

Universität Stuttgart

IER Institut für Energiewirtschaft
und Rationelle Energieanwendung

Forschungsarbeit

Vergleichende Analyse von PV- Prognosemodellen und Implementierung eines Echtzeit- Prognosedienstes für die dynamische Prognose der lokalen PV-Leistung

Haoyan
Li

Bandnummer 1015

Vergleichende Analyse von PV-Prognosemodellen und Implementierung eines Echtzeit-Prognosedienstes für die dynamische Prognose der lokalen PV-Leistung

Forschungsarbeit

Haoyan Li
Siriusweg 18b
70565 Stuttgart
Matr.-Nr. 3237579

Prüfer: Prof. Dr.-Ing. K. Hufendiek, IER (Erstprüfer)
Prof. Dr.-Ing. P. Radgen, IER (Zweitprüfer)

Betreuerin: Dr.-Ing. J. Kumm, IER

Studiengang: Nachhaltige Elektrische Energieversorgung

Beginn der Arbeit: 09. Januar 2019

Ende der Arbeit: 04. Juni 2019

Institut für Energiewirtschaft und Rationelle Energieanwendung
Heißbrühlstr. 49a
70565 Stuttgart
Prof. Dr.-Ing. K. Hufendiek
Abteilung Energiemärkte und Intelligente Systeme (EI)

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung.....	2
2	Grundlagen	3
2.1	Clear-Sky-Modell.....	3
2.2	Autoregressives Modell mit exogenem Eingang (ARX-Modell)	4
2.2.1	Mathematische Beschreibung des ARX-Modells	4
2.2.2	Parameterberechnung des ARX-Modells	5
2.3	Neuronales Netz	5
2.3.1	Struktur und Arbeitsweise des neuronalen Netzes	6
2.3.2	Training des neuronalen Netzes	7
2.4	Random Forest.....	7
2.4.1	Grundlagen des Random Forests.....	8
2.4.2	Grundlagen des Quantile Regression Forests	9
2.5	Quantilsregression	10
2.6	Fehlermaße.....	10
3	Konzeption der Prognosemodelle	13
3.1	Konzeption des Hybridmodells.....	13
3.2	Konzeption der Regressionsmodelle	14
3.2.1	Konzeption des ARX-Modells	14
3.2.2	Konzeption des neuronalen Netzes	14
3.2.3	Konzeption der Kombination eines ARX-Modells mit neuronalem Netz.....	15
3.2.4	Konzeption des Random Forests	16
3.3	Konzeption des Clear-Sky-Modells.....	16
4	Implementierung und Parametrierung der Prognosemodelle	19
4.1	Struktur des Programms	19
4.2	Dynamisches Abrufen von Wetterprognose und PV-Leistungsdaten und Datenaufbereitung	20
4.3	Ablauf der Prognoseerstellung	20
4.4	Parametrierung der Prognosemodelle	22
4.4.1	Parametrierung des ARX-Modells	22
4.4.2	Parametrierung des neuronalen Netzes	26
4.4.3	Parametrierung der Kombination eines ARX-Modells mit einem neuronalen Netz	28
4.4.4	Parametrierung des Random Forests.....	29
4.4.5	Parametrierung des Clear-Sky-Modells	31
4.4.6	Kritische Auseinandersetzung.....	32
5	Ergebnisse und Vergleich der Prognosemodelle	33

5.1	Rahmenbedingungen	33
5.2	Deterministische Prognosen	34
5.2.1	Einfluss der Fehlermaße auf Prognoseergebnisse	36
5.2.2	Einfluss der Trainingsdatenmenge auf die Prognosequalität	38
5.3	Probabilistische Prognoseergebnisse	39
6	Zusammenfassung und Ausblick	43
6.1	Zusammenfassung	43
6.2	Kritik und Ausblick	44
Literaturverzeichnis		47
Anhang		49
A	Erfassung der Wetterprognosen	49
B	Erfassung der Leistungsdaten	50
Eigenständigkeitserklärung		53

Abbildungsverzeichnis

Abb. 1-1: Darstellung einer intelligenten Steuerung von dezentralen Energiesystemen	1
Abb. 2-1: Viertelstündlich aufgelöste PV-Leistung und entsprechende CS-Leistung	4
Abb. 2-2: Beispiel für den Aufbau eines neuronalen Netzes, nach (Kruse et al. 2015)	6
Abb. 2-3: Beispiel für den Aufbau eines künstlichen Neurons, nach (Kruse et al. 2015)	7
Abb. 2-4: Beispiel für den Aufbau eines Random Forests, nach (Breiman, 2001)	8
Abb. 3-1: Aufbau des Hybridmodells, nach (Decker, 2018)	13
Abb. 3-2: Aufbau des verwendeten ARX-Modells, nach (Bacher et al. 2009)	14
Abb. 3-3: Aufbau des verwendeten neuronalen Netzes, nach (Grossberg 1988)	15
Abb. 3-4: Aufbau der Kombination eines ARX-Modells mit neuronalen Netze	16
Abb. 3-5: Aufbau des verwendeten Random Forests, nach (Breiman, 2001)	16
Abb. 4-1: Ablaufplan des Hauptprogramms	19
Abb. 4-2: Übersicht der Teilprogramme	20
Abb. 4-3: Programmablaufplan mit TP3	21
Abb. 4-4: Autokorrelationsfunktion des CS-Faktors	23
Abb. 4-5: Scatterplot – PV-Leistung gegen Wetterprognosen	24
Abb. 4-6: Korrelationskoeffizienten zwischen der PV-Leistung und Wetterprognosen	25
Abb. 4-7: NRMSE des ARX-T-Modells und des ARX-B-Modells	26
Abb. 4-8: NRMSE der neuronalen Netze mit unterschiedlichen Layers	27
Abb. 4-9: NRMSE der neuronalen Netze mit unterschiedlichen Trainingsdatenmengen ..	28
Abb. 4-10: NRMSE der ARX-NN-Modelle mit unterschiedlichen Trainingsdatenmengen	29
Abb. 4-11: NRMSE der Random Forests mit unterschiedlicher Tree-Anzahl	30
Abb. 4-12: NRMSE der Random Forests mit unterschiedlicher Blattanzahl	30
Abb. 4-13: NRMSE der Random Forests mit unterschiedlichen Trainingsdatenmengen ..	31
Abb. 4-14: NRMSE der neuronalen Netze mit unterschiedlichen CS-Leistungen	32
Abb. 5-1: Messung und Punktprognosen der PV-Leistung	35
Abb. 5-2: NRMSE der verschiedenen Prognosemodelle	36
Abb. 5-3: Improvement der Prognosemodelle bezogen auf ARX-T-Modell mit Fehlermaßen NRMSE, MAE, MAPE	37

Abb. 5-4: NRMSE der Prognosemodelle mit beschränkten Trainingsdaten (5 Monate) ..	39
Abb. 5-5: Messung und probabilistische Prognosen der PV-Leistung	40
Abb. 5-6: Winkler Score der Prognosemodelle	41
Abb. B-1: Ablaufplan der Datenerfassung.....	51

Tabellenverzeichnis

Tabelle 4-1: Übersicht der in dieser Arbeit untersuchten Regressionsmodelle.....	22
Tabelle 5-1: Datenbeschreibung der verfügbaren PV-Leistung und der Wetterprognose	33
Tabelle 5-2: Übersicht der Prognoseanforderungen.....	33
Tabelle 5-3: Auswahl der Trainingdaten und Testdaten	34

Abkürzungsverzeichnis

ACF	Autokorrelationsfunktion
AR	Autoregressives Modell
ARX	AR-Modell mit exogenem Eingang
ARX-NN	Kombination eines ARX-Modells mit neuronalem Netz
CS	Clear Sky
IER	Institut für Energiewirtschaft und Rationelle Energieanwendung
Imp	Improvement
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NN	Neuronales Netz
NRMSE	Normalized Root Mean Square Error
NWP	Numerical Weather Prediction
PI	Prognoseintervall
PV	Photovoltaik
RF	Random Forest
RMSE	Root Mean Square Error
TP	Teilprogramm
Winkler	Winkler Score

Symbole

a	ARX-Modellparameter
b	ARX-Modellparameter
e	Error
f_{act}	Aktivierungsfunktion
f_{nwp}	Modellierungsfunktion der Wetterprognose
F	Verteilungsfunktion
\hat{F}	Prognostizierte Verteilungsfunktion
h_x	Bandweite der Standardnormalverteilung auf x-Achse
h_y	Bandweite der Standardnormalverteilung auf y-Achse
I	Indikator
J	Zielfunktion
k	Prognosestep
m	ARX-Modellparameter
n	Anzahl der Daten
net	Netzeingabe eines Neurons
P	PV-Leistung
P_{CS}	Clear-Sky-Leistung
q	Wahrscheinlichkeit des Quantilwertes
s	Lag
t	Zeitpunkt
w	Gewicht
X	Eingangsvektor
Y	Zielgröße
\hat{Y}	Prognostizierte Zielgröße
\hat{Y}_q	Quantil der prognostizierten Zielgröße
Z	Exogener Eingang des ARX-Modells
β	Quantilfaktor
τ	Clear-Sky-Faktor
$\hat{\tau}$	Prognostizierter Clear-Sky-Faktor
τ^{nwp}	Durch Wetterprognosen modellierter Clear-Sky-Faktor

$\hat{\mu}$	Punktprognose des RF-Modells
θ	Modellparametervektor
δ	Abweichung des Prognoseintervalls
ρ	Fehlerfunktion

1 Einleitung

1.1 Motivation

Mit der Energiewende geht die Wandlung von der klassischen zentralen Versorgungsstruktur über zu einer Struktur mit einem zunehmenden Anteil an intelligenten dezentralen Energiesystemen einher. Anhand von einem zunehmenden Anteil der elektrischen Einspeisung aus erneuerbaren Energien und entsteht mithilfe eines echtzeitfähigen Datenaustausches die Möglichkeit, intelligente Steuerungen für dezentrale Energiesystemen mit Echtzeitdaten zu entwickeln, welche Flexibilität für das Energiesystem bereitstellen können. Ein Hauptziel des Projektes „Heat4SmartGrid_BW“ des Instituts für Energiewirtschaft und Rationelle Energieanwendung der Universität Stuttgart besteht darin, mehr Flexibilität für das Energiesysteme durch das Berechnen optimaler Fahrpläne von Wärmepumpen bereitzustellen.

Die Einflussgrößen für eine intelligente Steuerung auf Gebäudeebene sind in Abb. 1-1 dargestellt. Sie basiert auf der zu erwartenden Erzeugungsleistung von PV-Anlage, elektrischen und thermischen Last des Gebäudes. Wegen der hohen Volatilität der PV-Einspeisung und der großen Schwankung des Beitrags an der Tageslast sind Speichermöglichkeiten zusätzlich zu berücksichtigen, sowohl durch elektrische Energiespeicher als auch durch Wärmespeicher, um eine Flexibilisierung des Energiesystems zu verwirklichen.

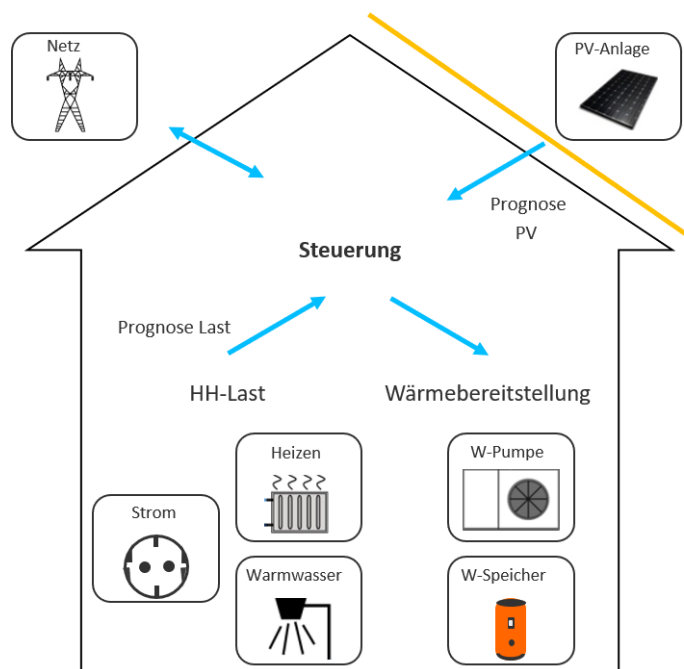


Abb. 1-1: Darstellung einer intelligenten Steuerung von dezentralen Energiesystemen

1.2 Zielsetzung

Als wichtige Inputdaten für eine intelligente Steuerung werden Prognosen von PV-Leistung, elektrischer und thermischer Last benötigt. In dieser Forschungsarbeit werden Methoden für die Prognose der PV-Leistung untersucht.

Das Ziel der Arbeit besteht darin, nach der Konzeptionierung, Implementierung, Optimierung und Evaluierung von Prognosemethoden die Echtzeitdaten von PV-Leistung und Wetterprognose mit dem Prognosemodell zu verknüpfen und PV-Prognosen in Echtzeit zu erstellen. Somit wird ein Prognosedienst für eine PV-Leistungsprognose unter Einbindung von Echtzeitdaten erstellt und an der PV-Anlage auf dem Dach des IER getestet. Der Dienst soll PV-Prognosen mit einer 15-minütigen Auflösung für einen Prognosehorizont von 72 Stunden erstellen.

Die Prognosemethode soll unter Verwendung eines bestehenden Clear-Sky-Modells erstellt werden. Dieses soll mit passenden Prognosemodellen zur Berücksichtigung der Wetterbedingungen ergänzt werden. Im Vergleich zu einer deterministischen Prognose werden über eine probabilistische Prognose mehr Informationen über die Unsicherheit und Wahrscheinlichkeit transportiert. Daher sollen auch stochastische Prognoseverfahren betrachtet werden.

2 Grundlagen

In diesem Kapitel werden die wichtigsten Zusammenhänge zum Verständnis über die Erstellung einer Prognose der PV-Leistung vorgestellt. Zuerst wird der Ansatz des Clear-Sky-Modells erläutert. Anschließend sind Charakteristika und mathematische Zusammenhänge der verwendeten Prognosemethoden detailliert zu untersuchen. Zusätzlich wird die Quantile-Regression-Methode berücksichtigt, um probabilistische Prognose zu erstellen. Anschließend werden Fehlermaße für Modellvergleich und Prognoseauswertung dargelegt.

2.1 Clear-Sky-Modell

Die Clear-Sky-Leistung (CS-Leistung) beschreibt die durch eine PV-Anlage erzeugte Leistung, wenn die Wetterbedingung wolkenlos ist. In einem Clear-Sky-Ansatz lässt sich die PV-Leistung (P) aus einer wolkenlosen Komponente (P_{CS}) und einer wetterabhängigen Komponente (τ) berechnen durch:

$$P = \tau \cdot P_{CS} \quad (2-1)$$

Dann ist P_{CS} die bei wolkenlosen Bedingungen erzeugte Leistung (Clear-Sky-Leistung) für den Prognosezeitpunkt. Um aus der Clear-Sky-Leistung die tatsächliche Leistung zu berechnen, wird der Clear-Sky-Faktor (τ) verwendet, der die Wetterbedingungen, d.h. die Durchlässigkeit der Atmosphäre aufgrund von Bewölkung und anderen Wetterparametern, abbildet. Zur Berechnung der Clear-Sky-Leistung wird in dieser Arbeit eine statistische Methode, die Kernel-Regression (Bacher 2008) verwendet, die bereits implementiert vorliegt (Decker 2018). Bevor das Clear-Sky-Modell anhand der Kernel-Regression mit den Parametern einer spezifischen Anlage aufgebaut werden kann, müssen die Leistungsdaten dieser PV-Anlage für ein ganzes Jahr zur Verfügung stehen. Hierfür werden die vorliegenden Messdaten der PV-Anlage auf dem Dach des IER verwendet. Die Messwerte der am Standort beobachteten PV-Leistung werden gegen die Achsen ‚Stunde am Tag‘ und ‚Tag im Jahr‘ aufgetragen, siehe Abb. 2-1. Nachdem die PV-Messwerte für ein Jahr bekannt sind, wird die CS-Leistung für jede Stunde des Jahres nach der Methode des Smoothing-Kernels (Bacher 2008) über die gewählten Modellparameter der Bandweiten h_x und h_y bestimmt und für die Abb. 2-1 auf der Achse Stunde am Tag und auf der Achse Tag im Jahr aufgetragen.

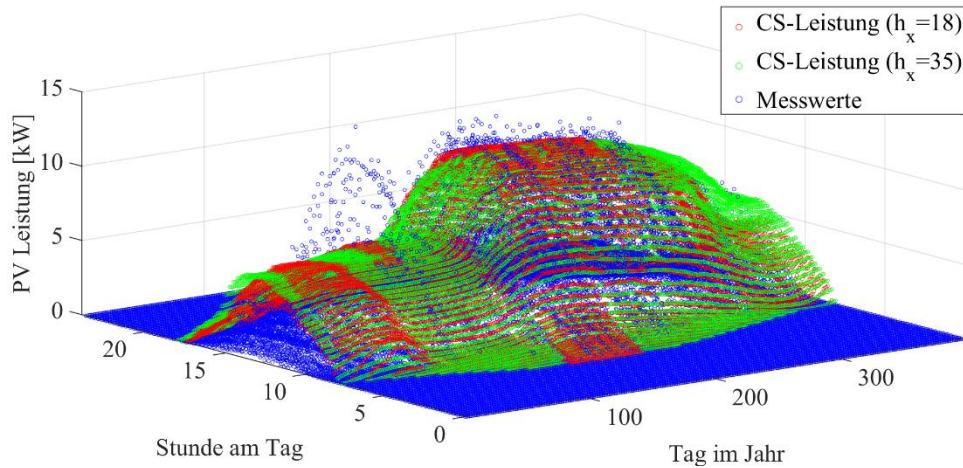


Abb. 2-1: Viertelstündlich aufgelöste PV-Leistung und entsprechende CS-Leistung

Der Hauptgrund hinter der Verwendung eines Clear-Sky-Modells liegt in zwei Aspekten. Für klassische Zeitreihenmodelle spielt die CS-Leistung eine entscheidende Rolle, da die PV-Leistung mit dem Clear-Sky-Faktor auf die CS-Leistung normalisiert wird. Anschließend werden die Zeitreihenmodelle für die Ermittlung des Clear-Sky-Faktors aufgebaut. Im Vergleich zur PV-Leistung erfüllt die Clear-Sky-Leistung eher die Bedingung der Stationarität, was zur besseren Modellperformance führt. Auch in Kombination mit Machine-Learning-Methoden gewinnt die Verwendung von Clear-Sky-Modellen an Bedeutung. Ein Vorteil der hier verwendeten Methode stellt ihre Selbstadaptivität dar.

2.2 Autoregressives Modell mit exogenem Eingang (ARX-Modell)

Ein autoregressives Modell mit exogenem Eingang (ARX-Modell) ist eine erweiterte Form des autoregressiven Modell (AR-Modell). Das AR-Modell hat die Form einer Linearkombination der gewichteten vergangenen Werten der Zeitreihe. In (Bacher 2008) wurde gezeigt, dass sich die PV-Prognoseergebnisse verbessern, wenn das AR-Modell um einen exogenen Eingang (X) ergänzt wird (ARX-Modell).

2.2.1 Mathematische Beschreibung des ARX-Modells

Ein ARX-Modell besteht aus zwei Eingängen, nämlich dem endogenen Eingang und dem exogenen Eingang. Das ARX-Modell ist wie folgt definiert:

$$Y_{t+k} = m + a_1 Y_t + a_2 Y_{t-s(k)} + b Z_{t+k} + e_{t+k} \quad (2-2)$$

Gleichung 2-2 verdeutlicht, dass der Zielwert zum Zeitpunkt $t+k$ (Y_{t+k}) von zwei Faktoren beeinflusst wird. Zum einen haben der endogene Eingang zum aktuellen Zeitpunkt t (Y_t) und der historische endogene Eingang ($Y_{t-s(k)}$) Einfluss auf den Zielwert Y_{t+k} , zum anderen ist der Zielwert vom exogenen Eingang zum Zeitpunkt $t+k$ (Z_{t+k}) abhängig. Für die Auswahl

des historischen endogenen Eingangs ($Y_{t-s(k)}$) werden die entsprechenden Lags ($s(k)$) durch eine Autokorrelationsanalyse bestimmt. Als exogenen Eingang sollen die zum endogenen Eingang relativ stark korrelierten Werte ausgewählt werden. Die Modellperformance hängt mit der Korrelation zwischen endogenem Eingang und exogenem Eingang zusammen.

2.2.2 Parameterberechnung des ARX-Modells

Der Prozess zur Parameterberechnung der optimalen Koeffizienten wird mithilfe der Maximum-Likelihood-Methode durchgeführt. Angenommen wird, dass einige stochastisch unabhängige Beobachtungen (X_i, Y_i) bekannt sind. Der Eingangsvektor ist

$$X_i = (1, Y_t, Y_{t-s(k)}, Z_{t+k}) \quad (2-3)$$

und der Modellparametervektor ist

$$\theta = (m, a_1, a_2, b). \quad (2-4)$$

Das ARX-Modell wird dann wie in Gleichung 2-5 umgeschrieben.

$$Y_i = X_i \theta^T + e_i \quad (2-5)$$

Die Zielfunktion

$$J(\theta) = \frac{1}{2} \sum_{i=1}^n (Y_i - X_i \theta^T)^2 \quad (2-6)$$

ist aus der Likelihood-Funktion und Dichtefunktion abgeleitet (Box et al. 2008) und wird minimiert, um die optimalen Koeffizienten des Modellparameters zu berechnen. Dafür wird die Zielfunktion nach θ differenziert und gleich Null gesetzt. Daher ergibt Gleichung 2-7 die Lösung des Modellparametervektors.

$$\theta = (X^T X)^{-1} X^T Y \quad (2-7)$$

2.3 Neuronales Netz

In zahlreichen Bereichen werden heutzutage Verfahren der neuronalen Netze eingesetzt, um sowohl Regressionsaufgaben als auch Klassifikationsaufgaben zu erledigen. Wegen der mächtigen Leistung des Ansatzes dominieren neuronale Netze im Bereich der künstlichen Intelligenz.

2.3.1 Struktur und Arbeitsweise des neuronalen Netzes

So wie viele andere Erfindungen in der Welt von der Natur inspiriert sind, werden künstliche neuronale Netze dem Gehirn von Menschen nachgebildet. Aus Neuronen und Kanten setzt sich ein neuronales Netz zusammen. Die Neuronen unterteilen sich in Eingangsneuronen, versteckte Neuronen und Ausgangsneuronen. Die Information der externen Umgebung wird zuerst von den Eingangsneuronen empfangen. Nach dem Informationsverarbeitungsprozess in den versteckten Neuronen wird die verarbeitete Information von den Ausgangsneuronen nach draußen gesendet. Alle Neuronen sind durch Kanten miteinander verknüpft, welche die Abhängigkeit zwischen den Neuronen beschreiben. So stellt ein neuronales Netz eine mehrschichtige, aus zahlreichen verknüpften Knotenpunkten aufgebaute Struktur dar, siehe Abb. 2-2.

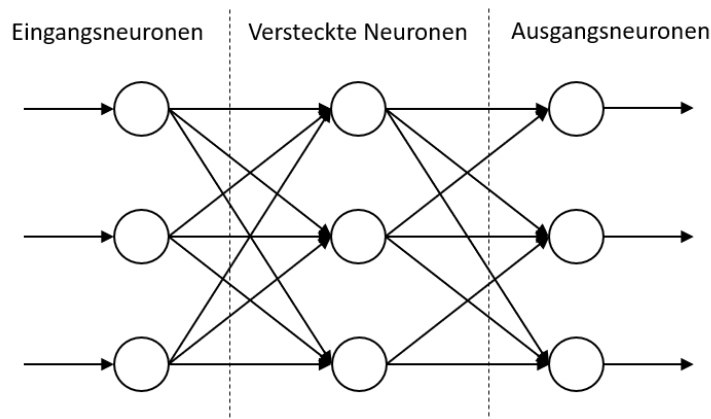


Abb. 2-2: Beispiel für den Aufbau eines neuronalen Netzes, nach (Kruse et al. 2015)

Um die Arbeitsweise eines neuronalen Netzes zu beschreiben, soll zunächst die Funktionsweise der Neuronen erläutert werden, siehe Abb. 2-3. Ein Neuron ist das Basiselement der neuronalen Netze. Das Neuron berechnet eine gewichtete Summe (net) ihrer Eingaben und wendet dann eine Aktivierungsfunktion (f_{act}) auf diese Summe an, um den Ausgabewert zu errechnen.

Aus Gleichung 2-8 ergibt sich die gewichtete Summe (net)

$$net = w_1x_1 + \dots + w_nx_n. \quad (2-8)$$

Eine im künstlichen Neuron häufig verwendete Aktivierungsfunktion (f_{act}) ist eine Schwellenwertfunktion.

$$f_{act}(net) = \begin{cases} 1, & \text{falls } net \geq 0 \\ 0, & \text{falls } net < 0 \end{cases} \quad (2-9)$$

Es stehen auch viele andere Möglichkeiten der Aktivierungsfunktion zur Verfügung. Nachdem die Berechnung eines einzelnen Neurons erläutert wurde, wird nun die Funktionsweise des neuronalen Netz veranschaulicht. Das ist als Netzwerk aus vielen Neuronen organisiert, indem die unterschiedlichen Neuronen durch Verbindungen miteinander verknüpft werden. Auf diese Weise lassen sich auf Basis vieler dieser einfachen Neuronen hochgradig nicht-lineare Zusammenhänge modellieren.

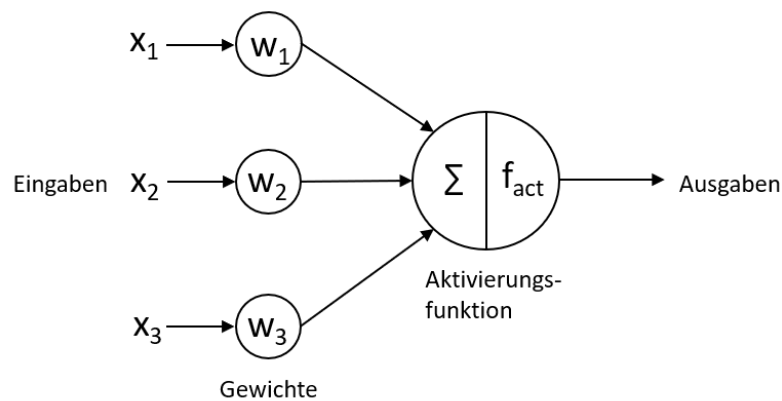


Abb. 2-3: Beispiel für den Aufbau eines künstlichen Neurons, nach (Kruse et al. 2015)

2.3.2 Training des neuronalen Netzes

Eine hohe Verfügbarkeit an Daten und der erhebliche Zuwachs an Rechenkapazität ermöglicht das Training von großen neuronalen Netzen. Bei der Trainingsphase werden Gewichte und Biaswerte entsprechend der Auswertung einer Fehlerfunktion angepasst und aktualisiert, bis die berechneten Ausgangswerte bestimmte Anforderungen an die entsprechenden Fehlermaße erfüllen. Je nach zu optimierendem Kriterium unterscheiden sich die Algorithmen der neuronalen Netze in vielen Arten. Auf Grund der schnellen Berechnungsgeschwindigkeit und hohen Robustheit wird der Levenberg-Marquardt-Algorithmus heutzutage häufig verwendet (Hagan et al. 1994). Es gibt allerdings bei der Wahl der Optimierungsalgorithmen keine eindeutig beste Lösung. Der sinnvollste Algorithmus ist stark von den verfügbaren Trainingsdaten und gewählten Fehlermaßen abhängig.

2.4 Random Forest

Eine Menge der unterschiedlichen Prädiktoren (z.B. Regressoren oder Klassifikatoren) bezeichnet man als ein Ensemble. Unter der Ensemble-Methode versteht man daher einen Algorithmus zum Ensemble-Learning. Im Vergleich zu einzelnen Vorhersagemodellen hat sich eine Verbesserung durch Ensemble-Methoden herausgestellt. Beispielweise lässt sich eine Gruppe von Entscheidungsbäumen auf unterschiedlichen, zufällig ausgewählten

Teilmengen der Trainingsdaten trainieren (Breiman, 2001). Solch ein Ensemble von Entscheidungsbäumen nennt man Random Forest.

2.4.1 Grundlagen des Random Forests

Entscheidungsbäume sind die funktionellen Komponenten von Random Forests. Die Struktur und Arbeitsweise eines Random Forests sind wie in Abb. 2-4 dargestellt. Auf den genauen mathematischen Hintergrund von Entscheidungsbäumen wird hier nicht eingegangen. Die Grundfunktionsweise von Entscheidungsbäumen ist in (Breiman, 2017) vorgestellt. Ein Nachteil der Entscheidungsbäume besteht darin, dass sie sehr empfindlich auf kleine Variationen in den Trainingsdaten reagieren (Raschka, 2018). Um dieser Instabilität des Entscheidungsbaumes entgegenzuarbeiten, soll eine effektivere Methode bei der Aufbereitung der Trainingsdaten eingesetzt werden. Das führt dazu, dass jeder einzelne Entscheidungsbaum anhand der Bagging-Methode mit einer zufällig ausgewählten Teilmenge der Trainingsdaten als Teil eines Random Forests trainiert wird (Breiman 1996).

Angenommen wird, dass ein paar unabhängige Beobachtungen (X_i, Y_i) bekannt gegeben. Für einen gegebenen Eingang (x) ist die Prognose durch Random Forests

$$\hat{\mu}(x) = \sum_{i=1}^n w_i(x) Y_i, \quad (2-10)$$

wobei $w_i(x)$ die Gewichte des Random Forests beschreibt.

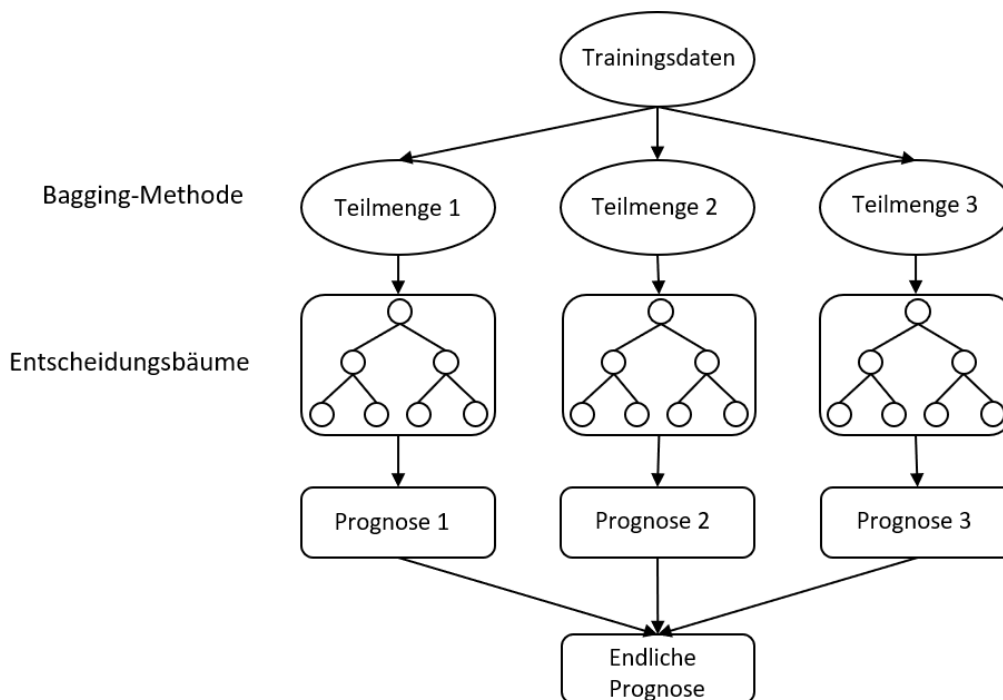


Abb. 2-4: Beispiel für den Aufbau eines Random Forests, nach (Breiman, 2001)

2.4.2 Grundlagen des Quantile Regression Forests

Für die Beschreibung von Aktivitäten mit impliziter Unsicherheit werden häufig probabilistische Prognosemethoden verwendet. Im Vergleich zur traditionellen Punktprognose ist bei der probabilistischen Prognosemethode ein Prognoseintervall vorherzusagen. Die bedingte Wahrscheinlichkeitsverteilung lässt sich durch Quantile Regression Forests modellieren. Aus der Wahrscheinlichkeitsverteilung ergibt sich daher das entsprechende Prognoseintervall, das aus zwei Quantilwerten besteht, siehe Gleichung 2-15.

Die bedingte Verteilungsfunktion von der Zielgröße (Y) mit gegebenen Eingang (X) ist

$$F(y | X = x) = P(Y \leq y | X = x). \quad (2-11)$$

Die durch Quantile Regression Forests berechnete bedingte Verteilungsfunktion ist

$$\hat{F}(y | X = x) = \sum_{i=1}^n w_i I_{\{Y_i \leq y\}}, \quad (2-12)$$

wo der Indikator (I) als folglich in Gleichung 2-12 definiert.

$$I_{\{Y_i \leq y\}} = \begin{cases} 1, & Y_i \leq y \\ 0, & Y_i > y \end{cases} \quad (2-13)$$

Aus der in Gleichung 2-12 bedingten Verteilungsfunktion ergibt sich der Quantilwert der Zielgröße, siehe Gleichung 2-14.

$$\hat{Y}_q = \inf\{y: \hat{F}(y | X = x) \geq q\} \quad (2-14)$$

Der Quantilwert von q beschreibt, dass es eine Wahrscheinlichkeit von q gibt, dass vorherzusagende Zielgröße unterhalb der Quantilwert (\hat{Y}_q) liegt.

Das Prognoseintervall setzt sich aus zwei Quantilwerten mit entsprechenden q -Quantilen zusammen, siehe Gleichung 2-15. Beispielsweise besteht ein 90%-Prognoseintervall aus dem 5%-Quantil-Wert als Untergrenze und dem 95%-Quantil-Wert als Obergrenze.

$$PI_{(1-q)} = [\hat{Y}_{\frac{q}{2}}, \hat{Y}_{1-\frac{q}{2}}] \quad (2-15)$$

2.5 Quantilsregression

Alternativ werden für eine probabilistischen Prognose häufig klassische statistische Methoden eingesetzt, um Prognoseintervalle vorherzusagen. Hier wird die relativ einfache lineare Quantilsregressionsmethode vorgestellt. Mit anderen Worten werden Quantilwerte mit der linearen Quantilsregression berechnet.

Es wird angenommen, dass ein paar unabhängige Beobachtungen (X_i, Y_i) bekannt sind. Die Definition der linearen Quantilsregression ist in Gleichung 2-16 beschrieben, wobei β der zu optimierende Parameter ist und e den Fehler darstellt.

$$Y = \beta X + e \quad (2-16)$$

Ähnlich wie andere Regressions-Methoden soll der Parameter in der linearen Quantilsregression nach einer bestimmten Fehlerfunktion optimiert werden (Koenker et al. 1978). Die Fehlerfunktion ist in Gleichung 2-17 beschrieben.

$$\rho(q, e) = \begin{cases} q e, & e \geq 0 \\ (1 - q) e, & e < 0 \end{cases} \quad (2-17)$$

Die Quantilwerte lassen sich dann in der linearen Quantilsregression wie in Gleichung 2-18 berechnen.

$$\hat{Y}_q = \beta_q X \quad (2-18)$$

Um diese Quantilwerte (\hat{Y}_q) zu erzielen, muss der Quantilfaktor (β_q) berechnet werden. Dabei handelt es sich um eine Lösung des Quantilfaktors bei der Lösung des Minimierungsproblems in Gleichung 2-19.

$$\beta_q = \underset{\beta}{\operatorname{argmin}} \sum_1^n \rho(q, e) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n \rho(Y_i, -\beta X_i) \quad (2-19)$$

Dann lassen sich Gleichung 2-18 und Gleichung 2-19 in die Gleichung 2-15 einsetzen, um das gewünschte Prognoseintervall zu erhalten.

2.6 Fehlermaße

In diesem Abschnitt werden ein paar in der Literatur zur Bewertung der Prognosequalität häufig verwendete Fehlermaße vorgestellt. Ein typisches Fehlermaß ist der Root Mean Square Error (RMSE). Der RMSE ist die Wurzel der gemittelten Summe der Abweichungen

zwischen prognostiziertem Zielwert (\hat{Y}) und tatsächlichem Zielwert (Y) in einem Datensatz mit n Daten, siehe Gleichung 2-20.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2} \quad (2-20)$$

Durch eine Normalisierung auf den mittleren tatsächlichen Zielwert wird der Normalized Root Mean Square Error (NRMSE) berechnet, siehe Gleichung 2-21.

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2}}{\frac{1}{n} \sum_{i=1}^n Y_i} \quad (2-21)$$

Obwohl der RMSE und der NRMSE bei der Prognoseauswertung häufig verwendet werden, sind in manchen Situationen andere Fehlermaße zu empfehlen. Beispielsweise ist der Mean Absolute Error (MAE) für einen Datensatz mit vielen Ausreißern geeignet, siehe Gleichung 2-22.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{Y}_i - Y_i| \quad (2-22)$$

Ein weiteres Fehlermaß ist der Mean Absolute Percentage Error (MAPE), siehe Gleichung 2-23.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \frac{|\hat{Y}_i - Y_i|}{Y_i} \quad (2-23)$$

Die oben genannten Fehlermaße werden häufig bei der Auswertung von deterministischen Prognosen benutzt. Andere Fehlermaße werden für die Auswertung probabilistischer Prognosen benötigt.

Ein komprehensives Fehlermaß für probabilistische Prognose ist der Winkler Score (*Winkler*), der eine gemeinsame Auswertung des Prognoseintervalls ermöglicht (Hong et al. 2016). Für ein wie in Gleichung 2-15 definiertes Prognoseintervall ($PI_{(1-q)}$) ist der Winkler Score

$$Winkler(1 - q) = \begin{cases} \delta, & \text{falls } \hat{Y}_{\frac{q}{2}} \leq Y \leq \hat{Y}_{1-\frac{q}{2}} \\ \delta + \frac{2\left(\hat{Y}_{\frac{q}{2}} - Y\right)}{q}, & \text{falls } Y < \hat{Y}_{\frac{q}{2}} \\ \delta + \frac{2\left(Y - \hat{Y}_{1-\frac{q}{2}}\right)}{q}, & \text{falls } Y > \hat{Y}_{1-\frac{q}{2}} \end{cases}, \quad (2-24)$$

wobei δ die Abweichung zwischen den beiden Grenzwerten des Prognoseintervalls ist, siehe Gleichung 2-25. Der Winkler Score gibt eine Strafe, wenn die tatsächlichen Zielwerte (Y) außerhalb des berechneten Prognoseintervalls (PI) liegen. Im Gegensatz dazu wird es belohnt, wenn sich die tatsächlichen Zielwerte (Y) im Prognoseintervall befinden (Hong et al. 2016).

$$\delta = \hat{Y}_{\frac{q}{2}} - \hat{Y}_{1-\frac{q}{2}} \quad (2-25)$$

Sowohl bei deterministischen Prognosen als auch bei probabilistischen Prognosen wird das Improvement (*Imp*) häufig verwendet, um die Modellperformance von unterschiedlichen Prognosemethoden hinsichtlich eines Prognosemaßes miteinander zu vergleichen. Dadurch erkennt man eine Erhöhung oder eine Senkung der Modellperformance bezogen auf ein vorher ausgewähltes Referenzmodell, siehe Gleichung 2-26.

$$Imp = \frac{\text{Fehlermaß}_{\text{Referenz}} - \text{Fehlermaß}}{\text{Fehlermaß}_{\text{Referenz}}} 100\% \quad (2-26)$$

3 Konzeption der Prognosemodelle

In diesem Kapitel wird zuerst das Konzept des Hybridmodells vorgestellt, das anschließend für die Erstellung der PV-Prognose in dieser Arbeit implementiert wurde. Dann wird jede in dieser Arbeit verwendete Konzeption des Regressionsmodells erläutert. Anschließend wird die Wahl des in dieser Arbeit benutzten Clear-Sky-Modells erörtert.

3.1 Konzeption des Hybridmodells

Das in dieser Arbeit umgesetzte Hybridmodell ist eine Kombination des in Kapitel 2.1 vorgestellten Clear-Sky-Modells mit verschiedenen Regressionsmodellen. Das Konzept des Hybridmodells ist in Abb. 3-1 dargestellt. Die CS-Leistung wird vom CS-Modell berechnet und der CS-Faktor wird als Zielgröße des Regressionsmodells geliefert. Schließlich ergibt sich die prognostizierte PV-Leistung für den angeforderten Prognosehorizont aus dem Produkt von CS-Leistung und CS-Faktor.

Ein Hauptgrund für den Ansatz des Hybrid-Modells besteht darin, dass der CS-Faktor bei der Anwendung eines klassischen Zeitreihenmodells (z.B. ARX-Modell) eine bessere Stationarität der Zeitreihe aufweist. Daher wird die Modellperformance tendentiell verbessert (Bacher 2008). Ein weiteres Merkmal des verwendeten Hybrid-Modells liegt darin, dass die durch das CS-Modell ermittelte CS-Leistung bei der Trainingsphase in den Machine-Learning-Algorithmen eingespeist wird. Mit dem Ansatz des CS-Modells und der Ausnutzung der CS-Leistung wird eine höhere Genauigkeit des Prognoseergebnisses ermöglicht.

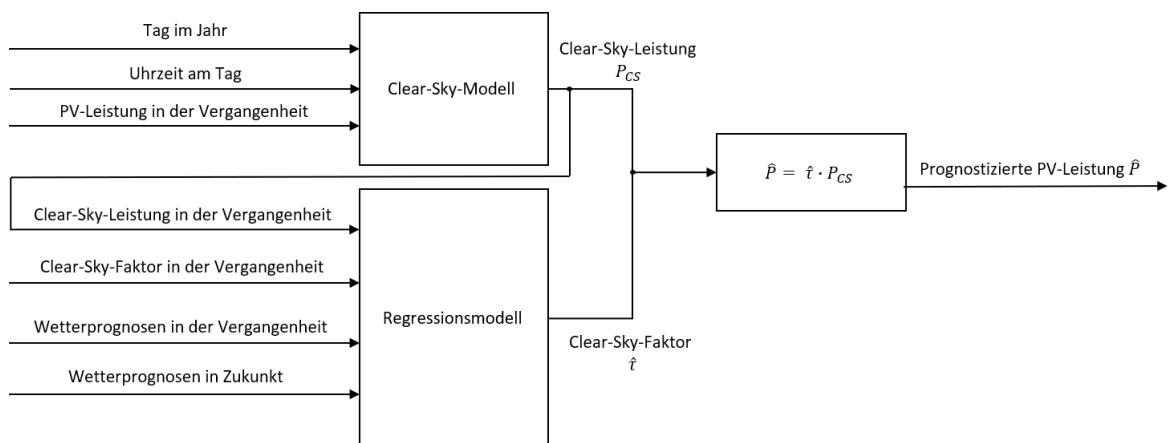


Abb. 3-1: Aufbau des Hybridmodells, nach (Decker, 2018)

3.2 Konzeption der Regressionsmodelle

Die heutzutage in der Literatur häufig verwendeten Regressionsmodelle sind im Wesentlichen in zwei Gruppen aufgegliedert. Eine Gruppe der Methoden lässt sich den klassischen statistischen Methoden zuordnen, für die normalerweise historische Daten erforderlich sind, um die Saisonalität und Periodizität der PV-Erzeugung zu erfassen. Die andere Gruppe der Methoden gehört zu den Machine-Learning-Algorithmen, die sehr große Datensätze als Trainingsdaten benötigen. In dieser Arbeit wird ein ARX-Modell als ein Vertreter der statistischen Methoden untersucht. Außerdem werden ein neuronales Netz und ein Random Forest als Anwendungen der Machine-Learning-Algorithmen trainiert und angewendet.

3.2.1 Konzeption des ARX-Modells

Der Aufbau des verwendeten ARX-Modells ist in Abb. 3-2 dargestellt. Das ARX-Modell nimmt den CS-Faktor zum aktuellen Zeitpunkt t (τ_t) und den zum vergangenen Zeitpunkt $t-s$ beobachteten CS-Faktor ($\tau_{t-s(k)}$) als endogene Eingänge. Als exogener Eingang wird ein Wetter-Faktor (τ_{t+k}^{nwp}) verwendet, der aus einer Wetterprognose (*NWP – numerical weather prediction*) für den zukünftigen Zeitpunkt $t+k$ abgeleitet wird (Bacher et al. 2009), um den Wettereinfluss zu berücksichtigen. Die genaue Ausgestaltung der exogenen und endogenen Eingänge wird in Abschnitt 4.2 beschrieben.

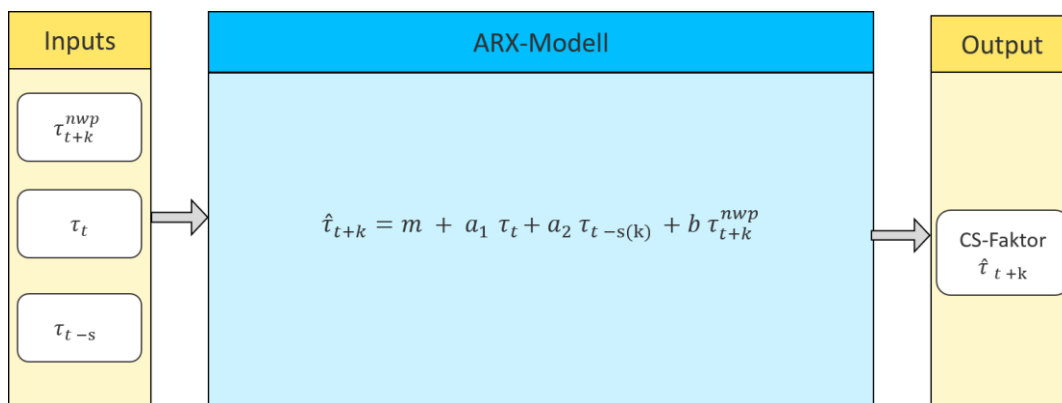


Abb. 3-2: Aufbau des verwendeten ARX-Modells, nach (Bacher et al. 2009)

3.2.2 Konzeption des neuronalen Netzes

Das in dieser Arbeit benutzte neuronale Netz wird als Feed-Forward-Netz bezeichnet (Grossberg 1988). Die relativ einfach aufgebaute Struktur des neuronalen Netzes ist in Abb. 3-3 zu sehen.

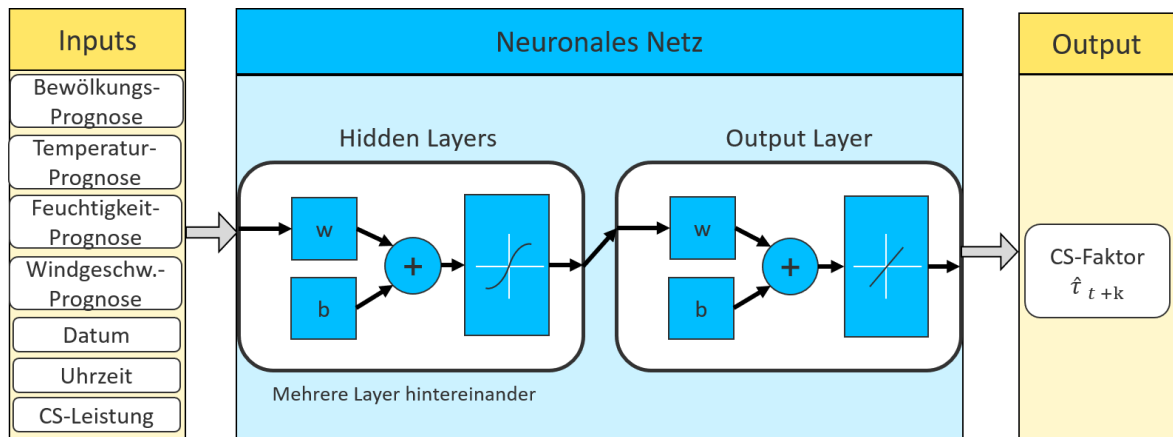


Abb. 3-3: Aufbau des verwendeten neuronalen Netzes, nach (Grossberg 1988)

Die durch das CS-Modell berechnete CS-Leistung und die Wetterprognose-Parameter lassen sich als Inputdaten ins Feed-Forward-Netz einspeisen, um das Netz zu trainieren. Anschließend wird mit dem trainierten Modell die Prognose des CS-Faktors als Zielgröße getroffen.

3.2.3 Konzeption der Kombination eines ARX-Modells mit neuronalem Netz

Eine durchdachte Auswahl des exogenen Eingangs für ein ARX-Modell trägt einen wichtigen Beitrag zum Zuwachs der Modellleistung. In (Bacher 2008) wird der exogene Eingang über die prognostizierte Sonneneinstrahlung modelliert. Ein Hauptgrund dafür besteht in der hohen Abhängigkeit des CS-Faktors von der Sonneneinstrahlung. Anders ausgedrückt, besitzt die Sonneneinstrahlung eine hohe Korrelation mit dem CS-Faktor. Jedoch steht in dieser Arbeit die Sonneneinstrahlung im Umfang der vorliegenden Wetterprognosen nicht zur Verfügung. Daher wird hier als externer Input Temperatur oder Bewölkungsgrad verwendet. Da die Temperatur oder der Bewölkungsgrad nicht so hoch mit der PV-Leistung bzw. dem CS-Faktor korrelieren wie die Sonneneinstrahlung, erreicht das in dieser Arbeit aufgebaute ARX-Modell jedoch nicht die gleiche Modellleistung wie die des in (Bacher 2008) berechneten ARX-Modells.

Daher entstand die Idee, ein ARX-Modell mit einem neuronalen Netz zu kombinieren, wie es in Abb. 3-4 zu sehen ist. Somit können mehr Information aus den Wetterprognosen, d.h. mehrere Wetterparameter, durch das neuronale Netz ausgenutzt werden, um den Einfluss des Wetters umfassender wiederzugeben.

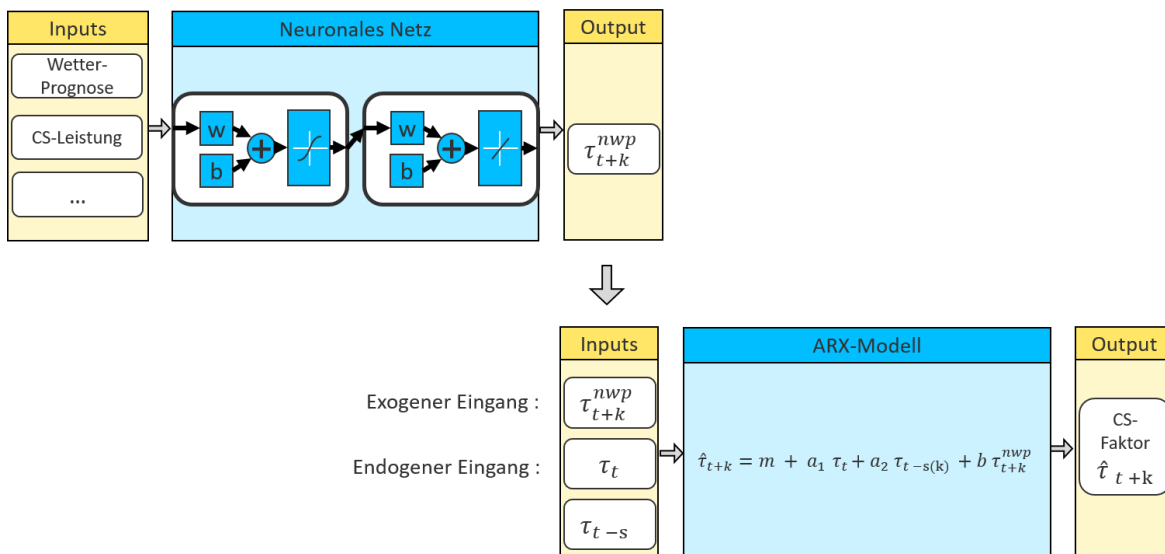


Abb. 3-4: Aufbau der Kombination eines ARX-Modells mit neuronalen Netze

3.2.4 Konzeption des Random Forests

Analog zum neuronalen Netz werden auch für den Random Forest die CS-Leistung und die Wetterprognoseparameter als Inputdaten verwendet. Der Aufbau des verwendeten Random Forest ist in Abb. 3-5 dargestellt. Weil die Inputdaten für das neuronale Netz und den Random Forest gleich gesetzt sind, lassen sich die beiden Machine-Learning-Algorithmen bei der Prognoseauswertung fair miteinander vergleichen.

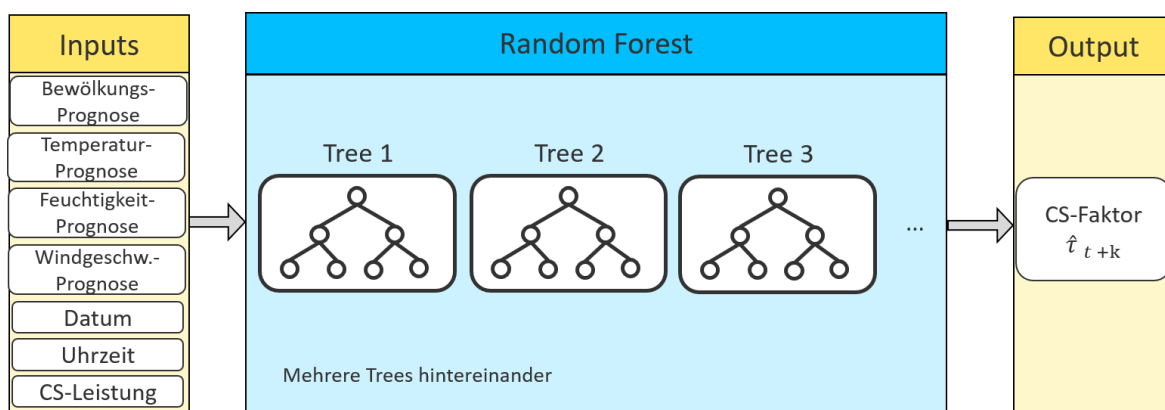


Abb. 3-5: Aufbau des verwendeten Random Forests, nach (Breiman, 2001)

3.3 Konzeption des Clear-Sky-Modells

Das Grundprinzip des in dieser Arbeit verwendeten CS-Modells ist in Abschnitt 2.1 vorgestellt. Das nichtparametrische CS-Modell ermöglicht die Berechnung der CS-Leistung

ohne die Angabe von physikalischen Kennzahlen der lokalen PV-Anlage (z.B. Neigungswinkel und Azimutwinkel der PV-Anlage) und berücksichtigt zudem lokale Gegebenheiten wie z.B. saisonale Verschattungen. Es setzt allerdings voraus, dass historische Daten der PV-Leistung über mindestens ein Jahr verfügbar sind, damit die CS-Leistung selbstadaptiv berechnet werden kann.

4 Implementierung und Parametrierung der Prognosemodelle

In diesem Kapitel wird beschrieben, wie die in Kapitel 3 vorgestellten Konzepte der Prognosemodelle in Matlab implementiert werden.

4.1 Struktur des Programms

Wie in Kapitel 1 beschrieben, soll das Prognosesystem echtzeitfähig sein. Sowohl die Wetterprognosedaten als auch die PV-Leistungsdaten müssen daher dynamisch abgefragt und in das Prognosemodell eingespeist werden können. Die Funktionen für das Auslesen der Echtzeitdaten von PV-Leistungen und Wetterprognosen werden in Python entwickelt. Um die in Matlab aufgebauten Prognosemodelle mit den mit Python ausgelesenen Echtzeitdaten zu verknüpfen, wird hier die Datenbank SQLite ausgewählt, die eine hohe Kompatibilität sowohl für Matlab als auch für Python anbietet. Dadurch wird ein Datenaustausch zwischen Matlab und Python ermöglicht. Der Programmablaufplan ist in Abb. 4-1 dargestellt. Zur Strukturierung des Gesamtprogramms wird das Programm in mehrere Teilprogramme zerlegt. In dieser Arbeit werden im Wesentlichen vier Teilprogramme entwickelt, wie in Abb. 4-2 zu sehen sind, und dann entsprechend des Ablaufplans verknüpft.

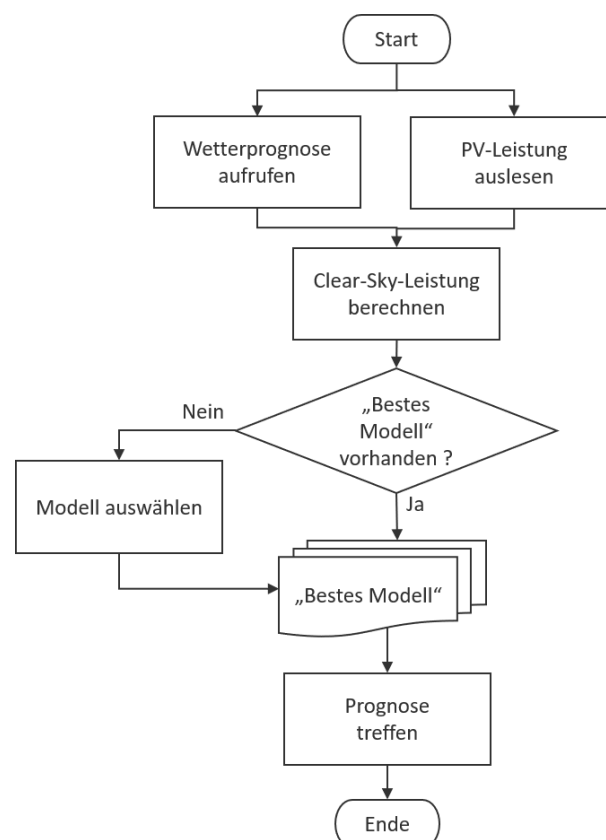


Abb. 4-1: Ablaufplan des Hauptprogramms

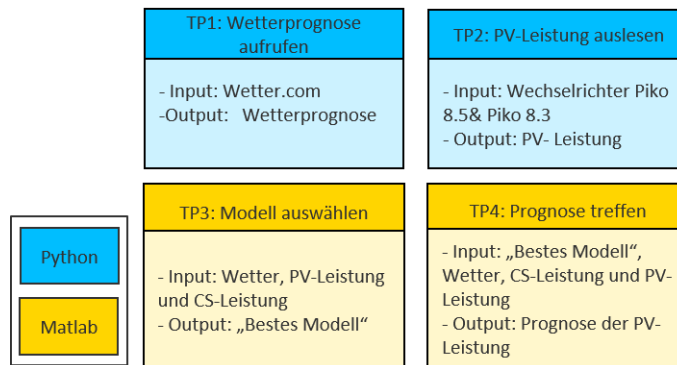


Abb. 4-2: Übersicht der Teilprogramme

4.2 Dynamisches Abrufen von Wetterprognose und PV-Leistungsdaten und Datenaufbereitung

Nach dem Start des Programms werden zuerst die beiden in Python implementierten Teilprogramme aufgerufen. Das Aufrufen der Wetterprognose für die nächsten 72 Stunden wird mit TP1 von der Webseite Wetter.com durchgeführt. Es kann hierbei auf ein bereits bestehendes Skript zurückgegriffen werden, das im Zuge der Arbeit etwas modifiziert und somit wieder lauffähig gemacht wird. Die heruntergeladenen Wetterdaten sind stündlich aufgelöst und bestehen aus der Temperatur, dem Luftdruck, der Feuchtigkeit, der Windrichtung, der Windgeschwindigkeit, der Niederschlagsmenge, dem Niederschlagsrisiko und dem Bewölkungsgrad für die nächsten 72 Stunden für den Standort Stuttgart-Vaihingen. Um später 15-minütig aufgelöste Prognosen zu erstellen, werden die stündlichen Wetterdaten durch lineare Interpolation in 15-minütig aufgelöste Daten überführt.

Mit TP2 ist es möglich, die Leistung der PV-Anlage auf dem Dach des IER durch zwei feste IP-Adressen dynamisch auszulesen, wenn die beiden PV-Wechselrichtern mit dem Uni-Netz verbunden sind. Die Daten der PV-Leistung werden 15-minütig aggregiert weiterverwendet.

Eine ausführliche Beschreibung der Datenerfassung der Wetterprognose und PV-Leistung wird im Anhang vorgestellt. Nachdem die beiden Daten von Python in die SQLite-Datenbank exportiert wurden, werden sie weiter zu Matlab geliefert.

4.3 Ablauf der Prognoseerstellung

Nachdem Wetterdaten und PV-Leistungen zeitlichen synchronisiert wurden, kann mit der Umsetzung der eigentlichen Prognose begonnen werden. Daher wird hier die Implementierung der Prognosemodelle vorgestellt und die Auswahl der jeweiligen Hyperparameter wird eingehend untersucht.

Den nächsten Schritt stellt das Clear-Sky-Modell auf Basis von (Bacher et al. 2009) vor, welches bei Beginn der Arbeit bereits in Matlab implementiert vorliegt (Decker 2018). In dem in Kapitel 2.1 vorgestellten CS-Modell werden hier nur ein paar Modellparameter an die Institutsanlage neu angepasst. Eine weitergehende Untersuchung des CS-Modells wird im Rahmen der Arbeit nicht vorgenommen.

Nachdem die Wetterprognose, die PV-Leistung und CS-Leistung bereitgestellt sind, kommt das TP3 zum Einsatz, das als ein Kernbestandteil des Programms betrachtet wird, siehe Abb. 4-3. Alle in Kapitel 3 vorgestellten Prognosemodelle werden zunächst in TP3 implementiert und mit einem Datensatz des Zeitraums vom 01.01.2018 bis zum 31.09.2018 trainiert. Danach werden alle trainierten Modelle mit dem Datensatz des Zeitraums vom 01.10.2018 bis zum 27.10.2018 getestet und ausgewertet. Ergebnis des TP3 ist somit das ‚beste‘ Modell, d.h. das Modell, für das für den Testzeitraum der geringste Fehler zwischen Prognose und Messdaten erzielt wurde. Das ‚beste‘ Modell ist in Anführungszeichen gesetzt, da das Ergebnis auch vom verwendeten Fehlermaß abhängt und somit keine allgemeingültige Rangfolge der Modelle gebildet werden kann. Schließlich wird das von TP3 ausgewählte Modell in TP4 mit den Echtzeitdaten verknüpft, um eine Prognose für die nächsten 3 Tage zu treffen. Aufgrund der stetigen Zunahme der Datengrundlage sollten die Prognosemodelle regelmäßig mit neuen Daten trainiert und getestet werden, damit stets das ‚beste‘ Modell verwendet werden kann.

Der Programmablaufplan vom Start bis zum Ende veranschaulicht den Ablauf einer Prognoseerstellung zu einem bestimmten Zeitpunkt. Sobald das Programm rund um die Uhr auf einem Server laufen kann, ist es in der Lage, mit Echtzeitdaten Prognosen dynamisch zu erstellen.

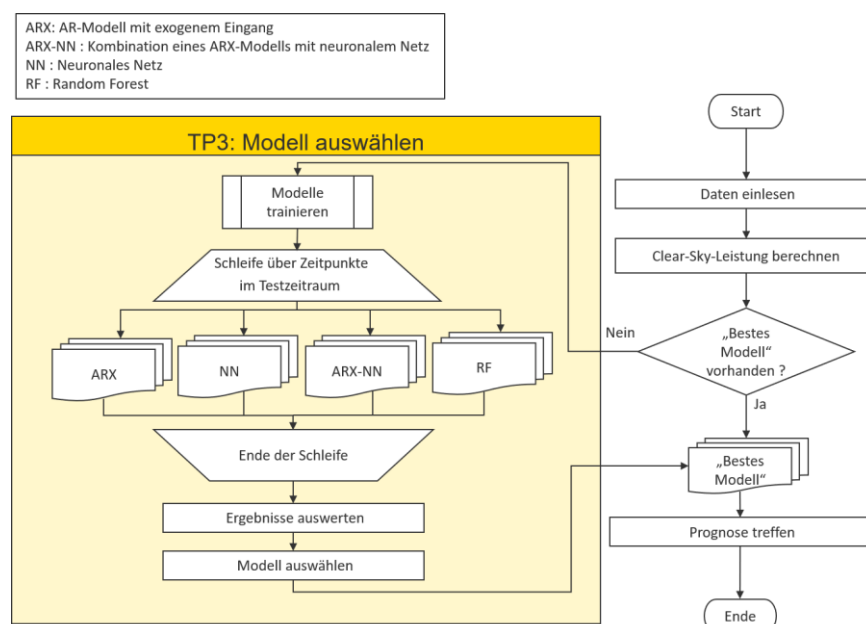


Abb. 4-3: Programmablaufplan mit TP3

4.4 Parametrierung der Prognosemodelle

Um eine hohe Modellleistung und eine hohe Präzision der Prognoseergebnisse zu erzielen, müssen zuerst die implementierten Regressionsmodelle hinsichtlich ihrer Parameter optimiert werden. Die Daten der Wetterprognose und PV-Leistung sind im Zeitraum vom 01.01.2018 bis zum 27.10.2018 verfügbar. Die Daten lassen sich in Trainingsdaten und Testdaten unterteilen. Alle Modelle werden mit Trainingsdaten des Zeitraums vom 01.01.2018 bis zum 30.09.2018 trainiert. Schließlich erfolgt die Auswertung der Regressionsmodelle mit Testdaten des Zeitraums vom 01.10.2018 bis zum 27.10.2018. Die Tabelle 4-1 gibt eine Übersicht der in diesem Kapitel untersuchten Regressionsmodelle.

Tabelle 4-1: Übersicht der in dieser Arbeit untersuchten Regressionsmodelle

Regressionsmodell	Beschreibung
ARX-T	ARX-Modell mit Temperatur als exogenem Eingang
ARX-B	ARX-Modell mit Bewölkungsgrad als exogenem Eingang
NN	Neuronales Netz
ARX-NN	Kombination eines ARX-Modells mit neuronalem Netz
RF	Random Forest

4.4.1 Parametrierung des ARX-Modells

Es gibt zwei wichtige Eingänge, die im ARX-Modell bestimmt werden müssen. Die Auswahl des endogenen Eingangs wird mit einer Autokorrelationsanalyse getroffen. Die Auswahl des exogenen Eingangs erfolgt durch eine Korrelationsanalyse.

Auswahl des endogenen Eingangs

Das in Gleichung 2-2 definierte ARX-Modell wird an unsere Prognosedaten angepasst. Es wird in Gleichung 4-1 transformiert.

$$\hat{\tau}_{t+k} = m + a_1 \tau_t + a_2 \tau_{t-s} + b \tau_{t+k}^{nwp} \quad (4-1)$$

Angenommen wird, dass τ_t der aktuelle CS-Faktor ist und τ_{t-s} ein historischer CS-Faktor zum Zeitpunkt $t-s$. Dann muss die Frage beantwortet werden, von welchen historischen CS-Faktoren der zu prognostizierende CS-Faktor abhängt. Diese Frage wird mit der Autokorrelationsanalyse beantwortet, siehe Abb. 4-4.

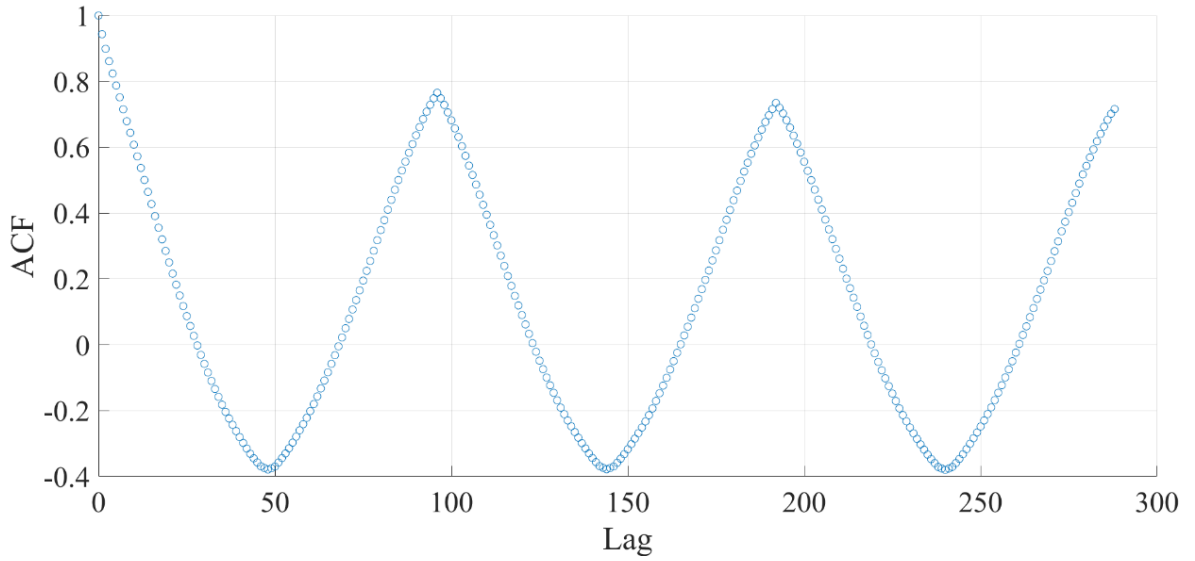


Abb. 4-4: Autokorrelationsfunktion des CS-Faktors

Durch die Autokorrelationsfunktion ist deutlich zu sehen, dass an Stellen der Lags 96, 192 und 288 die höchsten Werte auftreten. Wie in Abschnitt 1.2 beschrieben, ist der Prognosehorizont 72 Stunden und die Prognoseauflösung 15-minütig. In diesem Fall entsprechen die Lags 96, 192 und 288 den Zeitpunkten von 24 Stunden, 48 Stunden und 72 Stunden vor dem prognostizierten Zeitpunkt. Daher wird das Lag (s) in Gleichung 4-2 definiert,

$$s(k) = \begin{cases} 96 + k, & \text{wenn } k < 96 \\ 192 + k, & \text{wenn } 96 \leq k < 192 \\ 288 + k, & \text{wenn } 192 \leq k \leq 288 \end{cases} \quad (4-2)$$

wobei k der Prognosezeitpunkt vom aktuellen Zeitpunkt aus gesehen ist. Beispielsweise ist eine 1-Step-Prognose ($k=1$) die Prognose für den Zeitpunkt von 15 Minuten nach dem aktuellen Zeitpunkt (t) und in Gleichung 4-3 dargestellt.

$$\hat{\tau}_{t+1} = m + a_1 \tau_t + a_2 \tau_{t-95} + b \tau_{t+1}^{nwp} \quad (4-3)$$

Die Auswahl des exogenen Eingangs (τ^{nwp}) wird im nächsten Abschnitt erläutert.

Auswahl des exogenen Eingangs

In (Bacher et al. 2009) wird die Sonneneinstrahlung der Wetterprognose als exogener Eingang verwendet, um das ARX-Modell aufzubauen. In (Bacher et al. 2009) befindet sich keine konkrete Begründung für die Auswahl der Sonneneinstrahlung, jedoch korreliert die Sonneneinstrahlung stark mit den PV-Leistungsdaten, welche über die CS-Leistung in den

CS-Faktor umgerechnet werden können. In (Bacher et al. 2009) wird mit der Prognose der Sonneneinstrahlung zuerst der CS-Faktor durch eine lineare Regression modelliert

$$\tau_{t+k}^{nwp} = f_{nwp}(Z_{t+k}). \quad (4-4)$$

Die Variable τ_{t+k}^{nwp} ist der über die Wetterprognose (Z_{t+k}) modellierte CS-Faktor. Dementsprechend wird der erwartete Wettereinfluss zum Zeitpunkt $t+k$ durch diese Variable (τ_{t+k}^{nwp}) im ARX-Modell auch berücksichtigt (Bacher et al. 2009).

Aber im vorliegenden Fall ist in den Wetterprognosedaten keine Prognose der Sonneneinstrahlung verfügbar. Daher wird hier zunächst die Korrelation für jeden verfügbaren Wetterprognoseparameter analysiert, um eine mit dem CS-Faktor (τ) möglichst hoch korrelierte Variable (τ_{t+k}^{nwp}) herauszufinden. Die Temperatur, der Bewölkungsgrad, der Luftdruck und die Windgeschwindigkeit werden in die Korrelationsanalyse einbezogen. In Abb. 4-5 ist die PV-Leistung über jeden oben genannten Wetterprognoseparameter aufgetragen. Außerdem wird für jeden Wetterparameter eine lineare Regression daran angepasst.

Um die Korrelation zwischen der PV-Leistung und den Wetterprognoseparametern herauszufinden, werden die Korrelationskoeffizienten berechnet.

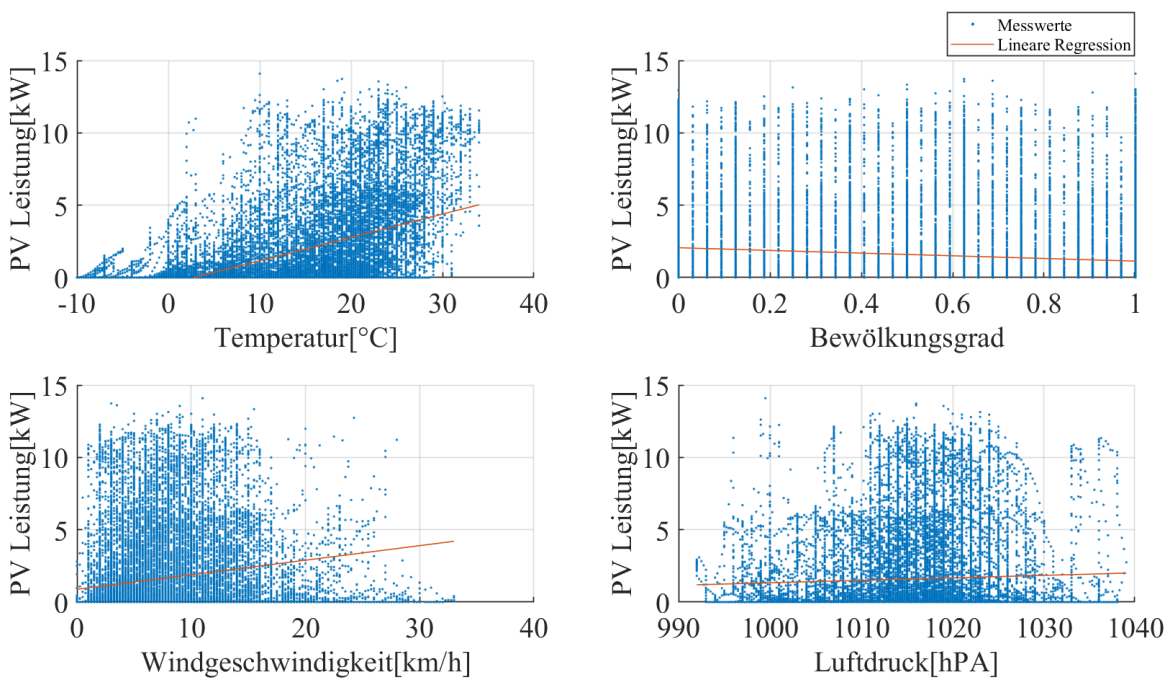


Abb. 4-5: Scatterplot – PV-Leistung gegen Wetterprognosen

Nach der Berechnung der Korrelationskoeffizienten zwischen der PV-Leistung und den Wetterprognoseparametern, siehe Abb. 4-6, ist es deutlich zu beobachten, dass die PV-

Leistung mit der Temperatur am stärksten korreliert ist. Die Korrelationskoeffizienten des Bewölkungsgrades und der Windgeschwindigkeit folgen danach.

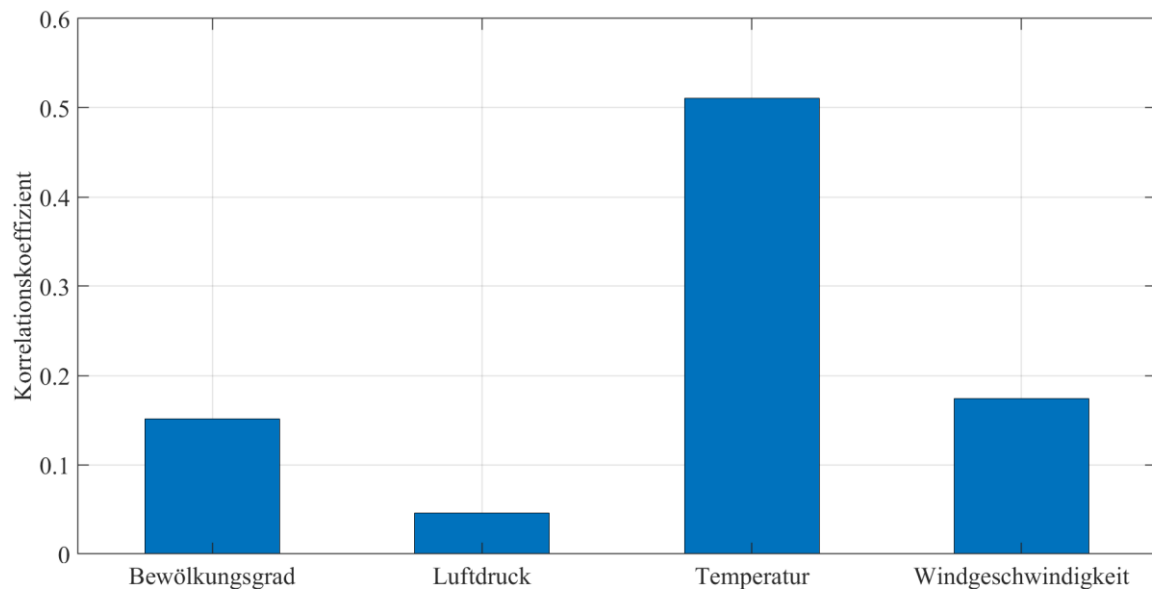


Abb. 4-6: Korrelationskoeffizienten zwischen der PV-Leistung und Wetterprognosen

Mit der Korrelationsanalyse wird somit festgestellt, dass für den exogenen Input des ARX-Modells die Temperatur das meistversprechende Wettermerkmal ist, obwohl auch hier der Korrelationskoeffizient mit $< 0,55$ recht gering ist. Es wird daher zuerst ein ARX-Modell mit Temperatur als exogenem Eingang (ARX-T-Modell) aufgebaut. Analog zur Temperatur wird für ein zweites Modell der Bewölkungsgrad als exogener Eingang (ARX-B-Modell) genommen. In Abb. 4-7 sind die Prognoseergebnisse beider ARX-Modelle mit dem NRMSE bewertet dargestellt. Der Vergleich zwischen den beiden Modellen zeigt, dass aufgrund der geringeren Korrelation das ARX-B-Modell den größeren Prognosefehler aufweist, wenn der in Abschnitt 2.6 vorgestellte NRMSE als Fehlermaß ausgewählt wird. Das ARX-B-Modell erreicht für den Testzeitraum ein Prognoseergebnis mit einem NRMSE von 0,6 bis 0,8 für den Prognosehorizont von 2 bis 72 Stunden, in den ersten 2 Stunden ist der NRMSE geringer.

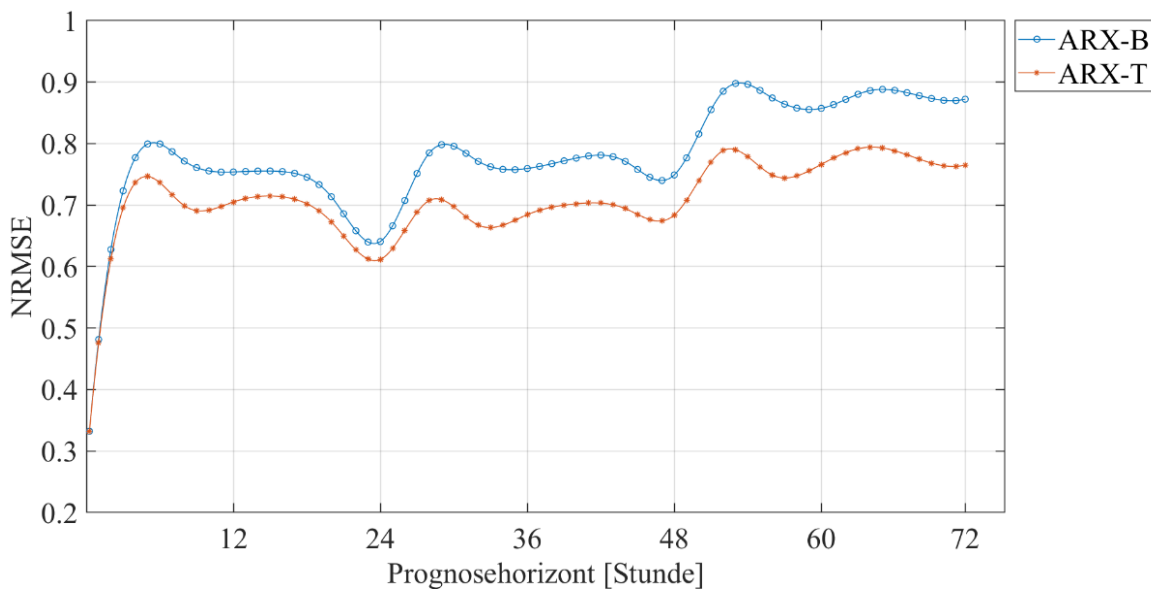


Abb. 4-7: NRMSE des ARX-T-Modells und des ARX-B-Modells

4.4.2 Parametrierung des neuronalen Netzes

Um mit dem implementierten neuronalen Netz eine hohe Modellleistung erzielen zu können, müssen einige Parameter sinnvoll gewählt werden. Einerseits haben die Hyperparameter (z.B. Struktur des neuronalen Netzes) einen wichtigen Einfluss auf die Modellperformance, andererseits spielen auch die Inputparameter und die Trainingsdaten eine entscheidende Rolle.

Auswahl der Merkmale

Ein neuronales Netz wird nur etwas erlernen können, wenn die Trainingsdaten ausreichend viele Informationen aufgrund von relevanten Merkmalen enthalten. Die Wahl der Inputfaktoren wird in dieser Arbeit nicht näher untersucht und es werden dieselben Merkmale verwendet wie in (Decker 2018). Damit werden folgende Merkmale als Inputparameter des Neuronalen Netzes verwendet: der Bewölkungsgrad, die Temperatur, die Windgeschwindigkeit, die relative Feuchtigkeit, die CS-Leistung, der Tag im Jahr und die Uhrzeit.

Auswahl des Hyperparameters

Wie in Abschnitt 2.3.1 vorgestellt, ist ein neuronales Netz im Allgemeinen eine mehrschichtige Struktur. Die Anzahl der Schichten (Layer) gehört daher zu einem wichtigen Hyperparameter des neuronalen Netzes. Für die Optimierung des neuronalen Netzes wird

daher die Anzahl der Schichten (Layer) von 4 bis 14 variiert und die resultierenden Prognosen werden mit dem NRMSE bewertet

In Abb. 4-8 ist der NRMSE für die Ergebnisse der NN-Modelle mit 4 bis 14 Schichten über dem Prognosehorizont von null bis 72 Stunden dargestellt. Das Optimum der Layeranzahl liegt hier für alle Zeithorizont bei 10 Layern. Im Vergleich dazu wird der größte Fehler mit Layers 12 verursacht. Dies unterstreicht, dass es eine umstrittene Frage ist, ob eine Zunahme der Layers auch einen Zuwachs der Modellperformance erbringen kann. Weil die Auswahl der Layeranzahl sehr situationsabhängig ist, sollte die Struktur des neuronalen Netzes für unterschiedliche Anwendungsfälle stets erneut angepasst werden. Eine verallgemeinerte Empfehlung für die Wahl der Hyperparameter wird hier nicht gegeben.

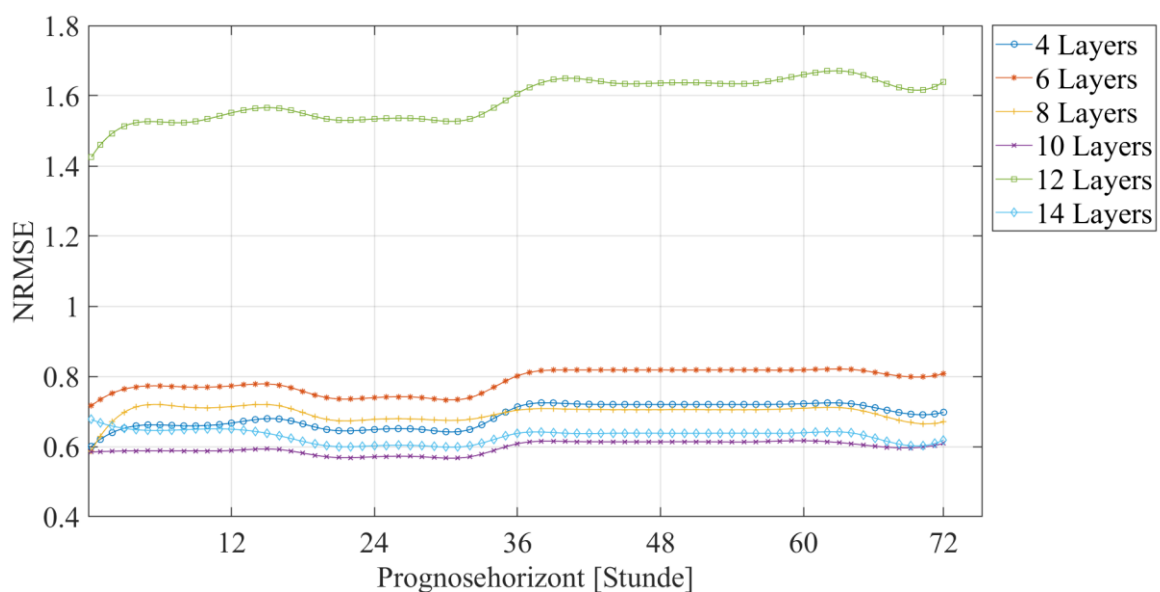


Abb. 4-8: NRMSE der neuronalen Netze mit unterschiedlichen Layers

Auswahl der Trainingsdatenmenge

Nachdem die Optimierung des Hyperparameters der Layeranzahl durchgeführt wurde, soll zunächst untersucht werden, welchen Einfluss die Trainingsdatenmenge auf das Prognoseergebnis des untersuchten neuronalen Netzes hat. Für ein neuronales Netz mit 10 Layern werden hier jeweils unterschiedliche Datenmengen verwendet. Das Diagramm in Abb. 4-9 veranschaulicht, dass der Prognosefehler mit der Zunahme der Trainingsdatenmenge von 3 Monaten auf 9 Monate eine sinkende Tendenz aufweist. Anders ausgedrückt, bedeuten für das in dieser Arbeit untersuchte neuronale Netz mehr Trainingsdaten eine bessere Prognosegüte.

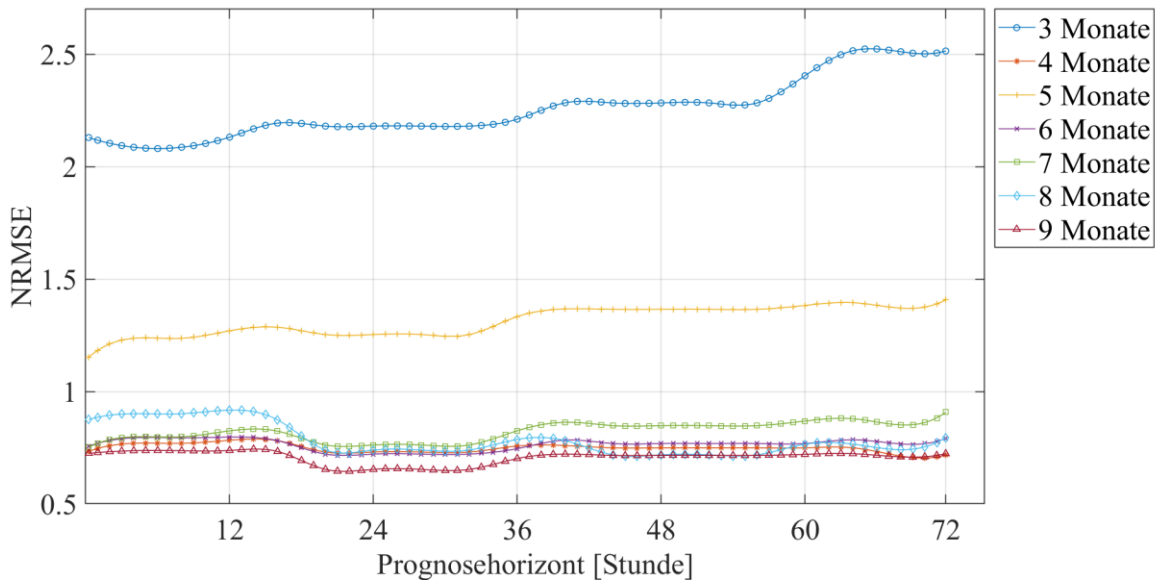


Abb. 4-9: NRMSE der neuronalen Netze mit unterschiedlichen Trainingsdatenmengen

4.4.3 Parametrierung der Kombination eines ARX-Modells mit einem neuronalen Netz

Hier versteht man unter der Kombination eines ARX-Modells mit einem neuronalen Netz (ARX-NN-Modell), dass der exogene Eingang für ein ARX-Modell (τ^{wpp}) durch ein neuronales Netz mit umfangreichen Wetterprognoseparametern modelliert wird. Anders gesagt, wird die lineare Regression (f_{wpp}) in Gleichung 4-4 durch ein neuronales Netz ersetzt. Die Konzeption des ARX-NN-Modells ist in Abschnitt 3.2.3 vorgestellt und die Struktur des ARX-NN-Modells ist in Abb. 3-4 dargestellt.

Für eine Parametrierung der endogenen Eingänge gibt es relativ wenige Spielraum, weil die Auswahl des endogenen Eingangs schon in Abschnitt 4.2.1 festgelegt wird. Der exogene Eingang (τ^{wpp}) wird hier mit dem prognostizierten CS-Faktor eines neuronalen Netzes als Ergebnis modelliert, das dieselben Parameter wie das in Abschnitt 4.4.2 vorgestellte neuronale Netz als Input verwendet. Die Anpassung des ARX-NN-Modells konzentriert sich dann auf die Wahl einer geeigneten Trainingsdatenmenge.

In Abb. 4-10 ist zu sehen, dass die Prognosegüte mit zunehmenden Trainingsdaten eine deutliche Veränderung erfährt. Nachdem der Prognosefehler zunächst auf etwa 0,6 absinkt, während die Trainingsdatenmenge auf 7 Monate zunimmt, steigt der Prognosefehler mit einer 8-monatigen Trainingsdatenmenge dann wieder auf 1,3 an. Für das ARX-NN-Modell kann hier keine allgemeingültige Aussage über die optimale Trainingsdatenmenge getroffen werden, aber es wird aufgrund der Ergebnisse hier im Folgenden eine Trainingszeit von 7 Monaten verwendet.

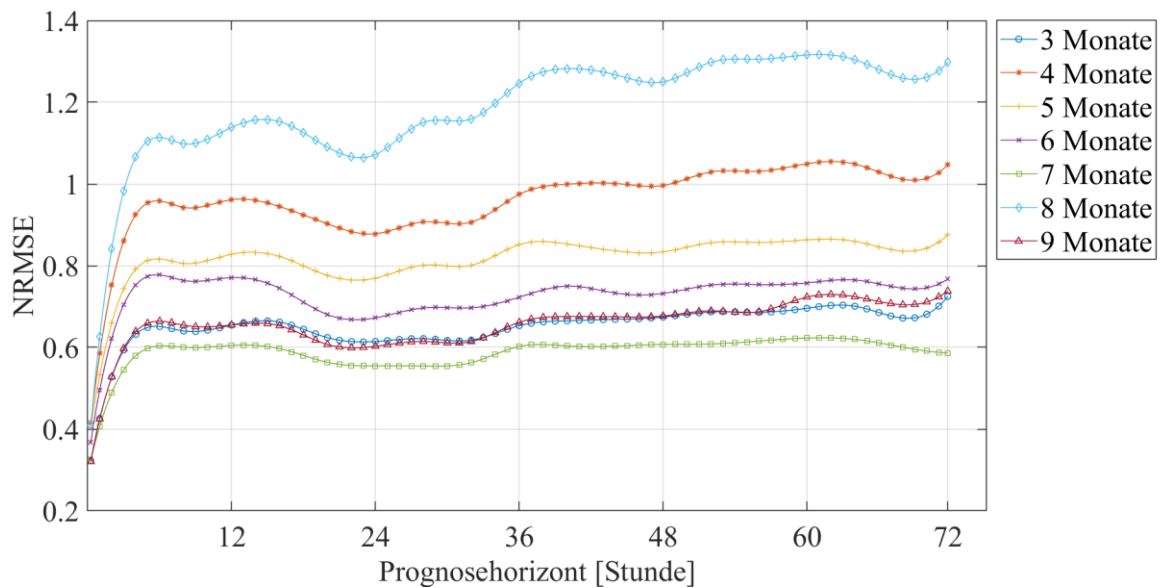


Abb. 4-10: NRMSE der ARX-NN-Modelle mit unterschiedlichen Trainingsdatenmengen

4.4.4 Parametrierung des Random Forests

Ähnlich wie bei dem für das neuronale Netz durchgeführten Parametrierungsprozess werden beim Random Forests die Hyperparameter und die Trainingsdatenmenge näher betrachtet.

Auswahl der Merkmale

Alle in das neuronale Netz eingespeisten Merkmale werden ebenfalls für den Random Forest verwendet. Die Auswahl der Merkmale wurde daher bereits in Abschnitt 4.2.2 ausführlich erläutert. Es ermöglicht zudem eine höhere Vergleichbarkeit der beiden Machine-Learning-Algorithmen, wenn die verwendeten Merkmale gleich sind.

Auswahl des Hyperparameters

Bei Random Forests gibt es zwei wichtige Hyperparameter. Die Anzahl der Entscheidungsbäume (Trees) beschreibt, wie viele Entscheidungsbäume im Random Forest aufgebaut werden. Die Anzahl der Blätter (Leaves) gibt die minimale Anzahl der Datenpunkte an, die ein Blatt im einzelnen Entscheidungsbaum haben muss (Breiman 2017). Zunächst werden alle anderen Hyperparameter des Random Forests als Default in Matlab eingestellt, dann wird die Anzahl der Entscheidungsbäume (Trees) von 100 bis 900 variiert. Die resultierenden Prognosefehler NRMSE sind in Abb. 4-11 dargestellt. Als Optimum wird hier die Anzahl der Entscheidungsbäume von 300 Trees bestimmt. Das Diagramm veranschaulicht aber auch, dass die Prognosegüte nur wenig von der Anzahl der Entscheidungsbäume beeinflusst wird. Weil die Abweichung zwischen allen berechneten

Prognosefehlern sehr gering ist, ist die Prognosegüte relativ unempfindlich gegenüber der Anzahl der Entscheidungsbäume.

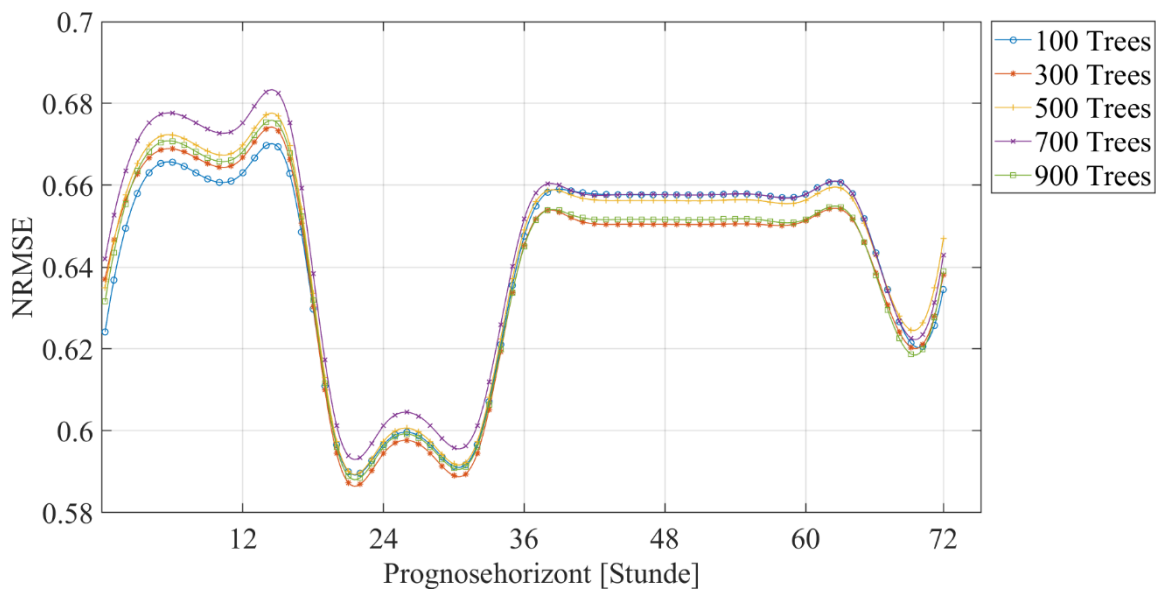


Abb. 4-11: NRMSE der Random Forests mit unterschiedlicher Tree-Anzahl

In ähnlicher Weise wird die optimale Anzahl der Blätter (Leaves) über Abb. 4-12 herausgefunden. Es werden 6 Blätter als optimal angenommen. Aber auch eine Veränderung der Anzahl der Blätter hat nur einen geringen Einfluss auf die Prognosegüte.

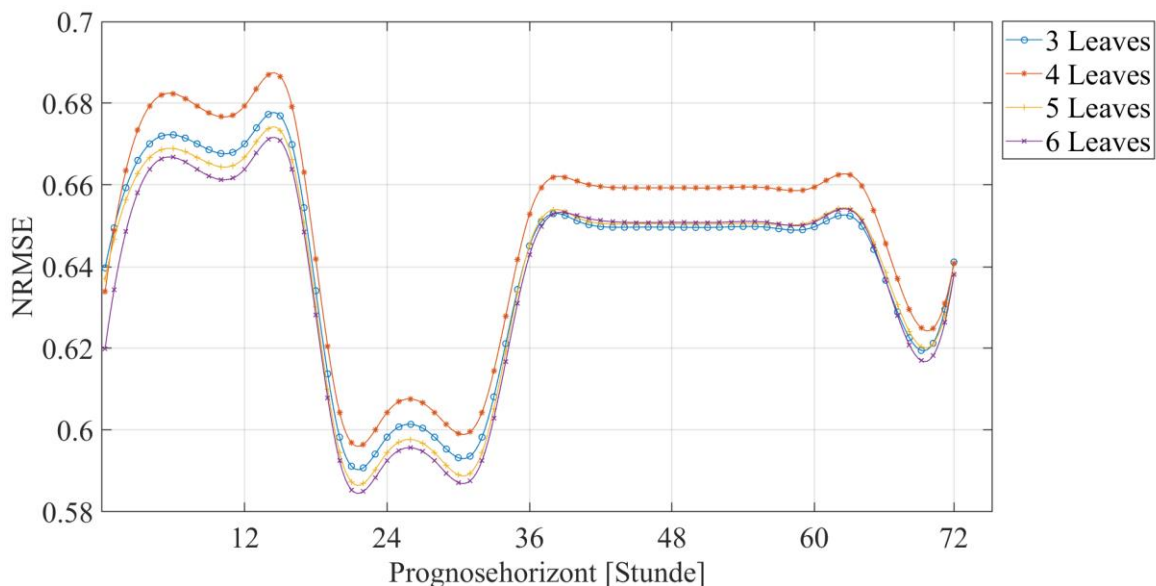


Abb. 4-12: NRMSE der Random Forests mit unterschiedlicher Blattanzahl

Auswahl der Trainingsdatenmenge

Die Prognosefehler (NRMSE) der Random Forests mit unterschiedlichen Trainingsdatenmengen sind in Abb. 4-13 über dem Prognosehorizont von null bis 72 Stunden dargestellt. Der geringste Fehler findet sich bei einer Trainingszeit von 7 Monaten. Für den Random Forest wird die Trainingszeit von 7 Monate als eine optimale Trainingsdatenmenge verwendet. Mit der Veränderung der Trainingsdatenmenge zeigt sich jedoch insgesamt keine starke Veränderung der Prognosegüte. Offensichtlich zeigt sich, dass im Vergleich zum in Abschnitt 4.2.2 untersuchten neuronalen Netz der Random Forest sowohl für die Änderung der Hyperparameter als auch für die Veränderung der Trainingsdatenmenge nicht so empfindlich ist.

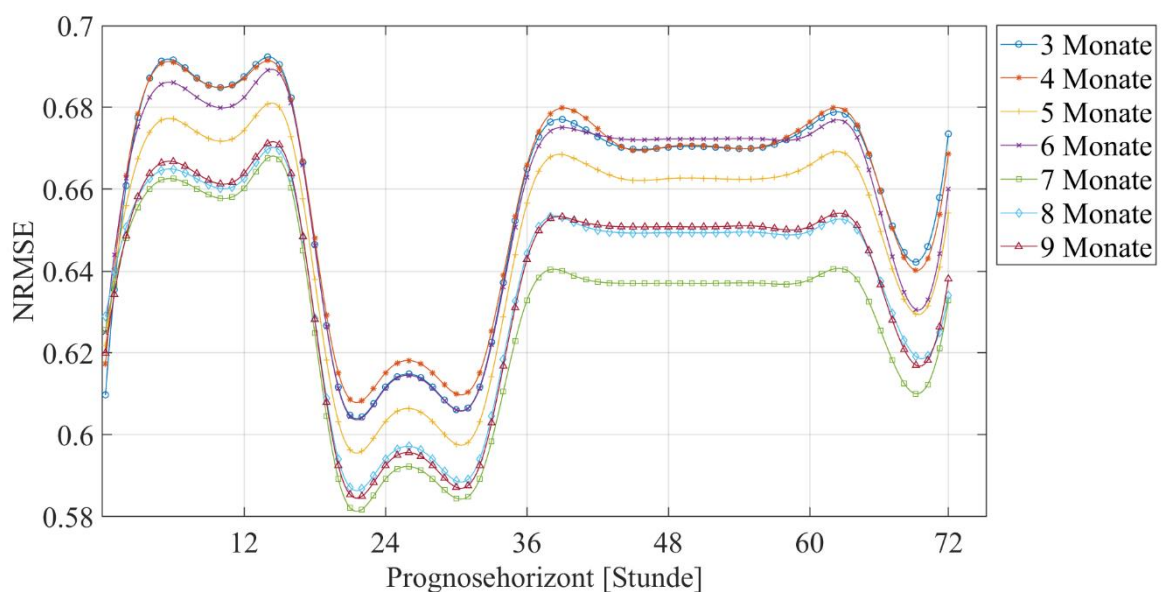


Abb. 4-13: NRMSE der Random Forests mit unterschiedlichen Trainingsdatenmengen

4.4.5 Parametrierung des Clear-Sky-Modells

Das in dieser Arbeit untersuchte Prognosemodell besteht aus zwei Teilen. Die Konzeption des Hybridmodells ist in Abschnitt 3.1 vorgestellt und der Aufbau des Prognosemodells ist in Abb. 3-1 dargestellt. Das aufgebaute Prognosemodell wird als eine Kombination eines Clear-Sky-Modells mit verschiedenen Regressionsmodellen bezeichnet. Nachdem die Parametrierung der Regressionsmodelle durchgeführt wurde, wird hier zusätzlich auch die Auswahl der CS-Modellparameter untersucht. Eine vollständige Parametrisierung des CS-Modells ist in dieser Arbeit nicht im Fokus, daher wird hier lediglich die Bandweite der Standardnormalverteilung über den Tag im Jahr (h_x) als ein besonders relevanter Parameter optimiert. Die Bestimmung der restlichen CS-Modellparameter (z.B. die Bandweite der Standardnormalverteilung über die Uhrzeit am Tag) wird in (Bacher et al. 2009) untersucht.

Zur Optimierung des CS-Modellparameters (h_x) wird hier das neuronale Netz als Regressionsmodell ausgewählt, weil von allen vier in Abschnitt 4.2 untersuchten Regressionsmodellen das neuronale Netz eine hohe Empfindlichkeit auf Veränderungen von Modellparametern besitzt.

Die Prognosefehler (NRMSE) des neuronalen Netzes mit den CS-Modellen mit unterschiedlichen CS-Modellparametern ist in Abb. 4-14 dargestellt. Das Optimum der Bandweite der Standardnormalverteilung über den Tag im Jahr (h_x) liegt hier bei 27.

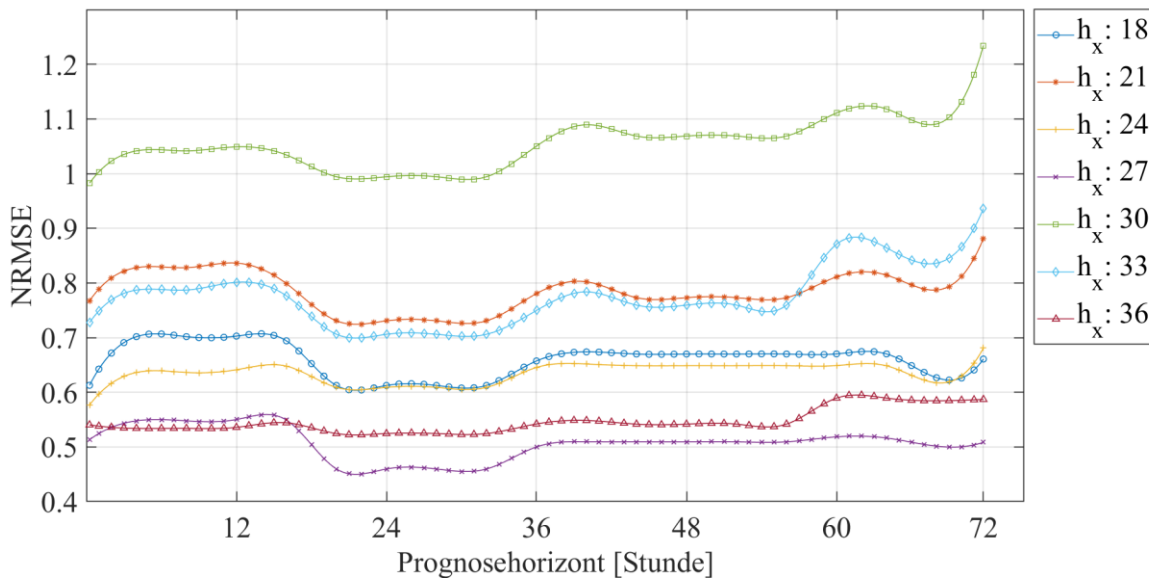


Abb. 4-14: NRMSE der neuronalen Netze mit unterschiedlichen CS-Leistungen

4.4.6 Kritische Auseinandersetzung

Es ist zu bedenken, dass bei der Auswahl der Inputfaktoren für das neuronale Netz auch alternative Möglichkeiten zur Verfügung stehen. Hier wurde vereinfachend die Auswahl der Inputparameter für das neuronale Netz aus (Decker 2018) übernommen, obwohl eine Parameterauswahl idealerweise systematisch über Signifikanztests untersucht werden müsste.

Die in Abschnitt 4.4 durchgeführten Parametrierungsprozesse stellen zudem lediglich eine lokale Optimierung dar. Anders gesagt, werden die Modellparameter nur jeweils einzeln variiert und dann in dieser einen Dimension optimiert.

Aufgrund des zeitlichen Aufwands ist eine globale Optimierung über ein Abtasten des kompletten Parameterraums im Rahmen der vorliegenden Arbeit nicht leistbar. Daher wird hier nicht die gesamte Kombinatorik aller Parameter untersucht.

5 Ergebnisse und Vergleich der Prognosemodelle

Im vorangegangenen Kapitel wurden die Implementierung und Parametrierung der Prognosemodelle dargestellt. Im Folgenden werden die Prognoseergebnisse und ein Modellvergleich betrachtet. Wie in Kapitel 1 bereits erläutert, werden die Prognosemethoden in deterministische und probabilistische Prognosemethoden eingeteilt.

5.1 Rahmenbedingungen

Um die mit den in Kapitel 4 implementierten Modellen berechneten Prognoseergebnisse zu evaluieren, werden die Prognosemodelle mit den Leistungsdaten der Institutsanlage getestet. Die PV-Anlage auf dem Dach des Institutsgebäudes des IER weist eine Nennleistung von 15,3 kW_{peak} auf. Die durch die Institutsanlage erzeugte PV-Leistung wird an den zwei Wechselrichtern gemessen. Die historischen Daten der 15-minütig aufgelösten Mittelwerte der PV-Leistung stehen von heute bis zum 24.10.2017 zur Verfügung. Die historischen Daten der Wetterprognose für den Standort Stuttgart-Vaihingen sind vom 27.10.2017 bis zum 27.10.2018 verfügbar. Danach wird die Wetterprognose mit dem in dieser Arbeit weiter entwickelten Python-Skript wieder seit dem 26.03.2019 gesammelt. Die Tabelle 5-1 gibt eine Übersicht der verfügbaren PV-Leistungswerte und der Wetterprognosen.

Tabelle 5-1: Datenbeschreibung der verfügbaren PV-Leistung und der Wetterprognose

Daten	Auflösung	Zeitraum
PV-Leistung	15-minütig	vom 24.10.2017 bis zum aktuellen Zeitpunkt
Wetterprognose	stündlich	vom 27.10.2017 bis zum 27.10.2018 und vom 26.03 bis zum aktuellen Zeitpunkt

Nachdem die verfügbaren Inputdaten zum Modellaufbau dargestellt wurden, lenkt sich nun die Aufmerksamkeit auf die Anforderungen an die Prognoseergebnisse. Die Anforderungen an die zu erzielenden Prognoseergebnisse, nämlich die Ausgänge der Prognosemodelle, wurden zu Beginn dieser Arbeit bereits angegeben und werden hierin Tabelle 5-2 nochmals zusammengefasst dargestellt.

Tabelle 5-2: Übersicht der Prognoseanforderungen

Rahmenbedingung	Anforderung
Prognosehorizont	72 Stunden im Voraus
Prognoseauflösung	15-minütig

Wenn die Anforderungen an die Prognoseergebnisse mit den verfügbaren Inputdaten verglichen werden, ist zu erkennen, dass die Anforderung hinsichtlich der 15-minütigen Prognoseauflösung strenggenommen nicht erfüllt werden kann, weil die Wetterprognose nur in stündlicher Auflösung verfügbar ist. Um die stündliche Wetterprognose mit der 15-minütigen PV-Leistung zu synchronisieren, werden die Wetterprognosedaten durch lineare Interpolation in viertelstündlicher Auflösung neu gesampelt. Nachdem die PV-Leistung und die Wetterprognosen in zeitliche Übereinstimmung gebracht wurden, sind die Voraussetzungen für eine 15-minütige Prognoseauflösung geschaffen. Für Zeitreihenmodelle (z.B. ARX-Modell) ist zudem eine vollständige Zeitreihe ohne Lücken erforderlich. Daher wird auf Zeiträume mit lückenhaften Daten verzichtet. Aus diesen Gründen werden die Trainingsdaten und Testdaten wie in Tabelle 5-3 angegeben ausgewählt.

Tabelle 5-3: Auswahl der Trainingsdaten und Testdaten

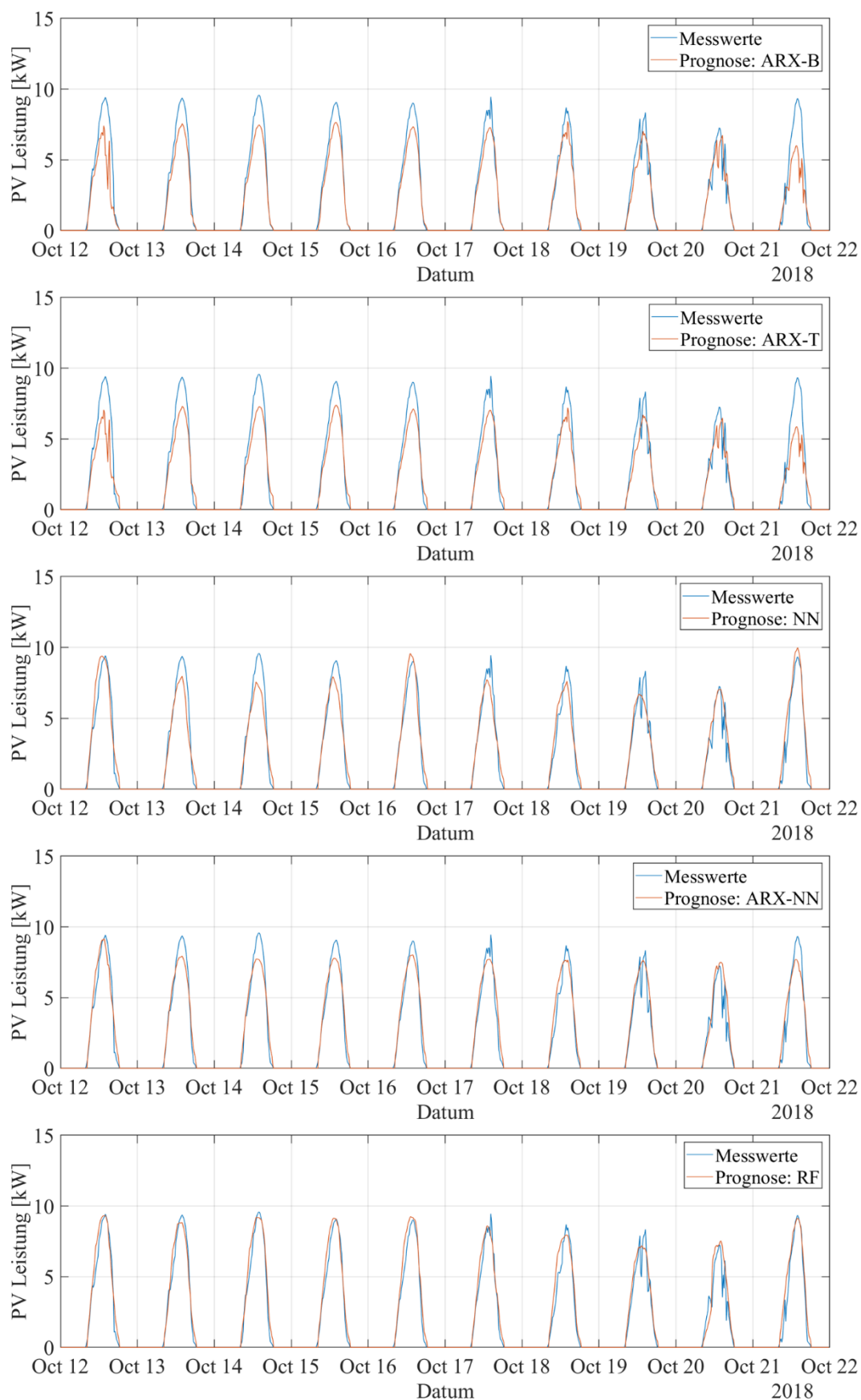
Datensatz	Zeitraum
Trainingsdaten	vom 01.01.2018 bis zum 30.09.2018
Testdaten	vom 01.10.2018 bis zum 27.10.2018

5.2 Deterministische Prognosen

Bei der deterministischen Prognose wird für jeden Prognosezeitpunkt eine konkrete Punktprognose der PV-Leistung getroffen. Die deterministischen Prognoseergebnisse der in Kapitel 4 implementierten Prognosemodelle sind für einen exemplarischen Zeitraum von einigen Tagen in Abb. 5-1 visuell dargestellt.

Das Diagramm veranschaulicht sowohl die tatsächliche PV-Leistung als auch die prognostizierte PV-Leistung in einem Teilzeitraum des Testdatensatzes (vom 12.10.2018 bis zum 21.10.2018). Es handelt sich hier um die Punktprognosen der PV-Leistung, die mit dem ARX-T-Modell, ARX-B-Modell, NN-Modell, ARX-NN-Modell und RF-Modell getroffen werden.

Es ist in Abb. 5-1 zu erkennen, dass die Prognosewerte des ARX-T-Modells und des ARX-B-Modells im dargestellten Zeitraum häufig niedriger als die Messwerte sind (systematische Unterschätzung). Die Ergebnisse der anderen drei Modellen weisen visuell soweit eine grobe Übereinstimmung der beiden Linien von Prognosewerten und Messwerten der PV-Leistung auf. Insgesamt fällt es allerdings allen Prognosemodellen schwer, plötzliche Veränderungen der tatsächlichen PV-Leistung präzise vorherzusagen (z.B. am 20.10.2018). Eine vergleichende Bewertung der Prognosequalität der verschiedenen Modelle ist mithilfe dieser visuellen Darstellung nicht möglich. Hierfür werden Fehlermaße verwendet, wie im folgenden Kapitel beschrieben.

**Abb. 5-1:** Messung und Punktprognosen der PV-Leistung

5.2.1 Einfluss der Fehlermaße auf Prognoseergebnisse

Die Auswertung der Prognoseergebnisse und der bewertende Modellvergleich hängen zunächst von der Auswahl eines Fehlermaßes ab.

Zunächst werden hier alle Prognosemodelle anhand des Fehlermaßes NRMSE miteinander verglichen. Alle NRMSE der in Kapitel 4 hergeleiteten Modelle sind in Abb. 5-2 über dem Prognosehorizont von null bis 72 Stunden dargestellt.

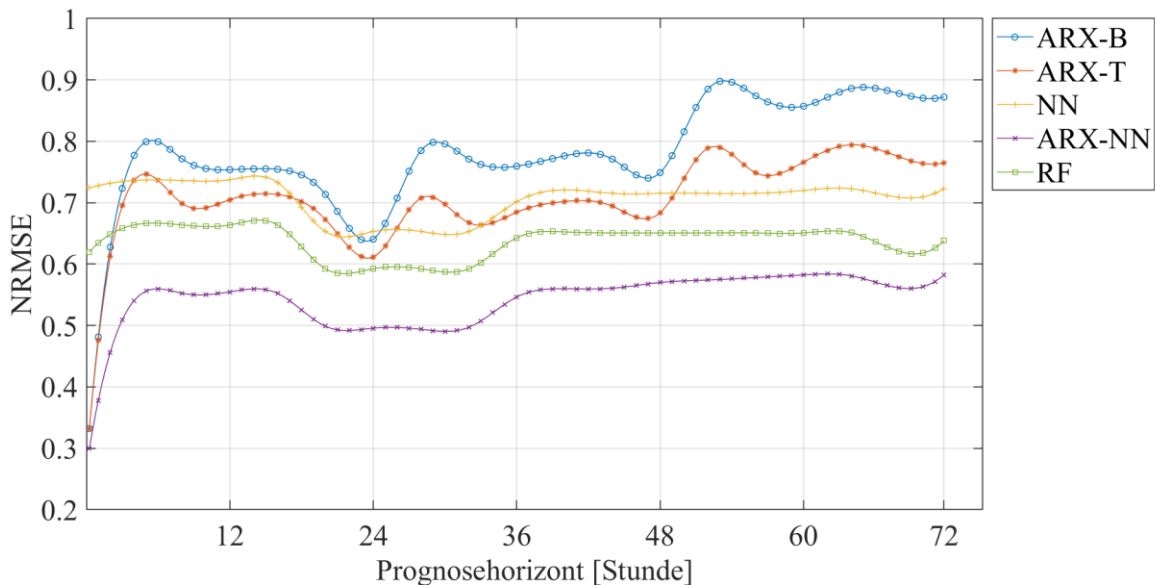


Abb. 5-2: NRMSE der verschiedenen Prognosemodelle

Im Fall der Varianten des ARX-Modells (ARX-T, ARX-B und ARX-NN) zeigt sich ein sehr niedriger Fehler für sehr kurze Prognosehorizonte und ein starker Anstieg des Fehlers in den ersten 5 Stunden des Prognosehorizonts. Dieser Verlauf ist hauptsächlich darauf zurückzuführen, dass bei der Autoregression für die Leistungsprognose der zukünftigen Stunden Prognosen der vorangegangenen Stunden als Input herangezogen werden müssen. Für kurzfristige Prognosen innerhalb von 24 Stunden werden die aktuelle Leistung und die 24 Stunden zuvor erzeugte und gemessene Leistung berücksichtigt. Für mittelfristige und langfristige Prognosen innerhalb von 72 Stunden werden ebenfalls die aktuelle Leistung und die 1 Tag vorher erzeugte historische Leistung verwendet. Dabei haben die beiden Leistungen allerdings einen unterschiedlichen Einfluss auf die kurzfristige und langfristige Prognose. Anders gesagt, haben die beiden Werte positivere Auswirkungen auf die kurzfristige Prognose. Daher ist ARX-Modell eher geeignet für eine kurzfristige Prognose. Wenn nun die ARX-T- und ARX-B-Modelle mit den zwei anhand von Machine-Learning-Methoden aufgebauten Modellen (NN- und RF-Modell) verglichen werden, dann ist deutlich zu sehen, dass die beiden ARX-Modelle weniger genau sind. Über den Prognosehorizont zeigt das Fehlermaß für RF- und NN-Modell dauerhaft einen relativ

flachen Verlauf. Die kleine unstabile Schwankung wird vermutlich durch den Fehler der Wetterprognose verursacht. Dieser Einfluss des Fehlers der Wetterprognosen kann nur quantitativ festgestellt werden, wenn die Prognosemodelle mit tatsächlichen Messwerten von Wetterdaten im Vergleich zu den Daten der Wetterprognose für denselben Zeitraum und dieselbe Anlage validiert werden.

Insgesamt zeigt sich der niedrigste Prognosefehler beim ARX-NN-Modell. Die Kombination eines ARX-Modells mit einem neuronalem Netz vereinigt offenbar die Vorteile von ARX- und NN-Modell. Die Eignung für kurzfristige Prognose stammt vom ARX-Modell. Für eine langfristige Prognose wird die durch das NN-Modell gelieferte Information ausgenutzt, so dass sich die Prognosequalität für den langfristigen Horizont auch verbessern kann. Dieses Ergebnis des Modellvergleichs gilt auf der Basis des verwendeten Prognosemaßes NRMSE.

Für die Betrachtung weiterer Fehlermaße wird hier das Improvement als ein Mittel zum Modellvergleich auf Basis mehrere Fehlermaße verwendet. Die Definition des Improvements befindet in Gleichung 2-26. Es beschreibt, wie ein Modell bezogen auf ein Referenzmodell abschneidet. Ein positives Improvement bedeutet, dass das entsprechende Modell besser als das Referenzmodell abschneidet. Hier wird das ARX-T-Modell als ein Referenz angesehen. Alle Improvements der Prognosemodelle bezogen auf das ARX-T-Modell für Verwendung von den Fehlermaßen NRMSE, MAE und MAPE sind in Abb. 5-3 gezeigt.

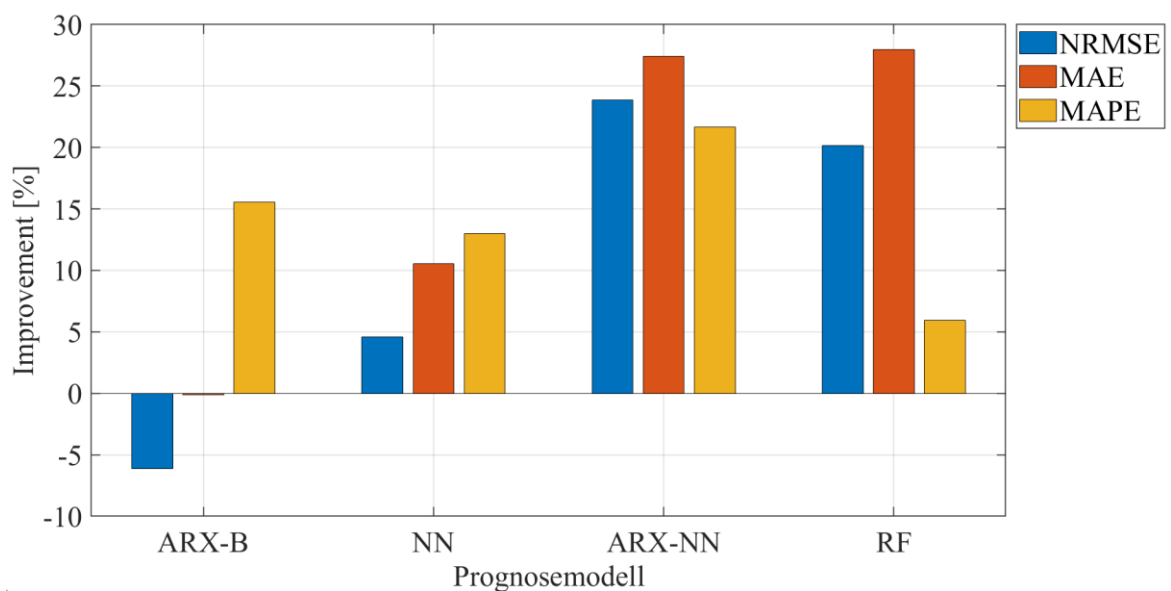


Abb. 5-3: Improvement der Prognosemodelle bezogen auf ARX-T-Modell mit Fehlermaßen NRMSE, MAE, MAPE

Das Diagramm in Abb. 5-3 verdeutlicht, dass die Modellperformance je nach ausgewähltem Fehlermaß variiert. Wenn der NRMSE als Fehlermaß ausgewählt wird, um die

Prognosemodelle miteinander zu vergleichen, steht die Modellleistung vom ARX-NN-Modell an erster Stelle. Danach folgen das RF-Modell und das NN-Modell. Am Ende stehen das ARX-T-Modell und ARX-B-Modell. Das Ergebnis entspricht dem Ergebnis in Abb. 5-2. Wenn nun aber der MAE als Fehlermaß verwendet wird, ergibt sich die höchste Modellperformance vom RF-Modell. Bei Auswahl des Fehlermaßes MAPE tritt die höchste Prognosegenauigkeit auch bei dem ARX-NN-Modell auf.

Dieses Diagramm verdeutlicht, dass das Prognoseergebnis und die Modellperformance auch vom verwendeten Fehlermaß abhängig sind. Wenn NRMSE oder MAPE als Fehlermaß verwendet werden, ist in diesem Fall die Anwendung des ARX-NN-Modells zu empfehlen.

5.2.2 Einfluss der Trainingsdatenmenge auf die Prognosequalität

Die Trainingsdatenmenge wird hier als eine weitere Einflussgröße betrachtet. Die Ergebnisse aus Abschnitt 5.2.1 setzen voraus, dass eine ausreichend große Menge an Trainingsdaten vorhanden ist. Jedoch muss berücksichtigt werden, dass in manchen nicht so idealen Situationen Trainingsdaten nur von einem kleineren Zeitraum vorhanden sind. In diesem Fall muss der Einfluss der Trainingsdatenmenge auf die Prognoseergebnisse für die Modellwahl berücksichtigt werden. Anders gesagt, wird hier die Empfindlichkeit der Prognosemodelle auf einen kleinen Datensatz getestet. Als ein Beispiel wird ein Trainingsdatensatz von 5 Monaten (vom 01.05.2018 bis zum 30.09.2018) ausgewählt. Und der in Abschnitt 5.1 festgelegte Testzeitraum wird hier unverändert beibehalten. Der berechnete Prognosefehler (NRMSE) der entsprechenden Prognosemodelle mit beschränkten Trainingsdaten (5 Monate) ist in Abb. 5-4 dargestellt. Das Ergebnis in Abb. 5-4 mit beschränkten Trainingsdaten unterscheidet sich signifikant vom in Abb. 5-2 gezeigten Ergebnis. Der größte Prognosefehler wird vom NN-Modell begangen. Im Mittelfeld liegen die Variationen des ARX-Modells. Bei einem kurzfristigen Prognosehorizont schneiden diese ebenfalls wieder gut ab. Während des gesamten Zeithorizonts ist zu sehen, dass der niedrigste Prognosefehler (NRMSE) vom RF-Modell erreicht wird. Durch dieses Diagramm wird die unterschiedliche Empfindlichkeit der Prognosemodelle hinsichtlich der Trainingsdatenmenge verdeutlicht. Beim NN-Modell wird die Modellleistung durch die Verringerung der Trainingsdatenmenge deutlich verschlechtert. Aufgrund der Eigenschaften der klassischen Zeitreihenmodelle wird die Modellperformance der Variationen des ARX-Modells relativ wenig von der Trainingsdatenmenge beeinflusst. Dafür spielt für die Zeitreihenmodelle die Kontinuität der Zeitreihe eine wichtigere Rolle. Für das RF-Modell ergibt die Variation der Trainingsdatenmenge den schwächsten Einfluss auf die Prognoseergebnisse. Zusammenfassend gesagt, hat eine Veränderung der Trainingsdatenmenge den stärksten Einfluss auf das NN-Modell. Im Gegensatz dazu verhält sich das RF-Modell kaum sensitiv

auf eine Veränderung der Größe des Trainingsdatensatzes. Deswegen wird hier empfohlen, für Anwendungsfälle mit relativ wenigen verfügbaren Daten das RF-Modell zu verwenden.

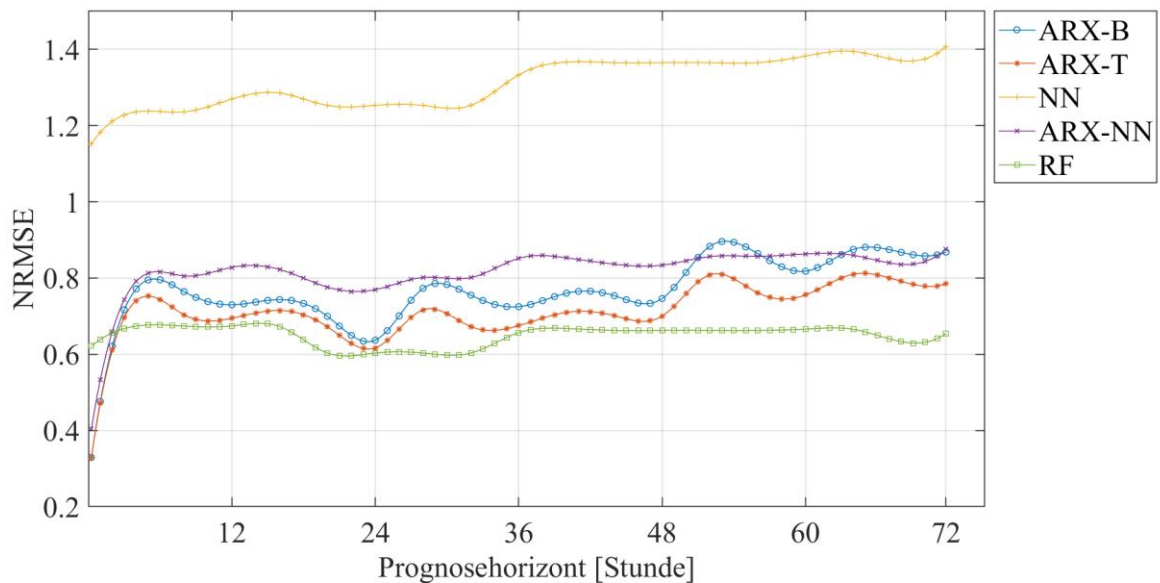


Abb. 5-4: NRMSE der Prognosemodelle mit beschränkten Trainingsdaten (5 Monate)

5.3 Probabilistische Prognoseergebnisse

Nachdem in den vergangenen Abschnitten die deterministische Prognose beschrieben wurde, soll der folgende Abschnitt einen Überblick über probabilistische Prognosen geben, die das Ziel haben, ein Prognoseintervall vorherzusagen.

In dieser Arbeit wird keine Annahme über die Form der Verteilungsfunktion getroffen. Deswegen werden probabilistische Prognosen mithilfe von nichtparametrischen Methoden getroffen. Wie in Abschnitt 2.4 und 2.5 beschrieben, werden hier die lineare Quantils-Regression und der Quantile Regression Forest als nichtparametrische Methoden benutzt, um probabilistische Modelle aufzubauen. Alle Varianten des ARX-Modells kombiniert mit der linearen Quantilsregression sind in der Lage, probabilistische Prognose zu treffen. Außerdem ist der Random Forest auch dazu in der Lage, Prognoseintervalle vorherzusagen. Daher werden in dieser Arbeit vier Prognosemodelle für probabilistische Prognosen untersucht, nämlich das ARX-T-Modell, das ARX-B-Modell, das ARX-NN-Modell und das RF-Modell.

Die Messwerte und die probabilistischen Prognosen der Prognosemodelle im exemplarischen Zeitraum vom 18.10.2018 bis zum 20.10.2018 sind in Abb. 5-5 dargestellt. Beim ARX-T- und ARX-B-Modell wird tendenzielle große Bereiche zwischen beiden Grenzwerten des Prognoseintervalls prognostiziert. Beispielsweise umfasst das 90%-Prognoseintervall des ARX-B-Modells zum Zeitpunkt 12: 00 am 18.10.2018 den

Leistungsbereich von 0 kW bis 18 kW. Obwohl die Messwerte innerhalb des Prognoseintervalls liegen, hat die Prognose eine sehr geringe Aussagekraft. Die probabilistischen Prognoseergebnisse des ARX-NN- und RF-Modells sind kompakter und transportieren daher mehr Informationen. Die Prognosequalität lässt sich anhand von probabilistischen Fehlermaßen wie dem Winkler Score quantitativ vergleichen.

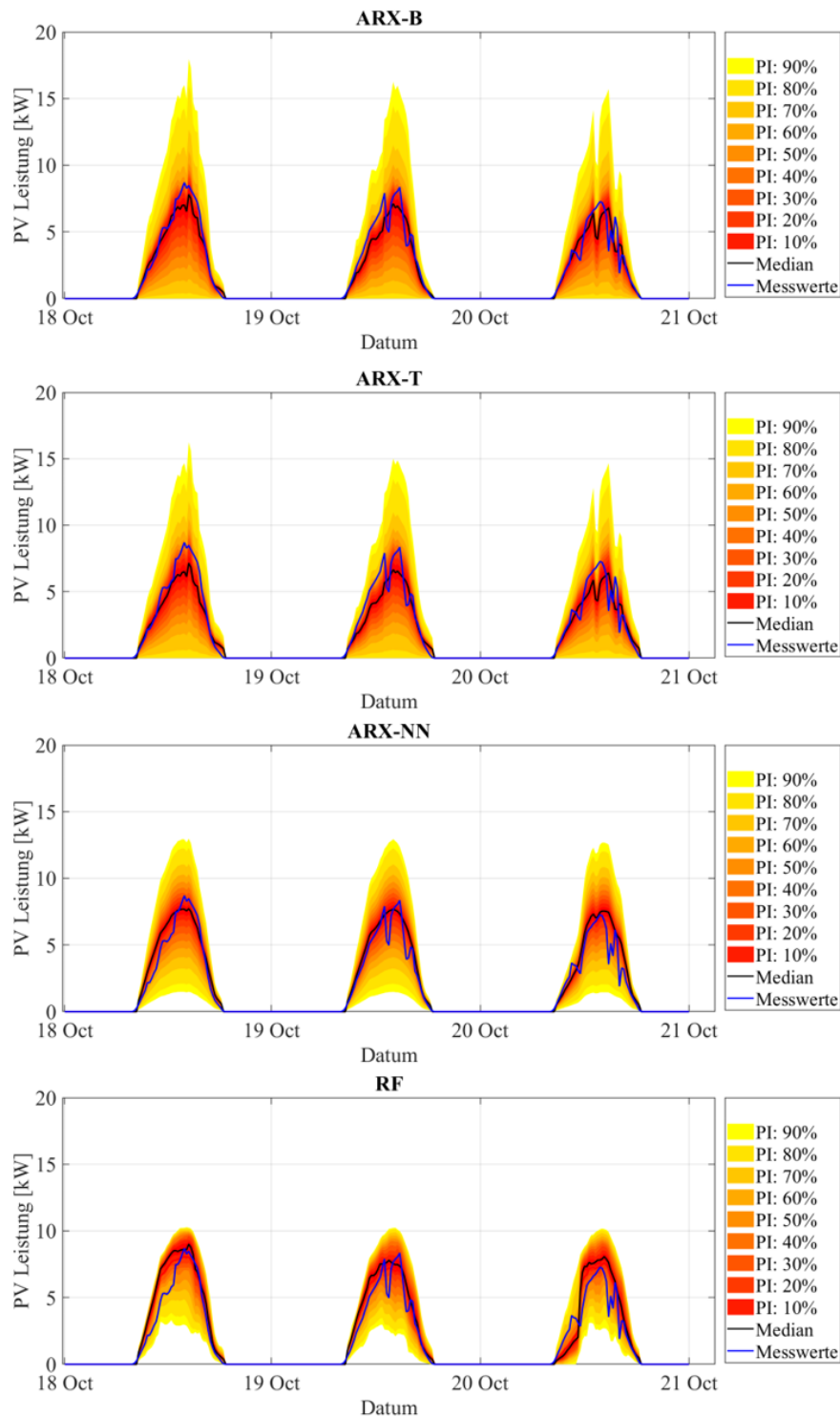


Abb. 5-5: Messung und probabilistische Prognosen der PV-Leistung

Ein quantitativer Modellvergleich zwischen allen Prognosemodelle wird mit dem in Geichung 2-24 vorgestellten Fehlermaß Winkler Score durchgeführt. Ein niedriges Winkler Score entspricht einem kleinem Prognosefehler. Der Winkler Score in Abb. 5-6 verdeutlicht, dass die probabilistischen Prognosen der ARX-T- und ARX-B-Modelle weniger befriedigend sind. In den 10%- bis 20%-Prognoseintervallen zeigt der Winkler Score zwischen ARX-NN- und RF-Modell nur geringe Unterschiede. Danach geht der Winkler Score des ARX-NN-Modells mit zunehmenden Prognoseintervall immer weiter hoch. Insgesamt wird der niedrigste Prognosefehler über dem Prognoseintervall von 10% bis 90% durch das RF-Modell erzielt. Demnach empfiehlt es sich, für probabilistische Prognosen das RF-Modell zu verwenden.

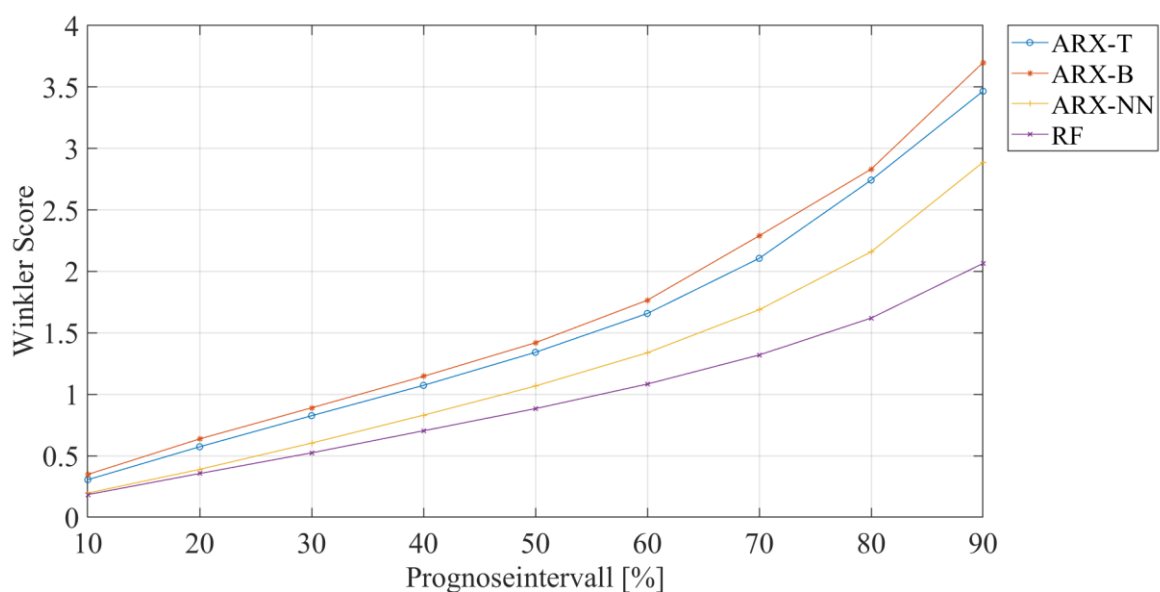


Abb. 5-6: Winkler Score der Prognosemodelle

6 Zusammenfassung und Ausblick

Dieses Kapitel gibt einen Überblick über Inhalte und Ergebnisse dieser Forschungsarbeit. Die in den vergangenen Kapiteln gewonnenen Erkenntnisse werden hier übersichtlich zusammengefasst. Schließend wird ein Ausblick auf weiterführende Forschungsrichtungen für kommende Arbeiten gegeben.

6.1 Zusammenfassung

Ziel dieser Arbeit ist das Erstellen eines echtzeitfähigen Prognosedienstes für die Leistung einer PV-Anlage. Es werden hierfür zum einen die aktuellen Leistungsdaten der PV-Anlage auf dem Dach des IER kontinuierlich ausgelesen und zum anderen Wetterprognosen für den Standort Stuttgart-Vaihingen regelmäßig abgerufen. Um die Echtzeitfähigkeit umzusetzen, wird das Einlesen der Daten durch zwei in Python implementierte Skripte durchgeführt und die Daten werden in einer SQLite-Datenbank gespeichert und für die weiteren Schritte zur Verfügung gestellt. Um die eigentliche Prognose zu erstellen, werden vier Prognosemodelle in Matlab implementiert und anhand von historischen Wetterprognose- und Leistungsdaten trainiert, parametrisiert und evaluiert. Alle Prognosemodelle sind als Hybrid-Modelle konzipiert, die eine Zusammensetzung aus einem Clear-Sky-Modell und einem Regressionsmodell darstellen. Mit dem Clear-Sky-Modell werden die historisch gemessenen absoluten PV-Leistungswerte zuerst auf die damit ermittelten Clear-Sky-Leistungen der jeweiligen Zeitpunkte normalisiert. Die dadurch berechneten CS-Faktoren der historischen Messwerte werden dann als Eingangsdaten für die Regressionsmodelle verwendet. Zudem fließt die Wetterprognose für den Prognosezeitpunkt in die Regressionsmodelle ein. Als Zielgröße der Regressionsmodelle wird der CS-Faktor des Prognosezeitpunktes berechnet. Durch die Multiplikation mit der CS-Leistung wird der CS-Faktor wieder in die absolute prognostizierte PV-Leistung umgerechnet. Auf diese Weise wird nicht nur eine deterministische Prognose erstellt, sondern es werden in der Arbeit über entsprechende Modelle auch probabilistische Prognosen für die PV-Leistung getroffen.

Das Clear-Sky-Modell basiert auf einer statistischen, selbstadaptiven Methode, so dass seine Parameter mit der Zunahme der historischen Datenmenge der PV-Leistung aktualisiert werden können, wodurch auch sich verändernde Umgebungseinflüsse berücksichtigt werden. Das Modell lag zu Beginn der Arbeit bereits implementiert vor und wird im Zuge der Arbeit hinsichtlich der gewählten Parameter noch leicht angepasst.

Als Regressionsmodelle werden in der Arbeit vier Modelle untersucht. Nach der Implementierung und Parametrisierung der Modelle werden die Prognosemodelle für den Anwendungsfall der Institutsanlage getestet und ausgewertet. Für die Bewertung der Prognosequalität der deterministischen Prognosen wird zumeist der NRMSE als Fehlermaß verwendet, für die Bewertung der probabilistischen Prognosen der Winkler Score. Die

wichtigsten Erkenntnisse werden über die vier Modelle hier zusammengefasst: Das autoregressive Modell mit exogenen Eingang (ARX-Modell) gehört zu den klassischen Zeitreihenmodellen. Es zeigt sich, dass aufgrund der Eigenschaften der Zeitreihenmodelle vom ARX-Modell häufig eine relativ genaue kurzfristige Prognose für einen Prognosehorizont von maximal 3 Stunden erreicht wird. Aber mit steigendem Prognosehorizont nimmt die Prognosequalität des ARX-Modells ab. Als zweites Regressionsmodell wird ein neuronales Netz (NN-Modell) untersucht. Das verwendete Feed-Forward-Netz kann die umfangreichen Informationen der Wetterprognosen ausnutzen und präsentiert keine signifikanten Unterschiede der Prognosegüte über den gesamten Prognosehorizont. Außerdem zeigt sich, dass das neuronale Netz sehr empfindlich auf eine Veränderung der Trainingsdatenmenge reagiert, was bedeutet, dass eine Vergrößerung des Trainingsdatensatzes zu einer deutlichen Erhöhung der Modellleistung führt. Ein weiteres Regressionsmodell stellt die Kombination eines ARX-Modells mit einem neuronalen Netz (ARX-NN-Modell) dar, welches aufgrund seiner Eigenschaften die Vorteile des ARX- und des NN-Modells vereinen kann. Die hohe Genauigkeit seiner kurzfristigen Prognose entsteht hauptsächlich aufgrund der vergangenen Leistungsdaten, die durch das ARX-Modell modelliert werden können. Für die langfristige Prognose spielt der Wettereinfluss eine wichtigere Rolle, der durch das neuronale Netz genutzt werden kann. Deswegen wird häufig die präziseste Prognose sowohl für den kurzfristigen als auch für den längerfristigen Zeithorizont vom ARX-NN-Modell getroffen. Als letztes Modell wird hier der Random Forest (RF-Modell) angewendet, welcher eine sehr überlegene Modellperformance zeigt, wenn der verfügbare Datensatz auf relativ wenige Trainingsdaten begrenzt ist. Auch hinsichtlich der probabilistischen Prognosen erzeugt der Random Forest die präzisesten Prognosen. Insgesamt zeigt sich, dass die Modellleistungen nicht nur mit der Veränderung der verfügbaren Datenmenge variieren, sondern dass die Prognosemodelle auch je nach der Auswahl des für die Bewertung genutzten Fehlermaßes unterschiedlich abschneiden können. Ein quantitativer Modellvergleich ist daher nur unter Berücksichtigung der gegebenen Rahmenbedingungen möglich. Für eine deterministische Prognose erweist sich für den untersuchten Anwendungsfall die Verwendung des ARX-NN-Modells als vorteilhaft. Außerdem wird für probabilistische Prognosen des untersuchten Falls das RF-Modell empfohlen. Alle Modelle sind in dem in der Arbeit erstellten Prognosedienst enthalten, der die Leistung der PV-Anlage auf dem IER-Institutsdach in Echtzeit prognostiziert.

6.2 Kritik und Ausblick

Bei der Auswahl der Inputdaten für das ARX-Modell wird in dieser Arbeit eine Korrelationsanalyse zwischen der PV-Leistung und den Wetterprognosedaten durchgeführt. Es wäre aber auch sinnvoll, vor dem Modellaufbau die Korrelation mit dem CS-Faktor zu

analysieren, um möglicherweise eine aussagekräftigere Auswahl des exogenen Eingangs darstellen zu können und ggf. eine Erhöhung der Modellperformance erzielen zu können. Für das NN-Modell gibt es ebenfalls noch Spielraum bei der Auswahl der Inputparameter, die weiter zu untersuchen wäre. Weil in dieser Arbeit vereinfachend die Auswahl der Inputmerkmale für ein neuronales Netz von (Decker 2018) übernommen werden, werden hier nicht wie sonst üblich statistische Signifikanztests für die Parameterwahl durchgeführt. Aufgrund der in dieser Arbeit ermittelten Ergebnisse schneidet das NN-Modell nur mittelmäßig ab. Daher empfiehlt es sich, systematische Analysen bzgl. der Inputparameter für das neuronale Netz in weitergehenden Arbeiten durchzuführen, so dass die Auswahl der Inputmerkmale optimiert und die erwartete Modellleistung eines neuronalen Netzes erreicht werden könnte.

Außer der systematischen Parameterauswahl wäre eine globale Parameteroptimierung für die vorhandenen Prognosemodelle auch sinnvoll. Alle in Abschnitt 4.4 durchgeführten Parametrierungsprozessen fokussieren zunächst auf die lokale Optimierung. Daher könnte eine vollständige Parameteroptimierung noch weitergetrieben werden.

Zusätzlich könnte die entsprechende Funktion zur Modelloptimierung so vervollständigt werden, dass sie in bestimmten zeitlichen Abständen selbstständig die Modelle mit den aktuellen Daten neu parametriert und immer wieder das ‚beste‘ Modell aussucht.

Aufgrund des maßgeblichen Einflusses des gewählten Fehlermaßes auf die Bewertung der Prognoseergebnisse wird ein allgemeingültiger Modellvergleich erschwert. Daher wäre es eine interessante Forschungsrichtung, den Einfluss der Fehlermaßwahl auf die Bewertung der Prognoseergebnisse hinsichtlich der Prognoseanwendung weiter zu untersuchen.

Darüber hinaus ließe sich ein genauerer Blick auf Ensemble-Methode werfen, die häufig zum Einsatz kommen, wenn bereits einige Prognosemodelle aufgebaut sind, welche sich dann zu einem noch präziseren Vorhersagemodell vereinen lassen. Bisher funktionieren alle in dieser Arbeit aufgebauten Prognosemodelle unabhängig voneinander, aber mithilfe einer Ensemble-Methode könnten die einzelnen Prognosen in einer gemeinsamen Vorhersage zusammengefasst werden, um ein noch präziseres Prognoseergebnis zu erzielen.

Schließlich wäre eine Erweiterung des echtzeitfähigen Prognosedienstes um die Prognose der lokalen elektrischen und thermischen Last sinnvoll, wenn der Prognosedienst einer intelligenten Gebäudeenergiesystem-Steuerung dienen soll.

Literaturverzeichnis

Bacher, Peder (2008): Short-term Solar Power Forecasting. Masterarbeit. Technical University of Denmark, Lyngby, Dänemark. Informatics and Mathematical Modelling.

Bacher, Peder; Madsen, Henrik; Nielsen, Henrik Aalborg (2009): Online short-term solar power forecasting. In: *Solar Energy* 83 (10), S. 1772–1783. DOI: 10.1016/j.solener.2009.05.016.

Box, George E. P.; Jenkins, Gwilym M.; Reinsel, Gregory C. (2008): Time series analysis. Forecasting and control. Fourth edition. Hoboken, New Jersey: Wiley a John Wiley & Sons Inc (Wiley series in probability and statistics). Online verfügbar unter <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10713671>.

Breiman, Leo (1996): Bagging predictors. In: *Machine Learning* 24 (2), S. 123–140. DOI: 10.1007/BF00058655.

Breiman, Leo (2001): Random Forests. In: *Machine Learning* 45 (1), S. 5–32. DOI: 10.1023/A:1010933404324.

Breiman, Leo; Friedman, Jerome H.; Olshen, Richard A.; Stone, Charles J. (2017): Classification And Regression Trees: Routledge.

Decker, Julian (2018): Prognose lokaler PV-Erzeugung unter Anwendung von Clear-Sky-Modellen und neuronalen Netzen. Studienarbeit. Universität Stuttgart, Stuttgart, Deutschland. Institut für Energiewirtschaft und Rationelle Energieanwendung.

Grossberg, Stephen (1988): Nonlinear neural networks: Principles, mechanisms, and architectures. In: *Neural Networks* 1 (1), S. 17–61. DOI: 10.1016/0893-6080(88)90021-4.

Hagan, M. T.; Menhaj, M. B. (1994): Training feedforward networks with the Marquardt algorithm. In: *IEEE transactions on neural networks* 5 (6), S. 989–993. DOI: 10.1109/72.329697.

Hong, Tao; Fan, Shu (2016): Probabilistic electric load forecasting: A tutorial review. In: *International Journal of Forecasting* 32 (3), S. 914–938. DOI: 10.1016/j.ijforecast.2015.11.011.

Koenker, Roger; Bassett, Gilbert (1978): Regression Quantiles. In: *Econometrica* 46 (1), S. 33. DOI: 10.2307/1913643.

Kruse, Rudolf; Borgelt, Christian; Braune, Christian; Klawonn, Frank; Moewes, Christian; Steinbrecher, Matthias (2015): Computational Intelligence. Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze. 2., überarb. und erw. Aufl. 2015. Wiesbaden: Springer Fachmedien Wiesbaden GmbH (Computational Intelligence).

Lauret, Philippe; David, Mathieu; Pedro, Hugo (2017): Probabilistic Solar Forecasting Using Quantile Regression Models. In: *Energies* 10 (10), S. 1591. DOI: 10.3390/en10101591.

Raschka, Sebastian (2018): Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow. Konzepte, Tools und Techniken für intelligente Systeme. 1. Auflage. Heidelberg: O'Reilly; dpunkt.verlag.

Anhang

A Erfassung der Wetterprognosen

Die Wetter-Prognosedaten für den Standort Stuttgart-Vaihingen werden von der Website www.wetter.com heruntergeladen. Dort sind die stündlichen aufgelösten Prognosen der Temperatur, des Bewölkungsgrades, der relativen Feuchtigkeit, der Windgeschwindigkeit, der Windrichtung, des Niederschlagsrisikos, der Niederschlagsmenge vom aktuellen Zeitpunkt bis über die nächsten 3 Tage verfügbar.

Der in dieser Arbeit erstellte Prognosedienst trifft jede Stunde eine Prognose der PV-Leistung für die nächsten 72 Stunden. Entsprechend wird die Wetterprognose jede Stunde einmal heruntergeladen und gespeichert. Das stündlich Auslesen der Wetterprognose wird mit einer Screen-Scraping-Bibliothek (BeautifulSoup) in Python automatisiert. Alle benötigten Wetter-Prognosedaten befinden sich auf der HTML-Seite und durch die BeautifulSoup-Bibliothek wird jeder Typ der Wetterprognose identifiziert und lokalisiert. Die automatische Datenerfassung der Wetterprognose erfolgt dann über das weiterentwickelte Python-Skript, das in einer Endlosschleife läuft. Sobald eine Stunde vergangen ist, wird eine Unterfunktion aufgerufen, die das automatische Auslesen der Wetterprognosen umsetzt. Nachdem die entsprechenden Wetterprognosen heruntergeladen wurden, werden die Daten in einer SQLite-Datenbank gespeichert. Der Grund der Auswahl der Datenbank besteht darin, dass diese Datenbank hochkompatibel sowohl für Python als auch für Matlab ist. Die durch das Python-Skript ausgelesenen Daten werden weiter zum Matlab-Programm geliefert, mit dem die Prognose der PV-Leistung getroffen wird. Es zeigt sich, dass durch diese Umsetzung ein schneller und zuverlässiger Datenaustausch zwischen Python und Matlab gewährleistet ist.

B Erfassung der Leistungsdaten

Die durch die Instituts-PV-Anlage erzeugte elektrische Leistung wird an den zwei Wechselrichtern vom Hersteller KOSTAL Solar gemessen. Die beiden Wechselrichter haben jeweils eine feste IP-Adresse und sind mit dem Uni-Netz verbunden. Die aktuellen Leistungsdaten sind über die beiden IP-Adressen abrufbar. Weil die beiden Wechselrichter jeweils unterschiedliche Firmwares besitzen, werden die Abfragen der aktuellen Leistungsdaten mit zwei unterschiedlichen Prozessen durchgeführt. Für den Wechselrichter PIKO 8.5 ist die IP-Adresse ohne Passwortabfrage zugänglich. Die Messwerte des Wechselrichters PIKO 8.5 (z.B. DC-Spannung, DC-Leistung, AC-Spannung und AC-Leistung) werden über HTTP im Form eines JSON-Formats mit einem Python-Skript ausgelesen. Für den Wechselrichter PIKO 8.3 ist es allerdings erforderlich, sich mit einem Passwort einzuloggen. Um das Auslesen der Leistungsdaten des Wechselrichters PIKO 8.3 umzusetzen, wird dieser Prozess ebenfalls mit einem Python-Skript durchgeführt, das mithilfe von der Bibliothek-urllib entwickelt wurde und online gefunden werden kann. Nachdem die Leistungsdaten der beiden Wechselrichter erfolgreich ausgelesen wurden, werden die Daten vorübergehend in Python in Form einer Liste gespeichert. Sobald der Zeitpunkt zum Downloaden erreicht ist, werden die Daten aus Python in dieselbe SQLite-Datenbank exportiert, in der sich die Wetter-Prognosedaten befinden. Der Ablaufplan der gesamten Datenerfassung ist in Abb. B-1 dargestellt.

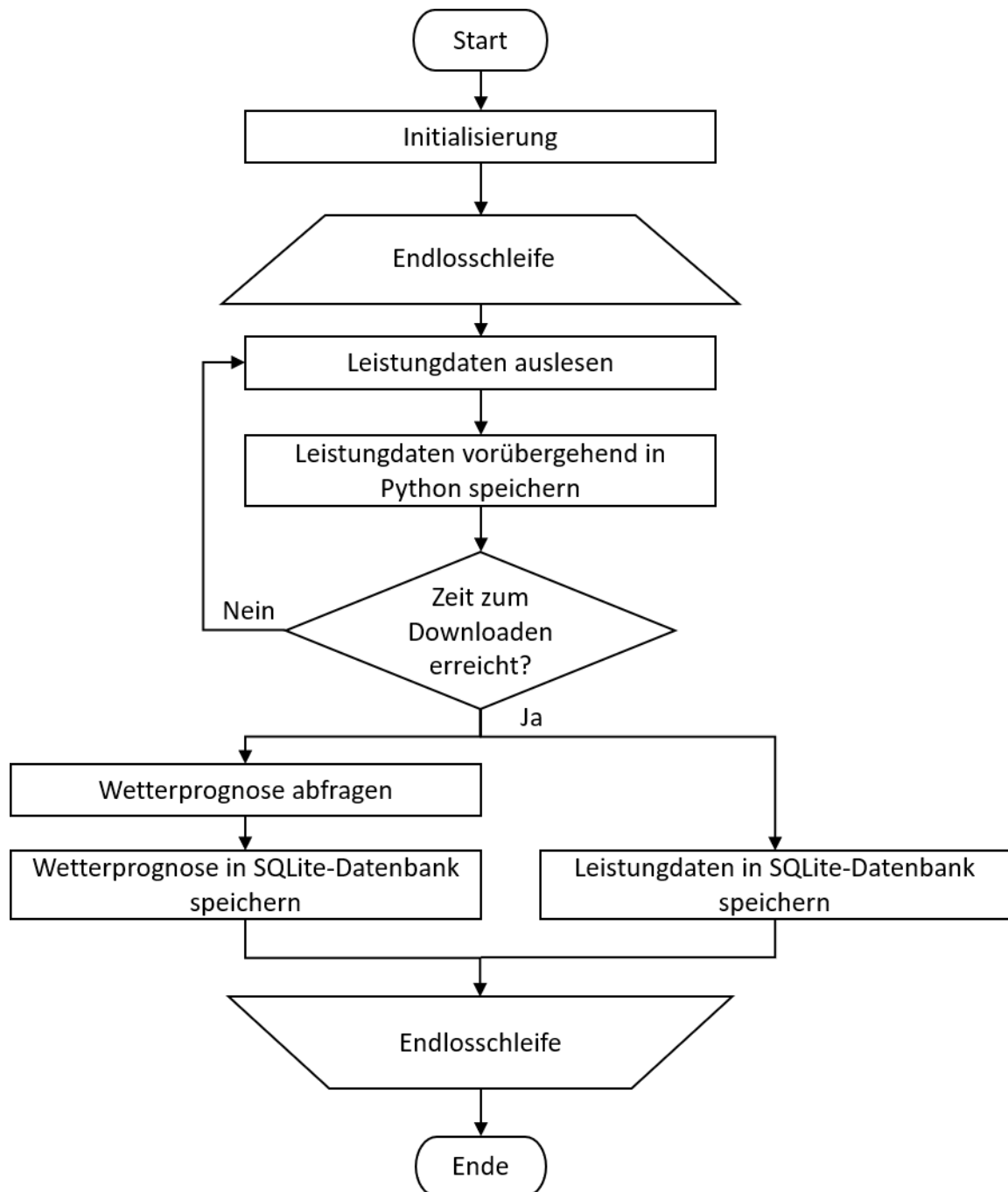


Abb. B-1: Ablaufplan der Datenerfassung

Eigenständigkeitserklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate kenntlich gemacht zu haben. Des Weiteren versichere ich, dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfverfahrens gewesen ist und noch nicht veröffentlicht wurde. Das eingereichte elektronische Exemplar stimmt mit den anderen Exemplaren überein.

Haoyan Li Stuttgart , den 04.06.2019