

Implementierung eines Vorhersage- und Optimierungsalgorithmus in eine Datenschnittstelle für Smart Meter Gateways (SmartMinCO₂)

Verfasser:

Forschungsgesellschaft

Heizung-Lüftung-Klimatechnik Stuttgart mbH

Joel Kimmich, M.Sc.

Matthias Hartnagel, M.Sc.

Dipl.-Ing. Armin Ruppert

Forschungsvorhaben 2021-01

gefördert vom

Verein der Förderer der Forschung im Bereich

Heizung-Lüftung-Klimatechnik Stuttgart e.V.

Verein der Förderer
der Forschung im Bereich
Heizung • Lüftung • Klimatechnik
Stuttgart e.V.



Januar 2022

Vorwort und Danksagung

An dieser Stelle möchten wir uns recht herzlich beim Verein der Förderer der Forschung im Bereich Heizung-Lüftung-Klimatechnik Stuttgart e.V. für die Förderung des Forschungsvorhabens bedanken. Daneben gilt auch der Tecomon GmbH unser Dank für die Unterstützung bei der Entwicklung des Applikationsmodells. In diesem Rahmen ist eine studentische Arbeit [1] betreut worden, deren Ergebnisse in den vorliegenden Endbericht eingeflossen sind.

Kurzfassung

Die zunehmende Nutzung von erneuerbaren Energieträgern bewirkt eine fluktuierende Stromerzeugung in Abhängigkeit von Wind und Solarstrahlung. Zur Stabilisierung des Netzbetriebs müssen daher zukünftig vermehrt Maßnahmen auch auf der Verbraucherseite ergriffen werden, wie beispielsweise die Verschiebung von Lasten in Zeiten mit erhöhtem Stromangebot mittels Speicher und individueller Regelung. Ein Smart-Meter-Gateway soll zwischen Netzbetreiber und Gebäude bzw. Anlagentechnik zur Wärme- und Kälteübertragung als Datenschnittstelle dienen, um solche Regelungskonzepte auch für Eigenheime zu ermöglichen.

Hierzu wird im Forschungsvorhaben ein intelligentes Regelungskonzept zur Be- und Entladung von Energiespeichern mit dem Ziel der Umverteilung von Lasten in Python entwickelt. Dafür wird ein künstliches neuronales Netz mit Simulations- und Messdaten trainiert, um Heizlasten eines Beispielgebäudes für mehrere Stunden bis Tage vorherzusagen. Diese modellprädiktiven Daten werden einem hierfür erstellten Optimierer übergeben, der anhand von Marktsignalen als Kostenfunktion einen Zeitplan zur Beladung eines thermischen Energiespeichers erstellt. Wird als Marktsignal beispielsweise der Anteil fossiler Energieträger im Strommix verwendet, kann durch das entwickelte Applikationsmodell die individuelle CO₂-Emission bei der Stromerzeugung minimiert werden, indem der Speicher in Zeiten mit niedrigen CO₂-Emissionen im Netz beladen und bei hohen Emissionen entladen wird. Zur Validierung des entwickelten Programms werden Model-in-the-Loop-Simulationen durchgeführt. Diese basieren auf Simulations- und Messdaten als Heizlasten sowie realen Wetterdaten und bilden ein Jahr ab. Als Vergleichsbasis dient ein bedarfsorientiertes Modell ohne Energiespeicher.

Die Simulationen weisen die Funktionalität des Applikationsmodells nach und zeigen für das Jahr 2019 ein Einsparpotential von 9,5 % der CO₂-Emissionen respektive 19 % der Stromkosten für ein Einfamilienhaus mit 200 m² beheizter Wohnfläche nach EnEV 2014-Standard. Bei Berücksichtigung einer Nachtabsenkung erhöht sich die CO₂-Einsparung um 50 %.

Abstract

The increasing use of renewable energy sources causes fluctuating power generation depending on wind and solar radiation. In order to stabilize grid operation, it will therefore be necessary in the future to take more measures on the consumer side as well, such as shifting loads at times of increased electricity supply by means of storage and individual control. A Smart-Meter-Gateway is to serve as a data interface between the grid operator and the building or system technology for heat and cold transmission in order to enable such control concepts for homes as well.

For this purpose, an intelligent control concept for the charging and discharging of energy storage units with the aim of redistributing loads is being developed in Python in the research project. For this purpose, an artificial neural network is trained with simulation and measurement data to predict heating loads of an example building for several hours to days. This model-predictive data is passed to an optimizer created for this purpose, which uses market signals as a cost function to generate a schedule for loading a thermal energy storage unit. For example, if the share of fossil fuels in the electricity mix is used as the market signal, the developed application model can minimize individual CO₂ emissions during electricity generation by loading the storage unit during periods of low CO₂ emissions in the grid and unloading it when emissions are high. Model-in-the-loop simulations are performed to validate the developed program. These are based on simulation and measurement data as heating loads as well as real weather data and represent one year. A demand-oriented model without energy storage serves as a basis for comparison.

The simulations prove the functionality of the application model and show a savings potential of 9.5 % of CO₂ emissions and 19 % of electricity costs, respectively, for a single-family house with 200 m² of heated living space according to EnEV 2014 standard for the year 2019. If a night setback is taken into account, the CO₂ saving increases by 50 %.

Inhaltsverzeichnis

Kurzfassung

Abstract

Bildverzeichnis **III**

Tabellenverzeichnis **V**

Nomenklatur **VI**

Abkürzungsverzeichnis **IX**

1	Einleitung	1
1.1	Einleitung	1
1.2	Ziele dieser Studie	2
2	Grundlagen	3
2.1	Wärmeübertragung und Thermodynamik	3
2.1.1	Berechnung von Wärmeströmen	3
2.1.2	Wärmestrahlung	4
2.2	Thermische Gebäude- und Anlagensimulation	5
2.3	Smart-Meter-Gateways	5
2.4	Optimierung	6
2.5	Funktionsweise neuronaler Netze	7
2.5.1	Perzeptron	7
2.5.2	Mehrschichtige neuronale Netze	8
2.5.3	Training	9
2.5.4	Rekurrente neuronale Netze und die LSTM-Zelle	9
2.5.5	Normalisierung	11
3	Applikationsmodell	13
3.1	Beispielgebäude und Umgebungsbedingungen	14
3.2	Vorhersage	15
3.2.1	Aufbau neuronales Netz	15
3.2.2	Datenaufbereitung	16
3.2.3	Training	18
3.2.4	Vorhersage für SMGW und Simulation	19
3.3	Optimierung	20
3.3.1	Beschreibung des Problems	20
3.3.2	Aufbau des Optimierers	21
3.3.3	Extremfälle der Optimierung	23
3.4	Speichermodell	24
3.5	Model-in-the-Loop-Simulation	25
4	Ergebnisse und deren Auswertung	28
4.1	Einstellungen des neuronalen Netzes	28

4.2	Eingabeparameter	29
4.3	Simulationsergebnisse	33
4.3.1	CO ₂ -Ausstoßreduzierung	33
4.3.2	Stromkostenreduzierung	34
4.3.3	Kosten und CO ₂ -Ausstoß beider Optimierungen	34
4.3.4	Einfluss Trainingsdatenzeitraum	36
4.3.5	Realdaten-Simulation	37
5	Zusammenfassung	40
6	Literatur	42
	Anhang	46
A	Verwendete Software und deren Versionen	46
B	Randbedingungen für die Gebäudesimulation	47
C	Eingabeparameterzuordnung	48

Bildverzeichnis

Bild 2-1	Funktionsweise Perzeptron	7
Bild 2-2	Beispiel eines Fully-Connected Netzes	8
Bild 2-3	Darstellung des Trainingsverlaufs mit Bereich des Übertrainings	10
Bild 2-4	Aufwärmen und autoregressives Vorgehen bei LSTM-Zellen	11
Bild 3-1	Grafische Darstellung des Applikationsmodells	13
Bild 3-2	Verwendetes Beispielgebäudes	14
Bild 3-3	Klassische und gleichmäßige Aufteilmethoden im Vergleich für den Parameter „Elevation“	17
Bild 3-4	Werte nach der Normalisierung mit dem Median (grob gestrichelte Linie) und der Standardabweichung von ausgewählten Parametern	18
Bild 3-5	Darstellung des Moving-Window-Verfahrens	18
Bild 3-6	Vorhersage mit der LSTM-Zelle	20
Bild 3-7	Flussdiagramm des Optimierers	22
Bild 3-8	Heizlastverlauf des simulierten Beispielgebäudes und die gemessenen, in stündliche Auflösung reduzierten Messdaten des Realgebäudes für die Kalenderwoche 50 im Jahr 2019	26
Bild 4-1	Vorhersagegenauigkeit bei verschiedenen Einstellungen während des Trainings über die LSTM-Unit-Anzahl	29
Bild 4-2	Vorhersagegenauigkeit bei verschiedenen Eingabeparametern	31
Bild 4-3	Vorhersagegenauigkeit bei verschiedenen Eingabeparametern aufbauend auf der Vorhersage einzig mit der flächenbezogenen Heizlast (Modell X01 = A01)	32
Bild 4-4	Vorhersagegenauigkeit bei verschiedenen Eingabeparametern aufbauend auf dem (Basis-)Modell X27 mit den Parametern: flächenbezogene Heizlast; Globalstrahlung; positive Elevation und Windgeschwindigkeiten Nord und Ost	32
Bild 4-5	CO ₂ -Ausstoß bei der Optimierung hinsichtlich hohen Strommix-Anteils an Wind- und Solarkraft. Bedarfsorientiert ergeben sich 15,8 kg CO ₂ /(m ² ·a); optimiert 14,3 kg CO ₂ /(m ² ·a)	34
Bild 4-6	Kosten bei der Optimierung hinsichtlich geringen Börsenstrompreises. Bedarfsorientiert ergeben sich 4,7 €/ (m ² ·a); optimiert 3,8 €/ (m ² ·a)	35
Bild 4-7	Stromkosten bei beiden Optimierungen im Vergleich. Bedarfsorientiert ergeben sich 4,7 €/ (m ² ·a); CO ₂ -optimiert 4,2 €/ (m ² ·a); Kosten-optimiert 3,8 €/ (m ² ·a)	35
Bild 4-8	CO ₂ -Ausstoß bei beiden Optimierungen im Vergleich. Bedarfsorientiert ergeben sich 15,8 kg CO ₂ /(m ² ·a); Kosten-optimiert 14,9 kg CO ₂ /(m ² ·a); CO ₂ -optimiert 14,3 kg CO ₂ /(m ² ·a)	35
Bild 4-9	Auswirkung der Trainingszeit auf das Simulationsergebnis bei der Optimierung hin zu geringem CO ₂ -Ausstoß	36

Bild 4-10	Auswirkung der Trainingszeit auf das Simulationsergebnis bei der Optimierung hin zu geringen Stromkosten	37
Bild 4-11	CO ₂ -Ausstoß bei der Optimierung hinsichtlich hohen Strommix-Anteils an Wind- und Solarkraft mit verwendeten Realdaten. Bedarfsorientiert ergeben sich 8,8 kg CO ₂ /(m ² ·a); optimiert 7,8 kg CO ₂ /(m ² ·a); optimiert verbessert 7,7 kg CO ₂ /(m ² ·a)	38
Bild 4-12	Stromkosten bei der Optimierung hinsichtlich geringen Börsenstrompreises mit verwendeten Realdaten. Bedarfsorientiert ergeben sich 3,2 €/ (m ² ·a); optimiert 2,1 €/ (m ² ·a)	38

Tabellenverzeichnis

Tabelle 3-1	Für das Applikationsmodell und in der Gebäudesimulation berücksichtigte Umgebungsparameter	15
Tabelle 3-2	Berechnung der Energien im Optimierer	23
Tabelle 3-3	Notwendige Parameter für das Speichermodell	24
Tabelle 4-1	Verwendete Trainings- und LSTM-Einstellungen	29
Tabelle 4-2	Mögliche Eingangsparameter und deren Zuordnung zu den Modellen A01 bis A26	30
Tabelle 4-3	Verwendete Emissionsfaktoren für das Jahr 2019 [31]	33
Tabelle 4-4	Vorhersagefehler bei verschiedenen Trainingsdatensatzlängen während der Simulation	36
Tabelle 4-5	Vorhersagefehler der mit den Realdaten trainierten neuronalen Netze	38
Tabelle A-1	Verwendete Software	46
Tabelle A-2	Installierte Python-Packages mit Versionsnummern Teil 1	46
Tabelle A-3	Installierte Python-Packages mit Versionsnummern Teil 2	47
Tabelle B-1	Thermische Eigenschaften des Beispielgebäudes nach [27]	47
Tabelle C-1	Eingangsparameterzuordnung zu den Modellen B01 bis B32 und den Basismodellen X	48
Tabelle C-2	Eingangsparameterzuordnung zu den Modellen B33 bis B50 und den Basismodellen X	49

Nomenklatur

Lateinische Notation

Symbol	Beschreibung	Einheit
A_{gk}	atmosphärische Strahlung bei wolkenlosem Himmel	W/m ²
A_{gw}	atmosphärische Strahlung bei bedecktem Himmel	W/m ²
A_g	atmosphärische Strahlung	W/m ²
A	Fläche	m ²
$area$	spezifischer Bereich innerhalb des Optimierers	-
\vec{b}_L	Bias-Vektor einer Schicht L	-
b_i	Bias eines Neurons N an Stelle i	-
b_N	Bias eines Neurons N	-
c_1	Schwarzkörperkonstante 1	W/m ²
c_2	Schwarzkörperkonstante 2	m·K
c_p	spezifische Wärmekapazität	J/(kg·K)
D	Teiler	-
\dot{e}_{spez}	spezifische Strahlung	W/m ²
J_{mse}	Kostenfunktion, die den mittleren quadratischen Fehler verwendet	-
j	Anzahl betrachteter Werte	-
\vec{k}	Kostenfunktionswertvektor	-
L	Fehler eines neuronalen Netzes	-
\mathcal{L}	Verlustfunktion	-
m	Anzahl von Datenpunkten	-
m	Masse	kg
n	Anzahl von Zeitschritten der Vorhersage	-
N	Bedeckungsgrad	-
\vec{o}	Ausgabevektor	-
\vec{o}_1	erster Ausgabevektor	-
\vec{o}_2	zweiter Ausgabevektor	-
\vec{o}_i	Ausgabevektor einer Schicht i	-
\vec{o}_t	t-ter Ausgabevektor	-
$P_{Erzeuger}$	Leistung Energieerzeuger	W
p_H	Wasserdampfpartialdruck	hPa
P_{Haus}	arithmetisches Mittel der Leistung des Gebäudes eines Zeitschrittes	W
\vec{P}_{Haus}	Vektor mit Heizlastvorhersage des Gebäudes	-
Q	Wärme	J
Q_0	Initialbeladung Speicher	J
Q_{area}	Speicherbeladung in $area$	J
Q_B	Beladungswärme	J
Q_{Haus}	Wärmebedarf des Gebäudes	J
$Q_{i\text{ Erzeuger}}$	erzeugte Energie in Zeitschritt i	J
$Q_{i\text{ Haus}}$	Wärmebedarf des Gebäudes in Zeitschritt i	J
$Q_{i\text{ Speicher}}$	Beladung des Speichers im Zeitschritt i	J
$Q_{i\text{ Verlust}}$	Verlustenergie des Speichers in Zeitschritt i	J
Q_{max}	Maximalbeladungswärme	J
Q_s	Speicherbeladung in Schritt s	J

Symbol	Beschreibung	Einheit
Q_{soll}	erforderliche Wärmemenge in <i>area</i>	J
$Q_{\text{Speicher max}}$	Maximalwärmemenge des Speichers	J
Q_{Speicher}	Wärmemenge im Speicher	J
\dot{Q}	Wärmestrom	W
s	Stelle des Minimums der Kostenfunktion im Optimierer	-
T_{dSt}	Taupunkttemperatur in Stationshöhe	K
T_{K}	Temperatur des thermischen Körpers	K
T_{St}	Lufttemperatur in Stationshöhe	K
t	Anzahl an Ausgaben	-
U	Wärmedurchgangskoeffizient	W/(m ² ·K)
$(UA)_{\text{sa}}$	Wärmeverlustkoeffizient	W/K
$w_{\text{N}1}$	Gewicht eines Neutrons N der Stelle 1	-
$w_{\text{N}2}$	Gewicht eines Neutrons N der Stelle 2	-
$w_{\text{N}d}$	Gewicht eines Neutrons N der Stelle d	-
\vec{w}_i	Gewichtsvektor einer Schicht i	-
\vec{w}_{N}	Gewichtsvektor eines Neutrons N	-
\mathbf{W}	optimale Gewichtsverteilung	-
$\widehat{\mathbf{W}}_{\text{L}}$	Gewichtsmatrix einer Schicht L	-
$\widehat{\mathbf{W}}$	Gesamtheit aller Gewichtsmatrizen und Bias-Vektoren eines neuronalen Netzes	-
x_1	Eingangsvektorwert an Stelle 1	-
x_2	Eingangsvektorwert an Stelle 2	-
x_3	Eingangsvektorwert an Stelle 3	-
$x_{\text{N}1}$	Eingangsvektorwert eines Neutrons N an Stelle 1	-
$x_{\text{N}2}$	Eingangsvektorwert eines Neutrons N an Stelle 2	-
$x_{\text{N}d}$	Eingangsvektorwert eines Neutrons N an Stelle d	-
\vec{x}	Eingangsvektor	-
\vec{x}_i	Eingangsvektor einer Schicht i	-
\vec{x}_{N}	Eingangsvektor eines Neutrons N	-
\vec{x}_{w}	Eingangsvektor mit Wetterdaten	-
y	Wert eines betrachteten Parameters	-
\bar{y}	arithmetisches Mittel	-
\hat{y}	Ausgabewert	-
$\vec{\hat{y}}$	Ausgabevektor	-
y_i	zu normalisierender Wert	-
y_{norm}	normalisierter Wert	-
\vec{y}	Sollwertvektor	-
z	Lösungswert des Optimierers	-
\vec{z}	Lösungsvektor des Optimierers	-

Griechische Notation

Symbol	Beschreibung	Einheit
α	Absorptionskoeffizient	-
Δt	Zeitschrittlänge	s
ΔT	Temperaturdifferenz	K
ΔV	Versatz	-
ε	Emissionskoeffizient	-
ϑ	Systemtemperatur	°C
ϑ_{amb}	Umgebungstemperatur	°C
ϑ_{Bezug}	Bezugstemperatur	°C
λ_{W}	Wellenlänge	m
ϱ	Reflexionskoeffizient	-
σ	Stefan-Boltzmann-Konstante	W/(m ² ·K ⁴)
τ	Transmissionskoeffizient	-
φ	Aktivierungsfunktion	-

Abkürzungsverzeichnis

API	Programmierschnittstelle (engl. Application Programming Interface)
DWD	Deutscher Wetterdienst
EE	erneuerbare Energie
EnEV	Energieeinsparverordnung
ENTSO-E	Verband Europäischer Übertragungsnetzbetreiber (engl. European Network of Transmission System Operators for Electricity)
LSTM	Long Short-Term Memory
MIL	Model-in-the-Loop
MOSMIX	Model Output Statistics-MIX
MPC	modellprädiktive Regelung (engl. Model Predictive Control)
MSE	mittlerer quadratischer Fehler (engl. Mean Squared Error)
NN	neuronales Netz
PV	Photovoltaik
RMSE	Wurzel des mittleren quadratischen Fehlers (engl. root mean squared error)
RNN	rekurrentes neuronales Netz
SMGW	Smart-Meter-Gateway

1 Einleitung

1.1 Einleitung

Konventionelle Energieerzeuger wie Kohle- und Kernkraftwerke werden im Zuge der von der Bundesregierung beschlossenen Energiewende sukzessive durch erneuerbare Energien (EE) ersetzt [2], deren Kohlenstoffdioxidemissionen im Betrieb vernachlässigbar sind. Die größten elektrischen Erzeuger in Deutschland werden voraussichtlich Windkraft- und Photovoltaikanlagen sein. Die Möglichkeit der anderen EE wie Wasserkraft sind durch geografische Begebenheiten begrenzt. Dadurch ändert sich die Stromerzeugung von nachfragegesteuert zu angebotsgesteuert, da die Erzeugung aus Wind und Sonne im Gegensatz zur konventionellen Stromerzeugung nicht steuerbar ist.

Eine Folge ist die Notwendigkeit von Lastumverteilungen aus Zeiten mit geringem Angebot und hoher Nachfrage in Zeiten mit hohem Angebot und geringer Nachfrage. Ein Indikator für angebotsreiche Stunden ist der Strompreis, der – gesteuert durch Angebot und Nachfrage – bei höherer Energieerzeugung als Energienutzung sinkt. Somit haben Verbraucher einen wirtschaftlichen Vorteil durch das Umverteilen von Lasten hin zu Zeiten niedrigen Energiepreises. Das kann z. B. durch Zwischenspeichern und eine zeitversetzte Verwendung realisiert werden. Für Strompreiseinsparungen wird das Durchreichen der Strompreise von Strompreisbörse zu Verbraucher durch den Energieerzeuger bzw. Netzbetreiber vorausgesetzt.

Die Bundesregierung hat 2016 mit dem Messstellenbetriebsgesetz (MsbG) [3] Maßnahmen zur Möglichkeit der Datenkommunikation zwischen Stromverbraucher und -erzeuger ergriffen. Dafür sollen Smart-Meter-Gateways (SMGWs), als intelligente Stromzähler bekannt, eingesetzt werden. Diese dienen als Datenschnittstelle zwischen den Akteuren im Stromnetz und ermöglichen es dem Netzbetreiber bzw. -erzeuger, das Stromangebot dem Verbrauch anzugleichen. Zusätzlich kann ein SMGW als Recheneinheit verwendet werden. Das ermöglicht neue Regelungsstrategien, die Angebot und Nachfrage von Strom berücksichtigen und die oben erwähnten Herausforderungen in der Lastumverteilung bewerkstelligen können. Ein Stromnetz, das mit SMGWs ausgerüstet ist, wird zu einem „Smart Grid“, engl. für „Intelligentes Netz“. Dieser Begriff steht für die automatische Anpassung eines jeden Verbrauchers im Stromnetz an spezielle Anforderungen (wie z. B. Lastumverteilung) und der Energieerzeugung.

Die Umverteilung von Lasten ist ein multiparametrischer Prozess und erfordert eine Optimierung, um eine hohe Effizienz des Systems erreichen zu können. Dabei müssen z. B. Nutzerkomfort, Stromangebot, Speichermöglichkeit und Speicherverluste berücksichtigt werden. Auch die Minimierung des CO₂-Ausstoßes kann Ziel der Optimierung sein. Bisher existiert noch keine Umsetzung der Lastumverteilung mittels SMGW. Vor diesem Hintergrund wird in dieser Arbeit ein Applikationsmodell zur Lastumverteilung mittels eines Speichers erstellt und dessen Funktionsfähigkeit durch Model-in-the-Loop-(MIL)-Simulationen validiert.

1.2 Ziele dieser Studie

Diese Studie befasst sich mit der Erstellung eines Applikationsmodells für ein Smart-Meter-Gateway (SMGW). Das umgesetzte Programm beruht auf einem Ansatz mit modellprädiktiver Regelung und anschließender Optimierung unter Berücksichtigung der Stromnetzauslastung. Dadurch können Lasten umverteilt und ein netzdienlicher Betrieb gewährleistet werden. Das Applikationsmodell soll eine Vorhersage durch ein künstliches neuronales Netz (NN) erstellen, mit der ein Optimierer die Zeitpunkte bestimmen kann, während denen der thermische Energiespeicher des Gebäudes beladen wird. Für die Vorhersage wird das NN mit gegebenen Gebäude- und Wetterdaten trainiert. Zusätzlich werden im Betrieb automatisiert die aktuelle Wetterprognose und Strommarktdaten abgerufen. Diese werden in die Vorhersage und Optimierung miteinbezogen. Als Kostenfunktion für den Optimierer werden minimale Stromkosten und ein maximaler Anteil erneuerbarer Energien im Stromnetz für eine Reduktion des gebäudespezifischen CO₂-Ausstoßes untersucht.

Eine Model-in-the-Loop-(MIL)-Simulation ermöglicht es, das Applikationsmodell auf seine Funktion hin zu testen und zu bewerten. Dafür wird ein Gebäude- und Speichermodell erstellt. Die Heizlastverläufe des Beispielgebäudes werden durch eine Simulation in der Software TRNSYS erstellt und sowohl für das Training des NNs als auch für die MIL-Simulation verwendet. Als Vergleichsbasis dient ein bedarfsorientiertes Regelverfahren. Außerdem wird das NN auf die Qualität der Vorhersage hin untersucht.

2 Grundlagen

2.1 Wärmeübertragung und Thermodynamik

Das in Kapitel 3 und nachfolgend beschriebene Gebäude- und Speichermodell ist von thermischem Verhalten beeinflusst, das auf Grundlagen der Wärmeübertragung und Thermodynamik basiert. Da diese Themen weitreichend sind, wird hier ein Überblick der für die nachfolgenden Kapitel notwendigen Gesetzmäßigkeiten gegeben. Eine ausführliche Beschreibung dieser Grundlagen findet sich in [4, 5, 6].

2.1.1 Berechnung von Wärmeströmen

Ein Wärmestrom \dot{Q} (in W) von einem System zur Umgebung über eine definierte Systemgrenze berechnet sich mit einem Wärmedurchgangskoeffizienten U (in $W/(m^2 \cdot K)$) durch eine wärmeübertragende Fläche A (in m^2) nach

$$\dot{Q} = U \cdot A \cdot (\vartheta - \vartheta_{\text{amb}}) = U \cdot A \cdot \Delta T \quad (2-1)$$

mit

ϑ = Temperatur des Systems in °C

ϑ_{amb} = Umgebungstemperatur in °C

ΔT = Temperaturdifferenz in K

Bei einem negativen Wert wird dem System Wärme zugeführt (Systemtemperatur ϑ ist geringer als Umgebungstemperatur ϑ_{amb}), bei einem positiven Wert wird Wärme abgeführt. Der Wärmedurchgangskoeffizient hängt von der Wärmeleitfähigkeit der Materialien der Systemgrenze, der auftretenden Wärmeübertragungsarten an Grenzschichten und der Form der Systemgrenze ab. Die Bestimmung ist empirisch, weil die auftretenden Strömungen im Fluid um das wärmeübertragende Element sich nach den Gesetzmäßigkeiten der Navier-Stokes-Gleichungen ohne bekannte analytische Lösung verhalten. Die Komplexität des Themas wird durch den Umfang des VDI-Wärmeatlas deutlich, der eine Sammlung von empirischen Gleichungen zur Wärmeübertragung enthält [6]. Einen einfacher zu bestimmenden Wert stellt der Wärmeverlustkoeffizient $(UA)_{\text{sa}}$ (in W/K) dar, der spezifisch für jedes Bauteil gemessen werden kann und Berechnungen vereinfacht. Dessen Bestimmung wird in DIN EN 12977-3 erläutert [7]. Die Berechnung des Wärmestroms vereinfacht sich zu:

$$\dot{Q} = (UA)_{\text{sa}} \cdot (\vartheta - \vartheta_{\text{amb}}) = (UA)_{\text{sa}} \cdot \Delta T \quad (2-2)$$

Die Wärme Q (in J) stellt die Menge an Wärme dar, die im Vergleich zu einer Bezugstemperatur ϑ_{Bezug} (in °C) in dem System gespeichert ist. Hierbei entspricht ϑ_{Bezug} der Temperatur, bei dem das System keine nutzbare Wärme enthält. Oft wird als Bezugstemperatur 0°C verwendet. Die Berechnung von Q erfolgt nach

$$Q = c_p \cdot m \cdot (\vartheta - \vartheta_{\text{Bezug}}) = c_p \cdot m \cdot \Delta T_{\text{Bezug}} \quad (2-3)$$

mit

m = Masse des Systems in kg

c_p = spezifische Wärmekapazität der Masse des Systems in J/(kg·K)

ΔT_{Bezug} = Temperaturdifferenz zur Bezugstemperatur in K

Im Fall von Wasser als Medium ist die spezifische Wärmekapazität temperaturabhängig.

2.1.2 Wärmestrahlung

Wärmestrahlung wird elektromagnetische Strahlung mit einem Wellenlängenbereich von 0,1 μm ... 1000 μm genannt, die jeder Körper über dem absoluten Nullpunkt ausstrahlt. Damit ist Wärmestrahlung die einzige Wärmeübertragungsform, die keine Wechselwirkung von benachbarten Molekülen in einem Trägermedium beinhaltet und im Vakuum funktioniert. Die bekannteste Wärmestrahlung ist die Solarstrahlung der Sonne.

Die Wärmestrahlung eines Körpers hängt von seinen Oberflächeneigenschaften ab und der Fähigkeit, Strahlung zu absorbieren, reflektieren und transmittieren. Diese Eigenschaften werden mit den Werten Absorptionskoeffizient α , Reflexionskoeffizient ρ und Transmissionskoeffizient τ beschrieben, die den Anteil der jeweiligen Strahlung an der gesamten einfallenden Strahlung angeben [8]. Das Kirchoff'sche Gesetz erlaubt es, eine Beziehung zwischen Absorption und Emission eines thermischen Körpers im Gleichgewicht herzustellen [9] und besagt, dass Emissionskoeffizient ε und Absorptionskoeffizient gleichzusetzen sind:

$$\alpha = \varepsilon \quad (2-4)$$

Ein idealisierter Grenzfall eines thermischen Körpers ist der schwarze Strahler, der jegliche Strahlung absorbiert und somit nach Gleichung 2-4 mit einem Emissionskoeffizienten von $\varepsilon = 1$ ideal strahlt. Die emittierte spezifische Strahlung \dot{e}_{spez} (in W/m^2) eines solchen schwarzen Strahlers ist mit dem Plank'schen Strahlungsgesetz berechenbar [8]:

$$\dot{e}_{\text{spez}} = \int \frac{c_1}{\lambda_W^5 \cdot \left(\exp\left(\frac{c_2}{\lambda_W T_K}\right) - 1 \right)} d\lambda_W = \sigma T_K^4 \quad (2-5)$$

mit

$c_1 = 3,7418 \cdot 10^{-16} \text{ W}/\text{m}^2$ als Schwarzkörperkonstante 1

$c_2 = 1,4388 \cdot 10^{-2} \text{ m} \cdot \text{K}$ als Schwarzkörperkonstante 2

λ_W = Wellenlänge in m

T_K = Temperatur des thermischen Körpers in K

$\sigma = 5,6696 \cdot 10^{-8} \text{ W}/(\text{m}^2 \cdot \text{K}^4)$ als Stefan-Boltzmann-Konstante

Die terrestrische Wärmestrahlung ist ein Teil in der Strahlungsbilanzierung der Erde und gibt an, wie viel Wärmestrahlungsleistung von der Erde emittiert wird. Die in der Atmosphäre zurückgestrahlte Gegenstrahlung, atmosphärische Wärmestrahlung genannt, erfasst die von der Atmosphäre emittierte Strahlung [10]. Eine Berechnung dieser Wärmestrahlungen ist wegen der ungleichmäßigen Erdoberfläche und der unbestimmbaren Strahlungskoeffizienten lediglich eingeschränkt möglich. Ångström [11] hat ein Modell entwickelt, das diese Strahlungsleistungen modelliert [10]. Demnach lässt sich die atmosphärische Strahlung A_g (in W/m^2) wie folgt abschätzen:

$$A_g = N \cdot A_{gW} + (1 - N)A_{gK} \quad (2-6)$$

wobei

$$A_{g_w} = \sigma \cdot T_{dSt}^4 \quad (2-7)$$

$$A_{g_k} = \sigma \cdot T_{St}^4 \left(0,790 - 0,174 \cdot 10^{-0,041 \cdot 1/hPa \cdot p_H} \right) \quad (2-8)$$

mit

A_{g_w} = atmosphärische Gegenstrahlung bei bedecktem Himmel in W/m^2

A_{g_k} = atmosphärische Gegenstrahlung bei wolkenlosem Himmel in W/m^2

T_{dSt} = Taupunkttemperatur in Stationshöhe in K

T_{St} = Lufttemperatur in Stationshöhe in K

N = Bedeckungsgrad ohne Einheit

p_H = Wasserdampfpartialdruck in hPa

2.2 Thermische Gebäude- und Anlagensimulation

In einer thermischen Gebäudesimulation wird das Verhalten eines Gebäudemodells über einen definierten Zeitraum (z. B. ein Jahr) unter Berücksichtigung äußerer und innerer Einflüsse abgebildet. Ziel ist es, ein realitätsnahes Abbild des Gebäudes zu erstellen und mit gegebenen Lasten unbekannte Größen zu berechnen. Die häufigste Anwendung liegt in der Berechnung der Heizlast und Kühllast unter Berücksichtigung der Sonneneinstrahlung, Außenlufttemperatur, Luftfeuchte, vorherrschenden Windverhältnisse, Nutzeranwesenheitszeiten, Luftaustausch und innerer thermischer Lasten. Das Gebäudemodell muss hierfür die physikalischen Zusammenhänge der Einflüsse auf die zu berechnende Variable abbilden.

Äquivalent zur Gebäudesimulation lässt es sich die Anlagensimulation beschreiben. Anstatt eines Gebäudes wird eine (technische) Anlage und deren Verhalten modelliert. Verwendet wird diese Art der Simulation z. B. zur Untersuchung von Anlagenverhalten bei gegebenen Randbedingungen und Einflüssen. Ist eine Anlagensimulation in eine Simulation der Umgebung eingebettet, wird von Model-in-the-Loop-(MIL)Simulationen gesprochen. So kann unter anderem das Verhalten von raumluftechnischen Anlagen, Energieerzeugern, Energiespeichern und Wärmeübertragern während des Simulationszeitraums eingeschätzt werden, ohne aufwändige Echtzeitversuche durchführen zu müssen. In der Praxis werden die Simulationen virtuell durch numerische Programme durchgeführt.

2.3 Smart-Meter-Gateways

Ein Smart-Meter-Gateway (SMGW) ist ein sogenanntes intelligentes Messsystem und wird mit dem Messstellenbetriebsgesetz (BsbG) von der deutschen Bundesregierung eingeführt. SMGWs sollen im Rahmen der Digitalisierung heutige Stromzähler ersetzen und zusätzliche Funktionen bieten, die die Effizienz und Regelbarkeit von Gebäuden erhöhen. Dazu gehört die Kommunikation mit Netzbetreiber und Energielieferant sowie die Möglichkeit, die Rechenleistung von SMGWs zu nutzen, um anwendungsspezifische Regelungen umzusetzen. Dafür stehen dem SMGW Schnittstellen zur Verfügung, durch die es Informationen von Gebäude, Nutzer und aus dem Internet abrufen und Steuer- und Regelsignale bereitstellen kann. SMGWs müssen bestimmten Standards genügen und vom Bundesamt für Sicherheit in der Informationstechnik zertifiziert werden. Hierbei steht die sichere Kommunikation und der Datenschutz im Vordergrund [12].

Für Netzbetreiber und Energieerzeuger verspricht die Einführung der SMGWs eine bessere Plan-

barkeit von abgenommenen Lasten und erzeugten Leistungen. Dies wird mit der zunehmenden Dezentralisierung des Strommarktes durch private Photovoltaikanlagen und der Veränderung auf Nutzerseite weg von den Verbrauchern hin zur Eigenerzeugung aufwändiger. Die Vorteile für den Nutzer von SMGWs sind Potentiale der Energieeinsparung und der Effizienzsteigerung von Gebäuden. Ein häufiger Fall wird beispielsweise sein, dass ein Nutzer eine Photovoltaikanlage betreibt und Strom erzeugt. Durch den Verkauf des Stroms in das Netz erwirtschaftet der Anlagenbetreiber Geld. Für den Zukauf muss er einen höheren Preis zahlen, als er mit dem verkauften Strom verdient. Somit lohnt es sich für den Betreiber, den Strom selbst zu nutzen. Hierbei kann ein SMGW helfen, indem es Lasten zuschalten kann, wenn die Photovoltaikanlage mehr Strom erzeugt, als anliegende Lasten benötigen. Diese Lasten, die z. B. das Aufladen von Akkus in elektronischen Geräten sein können, liegen nicht mehr in Zeiten mit einem Mangel an Strom. Die Folge davon sind Kosteneinsparungen, wenn variable Preise an Endkunden weitergegeben würden. Dieses Prinzip nennt sich Lastumverteilung, die mit dem Ausbau von erneuerbaren Energien notwendig werden, um deren fluktuierende Energieerzeugung der Nachfrage anzugleichen. Eine Verwendung von Energiespeichern zur Lastumverteilung mindert jedoch den Wirkungsgrad durch die entstehenden Speicherverluste.

In der Gesellschaft sind zu SMGWs Bedenken geäußert worden, die sich auf den Datenschutz beziehen. Tatsächlich stellt der Energienutzungsverlauf von einzelnen Gebäuden ein Werkzeug zur Identifikation von personengebundenen Daten dar, weil auf verschiedene Verhaltensweisen der Nutzer geschlossen werden kann. Datenschützer fordern, abseits von der ohnehin angestrebten Verschlüsselung der Kommunikation und besonderem Augenmerk auf Cyber Security, die Daten von SMGWs mit Verfahren wie Zusammenfügen von mehreren Gebäudewerten unschärfer zu machen und somit ein Rückschließen auf Nutzer zu erschweren beziehungsweise zu verhindern.

Der aktuelle Status des Rollouts der SMGWs kann auf der Website des Bundesamt für Sicherheit in der Informationstechnik nachverfolgt werden [12]. Ende 2021 waren demnach fünf SMGWs zertifiziert, die eines von drei zertifizierten Sicherheitsmodulen verwenden. Fünf weitere SMGWs befinden sich derzeit in der Zertifizierungsphase.

2.4 Optimierung

Eine Optimierung ist das Bestimmen eines Lösungsvektors eines Systems mit mehreren Dimensionen, sodass der Lösungsvektor auf ein globales Extremum weist. Der Lösungsraum des zu optimierenden Systems ist groß, sodass ein vollständiges Durchsuchen nicht in praktikabler Zeit möglich ist. Darum werden Algorithmen eingesetzt, die eine Lösung in wenigen Schritten zielgerichtet finden sollen. Ein bekannter Algorithmus der Numerik ist das Newton-Raphson-Verfahren zur Nullstellenfindung [13]. Dieses erfordert eine differenzierbare Funktion und erlaubt keine Randbedingungen, was nicht für alle Optimierungsprobleme gegeben ist. Die Lösungsraumgröße von hochdimensionalen Problemen erfordert andere Algorithmen, die mit geringem Aufwand ein akzeptables Ergebnis liefern sollen und Optimierer genannt werden. Dabei wird der Kompromiss eingegangen, dass nur ein Teil des Lösungsraums untersucht wird und somit das Ziel des Ergebnisses eines Optimierers nicht zwangsweise ein globales Extremum ist.

Lösungsräume, deren Funktion nicht bekannt sind oder auf einer nicht differenzierbaren Funktion basieren, können mit stochastischen Methoden untersucht werden. Solche Optimierer verwenden evolutionäre Methodiken der Vererbung oder soziale Verhaltensweisen wie Schwarmverhalten [14, 15], um sich im Suchraum systematisch zu bewegen und die Lösungszeit zu reduzieren. Dabei werden anhand definierter Prinzipien Lösungsvektoren erstellt, evaluiert und teilweise behalten bzw. verworfen. Im Falle evolutionärer Algorithmen bestehen diese Prinzipien aus Rekombination

und Mutation. Zur Evaluierung wird die Selektion angewendet, um eine neue Generation an Lösungsvektoren zu erzeugen. Eine genauere Erklärung verschiedener Optimierungsmethoden ist in [13, 14, 15] zu finden.

Optimierung findet in der modellprädiktiven Regelung (engl. Model Predictive Control) (MPC) Anwendung. MPC ist eine Methode zur Regelung von nichtlinearen, multivariablen Systemen ohne Linearisierung. Dabei wird anhand von bekannten Eingangsparametern eine Vorhersage des Systemverhaltens durchgeführt und als Randbedingung einer Regelungsoptimierung genutzt (vgl. [16]). Die Implementierung einer MPC wird durch die Methoden der Vorhersage und der Optimierung charakterisiert.

2.5 Funktionsweise neuronaler Netze

Neuronale Netze (NNs) sind eine Untergruppe des maschinellen Lernens, die – angelehnt an das menschliche Gehirn – mit Neuronen aus Datensätzen Strukturen erkennen können (vgl. [17]). Die Architektur eines NNs besteht aus Schichten (Layers) von Neuronen, die miteinander verknüpft sind [18]. Dem NN werden die Eingabeparameter gebündelt in einem Eingangsvektor \vec{x} übergeben. Der nach den Berechnungen durch das NN entstehende Ausgabevektor \vec{o} enthält die Ergebnisse.

2.5.1 Perzeptron

Das einfachste NN besteht aus einem Neuron, dem Perzeptron, das schematisch in Bild 2-1 dargestellt ist.

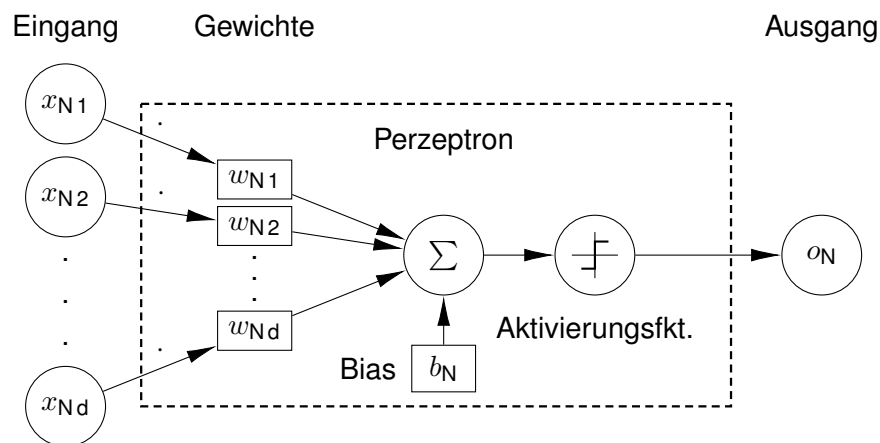


Bild 2-1: Funktionsweise eines Perzeptron bzw. Neurons

Das Perzeptron multipliziert den Eingangsvektor \vec{x}_N mit dem Gewichtsvektor \vec{w}_N und summiert das Ergebnis zusammen mit dem Bias-Wert b_N auf. Die Aktivierungsfunktion φ löst die Linearität zwischen Eingabe und Ausgabe auf und generiert den Ausgabevektor. Der Bias-Wert ermöglicht das Beeinflussen des Aktivierungsniveaus des Perzeptrons durch Verschieben entlang der Abszissenachse. Eine Berechnung des Ausgangs o_N erfolgt anhand der Gleichung 2-9, wobei der Index N für „Neuron“ steht:

$$o_N = \varphi(\vec{x}_N \cdot \vec{w}_N + b_N) \quad (2-9)$$

Als Aktivierungsfunktion können verschiedene Funktionen verwendet werden. Üblich ist die Sigmoid-Funktion, tanh oder ReLU [19]. Die Verwendung einer Aktivierungsfunktion ermöglicht

dem Perzeptron, nichtlineare Zusammenhänge abbilden zu können. Dafür bildet die Funktion einen Schwellenwert ab, ab dem sich die Ausgabe von der unteren Grenze zur oberen hin verändert.

2.5.2 Mehrschichtige neuronale Netze

Das Neuron ist der Grundbaustein für komplexere NNs und besitzt dieselbe Funktionsweise wie ein Perzeptron. Eine Schicht wird aus Neuronen gebildet. Ein NN besteht aus drei oder mehr Schichten. Dabei leitet die erste Schicht, die Eingabeschicht, an die zweite Schicht weiter. Die letzte Schicht wird Ausgabeschicht genannt und stellt das Ergebnis dar. Zwischen Eingabe- und Ausgabeschicht liegen sogenannte versteckte Schichten (engl. „Hidden Layers“).

Die Verbindungen zwischen den Neuronen und Schichten kann allumfassend von Eingabe- zur Ausgabeschicht erfolgen oder teilweise. So können verschiedene Neuronen auf andere verbundene Neuronen Einfluss nehmen. Nicht verbundene Elemente werden nicht oder indirekt beeinflusst. Diese Verbindungen machen NNs charakteristisch, sodass NNs danach benannt werden. So heißt ein NN, dessen Neuronen der Schichten komplett mit den Neuronen der nachfolgenden Schicht verbunden sind, „Fully-Connected“ (in dt. „vollständig verbunden“). Ein solches Netz ist in Bild 2-2 mit Eingabeschicht, einer versteckten Schicht mit vier Neuronen und einer Ausgabeschicht dargestellt. Ein derartiges Netz kann verwendet werden, um aus drei Variablen eine Ausgabe zu erzeugen.

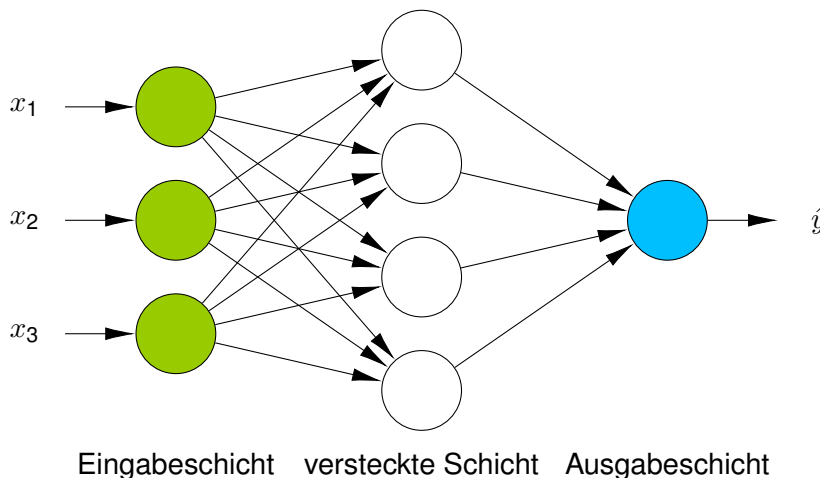


Bild 2-2: Beispiel eines Fully-Connected Netzes

Die Berechnung eines mehrschichtigen NN basiert auf Gleichung 2-9 mit zusätzlicher Dimension. So können alle parallelen Neuronen (alle Neuronen einer Schicht) mit der Gewichtsmatrix $\widehat{\mathbf{W}}_L$ einer Schicht, bestehend aus allen Gewichtsvektoren \vec{w}_i berechnet werden:

$$\vec{o}_i = \varphi(\widehat{\mathbf{W}}_L \cdot \vec{x}_i + \vec{b}_L) \quad (2-10)$$

wobei der Bias einen Vektor \vec{b}_L darstellt, bestehend aus allen Bias b_i der einzelnen Neuronen. Die Anzahl der Neuronen in der vorangegangenen Schicht wird durch die Laufvariable i repräsentiert. Die Aktivierungsfunktion φ wird elementweise auf den resultierenden Vektor aus $\widehat{\mathbf{W}}_L \cdot \vec{x}_i + \vec{b}_L$ angewandt. Die Ausgabe eines gesamten NN \vec{y} kann somit mit der Gesamtheit der Gewichtsmatrizen und Bias-Vektoren $\widehat{\mathbf{W}}$ und dem Eingangsvektor \vec{x} berechnet werden:

$$\vec{y} = f(\vec{x}; \widehat{\mathbf{W}}) \quad (2-11)$$

2.5.3 Training

Das Training eines NN ist notwendig, um die Gewichte und Bias anzupassen, sodass der Ausgang dem gewünschten Ergebnis entspricht. Dafür werden Daten (Trainingsdaten) benötigt, die als Eingang dienen, von denen der Ausgang bekannt ist¹. Vor dem ersten Durchlauf des NN wird $\widehat{\mathbf{W}}$ initial mit Werten gefüllt. Eine Verlustfunktion \mathcal{L} definiert mit dem Sollwertvektor \vec{y} aus den Trainingsdaten den Fehler L [20]:

$$L = \mathcal{L}(f(\vec{x}; \widehat{\mathbf{W}}); \vec{y}) = \mathcal{L}(\vec{\hat{y}}; \vec{y}) \quad (2-12)$$

Der Fehler wird in der Praxis über mehrere Datenpunkte gebildet und durch eine Kostenfunktion auf einen Skalar reduziert, der als Maß für die Genauigkeit des NN angesehen werden kann. Eine übliche Kostenfunktion J_{mse} für m Datenpunkte ist der mittlere quadratische Fehler (MSE, engl. „Mean Squared Error“):

$$J_{\text{mse}}(\widehat{\mathbf{W}}) = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 \quad (2-13)$$

Zur Minimierung der Kostenfunktion wird u. a. die Backpropagation (engl. für Fehlerrückführung) verwendet. Diese benutzt das Gradientenverfahren, um die optimale Gewichtsverteilung \mathbf{W} zu finden. Ziel ist die Minimierung der Kostenfunktion, die auf eine Dimension vereinfacht wird. Für eine genauere Beschreibung der Backpropagation wird auf [20] verwiesen. Eine weitere Methode der Gewichtsoptimierung ist der Optimierer Adam [21].

Um das Ergebnis des Trainings bewerten zu können, werden nicht alle vorhandenen Daten zum Training verwendet, sondern in drei Teile aufgeteilt. Jeder Datensatz ist für eine Aufgabe gedacht, wobei das Verhältnis bei der Datenaufteilung ca. 70 % Trainingsdaten, ca. 20 % Validierungsdaten und ca. 10 % Testdaten beträgt. Je nach Anwendung kann das Verhältnis variieren. Die Validierungsdaten ermöglichen es, das Verhalten des Netzes während des Trainings mit unabhängigen, dem Netz unbekanntem Daten zu validieren. Dadurch wird vermieden, dass das NN bei zu häufigem Anpassen der Gewichte an den Trainingsdatensatz die Ausgaben durch Auswendiglernen erzeugt, anstatt anhand des Eingabevektors. Dieses Problem wird als Übertraining bezeichnet. In diesem Fall nehmen die Fehler bei der Validierung zu und das Training wird abgebrochen. Bild 2-3 stellt den Bereich, ab dem Übertraining stattfindet, grafisch dar. Der Testdatensatz wird mit dem fertig trainierten Netz angewendet und dient als Referenz der Leistung des NNs.

Rauschende Daten können während des Trainings zu Problemen führen, weil die Gewichte fluktuierend angepasst werden müssen. Dies kann unter anderem bei der Online-Adaption auftreten, die nach jedem Durchlauf eines einzigen Eingabevektors die Gewichtung vornimmt. Abhilfe schafft das Gruppieren von Eingabevektoren zu Stapeln (engl. „Batch“) und die Bestimmung der Gewichte anhand des Mittelwerts von allen Eingabevektoren im Stapel. Das birgt den Vorteil der Parallelisierung und die Rechenzeit kann verkürzt werden. Dafür müssen Recheneinheiten mit mehreren Rechenkernen verwendet werden, wie z. B. Grafikkarten.

2.5.4 Rekurrente neuronale Netze und die LSTM-Zelle

Neuronen können zu Neuronen derselben und darauffolgender Schicht, sowie zu Neuronen vorheriger Schichten eine Verbindung haben. Ein solches NN wird als rekurrentes neuronales Netz

¹Dies gilt für überwachtes Lernen. Für den Unterschied zu unüberwachtem Lernen sei auf [20] verwiesen.

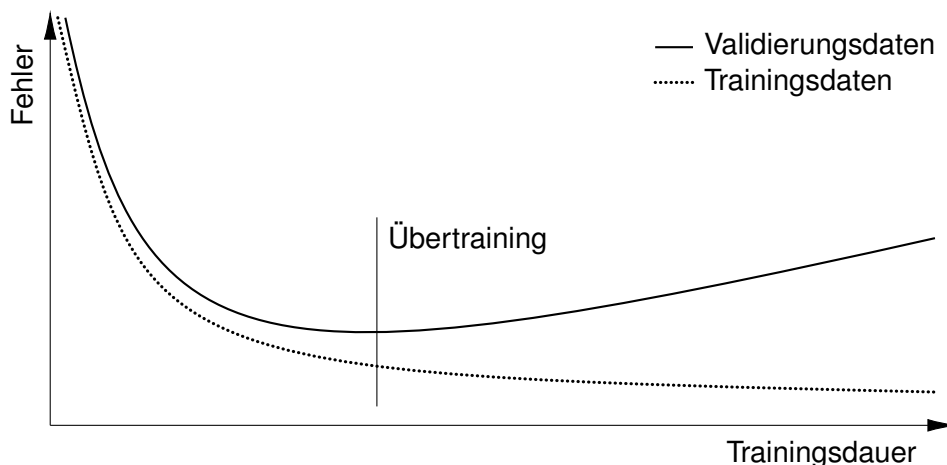


Bild 2-3: Darstellung des Trainingsverlaufs mit Bereich des Übertrainings

(RNN) bezeichnet. Es ermöglicht das Transferieren von Aktivierungspotentialen von einem RNN-Durchlauf in den Nachfolgenden. So können frühere Eingangsvektoren spätere beeinflussen. Diese Eigenschaft prädestinieren RNNs für zeitabhängige Eingabevektoren.

Die Long Short-Term Memory (LSTM)-Zelle ist eine besondere Form eines RNNs. Sie enthält vier Fully-Connected-Unternetze mit unterschiedlichen Aufgaben. Die Unternetze heißen Forget-Gate, Input-Gate, Cell-State-Gate und Output-Gate (deutsch: Vergessens-, Eingangs-, Zellstatus- und Ausgangstor). Diese ermöglichen der LSTM-Zelle, Informationen aus einem Durchlauf mit einem Eingangsvektor zu speichern und späteren Eingangsvektoren zuzuführen. Vereinfacht ausgedrückt kann eine LSTM-Zelle analog zum menschlichen Gehirn sich erinnern und vergessen. Das impliziert, dass eine LSTM-Zelle mit aufeinanderfolgenden Durchläufen zeitabhängiger Eingangsvektoren Vorteile besitzt. Der Vorlauf, während dem die produzierten Ausgangsvektoren nicht verwendet und die Parameter in den vier Unternetzen generiert werden, wird als Aufwärmvorgang bezeichnet. Da diese Unternetze die Vorhersage beeinflussen, können alle für den Aufwärmvorgang verwendeten Eingangsparameter das Ergebnis beeinflussen. Nach dem Aufwärmvorgang generiert eine LSTM-Zelle den ersten Ausgangsvektor \vec{o}_1 , der anhand der im Training gewonnenen und im Aufwärmvorgang angewendeten Erkenntnisse generiert wird. Die während des Aufwärmvorgangs generierten Ausgabevektoren werden nicht als Vorhersage verwendet, weil zu diesem Zeitpunkt die Parameter der Unternetze noch nicht berechnet wurden. Um mehr als eine Ausgabe \vec{o}_1 im Anschluss an die erste zu bekommen ($\vec{o}_2 \dots \vec{o}_t$), wird jede Ausgabe autoregressiv als Eingang verwendet, bis die gewünschte Anzahl an Ausgaben t erzeugt ist. Diese Funktionsweise wird im Schema in Bild 2-4 grafisch dargestellt.

Eine LSTM-Zelle wird durch die Dimensionen der Unternetze definiert, die LSTM-Einheiten (engl. „Units“) genannt werden. Durch die in der Zelle stattfindenden Multiplikationen und Additionen besitzt jedes Unternetz dieselbe Dimension. Das Training des Netzes funktioniert im Groben äquivalent zu einem Feedforward-NN bzw. einem NN mit Fully-Connected-Schichten. Für eine genauere Beschreibung einer LSTM-Zelle wird auf Westhauser [22] verwiesen.

Der Vorteil einer LSTM-Zelle ist, dass bisherige Werte zukünftige beeinflussen können und deutlich bessere Vorhersagen als mit anderen NNs gemacht werden. Dies wird mit der Zeit durch ihren Nachteil der Fehleraufsummierung verdrängt. Denn durch die Autoregression werden falsche Werte zur Vorhersage genutzt. Folglich muss bei der Anwendung eines NNs mit LSTM-Zelle beachtet werden, dass das Ergebnis durch eine Vielzahl an Autoregressionsschritten nicht unbrauchbar gemacht wird.

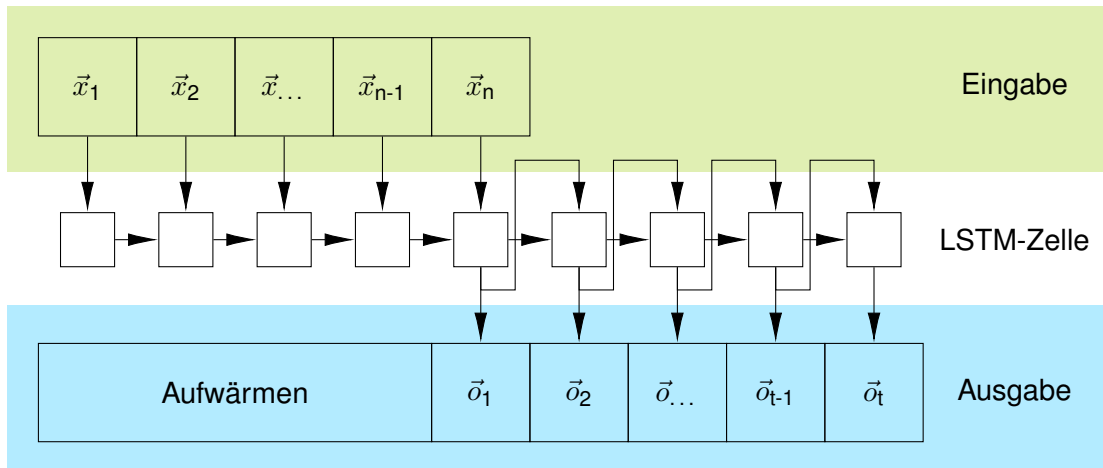


Bild 2-4: Aufwärmen und autoregressives Vorgehen bei LSTM-Zellen

2.5.5 Normalisierung

Unabhängig vom verwendeten NN ist es ratsam, die Eingabeparameter zu normalisieren [23, 24]. So wird der Vorgang genannt, bei dem alle Werte gestaucht und verschoben werden, sodass sie größtenteils in einem Bereich von $-1 \dots 1$ oder $0 \dots 1$ liegen. Dafür wird der betrachtete Wert y_i mit einem Versatz ΔV mit gleicher Einheit wie der Wert addiert und mit einem Teiler D (ohne Einheit) geteilt, um seinen normalisierten Wert y_{norm} zu bekommen:

$$y_{\text{norm}} = \frac{y_i - \Delta V}{D} \quad (2-14)$$

Gängige Praxis ist, als Versatz die Standardabweichung aller j Werte y des betrachteten Parameters und als Teiler das arithmetische Mittel \bar{y} oder den Median \hat{y} von y zu verwenden (die Werte besitzen dieselbe Einheit wie y) (Berechnung nach [25]).

$$\Delta V = \sqrt{\sum_{i=1}^j (y_i - \bar{y})^2} \quad (2-15)$$

$$D \in \{\bar{y}, \hat{y}\} \quad (2-16)$$

mit

$$\bar{y} = \frac{1}{j} \cdot \sum_{i=1}^j y_i \quad (2-17)$$

$$\hat{y} = \begin{cases} y_{j/2+0,5} & \text{wenn } j \text{ ungerade} \\ \frac{1}{2}(y_{j/2} + y_{j/2+1}) & \text{wenn } j \text{ gerade} \end{cases} \quad (2-18)$$

Anhand der verwendeten Daten und deren Bedeutung können manuell gewählte Werte für Teiler und Versatz ermittelt werden. So kann es sinnvoll sein, die Werte der spezifischen Globalstrahlung auf der Erde, die aus physikalischen Gründen zwischen null und der Solarkonstante bei ca. 1367 W/m^2 liegen kann [26], mit dem Teiler $D = 1367$ und dem Versatz $\Delta V = 0 \text{ W/m}^2$ zu normalisieren. Eine Untersuchung hinsichtlich einiger Normalisierungsmethoden wird in Bild 4-1 (Kapitel 4) durchgeführt.

Teiler und Versatz werden für jeden Parameter unabhängig von den anderen berechnet. In der Theorie soll auf die Verwendung der Validierungs- und Testdaten verzichtet werden, weil das NN über die Normalisierung Informationen aus den Datensätzen erhält und das Training unabhängig

von den Test- und Validierungsdaten sein soll. Komplexere Normalisierungsmethoden verhindern im Training den Zugang zu in der Zukunft liegenden Daten, indem z. B. ein gleitender Mittelwert verwendet wird.

3 Applikationsmodell

In folgendem Kapitel wird die Funktionsweise und der Aufbau des in dieser Studie erstellten Applikationsmodells zur Lastumverteilung auf Basis eines Netz-Signals für die Anwendung in einem Smart-Meter-Gateway (SMGW) erläutert. Der Aufbau des Applikationsmodells lässt sich in die Unterbereiche Datenaufbereitung, Vorhersage und Optimierung aufteilen. Außerdem wird für die Model-in-the-Loop-(MIL)-Simulation zur Untersuchung und Bewertung der Funktionsweise des Modells ein Beispielgebäude mit thermischen Eigenschaften und Anlagentechnik (Speicher, Energieerzeuger) definiert, sodass mit der Gebäude- und Anlagensimulation Trainings- und Testdaten erzeugt werden können.

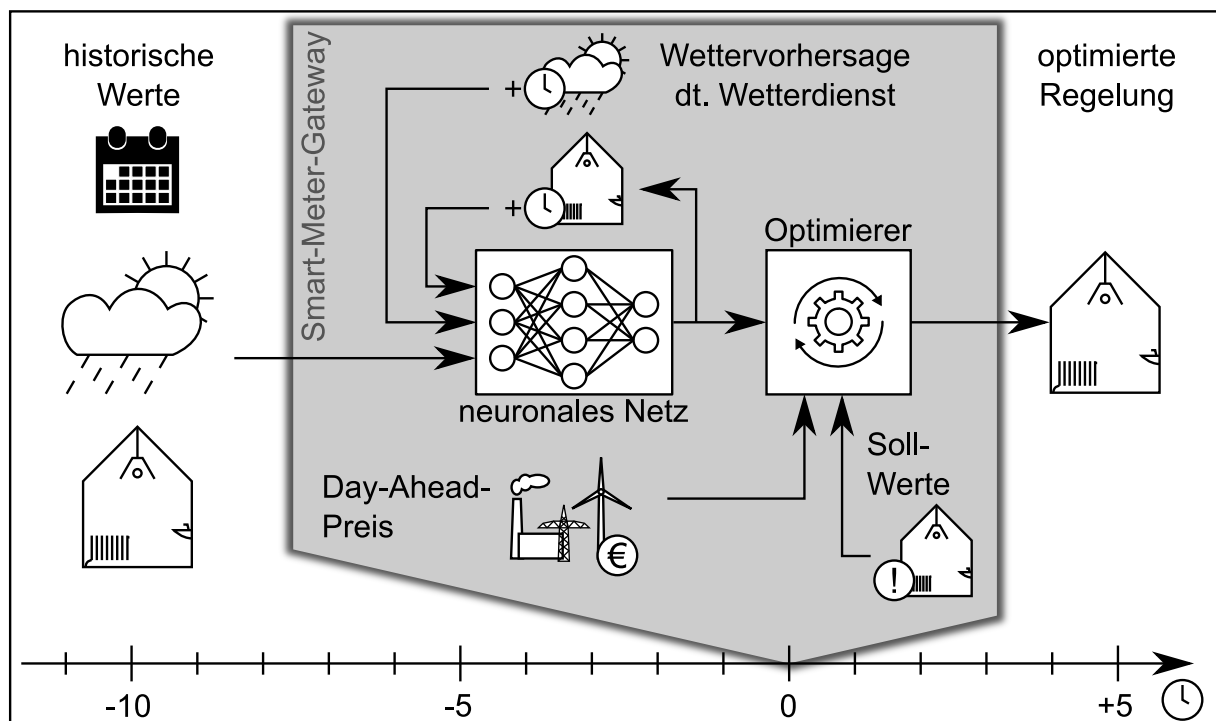


Bild 3-1: Grafische Darstellung des Applikationsmodells

Bild 3-1 stellt schematisch den Aufbau und die Funktionsweise des Applikationsmodells dar. Das Grundprinzip ist die Nutzung einer Vorhersage von bevorstehenden Lasten zur Optimierung der Speicherbeladung anhand eines netzseitigen Steuersignals, beispielsweise des Strompreises oder des Anteils erneuerbarer Energien (EE) am Strommix. Dies erfolgt „technologieoffen“ und unabhängig vom Energieerzeuger oder der Art des Speichers. So kann z. B. ein Gebäude mit Wärmepumpe und Warmwasserspeicher hinsichtlich geringen CO₂-Ausstoßes oder die Beladung eines elektrisch angetriebenen Fahrzeugs hinsichtlich geringen Strompreises optimiert werden. Die Evaluation des Modells durch die MIL-Simulation erfolgt anhand eines Beispielgebäudes, das hinsichtlich der Stromaufnahme des Wärmeerzeugers optimiert wird. Es wird ein Warmwasserspeicher als Kurzzeitwischenspeicher betrachtet, um anhand einer Tagesvorhersage mit stündlicher Auflösung die Kosten zur Deckung der Gebäudeheizlast zu minimieren. Als Netzsignal wird der Anteil fossiler Energieerzeuger am Strommix durch eine Optimierung hin zu Betriebszeiten mit hohem Anteil von EE sowie der Day-Ahead-Strompreis untersucht. Zu beachten ist, dass die

Optimierung hinsichtlich geringer CO₂-Emissionen ohne großen Aufwand in der Praxis umsetzbar ist. Für die Optimierung des Strompreises ist eine Voraussetzung, dass der Börsenstrompreis vom Energielieferant an den Nutzer durchgereicht würde.

3.1 Beispielgebäude und Umgebungsbedingungen

Das für die Simulation verwendete Beispielgebäude ist ein Einfamilienhaus aus Stuttgart, das im Rahmen einer Bachelorarbeit von Kienzlen erstellt wurde [27]. Dort wird der Wohngebäudebestand in Deutschland untersucht und ein repräsentatives Gebäudemodell erstellt, das in der Simulationssoftware TRNSYS für die Simulation vorbereitet wird. Das Einfamilienhausmodell hat eine Grundfläche von 98,7 m² und ist zweigeschossig aufgebaut. Beide Geschosse werden im Modell als eine Regelzone zusammengefasst. Bild 3-2 zeigt eine gerenderte räumliche Ansicht des Gebäudes [27].

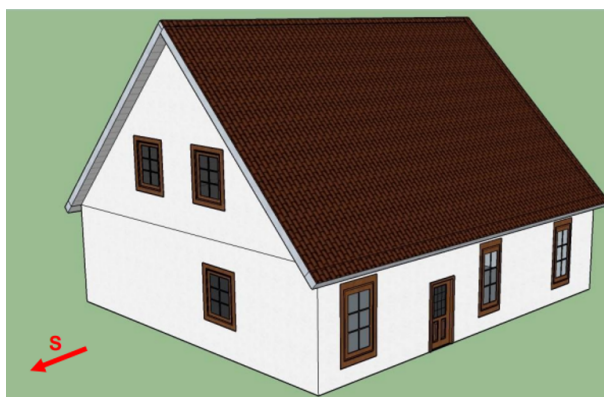


Bild 3-2: Verwendetes Beispielgebäude [27]

Die thermischen Eigenschaften des Beispielgebäudes sind in Anhang B zu finden. Sie sind an die Energieeinsparverordnung von 2014 angelehnt. Die Belegungswerte sowie innere Lasten werden nach DIN EN 16798-1:2019 Anhang C „Wohngebäude, Wohnung“ festgelegt [28]. Die Solltemperatur beträgt 20 °C, solange Nutzer anwesend sind. Es wird keine Nachtabsenkung berücksichtigt, wie sie in der Praxis häufig angewandt wird, weil nach der Belegung aus der Norm zu jeder Zeit Nutzer anwesend sind und das Gebäude als eine Zone berechnet wird [28].

Die Wetterdaten für die Gebäudesimulationen mit dem Beispielgebäude werden vom öffentlich zugänglichen FTP-Server des Deutschen Wetterdienstes (DWD) heruntergeladen. Es werden die historischen Daten der Jahre 2010 bis 2020 verwendet (siehe Kapitel 3.2.2). Die Messstationen stehen in Stuttgart-Echterdingen und Stuttgart Schnarrenberg (Stationsnummern 4928 und 4931), wobei von letzterer alle Solarstrahlungswerte stammen. Einige Daten müssen für die Simulation berechnet werden, wobei Werte vom DWD als Berechnungsgrundlage dienen. Tabelle 3-1 zeigt eine Übersicht aller verwendeter und in der Simulation berücksichtigter Parameter sowie deren Quelle. Zur Berechnung von Werten der feuchten Luft wird das Python-Package CoolProp mit der Funktion HAPropsSI verwendet, das quelloffen auf Github zur Verfügung gestellt wird [29].

Die Werte der terrestrischen Wärmestrahlung werden mit Gleichung 2-6 für die atmosphärische Gegenstrahlung abgeschätzt. Die Werte sind üblicherweise 25 %¹ zu gering im Vergleich zur theoretisch auftretenden Strahlung. Der entstehende Fehler ist vernachlässigbar, weil die Funktionsweise eines neuronalen Netzes (NN) es erlaubt, den Einfluss der terrestrischen Wärmestrahlung

¹Dieser Wert stammt aus einem Vergleich von nach Gleichung 2-6 berechneten Werten und Werten aus [27].

Tabelle 3-1: Für das Applikationsmodell und in der Gebäudesimulation berücksichtigte Umgebungsparameter

Parameter	verwendete DWD-Werte (Kürzel)	Herkunft
Außenlufttemperatur	TT_10	DWD
Luftdruck	P0	DWD
Windgeschwindigkeit	F	DWD
Windrichtung	D	DWD
(Wolken-)Bedeckungsgrad	V_N	DWD
Wasserdampfgehalt	TD_10; TT_10, P0	HAPropsSI
relative Feuchte	TD_10; TT_10, P0	HAPropsSI
Direktstrahlung	FG_LBERG; FD_LBERG; SD_LBERG	DWD
Diffusstrahlung	FD_LBERG; SD_LBERG	DWD
Atmosphärische Wärmestrahlung	ATMO_LBERG	DWD
Terrestrische Wärmestrahlung	TD_10; TT_10; V_N	Abschätzung

auf andere Größen zu projizieren, wie die Taupunkt- und Außenlufttemperatur, von denen die Wärmestrahlung teilweise abhängt (siehe Gleichung 2-5). NNs kommen mit einem korrelierten Wert einer Größe aus, um auf das Ergebnis zu kommen. Des Weiteren ist der Einfluss der terrestrischen Wärmestrahlung auf das Gebäude- und Anlagenmodell in TRNSYS geringfügig.

3.2 Vorhersage

Dieses Unterkapitel erklärt die Bestimmung der Vorhersage für die modellprädiktive Regelung. Dafür werden ein NN aufgebaut, Trainingsdaten aufbereitet und das NN trainiert. Dieses kann für Vorhersagen in einem SMGW oder für Simulationen verwendet werden. Hierbei sind die verwendeten Eingangs- und Ausgabevektoren frei definierbar, ebenso wie verschiedene Einstellungen des NNs. Die nachfolgende Beschreibung bezieht sich auf die Vorhersage der Heizlast. In der Anwendung kann hier jeglicher anderer verfügbarer Parameter genutzt werden, weil das NN grundsätzlich in der Lage ist, auf Basis entsprechender Trainingsdaten anderweitige Parameter vorherzusagen.

3.2.1 Aufbau neuronales Netz

Das verwendete NN nutzt eine Long Short-Term Memory (LSTM)-Zelle mit nachfolgender Reduzierungsschicht (engl. „Dense-Layer“, `tf.keras.layers.Dense`) für die Vorhersage auf kurze bis mittlere Vorhersagehorizonte. Das Ziel ist es, hierdurch ein autoregressives NN zu erstellen, welches anhand vergangener Gebäude- und Wetterdaten eine Vorhersage mit der Länge von t Zeitschritten macht. Anders als in Kapitel 2.5.4 und Bild 2-4 beschrieben und in der Literatur praktiziert, gibt das NN nicht dieselben Parameter im Ausgabevektor aus, die im Eingabevektor verwendet werden. Stattdessen vermindert die Reduzierungsschicht die Dimension des Ausgabevektors aus der LSTM-Zelle auf die Anzahl aller gebäudespezifischen Daten. Die für den Eingabevektor fehlenden Wetterdaten werden aus der Wettervorhersage vom DWD, den sogenannten Model Output Statistics-MIX (MOSMIX)-Daten, extrahiert, die über eine Programmierschnittstelle (engl. Application Programming Interface) (API) öffentlich zugänglich sind und vor jeder Vorhersage heruntergeladen werden. Somit nutzt das NN keine selbsterstellten Wetterdaten, sondern nutzt Gebäudeparameter zur Autoregression. Das sollte eine Verbesserung der Vorhersagequalität zur Folge

haben, weil die Wettervorhersage des DWD anhand des Stands der Technik und des Wissens auf Großrechnern durchgeführt wird. Diese berechnen unterschiedliche Szenarien und verwenden verschiedenste Berechnungsverfahren und eine Vielzahl an Wetterdaten (vgl. [30]). Ein NN, das einzig mit den Daten einer Wetterstation trainiert wird, kann höchstwahrscheinlich keine qualitativ hochwertigeren Vorhersagen erzeugen. Nachteil dieser Methode ist eine Begrenzung der Eingabeparameter, weil nur die Werte genutzt werden können, die in den MOSMIX-Daten bereitgestellt werden. Außerdem müssen die gewählten Eingangsparameter als historische Daten vorliegen, um das NN trainieren zu können. Die Parameter, auf die das mit einer Auflösung von einer Stunde zutrifft, sind (in Klammern stehen die Kürzel des DWD in den historischen Wetterdaten):

- Taupunkt (TD_10)
- Bedeckungsgrad (V_N)
- Windgeschwindigkeit (F)
- Windrichtung (D)
- Globalstrahlung (FG_LBERG)
- Außenlufttemperatur (TT_10)
- Luftdruck (P0)

3.2.2 Datenaufbereitung

Alle ausgewählten Parameter, die als Berechnungsgrundlage des NNs dienen sollen, müssen im Zuge der Datenaufbereitung nach Anforderungen erstellt oder umgerechnet und zu Eingangsvektoren zusammengefasst werden. Zusätzlich wird jeder Eingangsvektor vor der Verwendung durch das NN normalisiert (siehe Kapitel 2.5.5).

In dieser Arbeit bestehen die Rohdaten aus Simulationsdaten der Software TRNSYS. Zur Verfügung stehen die Daten von fünf Jahren (2011, 2015, 2017, 2018 und 2019). Die vorliegenden Simulationsparameter sind die flächenbezogene Heizlast des Beispielgebäudes und ein Wert für die Nutzeranwesenheit nach DIN EN 16798 (siehe Kapitel 3.1). Diese werden als Matrix bzw. Tabelle mit dem Python-Package Pandas in das Format DataFrame mit einem eigens geschriebenen Parser (von lat. „pars“: Teil und engl. „to parse“: analysieren) umgewandelt.

Dem hinzugefügt werden die Wetterparameter aus obiger Aufzählung. Dafür wird das Python-Package `wetterdienst`^{II} verwendet. Dieses ermöglicht eine einfache Nutzung der API des DWDs mit automatischem Bereitstellen der angeforderten Daten in einem DataFrame. Davor muss vom Standort des Gebäudes aus betrachtet die nächstgelegene Wetterstation gefunden werden. Dies wird mit dem Herunterladen der Stationsmetadaten und dem Python-Package `haversine` mit der Funktion `haversine_vector` erreicht. Damit wird mit der Haversine-Gleichung, eine Gleichung zur Berechnung von Entfernungen auf der Erde, die in Luftlinie nächstgelegene Station ermittelt, die alle ausgewählten Wetterparameter erfasst. Außerdem werden inaktive Stationen und alle, die nicht die gesamte Zeitspanne der Simulationsdaten abdecken, automatisch aussortiert.

Die Wetterdaten des DWD enthalten stellenweise fehlerhafte Werte. Überschreitet diese Lücke eine definierte Zeitspanne, z. B. fünf Stunden, werden alle in dieser Lücke enthaltenen Zeitschritte nicht für das Training verwendet. Bei kleineren Lücken wird der fehlerhafte Parameter linear interpoliert.

Dem Datensatz können automatisiert verschiedene Metadaten hinzugefügt werden. Diese dienen dem NN als zusätzliche Orientierung und können die Vorhersage signifikant verbessern. In dieser Arbeit werden folgende Metadaten zu jedem Zeitpunkt bestimmt:

^{II}Eine Liste aller verwendeten Python-Packages einschließlich zugehöriger Versionsnummern kann Anhang A entnommen werden.

- Azimutwinkel (repräsentiert den Tageszyklus)
- Elevationswinkel der Sonne (repräsentiert den Sonnenverlauf eines Tages und den Jahreszyklus)
- Wert für Wochenendtage (repräsentiert den Wochenzyklus)

Alle Parameter werden in kontinuierliche Werte verändert, sodass keine Sprünge in den Daten sind, die bei Interpretation kontinuierliche Vorgänge repräsentieren. Zum Beispiel müssen angegebene Winkel wie der Azimutwinkel, der die Werte 0...360 annehmen kann, in für das NN interpretierbare Daten umgerechnet werden. Aus diesem Grund wird der Azimutwinkel mit der Kosinusfunktion kontinuierlich gemacht und die Windgeschwindigkeit und -richtung in einen Vektor umgerechnet, der X- und Y-Koordinaten aufweist.

Wie in Kapitel 2.5.3 beschrieben, wird der Datensatz in drei Unterdatensätze für das Training, die Validierung während des Trainings und den Test des NNs aufgeteilt. Dabei werden zwei Methoden angewandt: eine klassische und eine gleichmäßige Aufteilung. Die klassische Methode nutzt die ersten 70 % des Datensatzes als Trainings-, die darauffolgenden 20 % als Validierungs- und die letzten 10 % als Testdatensatz. Dies hat den Nachteil, dass die Testdaten die letzten 4,8 Monate eines Jahres beinhaltet und somit keine Werte aus der ersten Hälfte eines Jahres getestet werden. Auch der Validierungsdatensatz wird nicht statistisch gleichmäßig über ein Jahr verteilt. Dieser Effekt wird mit kleineren Basisdatensätzen größer, was in der Praxis wahrscheinlich ist. Darum wird eine gleichmäßige Aufteilung implementiert, die den gesamten Datensatz in kleine Teilstücke zerlegt und jeweils diese nach obigem Verhältnis in die Unterdatensätze aufteilt. Das Ergebnis der Aufteilmethoden wird in Bild 3-3 am Beispiel des Parameters „Elevation“ dargestellt. (Die auftretenden Datensatzinkonsistenzen sind verursacht durch fehlerhafte Wetterdaten.) Die Auswirkung der Aufteilmethode auf die Vorhersagequalität des NNs wird in Kapitel 4.1 aufgezeigt.

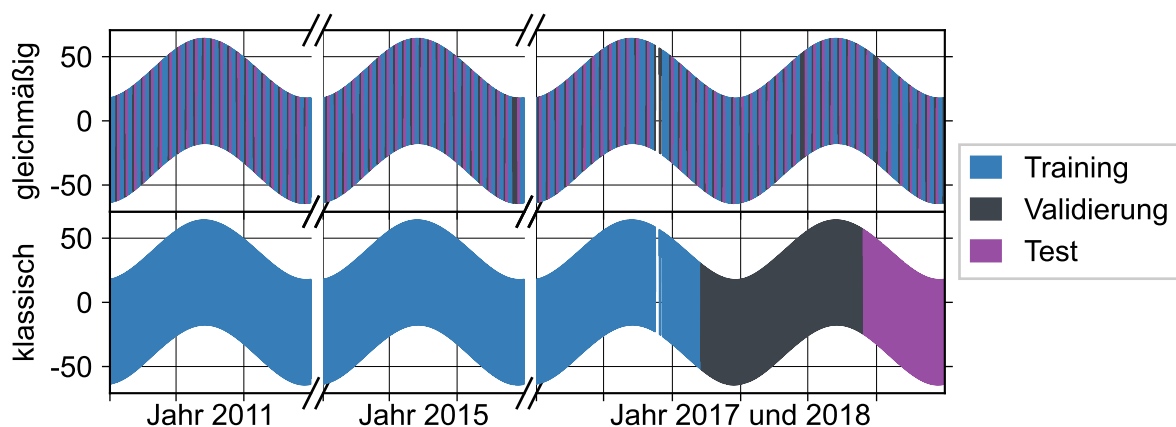


Bild 3-3: Klassische und gleichmäßige Aufteilmethoden im Vergleich für den Parameter „Elevation“

Zur Fehlervermeidung werden alle einheitenabhängigen Größen in SI-Einheiten umgewandelt. Danach werden für jeden Parameter die zugehörigen Teiler und Versätze nach Kapitel 2.5.5 bestimmt und alle Werte normalisiert. Als Versatz (vgl. Kapitel 2.5.5) wird der Median des Trainingsdatensatzes verwendet. Bild 3-4 zeigt die Verteilung aller Werte nach der Normalisierung, aufgeteilt nach Trainings-, Validierungs- und Testdatensatz. Durch das Verwenden des Medians des Trainingsdatensatzes und nicht des gesamten Datensatzes wird die Unabhängigkeit der Validierung und Testung des NNs sichergestellt. Darum besitzen die entsprechenden Datensätze Mediane ungleich Null (siehe Bild 3-4).

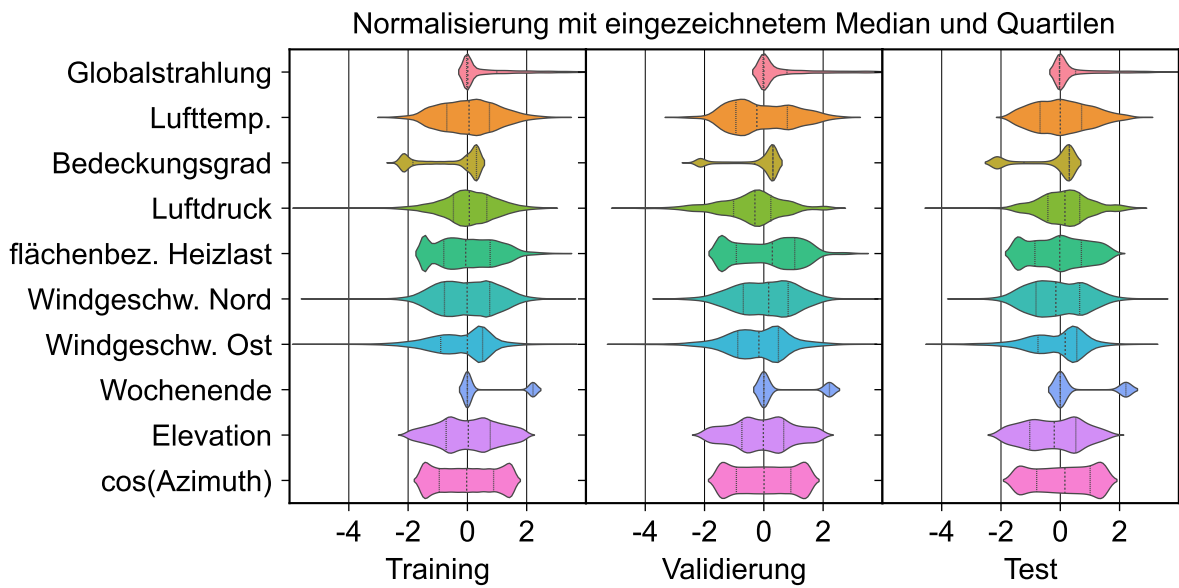


Bild 3-4: Werte nach der Normalisierung mit dem Median (grob gestrichelte Linie) und der Standardabweichung von ausgewählten Parametern

3.2.3 Training

Für das Training wird ein Generator von Eingabevektoren erstellt, der aus den Datensätzen einzelne Eingangsvektoren bzw. Stapel generieren kann. Dafür wird das Moving-Window-Verfahren verwendet, das bei gleicher Datensatzlänge ein Vielfaches an unterschiedlichen Eingabevektoren für das Training eines NNs generiert. Entgegen dem Tumbling-Window-Verfahren wird nicht jeder Eingabevektor nach dem Ende des vorherigen begonnen, sondern jeweils um einen Zeitschritt verschoben. Bild 3-5 stellt die Verfahren grafisch mit einem fünf Zeitschritte langen Fenster über eine Zeitspanne von zehn Zeitschritten dar. Jedes Fenster enthält die Daten für den Eingabevektor und die Sollwerte für die Vorhersage.

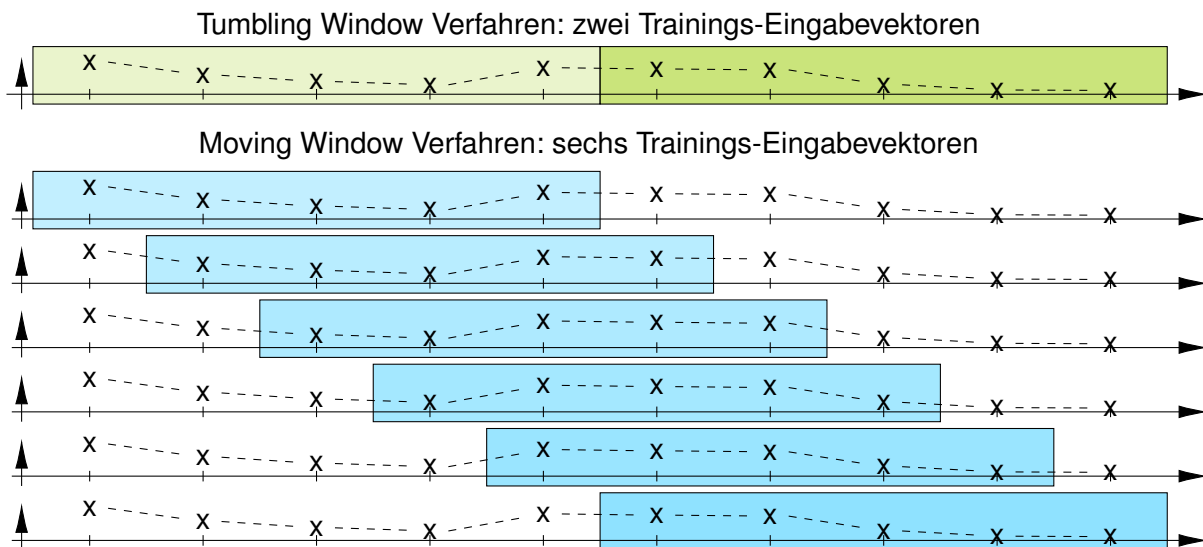


Bild 3-5: Darstellung des Moving-Window-Verfahrens im Vergleich zum klassischen Tumbling-Window-Verfahren bei gleicher Datensatzlänge

Als Verlustfunktion wird der mittlere quadratische Fehler (MSE) verwendet und für die Gewichts-

bestimmung der Adam-Optimierer (vgl. Kapitel 2.5.3) genutzt. Während des Trainingsvorgangs wird nach jedem vollständigen Training des Trainingsdatensatzes die Vorhersagegenauigkeit (der MSE) des Validierungsdatensatzes überprüft. Solange dieser fällt, wird eine neue Epoche gestartet, was einem neuen Durchlauf des Trainingsdatensatzes entspricht. Steigt der MSE der Validierungsdaten an, wird mit einer Verzögerung von zwei bis sechs Epochen weiter trainiert. Dabei wird die Gewichtsmatrix des leistungsfähigsten NN gespeichert. Sollte der MSE der Validierungsdaten nicht unter den Wert des bisherigen besten NNs fallen, wird das gespeicherte NN verwendet, um Übertraining vorzubeugen. Ansonsten wird die Vorgehensweise mit mehr Epochen wiederholt bis zu einer Maximalanzahl von 35 Epochen. Ein endgültiges Ergebnis zur Vorhersagegenauigkeit liegt nach dem Training vor, wenn das trainierte NN mit dem Testdatensatz evaluiert wird. Auch hier kommt der MSE zum Einsatz.

Im Rahmen der Untersuchungen werden NNs mit unterschiedlichen Einstellungen trainiert. Aus diesen werden geeignete Einstellungen für das NN abgeleitet, die anschließend für die MIL-Simulationen verwendet werden. Anstatt die Vorhersagegenauigkeit des Testdatensatzes als Validierung während des Trainings zu nutzen, wird das NN mit dem geringsten arithmetischen Mittel zwischen dem MSE von Test- und Validierungsdatensatz verwendet.

3.2.4 Vorhersage für SMGW und Simulation

Mit dem trainierten NN findet die Vorhersage für die Optimierung statt, wobei die notwendigen Daten wie in Kapitel 3.2.2 aufbereitet werden müssen. Das NN benutzt alle Einträge im übergebenen Eingangsvektor bis auf den letzten für den Aufwärmvorgang (vgl. Kapitel 2.5.4) und erzeugt mit dem aufgewärmten NN und dem letzten Eingabevektoreintrag einen einzigen Ausgabevektor mit den Ausgangsparametern (flächenbezogene Heizlast). Dieser bildet die Vorhersage des ersten Zeitschritts. Für die Autoregression wird ein neuer Eingangsvektor erstellt, wobei die historischen Werte und die Vorhersagewerte verwendet werden. Hierbei stammen, wie in Kapitel 3.2.1 beschrieben, die Gebäudedaten aus der bisherigen Vorhersage des NN, die Wetterdaten von der Vorhersage des DWDs (MOSMIX). Mit dem neuen Eingabevektor wird der Vorgang der Vorhersage – Aufwärmen des NN, einzelner Zeitschritt vorhersagen und neuen Eingabevektor bilden – solange wiederholt, bis die gewünschte Vorhersagelänge erreicht ist. Bild 3-6 stellt dieses Verfahren für die Vorhersagelänge t mit einem Eingangsvektor \vec{x} der Länge n und dem Wettervorhersagevektor \vec{x}_w grafisch dar.

Die Vorhersage, wie sie in einem SMGW stattfinden wird, unterscheidet sich von der Vorhersage während der MIL-Simulation. Letztere verwendet historische Wetterdaten, die zu Beginn für den gesamten Simulationszeitraum abgerufen werden können, während die Vorhersage im SMGW vor jeder Vorhersage die Werte (alle Wetterdaten vom DWD und Strommarktdaten vom Verband Europäischer Übertragungsnetzbetreiber (engl. European Network of Transmission System Operators for Electricity) (ENTSO-E)) aus Zweitquellen aktualisieren muss. Diese Werte reichen hierbei jeweils so weit in die Zukunft, wie sie von der Bezugsquelle bereitgestellt werden. Typischerweise sind die stündlich aufgelösten MOSMIX Daten für die nächsten zehn Tage verfügbar, wobei die Daten jede Stunde aktualisiert werden. Bei den Strommarktdaten kommt es vor, dass die Daten nur bis 24 Uhr verfügbar sind und somit weniger als 24 Zeitschritte in die Zukunft für die Optimierung verfügbar sind. Darum werden vor der Übergabe an den Optimierer die Vorhersagedaten auf denselben Zeitraum wie die Kostenfunktionswerte des Strommarkts angepasst. Alternativ kann die Autoregression des NNs frühzeitig abgebrochen werden.

Nach jeder Aktualisierung der Daten muss die Datenaufbereitung ausgeführt und die neuen in die vorhandenen Daten eingepflegt werden. Dafür ist eine Methode entwickelt worden, die Ak-

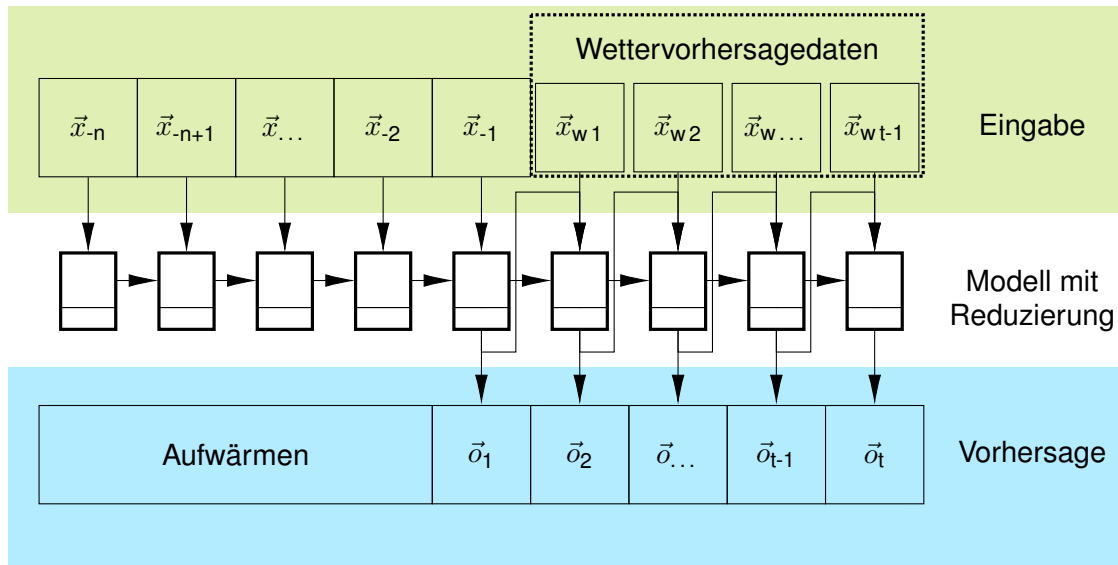


Bild 3-6: Vorhersage mit der LSTM-Zelle

tualisierungen dem Vorhersageobjekt zur Verfügung stellt, damit daraufhin die neue Vorhersage und anschließend die Optimierung durchgeführt werden kann. All diese notwendigen Funktionen zur Vorhersage in der Praxis sind im Applikationsmodell implementiert. Hierbei wurde die Funktionsweise nicht überprüft und in der Vorhersage im SMGW auftretende Probleme, wie fehlerhafte Daten und daraufhin fehlende Regelungswerte, nicht programmatisch gelöst, weil dies erst in der Anwendung effektiv möglich ist. Stattdessen wird das Augenmerk auf die Simulation gelegt, die als Unterklasse implementiert wird. Diese nutzt nach den Regeln der Vererbung durch das objektorientierte Programmieren dieselben Methoden wie die übergeordnete Vorhersageklasse, die für die Vorhersage in einem SMGW verwendet werden kann. Einige Methoden werden für ein schnelleres Simulationsergebnis optimiert. Unter anderem werden alle für die Simulation benötigten Wetter- und Strommarktdaten zu Beginn der MIL-Simulation gesammelt heruntergeladen. Entsprechend wird die Datenverarbeitung und Eingabevektorerstellung angepasst.

3.3 Optimierung

Dieses Unterkapitel behandelt die Erzeugung der Regelung durch die Optimierung der Kostenfunktion des CO₂-/Strompreises mit der vorhergesagten Heizlast des Modellgebäudes. Dafür wird zunächst die generelle Problemstellung mit den entsprechenden Randbedingungen erläutert, Lösungsmethoden aufgezeigt und im Anschluss daran ein neuartiger Optimierer vorgestellt.

3.3.1 Beschreibung des Problems

Die Vorhersage der Heizlast für n Zeitschritte in die Zukunft ermöglicht eine prädiktive Regelung für das betrachtete Gebäude. Gemeinsam mit einer Kostenfunktion wird die Beladung des Speichers hinsichtlich geringen Kosten optimiert, indem Lasten von späteren Zeitschritten auf frühere umverteilt werden. Im Falle dieser Arbeit wird als Kostenfunktion der Strompreis bzw. der Anteil fossiler Kraftwerke mit hohem CO₂-Ausstoß am Strommix berücksichtigt. Eine Lastenumverteilung

ist mit einer Zwischenspeicherung realisierbar. Umgesetzt wird das durch einen Warmwasserspeicher, der eine bestimmte Beladung Q_B (in J) und eine Maximalbeladung Q_{\max} (in J) besitzt. Durch die physikalischen Grenzen kommen folgende Bedingungen hinzu:

- Der Speicher muss mit einem positiven Betrag beladen sein ($Q_{\text{Speicher}} \geq 0$).
- Der Speicher darf niemals überladen werden ($Q_{\text{Speicher}} \leq Q_{\text{Speicher max}}$).

Daraus leiten sich folgende Randbedingungen ab:

$$\forall i \in 1, 2, \dots, n : Q_i \text{ Speicher} + Q_i \text{ Erzeuger} - Q_i \text{ Verlust} \geq Q_i \text{ Haus} \quad (3-1)$$

$$\forall i \in 1, 2, \dots, n : Q_i \text{ Speicher} + Q_i \text{ Erzeuger} - Q_i \text{ Verlust} \leq Q_{\text{Speicher max}} \quad (3-2)$$

Die Summe der zu Beginn eines Zeitschritts i im Speicher vorhandenen Energie $Q_i \text{ Speicher}$, der Verlustenergie des Speichers $Q_i \text{ Verlust}$ und der erzeugten Energie $Q_i \text{ Erzeuger}$ muss gleich groß oder größer als die in diesem Zeitschritt benötigte Energie $Q_i \text{ Haus}$ sein und darf die maximale Speichermenge $Q_{\text{Speicher max}}$ nicht überschreiten.

In Kapitel 2.1 wird deutlich, dass die Wärmeleistung von vielen Faktoren abhängig ist, darunter vor allem die Temperatur des Speichers. Das bedeutet, um denselben Betrag an Energie in den Speicher zu laden, ist abhängig von der Speicherbeladung ein anderes Energieniveau bzw. eine höhere Temperatur notwendig, die vom Energieerzeuger bereitgestellt werden muss. Dadurch ändert sich der Wirkungsgrad desselben, was Auswirkungen auf die Kostenfunktion hat, weil ein anderer Betrag an Erzeugerenergie für dieselbe Speicherbeladung benötigt wird. Auch die Verlustenergie des Speichers ist von der Speichertemperatur abhängig. Um diese Problematik zu vereinfachen, werden entsprechende Annahmen getroffen:

- Der Energieerzeuger arbeitet mit konstantem Wirkungsgrad.
- Der Speicher ist durchmischt (ungeschichtet) und besitzt eine einheitliche Temperatur.

Die Lösung des zu optimierenden Problems besteht in einem Wert für jeden Zeitschritt, der darstellt, wie viel Energie der Energieerzeuger bereitstellt. Umgesetzt wird das mit einem Wert z zwischen Null und Eins, der prozentual zur Zeitschrittlänge angibt, wie lange der Energieerzeuger mit voller Leistung und höchstem Wirkungsgrad in Betrieb ist. Die gesuchte Lösung ist somit ein Vektor \vec{z} der Dimension n :

$$\min \left[(k_1 \quad k_2 \quad \dots \quad k_n) \cdot \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \right] = \min [\vec{k} \cdot \vec{z}] \quad (3-3)$$

mit

\vec{k} = Kostenfunktionswerte

n = Anzahl Zeitschritte

Es ist zu betonen, dass die Kostenfunktion unabhängig vom Lösungsvektor \vec{z} ist. Wäre dem nicht so, würde sich die Kostenfunktion abhängig vom gewählten Lösungsvektor verändern, weil der Wirkungsgrad die Kosten beeinflussen würde. Es gilt: $\vec{k} \neq f(\vec{z})$.

3.3.2 Aufbau des Optimierers

Die Umsetzung des Optimierers wird durch die einzuhaltenden Randbedingungen aus Gleichung 3-1 und 3-2 erschwert. Darum muss ein neuartiger Optimierer erstellt werden, der den Zustand des Speichers aus Kapitel 3.4 nach jeder Veränderung des Lösungsvektors überprüft und

gegebenenfalls die Lösung anpasst. Bild 3-7 zeigt ein Flussdiagramm des Optimierers. Zur besseren Übersicht werden die einzelnen Schritte mit Kurzbezeichnungen dargestellt, die im Fließtext und Tabelle 3-2 erklärt werden.

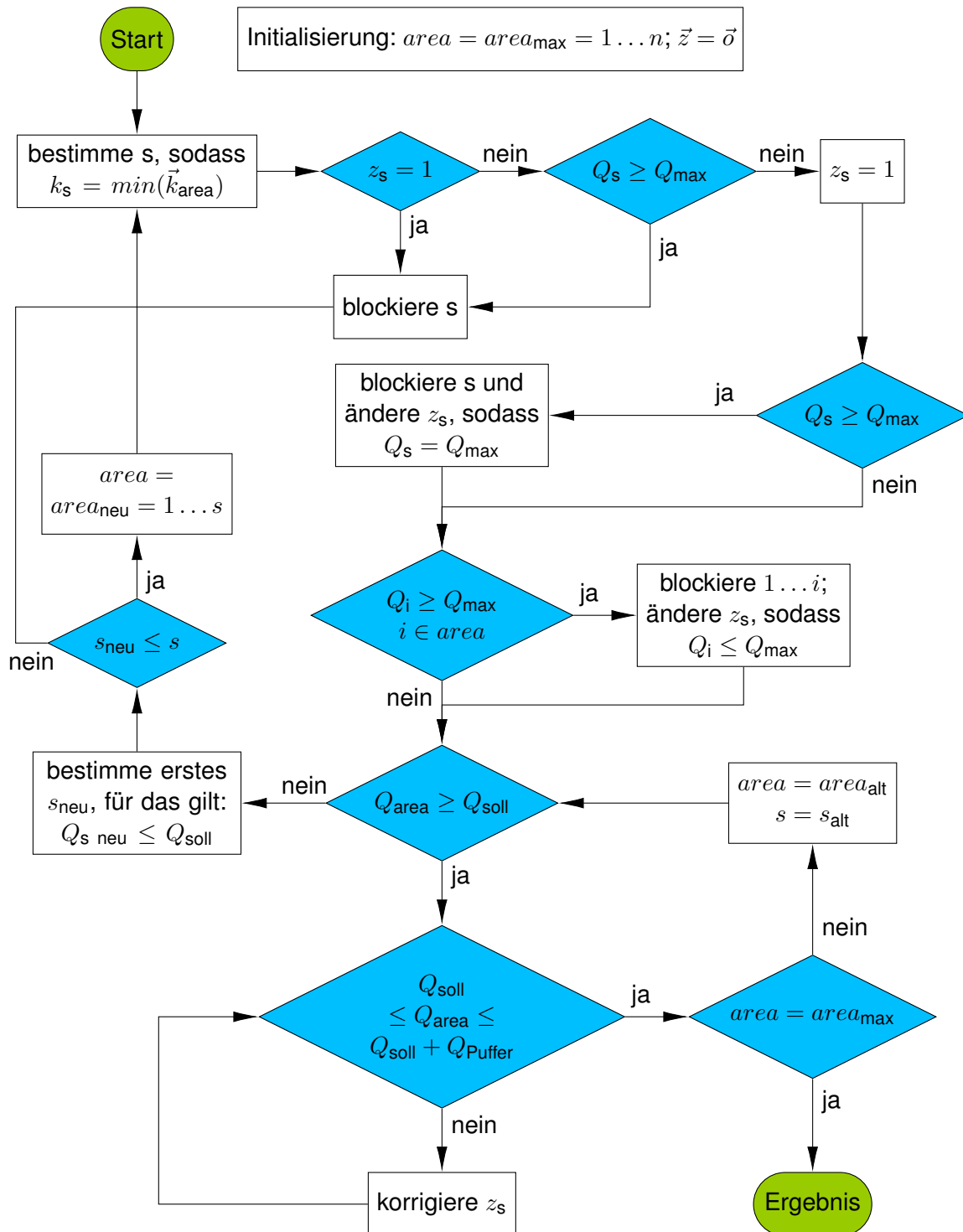


Bild 3-7: Flussdiagramm des Optimierers

Das Hauptprinzip des Optimierers ist es, einen spezifischen Bereich (*area*) zu behandeln, und darin an dem Minimum der Kostenfunktion (repräsentiert durch \vec{k}) den Speicher soweit zu beladen bzw. den Lösungsvektor \vec{z} zu erhöhen, bis ausreichend Energie für den ganzen Abschnitt zur Ver-

fügung steht. Diese betrachtete Stelle ist s . Wenn die Energie nicht ausreicht, wird entschieden, ob vor s ausreichend Energie vorhanden ist. Trifft dies zu, wird $area$ verkleinert auf den Bereich vor und einschließlich s ($area = area_{neu} = 0 \dots s$). Andernfalls bleibt $area$ unverändert. Danach wiederholt sich der Vorgang. Um eine Endlosschleife zu verhindern, wird das Minimum der Kostenfunktion mit einer Maskierung versehen, die alle Stellen blockiert, in denen der Lösungsvektor den Maximalwert besitzt ($z_s = 1$) oder der Speicher vollständig beladen ist ($Q_s = Q_{max}$).

Sobald ausreichend Energie in $area$ vorhanden ist, wird eine zweite Schleife betreten, in der die Energie auf einen Wert zwischen Minimal- und Maximalbetrag (Minimalbetrag plus Pufferbetrag) gebracht wird ($Q_{soll} \leq Q_{area} \leq Q_{soll} + Q_{Puffer}$). Das geschieht durch iterative Anpassung des Lösungsvektoreintrags z_s . Die Puffergröße beträgt 20 kJ^{III}, wenn $area$ ungleich $area_{max}$ ist, und sonst 0,2 kWh (0,72 MJ). (Beide Werte werden anhand von Vorsimulationen bestimmt und sind anpassbar, wobei eine zu geringe Puffergröße mehr Iterationen und somit eine längere Rechenzeit zur Folge hat.) Liegt die Energie im Sollwertbereich wird $area$ zum zuletzt betrachteten Bereich $area_{alt}$ vergrößert, der gespeichert ist, und s wird zu s_{alt} geändert. Daraufhin beginnt die zweite Schleife nach Prüfung der Energiemenge Q_{area} , indem die erste Schleife betreten wird. Wenn $area$ gleich $area_{max}$ gilt, ist die Optimierung abgeschlossen und gibt das Ergebnis aus. Die Pufferzone kann als Repräsentation der Regelungsgenauigkeit verstanden werden, wirkt Rechenungenauigkeiten entgegen und beschleunigt die Iteration bis zum Erreichen des erforderlichen Betrags.

Zwischen allen Schritten des Optimierers muss gewährleistet sein, dass die Randbedingungen eingehalten werden (Gleichung 3-1 und 3-2). Darum wird nach jeder Veränderung von \vec{z} bzw. z_s überprüft, ob der Speicher überladen ist ($Q_s \geq Q_{max}$ bzw. $Q_i \geq Q_{max}$). Falls dem so ist, wird z_s angepasst, sodass der Speicher vollständig beladen ist (mit der Genauigkeit der Puffergröße). Ein Unterschreiten der Minimalbeladung des Speichers wird durch die Funktionsweise des Optimierers verhindert (Genaueres in Kapitel 3.3.3).

Die im Optimierer verwendeten Energien Q (alle in J) und Leistungen P (alle in W) berechnen sich wie in Tabelle 3-2 erläutert. Kapitel 2.1 erklärt die Grundlagen zu den Energie- und Leistungsrechnungen, wobei eine Leistung P als Wärmeleistung/-strom \dot{Q} bezeichnet wird.

Tabelle 3-2: Berechnung der Energien im Optimierer

Zeichen	Berechnung	Beschreibung
Q_{max}	gegeben	Maximalbeladung Speicher
Q_0	gegeben	Initialbeladung Speicher
$P_{Erzeuger}$	gegeben	therm./elektr. Leistung Energieerzeuger
\vec{P}_{Haus}	gegeben	Vektor mit Heizlastvorhersage d. Gebäudes
Q_{Haus}	$P_{Haus} * \Delta t$	Wärmebedarf Gebäude
Q_{area}	$Q_{i-1} + z_i * P_{Erzeuger} - Q_{Verlust\ i} \mid i \in area$	Speicherbeladung in $area$
Q_{soll}	$Q_{i-1} + Q_{Haus\ i} - Q_{Verlust\ i} \mid i \in area$	Erforderliche Wärmemenge in $area$
Q_s	$Q_{i-1} + Q_{Haus\ i} - Q_{Verlust\ i} \mid i \in [1, s]$	Speicherbeladung in Schritt s

3.3.3 Extremfälle der Optimierung

Es gibt Heizlastvorhersagen, bei denen der Optimierer kein vollendetes Ergebnis ermitteln kann. Die kann auftreten, wenn der Speicher oder der Energieerzeuger falsch dimensioniert wird oder

^{III}Bei einem 1 m³-Wasserspeicher mit einer Temperaturdifferenz von 60 K entspricht das ca. 0,008 % der Maximalbeladung

die Vorhersage fehlerhafte zu hohe Werte beinhaltet. In allen Fällen – 1. zu kleiner Speicher, 2. zu leistungsschwacher Energieerzeuger und 3. zu hohe Heizlastvorhersagen – kann es passieren, dass die erforderliche Wärmemenge selbst bei ständiger Volllast des Energieerzeugers nicht erreicht werden kann. Alle Fälle können durch eine konservative Bauteilauslegung und eine Begrenzung der Heizlastvorhersage nach oben gelöst werden. Das ist ineffizient, weil Energieerzeuger hinsichtlich hoher Auslastung im Betrieb ausgelegt werden sollen. So kann es vorteilhaft sein, eine kurze Zeit der Unterversorgung in Kauf zu nehmen, um einen Arbeitspunkt näher am Effizienzoptimum zu erreichen. Die Zeit der Unterversorgung soll durch den Speicher ausgeglichen werden können.

Daher wird durch die Konzeption des Optimierers bei Unterversorgung versucht, den Speicher in den vorherigen Zeitschritten mehr als benötigt zu beladen, um einige Stunden mit höherer Wärmeleistung des Gebäudes als der des Energieversorgers überbrücken zu können. Einzig bei zu langer Zeit der Unterversorgung des Energieerzeugers gibt der Optimierer einen Fehler aus. In der Praxis wäre das mit geringfügig geringeren Temperaturen im Gebäude verbunden. Diese Entscheidung wird aus Gründen der Vergleichbarkeit zu bedarfsorientierten Regelstrategien getroffen.

3.4 Speichermodell

Das Speichermodell zur Optimierung aus Kapitel 3.3.2 bildet das Verhalten eines Warmwasserspeichers nach und kann mit Energie beladen und entladen werden. Außerdem wird die Verlustleistung nach Gleichung 2-2 berechnet und die entsprechende Energie über der Zeit als Speicherverlust bilanziert. Zusätzlich muss das Speichermodell ein Über- oder Unterbeladen des Speichers vermeiden. Um all dies zu ermöglichen, werden die Speicherparameter aus Tabelle 3-3 implementiert. Alle Parameter sind variabel, weshalb die für die MIL-Simulation verwendeten Werte in Tabelle 3-3 angegeben sind.

Tabelle 3-3: Notwendige Parameter für das Speichermodell

Parameter	Wert für die MIL-Simulation ^{IV}	Bemerkung
Speichermedium	Wasser	
Druck	1,25 bar	
Speichergröße	1 m ³	
Wärmeverlustrate	3,63 W/K	
min. Speichertemperatur	35 °C	untere Grenztemperatur, ab der der Speicher als leer gilt
Umgebungstemperatur	24 °C	wird für die Verlustrechnung verwendet

Der Wärmeverlust wird mit einer nach DIN EN 12977-3:2018 bestimmten Wärmeverlustrate berechnet. Dies birgt Verbesserungspotential, weil Speicher in der Praxis geschichtet werden. Für die Ziele dieser Arbeit ist irrelevant, wie die Energie im Speicher gespeichert ist, weshalb auf ein komplexes Schichtspeichermodell verzichtet wird.

^{IV}Die Betriebsparameter basieren auf dem Kombispeicher TERMO der Firma „Wagner Solar GmbH“; Art.-Nr.: 130 150 10

Die Berechnungen von Verlustenergie und Be- bzw. Entladung werden in festen Zeitschritten vorgenommen. Hierbei wird bei jeder Änderung der Speicherbeladung eine Iteration durchgeführt, bis ein definierter Fehler – in der MIL-Simulation liegt dieser bei 0,05 K – in der neuen Speichertemperatur unterschritten wird. Die Iteration ist wegen der Temperaturabhängigkeit der spezifischen Wärmekapazität von Wasser notwendig, wobei für die Berechnung der Wärmeleistung nach Gleichung 2-2 eine mittlere spezifische Wärmekapazität verwendet wird:

$$\bar{c}_p = \frac{c_{p1} + c_{p2}}{2} \quad (3-4)$$

wobei Index 1 für den Zustand vor und Index 2 für den Zustand nach der Speicherbeladungsänderung steht.

3.5 Model-in-the-Loop-Simulation

Die Funktionen des Applikationsmodells werden in MIL-Simulationen untersucht und bewertet. Aus deren Ergebnisse lassen sich Einsparpotentiale der gewählten Kosten gewinnen und die Qualität der Vorhersage und Optimierung in einer der Realität entsprechenden Situation bestimmen. Dabei wird der Strompreis und der erwartete CO₂-Ausstoß während der Stromerzeugung als Kostenfunktion für den Optimierer verwendet. Außerdem wird die Auswirkung der Länge der verfügbaren Trainingsdaten untersucht. Als Versuchsjahr werden die Daten aus dem Jahr 2019 vom DWD und dem ENTSO-E verwendet. In einer Simulationsreihe werden NNs mit einer während dem Training verfügbaren Datensatzlänge von vier Jahren, einem Jahr und vier Monaten getestet. Das Verhältnis zw. Trainings-, Validierungs- und Testdaten ist bei 0,7 zu 0,2 zu 0,1. Wenn nicht anders beschrieben, wird ein NN mit Trainingsdaten über vier Jahre verwendet.

Jede optimierte Regelung wird mit einer bedarfsorientierten Regelung verglichen, bei der kein Speicher vorhanden ist. Somit entspricht theoretisch der Energieaufwand der bedarfsorientierten Regelung in jedem Zeitschritt dem Energiebedarf des Gebäudes. In der Realität sind Regelungen mit Speicher mehr verbreitet, die mit einer Hysterese-Regelung auf einem bestimmten minimalen Temperaturniveau gehalten werden. Darum wird zusätzlich zum Gebäudeenergiebedarf eine konstante Speicherverlustleistung hinzugerechnet, die einer Speichertemperatur von 37 °C entspricht. Der bedarfsorientierte Energieaufwand setzt sich folglich aus der Gebäudeheizlast addiert mit der konstanten Verlustleistung von 47 W zusammen. Ein Wirkungsgrad bzw. COP-Wert (engl. von „Coefficient of Performance“) einer in der Praxis verwendeten Wärmepumpe zur Gebäudeheizung wird nicht berücksichtigt, sodass von einer Stromheizung ausgegangen wird. Außerdem sind alle Stromkosten in Börsenstrompreisen angegeben; infolgedessen werden Steuern und Abgaben nicht berücksichtigt. Jegliche Absolutwerte müssen somit als Anhaltspunkt aufgefasst werden, die nicht ohne besondere Umrechnung in reale Situationen übertragbar sind.

Die Simulationsvorhersage verwendet historische Wetterdaten für das Aufwärmen des NNs und Vorhersagen der Heizlast. Dadurch entsteht ein systematischer Fehler, weil die historischen Daten nicht den Fehler der Wettervorhersage bzw. der Strommarktvorhersage (Day-Ahead-Vorhersage des Preises bzw. der Leistung EE) beinhalten. Außerdem sind die Heizlastdaten, mit denen trainiert wird, mit denselben historischen (Wetter-)Daten simuliert worden (Kapitel 3.1). Während der Anwendung des Applikationsmodells in einem echten Gebäude werden real gemessene Gebäudewerte für Training und Vorhersage verwendet, wohingegen die Wetterdaten in der Praxis nicht am Gebäude, sondern in einer geografisch entfernten DWD-Wettermessstation gemessen werden. (Die erwartete maximale Entfernung liegt bei ca. 50 km in gering besiedelten Gebieten im Norden Deutschlands.) Dadurch entsteht ein geografischer Versatz zwischen Wetter- und Gebäudedaten, der im besten Fall einen zeitlichen Versatz bedeutet. In dem Fall bewirkt das Wetter

durch Windbewegungen zeitversetzt die gleichen Werte an den Standorten von Gebäude und Wetterstation. Im schlechtesten Fall eliminiert der geografische Versatz die Korrelation zwischen Wetter und Gebäudeverhalten. Die Auswirkungen dieser systematischen Fehler können durch diese Arbeit nicht bewertet werden und erfordern weiterführende Studien. Es kann erwartet werden, dass zumindest ein Teil des Fehlers durch zeitlich versetzte Daten während des Trainings vom NN erlernt und kompensiert werden kann.

Zusätzlich zu den aus der Gebäudesimulation stammenden Werten (siehe Kapitel 3.1) wird das Potential des Applikationsmodells mit gegebenen real gemessenen Werten eines Einfamilienhauses mit 160 m² Wohnfläche, Baujahr 1996, 30 km nördlich von Stuttgart untersucht. Damit sollen die Werte aus den Simulationsdaten verifiziert und die Funktionsweise des Applikationsmodells unter realen Bedingungen überprüft werden. Die verwendeten Realdaten sind Messungen der Heizleistung mittels Vorlauf- und Rücklauf- und Volumenstromes im Heizkreislauf des Realgebäudes. Diese liegen mit einer zeitlichen Auflösung von einer Minute für den Zeitbereich zwischen dem 21.1.2018 und 10.5.2021 vor. Die Besonderheit im Vergleich zu den Simulationsdaten liegt in der Nachtabsenkung, die über Nacht im Betrieb manuell einprogrammiert wird und sich beeinflusst vom Nutzerverhalten in zeitlichem Beginn und Ende unterscheidet. Das stellt das NN vor die Herausforderung, dieses Verhalten vorhersagen zu müssen. Ein typischer Verlauf der Heizleistung über eine Woche ist in Bild 3-8 im Vergleich zu den Simulationsdaten des Beispielgebäudes für die Kalenderwoche 50 des Jahres 2019 abgebildet. Die Absolutwerte sollten nicht direkt verglichen werden, weil es sich um zwei verschiedene Gebäude an Standorten mit 30 km Entfernung handelt. Da der Nutzer den Beginn des Heizens nach der Nachtabsenkung manuell

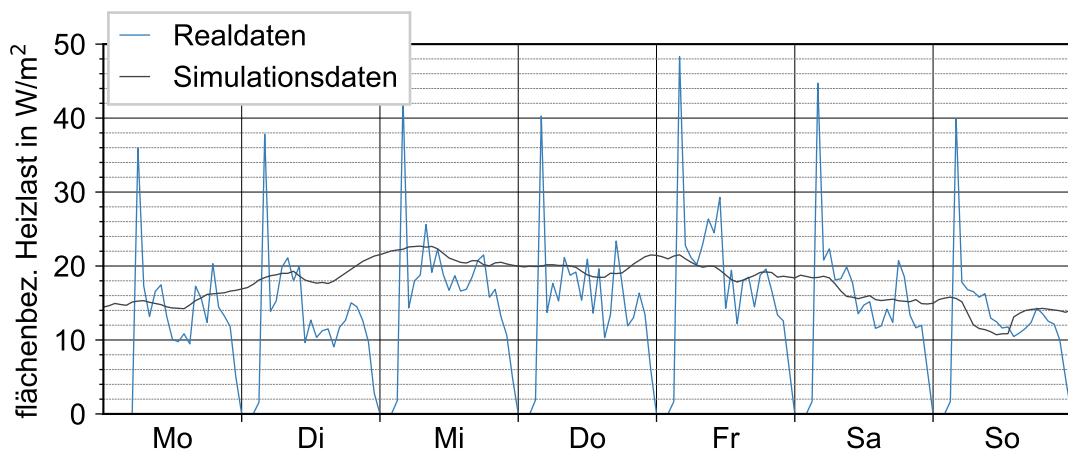


Bild 3-8: Heizlastverlauf des simulierten Beispielgebäudes und die gemessenen, in stündliche Auflösung reduzierten Messdaten des Realgebäudes für die Kalenderwoche 50 im Jahr 2019

einstellt, steht dieser Wert auch einem hypothetisch installierten SMGW zur Verfügung. Darum werden zwei NNs mit den Realdaten untersucht, wobei eines als Eingabeparameter dieselben nutzt, wie die NNs, die mit den Simulationsdaten trainiert wurden (siehe Kapitel 4.2). Das zweite bekommt zusätzlich die Information der manuellen Steuerung. Dieser Wert wird aus den Rohdaten anhand des auftretenden Volumenstroms extrahiert, weil die manuelle Steuerung nicht protokolliert ist. Der Wert ist bei abgeschalteter Heizkreispumpe Null (kein Durchfluss möglich und somit keine Abgabe von Heizleistung an das Realgebäude) und über den Tag oder bei eingeschalteter Heizkreispumpe Eins.

Der Realdatensatz wird in einen Trainingsdatensatz und einen Simulationsdatensatz aufgeteilt. Die Simulationsdauer wird als Vergleich zu den Simulationsdaten verwendet und umfasst ein Jahr

am Ende des Realdatensatzes (10.5.2020 bis 10.5.2021). Der Rest wird mit demselben Verhältnis wie bei den Simulationsdaten in Trainings-, Validierungs- und Testdatensatz aufgeteilt und zum Training für die NNs verwendet.

Das Realgebäude besitzt eine die flächenbezogene Heizleistung beeinflussende Besonderheit. In einem Wohnraum ist ein Kaminofen installiert, der an den Heizkreis angebunden ist. Dadurch wird dieser teilweise in der flächenbezogenen Heizleistung repräsentiert, weil der Ofen den Speicher belädt, teilweise wird der Raum geheizt, ohne dass es im Heizkreislauf gemessen werden kann. Dadurch sind die auftretenden Heizleistungsdaten vor allem am Abend geringer als sie es ohne Ofen wären. In Bild 3-8 ist dieser Fall am Dienstag zu erkennen. Die Ergebnisse beeinflusst dieser systematische Fehler, indem die absoluten Werte als zu gering angenommen werden. Da die Werte nur untereinander verglichen werden, spielt dieser Fehler eine untergeordnete Rolle.

4 Ergebnisse und deren Auswertung

Dieses Kapitel stellt die Ergebnisse dar und diskutiert diese. Zunächst werden verschiedene Arten an Normalisierungsmethoden und Long Short-Term Memory-(LSTM)-Zellen sowie Einstellungen während des Trainings des neuronalen Netzes (NN) untersucht. Anhand dieser werden die Parameter mit den qualitativ hochwertigsten Ergebnissen verwendet, um Model-in-the-Loop-(MIL-)Simulationen durchzuführen. Die MIL-Simulationen erlauben es, die Vorhersagequalität während des Betriebs eines Smart-Meter-Gateways (SMGW) abzuschätzen und die Qualität des Optimierers zu bewerten. Zuletzt wird das Applikationsmodell mit einer MIL-Simulation basierend auf Realdaten mit Nachtabsenkung getestet und auf das Einsparpotential hin untersucht.

4.1 Einstellungen des neuronalen Netzes

Zu Beginn werden im Hinblick auf die Vorhersagequalität des NNs die in Kapitel 3.2.2 beschriebenen Normalisierungsmethoden, die Anzahl an LSTM-Einheiten und verschiedene Trainingsparameter (Stapelgröße, TensorFlow-Einstellungen `patience` und `restore_best_weights`) untersucht und die hochwertigsten gefundenen Einstellungen für die Bewertung des Applikationsmodells in den MIL-Simulationen verwendet. Da die Eingabeparameter zeitlich nach der Auswahl der LSTM-Parameter optimiert werden, wird für die Evaluationen der genannten LSTM- und Trainingsparametern immer derselbe Eingabeparameter verwendet. Dieser besteht aus den Eingabeparametern flächenbezogene Heizlast, Außenlufttemperatur, Globalstrahlung und Elevation.

Erste Tests mit dem TensorFlow-Parameter `patience` für die Verzögerung im Training (siehe Kapitel 3.2.3) bringen ein eindeutiges Ergebnis. So wird die Trainingszeit durch die Erhöhung der Verzögerung von zwei auf sechs Epochen gesteigert, während die Qualität der Vorhersage zunimmt. Hierbei wird der TensorFlow-Parameter `restore_best_weights` aktiviert. Dieser Wert speichert die trainierte Gewichtsmatrix nach jeder Epoche, wenn diese eine bessere Vorhersagequalität als die bisherige ermöglicht. So wird ein Übertraining vermieden, welches bei größeren Werten für die Verzögerung (`patience`) eine mögliche Folge darstellt. Alle weiteren in dieser Arbeit verwendeten NNs nutzen eine Verzögerung von sechs Epochen, wobei die von der Vorhersagequalität leistungsstärkste Gewichtsmatrix am Ende verwendet wird.

Die Anzahl der LSTM-Einheiten wird in Versuchsreihen ermittelt, die verschiedene Stapelgrößen („Batch size“) und unterschiedliche Normalisierungsmethoden verwenden. Als Vergleichsbasis wird die gleichmäßige Normalisierung angewendet bei einer Stapelgröße von 32. In vier Versuchsreihen werden LSTM-Parameter variiert. Bild 4-1 zeigt die Ergebnisse, wobei der mittlere quadratische Fehler (engl. Mean Squared Error, MSE) als ein Wert für die Vorhersagequalität verwendet wird. Die Normalisierung ist hierbei herausgerechnet, was einen Vergleich ermöglicht. Bild 4-1 zeigt, dass weniger als 15 LSTM-Einheiten für jede Einstellung nicht sinnvoll sind. Ab 20 LSTM-Einheiten zeigen sich bei der klassischen Normalisierung keine großen Veränderungen. Dem entgegengesetzt weist die gleichmäßige Normalisierung einen Trend zur Verschlechterung der Fehlerquote auf. Hierbei ist der Absolutfehler höher als bei der klassischen Normalisierung. Somit ist entgegen den Erwartungen die klassische der gleichmäßigen Normalisierung vorzuziehen. Die Stapelgröße ist im Vergleichsmodell am leistungsfähigsten. Sowohl bei einer Verringerung auf acht als auch bei einer Vergrößerung auf 128 nimmt die Fluktuation des MSE über die Anzahl der LSTM-Einheiten zu. Der Absolutfehler verhält sich äquivalent.

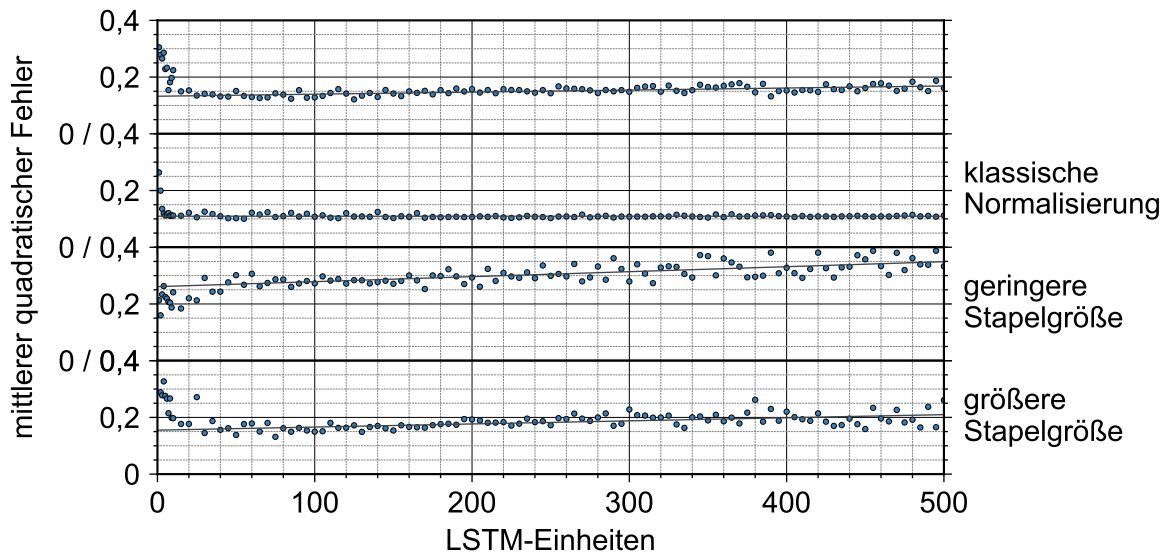


Bild 4-1: Vorhersagegenauigkeit bei verschiedenen Einstellungen während des Trainings über die LSTM-Unit-Anzahl

Auf Basis der Ergebnisse werden in allen weiteren Berechnungen für das NN folgende Einstellungen verwendet:

Tabelle 4-1: Verwendete Trainings- und LSTM-Einstellungen

Variable	verwendeter Wert
Normalisierung	klassisch
Verzögerung (patience)	6
restore_best_weights	True
Stapelgröße	32
LSTM-Units	60

4.2 Eingabeparameter

Es stehen 12 Parameter des Wetters aus der Gebäudesimulation und aus entsprechenden Berechnungen zur Verfügung, die als Eingabeparameter für das NN verwendet werden können. Diese sind in Tabelle 4-2 aufgelistet. Hierbei wird die Verwendung des Wertes zur Nutzeranwesenheit aus der Simulation bewusst nicht berücksichtigt, weil dieser Eingangsparameter in der Praxis selten zur Verfügung steht. Somit wäre ein Algorithmus, der die Nutzeranwesenheit benötigt, in den meisten Fällen nicht realisierbar, obwohl dieser Wert nach Abhandlungen aus der Literatur eine Verbesserung der Vorhersage bedeutet.

Da eine Untersuchung aller Kombinationen der Eingangsvektoren nicht in praktikabler Zeit durchführbar ist (12 Parameter bedeuten $12! \approx 479$ Millionen unterschiedliche Kombinationen) werden die verwendeten Eingabeparameter zuerst nach Erfahrung und erwarteter Abhängigkeit von der Heizlast selektiert. Dafür werden die Modellvarianten A01 bis A26 erstellt, die jeweils unterschiedliche Kombinationen der Eingabeparameter verwenden. Die entsprechende Zuordnung der Para-

meter zu den Modellen ist in Tabelle 4-2 gegeben und die zugehörigen Vorhersageergebnisse der Modelle sind mit denormalisierten MSEs der Test- und Validierungsdaten, deren Mittelwert und trainierter Epochenanzahl in Bild 4-2 abgebildet.

Tabelle 4-2: Mögliche Eingangsparameter und deren Zuordnung zu den Modellen A01 bis A26

Modellvariante	Eingabeparameter												
	flächenbez. Heizlast	Bedeckungsgrad	Außenlufttemperatur	Globalstrahlung	Taupunkt	Luftdruck	positive Elevation	Windgeschw. Nord	Windgeschw. Ost	Wochenende	Elevation	cos(Azimuth)	
A01	✓												
A02	✓		✓										
A03	✓			✓									
A04	✓		✓	✓									
A05	✓		✓	✓							✓		
A06	✓		✓	✓						✓	✓		
A07	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
A08			✓	✓									
A09	✓		✓	✓			✓						
A10	✓						✓	✓					
A11	✓						✓					✓	
A12	✓	✓											
A13	✓	✓	✓	✓	✓	✓							
A14	✓						✓	✓	✓	✓	✓	✓	
A15	✓		✓	✓	✓	✓	✓	✓					
A16	✓		✓	✓		✓	✓	✓					
A17	✓		✓	✓							✓	✓	
A18	✓		✓	✓						✓	✓	✓	
A19	✓		✓	✓			✓	✓	✓	✓	✓	✓	
A20	✓		✓	✓			✓	✓	✓	✓	✓	✓	
A21	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	
A22	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
A23	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
A24	✓	✓	✓	✓	✓	✓	✓				✓	✓	
A25	✓		✓	✓			✓				✓	✓	
A26	✓		✓	✓			✓					✓	

Mit der Zuordnung aus Tabelle 4-2 wird ein Überblick über den Einfluss verschiedener Parameter auf die Vorhersagequalität gegeben. So kann mit den Modellen A01 bis A05 bestätigt werden, dass sowohl die Außenlufttemperatur als auch die Globalstrahlung die Vorhersagegenauigkeit der Heizlast verbessern. Gleichzeitig wird deutlich, dass mehr Parameter die Vorhersagequalität negativ beeinträchtigen. So verschlechtert sich die Vorhersage von Modell A05 auf Modell A07, das alle Parameter als Eingabe verwendet. Des Weiteren wird das nicht nachvollziehbare Verhalten des NN deutlich, weil die hochwertigste Vorhersagequalität mit Modell A20 erreicht wird, wohingegen z. B. Modell A17 schlechter als Modell A05 abschneidet, obwohl es dieselben Parameter wie Modell A05 verwendet und zusätzlich teilweise die von Modell A20. Die Vorhersagequalität nimmt

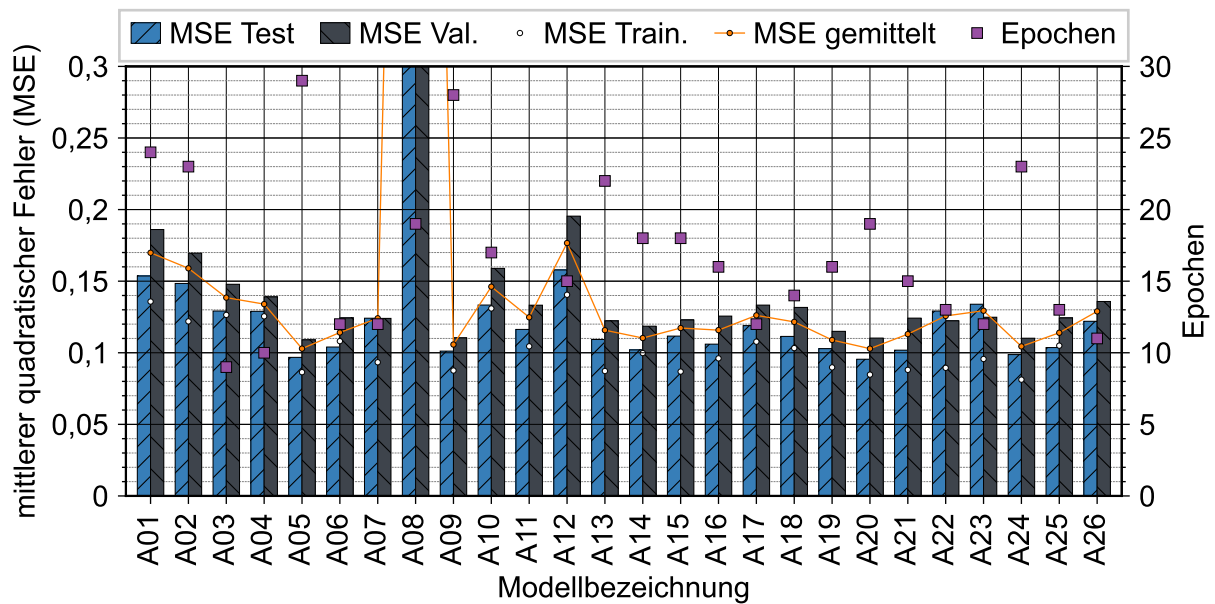


Bild 4-2: Vorhersagegenauigkeit bei verschiedenen Eingabeparametern

bei Modell A17 durch das Hinzufügen der Parameter ab, während sie bei Modell A20 durch mehr Parameter zunimmt. Daraus folgt, dass eine geeignete Vorhersage der Leistung des NNs anhand der Eingangsparameter nicht durch eine lineare Methode zu bewältigen ist.

Das Ausreißermodell A08 zeigt, dass der mit Abstand wichtigste Parameter die flächenbezogene Heizlast ist, wohingegen es keine erkennbare Korrelation von der Vorhersagequalität mit der Anzahl der trainierten Epochen gibt. Das bessere Verhältnis von Test-MSE zu Validierungs-MSE ist der Grund für das in dieser Arbeit benutzte Verfahren, das sich nicht an der Vorhersagequalität der Testdaten orientiert, sondern in gleichen Teilen an der Vorhersagequalität von Test- und Validierungsdaten. Dadurch vergrößert sich der für die Vorhersagequalität verwendete Bereich auf über ein Jahr an Daten. Damit sind die im Betrieb des Applikationsmodells auftretenden Daten genauer repräsentiert.

Das Optimierungsproblem der Auswahl der Eingabeparameter wird mit einem manuellen Ausführen eines an genetische Optimierer angelehnten Optimierungsverfahrens gelöst. Dabei wird mit dem Grundmodell A01 begonnen und eine Versuchsreihe erstellt, in der Modelle enthalten sind, die alle Eingangsparameter des Grundmodells beinhalten, sowie jeweils einen zusätzlichen Wert¹. Diese Modelle B01 bis B13 werden trainiert und das Modell mit der höchsten Vorhersagequalität ausgewählt. Dieses stellt daraufhin das neue Basismodell dar. Dann wird das Vorgehen wiederholt, bis keine Verbesserungen durch neue Parameter mehr erkennbar sind. Die untersuchten Modelle sind in Bild 4-3 und Bild 4-4 dargestellt, wobei die Basismodelle mit X bezeichnet und die zugehörigen Versuchsreihen mit vertikalen Linien voneinander abgegrenzt sind. Die zugehörige Parameterzuordnung ist in Anhang C zu finden.

¹mit einer Ausnahme: Die Parameter „Windgeschwindigkeit Nord“ und „Windgeschwindigkeit Süd“ werden zusätzlich gemeinsam getestet.

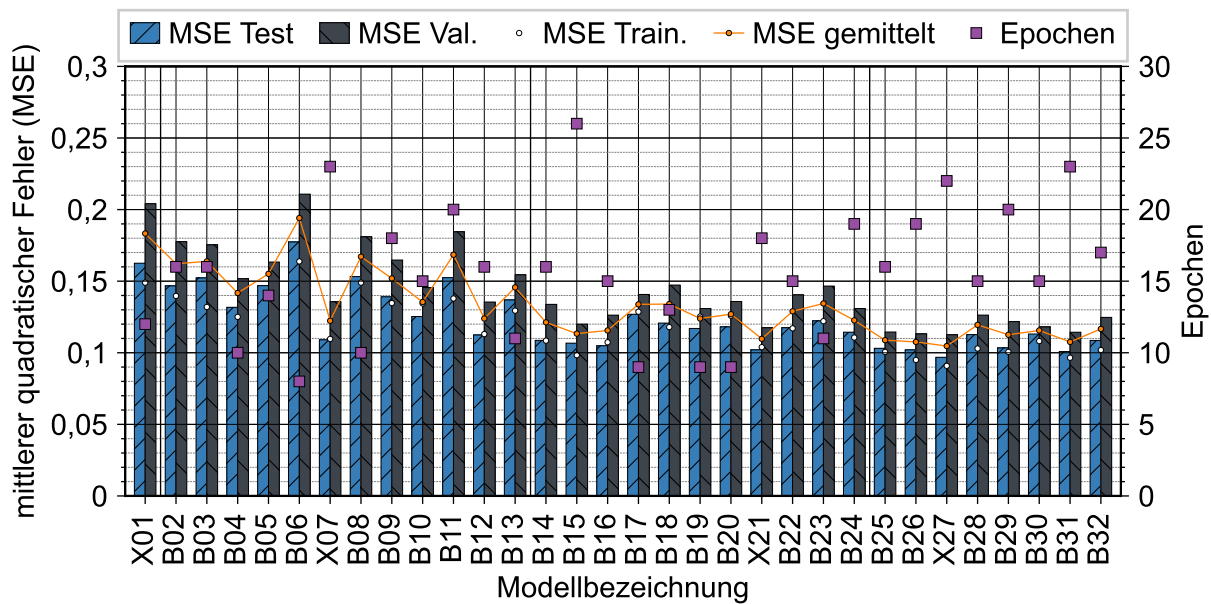


Bild 4-3: Vorhersagegenauigkeit bei verschiedenen Eingabeparametern aufbauend auf der Vorhersage einzig mit der flächenbezogenen Heizlast (Modell X01 = A01)

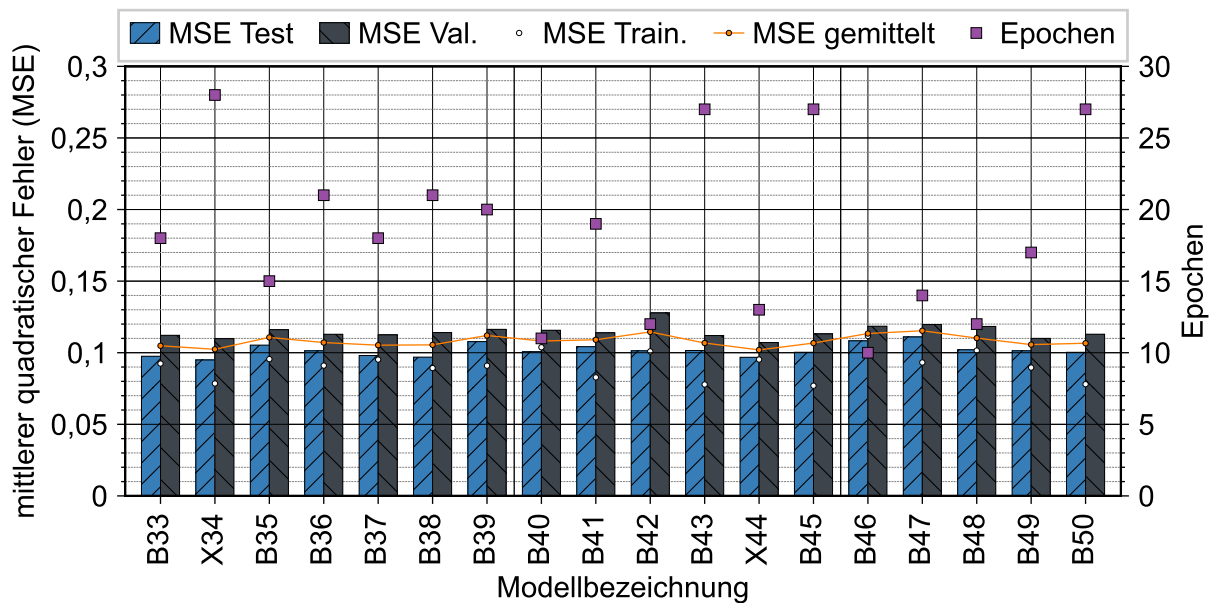


Bild 4-4: Vorhersagegenauigkeit bei verschiedenen Eingabeparametern aufbauend auf dem (Basis-)Modell X27 mit den Parametern: flächenbezogene Heizlast; Globalstrahlung; positive Elevation und Windgeschwindigkeiten Nord und Ost

Die Vorhersagequalität verbessert sich stetig mit jedem neuen Basismodell. Durch größere Eingangsvektoren steigt die Anzahl benötigter Epochs für das durchschnittliche Training bis zum Vorhersageoptimum mit geringer Korrelation. Es stellt sich das Modell X44 mit einem MSE von $0,10 \text{ (W/m}^2\text{)}^2$ bei der Vorhersage von einem Zeitschritt (ohne Autoregression) als das leistungsstärkste NN heraus. Es wird für alle Vorhersagen in den MIL-Simulationen, die auf Gebäudesimulationsdaten aus TRNSYS basieren, verwendet. Die Eingabeparameter sind:

- flächenbezogene Heizlast
- Außenlufttemperatur
- Globalstrahlung
- positiver Teil der Elevation
- Elevation
- Windgeschwindigkeiten Nord und Ost

Es ist anzumerken, dass alle Werte in Bild 4-4 nahe beieinanderliegen, sodass jedes dieser Modelle für die Vorhersage verwendet werden kann. Für derartig geringe Größenunterschiede bezüglich der Vorhersagequalität hat die initiale Einstellung der Gewichte einen größeren Einfluss auf das Ergebnis als die unterschiedlichen Eingabedaten. So veränderten sich bei erneuter Ausführung des manuellen Optimierungsverfahrens die Verhältnisse zwischen den Modellen B33 bis B50 geringfügig, während der Durchschnittswert annähernd konstant bleibt. Dies wird beim Vergleichen von den Ergebnissen der Modelle A01 und X01 aus Bild 4-2 und Bild 4-3 deutlich. Beide Modelle sind von den Einstellungen dieselben, wohingegen die Ergebnisse von zwei unabhängigen Trainingsdurchläufen sich unterscheiden. In diesem Fall bewirkt die Veränderung der initialen Gewichte eine Änderung der Anzahl trainierter Epochen. Dadurch verringert sich der auftretende Fehler in den Trainingsdaten bei gleichzeitig höherer Abweichung zum Fehler in den Test- und Validierungsdaten und höherer Wahrscheinlichkeit des Übertrainings.

4.3 Simulationsergebnisse

4.3.1 CO₂-Ausstoßreduzierung

Zur Reduzierung des CO₂-Ausstoßes wird dem Optimierer der momentane Anteil an Photovoltaik (PV)- und Windkraftanlagen am Strommix übergeben^{II}. Einen Anteil aller erneuerbarer Energien stellt der Verband Europäischer Übertragungsnetzbetreiber ENTSO-E nicht als Day-Ahead-Vorhersage zur Verfügung. Darum wird hinsichtlich der Energieerzeuger, die als Energiequelle die Sonne und den Wind nutzen, optimiert. Auf die Verwendung der historischen Daten wird verzichtet, weil diese im SMGW nicht während der Vorhersage vor Ort zur Verfügung stehen und die Simulationen diesen Fall modellieren sollen.

Zur Bewertung der Daten wird mit dem Applikationsmodell vom ENTSO-E die Erzeugerleistung je Energieerzeugerart abgerufen und mit Emissionsfaktoren vom Umweltbundesamt für das Jahr 2019 in erzeugte Tonnen CO₂ umgerechnet [31]. Die verwendeten Emissionsfaktoren für die fossilen Energieerzeuger sind in Tabelle 4-3 aufgelistet.

Tabelle 4-3: Verwendete Emissionsfaktoren für das Jahr 2019 [31]

Energieerzeugerart	Biomasse	Braunkohle	Gas aus Kohle	Gas	harte Kohle	Öl	Kernenergie	Sonstige nicht Erneuerbare	sonstige Erneuerbare
Emissionsfaktor in g CO ₂ /kWh	270	396	147	201	341	264	0	288	0

Bild 4-5 zeigt die monatliche Menge an ausgestoßenem CO₂ durch den aufgewendeten Strom im Jahr 2019. Daraus wird ersichtlich, dass die höchsten Einsparpotentiale in den Monaten März

^{II}Da der Optimierer die ihm übergebene Kostenfunktion minimiert, wird der Anteil von PV- und Windkraft von 1 subtrahiert, um den Anteil aller anderen Energieerzeuger am Strommix zu erhalten und diesen zu minimieren.

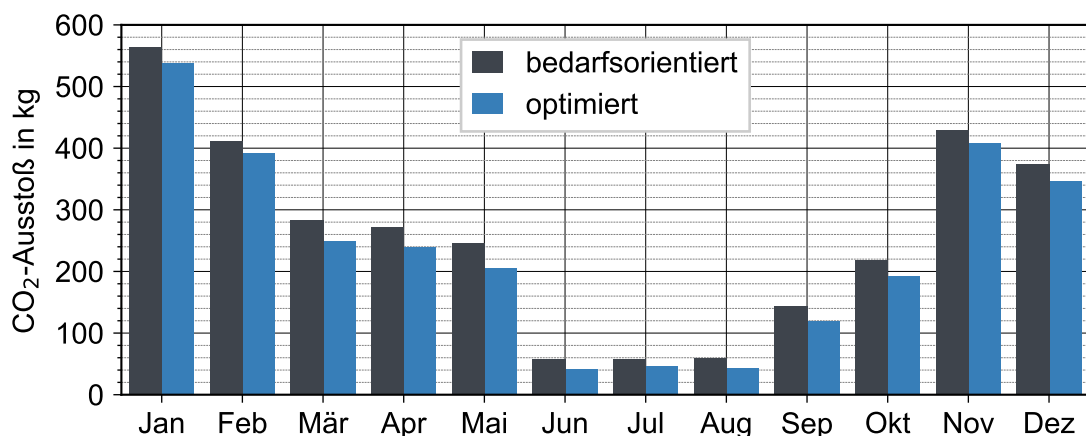


Bild 4-5: CO₂-Ausstoß bei der Optimierung hinsichtlich hohen Strommix-Anteils an Wind- und Solarkraft. Bedarfsorientiert ergeben sich 15,8 kg CO₂/(m²·a); optimiert 14,3 kg CO₂/(m²·a)

bis Mai gehoben werden können. Insgesamt ergibt sich eine CO₂-Einsparung von 9,5 % pro Jahr im Vergleich zur bedarfsorientierten Regelung, was einer Einsparung von 300 kg CO₂/a entspricht. Auf den Quadratmeter gerechnet ergibt dies eine CO₂-Ersparnis von 1,5 kg CO₂/(m²·a). Hochgerechnet auf die Fläche aller Einfamilienhäuser in Deutschland aus der Fortschreibung des Wohngebäude- und Wohnungsbestand vom Statistischen Bundesamt (Destatis), 2021 mit einem Betrag von 1,66 Mrd. m² ergibt sich ein Einsparpotential von über 2,5 Mio. t CO₂/a. Das Ergebnis wird verschlechtert, weil in der Praxis Wärmepumpen zur Wärmebereitstellung verwendet werden, die mit einer konservativ angenommenen Jahresarbeitszahl von mindestens 2,3 die benötigte Energie und die CO₂-Einsparung mehr als halbieren.

Zusammenfassend zeigt sich, dass das Applikationsmodell den CO₂-Ausstoß bei der Wärmebereitstellung von Einfamilienhäusern um ein Zehntel reduzieren kann und gleichzeitig eine Änderung der Regelung hin zu einer anderen Kostenfunktion, wie das Verringern von Lastspitzen, ermöglicht.

4.3.2 Stromkostenreduzierung

Die MIL-Simulation mit der Optimierung hin zu geringen Stromkosten wird in Bild 4-6 mit der bedarfsorientierten Regelung (siehe Kapitel 3.5) verglichen. Eine Stromkostenoptimierung bewirkt eine Kostenreduktion von durchschnittlich 11 % bis maximal 59 % im Monat. Hohe absolute Einsparungen werden in den Monaten Januar, März und Dezember erzielt. Das ganze Jahr betrachtend resultiert eine Einsparung von 180 €, was einer Einsparung von 19 % entspricht. Die bedarfsorientierte Regelung verursacht jährliche Kosten von 940 €. Äquivalent zur CO₂-Ausstoß-Optimierung reduzieren sich die Einsparungen zusammen mit dem Stromaufwand bei Nutzung einer Wärmepumpe. Die Werten können nicht direkt mit Verbraucherpreisen verglichen werden, weil die absoluten Zahlen für private Haushalte durch Steuern und Abgaben auf den Strompreis steigen. Der Durchschnittsstrompreis in den präsentierten Ergebnissen liegt mit 3,8 ct/kWh bei ca. 13 % des Strompreises für Privatanwender (bei angenommenem Strompreis von 30 ct/kWh).

4.3.3 Kosten und CO₂-Ausstoß beider Optimierungen

Beide Optimierungen werden auf ihren Einfluss auf die andere optimierte Größe untersucht. Die Ergebnisse sind in Bild 4-7 und Bild 4-8 dargestellt. Es wird deutlich, dass zwischen den Zielgrößen der Optimierungen eine Korrelation besteht. Wird auf geringe Stromkosten optimiert hat

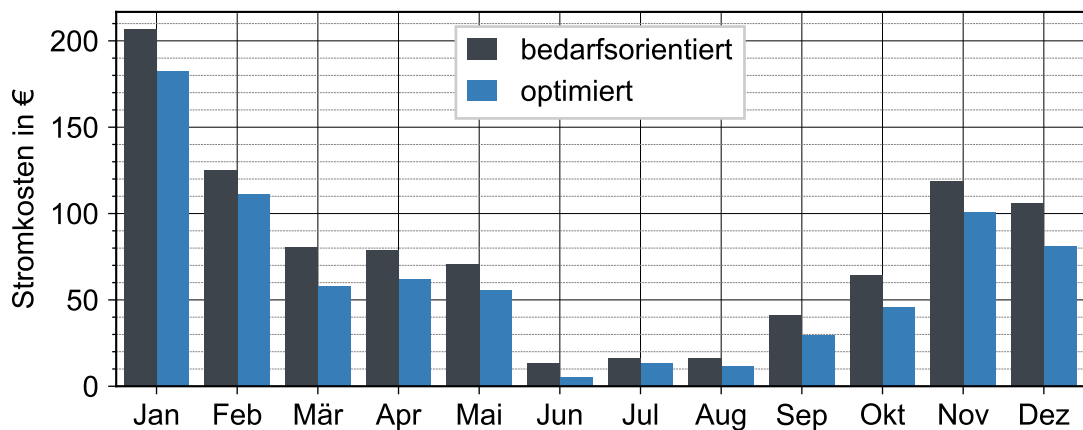


Bild 4-6: Kosten bei der Optimierung hinsichtlich geringen Börsenstrompreises. Bedarfsorientiert ergeben sich $4,7 \text{ €/}(m^2 \cdot a)$; optimiert $3,8 \text{ €/}(m^2 \cdot a)$

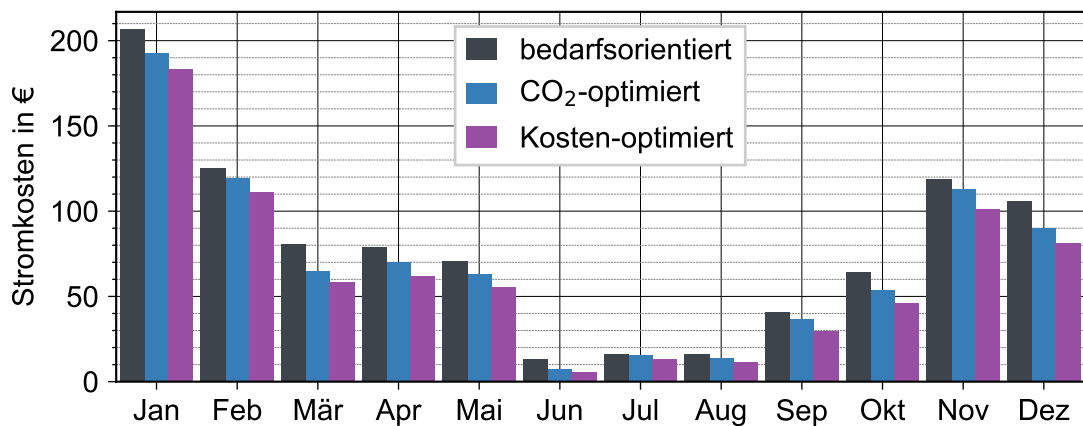


Bild 4-7: Stromkosten bei beiden Optimierungen im Vergleich. Bedarfsorientiert ergeben sich $4,7 \text{ €/}(m^2 \cdot a)$; CO₂-optimiert $4,2 \text{ €/}(m^2 \cdot a)$; Kosten-optimiert $3,8 \text{ €/}(m^2 \cdot a)$

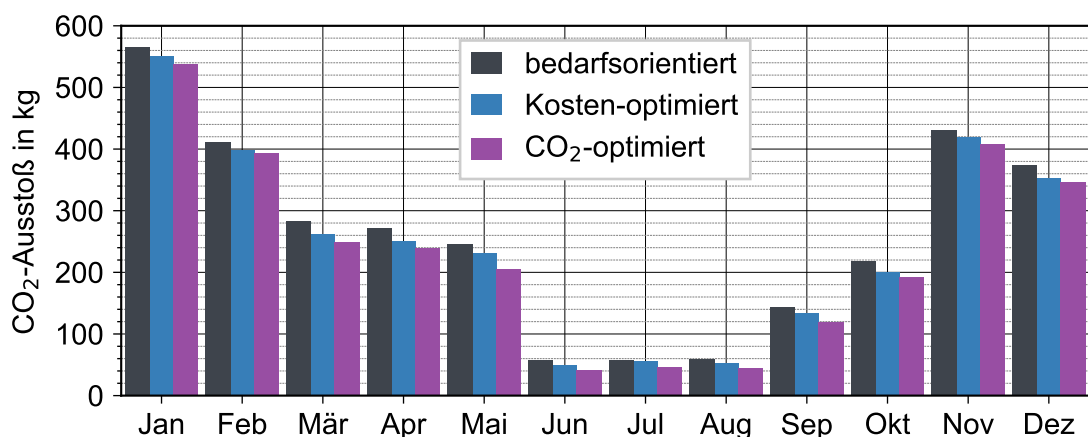


Bild 4-8: CO₂-Ausstoß bei beiden Optimierungen im Vergleich. Bedarfsorientiert ergeben sich $15,8 \text{ kg CO}_2/(m^2 \cdot a)$; Kosten-optimiert $14,9 \text{ kg CO}_2/(m^2 \cdot a)$; CO₂-optimiert $14,3 \text{ kg CO}_2/(m^2 \cdot a)$

das zur Folge, dass auch der CO₂-Ausstoß sinkt (Bild 4-8). So kann mit der Stromkostenoptimierung der CO₂-Ausstoß der Gebäudeheizung um 5,1 % (160 kg CO₂) reduziert werden, was

54 % der Reduktion durch die CO₂-Optimierung bedeutet. Äquivalent reduzieren sich die Stromkosten durch die CO₂-Optimierung um 11 % (99 €), was 55 % der Stromkostensenkung durch die Stromkostenoptimierung entspricht (Bild 4-7). Folglich bewirken beide Optimierungsmethoden gleichzeitig einen Vorteil bezüglich der anderen Optimierungsgröße. Dadurch entsteht ein wirtschaftlicher Vorteil, CO₂ einzusparen. Ebenso hat die Stromkostensenkung eine CO₂-Einsparung zur Folge.

4.3.4 Einfluss Trainingsdatenzeitraum

Die Länge des vorhandenen Trainingsdatenzeitraums ist in der Praxis begrenzt. Darum werden MIL-Simulationen durchgeführt, in denen NNs zum Einsatz kommen, die mit unterschiedlich umfangreichen Trainingsdatensätzen trainiert werden. Dabei wird der Zeithorizont des Trainingsdatensatzes von vier Jahren auf ein Jahr bzw. vier Monate reduziert. Das Ergebnis der Simulation mit der CO₂-Ausstoßoptimierung ist in Bild 4-9 der bedarfsorientierten Regelung gegenübergestellt. Äquivalent sind die Ergebnisse für die Stromkosten in Bild 4-10 dargestellt. Die Vorhersagefehler der NNs bei unterschiedliche langen Trainingsdatensätzen sind in Tabelle 4-4 aufgelistet. Die Werte sind als Wurzel des mittleren quadratischen Fehlers (engl. root mean squared error) (RMSE) in W/m² angegeben, um die Anschaulichkeit zu erhöhen. Es muss beachtet werden, dass die Werte nicht mit den Ergebnissen in den vorherigen Schaubildern verglichen werden können, weil in diesen der Fehler einer Vorhersage zu einem Zeitschritt ohne Autoregression angegeben ist, die Vorhersagefehler der Simulationen in Tabelle 4-4 aber den Fehler mit Autoregression (mit Aufsummierung der Fehler von 24 Vorhersagen) angeben.

Tabelle 4-4: Vorhersagefehler bei verschiedenen Trainingsdatensatzlängen während der Simulation

Länge Trainingsdaten	RMSE in W/m ²
4 Jahre	1,12
1 Jahr	2,26
Monate Sep. bis Dez.	1,10

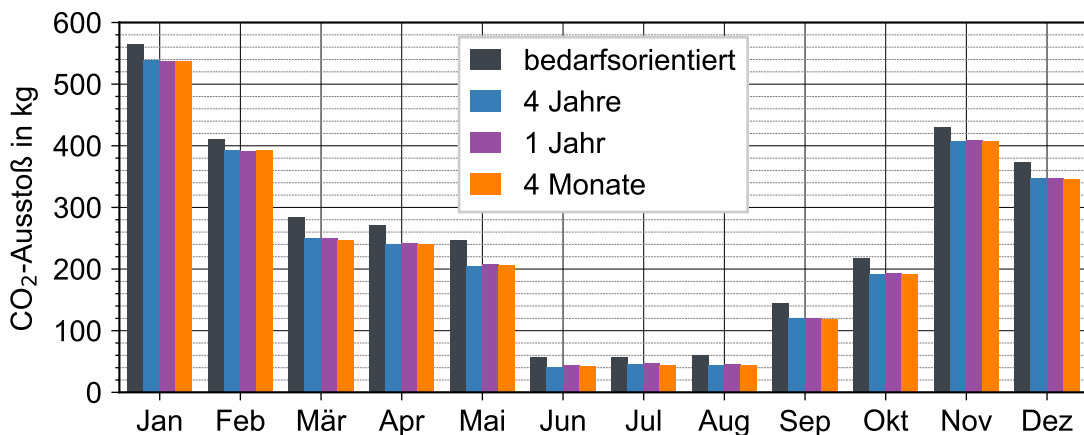


Bild 4-9: Auswirkung der Trainingszeit auf das Simulationsergebnis bei der Optimierung hin zu geringem CO₂-Ausstoß

Bild 4-9 zeigt, dass die Trainingsdatensatzlänge keinen wesentlichen Einfluss auf die Simulationsergebnisse hat, obwohl die Vorhersagequalität sich verändert hat. So verdoppelt sich der Fehler

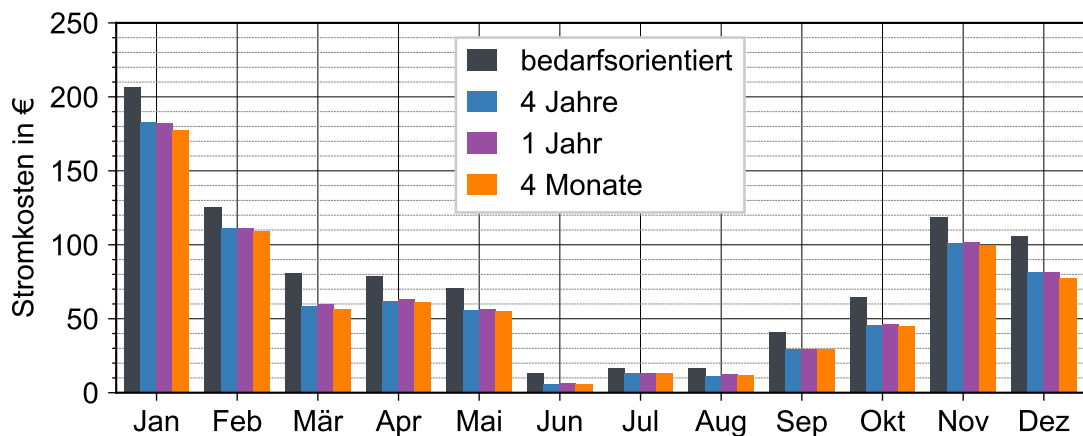


Bild 4-10: Auswirkung der Trainingszeit auf das Simulationsergebnis bei der Optimierung hin zu geringen Stromkosten

der Vorhersage bei der Reduktion des Datensatzes von vier auf ein Jahr. Die zugehörigen Simulationsergebnisse verändern sich ungeachtet dessen marginal von insgesamt 2816 kg ausgestoßenem CO₂ pro Jahr bei vier Jahren auf 12 kg CO₂ mehr bei einer Trainingsdatensatzlänge von einem Jahr, was einer Veränderung von 0,4 % entspricht. Die Reduktion der Trainingsdatensatzlänge auf vier Monate resultiert in einer verbesserten Vorhersagequalität über ein Jahr und verändern das Simulationsergebnis nicht (0,2 kg CO₂ mehr Ausstoß innerhalb eines Jahres). Dass mit einem kleineren Datensatz ein besseres Ergebnis erzielt werden kann als mit einem großen ist untypisch für NNs. Vermutlich liegt der Grund hierfür an den gewählten Monaten, die zum Training verwendet werden und repräsentativer bzw. einflussreicher für die Vorhersage eines Jahres sind.

Bei der Optimierung auf geringere Stromkosten zeigt sich ein ähnliches Bild. Auch hier spielt die Vorhersagequalität während der MIL-Simulation nur eine marginale Rolle. Die Verbesserung der Vorhersage durch die vier Monate bewirkt eine deutlichere Verbesserung des Ergebnisses im Vergleich zur Optimierung des CO₂-Ausstoßes. So können die Stromkosten um 2,1 % reduziert werden, was für das exemplarische Gebäude 16 €/Jahr entspricht.

Die Ergebnisse zeigen, dass die Vorhersagequalität einen geringen Einfluss auf die Simulationsergebnisse hat. Das bedeutet, dass der Optimierer das wichtigste Glied im Applikationsmodell ist. Das birgt den Vorteil, dass in der Praxis weniger Daten als ursprünglich erwartet benötigt werden, um das NN zu trainieren.

4.3.5 Realdaten-Simulation

Die Ergebnisse der MIL-Simulation mit den Realdaten werden in Bild 4-11 und Bild 4-12 für die CO₂- und Stromkostenoptimierung abgebildet. Als „nt“ betitelte Werte nutzen zusätzlich zu den in Kapitel 4.2 genannten Eingabeparametern den manuellen Wert als Repräsentation der manuellen Steuerung. Die verwendeten NNs besitzen eine Vorhersagequalität während der Simulation wie in Tabelle 4-5 angegeben.

Der Vergleich der Vorhersagequalität mit den Werten in Tabelle 4-4 ist nur bedingt möglich. Es muss aber bedacht werden, dass die Heizlast des Realgebäudes Baujahr 1996 höher ist als die des Beispielgebäudes, das thermisch an die Energieeinsparverordnung (EnEV) 2014 angelehnt ausgelegt ist. Dadurch resultiert ein gleicher relativer Vorhersagefehler beim Realgebäude in einem höheren Absolutfehler. Zusätzlich werden in der Simulation unterschiedliche Zeiträume betrachtet. Der erwartete Fehler dürfte bei maximal 25 % Veränderung des Absolutwerts liegen.

Tabelle 4-5: Vorhersagefehler der mit den Realdaten trainierten neuronalen Netze

verwendetes Trainingsdaten	RMSE in W/m ²	Eingangsparameter
Realdaten	3,9	wie Simulationsdaten
Realdaten verbessert	3,4	zusätzlich mit manuellem Steuerungswert

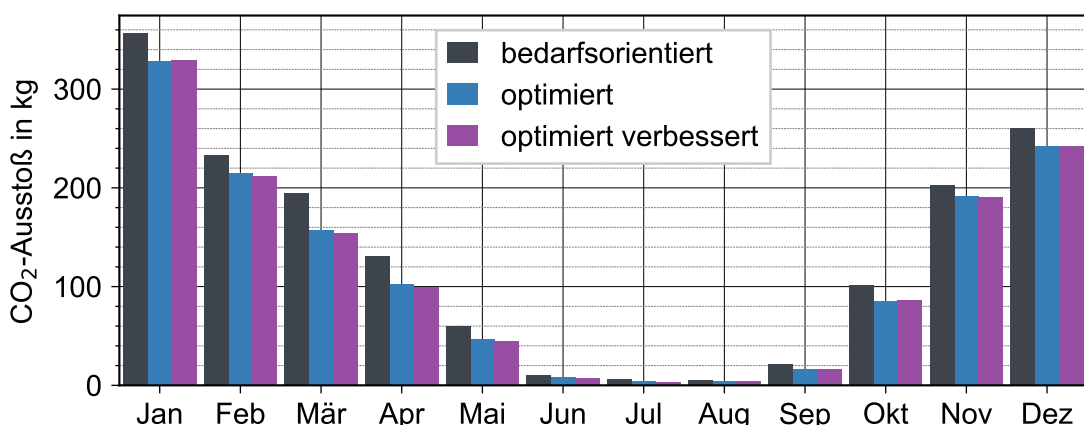


Bild 4-11: CO₂-Ausstoß bei der Optimierung hinsichtlich hohen Strommix-Anteils an Wind- und Solarkraft mit verwendeten Realdaten. Bedarfsorientiert ergeben sich 8,8 kg CO₂/(m²·a); optimiert 7,8 kg CO₂/(m²·a); optimiert verbessert 7,7 kg CO₂/(m²·a)

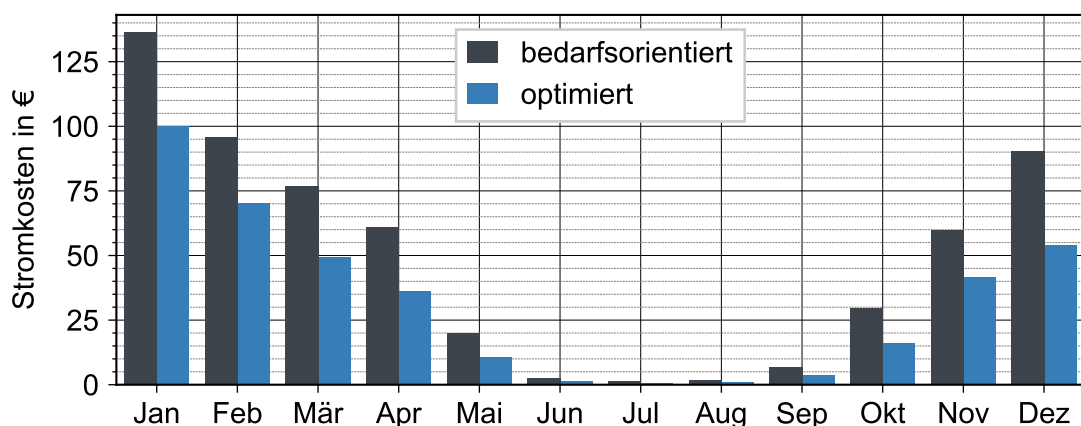


Bild 4-12: Stromkosten bei der Optimierung hinsichtlich geringen Börsenstrompreises mit verwendeten Realdaten. Bedarfsorientiert ergeben sich 3,2 €/m²·a; optimiert 2,1 €/m²·a

Somit lässt sich sagen, dass die Vorhersagequalität durch die Verwendung von Realdaten um ein Vielfaches verschlechtert wird. Gleichzeitig verbessert ein manueller Wert, der die Steuerung der Nachtabsenkung repräsentiert, den Fehler während der Vorhersage von 24 Stunden von einem RMSE von 3,9 W/m² auf 3,4 W/m² (vgl. Bild 4-11). Die Ergebnisse aus Kapitel 4.3.4 zeigen, dass für das Ergebnis in der Simulation der Einfluss der Optimierung überwiegt, weshalb eine schlechtere Vorhersage durch den Optimierer kompensiert werden kann. Dies wird mit den MIL-Simulationen mit den Realdaten bestätigt. So zeigt Bild 4-11, dass das Potential der CO₂-Einsparung in den Realdaten auf einem ähnlichen relativen Niveau liegt wie bei den Simulationsdaten. So wird im betrachteten Zeitraum durchschnittlich 12 % CO₂ eingespart, wobei das verbesserte NN die Einsparung um 0,7 %-Punkte erhöhen kann. Der geringe Einfluss der Vorher-

sagegenauigkeit wird damit bestätigt und durch den geringen Einfluss der optimierten Vorhersage bekräftigt. Im Vergleich zu den Simulationsdaten steigert sich die Einsparung bei den Realdaten, wobei dies nicht signifikant ist. Das ist durch die veränderten Simulationsumgebungen erklärbar. Allgemein zeigt sich durch den Vergleich der Ergebnisse mit Simulations- und Messwerten, dass die Simulationen die Realität ausreichend genau abbilden.

Mit der Optimierung hin zu geringen Stromkosten zeigen die Ergebnisse mit den Realdaten in Bild 4-12 ein gesteigertes Potential der Einsparungsmöglichkeiten. So können durch den Einsatz des Applikationsmodells im Jahr 34 % der Kosten gespart werden. Die Möglichkeit der Einsparung durch Lastumverteilung ist durch das veränderte Simulationsjahr erhöht. Dies wird durch die Standardabweichung in den Strompreisdaten abgebildet^{III}. Dieser Wert beträgt bei den Simulationsdaten 15 €/TWh über das Jahr, wohingegen die Standardabweichung bei den Realdaten einen Betrag von 20 €/TWh aufweisen. Wird diese Größe als repräsentativ für das Einsparpotential gesehen, erhöht sich dieses bei den Realdaten im Vergleich zu den Simulationsdaten um 29,8 %. Die in den MIL-Simulationen auftretenden Stromkosteneinsparungen erhöhen ihren Betrag um 79 % (von 19 % auf 34 %). Infolgedessen kann davon ausgegangen werden, dass die Einsparverbesserung mit den Realdaten nicht allein auf den veränderten Simulationszeitraum zurückzuführen ist. Somit zeigen die MIL-Simulationen mit den Realdaten bei Verwendung einer Nachtabsenkung ein höheres Einsparpotential als die Simulationsdaten und vergrößern das Potential des Applikationsmodells deutlich. Eine Begründung kann der Aufheizaufwand sein, der nach der Nachtabsenkung notwendig ist, um die beheizten Zonen auf Solltemperatur zu bringen (Bild 3-8). Im Zusammenspiel mit der nächtlich nicht vorhandenen Heizlast ergibt sich ein höheres Potential, die Tageslasten in die Nacht zu verschieben. Das hat den Vorteil, dass nachts die Nachfrage nach Strom geringer ist als am Tag und damit auch der Strompreis niedriger ist. Um diesen Umstand zu überprüfen, wird untersucht, wann der Optimierer während der Simulation den Speicher befüllt. Das Ergebnis ist, dass in der Zeit von 22 Uhr bis fünf Uhr durchschnittlich 56 % mehr Energie pro Tag in den Speicher geladen wird, als während der restlichen Zeit. Das bestätigt die Annahme und zeigt, dass durch das Applikationsmodell die Last vom Tag in die Nacht effektiver umverteilt wird.

Beide Optimierungsmethoden werden nur geringfügig von der verbesserten Vorhersage beeinflusst. Dies stimmt mit den Ergebnissen in Kapitel 4.3.4 überein.

^{III}Je größer die Standardabweichung, desto ausgeprägtere Fluktuation in den Daten und desto größer die Möglichkeiten, durch Lastumverteilung die Kosten zu senken.

5 Zusammenfassung

In der vorgelegten Studie wird ein Applikationsmodell zur Lastumverteilung mittels Energiespeicher entwickelt und durch Model-in-the-Loop-(MIL-)Simulationen auf seine Funktionsfähigkeit untersucht. Dabei wird ein neuronales Netz (NN) zur Modellprädiktion, Wetterdaten vom Deutschen Wetterdienst (DWD) und Strommarktsignale vom Verband Europäischer Übertragungsnetzbetreiber (engl. European Network of Transmission System Operators for Electricity) (ENTSO-E) verwendet. Ein entwickelter Optimierer erstellt ein Speicher-Beladungsplan für den Zeitraum der Vorhersage. Getestet wurde eine Heizlastvorhersage von 24 Stunden mit den Marktsignalen Strompreis und Anteil erneuerbarer Energien an der Stromerzeugung. Dafür wurde jeweils die Day-Ahead-Vorhersage verwendet. Als Testzeitraum wurden Simulationsdaten für das Jahr 2019 ausgewählt; die Lastumverteilung wird simulativ mit einem Warmwasserspeicher umgesetzt. Es werden zwei MIL-Simulationen durchgeführt, wobei die Heizlastverläufe einmal simulativ mit TRNSYS berechnet und einmal aus Messdaten stammt. Das Beispielgebäude hat eine beheizte Nutzfläche von 197 m² und eine Normheizlast von ca. 6,8 kW (ca. 35 W/m²).

Das NN verwendet für die Vorhersage die folgenden Eingabeparameter: die flächenbezogene Heizlast, die Außenlufttemperatur, die Globalstrahlung, den Elevationswinkel und die Windgeschwindigkeit in Nord- und Ost-Richtung. Entgegen den Erwartungen ist die gleichmäßige Aufteilung von Trainings-, Validierungs- und Testdaten über die Zeit nachteilig gegenüber der chronologischen Reihenfolge, obwohl die erste Methode das NN diverser trainiert. Ein Test mit unterschiedlichen Trainingsdatenzeiträumen lässt vermuten, dass dies an dem höheren Einfluss von Daten im Winter mit hohem Potenzial zur Lastumverteilung liegt. Wird die Vorhersagegenauigkeit im Winter besser, aber verschlechtert sie sich im Sommer, so wird integriert eine höhere Einsparung der Kosten erzielt, da im Winter die Absolutbeträge um ein Vielfaches höher sind und sich der Vorhersagefehler relativ bemerkbar macht. So ist ein Fokus des Trainings auf Wintermonate vermutlich vorteilhaft und sollte weiter untersucht werden.

Die MIL-Simulationen zeigen für das Beispieljahr 2019 ein CO₂-Einsparpotential von 9,5 % auf. Im selben Zeitraum lassen sich 19 % der Börsenstromkosten einsparen. Durch die Verluste des Warmwasserspeichers steigt der Gesamtenergieverbrauch im Vergleich zur bedarfsoptimierten Regelung an. Die Optimierung und Vorhersage in den Sommermonaten mit wenig bis keiner Heizlast ist ineffizient, weil die Vorhersage keinen Zustand „aus“ kennt. Um diesen Umstand in Zukunft zu umgehen, sollte für das Training auch die Kühllast als negative Heizlast dargestellt werden. Vorhersagewerte unter Null können vom Optimierer dann als Null aufgefasst werden und das Problem lösen. Für die Ergebnisse dieser Studie werden durch die geringe Genauigkeit im Sommer aber wenig beeinflusst, weil die Absolutwerte gering sind.

Ein weiteres Ergebnis der Studie zeigt die Korrelation zwischen der CO₂-Emission und dem Strompreis auf. So werden bei einer Optimierung der CO₂-Emissionen bei der Stromerzeugung gleichzeitig 11 % an Stromkosten eingespart. Umgekehrt wird die CO₂-Emission durch die Stromerzeugung um 5,1 % verringert, wenn bezüglich geringer Stromkosten optimiert wird.

Eine Verwendung von Messdaten des Heizlastverlaufs eines Einfamilienhauses Baujahr 1995 mit ca. 165 m² beheizter Wohnfläche und aktiver Nachtabsenkung bestätigt die vorangegangenen Simulationen und zeigt ein größeres Potential der Lastumverteilung auf: 34 % der Stromkosten können eingespart werden. Das entspricht einer Steigerung von fast 80 % im Vergleich zur Simulation

ohne Nachtabsenkung. Untersuchungen zeigen, dass die Nachtabsenkung und die morgendliche Aufheizphase die Lastumverteilung begünstigen. Sie steigern das entsprechende Potential durch hohe Heizlastfluktuation. So werden ein Großteil der Lasten von morgens und tagsüber auf die Nacht umverteilt.

Das in der vorgelegten Studie entwickelte Applikationsmodell ist funktionstüchtig und unterschiedliche Testreihen zeigten, dass hiermit entsprechende Potentiale gehoben werden könnten. Für den großflächigen Einsatz sollten verschiedene Sicherheitsaspekte bedacht und der Optimierer verbessert werden, weil die Vorhersagegenauigkeit nur einen eingeschränkten Einfluss auf das Ergebnis besitzt.

Für weitergehende Untersuchungen und zur Demonstration implementiert die Tecomon GmbH, die beratend im Projekt unterstützte, derzeit das entwickelte Applikationsmodell auf einem Raspberry Pi, um es zu Versuchszwecken in einem Gebäude mit Standort Esslingen (Neckar) einzusetzen. Dafür müssen Schnittstellen angepasst und verwendete Python-Packages vorkompiliert werden, um sie auf dem Raspberry Pi lauffähig zu machen. Das zu regelnde Gebäude wird von einer Wärmepumpe beheizt und verwendet zusätzlich eine Solarthermie- und Photovoltaikanlage, wodurch viele prinzipiell mögliche Regelstrategien und Kostenkurven untersucht werden könnten.

6 Literatur

Bücher, Monographien, selbständig erschienene bibliographische Einheiten

- [4] Böckh, P.: *Wärmeübertragung: Grundlagen und Praxis*. Heidelberg: Springer, 2011. ISBN: 9783642159589
- [5] Baehr, H. D. und Stephan, K.: *Wärme- und Stoffübertragung*. Springer Berlin Heidelberg, 30. Apr. 2019. 836 S. ISBN: 3662584409
- [6] Stephan, P. et al.: *VDI-Wärmeatlas*. Hrsg. von Chemieingenieurwesen, F. T. V.-G. V. und. Berlin, Germany: Springer Vieweg, 2019. ISBN: 9783662529881
- [8] Herwig, H. und Moschallski, A.: *Wärmeübertragung*. Springer Fachmedien Wiesbaden, 29. Mai 2019, S. 145–151. 388 S. ISBN: 3658264004. URL: https://www.ebook.de/de/product/36448912/heinz_herwig_andreas_moschallski_waermeuebertragung.html
- [10] Möller, F.: *Einführung in die Meteorologie*. Mannheim: Bibliograph. Inst, 1973. ISBN: 3411002883
- [13] Papageorgiou, M., Leibold, M. und Buss, M.: *Optimierung*. Springer Berlin Heidelberg, 13. Nov. 2015. 560 S. ISBN: 3662469359. URL: https://www.ebook.de/de/product/24197805/markos_papageorgiou_marion_leibold_martin_buss_optimierung.html
- [14] Weicker, K.: *Evolutionäre Algorithmen*. Vieweg+Teubner Verlag, 25. Juni 2015. ISBN: 3658099577. URL: https://www.ebook.de/de/product/24198849/karsten_weicker_evolutionaere_algorithmen.html
- [15] Grimme, C. und Bossek, J.: *Einführung in die Optimierung*. Springer Fachmedien Wiesbaden, 2018. ISBN: 978-3-658-21150-9. DOI: 10.1007/978-3-658-21151-6
- [16] Völz, A.: *Modellprädiktive Regelung nichtlinearer Systeme mit Unsicherheiten*. Gabler, Betriebswirt.-Vlg, 1. Nov. 2016. 55 S. ISBN: 3658162783. URL: https://www.ebook.de/de/product/27925522/andreas_voelz_modellpraediktive_regelung_nichtlinearer_systeme_mit_unsicherheiten.html
- [17] Ertel, W.: *Grundkurs künstliche Intelligenz eine praxisorientierte Einführung*. Wiesbaden: Springer Vieweg, 2016. ISBN: 9783658135485

- [18] Goodfellow, I., Bengio, Y. und Courville, A.: *Deep Learning. Das umfassende Handbuch*. MITP Verlags GmbH, 1. Nov. 2018. ISBN: 3958457002. URL: https://www.ebook.de/de/product/31366940/ian_goodfellow_yoshua_bengio_aaron_courville_deep_learning_das_umfassende_handbuch.html
- [20] Raschka, S.: *Machine Learning mit Python und Scikit-learn und TensorFlow das umfassende Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics*. Frechen: mitp, 2018. ISBN: 9783958457331
- [25] Bas, E.: *Einführung in Wahrscheinlichkeitsrechnung, Statistik und Stochastische Prozesse*. Springer-Verlag GmbH, 24. Dez. 2020. ISBN: 3658321199. DOI: <https://doi.org/10.1007/978-3-658-32120-8>. URL: https://www.ebook.de/de/product/39704660/esra_bas_einfuehrung_in_wahrscheinlichkeitsrechnung_statistik_und_stochastische_prozesse.html
- [26] Stieglitz, R. und Heinzl, V.: *Thermische Solarenergie*. Springer Berlin Heidelberg, 21. März 2013. 720 S. ISBN: 364229474X. DOI: <https://doi.org/10.1007/978-3-642-29475-4>. URL: https://www.ebook.de/de/product/19453724/robert_stieglitz_volker_heinzel_thermische_solarenergie.html

Unselbständig erschienene bibliographische Einheiten, Zeitschriften, Artikel

- [9] Kirchhoff, G. und Bunsen, R.: „Chemische Analyse durch Spectralbeobachtungen“. In: 189.7 (1861), S. 337–381. DOI: 10.1002/andp.18611890702
- [11] Angstrom, A.: „Solar and terrestrial radiation. Report to the international commission for solar research on actinometric investigations of solar and atmospheric radiation“. In: 50.210 (Apr. 1924), S. 121–126. DOI: 10.1002/qj.49705021008
- [19] Kalarik, B. und Olgac, A. V.: „Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks“. In: *International Journal of Artificial Intelligence and Expert Systems* (2011), S. 111–122
- [21] Kingma, D. P. und Ba, J.: „Adam: A Method for Stochastic Optimization“. In: (22. Dez. 2014). arXiv: 1412.6980 [cs.LG]
- [24] Salimans, T. und Kingma, D. P.: „Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks“. In: *Advances in Neural Information Processing Systems*. Hrsg. von Lee, D. et al. Bd. 29. Curran Associates, Inc., 2016. URL: <https://proceedings.neurips.cc/paper/2016/file/ed265bc903a5a097f61d3ec064d96d2e-Paper.pdf>
- [29] Bell, I. H. et al.: „Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp“. In: *Industrial*

& *Engineering Chemistry Research* 53.6 (2014), S. 2498–2508. DOI: 10.1021/ie4033999. eprint: <http://pubs.acs.org/doi/pdf/10.1021/ie4033999>. URL: <http://pubs.acs.org/doi/abs/10.1021/ie4033999>

Gesetze, Verordnungen, Richtlinien, Normen

- [3] *Messstellenbetriebsgesetz (MsbG)*. 2016
- [7] DIN EN 12977-3:2018: *Thermische Solaranlagen und ihre Bauteile – Kundenspezifisch gefertigte Anlagen – Teil 3: Leistungsprüfung von Warmwasserspeichern für Solaranlagen*. Hrsg. von Europäisches Komitee für Normung. Techn. Ber. Juli 2018
- [28] DIN EN 16798-1:2019: *Energetische Bewertung von Gebäuden – Lüftung von Gebäuden – Teil 1: Eingangsparameter für das Innenraumklima zur Auslegung und Bewertung der Energieeffizienz von Gebäuden bezüglich Raumluftqualität, Temperatur, Licht und Akustik – Modul M1-6*. Hrsg. von Europäisches Komitee für Normung. Techn. Ber. Apr. 2021

Unveröffentlichte Arbeiten wie Master- und Diplomarbeiten

- [1] Kimmich, J.: „Implementierung eines Vorhersage- und Optimierungsalgorithmus in eine Datenschnittstelle für Smart Meter Gateways“. Masterarbeit. Institut für Gebäudeenergetik, Thermotechnik und Energiespeicherung Stuttgart (IGTE), Jan. 2022
- [22] Westhauser, F.: „Neural Architecture Search zur Modellbildung von Gebäudeenergiesystemen“. Masterarbeit. Institut für Gebäudeenergetik, Thermotechnik und Energiespeicherung Stuttgart (IGTE), Mai 2020
- [27] Kienzlen, Y.: „Studie zum Thema Büro- und Wohngebäudebestand in Deutschland“. Bachelorarbeit. Institut für Gebäudeenergetik, Thermotechnik und Energiespeicherung Stuttgart (IGTE), 1. März 2018

Literaturangabe aus elektronischen Medien wie dem Internet

- [2] Kaiser, U.: *Energiewende im Überblick*. Hrsg. von Informationsamt der Bundesregierung, P. und. 12. Okt. 2021. URL: <https://www.bundesregierung.de/bregde/themen/energiewende/energiewende-im-ueberblick-229564> (besucht am 12.10.2021)
- [12] Seyfert, C.: *Smart Metering Systems. Intelligente Messsysteme*. Hrsg. von Sicherheit in der Informationstechnik, B. für. 11. Nov. 2021. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Smart-metering/smart-metering_node.html (besucht am 11.11.2021)

- [23] Stöttner, T.: *Why Data should be Normalized before Training a Neural Network*. Hrsg. von Inc., T. D. S. 11. Nov. 2021. URL: <https://towardsdatascience.com/why-data-should-be-normalized-before-training-a-neural-network-c626b7f66c7d> (besucht am 11.11.2021)
- [30] Bouras, A.: *Forschung und Entwicklung für die Wettervorhersage*. Hrsg. von Wetterdienst, D. 11. Nov. 2021. URL: https://www.dwd.de/DE/forschung/wettervorhersage/wettervorhersage_node.html (besucht am 11.11.2021)
- [31] Ittershagen, M.: *Treibhausgas-Emissionen*. Hrsg. von Bundesministerium für Umwelt, N. u. n. S. 24. Nov. 2021. URL: <https://www.umweltbundesamt.de/themen/klima-energie/treibhausgas-emissionen> (besucht am 24.11.2021)

Anhang

A Verwendete Software und deren Versionen

Tabelle A-1: Verwendete Software

Name	Version
PyCharm	2021.2.2 (Community Edition)
Python	3.8.8

Tabelle A-2: Installierte Python-Packages mit Versionsnummern Teil 1

Package	Versionsnummer	Package	Versionsnummer
absl-py	0.13.0	idna	2.10
aenum	3.1.0	importlib-resources	5.2.2
appdirs	1.4.4	ipython	7.26.0
astunparse	1.6.3	ipython-genutils	0.2.0
backcall	0.2.0	jedi	0.18.0
beautifulsoup4	4.9.3	keras-nightly	2.5.0.dev2021032900
cached-property	1.5.2	Keras-Preprocessing	1.1.2
cachetools	4.2.2	kiwisolver	1.3.1
certifi	2021.5.30	lxml	4.6.3
chardet	4.0.0	Markdown	3.3.4
click	7.1.2	matplotlib	3.4.3
click-params	0.1.2	matplotlib-inline	0.1.2
cloup	0.8.2	measurement	3.2.0
colorama	0.4.4	mpmath	1.2.1
CoolProp	6.4.1	munch	2.5.0
cycler	0.10.0	numpy	1.19.5
dateparser	1.0.0	oauthlib	3.1.1
decorator	5.0.9	opt-einsum	3.3.0
deprecation	2.1.0	packaging	21.0
dill	0.3.4	pandas	1.3.3
docopt	0.6.2	parso	0.8.2
dogpile.cache	1.1.2	pbr	5.6.0
entsoe-py	0.4.1	pickleshare	0.7.5
fasthaversine	0.1.2	Pillow	8.3.1
flatbuffers	2.0	Pint	0.17
fusepy	3.0.1	plotly	5.1.0
gast	0.5.2	pooch	1.4.0
google-auth	1.34.0	prompt-toolkit	3.0.19
google-auth-oauthlib	0.4.3	protobuf	3.17.3
google-pasta	0.2.0	pyasn1	0.4.8
grpcio	1.39.0	pyasn1-modules	0.2.8
h5netcdf	0.10.0	Pygments	2.9.0
h5py	3.3.0	pykml	0.2.0
haversine	2.3.1	pyparsing	2.4.7
html5lib	1.1	PyPDF2	1.26.0
htmllistparse	0.5.2	pyproj	3.1.0

Tabelle A-3: Installierte Python-Packages mit Versionsnummern Teil 2

Package	Versionsnummer	Package	Versionsnummer
pysolar	0.9	tensorboard-data-server	0.6.1
python-dateutil	2.8.2	tensorboard-plugin-wit	1.8.0
pytz	2021.1	tensorflow	2.5.0
rapidfuzz	1.4.1	tensorflow-addons	0.14.0
regex	2021.8.3	tensorflow-estimator	2.5.0
requests	2.25.1	termcolor	1.1.0
requests-cache	0.5.2	tqdm	4.62.0
requests-ftp	0.3.1	traitlets	5.0.5
requests-oauthlib	1.3.0	typing-extensions	3.10.0.0
retrying	1.3.3	tzlocal	2.1
rsa	4.7.2	urllib3	1.26.6
scipy	1.7.1	utm	0.7.0
seaborn	0.11.1	validators	0.18.2
six	1.16.0	wcwidth	0.2.5
soupsieve	2.2.1	webencodings	0.5.1
stevedore	3.3.0	Werkzeug	2.0.1
sympy	1.8	wetterdienst	0.17.0
tabulate	0.8.9	wrapt	1.12.1
tenacity	8.0.1	xarray	0.19.0
tensorboard	2.5.0	zipp	3.5.0

B Randbedingungen für die Gebäudesimulation

Tabelle B-1: Thermische Eigenschaften des Beispielgebäudes nach [27]

Bauteil	U-Wert in W/m ² /K
Kellerwand	0,35
Kellerdecke	0,35
Außenwand	0,24
Bodenplatte	0,3
Decke	0,2
Dach	0,2
Fenster	0,95

C Eingabeparameterzuordnung

Tabelle C-1: Eingangsparameterzuordnung zu den Modellen B01 bis B32 und den Basismodellen X

Modellbezeichnung	Eingabeparameter										
	flächenbez. Heizlast	Bedeckungsgrad	Außenlufttemperatur	Globalstrahlung	Taupunkt	Luftdruck	positive Elevation	Windgeschw. Nord	Windgeschw. Ost	Wochenende Elevation	cos(Azimuth)
B01	✓										
B02	✓	✓									
B03	✓		✓								
B04	✓			✓							
B05	✓				✓						
B06	✓					✓					
X07	✓						✓				
B08	✓							✓			
B09	✓								✓		
B10	✓						✓	✓			
B11	✓									✓	
B12	✓										✓
B13	✓										✓
B14	✓	✓					✓				
B15	✓		✓				✓				
B16	✓			✓			✓				
B17	✓				✓		✓				
B18	✓					✓	✓				
B19	✓						✓	✓			
B20	✓						✓		✓		
X21	✓						✓	✓	✓		
B22	✓						✓			✓	
B23	✓						✓				✓
B24	✓						✓				✓
B25	✓	✓					✓	✓	✓		
B26	✓		✓				✓	✓	✓		
X27	✓			✓			✓	✓	✓		
B28	✓				✓		✓	✓	✓		
B29	✓					✓	✓	✓			
B30	✓						✓	✓	✓	✓	
B31	✓						✓	✓	✓		✓
B32	✓						✓	✓	✓		✓

Tabelle C-2: Eingangsparameterzuordnung zu den Modellen B33 bis B50 und den Basismodellen X

Modellbezeichnung	Eingabeparameter										
	flächenbez. Heizlast	Bedeckungsgrad	Außenlufttemperatur	Globalstrahlung	Taupunkt	Luftdruck	positive Elevation	Windgeschw. Nord	Windgeschw. Ost	Wochenende	Elevation cos(Azimuth)
B33	✓	✓		✓			✓	✓	✓		
X34	✓		✓	✓			✓	✓	✓		
B35	✓			✓	✓		✓	✓	✓		
B36	✓			✓		✓	✓	✓	✓		
B37	✓			✓			✓	✓	✓	✓	
B38	✓			✓			✓	✓	✓		✓
B39	✓			✓			✓	✓	✓		✓
B40	✓	✓	✓	✓			✓	✓	✓		
B41	✓		✓	✓	✓		✓	✓	✓		
B42	✓		✓	✓		✓	✓	✓	✓		
X43	✓		✓	✓			✓	✓	✓	✓	
B44	✓		✓	✓			✓	✓	✓		✓
B45	✓		✓	✓			✓	✓	✓		✓
B46	✓	✓	✓	✓			✓	✓	✓	✓	
B47	✓		✓	✓	✓		✓	✓	✓	✓	
B48	✓		✓	✓		✓	✓	✓	✓	✓	
B49	✓		✓	✓			✓	✓	✓	✓	✓
B50	✓		✓	✓			✓	✓	✓	✓	✓