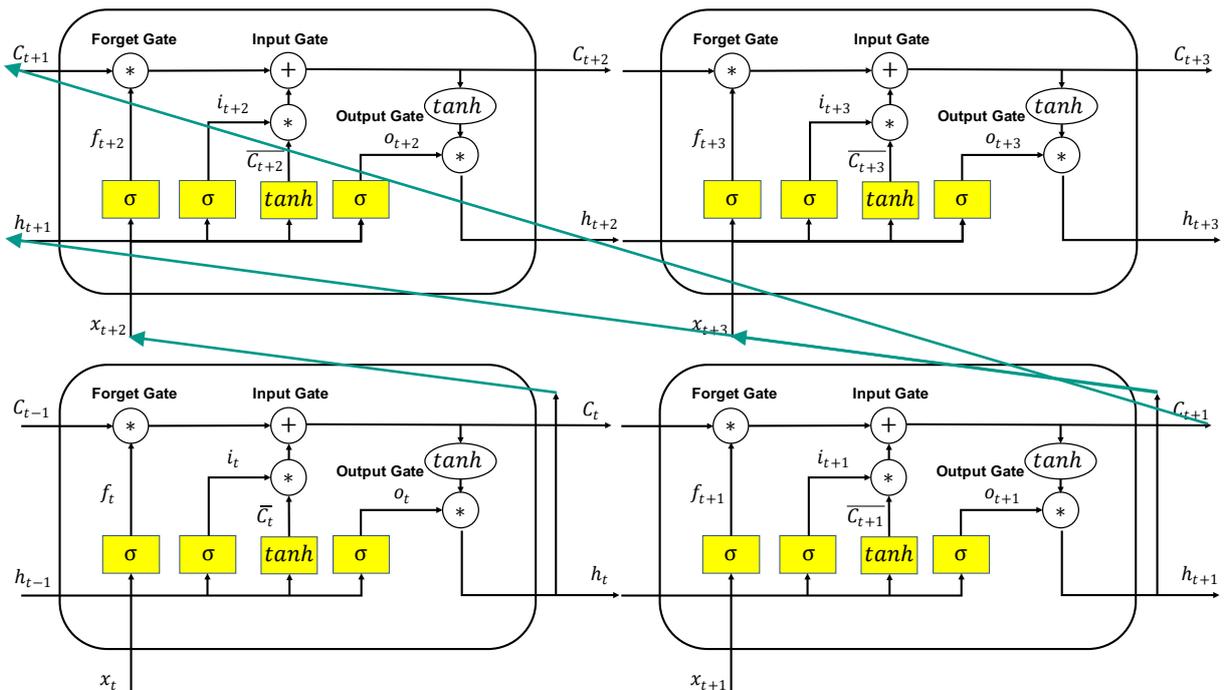


Abbildung 2-14: Schichtung von LSTM-Zellen III (return_state=True, return_sequences=True)



2.2 Datenaufbereitung

„Daten sind das neue Gold“ ([29], Titel). In der Transformation von analogen Systemen hin zur Digitalisierung werden immer größere Datenmengen verarbeitet und gespeichert. Dabei entstehen durch eine verbesserte Datenanalyse zum einen Optimierungsmöglichkeiten im Entwicklungsprozess, zum anderen ergeben sich neue Geschäftsmodelle, wie z.B. das Bereitstellen digitaler Produktbegleiter (engl. Digital Twins).

Oftmals sind Rohdaten aber fehlerbehaftet oder müssen vor der Verwendung aufbereitet werden. Sensoren können z.B. verrauschte Signale liefern oder Steuergeräte falsche Umrechnungsfaktoren enthalten. Außerdem wirkt sich eine gleichmäßige Verteilung der Trainingsdaten positiv auf die Generalisierung von neuronalen Netzwerken aus ([64] S. 22-25). In diesem Kapitel wird daher beschrieben, mit welchen Methoden die in dieser Arbeit verwendeten Datensätze aufbereitet werden, bevor die eigentlichen Modellierungen bzw. die Fehlererkennung beginnen.

2.2.1 Interpolation

Ein häufig auftretendes Problem bei Messsignalen ist eine Überlastung der Sensorik (engl. Overload), welche bei spannungsverändernden Sensoren z.B. durch eine beschädigte Kabelisolation auftreten kann. Dabei wird der minimal bzw. maximal mögliche Wert erreicht und gehalten, bis die Überlastung vorbei ist.



Zur Veranschaulichung ist in Abbildung 2-15 der Temperaturverlauf, gemessen in Stuttgart Mitte, für den 02.07.2022 aufgetragen (Datensatz aus [38]). Für einige Werte ist exemplarisch eine Überlastung in einem 8-Bit Steuergerät aufgetragen, bei dem das erste Bit positiv oder negativ stellt und die restlichen Bits die gemessenen Temperaturwerte widerspiegeln. Dementsprechend resultiert die Überlastung in einem Temperaturwert von 128 °C.

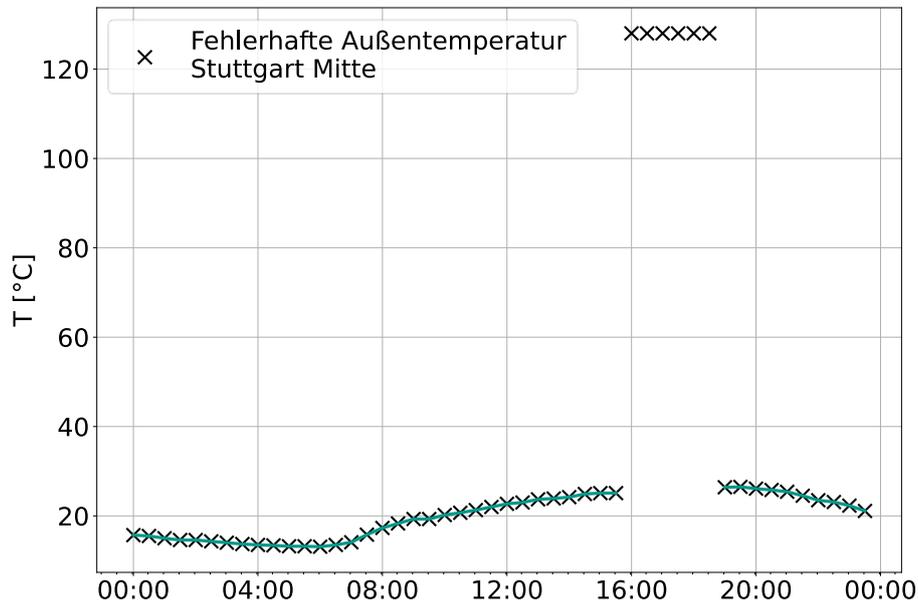


Abbildung 2-15: Temperaturverlauf mit Überlastungen [38]

Mit Hilfe von Minimum- bzw. Maximumschränken können fehlerhafte Punkte erkannt und entfernt werden. Wird die höchste zulässige Temperatur mit einem physikalischen Verständnis für realistisch auftretende Lufttemperaturen in Mitteleuropa vor dem Überschreiten des 2°C-Ziels der globalen Erderwärmung auf 60 °C festgelegt, werden alle Temperaturen darüber ignoriert und z.B. mit Hilfe einer linearen Interpolation berechnet. Die resultierende Funktion für die Berechnung der Zwischenpunkte berechnet sich nach Gleichung (2.2.1) für zwei gegebene Datenpunkte (x_0, y_0) und (x_1, y_1) folgendermaßen ([28] S.64).

$$f(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} * (x - x_0) = f_0 \frac{x_1 - x}{x_1 - x_0} + f_1 \frac{x - x_0}{x_1 - x_0} \quad (2.2.1)$$

Die korrigierte Außentemperatur ist dementsprechend in Abbildung 2-16 (rot) veranschaulicht. Ebenfalls können je nach Anwendungsfall quadratische (grün), kubische (magenta) oder polynomiale (braun, gelb) Interpolationen verwendet werden.



Mit Hilfe der Lagrange Formel kann die Lösung des Interpolationsproblems nach Gleichung (2.2.2) direkt angegeben werden ([57] S.39).

$$P(x) = \sum_{i=0}^n f_i \prod_{\substack{k=0 \\ k \neq i}}^n \frac{x - x_k}{x_i - x_k} \quad (2.2.2)$$

In der Praxis werden Polynome mit höherer Ordnung als fünf vermieden, da diese dazu neigen, instabil zu werden (vgl. Abbildung 2-16, gelb).

Als Vergleich sind die real gemessenen Außentemperaturen ohne Messfehler im Bereich der Interpolation zusätzlich aufgetragen. In diesem Beispiel werden die besten Ergebnisse mit einer linearen Interpolation erreicht, was daran liegt, dass mehrere Temperaturmessungen in Folge fehlgeschlagen sind und die Temperaturmessung dabei eine gewisse thermische Trägheit besitzt. Je nach Häufigkeit der Messfehler und Anwendungszweck kann es allerdings auch vorteilhaft sein, andere Interpolationen zu verwenden.

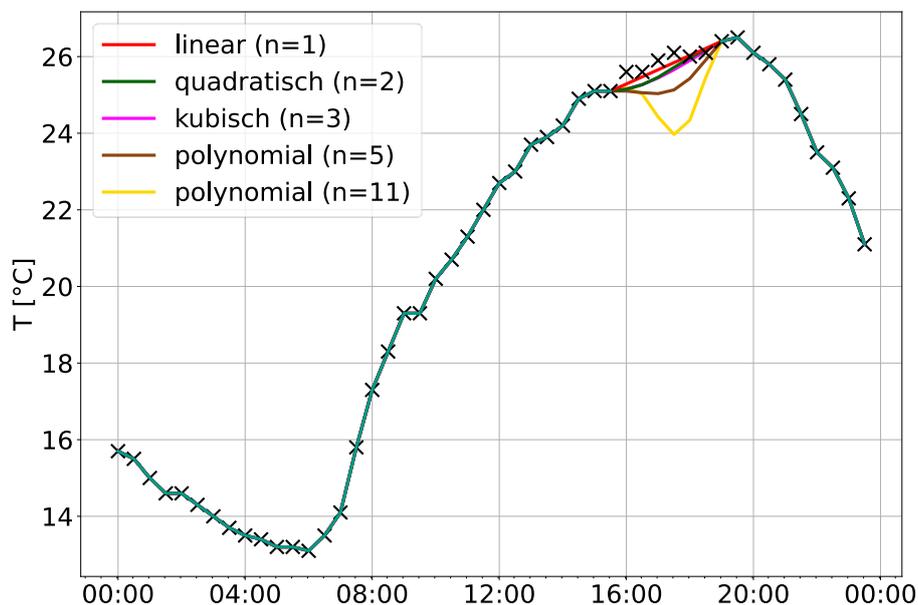


Abbildung 2-16: Korrigierter Temperaturverlauf ohne Überlastungen

Ein Nachteil dieser Methode ist die Voraussetzung einer Kenntnis über möglich realistisch vorkommende Wertebereiche des Gesamtsystems. In der Praxis bedeutet das für komplexere Zusammenhänge, wie z.B. Drücke oder Beschleunigungen, ein technisches Verständnis für jede der einzelnen Messgrößen.



Mit Hilfe der standardisierten Zufallsvariable z (engl. z-Score) ([34] S. 736) lässt sich dieses Problem umgehen. Diese wird mit Hilfe des Mittelwerts μ und der Standardabweichung σ nach Gleichung (2.2.3) folgendermaßen berechnet. z kann dabei auch als Entfernung zu einem standardisierten Mittelwert interpretiert werden.

$$z = \frac{x - \mu}{\sigma} \quad (2.2.3)$$

In Abbildung 2-17 ist die standardisierte Zufallsvariable zusätzlich zum fehlerbehafteten Temperaturverlauf aus Abbildung 2-15 aufgetragen. Es lässt sich erkennen, dass die z-Score bei den Ausreißern, bedingt durch die Überlastung, einen erhöhten Wert aufweist. Mit der Definition einer maximal zulässigen standardisierten Zufallsvariable lässt sich für jedes beliebige Signal, unabhängig von seinem Wertebereich, eine entsprechende Korrektur nach Gleichung (2.2.1) oder (2.2.2) vornehmen.

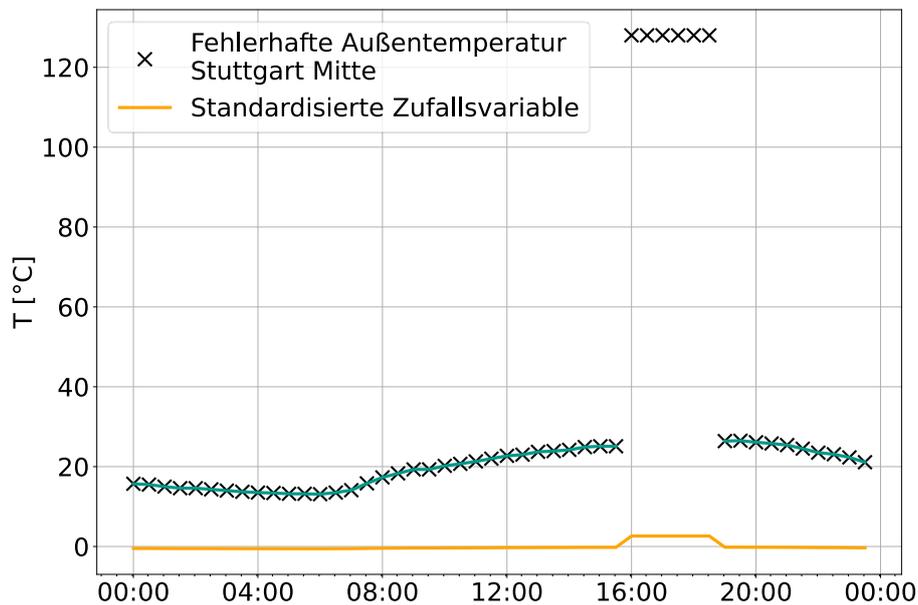


Abbildung 2-17: z-Score für einen Temperaturverlauf mit Überlastung

2.2.2 Veränderung der Abtastrate

Analoge Signale bestehen grundsätzlich aus einem kontinuierlichen Zeit- und Wertebereich. Bei digitalen Messungen werden Signalwerte zu bestimmten Zeitpunkten aufgezeichnet und gespeichert. Das bedeutet, die Signalverläufe werden diskretisiert, und es entstehen zeitaufgelöste Sequenzen.



Bei der Speicherung der Signalamplitude in binären Werten bzw. bei einer Quantisierung der Amplitudenachse, können durch den diskretisierten Wertebereich Messungenauigkeiten entstehen, welche auch als „Messrauschen“ bezeichnet werden ([44] S. 153). Bei einer Verwendung von A/D Wandlern mit ausreichend großem Wertebereich, z.B. entsprechen 24 Bit über 16 Millionen möglichen Messwerten, kann dieses Phänomen allerdings vernachlässigt werden.

Bei der idealen Abtastung ([44] S. 153), wird ein analoges Signal $x(t)$ mit einer Diracstoßfolge multipliziert und es entsteht dabei das zeitdiskrete und wertekontinuierliche Signal $x_A(t)$ entsprechend Gleichung (2.2.4).

$$x_A(t) = x(t) * \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x(t) * \delta(t - nT) \quad (2.2.4)$$

Aufgrund der Ausblendeigenschaft des Diracstoßes ([44], S.153) kann diese Gleichung folgendermaßen zu Gleichung (2.2.5) umgeschrieben werden.

$$x_A(t) = \sum_{n=-\infty}^{\infty} x(t) * \delta(t - nT) \triangleq x(nT) = x[n] \quad (2.2.5)$$

T ist dabei das Abtastintervall, für das $f_A = \frac{1}{T}$ gilt, und f_A dabei die Abtastfrequenz ist. Der Term $x[n]$ ist eine gewichtete Folge von Diracstößen und stellt die Abtastwerte dar, welche genau den Signalwerten $x(t)$ an den Stellen $t = nT$ entsprechen.

Mit Hilfe einer Fourier-Transformation kann der Frequenzgang des zeitdiskreten Signals nach Gleichung (2.2.6) analysiert werden.

$$X_A(j\omega) = \sum_{n=-\infty}^{\infty} x[n] * e^{-jn\omega T} = X_A\left(j\omega + jk \frac{2\pi}{T}\right) \quad (2.2.6)$$

Da der ideale Diracstoß in der Realität nicht existiert ([44], S.154), können x_A und $X_A(j\omega)$ nur angenähert werden. Grundsätzlich lässt sich ein zeitdiskretes Signal auch wieder in ein kontinuierliches (analoges) Signal überführen, wenn dabei das Abtasttheorem nach Gleichung (2.1.7) eingehalten wird ([44], S.158), welches besagt, dass ein Signal mit mehr als doppelter Frequenz abgetastet werden muss wie die höchste vorkommende Signalfrequenz.



Analoge Signale mit geringer und mittlerer Bandbreite werden daher häufig deutlich höher als nötig abgetastet (engl. Oversampling).

$$f_A > 2B; \quad B = \text{Bandbreite} \quad (2.2.7)$$

Beim Umgang und in der Verarbeitung von zeitdiskreten Daten für transiente Modellierungen ergeben sich nach den vorangegangenen Überlegungen folgende Möglichkeiten und Einschränkungen. Zum einen kann die Abtastrate eines Signals im Nachhinein verringert werden (engl. Downsampling), um den notwendigen Speicherbedarf von Messdaten zu verringern. Ist die neue Abtastrate ein ganzzahliges Vielfaches der alten, entsteht kein zusätzlicher Rechenaufwand, da nur jeder n-te Zeitschritt gespeichert wird. Falls nicht, muss zwischen den neuen Stützstellen entsprechend Kapitel 2.2.1 interpoliert werden.

Für eine künstliche Erhöhung der Abtastfrequenz kann ein digitaler Interpolator verwendet werden, wie z.B. in CD-Spielern verbaut, welcher ebenfalls zwischen den neuen und zusätzlich hinzugefügten Zeitschritten interpoliert ([44], S.164). Ein nachträgliches Erhöhen der Abtastrate (engl. Upsampling) wäre in der Theorie über eine Signalrekonstruktion mit anschließendem Abtasten an einer analogen Schaltung denkbar, wird aber aufgrund von Genauigkeitsverlusten bei physikalischen Abtastungen, im Gegensatz zu Idealen, in dieser Arbeit nicht untersucht.

2.2.3 Erstellen konsistenter Zeitreihen

Werden Teile eines diskreten Zeitsignals mit einheitlicher Fensterlänge ausgeschnitten, kann es passieren, dass die extrahierten Zeitreihen z.B. am Ende des Datensatzes, bei Stillstand oder Signalausfällen, nicht konsistent sind. Das bedeutet konkret, innerhalb eines Zeitfensters sind Sprünge in der Zeitdimension enthalten.

Für eine transiente Modellierung mit der Hilfe von neuronalen Netzwerken stellt dies Schwierigkeiten dar ([7], S.329), weil das physikalische Verhalten an diesen Stellen nicht dem der konsistenten Zeitreihe entspricht.

Daher werden die ausgeschnittenen Zeitfenster in dieser Arbeit so ausgewählt, dass für eine Fensterlänge Z und eine Schrittweite T nach Gleichung (2.1.8) stets folgender Zusammenhang gilt.

$$Z = \sum_{T_0}^{T_n} x_{t+T} \quad (2.2.8)$$



2.2.4 Leistungsklassifikation

Gemessene Fahrprofile im Realbetrieb eines Antriebsstrangs bestehen, anders als festgelegte Messkampagnen am Prüfstand, aus vielen verschiedenen Betriebszuständen. Die erforderlichen Daten für eine Modellierung müssen daraus mit Hilfe von geeigneten Methoden ausgeschnitten werden. Dafür werden konsistente Zeitfenster zuerst erstellt und anschließend klassifiziert, sodass diese verschiedenen Klassen eines entsprechenden Leistungsintervalls zugeordnet werden können.

Für die Aufteilung eines Leistungsintervalls zwischen L_{min} und L_{max} durch eine Schrittweite l ergeben sich nach Gleichung (2.2.9) k unterschiedliche Leistungen.

$$L_k = \sum_0^k L_{min} + k * l, \quad k \in [0, \frac{L_{max}}{l}] \quad (2.2.9)$$

Jede dieser Leistungen kann als Start- oder Endleistung auftreten. Daher werden alle möglichen Kombinationen mit der Dimension $r = 2$ gebildet und die Anzahl an möglich auftretenden Betriebszuständen ergibt sich nach Gleichung (2.2.10) folgendermaßen.

$$K = \frac{(k + r - 1)!}{r! + (k - 1)!}, \quad k > 0 \quad (2.2.10)$$

Beispielsweise ergeben sich bei einer Einteilung in 5% Schritten zwischen 0 und 100% Leistung 21 möglich auftretende Leistungen $[0, 5, 10, \dots, 90, 95, 100]$, welche sich in 231 unterschiedliche Betriebszustände kombinieren lassen $[(0, 0), (0, 5), (0, 10), \dots, (0, 100), (5, 5), (5, 10), \dots, (100, 100), (100, 95), \dots, (10, 5), (10, 0), (5, 0)]$.

Werden nun die Start- und Endleistung L_{start} und L_{Ende} und die Steigung eines jeden Zeitfensters betrachtet, lassen sich die Zeitfenster immer dann eindeutig klassifizieren, wenn L_{start} und L_{Ende} einer Betriebszustandsklasse zuzuordnen sind und sich das Vorzeichen der Steigung im betrachteten Zeitfenster nicht ändert.

Für diese Arbeit werden Schwellenwerte definiert, welche eine gewisse Abweichung von L_{start} und L_{Ende} , sowie eine kurzzeitige Änderung des Vorzeichens bei der Steigung zulassen. Abbildung 2-18 zeigt dafür exemplarisch ein klassifiziertes einstündiges Zeitfenster der Leistungsklasse (20, 60). L_{start} und L_{Ende} liegen dort bei 20.795 % und 62.298 %. Die Steigung wechselt zum Zeitpunkt 23:03:00 kurzzeitig das Vorzeichen, trotzdem wird dieses Zeitfenster in die entsprechende Leistungsklasse (20, 60) eingeteilt.

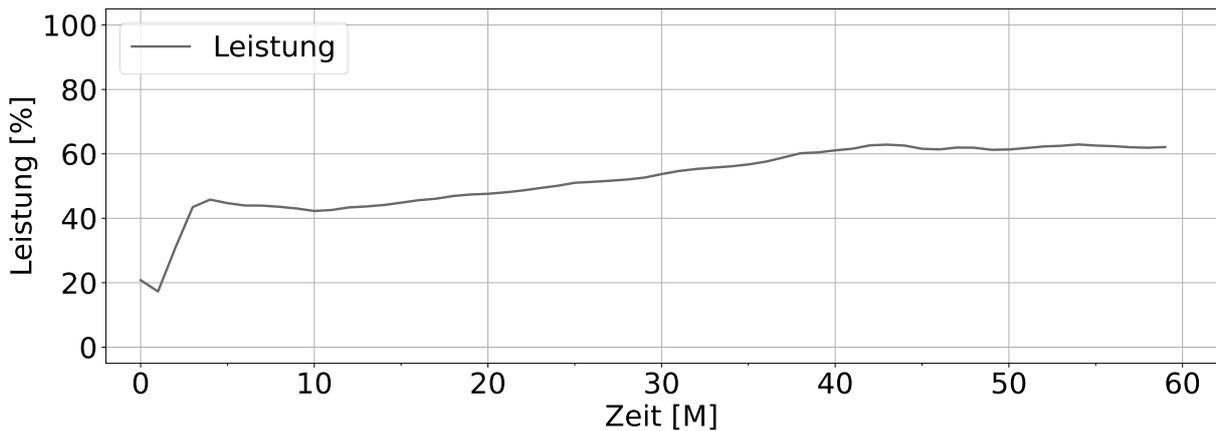


Abbildung 2-18: Klassifiziertes einstündiges Zeitfenster der Leistungsklasse (20, 60)

2.2.5 Datenaugmentation

In Datensätzen kann es vorkommen, dass nicht ausreichend Datenpunkte eines bestimmten Zustands für die Modellierungen mit Hilfe von neuronalen Netzwerken vorhanden sind. Dies stellt ein Problem dar, da in ungewichteten Datensätzen unterrepräsentierte Datenpunkte weniger beachtet werden als überrepräsentierte ([47], S.322). Beispielsweise kann eine Modellierung 99% Genauigkeit erreichen, ohne auch nur eine richtige Vorhersage für eine seltenen Klasse mit einer Verteilung unter einem Prozent getroffen zu haben. Gerade bei der Erkennung von Fehlern, die im Vergleich zum Normalzustand eher selten auftreten, können durch ungewichtete Datensätze falsche Modellgültigkeiten entstehen.

Eine Möglichkeit der Abhilfe besteht darin, die überrepräsentierten Datenpunkte zu verringern, was allerdings in einer Nichtnutzung wertvoller Messungen resultiert. Eine andere Möglichkeit ist die Vermehrung der unterrepräsentierten Datenpunkte. Für Klassifizierungsaufgaben wird in [47] die SMOTE- (engl. Synthetic Minority Over-sampling Technique) Methode vorgestellt. Diese beruht auf der künstlichen Vermehrung von unterrepräsentierten Klassen durch leichte Veränderungen ([47], S.328). Die Veränderungen werden dabei durch Interpolationen zwischen Datensätzen erreicht, die von einem K-Nearest Neighbor Modell [67] in ähnliche Klassen eingeteilt sind. In [37] wird diese Methodik auch für Regressionsprobleme, wie in dieser Arbeit ausschließlich untersucht, erfolgreich weiterentwickelt.

Bei der Augmentation von Zeitfenstern, welche bereits aufgrund ihrer vorkommenden Leistung klassifiziert sind (vgl. Kapitel 2.2.4), erübrigt sich die Einteilung mit Hilfe eines K-Nearest Neighbor Modells. Für Klassen, in denen es mehr als ein Zeitfenster gibt, können direkt durch die Interpolation zwischen mehreren Zeitfenstern neue Sequenzen generiert werden.



Dafür sind in Abbildung 2-19 exemplarisch drei gemessene und einständige Zeitfenster einer Leistungsklasse zwischen 80-40 % Leistung aufgetragen. Die vierte Sequenz (grün) stellt die Interpolation zwischen den gemessenen Sequenzen (grau) zu jedem Zeitschritt dar.

Über eine Variation der Anzahl an zufällig ausgewählten Zeitfenstern für die Interpolation können mit nur einigen wenigen Sequenzen mehrere neue synthetisch erzeugte Zeitfenster erstellt werden. Ebenfalls gilt dieses Verfahren für beliebig viele Messgrößen innerhalb der verwendeten Sequenzen, wobei die Ähnlichkeit mit zunehmender Dimension geringer wird, da die euklidischen Distanzen der Klassen potentiell größer werden können ([67], S.23).

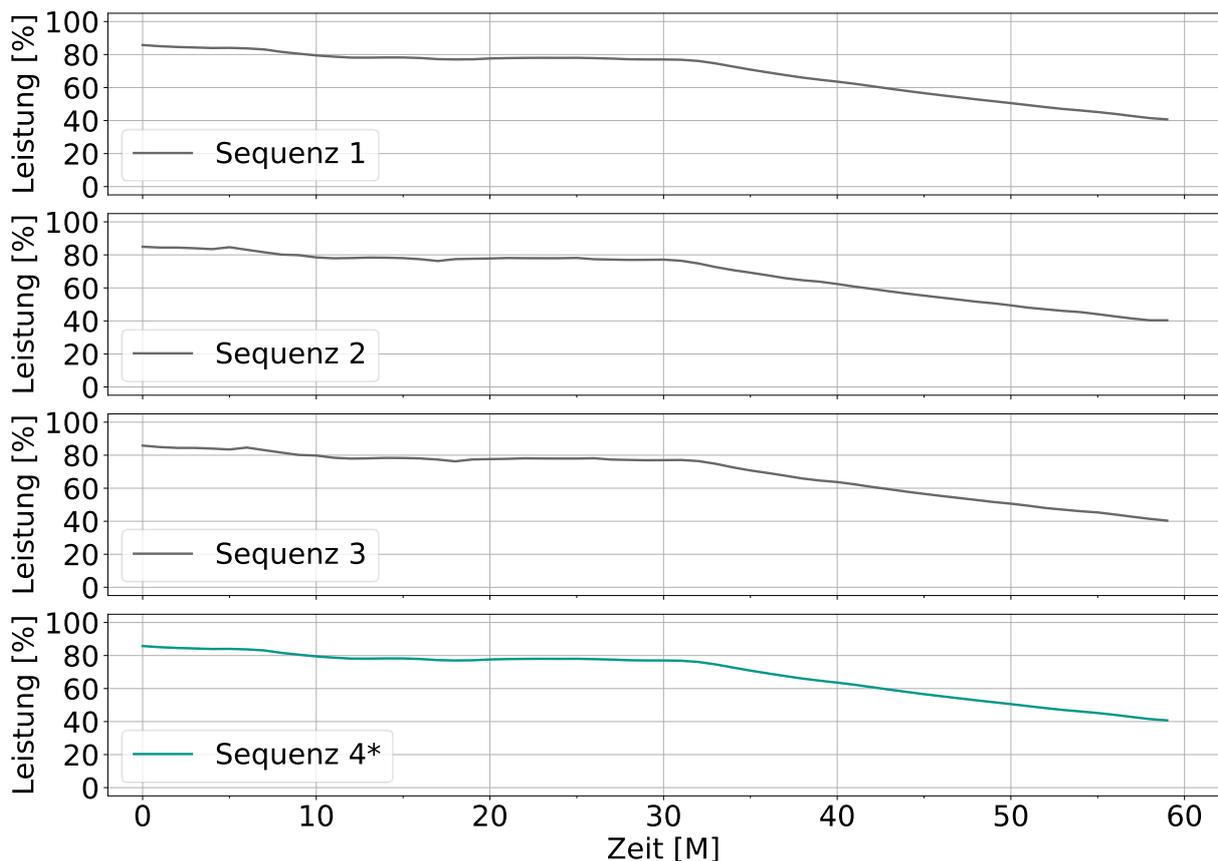


Abbildung 2-19: Klassifizierte einständige Zeitfenster der Klasse (80, 40) mit interpolierter Sequenz 4*

Bei Klassen, in denen es nur eine einzige Sequenz gibt, stößt dieses Verfahren an seine Grenzen, da keine Interpolation zwischen mehreren Zeitfenster möglich ist. Hier wird das Ausgangssignal mit einem leichten, niederfrequenten und zufällig initialisiertem Rauschen überlagert, um eine Augmentierung zu erreichen. Über Frequenz und Amplitude, die mit einer Standardabweichung um einen Mittelwert schwingen, lässt sich bei der Überlagerung der Abweichungsgrad zum Ausgangsfenster regulieren.



In Abbildung 2-20 ist dafür exemplarisch ein einstündiges Zeitfenster der Leistungsklasse 100-90 % in Sequenz 1 aufgetragen. Sequenz 2*, 3* und 4* stellen Zeitfenster dar, welche jeweils mit einem anderen Signal überlagert sind, um eine gewisse Varianz gegenüber dem Ausgangssignal zu erzeugen.

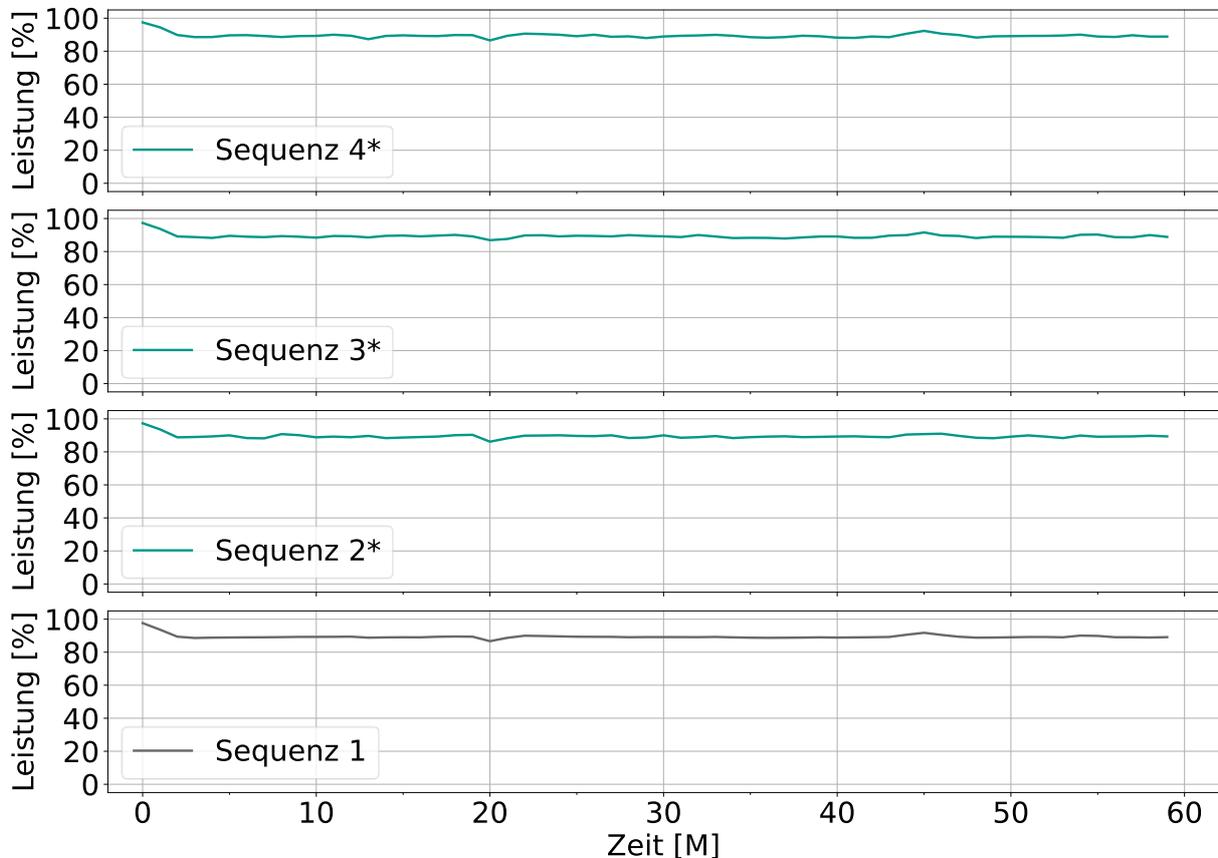


Abbildung 2-20: Klassifiziertes einstündiges Zeitfenster der Klasse (100, 90) mit überlagerten Sequenzen 2*, 3*, 4*

2.3 Zylindermodul

Das Zylindermodul ist ein 0D- bzw. 1D-Simulationsprogramm, mit dessen Hilfe die Vorgänge im Zylinder eines Verbrennungsmotors während des Hochdruckteils der Verdichtung berechnet werden können. Es ist Teil des Forschungsvorhabens Nr. 869 [41] der Forschungsvereinigung für Verbrennungsmotoren (FVV) und wird aufgrund seines universellen und modularen Aufbaus bis heute weiterentwickelt [42], [13], [6], [46], [9], [1], [59], [60]. Sowohl für unterschiedliche Brennverfahren als auch Kraftstoffarten können dabei zum einen gemittelte Einzelgrößen der Verbrennung, wie z.B. der indizierte Mitteldruck, und zum anderen kurbelwinkelaufgelöste Verlaufsgrößen, wie z.B. der Brennverlauf, berechnet werden.



2.4 GT-Suite

GT-Suite ist eine kommerzielle Simulationssoftware mit graphischer Oberfläche, in der mechanische und thermodynamische Systeme mit Hilfe von Bausteinen zusammengestellt, ausgelegt und berechnet werden können. Es lassen sich sowohl 1D-, als auch 3D-Berechnungen durchführen ([5], S.12), wobei sich die 1D-Berechnungen auf das Gesamtsystem beziehen und nur Vorgänge in einzelnen Komponenten mit Hilfe von 3D-Betrachtungen untersucht werden können.



3 Stand der Technik

Die Idee von der Verwendung lernfähiger Algorithmen für die Abbildung von physikalischen Zusammenhängen ist alt, wie bereits in Kapitel 1 und 2.1 dargelegt. Beispielsweise können neuronale Netzwerke darauf trainiert werden, Parameter von Differentialgleichungen abzuschätzen [51]. In [56] wird bereits im Jahr 1998 untersucht, Teilaufgaben des Motormanagements, wie z.B. die Verschiebung des Zündzeitpunkts zur Regelung der Schwerepunktage, durch Feed-Forward Netze zu ersetzen. Da dort allerdings Prüfstandsdaten zum Einsatz kommen, ist die Datengrundlage und somit die Generalisierung der Netze vergleichsweise eher gering.

Neu am Institut für Fahrzeugtechnik in Stuttgart ist die Kombination von konventioneller 0D-1D Simulationssoftware für die Antriebsstrangentwicklung (vgl. Kapitel 2.3 und 2.4) mit Algorithmen des maschinellen Lernens. Dies hat zum einen den Vorteil, Simulationen mit geringen Genauigkeitsverlusten stark zu beschleunigen ([15], S.1). Zum anderen können mit Hilfe von virtuellen Versuchsträgern breite DOEs (engl. Design of Experiments) durchgeführt und somit effektiv sehr große Datenmengen erzeugt werden.

Im Forschungsgebiet der Nutzung von neuronalen Netzen für die Antriebsstrangentwicklung ist die Zahl der Veröffentlichung dabei noch eher überschaubar ([63], [2], [49], [4]), wobei die Beiträge dazu stetig steigen. Bei stationären Betriebspunkten wird in ([15], S.16) gezeigt, dass neuronale Netze vielversprechende Ergebnisse bei der Modellierung von gemittelten Einzelgrößen der Verbrennung über ein Arbeitsspiel während des Hochdruckteils der Verdichtungen erzielen. Beispielsweise wird dort für einen Datensatz mit über einer Millionen Betriebspunkten eine Abweichung des vorhergesagten indizierten Mitteldrucks mit Hilfe von neuronalen Netzwerken von weniger als 1.87 % im Vergleich zur Simulation erzielt. Weiterhin konnte in ([16], S. 8) gezeigt werden, dass eine sinnvolle Verschaltung von Teilnetzen eine wesentliche Verbesserung der Vorhersagequalität bei komplexen Brennraumgrößen, wie z.B. der Lage des Spitzendrucks, zur Folge hat. Darauf aufbauend wird in dieser Arbeit die Modellierung von zyklischen und kurbelwinkelaufgelösten Brennraumgrößen untersucht sowie sich mit der Überführung von transienten Gesamtsystemen in neuronale Netzstrukturen auseinandergesetzt.



Bei der Detektion von Unregelmäßigkeiten in gemessenen Signalen mechatronischer Gesamtsystemen hat die Firma Airbus mit ihrem ISS-Projekt für die Überwachung funktionsrelevanter Messgrößen in [8] größere Aufmerksamkeit in der Gemeinschaft der Datenwissenschaftler erregt. Die Grundlagen zur Erkennung von Anomalien mit Hilfe von Auto-Encodern und LSTM-Zellen sind allerdings älter (vgl. [53], [52]).

Dort ist bereits beschrieben, wie mit Hilfe einer transientfähigen Encoder-Decoder Architektur und der Berechnung einer Fehlerkenngröße Unregelmäßigkeiten in verschiedenen gemessenen Signalen entdeckt werden.

Darauf aufbauend wird der Einsatz von Auto-Encodern mit Hilfe von LSTM-Zellen für komplexe höherdimensionale Sensorsignale verschiedener Subsysteme im Antriebsstrang untersucht. Außerdem wird die vorgestellte Methodik auf unterschiedliche Bauteilverbände angewandt und ausgewertet, sowie exemplarisch einige Unregelmäßigkeiten im Antriebsstrang mit Hilfe der Modellierungen untersucht.



4 Modellierung zyklischer Vorgänge

Bei der Beschreibung von Vorgängen im Brennraum einer Hubkolbenmaschine sind die physikalischen Prozesse abhängig vom Kurbelwinkel und wiederholen sich nach jedem Arbeitsspiel. Thermodynamisch handelt es sich dabei um einen rechtslaufenden Kreisprozess, bei welchem im Betrieb mit 4-Takten Frischgas zuerst angesaugt und anschließend verdichtet wird. Als nächstes wird der verwendete Kraftstoff eingebracht und abhängig vom Brennverfahren entzündet, sprich Energie freigesetzt und somit gearbeitet. Zuletzt muss das verbrannte Abgas ausgestoßen werden, bevor anschließend das neue Arbeitsspiel beginnt.

Da das mechanische und thermodynamische Grundprinzip bei jedem Arbeitsspiel gleich ist, kann auch von zyklischen Vorgängen gesprochen werden, bei denen sich die Randbedingungen für die Modellierung zwar ändern, aber diese in der Regel das Verhalten eines ganzen Arbeitsspiels beschreiben. Das bedeutet, es sind keine Informationen von vorangegangenen Arbeitsspielen notwendig, um den aktuellen Zyklus zu beschreiben, da diese in sich abgeschlossen sind.

4.1 Datengenerierung

In diesem Kapitel werden wichtige Größen für die Brennverfahrensentwicklung eines Wasserstoffmotors mit Hilfe von neuronalen Netzwerken modelliert. Ausgang für die Erstellung der neuronalen Netze sind kurbelwinkel aufgelöste Simulationen des Druck-, Druckgradienten-, Brenn- und Wandwärmeverlaufs während dem Hochdruckteil der Verdichtung von einem kalibrierten Forschungsmodell eines MTU BR-2000 Nutzfahrzeugmotors, welcher auf Wasserstoffverbrennung mit fremdgezündetem und vorgemischtem (Otto-)Brennverfahren umgerüstet ist. In Tabelle 4-1 sind die geometrischen Randbedingungen entsprechend für einen Zylinder gelistet. Da im Originalmotor kein variabler Ventiltrieb verbaut ist, sind die Steuerzeiten festgelegt.



Geometrische Größe	Wert	Einheit
Hub	150	mm
Bohrung	130	mm
Desachsierung	0.6	mm
Pleullänge	273	mm
Verdichtungsverhältnis	11.75	-
Einlass schließt	-163	°ZOT
Auslass öffnet	145	°ZOT

Tabelle 4-1: Geometrische Randbedingungen des H₂-Motormodells

Ausgehend von den konstruktiv bedingten Geometrien im Brennraum, werden für die Verbrennung ausschlaggebende Parameter variiert und mit Hilfe des Zylindermoduls berechnet. Vorangegangene Untersuchungen in ([64] S. 22-25), ([40] S.18) und ([65] S. 3-8) haben ergeben, dass sich eine zufällig Rasterung im Kennfeldbereich über Verteilungsfunktionen gegenüber einer vorgegebenen schrittweisen Rasterung, wie z.B. bei Messungen am Prüfstand üblicherweise verwendet, vorteilhaft auf die Generalisierung bei der Modellierung mit Hilfe von neuronalen Netzwerken auswirkt. Daher werden zwei Millionen Simulationen unter den Vorgaben von Tabelle 4-2 gerechnet.



Variable	Minimum	Maximum	Einheit	Verteilung
Drehzahl	400	2900	1/min	uniform
Luftmasse	100	5000	mg	uniform
Lambda	1	4	-	uniform
Rate AGR	0	20	%	uniform
Starttemperatur	250	500	K	uniform
Zündzeitpunkt	-60	40	°KW (ZOT)	normal
Temperatur Kopf	250	900	Kelvin	uniform
Temperatur Kolbenbuchse	250	800	Kelvin	uniform
Temperatur Kolben	250	800	Kelvin	uniform

Tabelle 4-2: Variablen der Kennfeldraasterung des H₂-Motormodells

Grundsätzlich liegen die Grenzen für alle Parameter etwas unter bzw. über den für Nutzfahrzeugmotoren üblichen Motorwerten. Das hat den Grund, dass für eine spätere Implementierung der Netze in andere Entwicklungsumgebungen ([15], S.14) möglichst viele verschiedene Brennverfahren und Randbedingungen abgedeckt sein sollen. Außerdem können dadurch Extrapolationen der Netze vermieden werden, da diese grundsätzlich zu einem unerwarteten Vorhersageverhalten führen können ([35], S.1).

4.1.1 Drehzahl

Die Drehzahl wird nach unten durch eine sinkende Brennrate und dadurch bedingt stärkere Zyklenschwankungen ([46], S. 7) begrenzt. Eine Drehzahl unter 400 1/min erscheint daher unrealistisch. Bei vorgemischter und fremdgezündeter Verbrennung, anders als bei der diffusiven Dieserverbrennungen, bei der die Brennrate durch die Vermischungsgeschwindigkeit begrenzt ist ([14], S. 3), können in homogenen Gemischen durch zunehmende Turbulenz in der Flammenfront höhere Brenngeschwindigkeiten, bzw. Drehzahlen erreicht werden ([33], S.612). Das bedeutet, es sind theoretisch höhere Drehzahlen wie bei vergleichbaren NFZ-Dieselmotoren denkbar. Allerdings entstehen bei langhubigen Großmotoren mit steigender Kolbenbeschleunigung hohe Trägheitskräfte, weshalb die obere Drehzahl auf 2900 1/min festgelegt wird.



4.1.2 Luftmasse

Durch eine Veränderung der Luftmasse im Zylinder kann bei ansonsten gleichen Randbedingungen die Last und respektive der indizierte Mitteldruck geregelt werden. Bei Wasserstoffmotoren lässt sich die Last allerdings auch gut qualitativ über das Luft-Kraftstoffverhältnis regeln, wie im nächsten Abschnitt erläutert wird. Nach unten wird die Luftmasse auf 100 mg begrenzt, um die Stabilität der numerischen Berechnungen zu gewährleisten, da bei zu kleinen Luftmassen, entsprechend der allgemeinen Gasgleichung sehr niedrige Drücke oder hohe Temperaturen im Unverbrannten entstehen können. Seit den 90er Jahren sind im Prinzip alle Nutzfahrzeugdieselmotoren aufgrund der einhergehenden Wirkungsgradsteigerung aufgeladen ([43], S.31). Das bedeutet, die im Zylinder enthaltene Luftmasse übersteigt das eigentliche Hubvolumen. Deshalb wird die maximale Luftmasse in der Simulation auf 5000 mg festgelegt, was bei gleichbleibender Temperatur einem Ladedruck von bis zu 3 bar entspricht. Abbildung 4-1 zeigt die Verteilung des indizierten Mitteldrucks in den Simulationsdaten. Bei hohen Luftmassen in Kombination mit einem stöchiometrischen Luft-Kraftstoffverhältnis können sehr hohe indizierte Mitteldrücke jenseits von 30 bar auftreten. Ebenfalls sind einige geschleppte Betriebspunkte, erkennbar am negativen indizierten Mitteldruck, enthalten, bei denen die Randbedingungen keine Verbrennung zustande bringen. Das ist bei der Modellierung aber durchaus vorteilhaft, um später einen möglichst großen Abbildungsbereich zu erzielen und einerseits Spielraum für Fortschritte in der Konstruktionstechnik zu lassen, sowie andererseits auch einen kleinen Teil des Schleppbetriebs mitabzubilden. Die größte Verteilung befindet sich im unteren Teillastbetrieb, was vorteilhaft ist, da die Brennaummodellierung dort erfahrungsgemäß herausfordernder als in der Volllast ist.

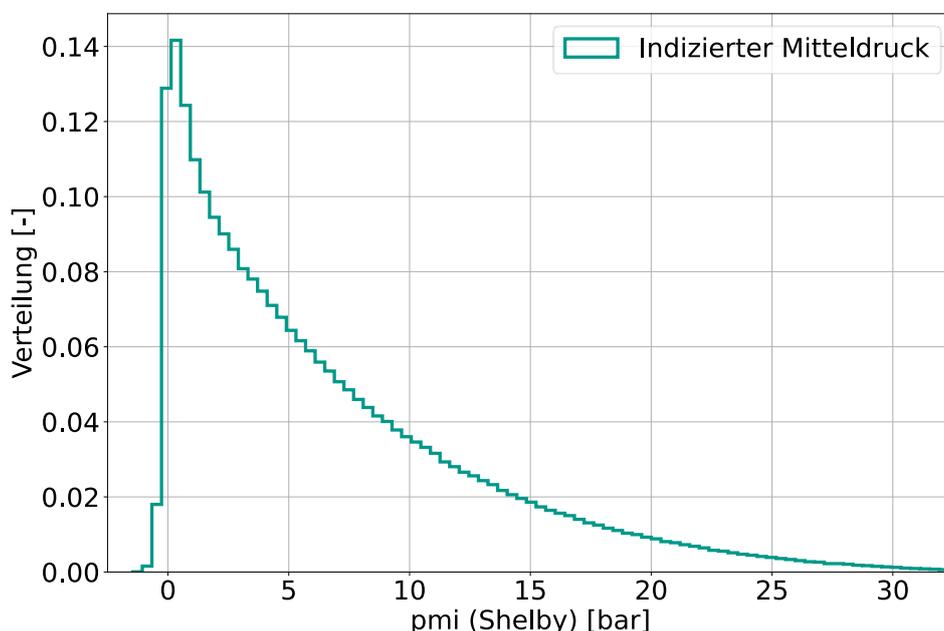


Abbildung 4-1: Verteilung des indizierten Mitteldrucks in der Simulation



4.1.3 Lambda

Wasserstoffmotoren werden aufgrund der hohen Reaktionsfreudigkeit von Wasserstoff mit Sauerstoff in der Regel nicht im fetten Bereich betrieben, da es sonst zu ungewünschten Nachreaktionen im Abgas kommen kann. Daher wird die untere Grenze für das Luft-Kraftstoffverhältnis auf $\lambda=1$ festgelegt. In Abbildung 4-2 sind die laminaren Flammgeschwindigkeiten für verschiedene Kraftstoffe ([25], S.8) in Abhängigkeit des Lambdas abgebildet. Im stöchiometrischen Bereich kann es aufgrund der fast achtmal höheren Flammgeschwindigkeit zu sehr hohen Brennraten kommen, was im Zusammenspiel mit großen Luftmassen zu extremen Druckverhältnissen im Zylinder führen kann. Deshalb findet die Wasserstoffverbrennung in der Regel in mageren Bereichen statt und kann auch vergleichsweise stärker abgemagert werden, da erst mit deutlich größerem Luftüberschuss eine ähnlich niedrige laminare Flammgeschwindigkeit, wie z.B. im Vergleich mit Benzin, erreicht wird. Um im Teillastbereich möglichst viele qualitative Regelstrategien abbilden zu können, wird die obere Lambda-Grenze in der Simulation daher auf vier festgelegt.

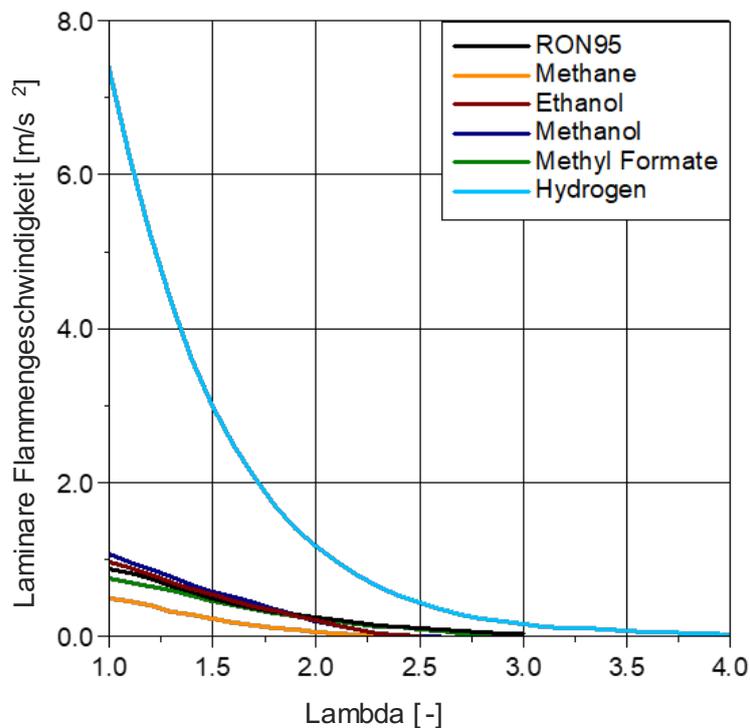


Abbildung 4-2: Laminare Flammgeschwindigkeit für verschiedene Kraftstoffe nach [25]



4.1.4 AGR-Rate

Da der in der Luft enthaltene Stickstoff bei magerer Verbrennung und hohen Temperaturen zu Stickoxiden reagieren kann ([33], S.620), wird auch für Wasserstoffmotoren eine Abgasrückführung untersucht, um die Prozesstemperatur zu senken. In Anlehnung an ([68], S. 33) wird die Rate der Abgasrückführung auf 0-20 % festgelegt.

4.1.5 Starttemperatur

Die Starttemperatur in der Arbeitsprozessrechnung bezieht sich auf die Temperatur bei „Einlass schließt“ und liegt in der Regel, je nach Ansaug- und Aufladesystem, über der Umgebungstemperatur. Anstatt den Startdruck vorzugeben, wird hier die Temperatur verwendet (vgl. Abbildung 4-3 - grün), da in Kombination mit der Vorgabe einer Luftmasse, entsprechend der allgemeinen Gasgleichung, unrealistische kleine Temperaturen (vgl. Abbildung 4-3 - grau) für z.B. hohe Massen und niedrige Drücke entstehen können und die Simulation somit stabiler läuft. Nach unten wird die Starttemperatur durch den Kaltstart begrenzt und deckt mit 250 Kelvin nahezu alle Regionen der Welt ab. Bei Saugmotoren korreliert die Starttemperatur mit der Umgebungstemperatur. Bei aufgeladenen Systemen dagegen kann die Temperatur aufgrund höherer Drücke stark ansteigen. In der Regel ist bei Turbomotoren deshalb ein Ladeluftkühler verbaut, trotzdem wird der obere Grenzbereich für die Starttemperatur auf 500 Kelvin festgelegt, um Spielraum für verschiedene Auslegungen zu bieten.

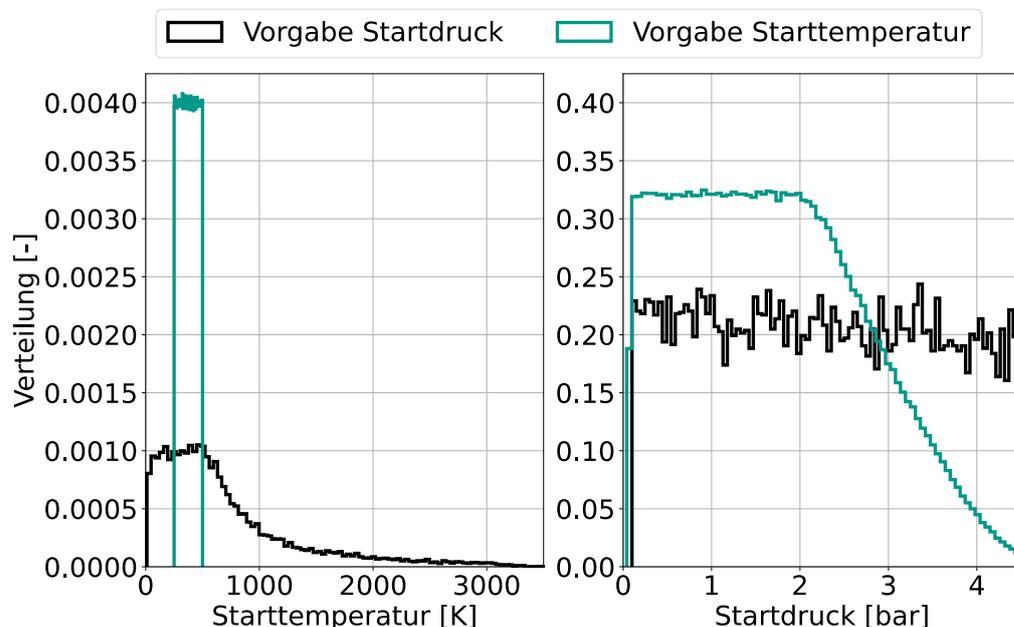


Abbildung 4-3: Unterschiede zwischen der Vorgabe der Starttemperatur oder des Startdrucks in der Simulation



4.1.6 Zündzeitpunkt

Der Zündzeitpunkt hat bei fremdgezündeten Brennverfahren einen maßgeblichen Einfluss auf den Brennverlauf, da ausgehend vom Zündfunken Kraftstoffenergie in Form einer sich ausbreitenden laminaren Flammenfront freigesetzt wird ([33], S.611). Je nach stöchiometrischem Verhältnis und Turbulenzniveau breitet sich die Flamme anschließend immer schneller im Brennraum aus, wodurch weitere Kraftstoffenergie umgesetzt und der Brennverlauf gestaltet wird. Nach Erreichen der maximalen Brennrate nimmt die Ausbreitungsgeschwindigkeit wieder ab und die Flamme kommt zum Erlöschen. Mit Grenzen für den Zündzeitpunkt zwischen -60 und 40 °KW um den oberen Totpunkt der Zündung (ZOT) sollen möglichst viele Brennverfahren und Betriebsstrategien abgedeckt werden.

Allerdings ergeben sich bei geeigneten Randbedingungen wie z.B. sehr früher Zündzeitpunkt und stöchiometrisches Lambda Schwerpunktlagen (50% Umsatzpunkt des Kraftstoffes) weit vor dem ZOT. Oder aber sehr späte Schwerpunktlagen bei magerem Lambda in Kombination mit späten Zündzeitpunkten. Daher wird die Zahl an unrealistischen Brennverläufen und Schwerpunktlagen in der Simulation mit Hilfe einer Gauß'schen Normalverteilung über die Zündzeitpunkte reduziert. In Abbildung 4-4 sind verschiedene Verteilungsfunktionen für den Zündzeitpunkt dargestellt. Der Mittelpunkt ($\mu=-10$) ist für alle Kurven gleich, weil für den Hauptanteil der Betriebspunkte eine Zündung vor ZOT angestrebt wird ([33], S.607). Mit Hilfe des σ -Anteils werden die Kurven gestaucht oder gestreckt. Die Wahl fällt auf $\sigma=20$ (grün), weil dort der vollständig gewünschte Zündzeitpunktbereich enthalten ist, und auch die Randbereiche gerade noch enthalten sind. Abbildung 4-5 zeigt den Unterschied der resultierend berechneten Schwerpunktlagen zwischen der uniformen und Gauß'schen Verteilung der Zündzeitpunktvariationen. Im üblicherweise gefahrenen Bereich der Schwerpunktlage zwischen 0 und 40 °KW (rot) um ZOT kann durch die Gauß'sche Verteilung ein deutlicher Zuwachs an realistischen Daten erzielt werden. Auch liegt die Maximalverteilung der Schwerpunktlage wirkungsgradoptimal zwischen 6 und 10 °KW ([39], S.1).

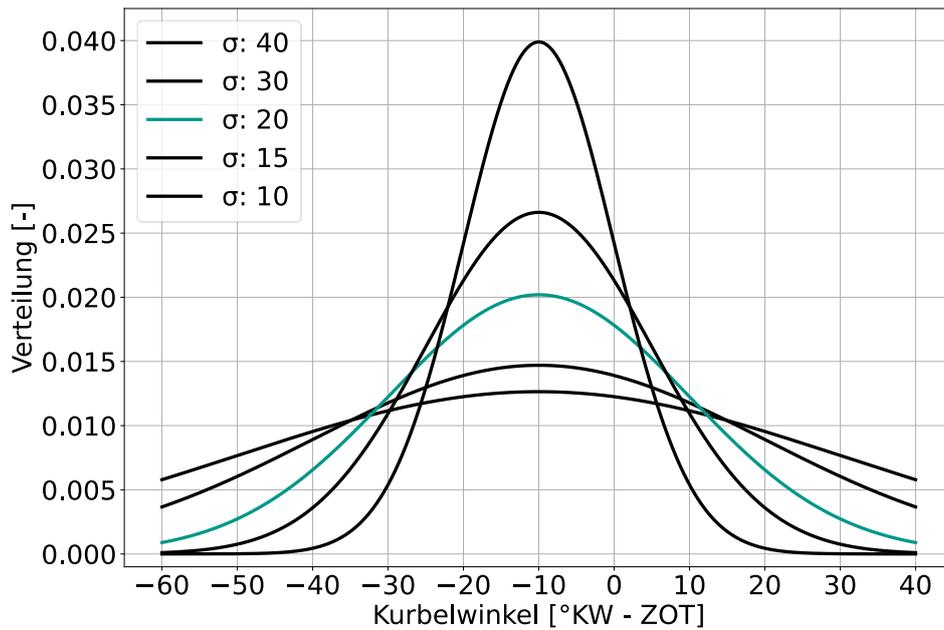


Abbildung 4-4: Normalverteilung des Zündzeitpunktes für $\mu=-10$ und verschiedene σ

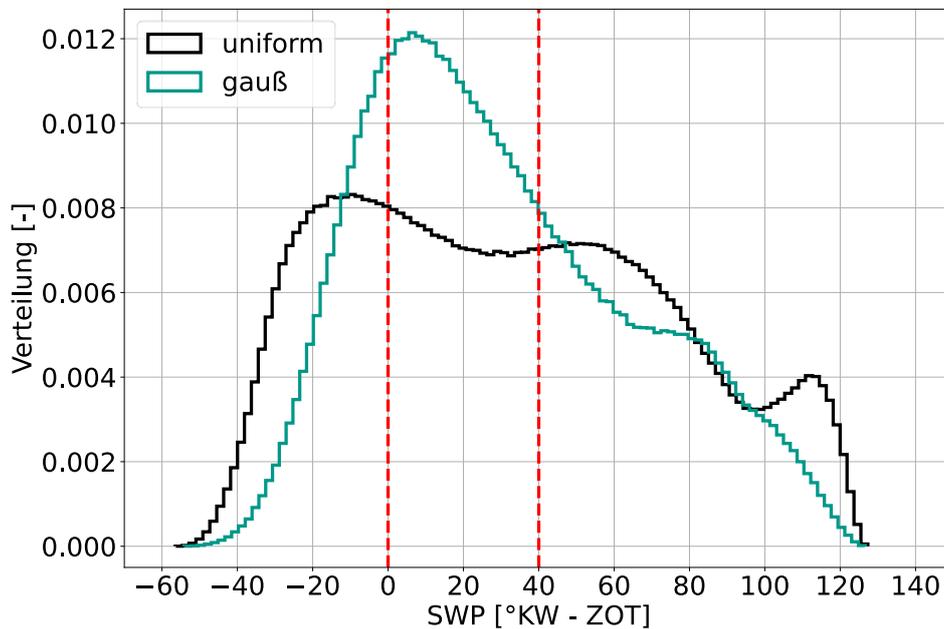


Abbildung 4-5: Verteilung der Schwerpunktlage bei uniformer und normaler Verteilung des Zündzeitpunkts

Theoretisch könnte der Anteil an realistischen Schwerpunktlagen mit komplexeren Verteilungsfunktionen, welche in Abhängigkeit zueinanderstehen, noch weiter erhöht werden. Allerdings kann die Abbildung in Randbereichen für eine Einbindung in andere Simulationsumgebung auch vorteilhaft sein, da während der Initialisierungsphase einer Simulation in GT-Suite durchaus unrealistische Zustände im Brennraum auftreten können.



Eine mögliche Verwendung des Zylindermodells mit Hilfe von neuronalen Netzwerken in anderen Simulationsumgebungen kann somit hinter geschalteten Regelungen konsistente Ergebnisse liefern und die Initialisierung bzw. Einregelung eines stationären Betriebszustands beschleunigen.

4.1.7 Bauteiltemperaturen

Obwohl in der folgenden Modellierung nur der Wandwärmeübergang im gesamte Brennraum betrachtet wird, werden die einzelnen Wandwärmeübergänge für Kopf, Kolben und Laufbuchse zum Erzielen genauere Ergebnisse berechnet und die entsprechenden, dort vorliegenden Temperaturen festgelegt. Diese werden nach unten wieder durch die Kaltstarttemperatur mit 250 Kelvin begrenzt. Abbildung 4-6 zeigt gemessene Kolbentemperaturen an verschiedenen Positionen für einen NFZ-Motor, bei dem die höchste Temperatur ca. 670 Kelvin beträgt. Nach ([45], S.458) wird der obere Grenzwert am Kolben und der Laufbuchse mit etwas Puffer für zukünftige Fortschritte in der Beschichtungstechnologie auf 800 Kelvin festgelegt. Die Temperatur am Kopf wird mit maximal 900 Kelvin etwas höher angesetzt, da dort i.d.R. weniger Kühlung ankommt und vereinzelt sehr heiße Zonen entstehen können.

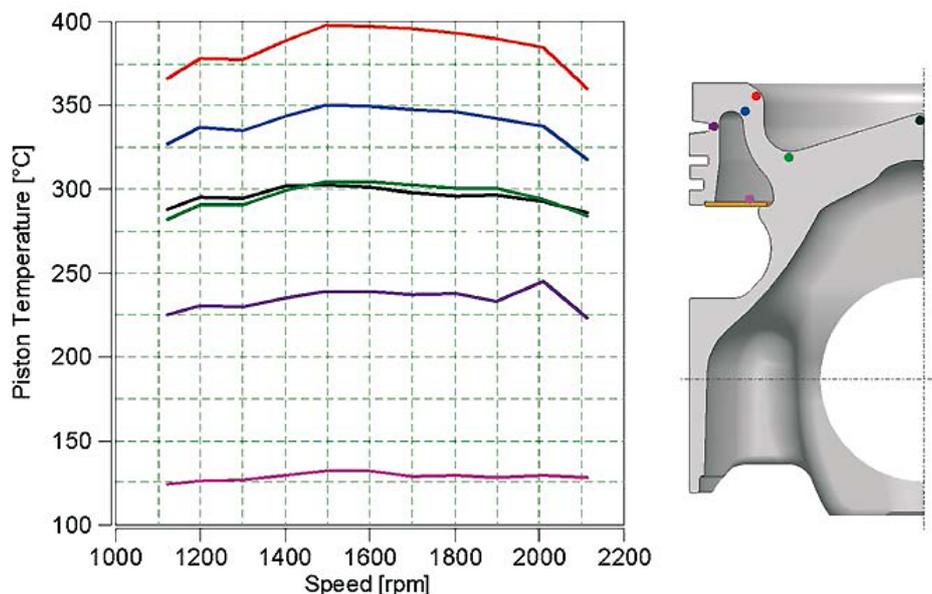


Abbildung 4-6: Kolbentemperaturen bei Vollast für NFZ-Motoren nach [45]

4.2 Modellierung I

Für die simulierten Kurbelwinkelverläufe wird zunächst keine Datenaufbereitung vorgenommen, da zunächst davon ausgegangen wird, dass die Simulationsumgebung Zylindermodul keine fehlerhaften Berechnungen zurückliefert.



Auf die Untersuchung der Berechnungsgenauigkeit des Zylindermoduls wird an dieser Stelle verzichtet, da diese verschiedenen vorangegangenen Veröffentlichungen (vgl. Kapitel 2.3) entnommen werden kann, und sich dieses Kapitel ausschließlich mit der Abbildung der Simulationsergebnisse mit Hilfe von neuronalen Netzwerken beschäftigt.

4.2.1 Datenaufteilung

Daher wird der vollständige Datensatz zuerst nach Tabelle 2-1 für die Modellierung aufgeteilt. Aufgrund der Vielzahl an berechneten Betriebspunkten, wird vorerst nur die Hälfte der Daten (Training + Validierung) für das Training der neuronalen Netze verwendet.

Anteil	Größe [%]
Training	30
Validierung	20
Test	50

Tabelle 4-3: Aufteilung des Ausgangsdatensatzes für die Modellierung I-I

4.2.2 Netzarchitektur

Da sich der zyklische Kurbelwinkelverlauf einer Kolbenrotation bei konstanten Ventilsteuerzeiten in der Hochdruckphase auf die Zeit zwischen Einlass schließt und Auslass öffnet begrenzt, können ausgehend von den Eingangsparametern an der Eingangsschicht Starttemperatur, Lambda, Luftmasse, Drehzahl, Anteil der Abgasrückführung, Zündzeitpunkt und Bauteiltemperaturen über drei Zwischenschichten L1 - L3 für jedes Ausgangsneuron auf der Ausgangsschicht ein Grad Kurbelwinkel abgebildet werden. Die Kurbelwinkelverläufe sind in sich abgeschlossen, also stationär, und über die Eingangsparameter ist der Bezug zu neuen Startbedingungen gegeben, weshalb für die Modellierung keine rekurrenten Netzarchitekturen verwendet werden, sondern kontinuierliche (vgl. Kapitel 2.1.6). In Abbildung 4-7 und Tabelle 4-4 ist die Netzarchitektur in Anlehnung an ([16], S.12) veranschaulicht und die entsprechende Parametrierung für den ersten Modellierungsversuch I-I gelistet. Das verwendete Optimierungsverfahren ist ADAM (vgl. Kapitel 2.1.6) verwendet.

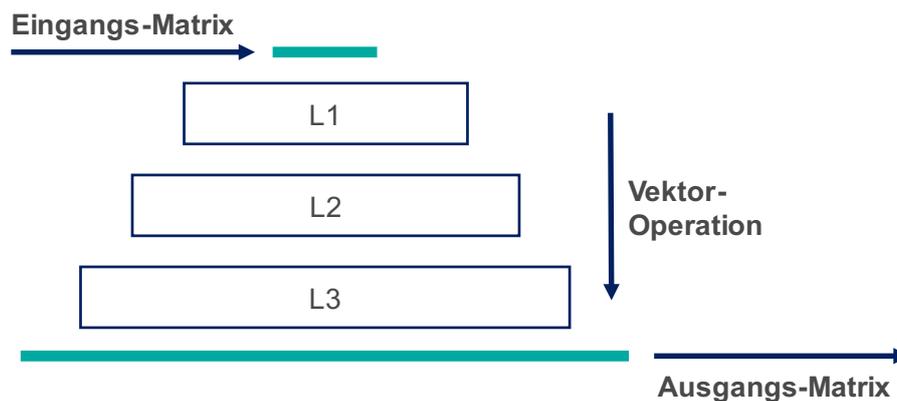


Abbildung 4-7: Netzlayout für die kurbelwinkelaufgelöste Modellierung I

Parameter	Wert
L1	100 Neuronen
L2	200 Neuronen
L3	400 Neuronen
Ausgang	308 Neuronen
Aktivierung	ReLu
Fehlerfunktion	MAE
Lernrate	10 ⁻³
Abbruchbedingung	5 Epochen
Eingangsdimension	(1x8)
Ausgangsdimension	(1x308)

Tabelle 4-4: Netzparameter für die kurbelwinkelaufgelöste Modellierung I-I

4.2.3 Druckverlauf I-I

Zuerst werden die Zylinderdruckverläufe betrachtet. Nach 21 Durchläufen wird entsprechend der Abbruchbedingung mit fünf Durchläufen ohne Modellverbesserung ein durchschnittlicher absoluter Fehler von 0.0862 bzw. 0.0736 bar für den Trainings- und Validierungsdatensatz mit Parametrierung I, gemittelt über alle Druckverläufe, erreicht. Der Fehler bei der Modellierung der ungesehenen Testdaten ist mit 0.0807 bar etwas höher als bei den Validierungsdaten, was zu erwarten war, liegt aber dennoch in einem zufriedenstellenden Bereich.



Da die Fehler bei der Abbildung der Test- und Validierungsdaten geringer als der Trainingsfehler ausfallen und nur wenige Epochen mit hoher Lernrate trainiert werden, kann davon ausgegangen werden, dass keine Überanpassung des Modells (engl. Overfitting) stattgefunden hat ([76], S.2). Dies soll im Folgenden jedoch näher untersucht werden.

Datensatz	MAE [bar]
Training	0.0862
Validierung	0.0736
Test	0.0807

Tabelle 4-5: Gesamtfehler für die Modellierung des Druckverlaufs I-I

Für detailliertere Untersuchungen wird exemplarisch zuerst ein Betriebspunkt (vgl. Tabelle 4-6) mit stöchiometrischem Lambda, hoher Drehzahl, frühem Zündzeitpunkt, aufgewärmten Bauteiltemperaturen und ohne Abgasrückführung betrachtet, wie er z.B. in der Vollast auftreten kann. In Abbildung 4-8 sind die simulierten Zylinderdruckverläufe für verschiedene Luftmassen in schwarz und deren Modellierung mit Hilfe der neuronalen Netzarchitektur in grün dargestellt. Aufgrund der hohen laminaren Flammgeschwindigkeit von Wasserstoff als verwendetes Brenngas steigen die Drücke nach der Zündung zunächst stark an. Nach Erreichen des Druckmaximums sinkt der Zylinderdruck hinter dem oberen Totpunkt auf Grund der Expansion wieder ab. Mit steigender Füllung (von unten nach oben) nimmt auch der erreichte Spitzendruck zu, da nach der allgemeinen Gasgleichung mehr Luftmasse auf dasselbe Volumen verdichtet werden muss, und bei gleichbleibendem Lambda mehr Kraftstoffmasse umgesetzt wird.

Da der Spitzendruck in der Regel konstruktiv bedingt auf einen maximal zulässigen Wert begrenzt wird, ist im Diagramm exemplarisch eine rote Linie für einen Druck von 250 bar gezeichnet, welcher gerade noch mit aktuellen Indizierkerzen messbar ist ([36], S.1). Damit soll veranschaulicht werden, wie das neuronale Netzwerk z.B. bei der Vorentwicklung für die Auslegung des Motorenkennfelds genutzt werden oder z.B. in Serienmotoren bei der Klopfregelung unterstützen kann. Falls diese Spitzendruckgrenze bei gegebenen Randbedingungen voraussichtlich überschritten werden könnte, kann z.B. der Zündzeitpunkt und somit die Schwerpunktlage präventiv nach hinten verschoben werden.



Konstante	Wert	Einheit
Drehzahl	2000	1/min
Lambda	1.0	-
Rate AGR	0	%
Starttemperatur	350	Kelvin
Zündzeitpunkt	-20	°KW - ZOT
Temperatur Kopf	800	Kelvin
Temperatur Kolbenbuchse	700	Kelvin
Temperatur Kolben	700	Kelvin

Tabelle 4-6: Konstante Parameter für die Auswertung der Zylinderdruckverläufe

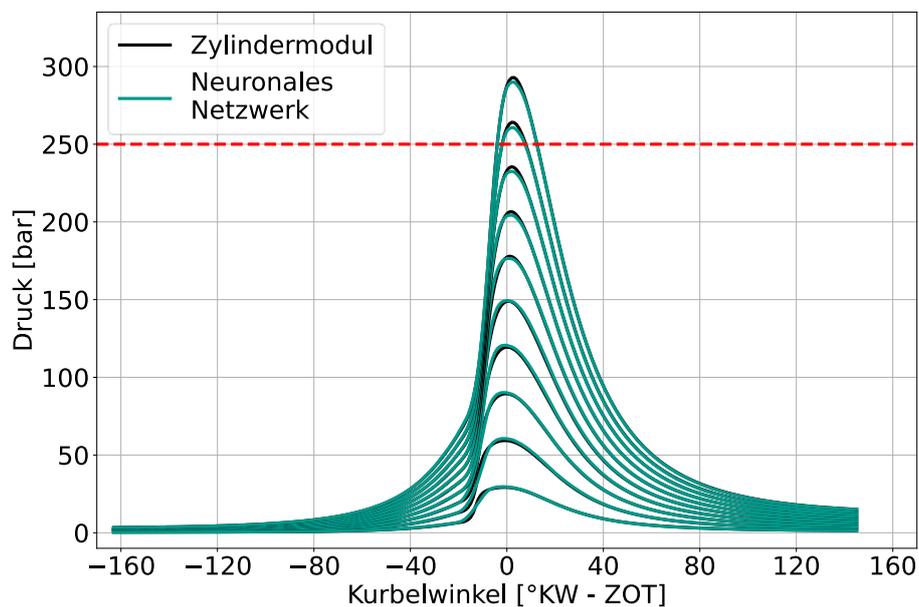


Abbildung 4-8: Auswertung der Druckverläufe für verschiedene Füllungen und Randbedingungen nach Tabelle 4-6 mit Parametrierung I-I

In Tabelle 4-7 sind die durchschnittlichen und maximalen absoluten Fehler für verschiedene Luftmassen (von unten nach oben) zwischen berechnetem und modelliertem Druckverlauf gelistet.



Der durchschnittliche Fehler beträgt durchgehend unter 0.3 bar, somit kann der qualitative Zusammenhang zwischen einer Änderung der Füllung und dem Verlauf des Zylinderdrucks als zufriedenstellend abgebildet gekennzeichnet werden.

Die maximalen Fehler liegen zwischen drei und 8 bar, was z.B. im unteren Teillastbetrieb bei einem Spitzendruck von 50 bar eher viel ist. Dort wird teilweise die Zündverzugszeit nicht richtig abgebildet, was in einem zu frühen Anstieg des modellierten Druckverlaufs gegenüber der Simulation resultiert. Zudem wird der Spitzendruck für hohe Füllungen etwas zu niedrig abgebildet. An dieser Stelle wird erwähnt, dass jeder der betrachteten Verläufe, wie in allen Kapiteln dieser Arbeit, Testdaten entsprechen, und diese nicht im Training oder in der Validierung enthalten sind.

Luftmasse [mg]	MAE [bar]	Maximaler Fehler [bar]
500	0.116	2.786
1000	0.260	4.681
1500	0.187	4.454
2000	0.240	2.987
2500	0.276	3.619
3000	0.299	7.227
3500	0.280	7.938
4000	0.263	7.239
4500	0.297	4.769
5000	0.241	3.088

Tabelle 4-7: MAE und maximaler Fehler bei der Modellierung des Druckverlaufs für verschiedene Füllungen I-I

4.2.4 Druckverlauf I-I.II

Ausgehend von den Ergebnissen im Modellierungsversuch I werden die Netzparameter optimiert. Dafür wird im Modellierungsversuch I-I.II der Anteil an Trainingsdaten auf die Hälfte erhöht. Außerdem werden mehr Trainingsdurchläufe ohne Modellverbesserung für das Abbruchkriterium zugelassen sowie die Anzahl an Neuronen verdoppelt. Dabei sollen die Netzgewichte zum einen durch den Zuwachs an Daten und veränderbaren Parametern verbessert werden, zum anderen soll die Wahrscheinlichkeit, das globale Abbildungsoptimum zu erreichen, erhöht werden.



Zusätzlich wird, anstatt der Optimierung über den MAE, auf die Fehlerbetrachtung nach Huber (vgl. Kapitel 2.1.4) gewechselt, da somit sowohl der qualitative Kurvenverlauf als auch die quantitativen Absolutwerte besser abgebildet werden können. Tabelle 4-8 listet die Änderungen der Trainingsarchitektur für den Modellierungsversuch I-I.II

Parameter	Wert
Trainingsdaten	50 %
Validierungsdaten	20 %
Testdaten	30 %
Abbruchbedingung	25 Epochen
Fehlerfunktion	Huber
L1	200 Neuronen
L2	400 Neuronen
L3	800 Neuronen

Tabelle 4-8: Änderungen der Trainingsarchitektur für die Modellierung I-I.II

In Abbildung 4-9 sind wieder die berechneten Zylinderdruckverläufe mit den Randbedingungen aus Tabelle 4-6 in schwarz und die modellierten Verläufe mit Hilfe des neuronalen Netzes in grün aufgetragen. Es lässt sich erkennen, dass die Zündverzugszeit zufriedenstellend abgebildet ist, da die Kurven am Druckanstieg durchweg übereinstimmen. Auch die Druckmaxima sind deutlich besser abgebildet als zuvor. Tabelle 4-9 bestätigt den aus der Abbildung gewonnenen Eindruck. Die durchschnittlichen absoluten und auch die maximalen Fehler haben sich mehr als halbiert und liegen nun durchgehend in einem zufriedenstellenden Bereich.

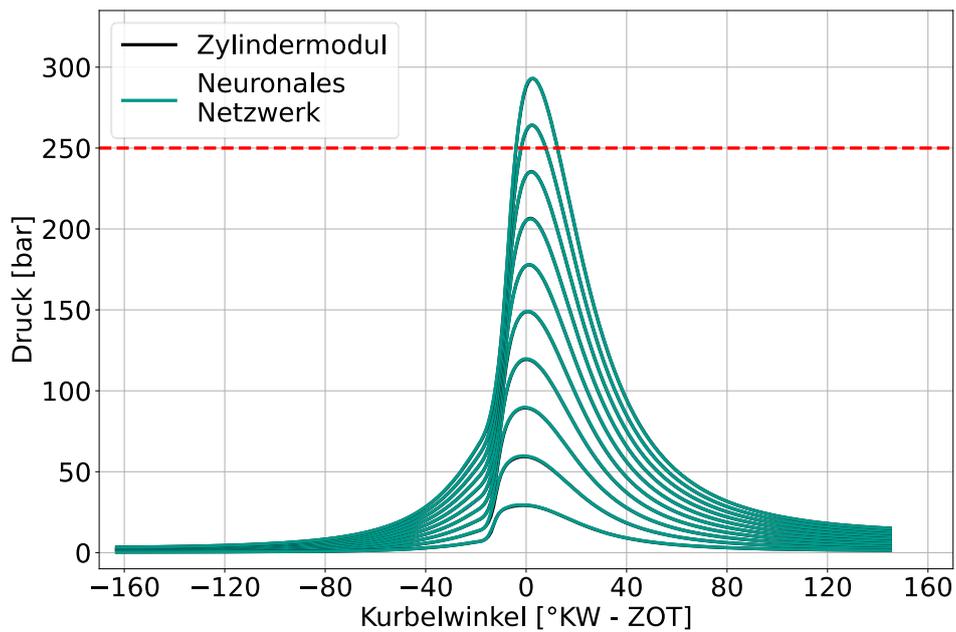


Abbildung 4-9: Auswertung der Druckverläufe für verschiedene Füllungen und Randbedingungen nach Tabelle 4-6 mit Parametrierung I-I.II

Luftmasse [mg]	MAE [bar]	Maximaler Fehler [bar]
500	0.059	1.094
1000	0.085	2.676
1500	0.054	1.559
2000	0.048	0.801
2500	0.063	1.645
3000	0.053	0.482
3500	0.075	1.902
4000	0.075	1.699
4500	0.062	1.052
5000	0.112	2.155

Tabelle 4-9: MAE und maximaler Fehler bei der Modellierung des Druckverlaufs für verschiedene Füllungen I-I.II



4.2.5 Druckgradient I-II

Ausgehend von der optimierten Trainingsarchitektur I-I.II wird nun der Gradient des Zylinderdruckverlaufs modelliert. Dieser ist bei der Auslegung von Motoren ebenso wichtig wie der Maximaldruck, da, durch einen hohen Druckanstieg bedingte Kräfte den Kolben oder das Kurbelgehäuse beschädigen können. Gerade bei der Verwendung von Wasserstoff als Brenngas, kann es aufgrund der hohen laminaren Flammgeschwindigkeit im Stöchiometrischen zu extremen Druckgradienten kommen. Nach ([21], S.130) wird der maximal zulässige Druckgradient auf 200 bar/ms festgelegt. Weil die Betrachtungen in diesem Kapitel kurbelwinkelaufgelöst sind, bedarf es daher einer Umrechnung des zeitaufgelösten Gradienten nach Gleichung (4.1).

$$2000 \frac{1}{min} = 33.333 \frac{1}{s} = 11999.88 \frac{^{\circ}KW}{s} \approx 12 \frac{^{\circ}KW}{ms}$$
$$\frac{200 \frac{bar}{ms}}{12 \frac{^{\circ}KW}{ms}} = 16.667 \frac{bar}{^{\circ}KW} \quad (4.1)$$

In Abbildung 4-10 sind die Druckgradienten für dieselbe Luftmassenvariation aus Abbildung 4-9 dargestellt. Die Kurbelwinkelachse konzentriert sich etwas stärker um den ZOT, da dort die merklichen Druckgradienten dargestellt sind. Es lässt sich erkennen, dass zum einen Luftmassen mit den gegebenen Randbedingungen nach Tabelle 4-6, welche aufgrund des auftretenden Spitzendrucks theoretisch fahrbar wären, durch die Begrenzung des Druckgradienten nicht im Brennraum auftreten dürfen. Zum anderen liefert die Modellierung mit Hilfe des neuronalen Netzwerks zufriedenstellende Ergebnisse, da der Gradientenverlauf sowohl qualitativ als auch quantitativ gut abgebildet ist. Der größte durchschnittliche Fehler liegt bei 0.023 bar/°KW und der größte maximale Fehler bei 0.850 bar/°KW. Ebenso ist der berechnete Datensatz, wie in Tabelle 4-10 gelistet, mit geringen Fehlern im Trainings-, Validierungs- und Testdatensatz zufriedenstellend abgebildet. Auch hier sind die Ergebnisse bei der Vorhersage von Validierungs- und Testdaten bei hohen Lernraten und wenigen Epochen besser als im Training, was darauf schließen lässt, dass die grundsätzlichen Zusammenhänge zwischen einer Änderung der Startbedingungen bei Einlass schließt und dem Verlauf des Druckgradienten vom Modell ohne Überanpassung abgebildet werden.



Datensatz	MAE [bar/°KW]
Training	0.00619
Validierung	0.00512
Test	0.00574

Tabelle 4-10: Gesamtfehler für die Modellierung der Druckgradienten I-II

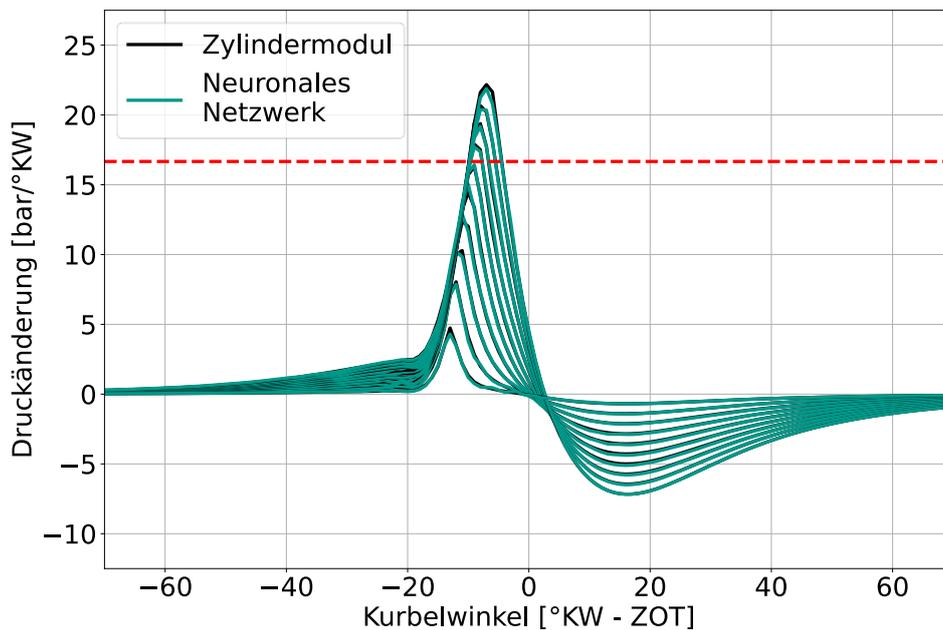


Abbildung 4-10: Auswertung der Druckgradienten aus Abbildung 4-9

4.2.6 Brennverlauf I-III

Als nächstes werden die Modelle für die Brennverlaufs- und Wandwärmemodellierung ausgewertet. Die Fehlerbetrachtung für den Gesamtdatensatz in Tabelle 4-11 und Tabelle 4-12 weist darauf hin, dass die Modellierung I-III auch hier zufriedenstellende Ergebnisse liefert.

Datensatz	MAE [J/°KW]
Training	0.304
Validierung	0.256
Test	0.289

Tabelle 4-11: Gesamtfehler für die Modellierung des Brennverlaufs I-III



Datensatz	MAE [J/°KW]
Training	0.0231
Validierung	0.0193
Test	0.0219

Tabelle 4-12: Gesamtfehler für die Modellierung des Wandwärmeverlaufs I-III

Für weitere und detailliertere Untersuchungen bei der Modellierung des Brenn- und Wandwärmeverlaufs wird dieses Mal eine komplexere Auswertemethodik herangezogen, bei der mehrere Eingangsgrößen in Abhängigkeit zueinander variiert werden. Ausgehend von einer konstanten Drehzahl und Starttemperatur wird die Luftmasse verringert, das Lambda abgemagert, der Zündzeitpunkt Richtung spät geschoben und die Rate der Abgasrückführung erhöht. Zusätzlich werden die Bauteiltemperaturen abgesenkt. Damit werden komplexe und umfangreiche Wechselwirkungen im Brennraum simuliert, welche die Abbildungsgenauigkeit der neuronalen Netzwerke darüber prüfen sollen.

Parameter	Start	Ende	Schrittweite
Drehzahl	2000 1/min	-	-
Starttemperatur	350 Kelvin	-	-
Luftmasse	4000 mg	1000 mg	-300 mg
Lambda	1	4	0.3
Zündzeitpunkt	-20 °KW-ZOT	10 °KW-ZOT	3 °KW
Rate AGR	0 %	20 %	2 %
Temperatur Kopf	800 Kelvin	500 Kelvin	-30 Kelvin
Temperatur Kolbenbuchse	700 Kelvin	400 Kelvin	-30 Kelvin
Temperatur Kolben	700 Kelvin	400 Kelvin	-30 Kelvin

Tabelle 4-13: Parametervariation für die Auswertung der Brenn- und Wandwärmeverläufe



In Abbildung 4-11 sind die berechneten Brennverläufe entsprechend der Parametrierung nach Tabelle 4-13 in schwarz und die modellierten Vorhersagen des neuronalen Netzes in grün aufgetragen. Zusätzlich ist der berechnete indizierte Mitteldruck aufgetragen, um eine greifbare Größe des Lastzustands daneben zu stellen. Ausgehend vom Zündzeitpunkt steigt der Brennverlauf aufgrund der zunehmenden Umsetzung des Brennstoffes je nach Lambda unterschiedlich stark an. Das Maximum der Brennstoffumsetzung wird mit sinkender Luftmasse und stärker abgemagertem Lambda zunehmend kleiner. Außerdem verschiebt sich die Schwerpunktlage mit späterem Zündzeitpunkt nach hinten, und die Brenndauer steigt mit größerem Restgasanteil im Brennraum. Dies führt dazu, dass der indizierte Mitteldruck zunehmend kleiner wird, teilweise sogar negativ ist und die Kurven unterschiedlich stark gestaucht bzw. gestreckt werden. Bei zwei Verläufen kommt es zu keiner Kraftstoffumsetzung, was in einem negativen indizierten Mitteldruck bzw. einem konstantem Nulldurchgang resultiert.

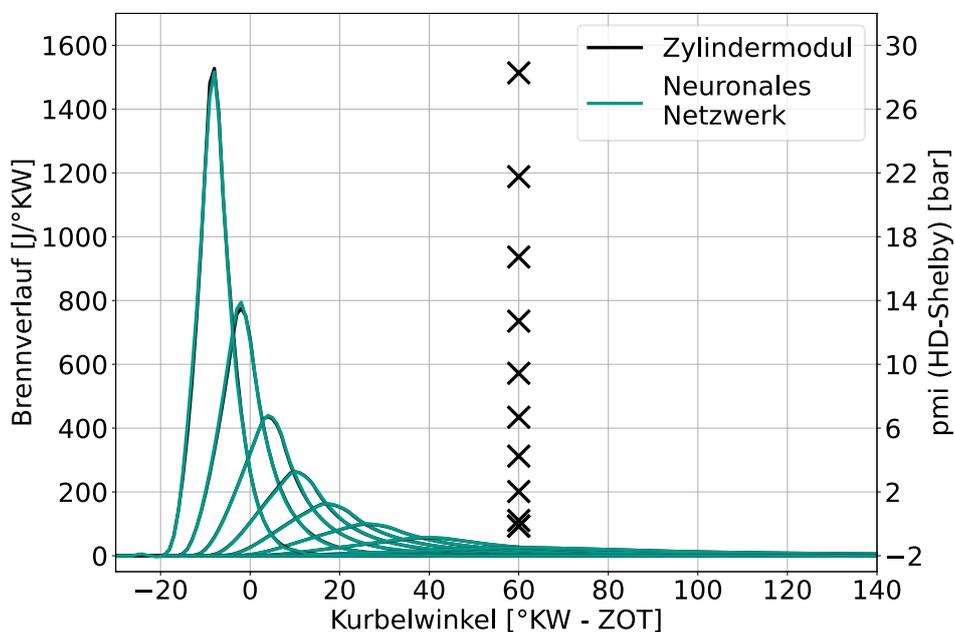


Abbildung 4-11: Auswertung der Brennverläufe für die Parametervariation nach Tabelle 4-13 mit Modellierung I-III

In Tabelle 4-14 sind die durchschnittlichen absoluten und maximalen Fehler bei der Modellierung der Brennverläufe für die verschiedenen indizierten Mitteldrücke aus Abbildung 4-11 gelistet. Die durchschnittlichen Fehler sind gering und liegen unter 2 J/°KW, was bedeutet, dass der qualitative Kurvenverlauf zufriedenstellend abgebildet ist. Auch die maximalen Fehler sind gering, wobei diese mit zunehmendem Umsatzmaximum steigen. Daher sind die maximalen Fehler zusätzlich prozentual, bezogen auf den dort vorkommenden Brennstoffumsatz, gelistet.



Für indizierte Mitteldrücke größer 4 bar liegen die maximalen Abweichungen bei der Modellierung des Brennverlaufs unter 4 %. Bei indizierten Mitteldrücken kleiner 4 bar liegen die maximalen Fehler zwischen 2.5 und 8.5 %, wobei die prozentuale Betrachtung bei Vergleichsgrößen kleiner null nicht aussagekräftig ist. Da keine Prüfstandsdaten vorliegen, lässt sich an dieser Stelle keine Aussage darüber treffen, ob die Simulation oder die Modellierung mit Hilfe der neuronalen Netzwerke für die etwas höheren Abweichungen in der unteren Teillast kleiner 4 bar pmi verantwortlich sind.

pmi [bar]	mae [J/°KW]	e_max [J/°KW]	e_max [%]
-0.143	0.069	0.351	-
-0.14	0.069	0.240	-
0.207	0.081	0.485	-
2.025	0.054	0.670	8.521
4.242	0.142	1.342	2.419
6.682	0.108	1.314	1.330
9.439	0.190	2.197	1.923
12.705	0.300	4.213	1.848
16.727	0.304	6.991	2.798
21.764	0.723	20.878	3.616
28.272	1.232	43.326	2.924

Tabelle 4-14: MAE und maximaler Fehler nach Modellierung I-III des Brennverlaufs für verschiedene indizierte Mitteldrücke

4.2.7 Wandwärmeverlauf I-IV

Bei den Wandwärmeübergängen in Abbildung 4-12 ist die Modellierungsgenauigkeit besser. Mit abnehmender Füllung und abmagerndem Lambda sinken die maximalen Wandwärmeverluste, da weniger Kraftstoff umgesetzt wird. Mit dem Zündzeitpunkt wird auch die Lage des maximalen Wandwärmeverlusts Richtung spät geschoben. Interessant ist, dass der Wandwärmeübergang nach der Verbrennung bei hohen Temperaturen im Brennraum größer ist als bei niedrigen.



Das bedeutet, je höher die Bauteiltemperaturen sind, desto mehr Wärme wird auch nach außen abgegeben und ergibt sich aus der Berechnung des Wandwärmeverlaufs nach Woschni über den Newtonschen Ansatz zur Berechnung des mittleren Wärmeübergangs und der Temperaturdifferenz ([12], S.252).

Sowohl die Kurvenverläufe als auch die Absolutwerte werden dabei vom Modell zufriedenstellend abgebildet.

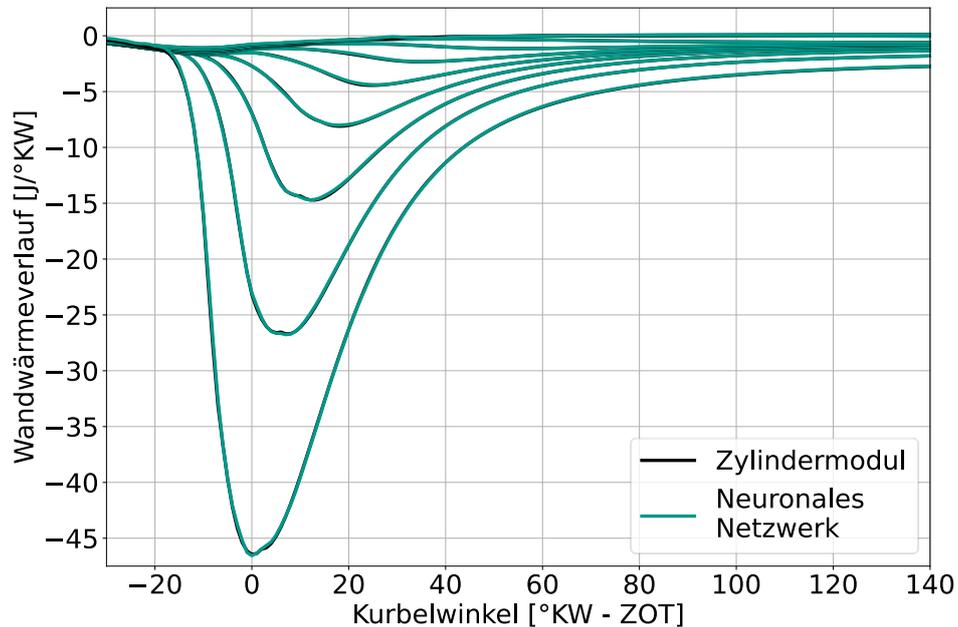


Abbildung 4-12: Auswertung der Wandwärmeverläufe für die Parametervariation nach Tabelle 4-13 mit Modellierung I-IV

Das bestätigt auch Tabelle 4-15, in der die durchschnittlichen absoluten und maximalen Fehler für die Modellierung des Wandwärmeeübergangs für verschiedene indizierte Mitteldrücke gelistet sind. Diese liegen durchgehend unter 1 J/°KW und können als zufriedenstellend abgebildet gekennzeichnet werden.



pmi [bar]	MAE [J/°KW]	Maximaler Fehler [J/°KW]
-0.143	0.0154	0.1321
-0.14	0.0106	0.0926
0.207	0.0152	0.0878
2.025	0.0095	0.0633
4.242	0.0104	0.0434
6.682	0.0098	0.0800
9.439	0.0109	0.0987
12.705	0.0157	0.1020
16.727	0.0167	0.1359
21.764	0.0175	0.2527
28.272	0.0497	0.7383

Tabelle 4-15: MAE und maximaler Fehler nach Modellierung I-IV des Wandwärmeverlaufs für verschiedene indizierte Mitteldrücke

4.3 Belastungstest

Als letztes werden die Modelle einem Belastungstest unterzogen. Dafür werden die Startbedingungen für den Zylinder innerhalb der Grenzen von Tabelle 4-2 fünf Millionen Mal zufällig initialisiert und die entstehenden, unbekanntenen Druck-, Druckgradienten-, Brenn- und Wandwärmeverläufe mit Hilfe der neuronalen Netzmodelle vorhergesagt. Als Referenz wird gleichzeitig eine entsprechende Berechnung mit dem Zylindermodul durchgeführt und die beiden Ergebnisse miteinander verglichen.

Somit sollen Grenzphänomene im Brennraum erzwungen werden, die erstmal nicht an das physikalische Verständnis des Menschen angelehnt sind, und die Vorhersagegrenzen der Algorithmen herausgearbeitet werden, da bisher keine wesentlichen Abweichungen zwischen den erstellten Modellen und der Simulation gefunden werden konnten.

In Tabelle 4-16 sind Grenzwerte für den durchschnittlichen absoluten und maximalen Fehler zwischen der Berechnung und Modellvorhersage gelistet, oberhalb derer die Ergebnisse gespeichert und ausgewertet werden.



Bei den durchschnittlichen absoluten Fehlern liegen nur einige wenige Verläufe der fünf Millionen zufällig initialisierten Startbedingungen knapp über dem Grenzwert, weshalb der qualitative Kurvenverlauf der Modelle endgültig als zufriedenstellend abgebildet gekennzeichnet werden kann. Bei den maximalen Fehlern gibt es einige Stellen, an denen hohe Abweichungen zwischen den Berechnungen und Modellierungen auftreten. Diese sollen im Folgenden analysiert werden.

Parameter	Grenze MAE	Anzahl	Grenze emax	Anzahl
Druck	1 bar	9	10 bar	1648
Druckgradient	0.2 bar/°KW	50	2 bar/°KW	10893
Brennverlauf	40 J/°KW	34	1000 J/°KW	3187
Wandwärmeverlauf	0.7 J/°KW	22	7 J/°KW	5138

Tabelle 4-16: Maximal zulässige Abweichungen und Überschreitungen während dem Belastungstest der neuronalen Netzmodelle

Dafür sind in Abbildung 4-13 die neun größten maximalen Fehler bei der Modellierung des Druckverlaufs abgebildet, die während des Belastungstests aufgetreten sind. Es lässt sich erkennen, dass das Modell teilweise Schwierigkeiten hat, den exakten Kurbelwinkel eines lokalen oder globalen Druckmaximums- bzw. Minimums zu treffen. Eine Verschiebung von wenigen Grad kann hier in Kombination mit hohen Druckgradienten bereits zu starken Abweichungen führen. Durch eine feinere Auflösung der Kurbelwinkelachse kann die Modellgenauigkeit an dieser Stelle weiter verbessert werden.

Grundsätzlich werden aber auch außergewöhnliche Druckverläufe wie z.B. oben links qualitativ gut abgebildet, bei denen der Druck durch die Dekompression nach ZOT zunächst erst abfällt und anschließend, bedingt durch eine späte Zündung und hohe Brennraten, wieder stark ansteigt.

Nur im mittleren rechten Bild gibt es im Druckverlauf beim Zusammenspiel zwischen Kompressionsdruck und Verbrennung eine kleinere Unstimmigkeit. Dort stagniert der modellierte Druckanstieg kurzzeitig, was physikalisch nicht erklärbar ist. Die Berechnung liefert hier einen konstant steigenden Druck. Das Druckverhalten im Ausbrand und in der Dekompression stimmt dort dagegen wieder vollständig mit der Simulation überein.

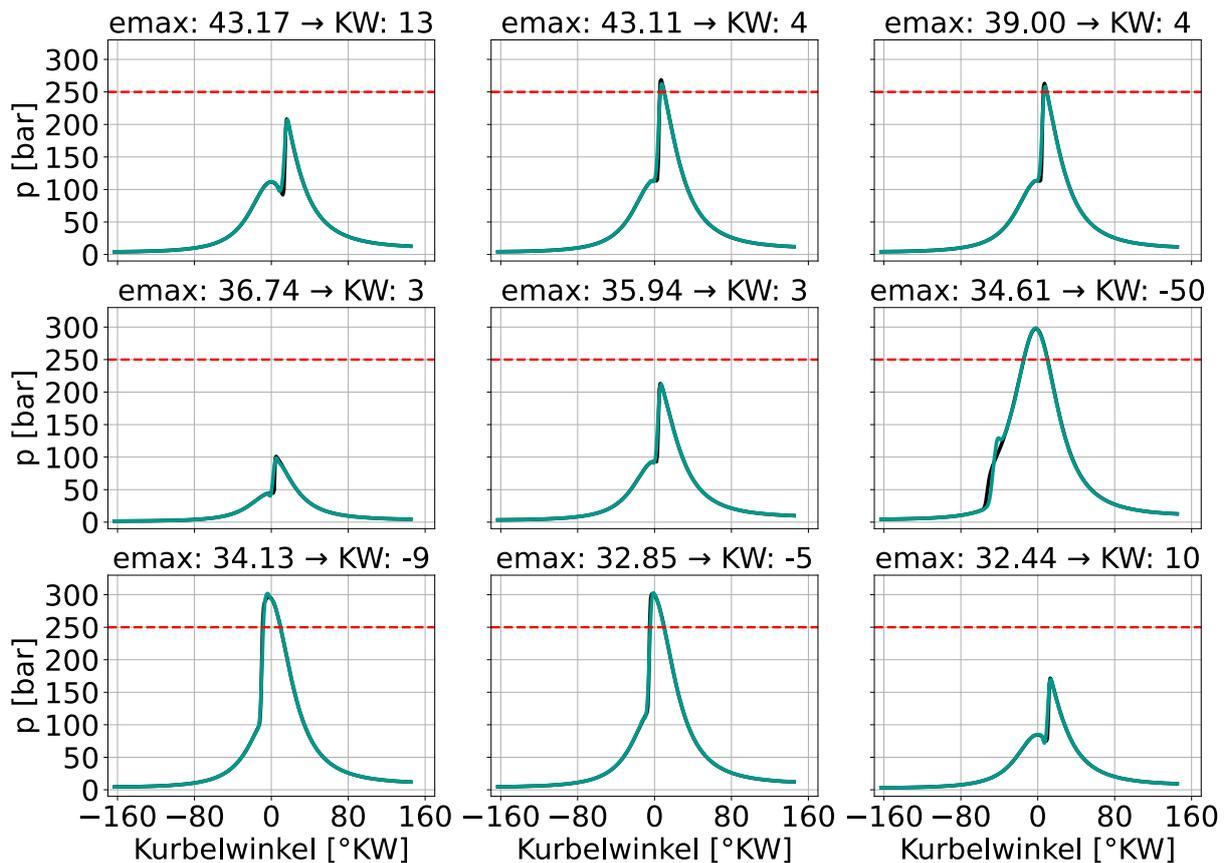


Abbildung 4-13: Größte maximale Fehler während des Belastungstests für die modellierten Druckverläufe I-I.II

Abbildung 4-14 zeigt die neun Verläufe der Druckgradienten mit den höchsten maximalen Abweichungen während des Belastungstests. Zusätzlich ist die Brenndauer aufgetragen (φ im Diagramm mit rotem Hintergrund), welche die Kurbelwinkeldauer beschreibt, zwischen welcher der Kraftstoff zu 10 bis 90 % umgesetzt wird. Generell können bei der Verbrennung von Wasserstoff, bedingt durch die hohe laminare Flammgeschwindigkeit im Stöchiometrischen, extreme Druckgradienten entstehen. Teilweise liegen diese viermal höher als die mechanisch fahrbare Obergrenze. Dies führt zu sehr kurzen Brenndauern, bei denen der Brennstoff in wenigen Kurbelwinkeln vollständig umgesetzt wird. Offensichtlich hat das Modell mit der in 1 °KW-Schritten aufgelösten Kurbelwinkelachse Probleme, diese extremen Gradienten abzubilden. Hier liegt ebenfalls nahe, das Modell mit einer feiner, z.B. in 0.1 °KW-Schritten aufgelösten Ausgangsschicht zu optimieren.

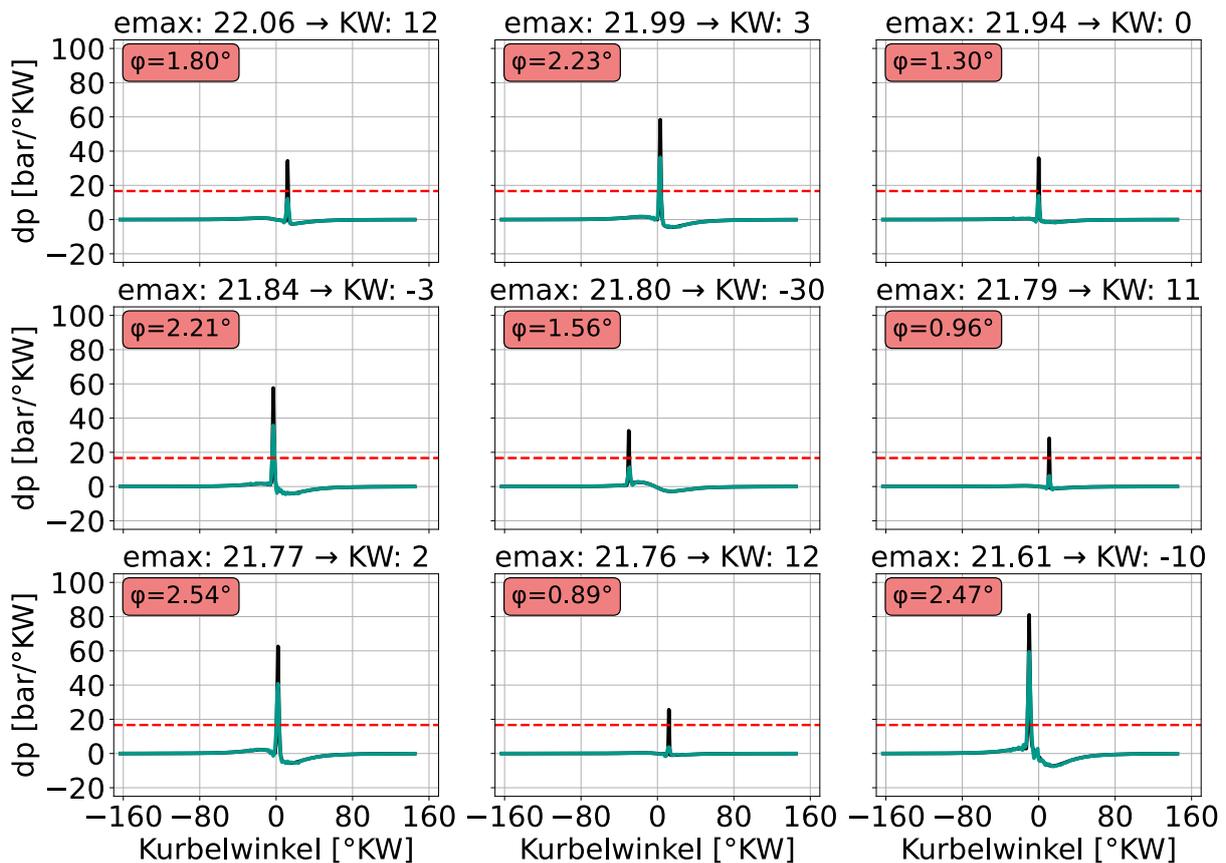


Abbildung 4-14: Größte maximale Fehler während des Belastungstests für die modellierten Druckgradienten I-II

Auch ein Blick in die Auswertung der modellierten Brennverläufe mit den höchsten Abweichungen während des Belastungstests zeigt ein ähnliches Verhalten in Abbildung 4-15. Hohe Brennraten in Kombination mit einer kurzen Brenndauer bringen das Modell an seine Grenzen. Dieses Verhalten zeigt sich ebenfalls bei den Verläufen mit den höchsten Abweichungen der Wandwärmeübergänge im Belastungstest auf Abbildung 4-16. Dort wird der Verlauf solange zufriedenstellend modelliert, bis es zu einem starken Anstieg bzw. Abfall des Wandwärmestroms aufgrund einer sehr kurzen Brenndauer kommt.

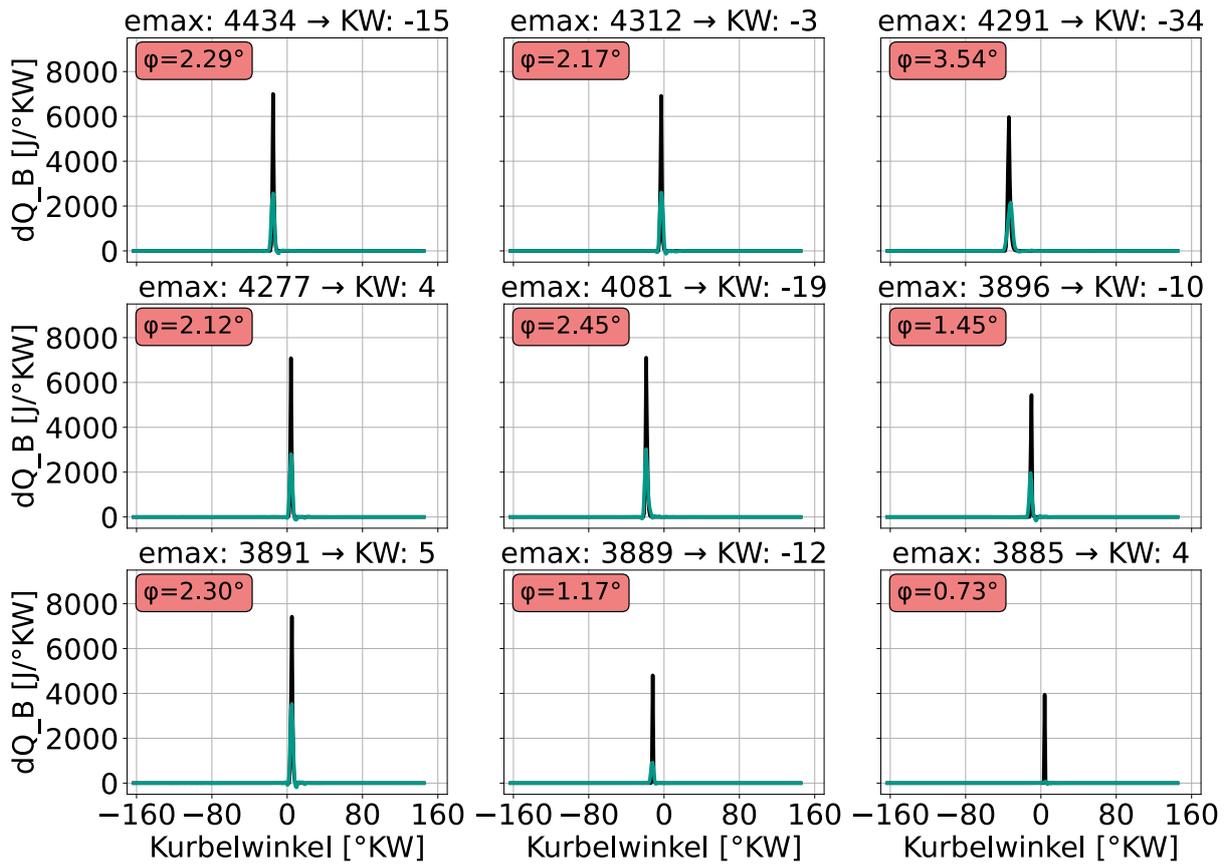


Abbildung 4-15: Größte maximale Fehler während des Belastungstests für die modellierten Brennverläufe I-III

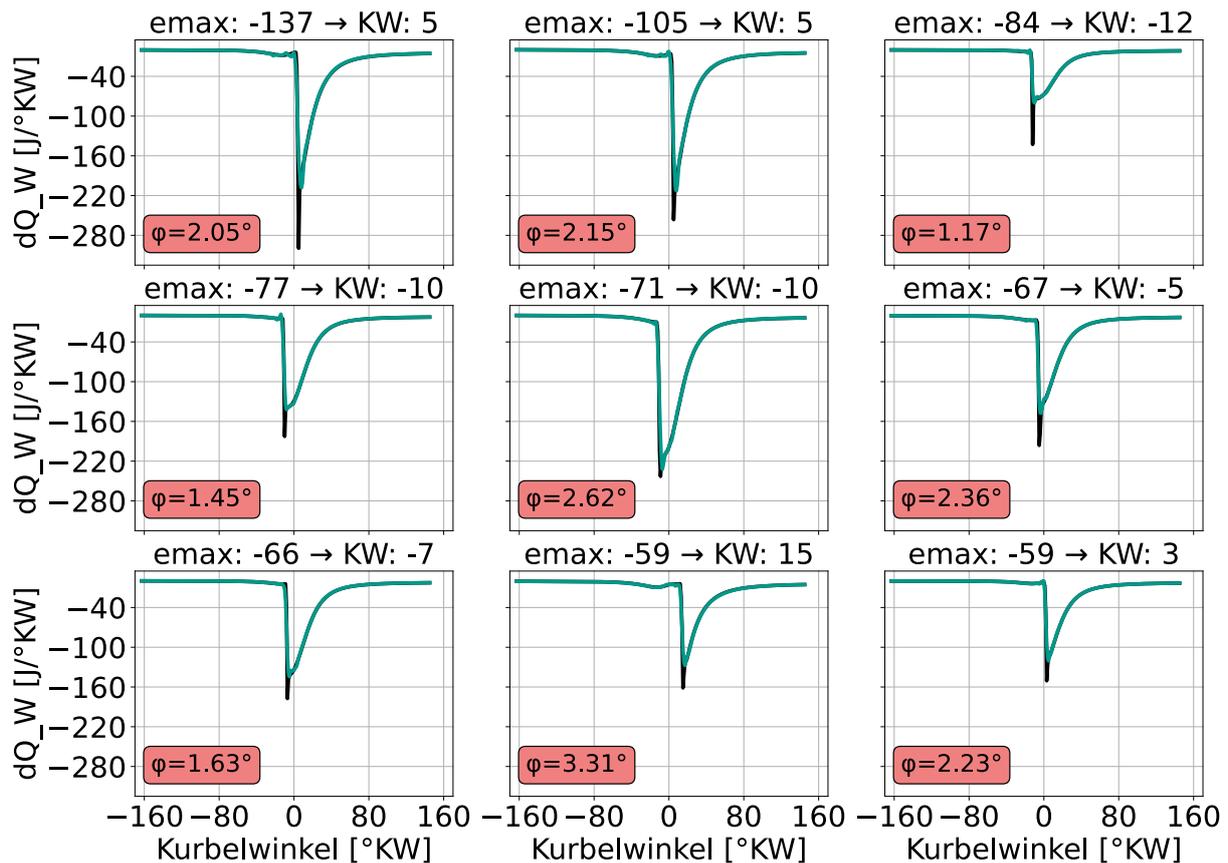


Abbildung 4-16: Größte maximale Fehler während des Belastungstests für die modellierten Wandwärmeverläufe I-IV

Grundsätzlich kann beobachtet werden, dass sich die vorgestellte Modellarchitektur im Zusammenhang mit den dazugehörigen Trainingsdaten nicht eignet, die physikalischen Verläufe mit extremen Gradienten abzubilden. Zwar handelt es sich bei den Phänomenen einer heftigen Verbrennung mit sehr kurzen Brenndauern um eher seltene Arbeitsspiele im Datensatz, welche weniger als ein Prozent darstellen. Da die Druckverläufe an diesen Stellen aber dennoch durchweg gut abgebildet werden, liegt ein Ansatz für mögliche weiterführende Forschungen eher in der Netzarchitektur und weniger in der Überarbeitung der Trainingsdaten.

Dafür sind zusätzliche Anpassungen an der Ausgangsschicht oder der Aktivierungsfunktion notwendig, um den möglich auftretenden Wertebereich zwischen zwei Kurbelwinkeln zu erhöhen. Ein logischer Ansatz wäre eine Verkleinerung der Schrittweite in der Ausgangsschicht auf $0.1 \text{ } ^\circ\text{KW}$, sodass sich die Änderung für jedes Ausgangsneuron um das Zehnfache verkleinert. Dies geht allerdings auch mit einer Verzehnfachung der Ausgangsneuronen und damit höherem Rechenaufwand einher, welche das Modelltraining und die Vorhersagen verlangsamen.



Im Gegensatz dazu eignet sich die Modellarchitektur gut, um komplexe Wechselwirkungen zwischen mehrdimensionalen Eingangsgrößen wiederzugeben und den qualitativen Verlauf von verschiedensten Ausgangsgrößen auch unter vielen unbekannt Zuständen beizubehalten. Dafür spricht einerseits, dass erst wahllos und zufällig initialisierte Eingangsparameter zu merklichen Abweichungen zwischen Simulation und Modellierung mit Hilfe der neuronalen Netzwerke führen. Andererseits sind bei der Betrachtung des maximalen Fehlers keine Ungenauigkeiten in der Berechnung durch das Zylindermodul, welche auch in kalibrierten Bereichen bis zu vier Prozent verglichen zu Prüfstandsmessungen betragen können, oder bedingt durch zyklische Schwankungen berücksichtigt. Diese werden vom Zylindermodul mit Parametrierung nach Kapitel 7 nicht abgebildet. Werden zuletzt die fünf Millionen zufällig vorhergesagten Betriebspunkte des Belastungstests als Referenz herangezogen und mit der Anzahl an Verläufen verrechnet, bei denen es zu Abweichungen in der Vorhersage über den definierten Grenzwerten kam, liegt der Gültigkeitsbereich der Modelle bei über 99.81 % im Vergleich zur Simulation.

Weiter dauern die fünf Millionen Berechnungen des Zylindermoduls für den Belastungstest auf 120 Intel Xeon Gold Prozessoren ca. drei Tage. Im Vergleich dazu benötigen die Vorhersagen mit Hilfe der neuronalen Netzwerke auf einem einzigen Intel Core i5-Prozessor 7.31 Minuten. Allerdings werden bei den Berechnungen des Zylindermoduls noch weitere Größen wie z.B. Stoffkonzentrationen oder gemittelte Einzelgrößen der Verbrennung ermittelt, welche in den Netzmodellen bislang nicht enthalten sind. Dies verdeutlicht das generelle Potential von neuronalen Netzwerken zur Beschleunigung von physikalischen Simulationen mit Hilfe des Zylindermoduls, welche bei zufriedenstellender Genauigkeit um den Faktor 500 liegt.



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



5 Modellierung transienter Vorgänge

Für eine Modellierung des gesamten Motorverhaltens oder zur Erstellung von virtuellen Sensoren mit Hilfe von neuronalen Netzwerken ist es notwendig, zeitübergreifende Vorgänge abzubilden. Dort weisen Ein- und Ausgänge transientes Verhalten auf und können nicht alleine über neue Parametrierungen abgebildet werden, sondern besitzen zusätzliche Zeitabhängigkeiten. Beispiele dafür in der Antriebsstrangentwicklung stellen mechanische oder thermische Trägheitsverhalten dar.

5.1 Datengenerierung

Zur Analyse und Berechnung der in diesem Kapitel betrachteten Verhaltensweisen eines 5X72DF-Motors der Firma WinGD wird ein geometrisch entsprechend aufgebautes Simulationsmodell in GT-Suite (s. Kapitel 2.4) verwendet, welches bereits mit Hilfe von Messdaten kalibriert ist.

Bei diesem Antriebssystem handelt es sich um einen langsam laufenden, nockenwellenlosen und direkt abgetriebenen zwei-Takt Großmotor, welcher sowohl mit verschiedenen Dieselsorten als auch mit Erdgas betrieben werden kann ([74], S.1-1). Außerdem lässt sich durch eine Umkehr der Drehrichtung ein Rückwärtslaufen des Motors ermöglichen. In Tabelle 5-1 sind die geometrischen Randbedingungen und Kennwerte des betrachteten Motors gelistet.



Bezeichnung	Wert
Hub	3086 mm
Bohrung	720 mm
ϵ	4.29
Zylinder	5
Nennleistung	2.606 kW/Zyl
Nenndrehzahl	71.5 1/min
Indizierter Nennmitteldruck	17.4 bar
Trockengewicht	4.81x106 kg
Länge	8.083 m
Höhe	13.630 m
Breite	4.780 m

Tabelle 5-1: Geometrische Randbedingungen und Kennwerte eines 5X72DF-Antriebsstrangs der Firma WinGD ([74], S.1-1)

5.1.1 Propellerkurve

Da der Motor für die Rotation an einer Schiffsschraube ausgelegt und durch den direkten Abtrieb kein Getriebe zwischengeschaltet ist, hängt die Leistung mit der Drehzahl im Betrieb ohne verstellbare Schraubenblätter (engl. Fixed Pitch Propeller - FPP) nach dem Propellergesetz entsprechend Gleichung (5.1) zusammen. Aufgrund der Wasserverdrängung steigt die notwendige Leistung beim Erhöhen der Drehzahl mit der dritten Potenz, ähnlich wie der Anstieg der Widerstandsleistung bei Fahrzeugen, bedingt durch den Luftwiderstand.

$$\frac{P}{P_N} = \left(\frac{n}{n_N}\right)^3 \quad (5.1)$$

Das bedeutet, die Leistung folgt der Drehzahl entlang einer Propellerkurve, wie in Abbildung 5-1 dargestellt. Dort ist die relative Leistung über der relativen Drehzahl bezogen auf den Nennbetriebspunkt aufgetragen, und es sind verschiedene mögliche Betriebsbereiche des Motors eingezeichnet.



Im Realbetrieb der Motoren können sich im FPP-Betrieb, z.B. im Notlauf (rot) oder bei starken Strömungen (gelb), weitere Zustände außerhalb der Propellerkurve ergeben ([74], S.2-7), welche aber nicht Teil dieser Arbeit sind. Bei Schiffsschrauben mit verstellbaren Schraubenblätter (engl. Controllable Pitch Propeller - CPP) lässt sich die Leistung bei konstanter Drehzahl verändern und der Motor dadurch auch in Bereichen außerhalb der Propellerkurve betreiben. Dieser Betrieb wird ebenfalls nicht in dieser Arbeit untersucht.

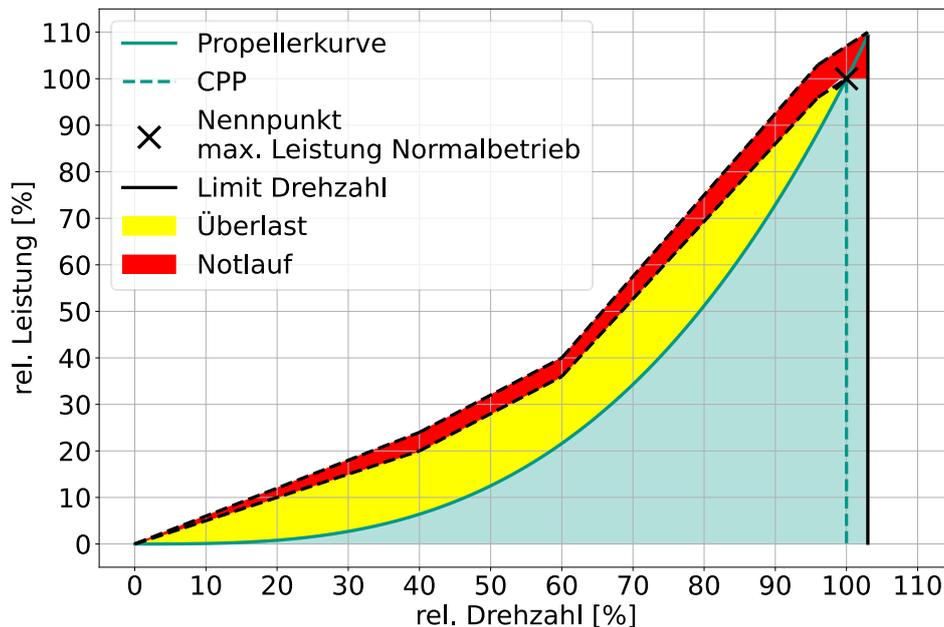


Abbildung 5-1: Exemplarische Propellerkurve für den 5X72DF im FPP-Betrieb (i.A. [74], S.2-7)

5.1.2 Zulässige Betriebsbereiche

Weil langsam laufende zwei-Takt Motoren mit langem Hub bei der Verbrennung hohen mechanischen und thermischen Trägheiten ausgesetzt sind, werden im Betriebshandbuch für verschiedene Leistungsbereiche zeitliche Grenzwerte angegeben, innerhalb denen die Leistung erhöht bzw. verringert werden darf ([74], S. 5-9). In Abbildung 5-2 ist die relative Leistung über der Dauer aufgetragen, innerhalb der sich Änderungen für das Erhöhen (links) oder das Verringern (rechts) der Leistung laut Handbuch bewegen dürfen. Unterhalb 25 % darf die Leistung innerhalb einer halben Minute beliebig verändert werden, dieser Bereich ist daher nicht im Diagramm enthalten. Für die anderen Bereiche sind in Tabelle 5-2 zusätzlich zu Abbildung 5-2 die entsprechenden Steigungen gelistet, da diese später ausschlaggebend für die Erzeugung der Trainingsdaten sind. Das Verringern der Leistungen kann grundsätzlich in halber Zeit der äquivalenten Leistungserhöhung erfolgen.

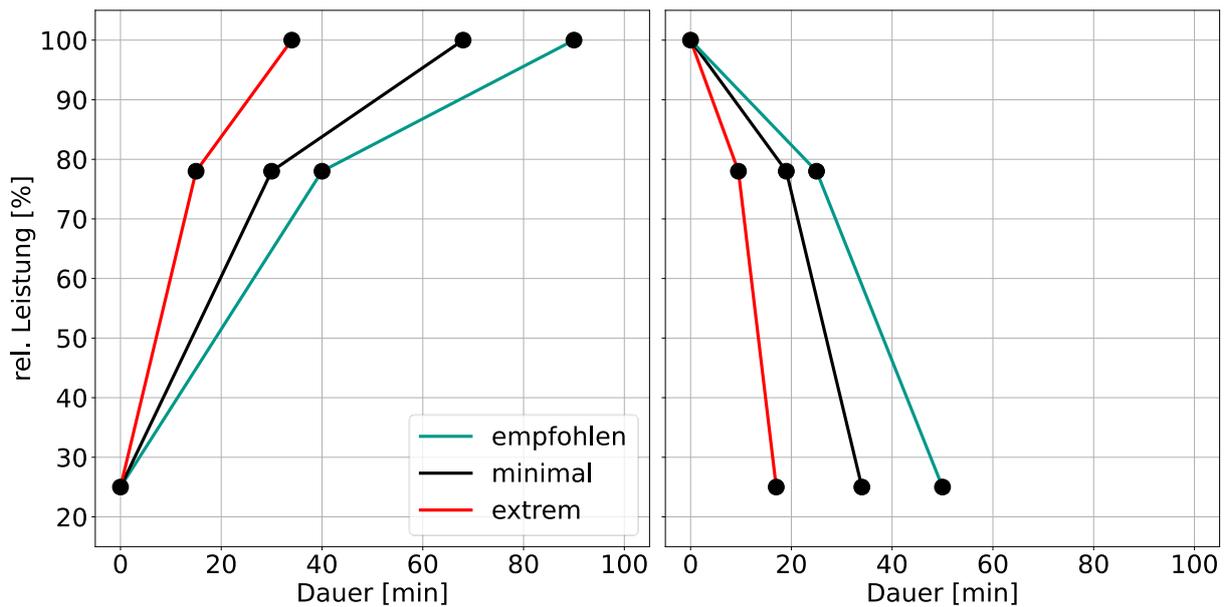


Abbildung 5-2: Zeitliche Grenzwerte für den 5X72DF bei einer Änderung der Leistung im FPP-Betrieb nach (i.A. [74],S. 5-9)

Leistungsbereich [%]	empfohlen [%/min]	minimal [%/min]	extrem [%/min]
<25	-	-	-
25-78	1.325	1.767	3.533
78-100	0.550	0.579	1.158
100-78	-1.1	-1.158	-2.316
78-25	-2.65	-3.534	-7.066

Tabelle 5-2: Zeitliche Grenzwerte für den 5X72DF bei einer Änderung der Leistung im FPP-Betrieb nach ([74],S. 5-9)

5.1.3 Simulierte Leistungsprofile

Basierung auf den Vorgaben von Tabelle 5-2 werden verschiedene Leistungsprofile zusammen mit der dazugehörigen Drehzahl nach Gleichung (2.2.10) erst mit zufälliger Steigung innerhalb vordefinierter Leistungssprünge zusammengestellt und anschließend das daraus resultierende Motorverhalten mit dem Vollmotormodell in GT-Suite berechnet. Vereinfacht wird dabei eine konstante Umgebungstemperatur von 20° C angenommen, um die Variationsmöglichkeiten der Eingangsparameter etwas einzugrenzen.



In der Ergebnisdatei können nach der Simulation wichtige Größen des Motorverhaltens bei zwei-Takt Motoren wie z.B. der Spüldruck, die Turboladerdrehzahl oder die Abgastemperatur ausgelesen werden, die während dem Fahrprofil simuliert werden.

Zum einen soll dabei der komplette Leistungsbereich zwischen 5-100 % abgebildet werden. Dieser ist nach unten durch die Leistung im Leerlauf, unterhalb der kein Motorbetrieb stattfinden kann, und nach oben durch die Leistung im Nennpunkt begrenzt. Zum anderen sollen gleiche Anteile von empfohlenen, minimalen und extremen Leistungssteigerungen bzw. -verringerungen enthalten sein. Ebenfalls wird ein gleichmäßig verteilter Stationärbetrieb über alle Leistungen berücksichtigt und dabei beachtet, dass jedes transiente Profil aus einem stationären Zustand beginnt. In Abbildung 5-3 sind exemplarisch einige der generierten Leistungsprofile abgebildet.

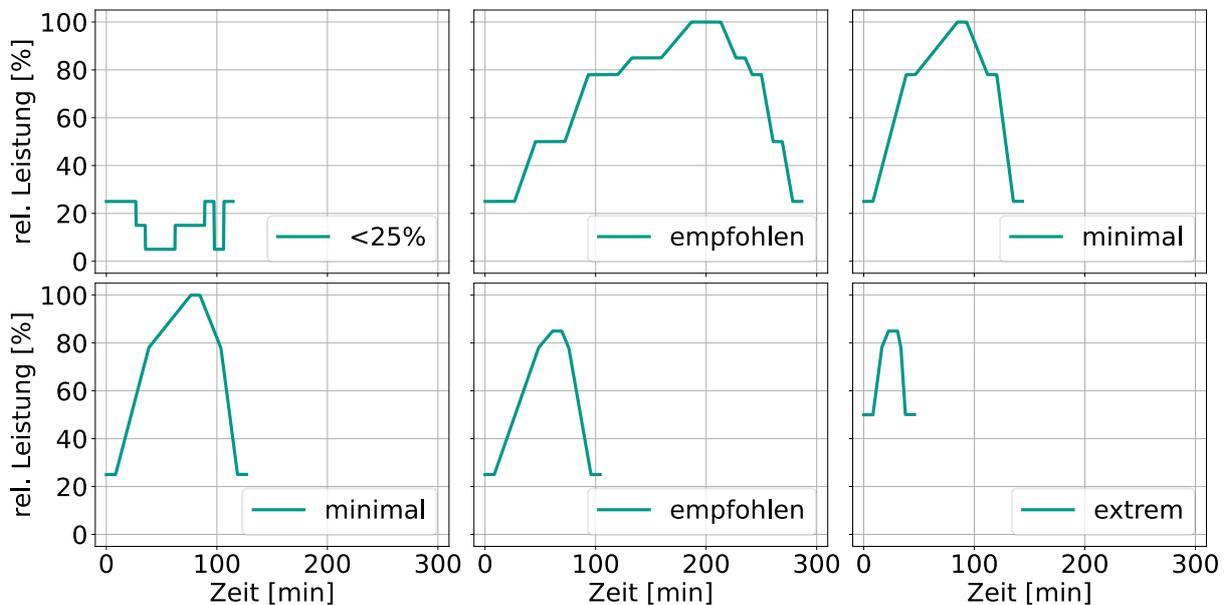


Abbildung 5-3: Auszug aus den generierten Leistungsprofilen entsprechend Tabelle 5-2

Auf hoher See können neben Strömungen auch größere Wellen einen Einfluss auf die Leistung haben. Dabei erhöht sich die Motorleistung beim Anstieg kurzzeitig, um eine eintreffende Wellenfront zu überwinden, und nimmt anschließend wieder ab ([72], S.38). Dieses Phänomen wird auch als „Waveloading“ bezeichnet und wird in den Trainingsdaten zusätzlich berücksichtigt, um die Allgemeingültigkeit des Modells zu erhöhen.



Dafür werden in verschiedenen stationären Leistungsbereichen zufällig initialisierte und sinusförmige Schwankungen überlagert, die sich in ihrer Amplitude und Frequenz leicht unterscheiden. Abbildung 5-4 zeigt exemplarisch einige solcher Leistungsverläufe, welche ein derartiges Ereignis veranschaulichen soll.

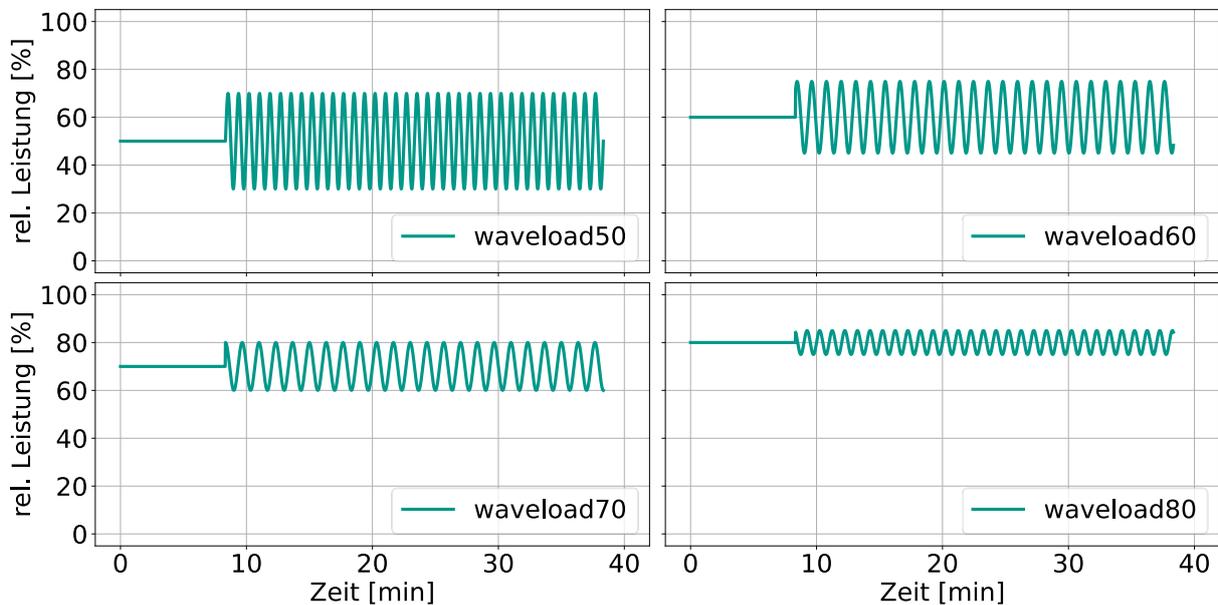


Abbildung 5-4: Auszug aus den Daten für die Simulation des Waveloadings bei unterschiedlicher Leistung, Amplitude und Frequenz

5.1.4 Datenextraktion

Als letztes werden die Datensätze der verschiedenen Simulationen automatisiert zugeschnitten, damit das Modell im Anschluss keine Fahrprofile, sondern die generellen Zusammenhänge zwischen einer Leistungsänderung und dem daraus resultierenden Motorverhalten erlernt. Später bei den Vorhersagen können so beliebige Fahrprofile innerhalb der Modellierungsgrenzen zusammengestellt und mit Hilfe des neuronalen Netzwerks abgebildet werden. Zusätzlich hat dieser Ansatz den Vorteil, dass auch Messdaten für die Modellentwicklung genutzt werden können, da sich z.B. aus sehr langen Fahrprofilen, teilweise über mehrere Monate oder Jahre, die notwendigen Daten ausschneiden lassen. Das hilft bei der Datengewinnung, falls kein kalibriertes Simulationsmodell zur Verfügung steht (vgl. Kapitel 5).

Dafür wird mit Hilfe mathematischer Funktion (vgl. Kapitel 2.2.4) herausgefunden, wann ein Fahrmanöver startet bzw. endet, und dieser Bereich aus dem Datensatz extrahiert. Exemplarisch ist dieser Ansatz in Abbildung 5-5 veranschaulicht.



Am Ende der Datenvorbereitung steht ein Trainingsdatensatz mit über 250 verschiedenen Fahrscenarien bei unterschiedlichen stationären Konstantleistungen und transienten Leistungsänderungen für die Modellbildung zur Verfügung, welcher in allen Fahrbereichen in etwa ausgeglichene Mengen an Datenpunkten enthält.

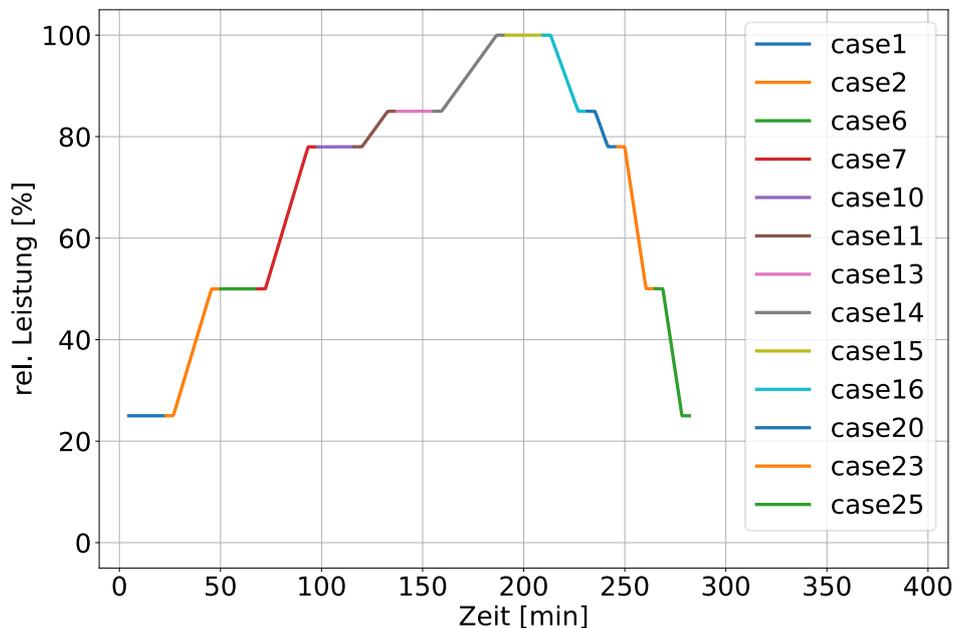


Abbildung 5-5: Auszug der Extraktion von verschiedenen Fahrmanöver aus den simulierten Leistungsprofilen

5.2 Modellierung II

Als nächstes werden die Simulationsdaten der berechneten Fahrmanöver in ein neuronales Netzmodell überführt. Dafür sind bei der Modellierung in erster Linie transientfähige Netzarchitekturen notwendig, die es ermöglichen, Informationen zeitübergreifend zu speichern (vgl. Kapitel 2.1.6).

5.2.1 Datenaufteilung

Von den generierten Fahrprofilen werden nach den Erkenntnissen aus dem vorangegangenen Kapitel und entsprechend Tabelle 5-3 50% der Fahrprofile für das Training verwendet. Mit 20% der Fahrprofile wird der Optimierungsvorgang während des Trainings validiert, und 30% der Daten stehen in Kapitel 5.2.5 - 5.2.10 zur Auswertung zur Verfügung.



Anteil	Größe [%]
Training	50
Validierung	20
Test	30

Tabelle 5-3: Aufteilung des Ausgangsdatensatzes für die Modellierung II

5.2.2 Netzarchitektur

In Abbildung 5-6 ist das in diesem Kapitel verwendete Netzlayout schematisch dargestellt, welches aus Eingangsschicht, drei hintereinandergeschalteten Schichten L1-L3 und einer Ausgangsschicht besteht und an vorangegangene Untersuchungen ([16]) angelehnt ist. Im Vergleich zu Abbildung 4-7 werden hier anstatt kontinuierlicher Neuronen LSTM-Zellen verwendet, bei denen die Weitergabe von Zeitreihen und Zellstatus aktiviert ist (vgl. Kapitel 2.1.6). Das bedeutet, sowohl das Kurzzeit- als auch das Langzeitgedächtnis werden in Folge des Trainings an die Simulationsdaten angepasst. Die Notwendigkeit dafür wird im folgenden Unterkapitel beschrieben. Um das Netzwerk auf Grafikkarten trainieren zu können, wird die Aktivierungsfunktion Tangens Hyperbolicus gewählt [69]. Die Lernrate, die Fehlerfunktion und das Abbruchkriterium werden zunächst an Kapitel 4 angelehnt. Die weitere Parametrierung wird entsprechend Tabelle 5-4 vorgenommen.

Anders als bei der kurbelwinkelaufgelösten Modellierung, bei der jeder Verlaufsgröße ein eigenes Modell zugeordnet ist, wird in diesem Kapitel getestet, alle Ausgangsgrößen simultan in einem Modell abzubilden. Das hat den Vorteil, dass Informationen über die Wechselwirkungen zwischen verschiedenen Einzelgrößen, wie z.B. der Abgastemperatur und Turboladerdrehzahl, vom Modell erkannt und erlernt werden können.

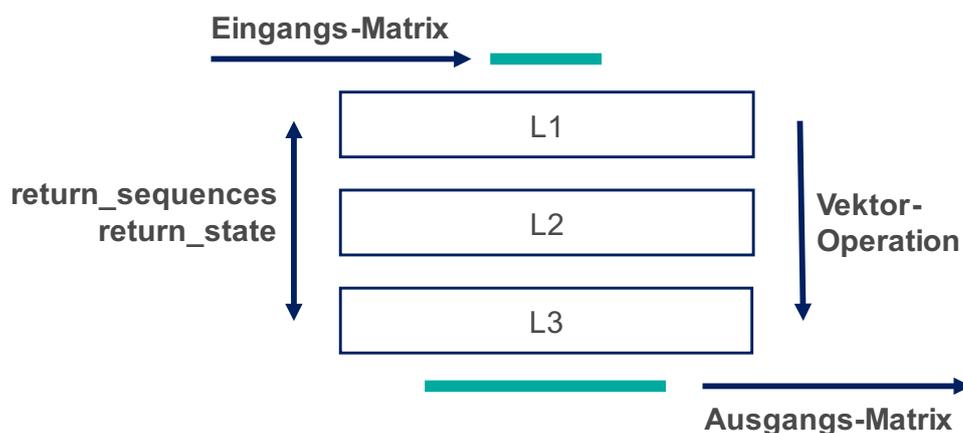


Abbildung 5-6: Netzlayout für die zeitaufgelöste Modellierung II



Parameter	Wert
L1	200 LSTM
L2	200 LSTM
L3	200 LSTM
Ausgang	3 Neuronen
Aktivierung	TanH
Fehlerfunktion	Huber
Lernrate	10-3
Zellstatus	aktiv
Zeitreihen	aktiv
Abbruchbedingung	25 Epochen
Eingangsdimension	(180x2)
Ausgangsdimension	(1x3)

Tabelle 5-4: Netzparameter für die zeitaufgelöste Modellierung II

5.2.3 Zeitabhängigkeit

Im Gegensatz zu stationären Zuständen, bei denen Eingangsgrößen feste Ausgangsgrößen zugeordnet werden können, sind bei zeitaufgelösten Vorgängen Informationen der Vergangenheit notwendig, um aktuelle oder in der Zukunft liegende Zustände zu beschreiben.

Als Beispiel dafür wird Abbildung 5-7 betrachtet. Dort sind die im Simulationsmodell berechnete Motorleistung und -Drehzahl, die Drehzahl des Abgasturboladers, die Abgastemperatur und der Spüldruck im Brennraum für eine Leistungsänderung von 25-100% und 100-25% innerhalb des minimalen, nach Handbuch empfohlenen Zeitraums aufgetragen. Alle Größen sind in diesem Kapitel grundsätzlich relativiert, da die Absolutwerte der Geheimhaltung unterliegen.

Die Abgastemperatur weist in den ersten und letzten zehn Minuten ein stark transientes Verhalten auf (grüne Kreise), da im Motor Zusatzgebläse für den unteren Teillastbetrieb erst zugeschaltet und anschließend wieder abgeschaltet werden. Abhängig davon, ob dieser Vorgang bereits stattgefunden hat oder nicht, und wie schnell der entsprechende Leistungsbereich durchlaufen wird, kann die Abgastemperatur für dieselbe Leistung und Drehzahl unterschiedliche Werte annehmen.



Aufgrund dessen ist der Informationsfluss, wie sich die Leistung in den vorangegangenen Minuten geändert hat, ausschlaggebend für die Bestimmung der aktuellen Abgastemperatur, und es wird der Zellstatus in der Netzarchitektur für die Modellierungen benötigt.

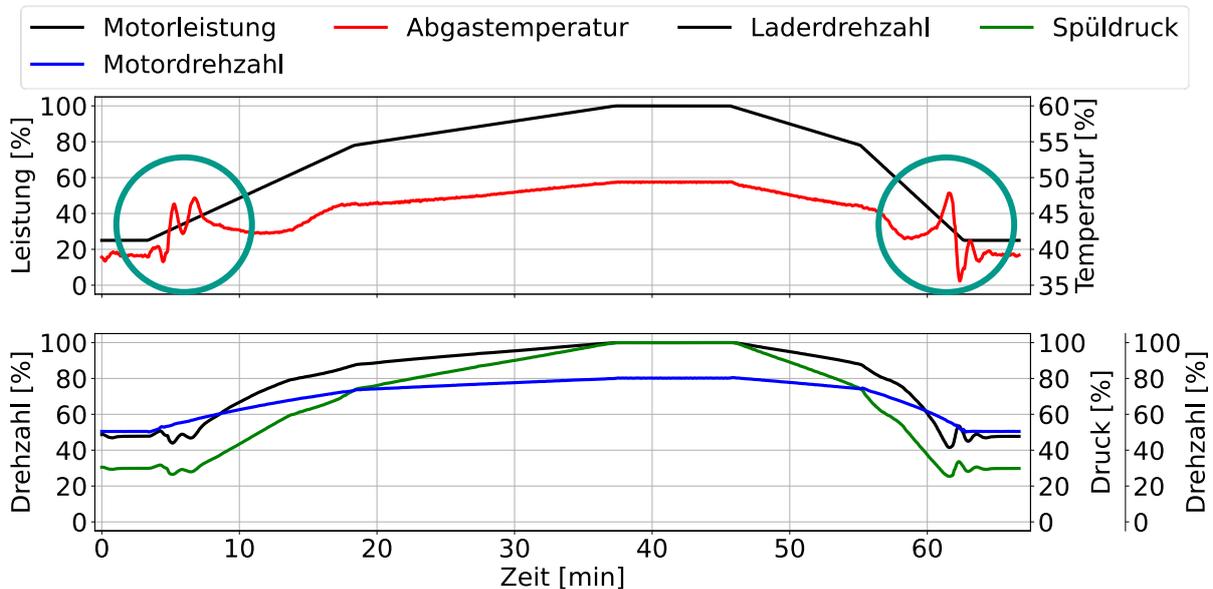


Abbildung 5-7: Transientes Verhalten der Abgastemperatur in der Simulation

5.2.4 Modellierungsschema

Die Simulationsdaten werden daher nach einem besonderen Schema modelliert, welches in Abbildung 5-8 dargestellt und an [70] angelehnt ist. Zuerst werden die States der LSTM-Zellen zufällig initialisiert. Basierend auf einem festen Zeitfenster von Leistung und Drehzahl, werden die Drehzahl des Abgasturboladers, die Abgastemperatur und der Spüldruck zum darauffolgenden Zeitpunkt vorhergesagt und die Gewichtungen daraufhin optimiert. Anschließend wird das Zeitfenster um einen Überlappungszeitschritt nach vorne verschoben und der Vorgang solange wiederholt, bis das vollständige Fahrprofil durchiteriert ist.

Am Ende des Fahrprofils werden die States zurückgesetzt, damit keine Informationen von vorangegangenen Fahrprofilen, welche z.B. mit einer anderen Leistung enden oder starten, übernommen werden. So soll erreicht werden, dass transiente Effekte im Motor, welche z.B. durch Regelungsstrategien im Steuergerät oder durch das Zu- und Abschalten von Nebenaggregaten auftreten, vom Modell erlernt werden. Später bei den Vorhersagen bedeutet das, der erste abgebildete Wert steht ebenfalls erst nach einer Initialisierungsphase in Zeitfensterlänge zur Verfügung.

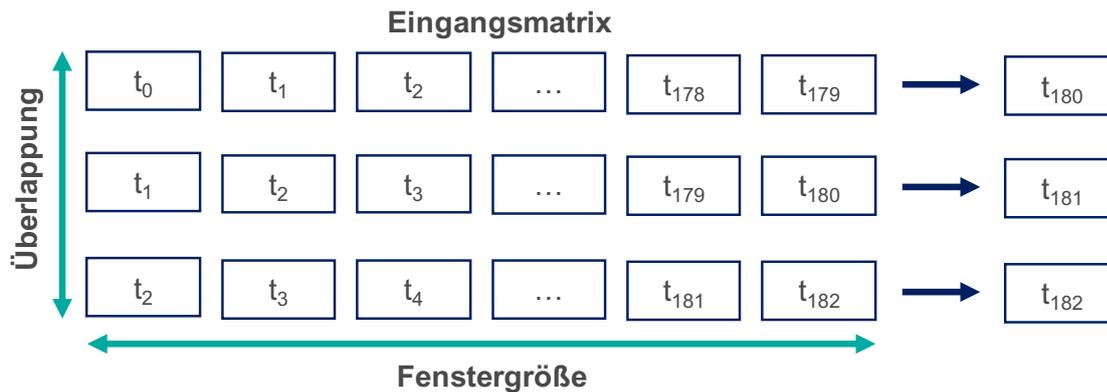


Abbildung 5-8: Modellierungsschema für die zeitaufgelöste Modellierung II

5.2.5 Leistungsänderung 25-100%

Mit einer parametrisierten Lernrate von 10-3 im Modellierungsversuch II-I nach Tabelle 5-4 wird das Training nach 27 Epochen entsprechend dem Abbruchkriterium von 25 Durchläufen ohne merklichen Lernerfolg beendet. Das bedeutet, die Gewichtungen des neuronalen Netzwerks wurden anfangs zu schnell und zu stark verändert, sodass am Ende keine zufriedenstellende Näherungslösung für die Verlaufsgrößen der Fahrprofile durch den Optimierer gefunden werden konnte bzw. das Training nicht konvergiert ist (vgl. Kapitel 2.1.5).

Für weiterführende Untersuchungen sind in Abbildung 5-9 exemplarisch die Berechnungen eines ungesehenen Fahrprofilprofils aus den Testdaten von 25-100% Leistung mit zufällig initialisierter Steigung innerhalb der Modellierungsgrenzen, abgebildet. Zusätzlich sind die Vorhersagen des erstellten Modells mit Parametrierung II-I aufgetragen. Es lässt sich erkennen, dass die transienten Motorgrößen aufgrund der zu hohen Lernrate interessanterweise nur über einen binären Anstieg vom Modell angenähert werden konnten. Diese einfachste Minimallösung resultiert natürlich in hohen durchschnittlichen und maximalen Fehlern bei der Abbildung der Abgastemperatur, des Spüldrucks und der Laderdrehzahl, welche bis zu 11 bzw. 34 % betragen und in Tabelle 5-5 gelistet sind.

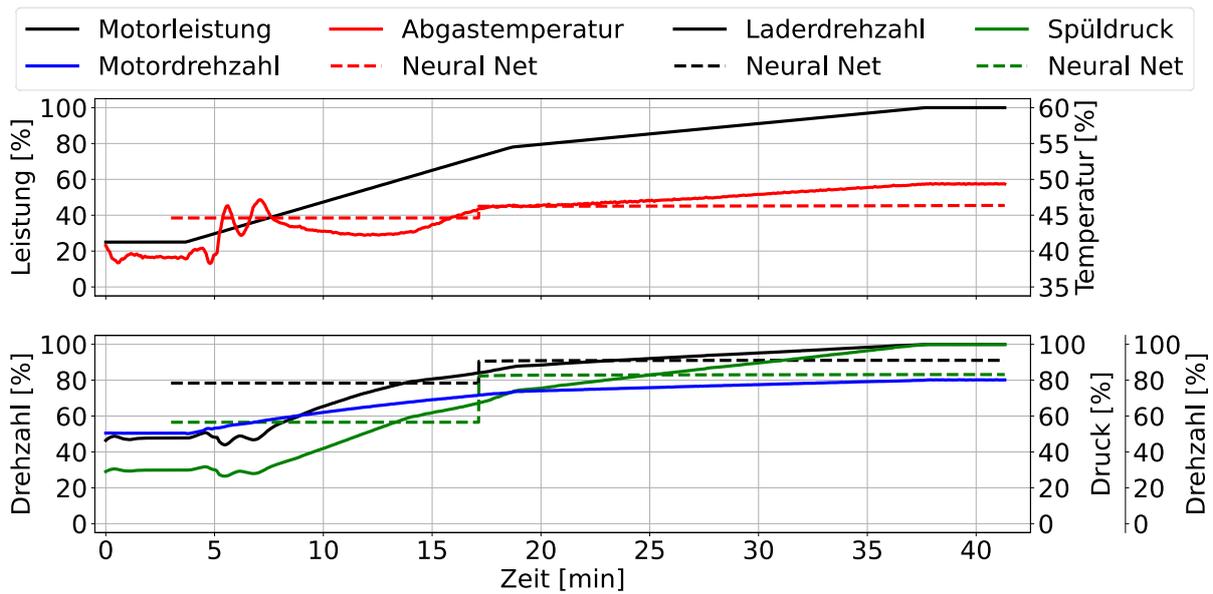


Abbildung 5-9: Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-I

Verlauf	mae [%]	e_max [%]
Abgastemperatur	4.726	6.349
Laderdrehzahl	8.572	34.436
Spüldruck	11.312	30.074

Tabelle 5-5: Durchschnittliche und maximale Fehler bei der Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-I

5.2.6 Leistungsänderung 25-100%

Mit einer Reduzierung der Lernrate auf 10^{-4} im Modellierungsversuch II-II werden die Zusammenhänge im Antriebstrang insgesamt mit 132 Epochen erlernt, bevor das Training entsprechend dem Abbruchkriterium beendet wird. Abbildung 5-10 zeigt erneut die Berechnungen und Vorhersagen der transienten Motorgrößen bei derselben Leistungsänderung mit gleicher Steigung wie in Abbildung 5-9. Dort wird deutlich, dass das Modell nun wesentlich präziser in der Lage ist, die zeitlich veränderlichen Verläufe im entsprechenden Leistungsbereich abzubilden. Das spiegelt sich auch in den geringeren durchschnittlichen und maximalen Fehlern aus Tabelle 5-6 wider. Diese liegen für fast alle Verläufe unter zwei Prozent.



Lediglich bei der Abgastemperatur wird gegen Ende beim Erreichen der Nennleistung eine merkliche Abweichung festgestellt. Diese wird dort im Vergleich zur Berechnung etwas zu gering modelliert.

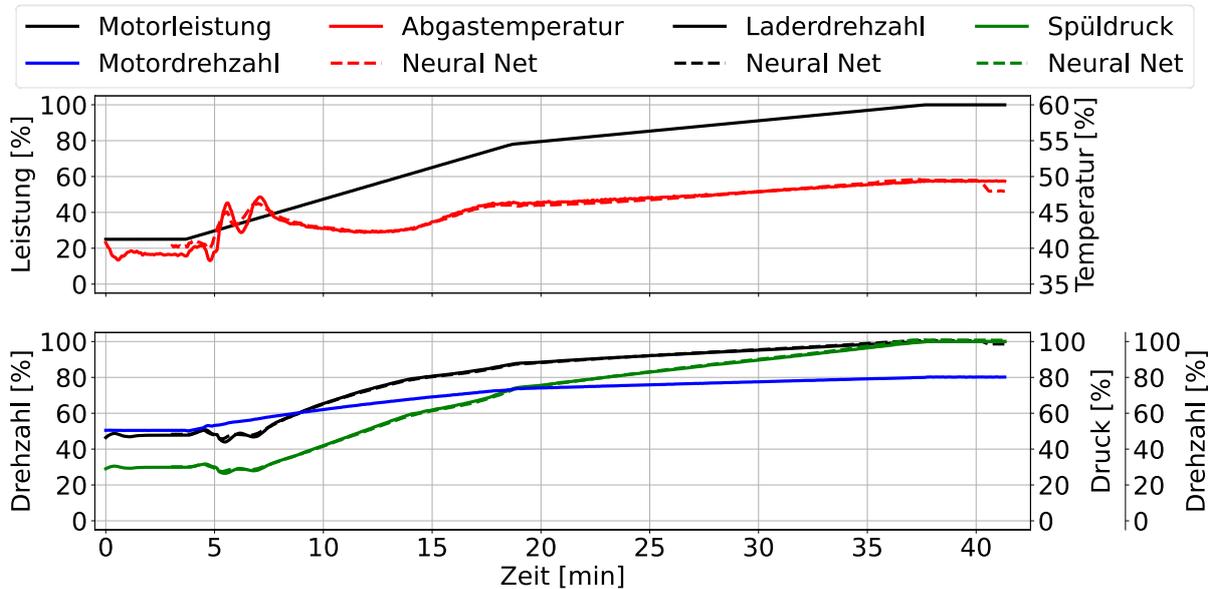


Abbildung 5-10: Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-II

Verlauf	mae [%]	e_max [%]
Abgastemperatur	0.316	2.942
Laderdrehzahl	0.411	1.567
Spüldruck	0.529	1.598

Tabelle 5-6: Durchschnittliche und maximale Fehler bei der Modellierung einer transienten Leistungsänderung von 25-100% mit Parametrierung II-II

5.2.7 Leistungsänderung 5-25%

Als nächstes wird exemplarisch eine bisher nicht für die Modellierung verwendete Leistungsänderung mit zufällig initialisierter Steigung im niederen Leistungsbereich betrachtet, welche z.B. beim Verlassen des Hafens auftreten kann. Dafür sind in Abbildung 5-11 die transienten Motorgrößen für eine schnelle Leistungsänderung von 5-25% abgebildet. Aufgrund der hohen Steigung der Leistungsänderung, welche sich nah an der zulässigen Grenze befindet, überdreht der Motor kurzzeitig und die Abgastemperatur steigt infolgedessen stark an.



Dieses hochtransiente Motorverhalten wird vom Modell qualitativ und quantitativ zufriedenstellend abgebildet, da die durchschnittlichen und maximalen Fehler für alle Größen unter zwei Prozent liegen, wie in Tabelle 5-7 gelistet.

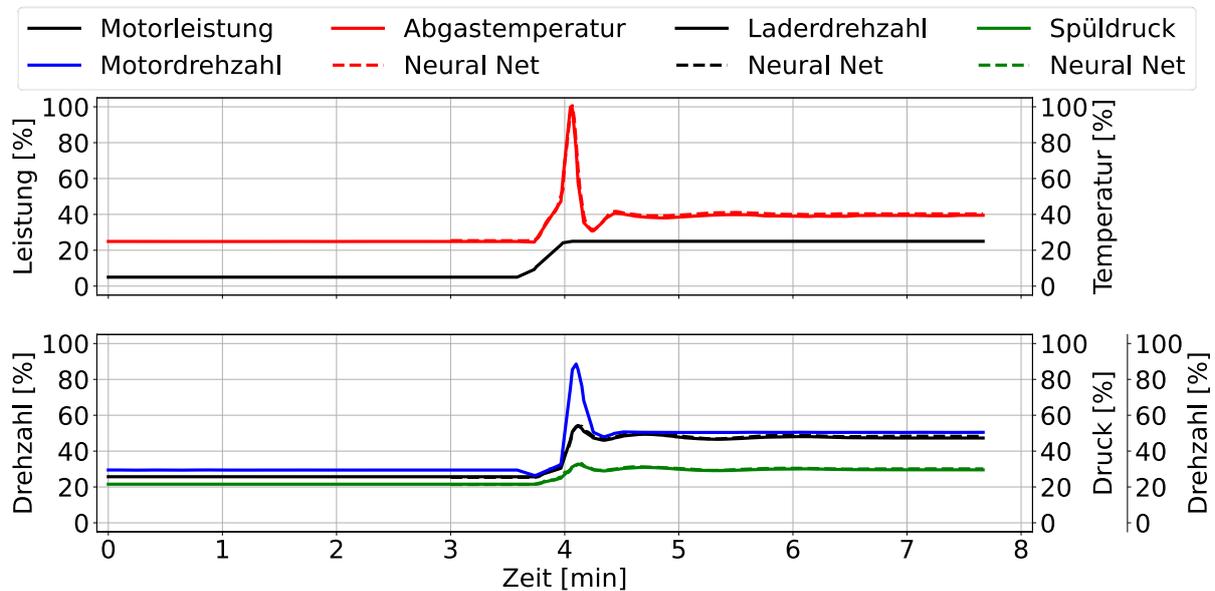


Abbildung 5-11: Modellierung einer transienten Leistungsänderung von 5-25% mit Parametrierung II-II

Verlauf	mae [%]	e_max [%]
Abgastemperatur	1.023	1.512
Laderdrehzahl	0.682	1.121
Spüldruck	0.365	1.282

Tabelle 5-7: Durchschnittliche und maximale Fehler bei der Modellierung einer transienten Leistungsänderung von 5-25% mit Parametrierung II-II

5.2.8 Waveloading

Als nächstes wird der Sonderfall des Waveloadings betrachtet, welcher beim Befahren von offenen Gewässern auftreten kann. Dabei schwankt die Leistung infolge des Wellengangs um einen Sollwert, und die transienten Motorgrößen ändern sich dementsprechend. Abbildung 5-12 zeigt eine zufällig initialisierte und sinusförmige Änderung der Leistung mit einer Periodendauer von 50 Sekunden und einer Amplitude von $\pm 15\%$ um einen Mittelwert von 50% Leistung.



Sowohl der qualitative als auch der quantitative Verlauf sind vom Modell zufriedenstellend abgebildet, da die Kurven nahezu deckungsgleich sind. Auch die durchschnittlichen und maximalen Fehler, welche in Tabelle 5-8 zu finden sind, liegen durchgehend unter zwei Prozent.

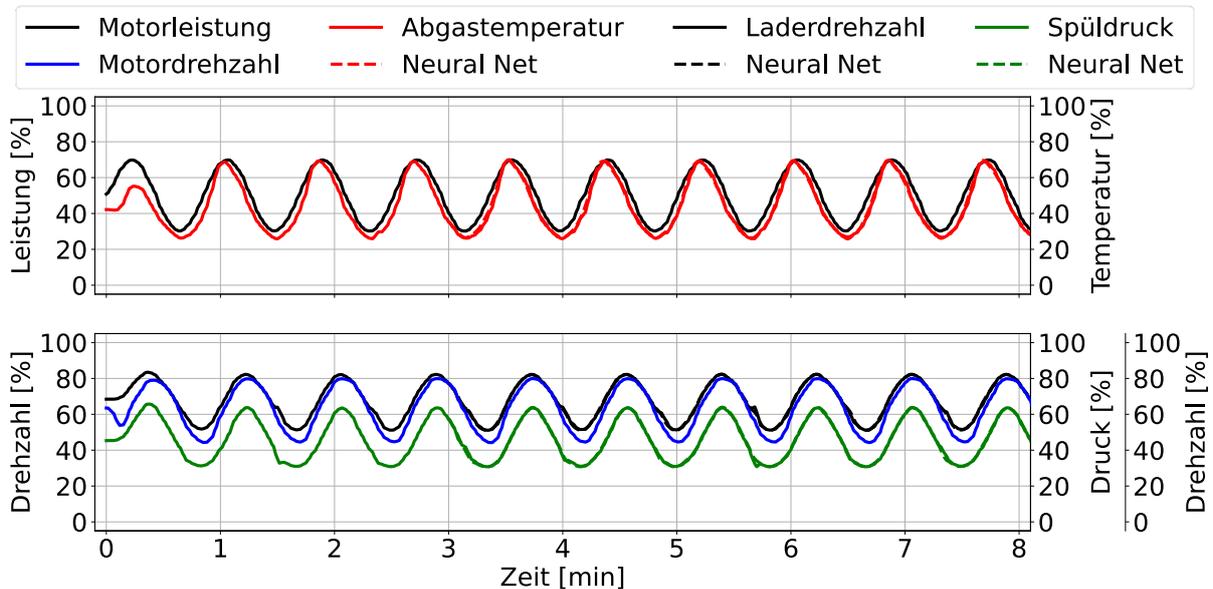


Abbildung 5-12: Modellierung von Waveloading bei 50% Leistung mit Parametrierung II-II

Verlauf	mae [%]	e_max [%]
Abgastemperatur	0.790	1.70
Laderdrehzahl	0.396	1.011
Spülndruck	0.454	1.376

Tabelle 5-8: Durchschnittliche und maximale Fehler während der Modellierung von Waveloading bei 50% Leistung mit Parametrierung II-II

5.2.9 Vollständiges Leistungsprofil

Als letztes wird ein vollständiges Fahrprofil mit zufällig initialisierten und unterschiedlich steigenden bzw. fallenden Leistungsänderungen betrachtet, welches weder in den Trainings-, Validierungs- oder Testdaten zu finden ist. Ausgehend von 25% wird dabei die Leistung in Abbildung 5-13 sukzessive bis zur Nennleistung erhöht und anschließend wieder reduziert. Die Initialisierungsphase der Simulation zu Beginn, bei der die Motorgrößen stark schwanken und erst eingeregelt werden müssen, werden bei der Auswertung nicht berücksichtigt



Für die Laderdrehzahl und den Spüldruck werden die Verläufe zufriedenstellend abgebildet, da sowohl der Kurvenverlauf als auch die Absolutwerte eine Abweichung von unter zwei Prozent bei der Modellierung besitzen (vgl. Tabelle 5-9). Bei der Abgastemperatur ist auch hier, wie in Abbildung 5-10 bereits erwähnt, eine merkliche Abweichung während des Betriebs bei konstanter Nennleistung zu erkennen. Dort liegt die Temperatur in der Modellierung etwa vier Prozent unter der der Berechnung.

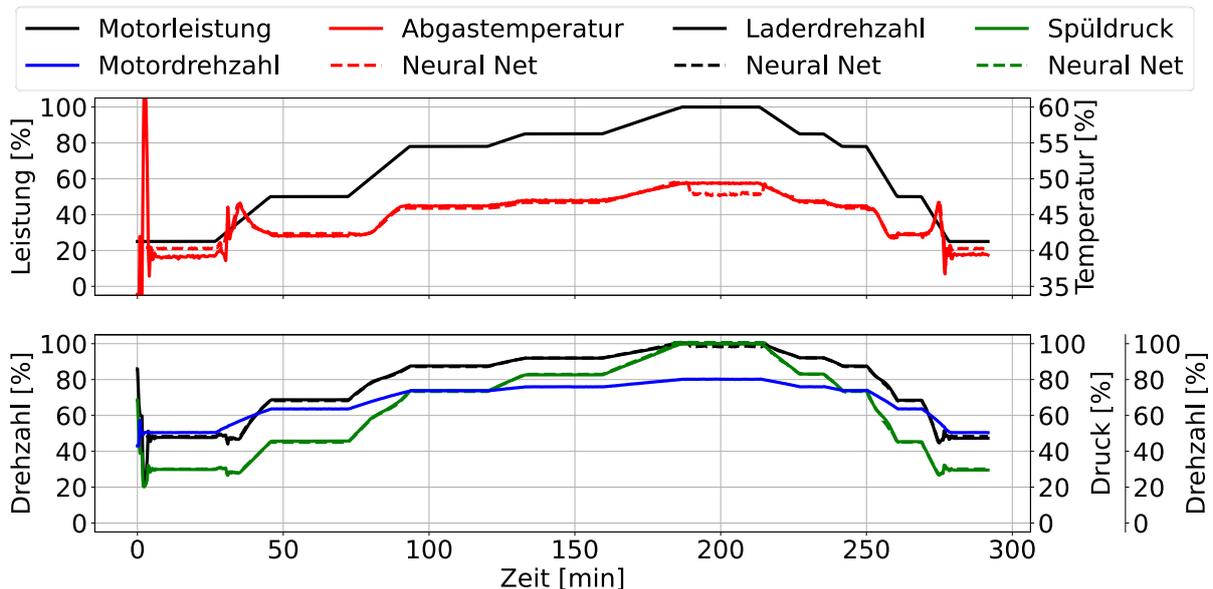


Abbildung 5-13: Modellierung eines vollständigen Fahrprofils mit Parametrierung II-II

Verlauf	mae [%]	e_max [%]
Abgastemperatur	0.493	3.67
Laderdrehzahl	0.510	1.540
Spüldruck	0.496	1.015

Tabelle 5-9: Durchschnittliche und maximale Fehler bei der Modellierung eines vollständigen Fahrprofils mit Parametrierung II-II

5.2.10 Transferlearning II-III

Abschließend wird die Lernrate noch einmal auf 10⁻⁸ verringert. Trainiert wird das bestehende Modell aus II-II im nächsten Modellierungsversuch II-III allerdings ausschließlich mit Daten im Bereich der Nennleistung und der niedrigen Leistungsbereichen. Damit soll erreicht werden, dass die Temperaturverläufe dort noch genauer abgebildet werden.



Nach fünf Epochen wird das Training beendet, um die bereits zufriedenstellend abgebildeten Größen nicht wesentlich zu beeinflussen. Dieses Vorgehen kann auch als Feinabstimmung bezeichnet werden und wird bei industriell eingesetzten neuronalen Netzwerken ebenfalls angewendet [10].

Abbildung 5-14 veranschaulicht erneut das vorangegangene Fahrprofil und die dazugehörigen Vorhersagen aus dem Modellierungsversuch II-III. Es zeigt sich, dass die Abweichung der Temperatur während des Betriebs bei konstanter Nennleistung nun deutlich geringer als in Abbildung 5-13 ausfällt. Auch der durchschnittliche und maximale Fehler konnten reduziert werden (vgl. Tabelle 5-10). Die Fehler bei der Abbildung des Spüldrucks und der Laderdrehzahl haben sich dabei zwar leicht erhöht, liegen aber nach wie vor in einem zufriedenstellenden Bereich. Das spricht dafür, das Training bei der Feinabstimmung sehr kurz zu halten und dafür eine sehr geringe Lernrate zu verwenden.

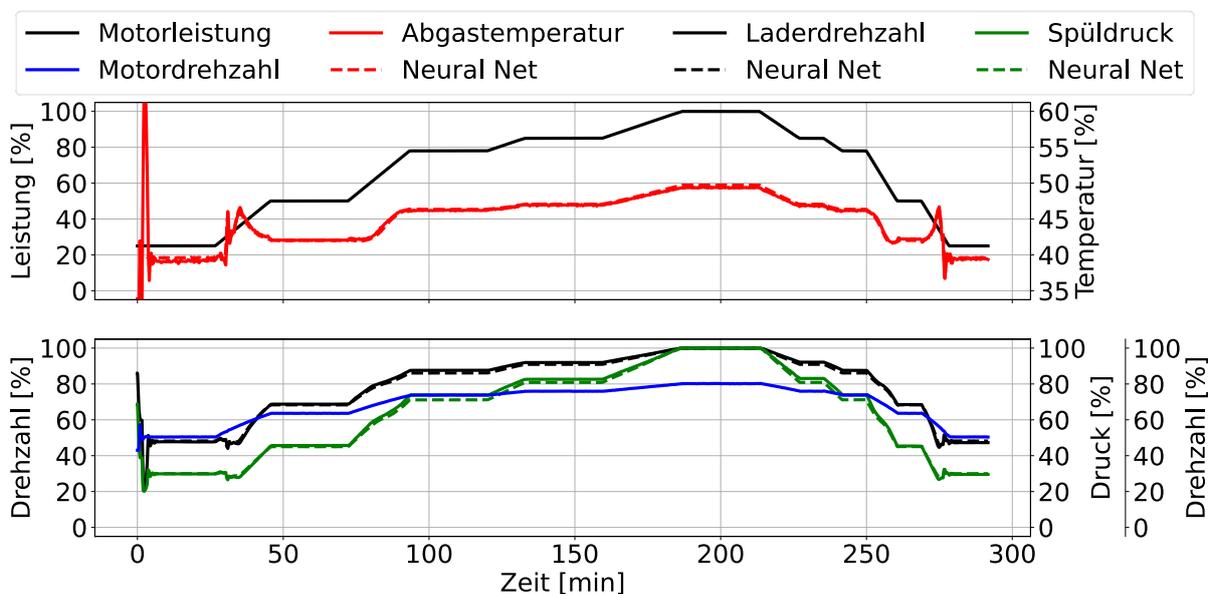


Abbildung 5-14: Modellierung eines vollständigen Fahrprofils mit Parametrierung II-III

Verlauf	mae [%]	e_max [%]
Abgastemperatur	0.220	1.251
Laderdrehzahl	0.777	1.012
Spüldruck	1.184	1.182

Tabelle 5-10: Durchschnittliche und maximale Fehler bei der Modellierung eines vollständigen Fahrprofils mit Parametrierung II-III



Grundsätzlich kann gesagt werden, dass sich die vorgestellte Methodik zur Datenerzeugung und die Modellarchitektur gut eignen, um die prinzipiellen Zusammenhänge im Motor mit Hilfe von neuronalen Netzwerken abzubilden. Im Gegensatz zum Erlernen von vollständigen Fahrprofilen werden hier nur ausgewählte Leistungsänderungen verwendet und somit eine höhere Generalisierung erreicht, da die Datenpunkte im Training zum einen den vollständigen Leistungsbereich enthalten und diesen zum anderen homogen verteilt abdecken.

Auch eignet sich die Netzarchitektur gut, um mehrere Ausgangsgrößen simultan abzubilden. Sowohl die Abgastemperatur als auch der Spüldruck und die Laderdrehzahl konnten zufriedenstellen modelliert werden. Hier sind jedoch Grenzen gesetzt. Im Vergleich zum GT-Modell, bei dem mehrere hunderte Größen gleichzeitig berechnet werden, sind im neuronalen Netzmodell nur einige charakteristische Kennwerte enthalten. Das ist während der Methodenentwicklung von Vorteil, da sich der Rechenaufwand mit wenigen Stunden Modelltraining in einem überschaubaren Bereich befindet. Für die Überführung eines vollständigen GT-Modells mit Luftpfad, Brennraum und Ladungswechsel sind allerdings aufwändigere Netzarchitekturen mit zusätzlichen Ein- und Ausgangsgrößen notwendig, da bei den Berechnungen des GT-Modells noch weitere Motorgrößen, wie z.B. Drehmomente oder Rohrdrücke und -Temperaturen, ermittelt werden, die im neuronalen Netzmodell bislang nicht enthalten sind.

Die Berechnung des fünf-stündigen Fahrprofils mit dem GT-Modell dauert auf einem Intel i5 Prozessor ca. 8 Stunden. Im Vergleich dazu benötigt die Vorhersage mit Hilfe des neuronalen Netzwerks auf demselben Prozessor 3.72 Minuten. Das resultiert, bei einer Genauigkeit von unter zwei Prozent Abweichung zur Simulation, in einer Rechenbeschleunigung um den Faktor 150.



6 Detektion von Unregelmäßigkeiten

Moderne Antriebssysteme werden immer komplexer. Langsam laufende zwei-Takt Motoren werden beispielsweise aufgrund der geringeren Schadstoffemissionen bei der Verbrennung von reinen Kohlenwasserstoffen ohne Schwefelanteil für einen Betrieb mit Erdgas anstelle von Schweröl umgerüstet ([77], S.1). Sobald der Motor warmgelaufen ist und die Verbrennung stabil abläuft, kann dabei das Brennverfahren und die umgesetzte Kraftstoffart gewechselt werden.

Dafür sind aufwändige Änderungen im Einspritzsystem notwendig. Auch die hinterlegte Regelungsstrategie im Steuergerät wird entsprechend umfangreicher, da es sich bei der Verbrennung von Erdgas im Gegensatz zur Dieselerbrennung um ein vorgemischtes und fremdgezündetes Brennverfahren handelt. Diese steigende Komplexität geht zwangsläufig mit einer erhöhten Fehleranfälligkeit einher. Allein durch die höhere Anzahl von notwendigen Bauteilen und Messsignalen, welche für die mechanische Umsetzung und deren entsprechenden Regelungsstrategien zum Einsatz kommen, erweitern sich die Möglichkeiten für Ausfälle und unerwünschte Auffälligkeiten im Gesamtsystem. Zudem lassen steigender Kostendruck und immer kürzere Innovationsintervalle die Entwicklungszeiten stetig sinken. Konkret bedeutet das, immer aufwändigere Systeme müssen in immer kürzerer Zeit entwickelt werden.

Für den globalen Warentransport liegt die Schifffahrt, gemessen an jährlich transportierten Kilogramm Gütern, vor der Eisenbahn und dem Flugzeug auf Platz eins ([62], S.5). Ausfälle wirken sich hier drastisch auf die weltweiten Lieferketten aus, welche durch das „just in time“ und „just in sequence“ -Prinzip sehr anfällig für Verzögerungen geworden sind ([54], S.127). Daher ist es wünschenswert, Anomalien im Antriebsstrang eines Containerschiffes so früh wie möglich zu erkennen, um eventuell notwendige Reparaturen schnellstmöglich anstoßen zu können. Neuronale Netzwerke sind durch speziell geformte Netzarchitekturen in der Lage, Unregelmäßigkeiten in Eingangssignalen zu detektieren. Dieses Kapitel beschäftigt sich mit dem Einsatz derartiger Modelle für die Überwachung von multidimensionalen Sensorsignalen bei verschiedenen Subsystemen im Antriebsstrang eines 12X92DF der Firma WinGD.

6.1 Messaufzeichnungen

Bei diesem Antriebssystem handelt es sich um einen langsam laufenden, nockenwellenlosen und direkt abgetriebenen zwei-Takt Großmotor, welcher sowohl mit verschiedenen Dieselsorten, als auch mit Erdgas betrieben werden kann ([75], S.1-1). Außerdem lässt sich durch eine Umkehr der Drehrichtung ein Rückwärtslaufen des Motors ermöglichen. In Tabelle 5-1 sind die geometrischen Randbedingungen und Kennwerte des betrachteten Motors gelistet.



Bezeichnung	Wert
Hub	3468 mm
Bohrung	920 mm
ϵ	3.77
Zylinder	12
Nennleistung	5.320 kW/Zyl
Nenndrehzahl	80 1/min
Indizierter Nennmitteldruck	17.3 bar
Trockengewicht	2.14x10 ⁹ kg
Länge	22.700 m
Höhe	19.110 m
Breite	5.550 m

Tabelle 6-1: Geometrische Randbedingungen und Kennwerte eines 12X92DF-Antriebsstrangs der Firma WinGD ([75], S.1-1)

Als Datengrundlage für die Modellierungen in diesem Kapitel dienen Sensoraufzeichnungen aus mehreren Schiffen mit vorgestelltem Antriebsstrang von insgesamt über zwei Jahren Fahrbetrieb. Da die Rohdaten unterschiedliche Zeitstempel und -Auflösungen besitzen, werden alle Messsignale als erstes in eine einheitliche Abtastrate überführt (vgl. Kapitel 2.2.2). Außerdem enthalten die Daten Messfehler, welche mit Hilfe von Grenzwerten gefiltert und anschließend interpoliert werden (vgl. Kapitel 2.2.1).

6.1.1 Datenextraktion

Wie im vorangegangenen Kapitel, werden die Rohdaten entsprechend ihrer Leistung klassifiziert. Dafür wird ein Zeitfenster von einer Stunde definiert, minutenweise nach vorne geschoben und der Vorgang solange wiederholt, bis alle Fahrprofile durchiteriert und dementsprechend zugeschnitten sind. Zeitfenster, die aufgrund von Stillstand oder Wartungsarbeiten keine zusammenhängende Stunde an Messdaten ergeben, werden aussortiert (vgl. Kapitel 2.2.3). Anschließend besteht der Gesamtdatensatz aus einzelnen Datenpaketen, welche exakt einstündig und in sich vollständig konsistent sind.



Anschließend werden die Zeitfenster in Leistungsschritten von 5% in eine der auftretenden Leistung entsprechenden Klasse eingeteilt (vgl. Kapitel 2.2.4). Zusätzlich werden die Daten zwischen dem Betrieb mit Schweröl (Typ I) und Erdgas (Typ III) getrennt. Es gibt weitere, selten vorkommende Betriebsarten (Typ II, I, VII), in denen beide Kraftstoffarten gleichzeitig verwendet werden, um z.B. die Verbrennung zu stabilisieren, welche aber nicht teil der Betrachtungen in diesem Kapitel sind und an dieser Stelle nur der Vollständigkeit wegen genannt werden.

6.1.2 Datengewichtung

Um eine möglichst hohe Generalisierung der neuronalen Netzwerke zu erreichen, werden die Daten für die Modellierung gleichmäßig über alle Leistungsklassen verteilt. Für die Klassen, in denen es mehr Datenfenster als erforderlich gibt, wird die gewünschte Anzahl an Zeitreihen zufällig ausgewählt. In Klassen, bei denen nicht genügend Datenfenster zu Verfügung stehen, werden die Zeitfenster augmentiert (vgl. Kapitel 2.2.5). Leistungsklassen, in denen überhaupt kein Betrieb stattfindet, wie z.B. im unteren Teillastbetrieb in der Betriebsart III (Erdgas), werden ignoriert. Abbildung 6-1, Tabelle 6-2 und Abbildung 6-2 veranschaulichen diese Methodik exemplarisch anhand eines realen halbjährigen Fahrprofils von einem Schiff mit 12X92DF Antriebsstrang und einer anschließender Gewichtung der Zeitanteile für das zugrundeliegende Fahrprofil.

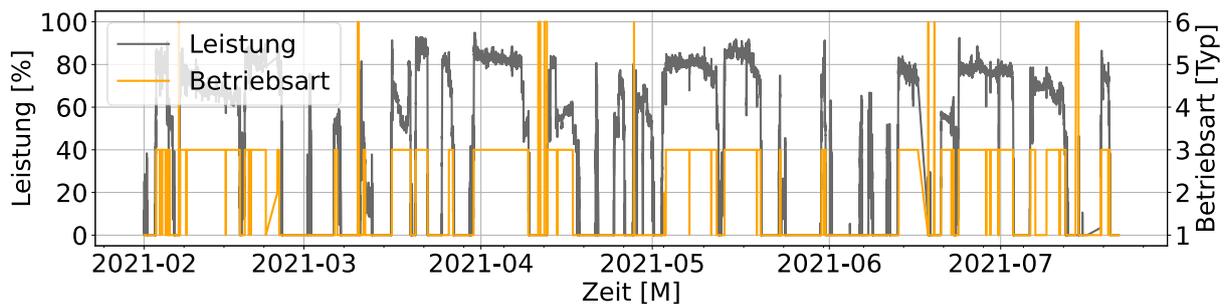


Abbildung 6-1: Reales Fahrprofil eines Schiffes mit 12X92DF Antriebsstrang

Leistungs- klasse [%]	10-10	10-40	15-15	...	65-65	65-70	65-60	...
Anzahl Schweröl [h]	40	4	7	...	19	8	5	...
Anzahl Erdgas [h]	-	-	-	...	135	17	7	...

Tabelle 6-2: Ausschnitt der Lastklassifikation aus einem Fahrprofil entsprechend Abbildung 6-1

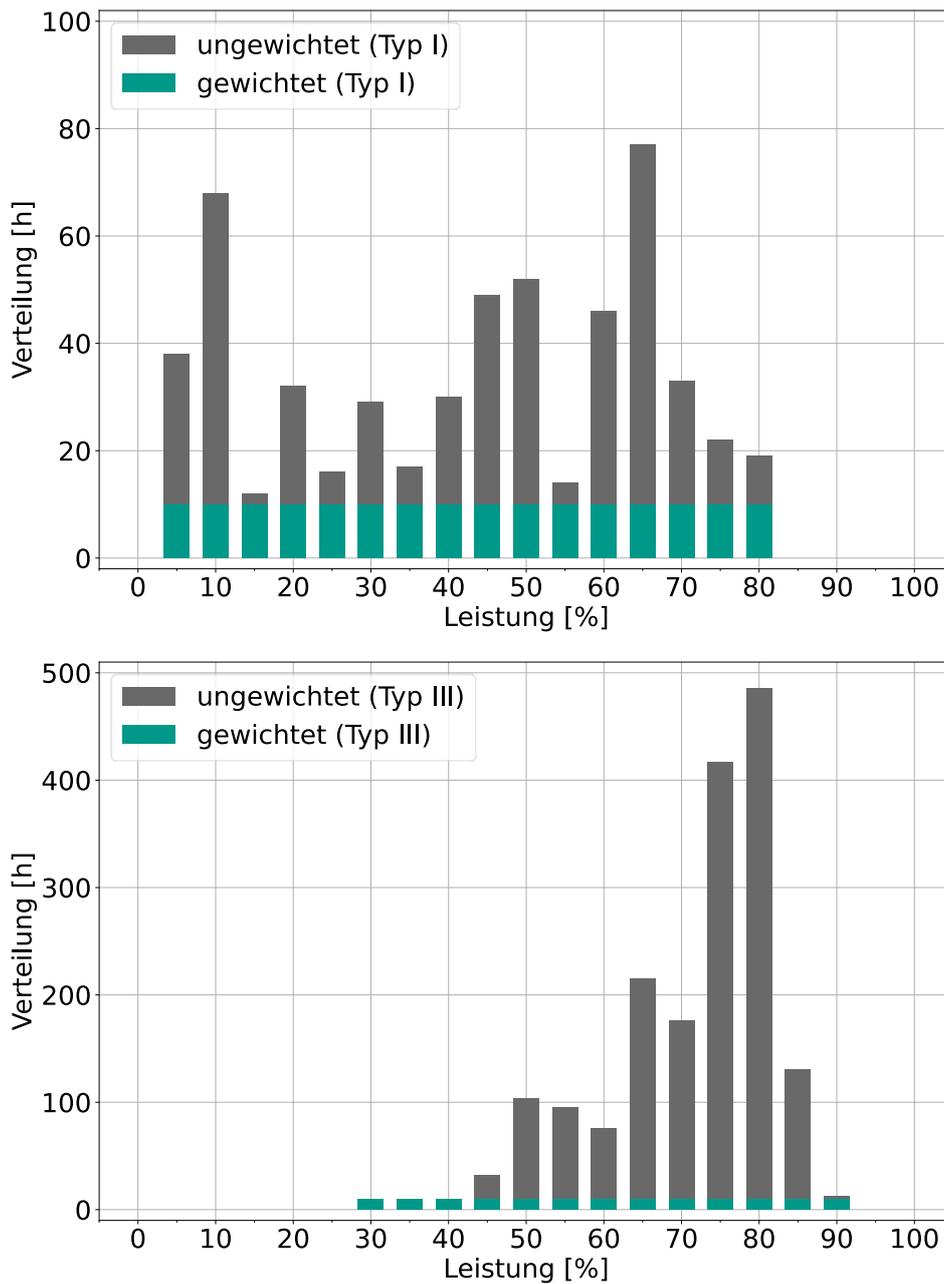


Abbildung 6-2: Gewichtung der Trainingsdaten für die Leistungsklassifikation
entsprechend Tabelle 2-1

6.2 Subsysteme

Nachdem alle verfügbaren Daten aufbereitet, zugeschnitten, klassifiziert und gewichtet sind, werden die vorliegenden Signale den verschiedenen Subsystemen zugeordnet, für welche diese maßgeblich zur Sicherstellung einer korrekten Funktionsweise sind. Um den allgemeinen Zustand des Antriebsstrangs zu beurteilen, werden die Motordrehzahl, die Leistung und die Außentemperatur bei allen Subsystemen als Eingangsgrößen betrachtet.



Im Folgenden werden die spezifischen Eingangssignale innerhalb ihrer Subsysteme kurz genannt und erläutert, bevor die Modellierungen beginnen.

6.2.1 Kolbenbewegung

In den langhubigen Kolbenwänden des 12X92DF Antriebsstrangs sind Thermoelemente eingelassen, sodass die Kolbenwandtemperatur auf einer horizontalen Ebene dreimal im radialen Abstand von 120° gemessen werden kann. Diese geben Aufschluss über die korrekte Funktionsweise des Schmierfilms und über die Laufbahn des Kolbens in der Buchse bzw. verändern sie sich, falls der Kolben frisst. Zusätzlich wird zur Modellierung die Anzahl der aktiven Einspritzdüsen herangezogen, da je nach Menge und Ort des eingespritzten Kraftstoffes unterschiedliche Temperaturniveaus entstehen können.

6.2.2 Verbrennung

Im Zylinder sind Drucksensoren eingebettet, welche den Kompressions- und Verbrennungsdruck messen. Diese geben unter anderem Aufschluss über unerwünschte Phänomene wie z.B. Vorzündungen oder Klopfen. Außerdem wird die Abgastemperatur gemessen, um einen Anhaltspunkt für die Enthalpie vor der Turbine zu bekommen.

6.2.3 Einspritzung

Für den Betrieb mit Schweröl sorgen mehrere Pumpen dafür, den Kraftstoff auf einen ausreichend hohen Druck in der Kraftstoffleitung zu verdichten. Zusätzlich wird das Schweröl im Betrieb mit Erdgas stetig umgewälzt, um eine Mindestviskosität aufrecht zu erhalten. Daher werden die Ansteuerung der Pumpen vom Steuergerät und der Druck im Nieder- und Hochdrucksystem für die Modellierungen betrachtet.

6.2.4 Hydraulik

Anders als bei einer Ventilsteuerung über eine Nockenwelle, werden die Auslassventile beim 12X92DF-Antriebsstrang über eine Hydraulik bewegt. Um die korrekte Funktionsweise zu beurteilen, werden die Öffnungs- und Schließzeiten des Abgasventils und die Ansteuerung der Hydraulikpumpen im Steuergerät betrachtet. Außerdem werden der Soll- und Istdruck im Hydrauliksystem verwendet.



6.3 Modellierung III

6.3.1 Datenaufteilung

Da viele Testdaten für die Auswertungen in den Kapiteln 6.3.5 - 6.3.8 von späteren Zeiträumen, wie die der ausgeschnittenen Leistungsprofile zu Verfügung stehen, werden entsprechend Tabelle 6-3 10% der Fahrprofile für das Training verwendet. Mit 5% der Fahrprofile wird der Optimierungsvorgang während dem Training validiert.

Anteil	Größe [%]
Training	10
Validierung	5
Test	85

Tabelle 6-3: Aufteilung des Ausgangsdatensatzes für die Modellierung III

6.3.2 Netzarchitektur

Um eine Einschätzung zu erhalten, ob die Signalverläufe der verschiedenen Systeme plausibel sind oder nicht, wird eine spezielle Encoder-Decoder Architektur verwendet (engl. Auto-Encoder), welche in Abbildung 6-3 schematisch dargestellt ist. Ausgehend von einer Eingangsmatrix werden die Signale im Verschlüsselungsteil (E1-E3) solange komprimiert, bis nur noch eine eindimensionale Repräsentation derer am Latentvektor vorliegt.

Diese verschlüsselten und komprimierten Informationen werden anschließend vom Rekonstruktionsteil (D1-D3) so wiederhergestellt, dass eine gewünschte Ausgangsmatrix vorliegt. Wie im vorangegangenen Kapitel 5 enthält die Netzarchitektur eine Zeitdimension, um transiente Vorgänge abbilden zu können. Der Latentvektor besitzt allerdings keine, um die beiden Teile sowohl im Daten als auch im Zeitbereich vollständig voneinander zu entkoppeln (i.A. [52]). Das bedeutet, die Gewichtungen der Zeitreihen sind im Encoder- und Decoderteil nicht miteinander verbunden.

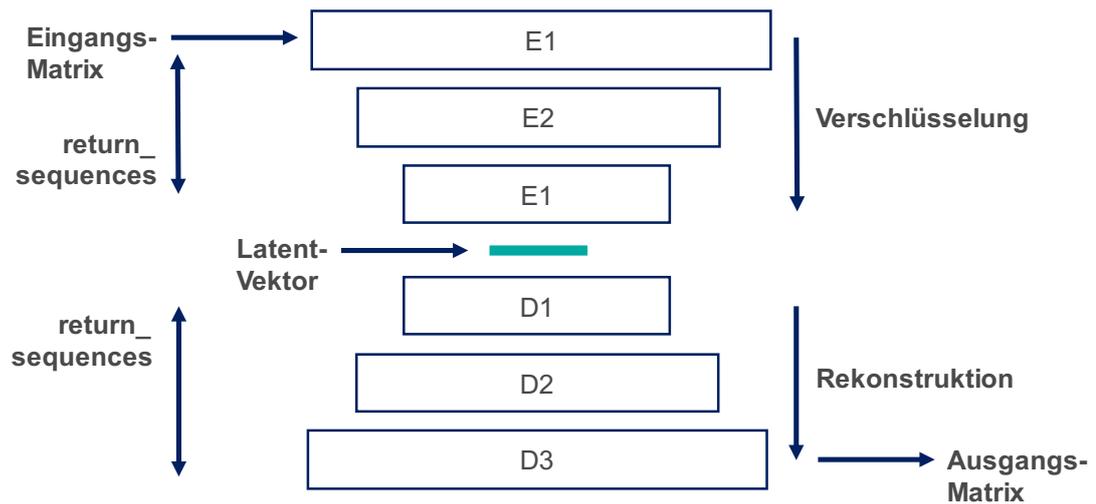


Abbildung 6-3: Netzlayout für die Fehlererkennung III

Da die Gewichtungen im Verschlüsselungs- und Rekonstruktionsteil während des Trainings an die Daten der einzelnen Subsysteme angepasst werden, wirken sich Unregelmäßigkeiten bzw. fehlerhafte Signalverläufe negativ auf die Rekonstruktion aus. Das bedeutet vereinfacht, verändern sich die Eingänge unerwartet, passen die Ausgänge nicht mehr dazu, ähnlich wie bei einem Schlüssel-Schloss Prinzip. Es ist daher wichtig, überwiegend fehlerfreie Daten für das Training zu verwenden [52] und [53]. In der Literatur konnten allerdings keine Angaben gefunden werden, bis zu welchem Anteil fehlerhafte Daten im Training grundsätzlich zulässig sind. Es wird daher in dieser Arbeit darauf geachtet, möglichst alle fehlerhaften Daten für die Modellierungen vollständig zu entfernen.

6.3.3 Rekonstruktionsfehler

Der Netzarchitektur folgend kann ein Fehlersignal nach Gleichung (2.2.10) zwischen dem Ein- und Ausgang gebildet werden, welches als Maß für den Grad einer Anomalie innerhalb der Signalverläufe steht. Mit der Einführung eines Grenzwertes θ , bis zu welchem von einer regulären Signalschwankung ausgegangen werden kann, wird das Fehlersignal relativiert und damit dimensionslos bzw. zwischen den verschiedenen Subsystemen vergleichbar gemacht.

$$\vec{A} = \frac{|\overrightarrow{Ein} - \overrightarrow{Aus}|}{\vec{\theta}} \quad (6.1)$$



6.3.4 Verschlüsselungsgrad

Mit der Einführung eines Verschlüsselungsgrades K nach Gleichung (2.2.10), welcher das Verhältnis zwischen den Dimensionen der Eingangsmatrix und des Latentvektors beschreibt, lassen sich die verschiedenen Netzarchitekturen miteinander vergleichen.

$$K = \frac{\text{Zeitfenster} * \text{Eingangssignale}}{\text{Neuronen Latentvektor}} \quad (6.2)$$

6.3.5 Kolbenwandtemperaturen III-I

Im Modellierungsversuch III-I werden zuerst die Kolbenwandtemperaturen betrachtet. Als Eingang sind zusätzlich zu Leistung, Drehzahl und Kolbenwandtemperaturen noch die Anzahl der aktiven Injektoren enthalten, da sich diese auf die Temperatur im Brennraum auswirken. Je nachdem, wie viele Injektoren aktiv sind, verteilt und entzündet sich der Kraftstoff unterschiedlich im Brennraum, und das Temperaturniveau an den einzelnen Messstellen variiert.

Der Ausgang besteht mit den drei Temperaturmessstellen aus entsprechend drei Signalen, daher wird ein eher kleineres Modell mit wenigen Zellen aufgebaut. Der Latentvektor wird mit einem Parameter gewählt, um eine vergleichsweise hohe Kompression in der Datendimension zu erreichen. Der Zellstatus ist im Gegensatz zur Zeitdimension nicht aktiv, da keine Zusammenhänge zwischen den einzelnen einstündigen Zeitfenstern gefunden, sondern nur die in sich konsistenten Datenreihen abgebildet und auf Plausibilität überprüft werden sollen.

Tabelle 6-4 listet die Netzparameter für das Modell der Kolbenwandtemperaturen. Da im Vergleich zum vorangegangenen Kapitel vergleichsweise viele Daten vorhanden sind, wird die Lernrate geringfügig auf 10^{-3} erhöht. Im Verschlüsselungsteil werden 10% der Gewichtungen pro Lernepoche zurückgesetzt, um das Modell robuster gegen Messfehler zu machen ([22], p.8609), was auch als „Dropout“ bezeichnet wird.



Parameter	Wert
E1	32 LSTM
E2	16 LSTM
E3	8 LSTM
Dropout E	0.1
Latentvektor	1
K	480
D1	8 LSTM
D2	16 LSTM
D3	32 LSTM
Ausgang	(60x3) Neuronen
Aktivierung	TanH
Fehlerfunktion	Huber
Lernrate	10 ⁻³
Zellstatus	nicht aktiv
Zeitdimension	aktiv
Abbruchbedingung	25 Epochen
Eingangsdimension	(60x7)
Ausgangsdimension	(60x3)

Tabelle 6-4: Netzparameter für die Fehlererkennung III-I

Nach 57 Epochen wird das Training entsprechend dem Abbruchkriterium beendet. Da das Modell für jede der drei Wandtemperaturen im Zylinder über ein Ausgangssignal verfügt, und die Signale alle an der gleichen vertikalen Stelle gemessen werden, wird ein Summenfehler mit folgenden Gewichtungen nach Tabelle 6-5 eingeführt.

Signal	Gewichtung
Wandtemperatur A	0.33
Wandtemperatur B	0.33
Wandtemperatur C	0.33

Tabelle 6-5: Gewichtung der Kolbenwandtemperaturen für die Fehlererkennung III-I



Als nächstes werden exemplarisch einige Fahrsituationen analysiert, um die Vorhersagequalität des Modells genauer zu untersuchen. Dafür werden neue und unbekannte Schiffsdaten herangezogen, in denen es bekanntlich zu einigen Auffälligkeiten gekommen ist und betrachtet, wie sich die berechnete Modellanomalie an diesen Stellen verhält. In Abbildung 6-4 (unten) sind dafür die Last und Betriebsart aus einem Fahrprofil I aufgetragen, welche von einem Schiff mit 12X92DF-Antriebsstrang stammen. Zusätzlich sind mittig die entsprechenden Wandtemperaturen für Zylinder 8 aufgetragen, welche während des Fahrprofils gemessen werden. Oben ist der Grad der Anomalie dargestellt, welcher nach Gleichung (6.1) berechnet und nach Tabelle 6-5 gewichtet ist.

Grundsätzlich lässt sich erkennen, dass die Wandtemperaturen mit der Leistung korrelieren und sich um einen Mittelwert bewegen, je nachdem, an welcher Stelle sich die Kolbenringe gerade befinden. Das bedeutet zum einen, je höher die Leistung ist, desto höher liegen die Wandtemperaturen und umgekehrt. Zum anderen sind die Wandtemperaturen A, B und C abwechselnd am höchsten bzw. am niedrigsten und zirkulieren um einen Mittelwert. Im Allgemeinen ist der Kurvenverlauf für das halbe Jahr an Fahrt normal, und es lassen sich keine Besonderheiten erkennen. Der Anomaliegrad beträgt daher durchgehend unter eins, was bedeutet, dass auch im Modell keine Auffälligkeiten für den Temperaturverlauf vorhergesagt werden.

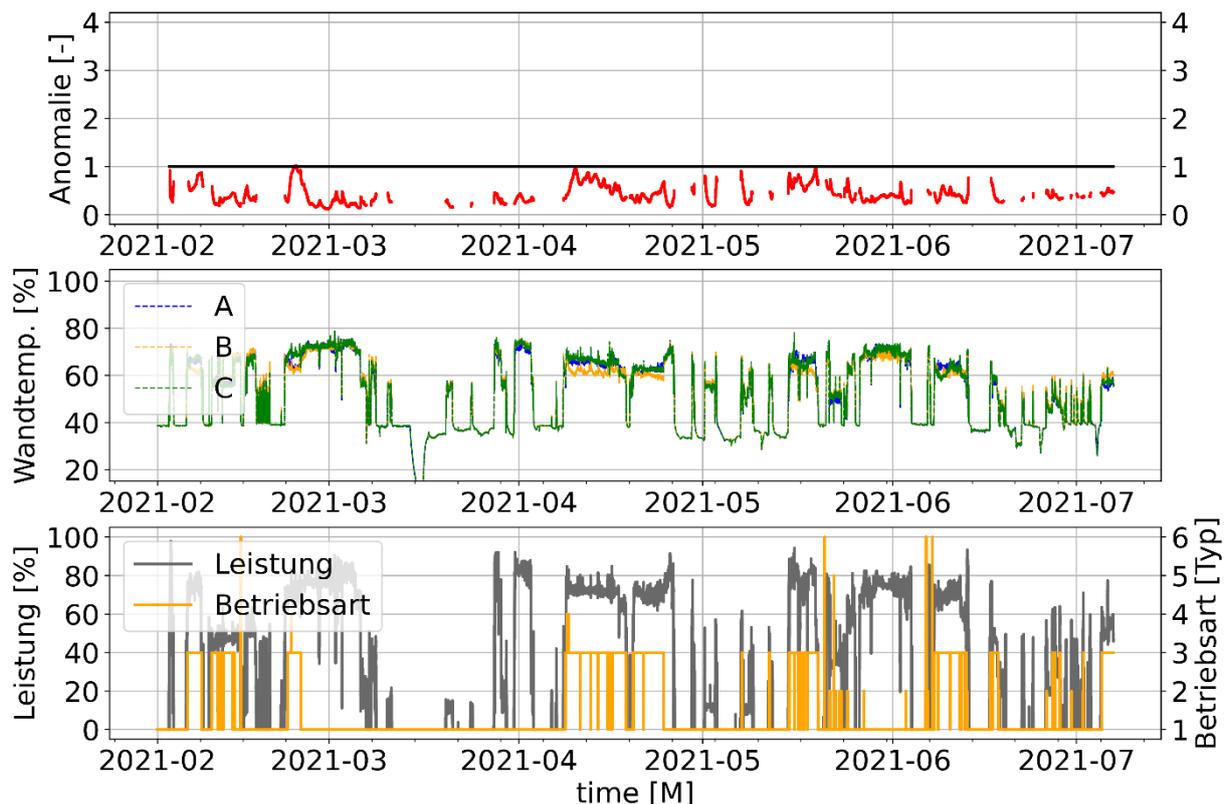


Abbildung 6-4: Auswertung der Fehlererkennung III-I am Fahrprofil 1 von Zylinder 8



Abbildung 6-5 zeigt Zylinder 7 desselben Schiffes während desselben Fahrprofils I. Dementsprechend sind das Leistungs- und Betriebsartsignal identisch, die Wandtemperaturen allerdings zu der vorangegangenen Betrachtung verschieden, da es sich um einen anderen Zylinder handelt. Bei den Temperaturkurven lässt sich Anfang März ein starker Einbruch von Wandtemperatur C feststellen. Dort liegt ein Problem mit dem Temperatursensor vor, welcher ein deutlich zu niedriges Temperatursignal ausgibt. Der Anomaliegrad steigt auf einen Wert von drei, was konkret bedeutet, die Abweichung ist mehr als dreimal so hoch wie eigentlich zulässig. Nachdem der Sensor im Juni erfolgreich getauscht werden konnte, befindet sich das Temperatursignal wieder in einem plausiblen, der Leistung entsprechenden Bereich, und der Anomaliegrad liegt unter eins.

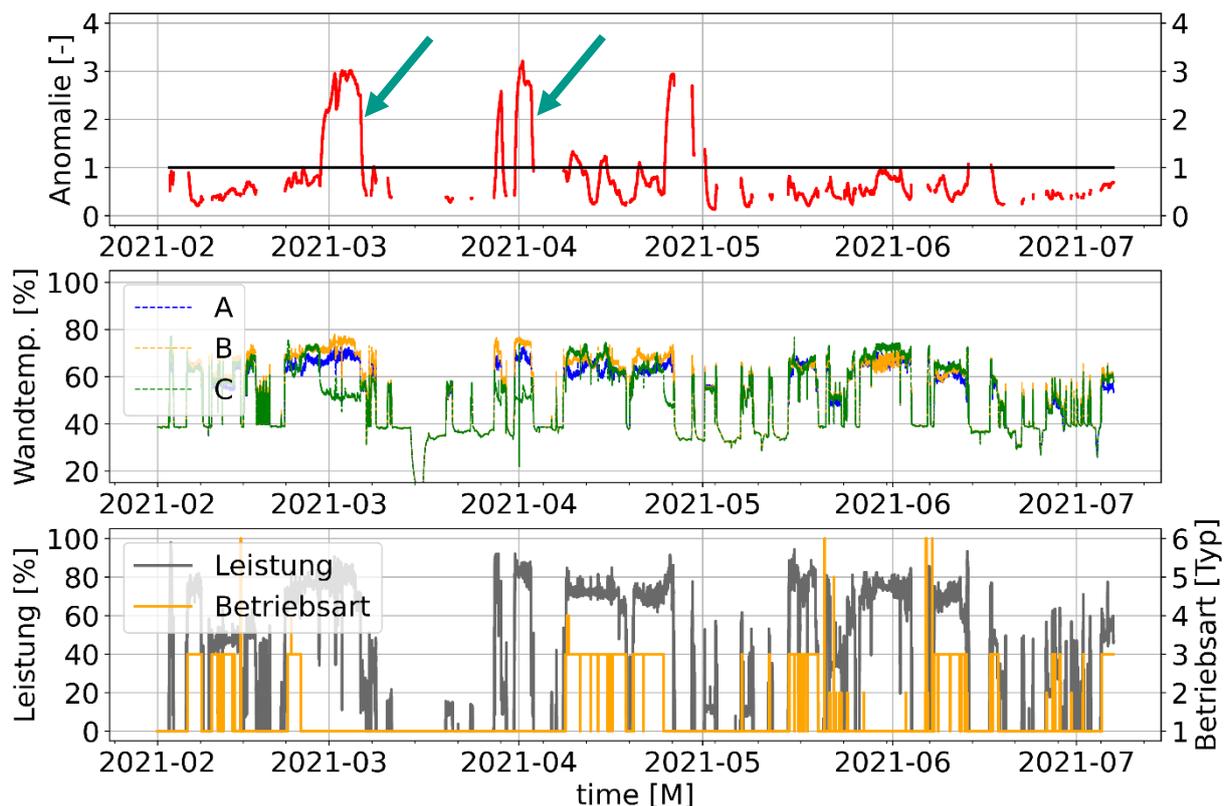


Abbildung 6-5: Auswertung der Fehlerkennung III-I am Fahrprofil 1 von Zylinder 7

Spannend für die Modellbewertung sind neben Sensorfehlern natürlich echte Schadensfälle. Bei den Wandtemperaturen bedeutet das, der Kolben hat unerwarteten Kontakt mit der Buchse und spant Teile davon ab bzw. frisst. Dies ist besonders gefährlich, da es im Extremfall zu schwerwiegenden Schäden am Zylinder bis hin zum Ausfall des gesamten Antriebsstrangs kommen kann. Derartige Ereignisse sind sehr selten, können aber dennoch gelegentlich vorkommen.



In Abbildung 6-6 ist daher ein zweites Fahrprofil eines anderen Schiffes mit demselben Antriebsstrang sowie die Wandtemperaturen von Zylinder 1 abgebildet. Sowohl in den Temperaturverläufen als auch im Anomaliegrad lassen sich keine Ereignisse entdecken. Im Gegensatz dazu repräsentiert Abbildung 6-7 den Verlauf der Wandtemperaturen von Zylinder 12, welcher im Laufe des Fahrprofils gefressen hat und die Kolbenbuchse anschließend ausgetauscht werden musste. Ende März lässt sich zum ersten Mal eine Anomalie in den Temperaturkurven feststellen. Die Wandtemperatur B (gelb) steigt deutlich an, während A und C abfallen, wie auf der Detailauswertung in Abbildung 6-8 zu erkennen ist. Hier kommt es zu einem ersten Kontakt von Kolben und Buchse in der Nähe des Temperatursensors B, wie in Abbildung 6-9 schematisch veranschaulicht, und in Folge dessen steigt die Temperatur leicht an. Aufgrund abrasiver Partikel werden nun vermutlich weitere Schäden an der Buchse in der Nähe der Messtelle C (grün) erzeugt, weshalb die Temperatur dort ebenfalls leicht ansteigt. Gegen Ende des Fahrprofils liegt ein schwerwiegender Kolbenfresser vor, und die Wandtemperaturen oszillieren deutlich stärker als üblich. Der Anomaliegrad liegt dabei bereits ab der ersten Auffälligkeit über dem zulässigen Grenzwert und steigt stetig bis auf einen Wert über drei.

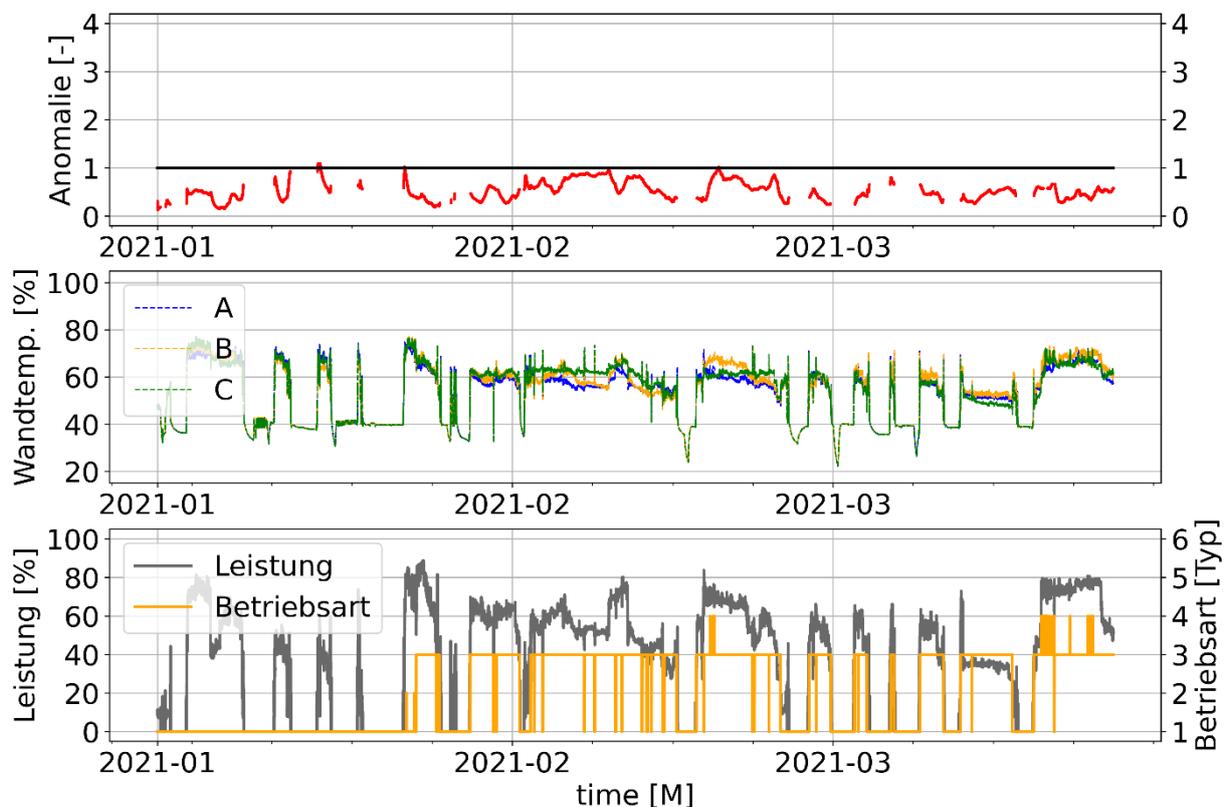


Abbildung 6-6: Auswertung der Fehlerkennung III-I am Fahrprofil 2 von Zylinder 1

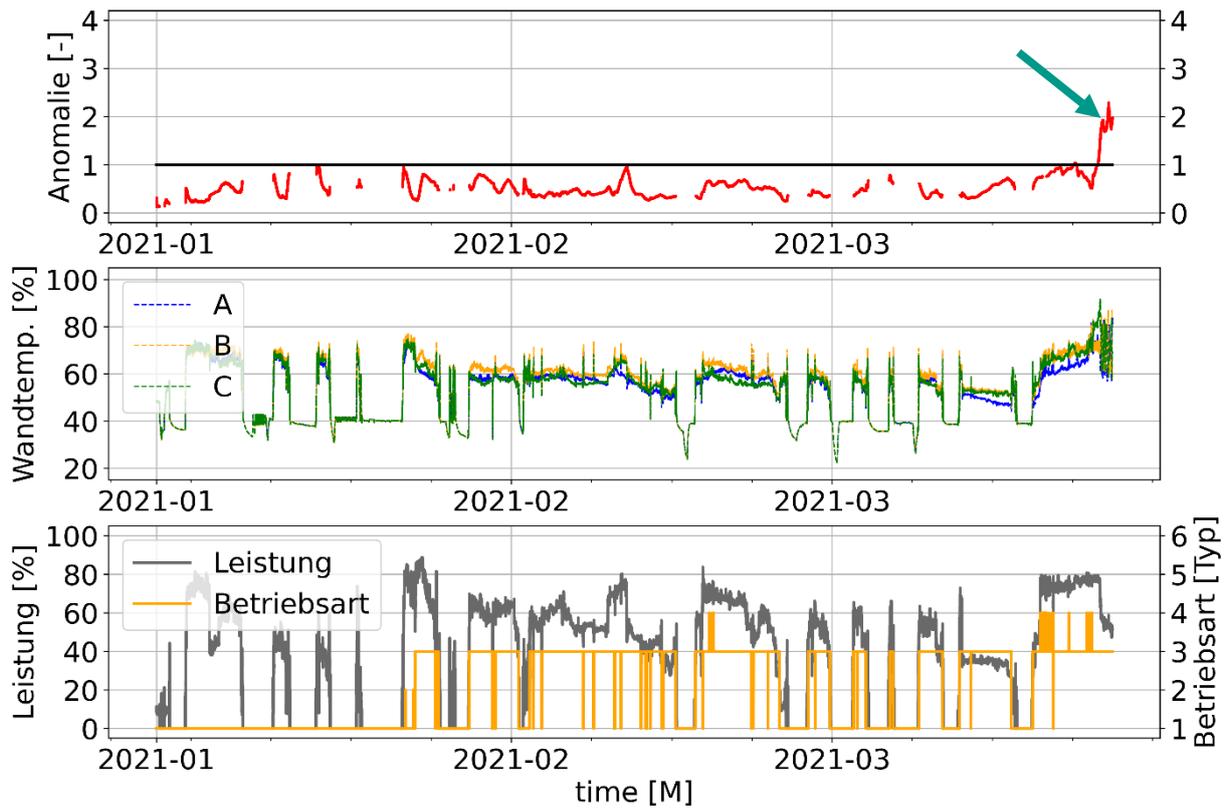


Abbildung 6-7: Auswertung der Fehlerkennung III-I am Fahrprofil 2 von Zylinder 12

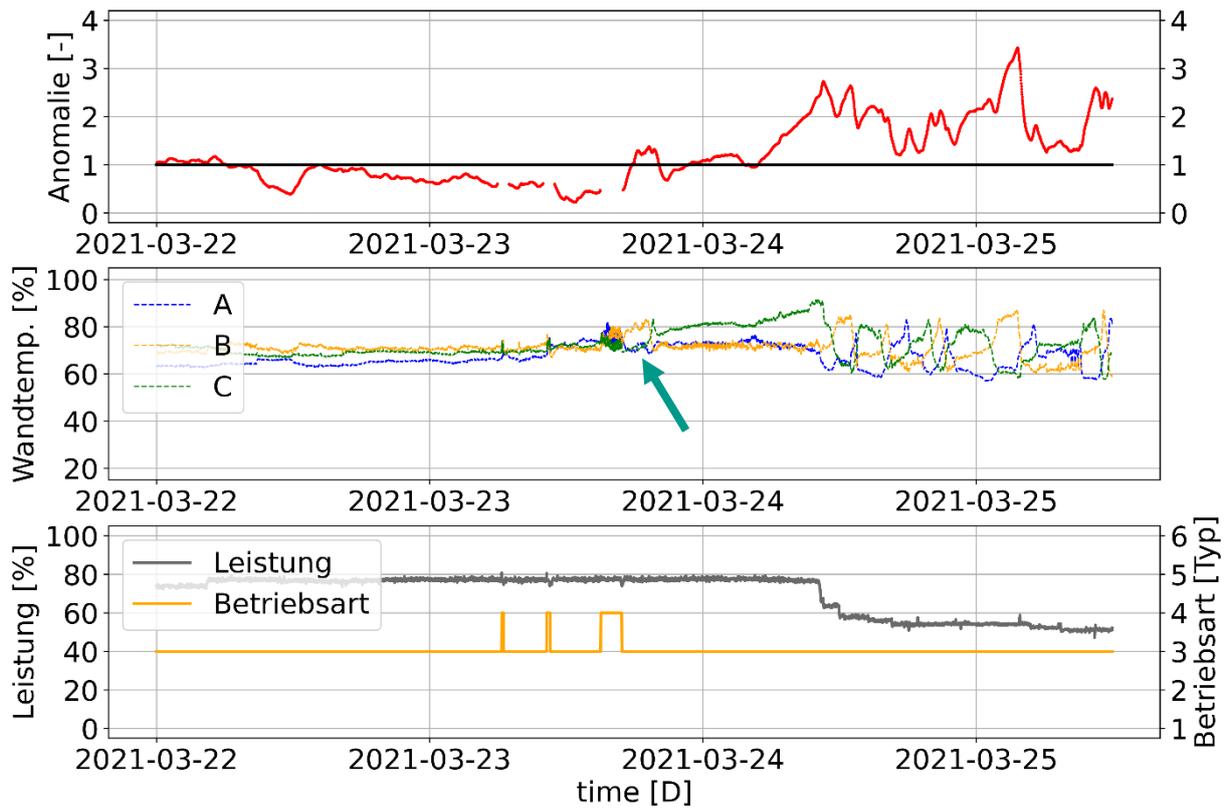


Abbildung 6-8: Detailauswertung der Fehlerkennung III-I am Fahrprofil 2 von Zylinder 12

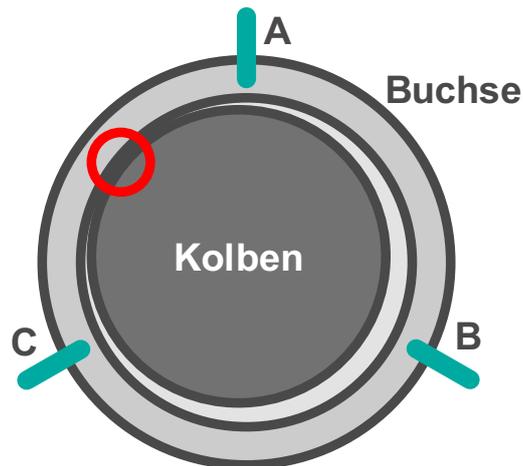


Abbildung 6-9: Schematische Ansicht des Kolbenkontakts von Zylinder 12 während dem Fahrprofil 2

Da das Schiff trotz eines beginnenden Schadens noch mehr als zwei Tage weitergefahren ist, zeigt, dass die menschliche Überwachung von Sensorsignalen in Echtzeit und über längeren Zeitraum fehleranfällig ist.



Eventuell hätte der Schaden mit Hilfe von Systemregeln entdeckt werden können, welche Alarm schlagen, wenn die Temperaturen z.B. zu weit auseinanderliegen oder sich in der Frequenz ändern. Daher soll abschließend ein Fall untersucht werden, bei dem das menschliche Verständnis vorerst nicht von einem Schaden ausgeht, das KI-basierte Modell allerdings schon.

Dafür ist in Abbildung 6-10 das Fahrprofil 3 mit den dazugehörigen Wandtemperaturen und der entsprechenden Bewertung aus der Modellierung III-I aufgetragen. Die Temperatur B liegt über das gesamte Fahrprofil höher als die Temperaturen A und C, weil dort durch das Ringschloss mehr Wärme entweicht als an den anderen beiden Messstellen. Im Juni steigt der Anomaliegrad mehrere Tage über einen Wert von eins, obwohl der Kurvenverlauf weiter plausibel erscheint. Mögliche Ursachen dafür könnten z.B. eine Bewegung der Kolbenringe oder ein tatsächlicher Kontakt mit der Buchse sein. Hier sind weitere Einschätzungen der Bauteilverantwortlichen notwendig, um den Fehlerverlauf zu verifizieren und ggf. zu klassifizieren.

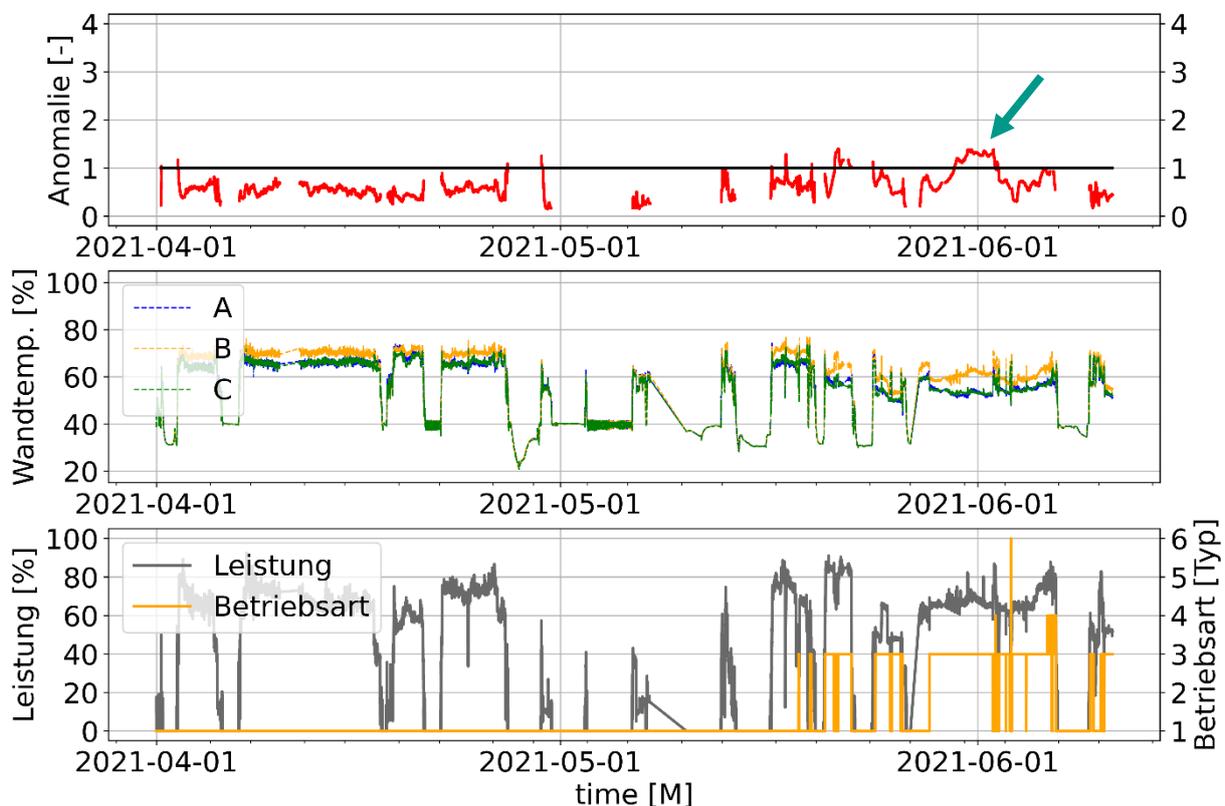


Abbildung 6-10: Detaillauswertung der Fehlererkennung III-I am Fahrprofil 3 von Zylinder 1



6.3.6 Verbrennung III-II

Das Modell für den Verbrennungsdruck enthält Ausgangsgrößen, welche physikalisch komplexere Wechselwirkungen zueinander besitzen wie die drei Kolbenwandtemperaturen. Deshalb wird der Latentvektor für die Fehlererkennung III-II von eins auf drei erhöht, um mehr Informationen in der Verschlüsselung speichern zu können, der Verschlüsselungsgrad sinkt damit auf $KVI-II = 140$. Die Anzahl der Neuronen wird verdoppelt und ansonsten bleibt die Netzarchitektur identisch (gelistete Parametrierung im Anhang). Bei der Fehlerbildung wird für eine detaillierte Betrachtung kein Summenfehler gebildet, sondern jedes Signal alleinstehend dargestellt.

In Abbildung 6-11 ist dafür ein neues Fahrprofil 4 mit dem entsprechenden Kompressions- und Verbrennungsdruck sowie der Abgastemperatur abgebildet. Die Drücke werden über ein Arbeitsspiel gemittelt aufgezeichnet und ausgewertet. Ab dem 08. April lässt sich zum ersten Mal eine erhöhte Anomalie im Kompressionsdruck feststellen. Anschließend steigt die Anomalie in der Abgastemperatur und bleibt durchgehend über eins. Hier kommt es zu ungewünschten Vorzündungen, welche aufgrund einer zu früh einsetzenden Verbrennung auftreten. Ende April sind diese teilweise so stark, dass der Kompressionsdruck längere Zeit über dem Verbrennungsdruck liegt. Interessant dabei ist, dass sich dieses Phänomen nicht nur im Kompressionsdruck, sondern bereits Mitte April auch in der Abgastemperatur und im Verbrennungsdruck feststellen lässt, obwohl diese plausibel erscheinen. Hier kann durch die Betrachtung von mehreren zusammenhängenden Signalen und deren Wechselwirkung ein kritischer Zustand wesentlich eindeutiger vom Modell detektiert werden als bei einer reinen Einzelbetrachtung der Temperatur bzw. Drücke über Grenzwertschranken oder mit dem menschlichen Auge.

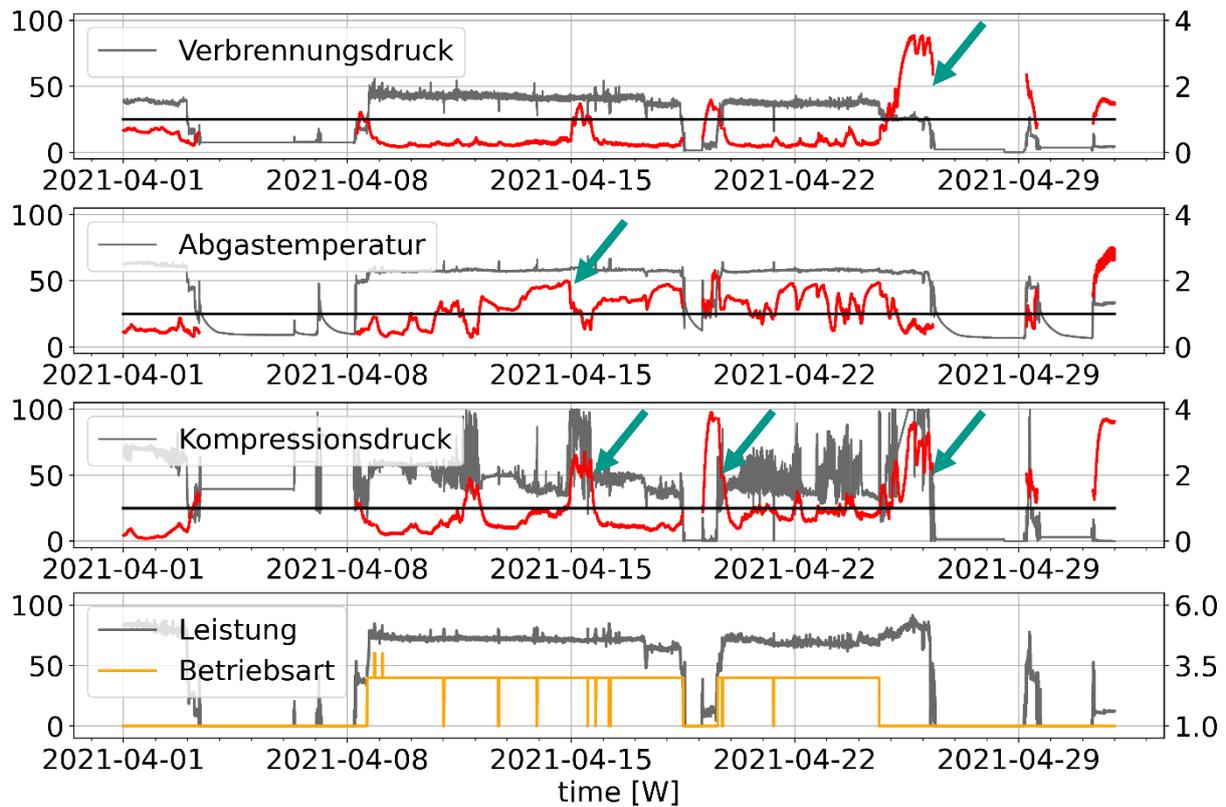


Abbildung 6-11: Detailauswertung der Fehlerkennung III-II am Fahrprofil 4 von Zylinder 7

Zum Vergleich ist in Abbildung 6-12 die Modellauswertung in demselben Zeitraum für den benachbarten Zylinder 8 abgebildet, an dem es zu keinen Problemen in der Verbrennung gekommen ist. Es lässt sich erkennen, dass hier die Anomaliebewertung vom Modell durchweg unter dem Grenzwert von eins liegt.

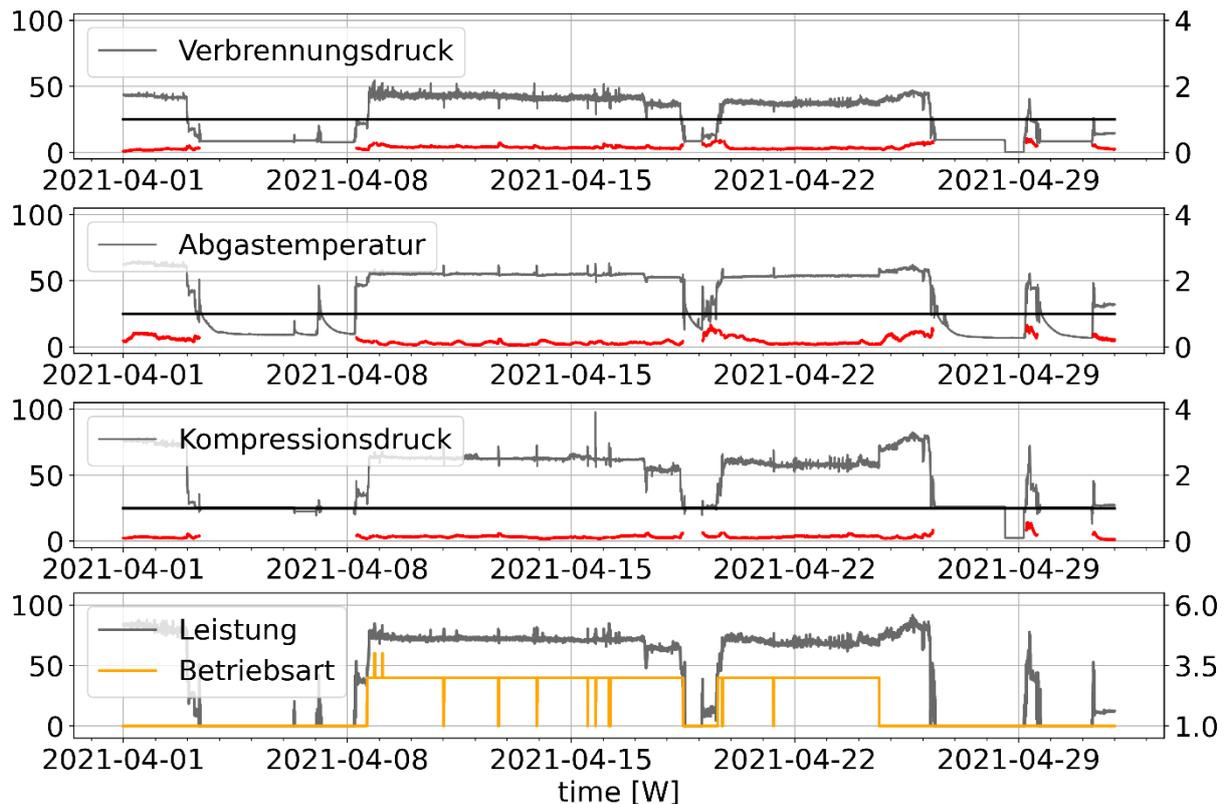


Abbildung 6-12: Detailauswertung der Fehlerkennung III-II am Fahrprofil 4 von Zylinder 8

6.3.7 Einspritzung III-III

Für die Fehlerkennung III-III bei der Einspritzung sind insgesamt sieben Ausgangsgrößen vom Modell zu überwachen. Damit der Verschlüsselungsgrad weiterhin bei $KVI-III = KVI-II = 140$ bleibt, wird der Latentvektor auf sieben vergrößert (gelistete Parametrierung im Anhang).

Abbildung 6-15 zeigt erneut das gesamte Fahrprofil 1, bei dem es während des Betriebs zu Problemen an den Pumpen gekommen ist. Zusätzlich sind die Drücke im Nieder- und Hochdrucksystem sowie die Ansteuerung der vier Kraftstoffpumpen und die Druckdifferenz zwischen Soll- und Istwert aufgetragen. Während der Betriebsart III (Erdgas) wird das Schweröl ständig auf Hochdruck gehalten und umgewälzt, um eine gewisse Mindestviskosität zu erhalten. Im Betrieb I (Schweröl) wird der Kraftstoffdruck entsprechend geregelt, um den gewünschten Druck für die Einspritzung zu erzielen.



Generell ist es normal, dass im Stillstand und beim Anfahren hohe Differenzen zwischen Soll- und Istdruck auftreten, da der Druck im Hochdrucksystem erst aufgebaut, bzw. abgebaut werden muss, um dem Sollwert zu entsprechen, bzw. die Druckregelung eine gewisse Trägheit besitzt, da sehr große Mengen Fluid verdichtet werden müssen. Das Modell wird daher erst ab einer Leistung von über 5% zur Bewertung der Systemgrößen eingesetzt, wie in Abbildung 6-13 erkennbar. Bei dem Bereich um den 18. Februar 2021 handelt es sich um eine Signalstörung, bei dem einige Signale rauschen, und dort ist daher ebenfalls keine Modellauswertung gezeichnet.

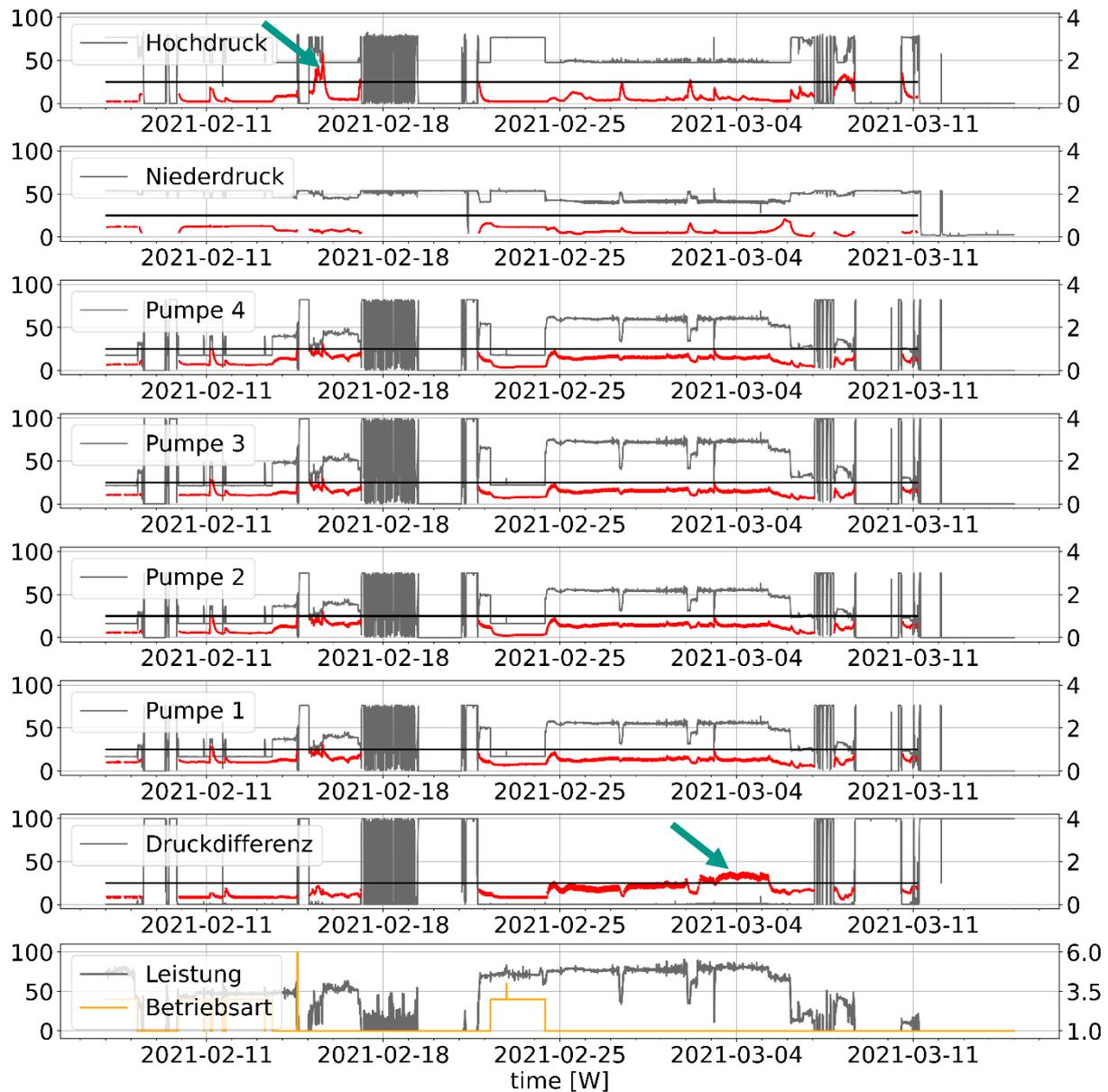


Abbildung 6-13: Detailauswertung 1 der Fehlerkennung III-III am Fahrprofil 1 (Februar-März)



Kurz zuvor, Mitte Februar lässt sich allerdings zum ersten Mal eine erhöhte Anomalie bei der Rekonstruktion des Hochdrucks vom Modell im Betrieb mit Schweröl erkennen.

Anfang März liegt auch die Anomalie für die Druckdifferenz über dem Grenzwert. Das bedeutet, die Leistung der Pumpen entspricht nicht mehr derer, welche für das Aufrechterhalten des Solldrucks im entsprechenden Betriebszustand eigentlich notwendig ist. Hier konnte bei einer Inspektion gegen Ende März eine Leckage an den Pumpen festgestellt und beseitigt werden.

Mitte Mai zeigt das Hochdrucksignal eine erhöhte Anomalie im Betrieb mit Erdgas (vgl. Abbildung 6-14). Es lässt sich im Signalverlauf konkret erkennen, dass der Druck beim Umwälzen des Schweröls mehrfach unerwartet absinkt. Als Anfang Juni anschließend in den Betrieb mit Schweröl gewechselt wird, zeigt das Signal für die Druckdifferenz eine erhöhte Anomalie. Hier wird laut Modell mehr oder weniger Arbeit von den Pumpen geleistet als eigentlich für den Betriebszustand üblich ist. Das spiegelt sich auch in einer leicht erhöhten Anomalie für die Pumpenansteuerung wider. Ende Juni konnte bei einer Inspektion eine gebrochene Feder an einer der Pumpen gefunden werden, weshalb die verbleibenden Pumpen mehr leisten müssen, um den Druck aufrecht zu erhalten.

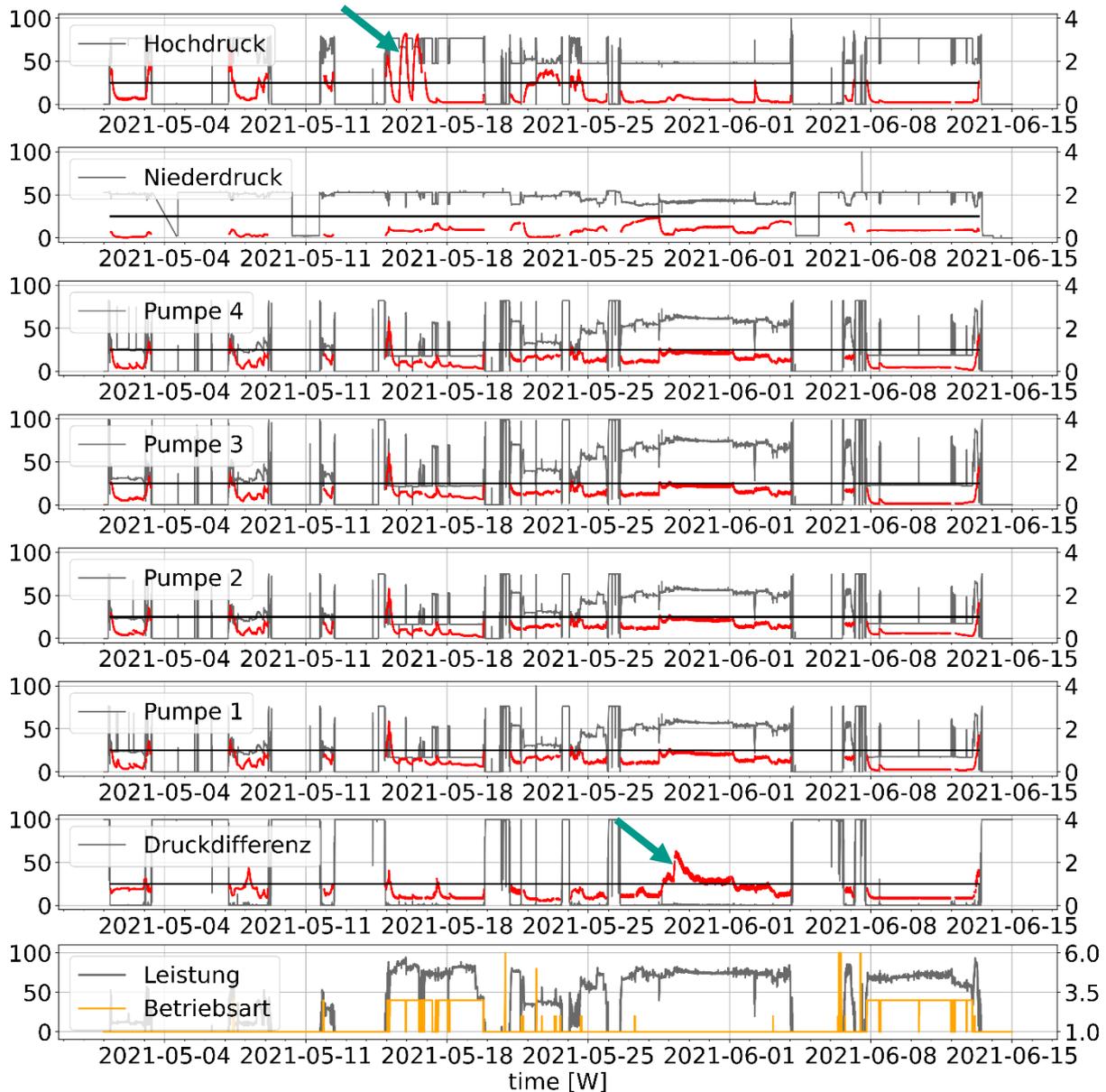


Abbildung 6-14: Detailauswertung 1 der Fehlerkennung III-III am Fahrprofil 1 (Februar-März)

Zum Vergleich ist in Abbildung 9-1 im Anhang ein Fahrprofil aufgetragen, in dem es zu keinen Problemen am Kraftstoffsystem gekommen ist. Dort liegt die Anomaliebewertung durchgehend unter dem vorgesehenen Grenzwert von eins.

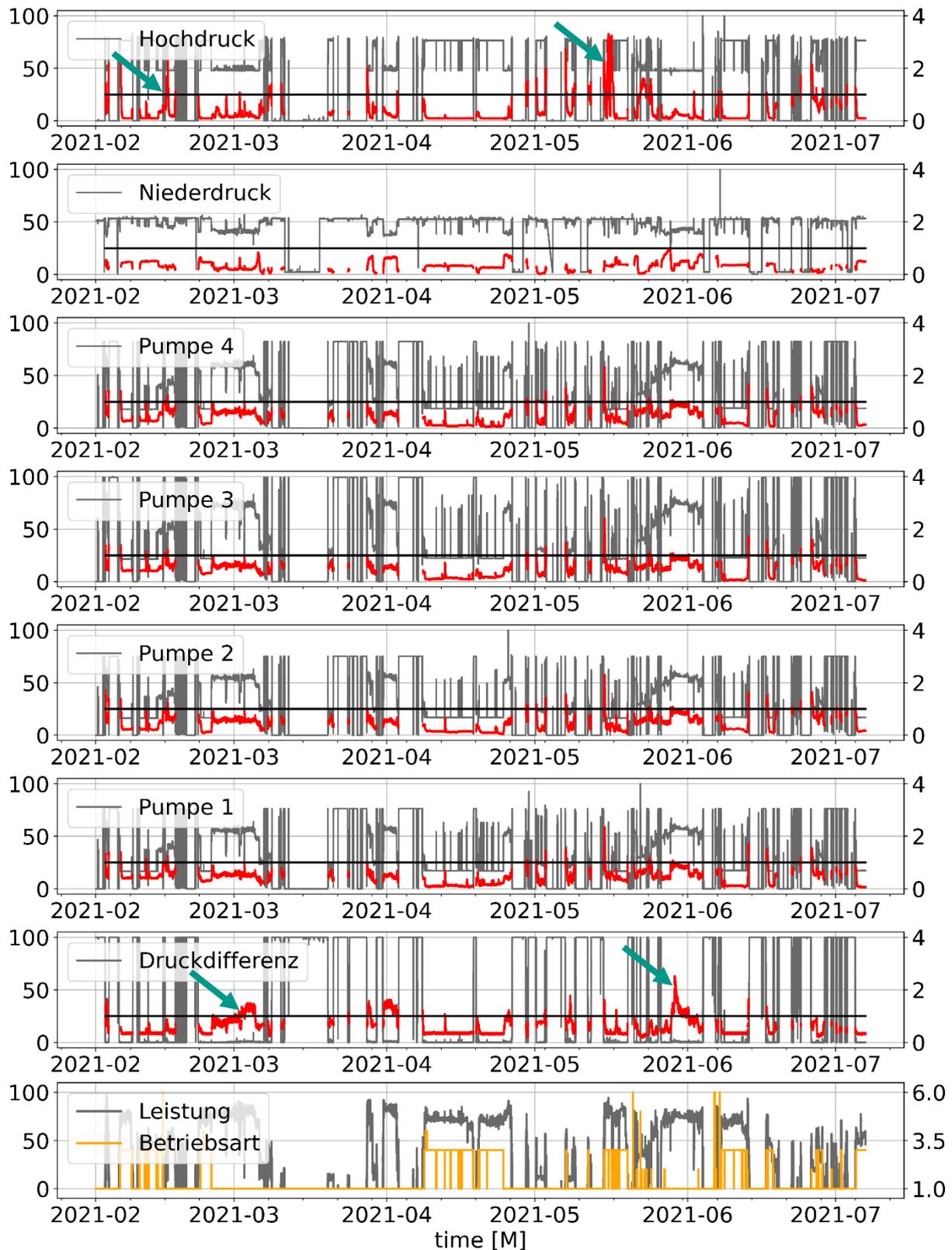


Abbildung 6-15: Auswertung der Fehlererkennung III-III am Fahrprofil 1



6.3.8 Hydraulik III-IV

Die Fehlererkennung III-IV für die Überwachung der hydraulischen Ansteuerung der Auslassventile besitzt insgesamt neun Signale (gelistete Parametrierung im Anhang), welche vom Modell auf Plausibilität überprüft werden. Dabei beschreiben der gemessene Hydraulikdruck, die Druckdifferenz zwischen Soll- und Istdruck und die Pumpenansteuerung der drei Hydraulikpumpen vom Steuergerät den allgemeinen Systemzustand und gehen zusammen mit der Betriebsart und der Leistung für alle Zylinder mit ein. Die zylinderspezifischen Signale, wie der Zeitpunkt für Auslass öffnet und schließt, sowie die Dauer für Auslass offen und geschlossen beschreiben die Ventilansteuerung durch das Hydrauliksystem.

Für die Analyse des Modells ist in Abbildung 6-16 und Abbildung 6-17 ein viertes unbekanntes Fahrprofil aufgetragen, welches, wie bei allen Auswertungen, nicht Teil des Modelltrainings war. Grundsätzlich wird der Hydraulikdruck über die Ansteuerung der Pumpen geregelt und das Auslassventil entsprechend dem Motorenkennfeld, welches im Steuergerät hinterlegt ist, bewegt. Ende März lässt sich erkennen, dass es zu einem Problem in der Regelung kommt, da alle drei Pumpen unerwartet auf voller Leistung laufen. Die Anomaliebewertung des Modells liegt entsprechend bei über drei. Infolgedessen steigt der Hydraulikdruck stark an und wird mit einer Anomalie von zwei ebenfalls als problematisch vorhergesagt. Da auch in der Druckdifferenz zwischen Soll- und Istdruck ein erhöhter Rekonstruktionsfehler auftritt, kann davon ausgegangen werden, dass für gegebenen Betriebspunkt eigentlich ein niedriger Druck vorgesehen ist, die Pumpenansteuerung aber mechanisch oder softwareseitig auf ihrem Maximum festhängt.

Bei der Analyse der Bewegungen des Auslassventils kann festgestellt werden, dass sich der erhöhte Druck auch auf den Öffnungszeitpunkt auswirkt. Aufgrund des höheren Drucks benötigt das Ventil weniger Zeit für die Öffnungsbewegung und öffnet daher früher als eigentlich vorgesehen. Dies resultiert Ende März in einer erhöhten Fehlerbewertung des Signals für „Auslass öffnet“. Die Öffnungs- und Schließdauern bleiben von der Störung unbeeinträchtigt genauso wie der Zeitpunkt für „Auslass schließt“. Anfang Mai lassen sich beim Signal für „Auslass öffnet“ drei weitere Fehler erkennen, bei dem die Anomalie teilweise über vier beträgt und auch der Messwert sehr früh liegt. Hier wird bei den ersten beiden Ereignissen von Messfehlern ausgegangen, da der Druck im Hydrauliksystem und die Pumpenansteuerung keine Probleme zu den entsprechenden Zeitpunkten aufweisen. Beim dritten Ereignis gegen Mitte Mai liegt dagegen auch das Fehlersignal bei allen drei Pumpenansteuerungen kurzzeitig über eins, weshalb hier mit Bauteilverantwortlichen abgeklärt werden muss, ob es zu weiteren Problemen im System gekommen ist.

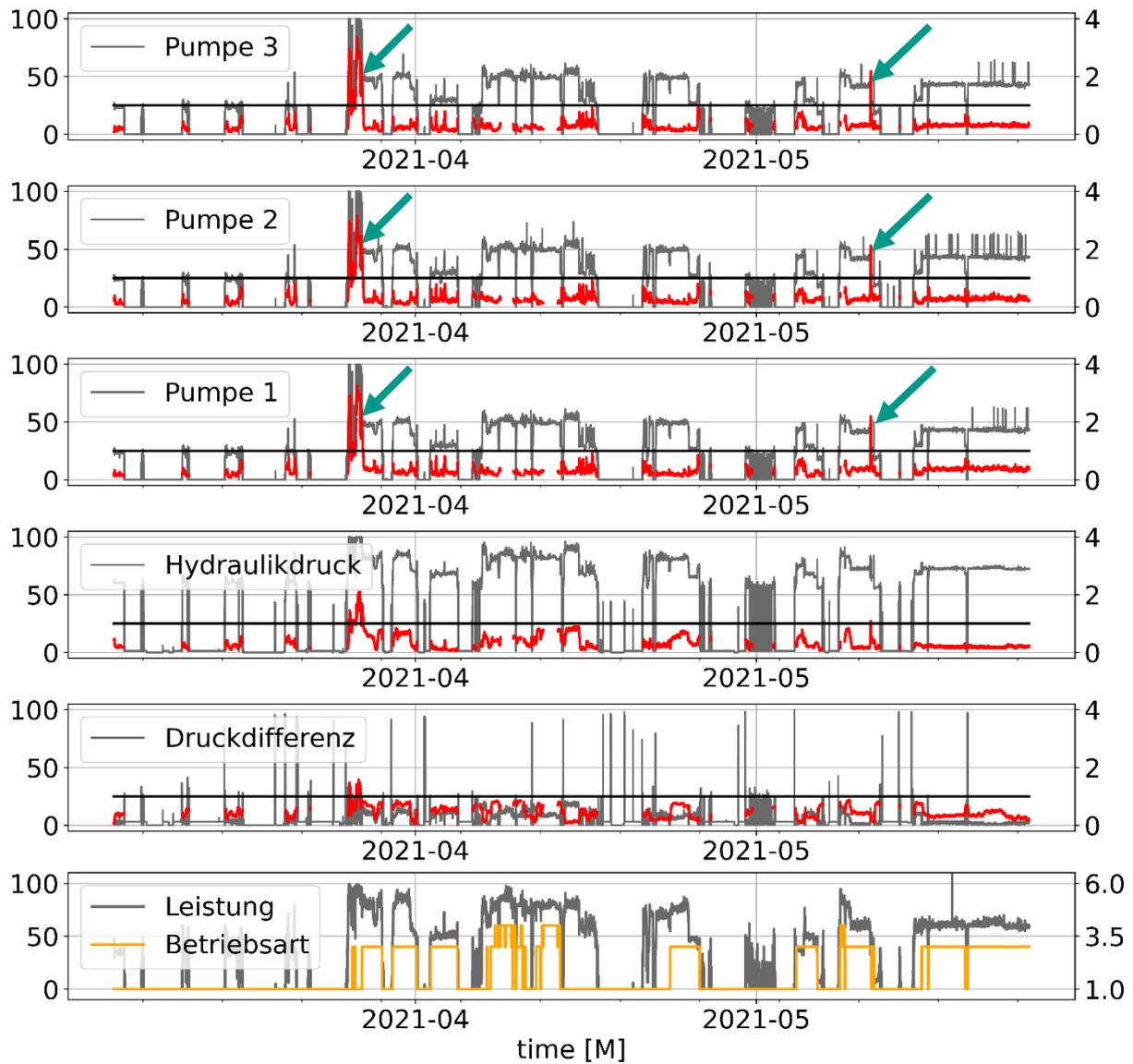


Abbildung 6-16: Auswertung der Fehlerkennung III-IV-A am Fahrprofil 4

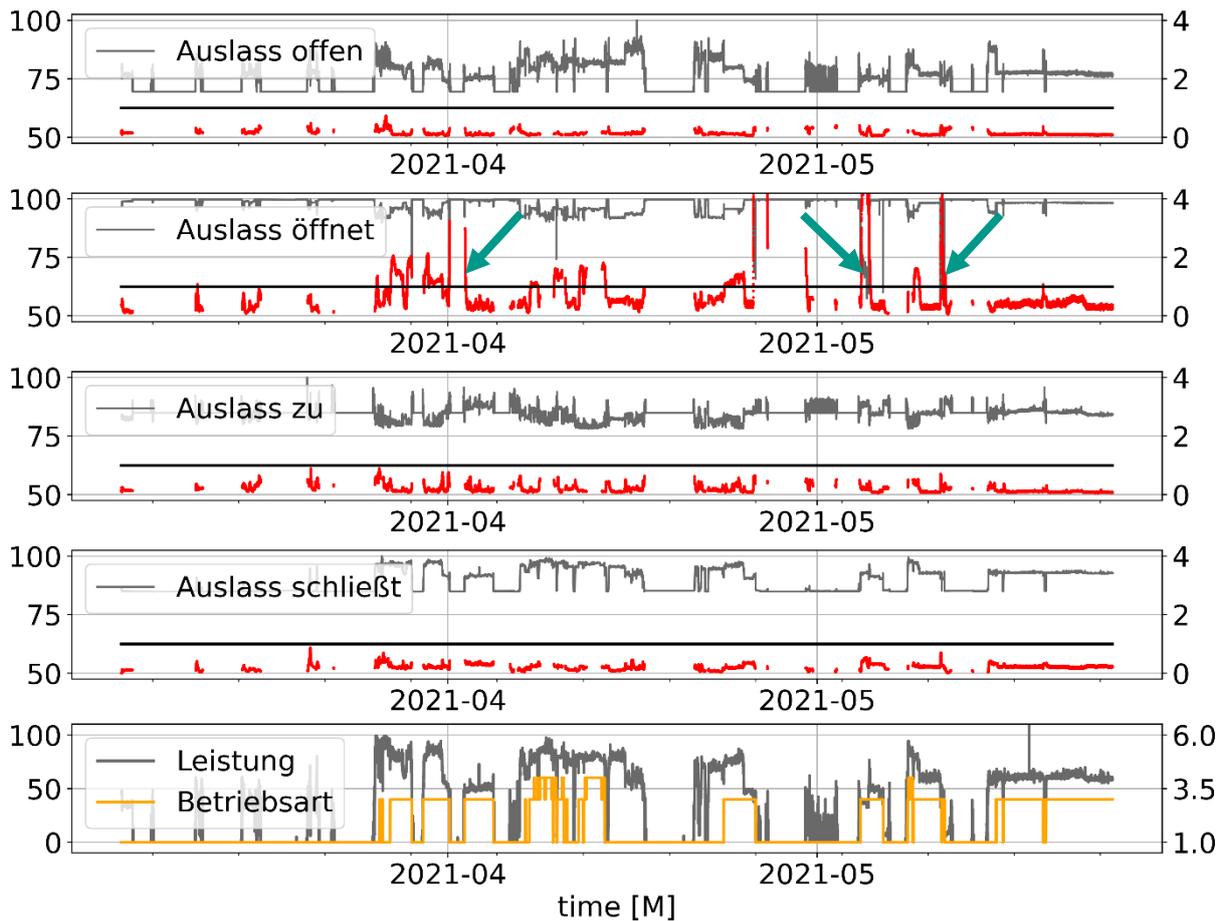


Abbildung 6-17: Auswertung der Fehlererkennung III-IV-B am Fahrprofil 4

Grundsätzlich kann gesagt werden, dass sich die vorgestellte Netzarchitektur eignet, um komplexe und multidimensionale Zusammenhänge abzubilden und auf Plausibilität zu überprüfen. Zum einen werden dabei eindeutige Fehler entdeckt, welche aufgrund der Vielzahl an Signalen, die im Antriebsstrang gemessen werden, oftmals übersehen oder in manuellen Fehlererkennungen nicht bedacht werden. Zum anderen werden Auffälligkeiten detektiert, die eine genaue Überprüfung erfordern, da das menschliche Auge oder Grenzwertbetrachtungen dort auf den ersten Blick nicht von einem Problem ausgehen. Analysen von Bauteilexperten können hier helfen, Fehler zu plausibilisieren und die Modellgenauigkeit weiter zu verbessern.

Für den Einsatz der vorgestellten Modelle im Steuergerät sind weitere Überlegungen für die Berechnung des Rekonstruktionsfehlers notwendig, da eine Warnmeldung überflüssig ist, wenn die Anomalieberechnung z.B. nur sehr kurzzeitig über einem Wert von eins liegt. Hier können integrale Betrachtungen der Fehlersignale helfen, den Grad einer Anomalie zu quantifizieren.



Da die Architektur bei allen vier Modellen ähnlich ist, lässt sich die Methodik gut auf beliebig viele weitere Subsysteme erweitern, wenn die entsprechende Datengrundlage geschaffen wird. Neben den vorgestellten thermischen, mechanischen und hydraulischen Systemen ist auch die Überwachung von digitalen Systemen denkbar, wie z.B. für Programmabläufe im Steuergerät. Bei Softwareupdates kann so sichergestellt werden, dass Funktionen und Methoden in der gewohnten Reihenfolge aufgerufen werden und dabei deren übliche Ausgabe erzeugt wird.



Diskussion

Grundsätzlich konnte gezeigt werden, dass sich neuronale Netzwerke eignen, komplexe Simulationen von mechanischen und thermodynamischen Systemen im Bereich der Antriebsstrangentwicklung um ein Vielfaches zu beschleunigen. Mit Hilfe der dargelegten Methoden für die Datenvorbereitung und -zusammenstellung sowie der passenden Netzarchitektur, kann die Abbildungsgenauigkeit dabei ähnliche Genauigkeiten wie die physikbasierte Simulation erreichen. Für eine Modellierung von periodischen und zyklischen Vorgängen eignen sich kontinuierliche Netzwerke (vgl. Kapitel 2.1.6), bei deren Aufbau die Zeitachse am Ausgang über die Neuronen der letzten Schicht abgebildet wird.

Bei den Vorhersagen am Beispiel von verschiedenen kurbelwinkelaufgelösten Kenngrößen im Brennraum während des Hochdruckteils der Verdichtung eines vorgemischten und fremdgezündeten H₂-Brennverfahrens liegt die Modellgültigkeit bei 500-facher Vorhersagebeschleunigung in einem Bereich von 99.81 %, bei denen die Abweichungen gegenüber der Simulation unter 4 % liegen (vgl. Kapitel 4.3). Somit eignen sich die Modelle gut für Prognosen in Echtzeit oder während der Applikation.

Während des Belastungstests der neuronalen Netzwerke aus Kapitel 4, bei dem die Startbedingungen für „Einlass schließt“ fünf Millionen Mal zufällig initialisiert werden, hat sich herausgestellt, dass die Abbildung der Vorgänge im Brennraum bei einer sehr kurzen Brenndauer, bedingt durch die hohe laminare Flammgeschwindigkeit im Stöchiometrischen bei der Verwendung von H₂ als Brenngas, eine feiner aufgelöste Ausgangsschicht benötigt. Hier reicht eine in 1° KW aufgelöste Zeitachse nicht, um die hohen Gradienten zwischen zwei Neuronen abzubilden, und es werden zusätzliche Neuronen in der Ausgangsschicht vorgeschlagen, um die Vorgänge z.B. in 0.1 °KW-Schritten zu modellieren. Dies geht allerdings mit einem erhöhten Rechenaufwand im Training sowie in den Vorhersagen einher, und es sollte vor einer erneuten Modellierung abgewägt werden, inwieweit die Abbildung derartiger Grenzphänomene mit Hilfe der neuronalen Netzwerke notwendig und sinnvoll ist. Weitere Verbesserungen durch eine Überarbeitung des Datensatzes werden eher ausgeschlossen, da die Druckverläufe bei Verbrennung mit extremer Brennrate trotz seltenem Vorkommen zufriedenstellend modelliert werden können.

Auch bei der Modellierung von zeitabhängigen Vorgängen, bei der ein Informationsfluss über vorangegangene Ereignisse ausschlaggebend für die Vorhersage aktueller Zeitschritte ist, konnten bei starker Simulationsbeschleunigung hohe Abbildungsgenauigkeiten von weniger als 2 % Unterschied im Vergleich zur Simulation erzielt werden (Tabelle 5-6 - Tabelle 5-10).



Anders als bei der kurbelwinkelaufgelösten Modellierung, bei der jeder Brennraumgröße ein eigenes Modell zugeordnet ist, werden hier mehrere charakteristische Größen eines Antriebsstrangs simultan vorhergesagt.

Interessant beim Training der zeitabhängigen LSTM-Zellen ist dabei, dass die Lernrate einen maßgeblichen Einfluss auf die Vorhersagegenauigkeit der transienten Effekte hat. Wird diese zu groß angesetzt, lassen sich teilweise nur binäre Minimallösungen für die Abbildung der Motorverhalten erzielen. Durch die aufwendige Datenvorbereitung und -zusammenstellung, bei der jeder möglich auftretende Leistungsbereich gleichmäßig verteilt wird, sind die erstellten neuronalen Netzwerke in der Lage, allgemeines Systemverhalten zu erlernen und geben keine vordefinierten Betriebsprofile wieder. Dies zeigt sich bei der Auswertung zufällig generierter und zusammengestellter Fahrprofile, in denen das Motorverhalten auch über längeren Zeitraum mit geringer Abweichung zur Simulation abgebildet wird.

Ein möglicher Ansatz für weiterführende Forschung ist herauszuarbeiten, ob die Schleusen der LSTM-Zellen, welche den zeitabhängigen Informationsfluss regeln, irgendwann Überlastungen oder Unregelmäßigkeiten bei der Abbildung von transienten Effekten erzeugen. Dafür können jahrelange und zusammenhängende Profile vorhergesagt werden und das Verhalten des Zellstatus genauer untersucht werden. Weiter sollte das Feintuning mit einer geringen Lernrate, welches für eine Erhöhung der Modellgenauigkeit angewandt wird, weiter untersucht werden. Lässt sich dieses in den Modellbetrieb implementieren, können neuronale Netzmodelle Alterungsprozesse oder Verschleiß im Antriebsstrang erlernen und so ihre Ausgabe stetig an den Systemzustand anpassen. Ebenfalls werden in dieser Arbeit bislang exemplarisch nur einige charakteristische Motorgrößen untersucht. Für die Überführung eines vollständigen Motormodells mit Luftpfad, Brennraum und Ladungswechsel sind aufwändigere Netzarchitekturen mit zusätzlichen Ein-, Zwischen- und Ausgangsgrößen notwendig.

Die Detektion von Unregelmäßigkeiten mit Hilfe von speziellen Encoder-Decoder Architekturen konnte bei verschiedenen Subsystemen eines realen Antriebsstrangs erfolgreich auf mehrere Ausgangsgrößen gleichzeitig angewandt werden. Eine Herausforderung liegt dabei im Umgang mit den echten Messdaten, welche gegenüber Simulationen oder frei zugänglichen und bereits aufbereiteten Datensätzen Sensorfehler enthalten. Zum anderen entsprechen reale Fahrprofile nicht zwangsläufig geeigneten Trainingsdaten für die Modellbildung, und diese müssen mit individuellen und vorgestellten Methoden zuerst zusammengestellt werden.

Mit der Einführung einer Kompressionsrate, welche das Verhältnis zwischen den Parametern am Latentvektor und den Ausgangsgrößen darstellt, lassen sich verschiedene Modelle von unterschiedlichen Subsystemen je nach Komplexität miteinander vergleichen.



Bei der Auswertung der einzelnen Anomalieerkennungen, bei der jedem Subsystem ein eigenes Modell zugeordnet ist, konnten Unregelmäßigkeiten und teilweise echte Schadensfälle gefunden und analysiert werden. Beispielsweise lassen sich Auffälligkeiten in der Kolbenbewegung sowie Rotationen am Kolbenring mit dem Modell der Brennraumwandtemperaturen entdecken. Weiter lassen sich hydraulische Systeme überwachen, in dem der Soll- und Istdruck sowie die Pumpenansteuerung über das Steuergerät vom Modell in Relation zueinander gestellt werden. Ein großer Vorteil besteht dabei darin, dass höherdimensionale Überlegungen häufig das menschliche Auffassungsvermögen übersteigen. Für drei oder vier Messgrößen lassen sich mehrdimensionale Regeln definieren, innerhalb derer sich Messwerte in Relation zueinander bewegen dürfen.

Bei der Überwachung von komplexen Subsystemen oder ganzen Antriebsträngen, in denen viele Messgrößen in Wechselwirkung zueinanderstehen, ist die Wahrscheinlichkeit hoch, einzelne Ereignisse mit der menschlichen Auffassung zu übersehen oder nicht bedacht zu haben. Hier können KI-basierte Überwachungen helfen, auch kleine oder schwer erkennbare Unregelmäßigkeiten festzustellen. Für eine serienreife Implementierung der erstellten neuronalen Netzwerke aus Kapitel 6 sind allerdings weitere Überlegungen für die Berechnung der Anomaliebewertung notwendig. Beispielsweise kann nicht für jedes Ereignis Alarm geschlagen werden, wenn der Anomaliewert über dem Grenzwert liegt. Hier sind integrale Betrachtungen denkbar, um die Schwere einer Unregelmäßigkeit herauszuarbeiten. Auch bedarf es bei gewissen Auswertungen noch Abklärungen mit den Bauteilverantwortlichen, um das Rekonstruktionsverhalten der Modelle weiter zu analysieren. Zwar wird bei den Modellierungen darauf geachtet, alle Betriebsbereiche im Training gleichmäßig verteilt zusammenzustellen, dennoch ist es denkbar, dass Anomalien auch aufgrund selten vorhandener Datenpunkte oder nicht bedateter Betriebszustände im Training ausgegeben werden. Beispielsweise ist ein Rückwärtslaufen des Antriebsstrang im Modell noch nicht berücksichtigt, und es kann davon ausgegangen werden, dass die Anomaliebewertung oder das Modell für einen solchen Fall entsprechend angepasst werden müssen.



Universität Stuttgart
Institut für Fahrzeugtechnik Stuttgart



Literaturverzeichnis

- [1] A. Fandakov, A Phenomenological Knock Model for the Development of Future Engine Concepts, Universität Stuttgart, Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik, Dissertation 2018, ISBN: 978-3-658-24875-8
- [2] A. Hanuschkin, S. Schober, J. Bode J, Schorr, B. Böhm, C. Krüger, S. Peters, Machine learning–based analysis of in-cylinder flow fields to predict combustion engine performance, International Journal of Engine Research, Volume 22 Issue 1, DOI: <https://journals.sagepub.com/doi/10.1177/1468087419833269>
- [3] A. M. Turing, Computing Machinery and Intelligence, Mind, New Series, Vol. 59, No. 236 (Oct., 1950), pp. 433-460, JSTOR, <http://www.jstor.org/stable/2251299>
- [4] A. Mirfendreski, Künstliche Intelligenz für die Entwicklung von Antrieben, Springer 2022, DOI: <https://doi.org/10.1007/978-3-662-63495-0>
- [5] A. P. Singh, P. S. Shukla, J. Hwang, A. K. Agarwal, Simulations and Optical Diagnostics for Internal Combustion Engines, Springer 2020, ISBN: 978-981-15-0335-1
- [6] A. Schmid, Virtuelle Optimierung und Applikation des Ottomotors durch 0D/1D-Simulation, Universität Stuttgart, Konstruktions-, Produktions- und Fahrzeugtechnik, Fakultät Maschinenbau, Dissertation 2011, ISBN: 9783816931744
- [7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang, Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. - 339 March 1989, DOI: 10.1109/29.21701
- [8] Airbus, Tensorflow, How Airbus Detects Anomalies in ISS Telemetry Data Using TFX, Blogeintrag, <https://blog.tensorflow.org/2020/04/how-airbus-detects-anomalies-iss-telemetry-data-tfx.html> (Stand 2020-04)
- [9] C. Bossung, Turbulenzmodellierung für quasidimensionale Motorprozessrechnung, Universität Stuttgart, Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik, Dissertation 2017, DOI: <http://dx.doi.org/10.18419/opus-9171>
- [10] C. Käding, E. Rodner, A. Freytag, J. Denzler, Fine-Tuning Deep Neural Networks in Continuous Learning Scenarios, Computer Vision – ACCV 2016 Workshops, Springer, DOI: https://doi.org/10.1007/978-3-319-54526-4_43
- [11] C. Olah, Understanding LSTM Networks, Google, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (Stand 2015-08)
- [12] D. Lejsek, A. C. Kulzer, A. Kufferath, G. Hohenberg, M. Bargende, Berechnung des Wandwärmeverlaufs im Motorhochlauf von DI-Ottomotoren, MTZ Motortech Z 71, 250–257 (2010), <https://doi.org/10.1007/BF03225562>



- [13] D. Rether, A. Schmid, M. Grill, M. Bargende, Quasi-Dimensional Modelling of CI-Combustion with Multiple Pilot- and Post-Injections, SAE Int. J. Engines 3(1):12-27, 2010, DOI: <https://doi.org/10.4271/2010-01-0150>
- [14] D. Rether, Modell zur Vorhersage der Brennrate bei homogener und teilhomogener Dieselerbrennung, Dissertation, IVK Universität Stuttgart 2012, DOI: <http://dx.doi.org/10.18419/opus-4506>
- [15] D. Rether, Il. Cimniak, M. Grill, M. Bargende, Nutzung neuronaler Netzwerke zur Virtualisierung am Beispiel von Brennraumvorgängen, Experten-Forum Powertrain: Simulation und Test 2020 (pp.65-85), DOI: http://dx.doi.org/10.1007/978-3-662-63606-0_5
- [16] D. Rether, Il. Cimniak, S. Bodza, M. Grill, M. Bargende, A new Class of AI-based Engine Models, Internationaler Motorenkongress 2021 (pp.267-282), DOI: http://dx.doi.org/10.1007/978-3-658-35588-3_16
- [17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. v. d. Driessche, T. Graepel, D. Hassabis, Mastering the game of Go without human knowledge, Nature volume 550, pp. 354–359 (2017), DOI: <https://doi.org/10.1038/nature24270>
- [18] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, 2017, DOI: <https://doi.org/10.48550/arXiv.1712.01815>
- [19] Diederik P. Kingma, Jimmy Lei Ba, Adam: A Method for Stochastic Optimization, Conference Paper at 3rd International Conference on Learning Representations (ICLR 2018), DOI: <https://doi.org/10.48550/arXiv.1412.6980>
- [20] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review (1958)., 65(6), 386–408. <https://doi.org/10.1037/h0042519>
- [21] FKFS, Anleitung UserCylinder 2.7.5, Handbuch 2022, <https://www.fkfs.de/kompetenzen/virtuelle-entwicklung/virtuelle-motorenentwicklung> (auf Anfrage)
- [22] G. E. Dahl, T. N. Sainath, G. E. Hinton, Improving Deep Neural Networks for LVCSR using Rectified Linear Units and Dropout, IEEE International Conference on Acoustic Speech and Signal Processing (IEEE Press, 2013), pp. 8609–8613, DOI: 10.1109/ICASSP.2013.6639346
- [23] G. Feng, C. Douligeris, Using Hopfield networks to solve traveling salesman problems based on stable state analysis technique, IJCNN 2000, pp. 521-526 vol.6, DOI: 10.1109/IJCNN.2000.859448
- [24] G. Philipp, D. Song, J. G. Carbonell, The Exploding Gradient Problem demystified - Definition, Prevalence, Impact, Origin, Tradeoffs, and Solutions, ICLR 2018, DOI: <https://doi.org/10.48550/arXiv.1712.05577>



- [25] G. Sgroi, G. Cornetti, D. Naber, M. T. Keskin, M. Grill, UserCylinder: Wasserstoff-Brennverfahren, FKFS 2021, https://www.fkfs.de/fileadmin/FKFS/2_Kompetenzen/2.2_Virtuelle_Entwicklung/2.2.3_Virtuelle_Motorenentwicklung/Simulation_H2_Brennverfahren.pdf
- [26] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, Sepp Hochreiter, Self-Normalizing Neural Networks, Conference Paper at 31st Conference on Neural Information Processing Systems (NIPS 2017), DOI: <https://doi.org/10.48550/arXiv.1706.02515>
- [27] Harro Heuser, Lehrbuch der Analysis 1+2, 17. Auflage, Teubner, ISBN: 978-3-8348-0777-9
- [28] I. N. Bronstein, K. A. Semendjajew, Taschenbuch der Mathematik, 8. Auflage, Harri Deutsch Verlag, ISBN: 978-3-8171-2008-6
- [29] I. Weidner, Big Data, Computerwoche, <https://www.computerwoche.de/a/daten-sind-das-neue-gold,3330102> (Stand 2017-08)
- [30] J. Gibson, B. Vlahov, D. Fan, P. Spieler, D. Pastor, A.-a. Agha-mohammadi, E. A. Theodorou, A Multi-step Dynamics Modeling Framework For Autonomous Driving In Multiple Environments, Mai 2023, DOI: <https://doi.org/10.48550/arXiv.2305.02241>
- [31] J. Moolayil, Lern Keras for Deep Neural Networks, Springer 2019, ISBN: 978-1-4842-4239-1
- [32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, Washington 2016, DOI: <https://doi.org/10.48550/arXiv.1506.02640>
- [33] J. Warnatz, U. Maas, R. W. Dibble, Combustion – Physical and Chemical Fundamentals, Modeling and Simulation, Experiments, Pollutant Formation, Springer, ISBN: 3-540-25992-9
- [34] J. Wooldridge, Introductory econometrics: A modern approach, 5. Auflage, Nelson Education, ISBN: 978-1-1115-3439-4
- [35] K. Xu, M. Zhang, J. Li, S. S. Du, K. Kawarabayashi, S. Jegelka, How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks, MIT 2021, DOI: <https://doi.org/10.48550/arXiv.2009.11848>
- [36] Kistler, Wassergekühlter PiezoStar Drucksensor, Handbuch 2021, <https://www.kistler.com/files/document/003-591d.pdf?callee=frontend>
- [37] L. Torgo, B. Pfahringer, R. P. Ribeiro, P. Branco, SMOTE for Regression, Conference Paper September 2013, DOI: 10.1007/978-3-642-40669-0_33
- [38] Landeshauptstadt Stuttgart, Amt für Umweltschutz, Abteilung Stadtklimatologie, https://www.stadtklima-stuttgart.de/stadtklima_filestorage/download/luft/S-Mitte-SZ-Jahres-Werte-1987-2021.zip
- [39] M. Bargende, Schwerpunkt-Kriterium und automatische Klingelerkennung. Bausteine zur automatischen Kennfeldoptimierung bei Ottomotoren, MTZ 1995 (Volume: 56), ISSN: 0024-8525



- [40] M. Briesch, D. Sobania, F. Rothlauf, The Randomness of Input Data Spaces is an A Priori Predictor for Generalization, KI 2022: Advances in Artificial Intelligence - 45th German Conference on AI 2022, Proceedings S. 17-28, ISBN: 978-3-031-15791-2
- [41] M. Grill, Zylindermodul. FVV-Vorhaben Nr. 869, Abschlussbericht, FVV, Frankfurt/Main, 2008
- [42] M. Grill., Objektorientierte Prozessrechnung von Verbrennungsmotoren, Dissertation, Universität Stuttgart 2006, DOI: <http://dx.doi.org/10.18419/opus-4076>
- [43] M. Hilgers, Dieselmotor, Springer 2016, ISBN: 978-3-658-14641-2
- [44] M. Meyer, Signalverarbeitung, Springer 2017, ISBN: 978-3-658-18320-2
- [45] M. Ullrich, R. Scharp, Nutzfahrzeugkolben für erhöhte mechanische und thermische Belastungen, MTZ-68 2007-06 S.454-458, DOI: <https://doi.org/10.1007/BF03227412>
- [46] M. Wenig, Simulation der ottomotorischen Zyklenschwankungen, Universität Stuttgart, Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik, Dissertation 2013, DOI: <http://dx.doi.org/10.18419/opus-4561>
- [47] N. Il. Chawla, L. O. Hall, K. W. Bowyer, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16 (2002) 321–357, DOI: <https://doi.org/10.48550/arXiv.1106.1813>
- [48] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, Il. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, D. Silver, Grandmaster level in StarCraft II using multi-agent reinforcement learning, Nature volume 575, pp. 350–354 (2019), DOI: <https://doi.org/10.1038/s41586-019-1724-z>
- [49] P. Engel, S. Meise, A. Rausch, W. Tegethoff, Modelling of Automotive HVAC Systems Using Long-Short-Term Memory Networks, ADAPTIVE 2019: The Eleventh International Conference on Adaptive and Self-Adaptive Systems and Applications, https://www.researchgate.net/publication/359550406_Data-Driven_Modeling_An_AI_Toolchain_for_the_Powertrain_Development_Process
- [50] P. J. Huber, Robust Estimation of a Location Parameter, Ann. Math. Statist. 35: 73-101 (1964). DOI: <https://doi.org/10.1214%2Faoms%2F1177703732>
- [51] P. Kidger, On Neural Differential Equations, Ph.D. Thesis Oxford 2021, DOI: <https://doi.org/10.48550/arXiv.2202.02435>
- [52] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, ICML 2016, https://www.researchgate.net/publication/304758073_LSTM-based_Encoder-Decoder_for_Multi-sensor_Anomaly_Detection



- [53] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, Long Short-Term Memory Networks for Anomaly Detection in Time Series, ESANN 2015 proceedings, ISBN: 978-287587014-8
- [54] R. Koether, Taschenbuch der Logistik, 5. aktualisierte Ausgabe 2008, ISBN: 9783446457676
- [55] R. Nikbakht, A. Jonsson, A. Lozano, Unsupervised Learning for Cellular Power Control, IEEE Communications Letters (Volume: 25, Issue: 3, March 2021), DOI: <https://doi.org/10.1109/LCOMM.2020.3027994>
- [56] R. O. Mueller, Modernes Motormanagement mit Neuronalen Netzen, Dissertation Würzburg 1998, ISSN: 1432-8801
- [57] R. W. Freund, R. H. W. Hoppe, Stoer/Bulirsch: Numerische Mathematik 1, 10. Auflage, Springer, ISBN: 978-3-540-45389-5
- [58] Raman Arora, Amitabh Basu, Poorya Mianjy, Anirbit Mukherjee, Understanding Deep Neural Networks with Rectified Linear Units, Conference Paper at 6th International Conference on Learning Representations (ICLR 2018), DOI: <https://doi.org/10.48550/arXiv.1611.01491>
- [59] S. Hann, M. Grill, M. Bargende, Laminare Flammengeschwindigkeit - Abmagerung und Hochlast-Abgasrückführung in der Motorsimulation, MTZ 79 (2018), Nr. 4, S. 28-35, DOI: <https://doi.org/10.1007/s35146-018-0004-z>
- [60] S. Hann, S. Palaveev, M. Grill, M. Veltman, M. Bargende, Predicting the Influence of Charge Air Temperature Reduction on Engine Efficiency, CCV and NOx-Emissions of a Large Gas Engine Using a SI Burn Rate Model, SAE Technical Paper 2020, DOI: 10.4271/2020-01-0575
- [61] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997), S. 1735-1780, DOI: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [62] S. Jeschke, Global Trends in Transport Routes and Goods Transport: Influence on Future International Loading Units, Discussion Paper - 16th Acea SAG Meeting - June 2011, https://www.acea.auto/files/SAG_16_Global_Trends_in_Transport_Routes_Goods_Transport.pdf
- [63] S. Kullmann, KI und DOE im Einsatz bei der Kalibrierung von elektrifizierten Antriebssträngen, Webinar, 2020, <https://www.avl.com/web/guest/-/ki-und-doe-im-einsatz-bei-der-kalibrierung-von-elektrifizierten-antriebsstraengen>
- [64] S. Milojevic, S. Bodza, D. Rether, M. Grill, M. Bargende, Next Generation Data Based Models powered by AI, 9th Int. Symposium on Development Methodology 10. Nov. 2021, <https://www.avl.com/documents/1982862/38398200/Conference+Program+9th+Symposium+on+Development+Methodology.pdf>
- [65] S. Milojevic, S. Bodza, II. Cimniak, M. Angerbauer, M. Grill, M. Bargende, Data-Driven Modeling: An AI Toolchain for the Powertrain Development Process, WCX SAE World Congress Experience, <http://dx.doi.org/10.4271/2022-01-0158>



- [66] S. Skansi, Introduction to Deep Learning, Springer (2018), ISBN: 978-3-319-73004-2
- [67] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory, January 1967, DOI: 10.1109/TIT.1967.1053964
- [68] T. Koch, E. Eßer, S. Kureti, A. Sousa, H₂-DeNO_x-Katalysator für H₂-Verbrennungsmotoren, MTZ-81 2020-06 S.33-38, DOI: <https://doi.org/10.1007/s35146-020-0234-8>
- [69] Tensorflow API Reference, Long Short-Term Memory Layer, https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM (Stand 2022-10)
- [70] Tensorflow API Reference, Time Series Forecasting, https://www.tensorflow.org/tutorials/structured_data/time_series (Stand 2022-10)
- [71] W. Bibel, R. Kruse, Prof. Dr. B. Nebel, Computational Intelligence (2011), Springer, ISBN: 978-3-658-10903-5
- [72] W. Huang, X. Wu, J. Liu, X. Bai, Dynamics of Deepwater Riser, Springer 2022, ISBN: 978-981-16-2887-7
- [73] W. S. Mc. Culloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics 5, 115–133 (1943). DOI: <https://doi.org/10.1007/BF02478259>
- [74] WinGD, Marine Installation Manual (MIM) X72DF, Oktober 2021, <https://www.wingd.com/en/engines/engine-types/x-df-dual-fuel/x72df/>
- [75] WinGD, Marine Installation Manual (MIM) X92DF, Oktober 2021, <https://www.wingd.com/en/engines/engine-types/x-df-dual-fuel/x92df/>
- [76] X. Ying, An Overview of Overfitting and its Solutions, IOP Conf. Series: Journal of Physics: 1168 (2019), DOI: 10.1088/1742-6596/1168/2/022022
- [77] Y. Lin, Y. Yu, G. Guan, Research on energy efficiency design index for sea-going LNG carriers, Journal of Marine Science and Application, DOI: <https://doi.org/10.1007/s11804-014-1282-6>
- [78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, DOI: <https://doi.org/10.48550/arXiv.1706.03762>



Anhang

7 Parametrierung Zylindermodul

Zylindermodul:

[GENERAL]

Rechenstart (grdKW) = -163

Rechenende (grdKW) = 145

Auflösung (grdKW) = 0.1

Auflösung_homBrenn (grdKW) = 0.1

Auflösung HRD (grdKW) = 1

Feste Schrittweite = yes

Faktor Zusatzspeicher (-) = 5

Drehzahl (1/min) = 1800

Release = FKFS

Takt (-) = 4

Savelter = no

max. Iterationszahl = 200

InterpolHrd = yes

Lmin (-) = 34.4110000000000

Hu (kJ/kg) = 119940.0000000000000

[#IniGeneral]

active = yes

Pseudo-Zonen = 0

speedup = no

Systeme = 1

SysMehrzonig = 1

Emissions-Spezies = no

Wassereinspritzung = yes

m_leer (mg) = 1

Zonen = 2

[#Startwerte]

active = yes

System = 1

FullCycle = no

II_Start (ccm) = -1

D_flag = no

m_Luft (mg) = 1000



m_Krst (mg) = -1
m_H2O_extern_ES (mg) = 0
x_AGR (%) = 5
x_H2O_ES (%) = -1
Lambda (-) = 1
T_Start (K) = 320
p_Start (bar) = -1

[#SpeichernZyl]

active = yes
System = 1
UV-Zone = 1
II-Zone = 2
m_Luft (mg) = -1
m_Krst (mg) = -1

[#VolumenFktStd]

active = yes
System = 1
ZOT (grdKW) = 0
Hub (mm) = 150.000000000000
Bohr (mm) = 130.000000000000
pII (mm) = 273.000000000000
desachs (mm) = 0.600000000000
eps = 11.750000000000
Kolben-OT = yes
realStroke = yes

[#KaloZyli]

active = yes
System = 1
Spezies = 11
T_freeze (K) = 1800
GG_mode = Burcat
Quasi2zone = auto
C_Krst (-) = 0.000000000000
H_Krst (-) = 2.000000000000
O_Krst (-) = 0.000000000000
N_Krst (-) = 0.000000000000

[#Entrainment]

active = yes



System = 1
UV-Zone = 1
II-Zone = 2
save = yes
Bohrh (mm) = -1
max_Hub (mm) = -1
max_Eps (-) = -1
interpolGeo = no
kompatib = no
rel_KW = no
fac_c (-) = 1
exp_n (-) = 1
ZZP (grdKW) = 0.000000000000
fac_Taylor (-) = 15.000000000000
SparkSize (mm) = 3.000000000000
ZK_hor (mm) = 6.000000000000
ZK_ver (mm) = 4.000000000000
importGeo = no
Geo-Datei = d:\Aflamme.hrd
CombEff (-) = 1.000000000000
min_sL_ZZP (m/s) = 0.00000000
a_ZZP (-) = -1
IT_c_grenz (-) = 1.000000000000
Schwankungsfaktor (-) = 1
Obere_Zuendgrenze (-) = 10
C_u (-) = 3
Entflammungsdauer (grdKW) = 0.000000000000
Entflammungsmodell = no
Hann Erweiterungen = yes
GeoDetailliert = no
Kraftstoff = Wasserstoff
intL_fac (-) = 0.022222
sT_fac (-) = 3
tau_fac (-) = 12
lokale TKE modellieren = yes
lokTKE_a (-) = 0.6
lokTKE_m (-) = 0.3
Max. rel. TKE (-) = 1.2
TDCclearance lokale TKE (mm) = -1

[#Turbulenz_QDM_Lbw2]
active = yes



System = 1
phi_start (grdKW) = -100000
Angepasste Brennraumgeometrie = no
h_d (mm) = 13
delta_phi_05 (grdKW) = 45
II_Mulde (ccm) = 0
d_Mulde (mm) = 0
Kolbengeometrie = no
C_tau (-) = 0.2
C_tau_Inflow_k (s^(1/3)) = 0.019
alpha_Inflow_k (-) = 0.667
C_tau_in (-) = 0.08
C_tau_H (-) = 2
C_Pis (-) = 1
a_f_Inflow_k (-) = 0.6
b_f_Inflow_k (-) = 0.2
C_Inflow (-) = 1
C_mu (-) = 0.09
f_T_spin_up (-) = 1
C_T (-) = 1
C_S (-) = 1
C_T_fric (-) = 1
C_S_fric (-) = 1
C_eps1 (-) = 5.5
C_eps2 (-) = 5.5
C_k_prod_SI (-) = 0
Ladungswechsel = no
Tumblezahl-Datei = def
Spalte Ventilhub Tumblezahl = h
Spalte Tumblezahl = T
Drallzahl-Datei = def
Spalte Ventilhub Drallzahl = h
Spalte Drallzahl = S
IdStr_Einlass_all = Einlass
IdStr_Einlass1 = ign
IdStr_Einlass2 = ign
Interne AGR = no
IdStr_Auslass1 = ign
IdStr_Auslass2 = ign
IdStr = def
HrdStr = def
ErwAusgabe = no



Ekin_Inflow_eff_IVC (kg*m²/s²) = 0
Ekin_Inflow_eff_1_IVC (kg*m²/s²) = 0.002554737485
Ekin_Inflow_eff_2_IVC (kg*m²/s²) = 0.002415070005
L_I_IVC (m) = 0
L_I_1_IVC (m) = 0.010279147743
L_I_2_IVC (m) = 0.010271998965
m_I_IVC (kg) = 0
m_I_1_IVC (kg) = 0.000002254717
m_I_2_IVC (kg) = 0.000002158205
L_Vol_IVC (m) = 0.160301514640
ekin_Pis_sec_IVC (m²/s²) = 0.000000000000
ekin_bowl_IVC (m²/s²) = 0.000000000000
C_k_IVC (-) = 0.008
I_IVC (-) = 0.000000000000
k_Turb_IVC (m²/s²) = 17.744497158150
d_EV (mm) = -1
h_EV (mm) = -1
n_EV (-) = 2.000000000000
f_dEV_B (-) = 0.4
Lambda_L (-) = 1
T_eng_IVC (-) = 0.000000000000
S_eng_IVC (-) = 1.945435365952
n_Spritzlöcher (-) = 6
d_Spritzlöcher (mm) = 0.125
Dichte Kraftstoff (l) (kg/m³) = 690
C_E (-) = 0.3
C_E_in (-) = 0.06
C_eps_DI (-) = 0.04
C_inj (-) = 1.000000000000
k_Turb_Inj_IVC (m²/s²) = 0.000000000000
Ekin_Inj_eff_IVC (kg*m²/s²) = 0.000000000000
Injektor Name = ign
Injektorlage x (mm) = 0
Injektorlage z (mm) = 0
Einspritzrichtung (grd) = 90
C_T_DI (-) = 1
C_lin (-) = 25
C_a (-) = 0.75
Lrd Datei = no

[#WandwZylinder]
active = yes



System = 1
UV-Zone = 1
Aufteiltyp = Entrainment
II_Dach (ccm) = 0
A_Kopf (cm²) = 0
II_Mulde (ccm) = 0
A_Kolben (cm²) = 0
h_Fst (mm) = 12.000000000000
HtMultipller (-) = 1.000000000000
HtMultipllerHead (-) = 1
HtMultipllerLiner (-) = 1
HtMultipllerPiston (-) = 1
aHead (-) = 1.000000000000
aPiston (-) = 1.300000000000
TdcClearance (mm) = 1.000000000000
analytischer Feuersteg = yes
b_Fst (mm) = 0.5
IdStr Leakage = ign
Vorgabe_Wandtemperaturverlauf = no
Dateiname = ign
T_Kopf (K) = 500
T_Buchse (K) = 430
T_Kolben (K) = 450

[#AlphaWoschni]

active = yes
System = 1
UV-Zone = 1
save = yes
Huber = no
C1 LW (-) = 6.18
C1_dk LW (-) = 0.417
C1 (-) = 2.28
C1_dk(-) = 0.308
C2 (-) = 0.00324
Polytrogenexpo (-) = 1.37
DRZUT (-) = 0
Original-Huber = no

[#Zeldovich]

active = yes
System = 1



UV-Zone = 1
T_freeze (K) = 1800
x_AGR (%) = \$Startwerte.x_AGR (%)
InfoStr = def
AusgabenAktiv = yes
Multipliert (-) = 1.000000000000
A1v = 4.93
A1r = 1.6
A2v = 1.48
A2r = 1.25
A3v = 4.22
A3r = 6.76
B1v = 0.0472
B1r = 0
B2v = 1.5
B2r = 1.612
B3v = 0
B3r = -0.212
E1v = 75.590
E1r = 0
E2v = 5.680
E2r = 37.690
E3v = 0
E3r = 49.340
II-Zone = 2
NO-Start (mol) = -1

[#LoeschLeereZonen]
active = yes

[#LamBrenn_FKFS]
active = yes
System = 1
UV-Zone = 1
save = no
BrennTyp = Wasserstoff
BeimischTyp = ign
Beimischung (Vol-%) = 0.000000000000
O_Krst (-) = 0.000000000000
M_Krst (g/mol) = 2.018000000000

[#AIFT]



active = yes
System = 1
UV_Zone = 1
Lambda_F (-) = -1
xR_zero = no
Ignoriere dQ_B = yes
InfoStr = Abwaertskompatibel

[#KaloKrstZyli]

active = yes
MolKrst (g/mol) = -1
a1_low = 40783.2321
a2_low = -800.918604
a3_low = 8.21470201
a4_low = -0.01269714457
a5_low = 1.753605076E-5
a6_low = -1.20286027E-8
a7_low = 3.36809E-12
a1_high = 560812.801
a2_high = -837.150474
a3_high = 2.975364532
a4_high = 0.00125229124
a5_high = -3.74071619E-7
a6_high = 5.936625E-11
a7_high = 3.61E-15
T_Grenz (K) = 1000



8 Modellübersicht

8.1 Druckverlauf I-I

Eingang:

Drehzahl, Luftmasse, Lambda, Rate AGR, Starttemperatur, Zündzeitpunkt, Temperatur Kopf, Temperatur Kolbenbuchse, Temperatur Kolben

Ausgang:

Druckverlauf von -163 bis 145 °KW bezogen auf ZOT

Netzparameter:

L1: 100 Neuronen, L2: 200 Neuronen, L3: 400 Neuronen, Aktivierung: ReLu, Ausgang: 308 Neuronen, Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss)

8.2 Druckgradientenverlauf I-II

Eingang:

Drehzahl, Luftmasse, Lambda, Rate AGR, Starttemperatur, Zündzeitpunkt, Temperatur Kopf, Temperatur Kolbenbuchse, Temperatur Kolben

Ausgang:

Druckgradientenverlauf von -163 bis 145 °KW bezogen auf ZOT

Netzparameter:

L1: 100 Neuronen, L2: 200 Neuronen, L3: 400 Neuronen, Aktivierung: ReLu, Ausgang: 308 Neuronen, Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss)

8.3 Brennverlauf I-III

Eingang:

Drehzahl, Luftmasse, Lambda, Rate AGR, Starttemperatur, Zündzeitpunkt, Temperatur Kopf, Temperatur Kolbenbuchse, Temperatur Kolben

Ausgang:

Brennverlauf von -163 bis 145 °KW bezogen auf ZOT

Netzparameter:

L1: 100 Neuronen, L2: 200 Neuronen, L3: 400 Neuronen, Aktivierung: ReLu, Ausgang: 308 Neuronen, Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss)



8.4 Wandwärmeverlauf I-IV

Eingang:

Drehzahl, Luftmasse, Lambda, Rate AGR, Starttemperatur, Zündzeitpunkt, Temperatur Kopf, Temperatur Kolbenbuchse, Temperatur Kolben

Ausgang:

Wandwärmeverlauf von -163 bis 145 °KW bezogen auf ZOT

Netzparameter:

L1: 100 Neuronen, L2: 200 Neuronen, L3: 400 Neuronen, Aktivierung: ReLu, Ausgang: 308 Neuronen, Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss)

8.5 Vollmotormodell II-II

Eingang:

Drehzahl, Leistung

Ausgang:

Abgastemperatur, Laderdrehzahl, Spüldruck

Netzparameter:

L1: 200 LSTM (return_state=True, return_sequences=True), L2: 200 LSTM (return_state=True, return_sequences=True), L3: 200 LSTM (return_state=False, return_sequences=False), Aktivierung: TanH, Ausgang: 3 Neuronen, Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-4, Abbruchkriterium: 25 (val_loss), Zeitfenster: 180 min

8.6 Kolbenwandtemperaturen (AD) III-I

Eingang:

Drehzahl, Leistung, Betriebsmodus, Umgebungstemperatur, Anzahl aktiver Injektoren, Kolbenwandtemperatur A, B, C

Ausgang:

Kolbenwandtemperatur A, B, C

Netzparameter:

E1: 32 LSTM, E2: 16 LSTM, E3: 8 LSTM, Latenvektor: 1, D1: 8 LSTM, D2: 16 LSTM, E3: 32 LSTM, Aktivierung: TanH, Ausgang: 3 Neuronen (time-distributed), Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss), Zeitfenster: 60 min, (return_state=False, return_sequences=True) bei allen LSTM



8.7 Verbrennung (AD) III-II

Eingang:

Drehzahl, Leistung, Betriebsmodus, Umgebungstemperatur, Verbrennungsdruck, Spüldruck, Abgastemperatur

Ausgang:

Verbrennungsdruck, Spüldruck, Abgastemperatur

Netzparameter:

E1: 64 LSTM, E2: 32 LSTM, E3: 16 LSTM, Latenvektor: 3, D1: 16 LSTM, D2: 32 LSTM, E3: 64 LSTM, Aktivierung: TanH, Ausgang: 3 Neuronen (time-distributed), Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss), Zeitfenster: 60 min, (return_state=False, return_sequences=True) bei allen LSTM

8.8 Einspritzung (AD) III-III

Eingang:

Drehzahl, Leistung, Betriebsmodus, Umgebungstemperatur, Pumpenansteuerung 1-4, Raildruck Setpoint, Raildruck Istdruck, Differenzdruck

Ausgang:

Pumpenansteuerung 1-4, Raildruck Setpoint, Raildruck Istdruck, Differenzdruck

Netzparameter:

E1: 128 LSTM, E2: 64 LSTM, E3: 32 LSTM, Latenvektor: 7, D1: 32 LSTM, D2: 64 LSTM, E3: 128 LSTM, Aktivierung: TanH, Ausgang: 7 Neuronen (time-distributed), Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss), Zeitfenster: 60 min, (return_state=False, return_sequences=True) bei allen LSTM



8.9 Hydraulik (AD) III-IV

Eingang:

Drehzahl, Leistung, Betriebsmodus, Umgebungstemperatur, Einlass öffnet, Einlass offen, Auslass öffnet, Auslass offen, Pumpenansteuerung 1-3, Hydraulik-Solldruck, Hydraulik-Istdruck, Druckdifferenz

Ausgang:

Einlass öffnet, Einlass offen, Auslass öffnet, Auslass offen, Pumpenansteuerung 1-3, Hydraulik-Solldruck, Hydraulik-Istdruck, Druckdifferenz

Netzparameter:

E1: 128 LSTM, E2: 64 LSTM, E3: 32 LSTM, Latenvektor: 10, D1: 32 LSTM, D2: 64 LSTM, E3: 128 LSTM, Aktivierung: TanH, Ausgang: 10 Neuronen (time-distributed), Aktivierung: linear, Fehlerfunktion: Huber, Lernrate: 10e-3, Abbruchkriterium: 25 (val_loss), Zeitfenster: 60 min, (return_state=False, return_sequences=True) bei allen LSTM



9 Zusätzliche Darstellungen

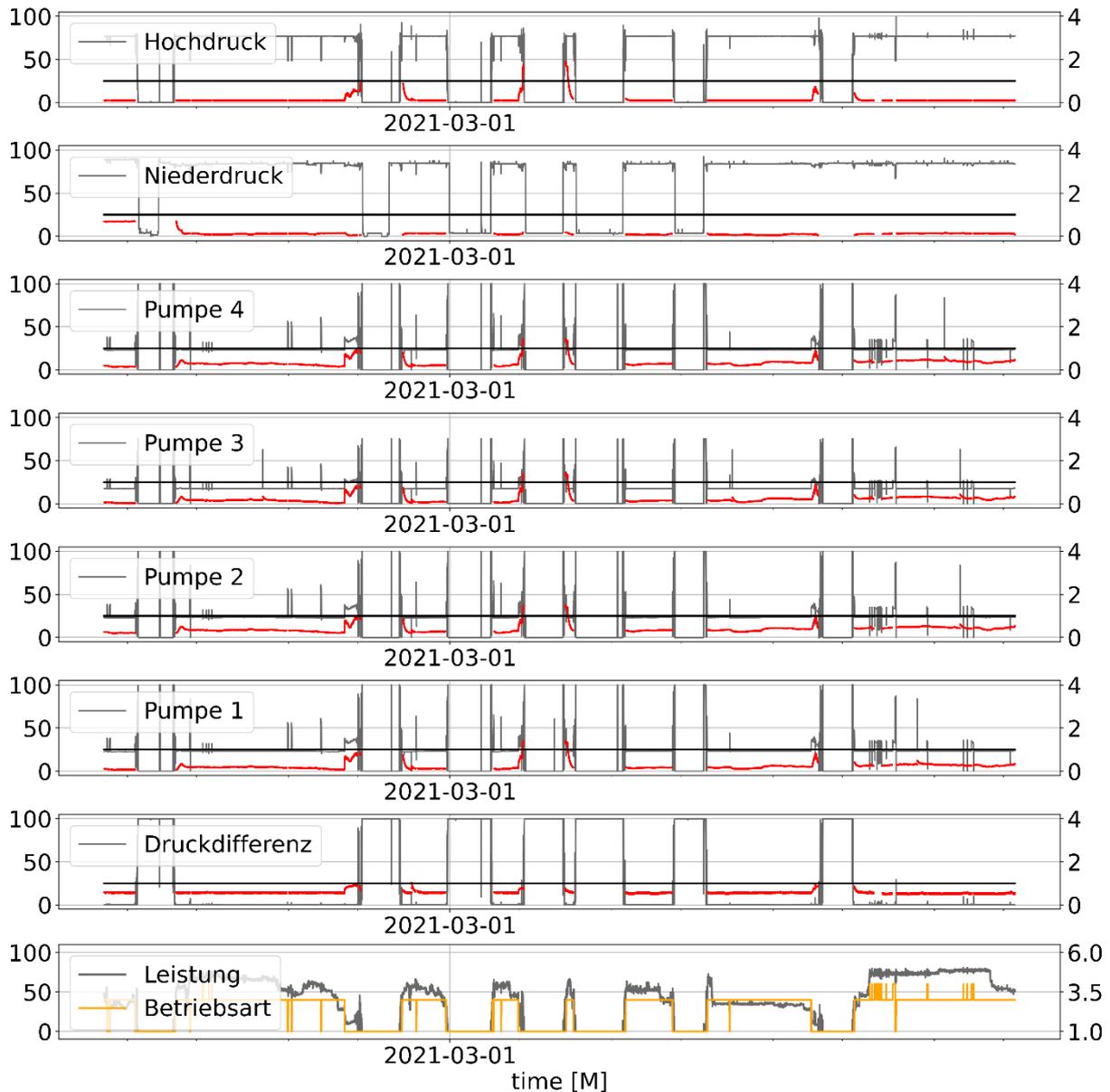


Abbildung 9-1: Auswertung ohne Anomalien der Fehlerkennung III-III

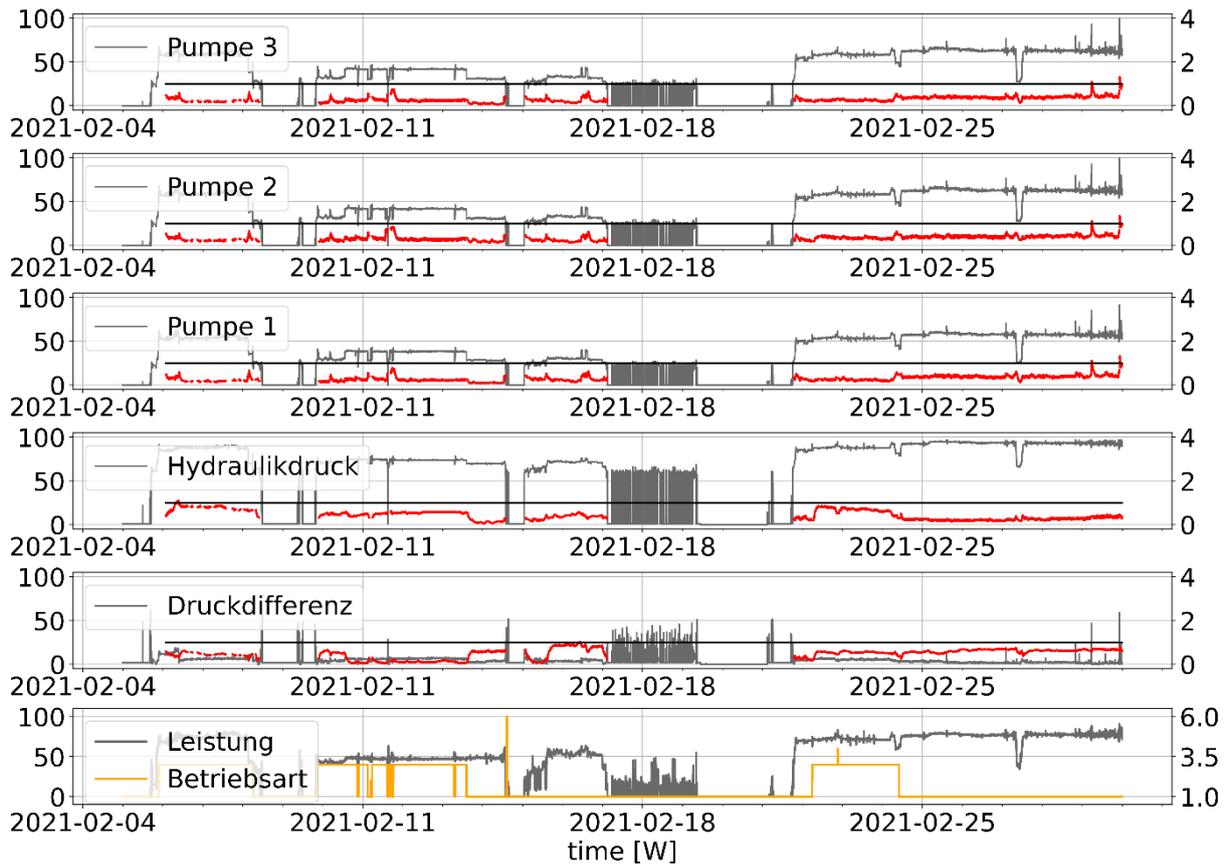


Abbildung 9-2: Auswertung ohne Anomalien der Fehlerkennung III-IV-A

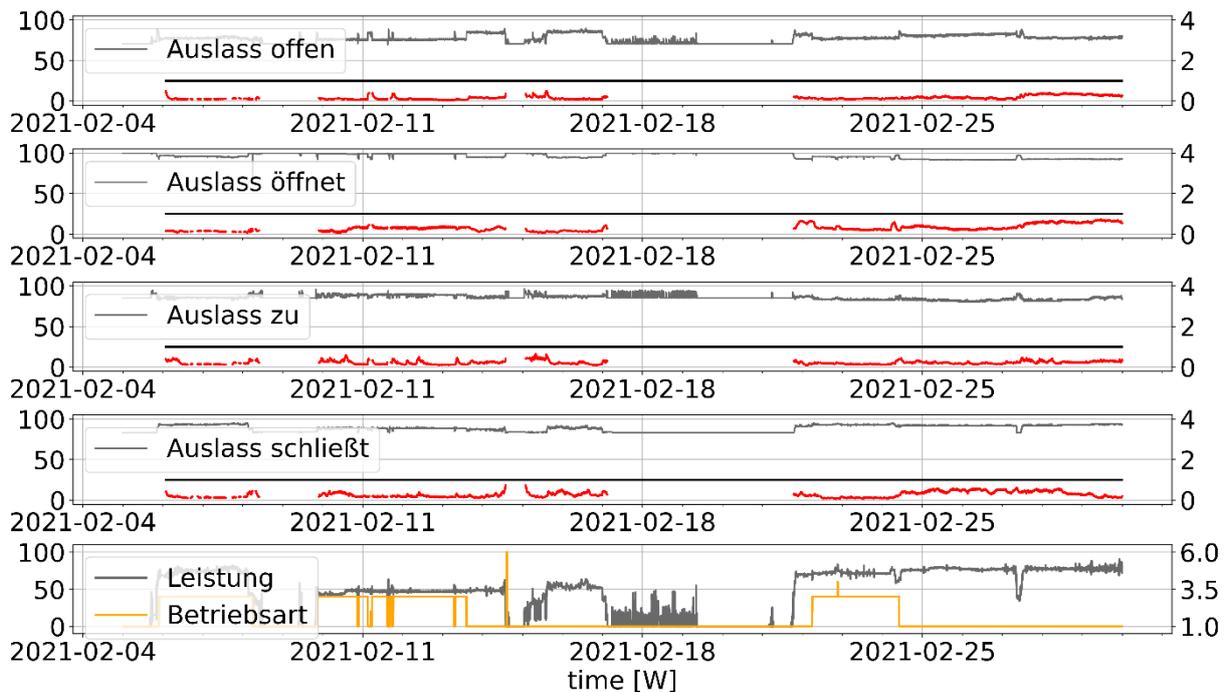


Abbildung 9-3: Auswertung ohne Anomalien der Fehlerkennung III-IV-B