

# Vergleich verschiedener Verfahren für die Fernkältelastprognose bei den Stadtwerken München

Wissenschaftliche Arbeit zur Erlangung des Grades Master of Science (Msc)  
an der Fakultät für Mathematik der Technischen Universität München.

**Betreut von** Dipl.-Ing. Inga Maria Müller  
Prof. Dr. Thomas Hamacher (Lehrstuhl für Erneuerbare und Nachhaltige Energiesysteme)  
Prof. Dr. Claudia Klüppelberg (Lehrstuhl für Mathematische Statistik)

**Eingereicht von** Ambroise Idoine

Begonienstr. 4

80939 München

+33 6 82 05 93 52

**Eingereicht am** München, den 15.04.2019



## **Kurzfassung**

Mit Fernkältenetze kann die zunehmende Nachfrage nach Klimakälte zentralisiert und effizient erzeugt werden. Zu diesem Zweck soll eine sorgfältige Analyse der zukünftigen Nachfrage als Basis für eine Optimierung der Fahrweise und des Netzausbaus vorliegen. Hierbei ist es wichtig, die kurzfristige Nachfrage sowie die langfristige Entwicklung des Netzes vorhersagen zu können und die wichtigsten Einflussparameter zu identifizieren. Im Rahmen dieser Arbeit wurden zuerst verschiedene Prognoseverfahren für die kurzfristige Nachfrage implementiert und miteinander verglichen. Anhand deren Ergebnisse kann eine ausführliche Sensitivitätsanalyse durchgeführt werden. Für dieses Anwendungsfall hat eine Regression basiertes Model mit ausgewählten Eingängen und verschiedene Vor- und Nacharbeitung Ansätze die beste Prognose erstellt.

Zum Schluss wurden diese Erkenntnisse benutzt, um langfristigen Prognosemodelle zu entwickeln, welche in die kommerziellen Software BoFiT implementieren worden sind.

## **Abstract**

District Cooling can help cover the increasing demand for air-conditioning in a centralized and optimized manner. However, any optimized operation and extension of the cooling network should be based on a careful analysis of the demand. Of interest is not only the short-term demand but also the long-term trend. The most impactful parameters for the demand should also be known. In the current work, various short-term forecasting methods were first implemented and compared. The results were then used to conduct a thorough sensitivity Analysis. In this use case, the most accurate forecast was obtained using a Regression based model with selected Inputs and various pre- and postprocessing steps.

Finally, long-term models were constructed using the acquired knowledge, and the forecast models were implemented in the commercial software BoFiT.



## **Rechtserklärung**

Hiermit erkläre ich,

Name: Idoine  
Vorname: Ambroise  
Matrikelnummer: 03697470

dass ich die beiliegende Masterarbeit zum Thema

### **Vergleich von verschiedenen Verfahren für Fernkältelastprognose bei der Stadtwerke München**

selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie alle wörtlichen und sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet und die entsprechende Quelle angegeben habe.

Ich bestätige, die „TUM Richtlinien zur Sicherung guter wissenschaftlicher Praxis und für den Umgang mit wissenschaftlichem Fehlverhalten an der Technischen Universität München, 2015“, wie sie auf der Internetseite der Stabsstelle Chancengleichheit nachzulesen sind, anzuerkennen.

Vom Lehrstuhl und seinen Mitarbeitern zur Verfügung gestellte Hilfsmittel, wie Modelle oder Programme, sind ebenfalls angegeben. Diese Hilfsmittel sind Eigentum des Lehrstuhls bzw. des jeweiligen Mitarbeiters. Ich werde sie nicht über die vorliegende Arbeit hinaus weiterverwenden oder an Dritte weitergeben.

Einer weiteren Nutzung dieser Arbeit und deren Ergebnisse (auch Programme und Methoden) zu Zwecken der Forschung und Lehre, stimme ich zu.

Ich habe diese Arbeit noch nicht zum Erwerb eines anderen Leistungsnachweises eingereicht.

München, 15.04,2019

(Bearbeiter: Idoine, Ambroise)

## Erklärung zur Weitergabe der Arbeit

Ich erkläre mich einverstanden mit der Weitergabe der vorliegenden Arbeit an:

- Aktuell oder zukünftig am Lehrstuhl wissenschaftliche Arbeit schreibende Studierende
  - Pauschal durch Mitarbeitende
  - Nur nach jeweiliger vorheriger Rücksprache.
- Aktuell oder zukünftig am Lehrstuhl wissenschaftliche Mitarbeitende
  - Pauschal durch andere Mitarbeitende
  - Nur nach jeweiliger vorheriger Rücksprache.

Mein Urheberrecht und persönliches Nutzungsrecht bleibt davon unberührt.

München, 18.04.2019

(Bearbeiter: Name, Vorname)

# Inhalt

Kurzfassung .....	3
Abstract .....	3
Rechtserklärung .....	5
Erklärung zur Weitergabe der Arbeit .....	6
1. Einleitung .....	9
2. Methodik.....	11
2.1. Wissensstand .....	11
2.2. Vorgehensweise .....	13
2.3. Bezeichnung.....	14
2.4. Bewertungsverfahren .....	15
3. Explorative Datenanalyse .....	17
3.1. Datenbeschreibung .....	17
3.2. Clustering von einzelnen Kunden.....	19
3.3. Korrelationsanalyse .....	23
4. Prognoseverfahren.....	27
4.1. Allgemeine Methodik .....	27
4.2. Multivariate Regression .....	34
4.3. Support Vektor Regression .....	39
4.4. K-nächste Nachbarn.....	44
4.5. Neuronale Netze.....	48
5. Verbesserung der Modelle und Nachbearbeitung den Prognosen .....	57
5.1. Zusätzliche Verbesserungen .....	57
5.2. Summe von Einzelprognosen.....	64
5.3. Parameterauswahl und Sensitivität der einzelnen Parameter.....	67
5.4. Fehleranalyse und Konfidenzintervall.....	80
6. Vergleich mit Fernwärmenetz-Prognose.....	81
7. Langfristige Szenarien .....	84
7.1. Netzzuwachs .....	85
7.2. Künstliche Zeitreihen.....	87
8. Umsetzung im BoFiT.....	99
8.1. Zusammenfassung BoFiT .....	99
8.2. Mögliche Verfahren .....	100
8.3. Prognosemodelle in BoFiT .....	101
8.4. Ergebnisse.....	105
9. Zusammenfassung und Ausblick .....	107
Literaturverzeichnis .....	110
Abbildungsverzeichnis.....	112
Abkürzungsverzeichnis .....	115
Tabellenverzeichnis.....	116

Anhänge .....	117
I. Support Vektor Regression .....	117
II. Parameterauswahl: Weitere Ergebnisse .....	119
III. Parametersensitivität .....	123



# 1. Einleitung

Seit 2005 betreibt die Stadtwerke München (SWM) ein Fernkältenetz in der Innenstadt. Inzwischen hat es eine gewisse Größe erreicht, bei der durch eine optimierte Fahrweise signifikante Kostenersparnisse zu erwarten sind [1]. Zu diesen Optimierungszweck sollen also präzise Kältelastprognose vorliegen.

Die relativ geringe Kundenzahl und das Wachstum des Münchner Fernkältenetzes stellen besondere Anforderungen für die Erstellung von Kältelastprognose dar. Die Last ist aus diesen Gründen anfälliger für die individuellen Kundenverhalten. Zudem sollen die Modelle soweit wie möglich die zukünftige Entwicklung berücksichtigen.

Abhängig von der Anwendung werden verschiedene Zeithorizont betrachtet. Kurzfristige Prognose (unter zwei Wochen) werden benötigt, um die Fahrweise der Kältemaschinen und Kältespeicher durch eine sorgfältige Einsatzplanung zu optimieren. Hingegen werden langfristige Prognose (auf mehreren Jahren) benutzt, um den Netzausbau zu planen, und neuen Erzeugungsanlage zu dimensionieren.

In der Literatur werden verschiedene Verfahren für kurz- und langfristige Prognosen aufgeführt. Im Energiebereich werden solche Methoden besonders für Fernwärme-, Stromverbrauchs- und Strompreisprognosen angewandt. Neben den klassischen Methoden (Multivariate Regression, ARMA, Random Forest...) werden maschinelle Lernverfahren (Neuronale Netze) immer häufiger implementiert [2]. Diese verschiedenen Ansätze können aber komplementär benutzt werden.

Ziel dieser Arbeit ist es, die mathematischen Verfahren zu vergleichen und angepasste Modelle zu entwickeln, um die beste Prognose für den ausgewählten Zeithorizont zu liefern. Dazu sollte eine quantitative Bestimmung der Einflussparameter aufgeführt werden.

In dieser Arbeit werden verschiedene Modelle implementiert, die auf den Methoden Multivariate Regression, ARMA, Support Vektor Regression, K-Nächste Nachbarn und Neuronale Netze basieren.

Nach einer explorative Datenanalyse (Kapitel 3) werden in Kapitel 274 kurzfristige Prognoseverfahren auf eine Zeithorizont von zwei Tage untersucht (Das entspricht die Zeithorizont für die Einsatzplanung). Anhand dieser Erkenntnisse, und nach eine sorgfältige Sensitivitätsanalyse (Kapitel 5) werden dann langfristige Prognoseverfahren implementiert (Kapitel 7).

Für die Prognose und Einsatzplanung verwenden die Stadtwerke München das kommerzielle Optimierungsprogramm BoFiT. Am Ende dieser Arbeit werden die erstellten Prognosemodelle in BoFiT integriert werden, um eine Anwendung im Arbeitsalltag zu ermöglichen (Kapitel 8). Hierbei ist zu berücksichtigen, dass das Tool benutzerfreundlich gestaltet wird. Im Vergleich zu herkömmlichen Programmiersprachen ist BoFiT weniger flexibel, was bei der Umsetzung eine besondere Herausforderung darstellt.

## 2. Methodik

### 2.1. Wissensstand

Auf dem Themengebiet für kurzfristige Prognosen für die Energiebranche wurden bereits viele Forschungen durchgeführt. Obwohl es einige Arbeiten gibt, die sich explizit mit Fernkälte beschäftigen [3], liegen die Schwerpunkte der meisten Untersuchungen bei Stromverbrauch, Strompreisentwicklung und Fernwärmelast, da diese Bereiche gegenwärtig von größerer wirtschaftlicher Bedeutung sind.

Die meisten Autoren benutzen auf Künstlichen Neuralen Netzen (NN) basierende Verfahren. Darüber hinaus werden weitere Ansätze wie Multivariate Regression [4], Support Vektor Regression (SVR) [5], Autoregressive Moving Average (ARMA) Varianten [3] [5], K-Nächste Nachbarn (KNN) [6] oder Entscheidungsbaum Methode [5] [7] verfolgt. Abhängig von der Anwendung werden unterschiedliche Methoden bevorzugt.

Unter dem Begriff Künstliche Neuronale Netze findet man zahlreiche Verfahren, die in zwei Gruppe aufgeteilt werden können: Feedforward NN und Rekursive NN. Feedforward NN berechnen jeden (Zeit-)punkt separat. Ein bekanntes Beispiel hierfür wäre das Perzeptron. Rekursive Netze hingegen speichern sämtliche Informationen im virtuellen Gedächtnis ab, um die nächsten Punkte besser prognostizieren zu können (z.B. LSTM oder GRU).

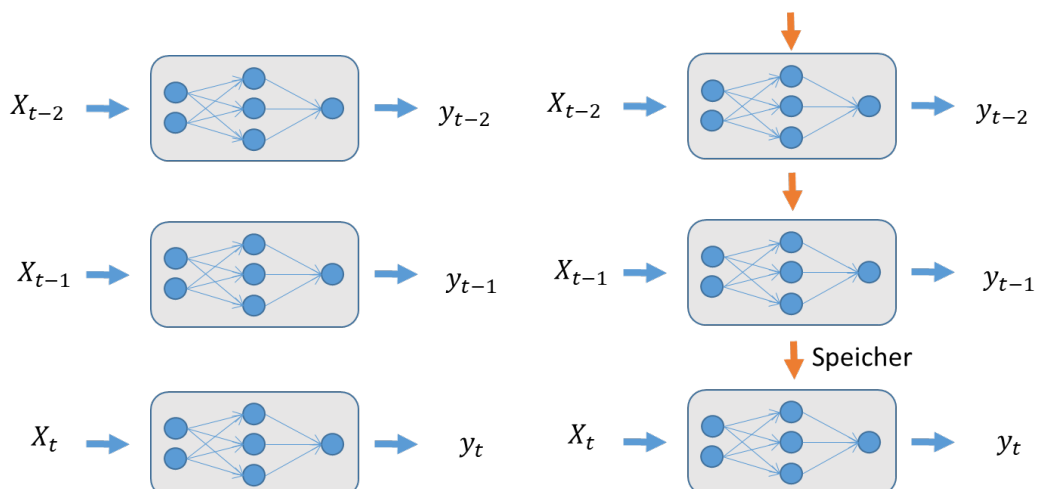


Abbildung 1: Feedforward NN (Links) vs Rekursive NN (Rechts)

Die Flexibilität von Rekursiven Netzen ist ein großer Vorteil. Besonders relevant für die Prognose sind die „*Long Short Term Memory*“ Netze (LSTM). Dieser Art von rekursiven neuronalen Netzen hat sehr gute Ergebnisse in der Spracherkennung erbracht [7] und wurde in den letzten Jahren verstärkt für Prognosen eingesetzt [5]. Allerdings befürwortet die leichtere Architektur von „*Gated Recurrent Units*“ (GRU) dessen Einsatz, die angeblich ähnlichen Ergebnisse liefern [8].

Es lassen sich ebenfalls mit Feedforward Netzen gute Prognosen erstellen, indem man zeitverschobene Daten als Eingangsparameter nutzt. Der Unterschied zu rekursiven Netzen liegt darin, dass bei Feedforward Netzen festgestellt wird, welche Zeitverschiebungen benutzt werden. So wird zum Beispiel in [3] ein NARX erstellt (Nicht-lineare Autoregressive Model mit Exogenen Variablen). Nach [6] fällt auch das „*Convolutional Net*“ (CNN) in diese Kategorie.

Wenn ein Verfahren gewählt wurde, stellt sich das Problem der Auswahl von passenden Eingangsparametern („feature selection“). Mit der Unterstützung von technischer Expertise und einer sorgfältigen Korrelationsanalyse kann erörtert werden, welche Parameter für das Prognosemodell relevant sind [9]. Trotzdem wird eine bessere Prognosegüte erreicht, wenn Parameterauswahlalgorithmen implementiert und ausgeführt werden (insbesondere für multivariate Regression) [5] [10]. Während einige Prognoseverfahren wie Partielle Kleinste Quadrate Methode (PLS) selbst eine Art Parameterauswahl implementieren (Eingebettete-Ansatz), wird meistens ein separater Algorithmus benutzt, um die Parameterauswahl durchzuführen (Wrapper-Ansatz) [11]. Im zweiten Fall muss ein „Teilmenge Auswahl Problem“ gelöst werden.

Bereits ab wenigen Parametern steigt die Anzahl von möglichen Parameterkombinationen signifikant an. Deshalb wurden heuristische Algorithmen entwickelt, die nur einen Teil der Kombinationen testen. Dazu gehören zum Beispiel genetische Algorithmen [10] wie „Forward/Backward Stepwise Regression“, die Methode LASSO [4], Schrittweise Entfernung von wenig aussagefähigen Parametern oder Schrittweise Gewichtung von Einflussparametern [11].

Obwohl der Schwerpunkt dieser Arbeit auf der kurzfristigen Prognose liegt, ist die SWM auch an langfristigen Prognosen sowie künstlichen Lastszenarien interessiert, um den Netzausbau zu planen. Hierzu gibt es auch ausreichend Literatur wie beispielsweise [12] oder [13]. In [12] werden drei mögliche Vorgehensweisen aufgelistet. Demnach können entweder künstliche Wetterzeitreihen als Eingang für die Szenarien genutzt werden, Quantil Methoden Anwendung finden oder Residuen Analysen durchgeführt werden, wie bei ARMA Modellen.

## 2.2. Vorgehensweise

Zuerst wurde den aktuellen Wissenstand untersucht. Folglich wurde vier Prognoseverfahren ausgewählt, die sich bereits in der Literatur bewährt haben:

- Multilineare Regression
- Support Vektor Regression
- K-nächste Nachbarn
- Neuronale Netze

Zweitens wurde eine sorgfältige Datenanalyse ausgeführt. Ziel dieser Analyse war es, vier Fragen zu beantworten:

- Was sind die Haupteinflussparameter?
- Wie lässt sich das zeitliche Lastmuster erklären?
- Ist das Netzwachstum vernachlässigbar?
- Können die Kunden auf Basis der Last in Gruppen eingeteilt werden?

Zunächst wurden verschiedenen Modelle auf Basis den ausgewählten Prognoseverfahren erstellt, verfeinert und anschließend miteinander verglichen.

Dazu wurde die Prognose zusätzlich für jeden einzelnen Kunden durchgeführt und anschließend kumuliert. Zeitgleich erfolgte die Implementierung in das Optimierungsprogramm BoFiT.

Zum Schluss wurden auch langfristige Prognosen für verschiedene Wetter- und Netzausbauszenarien erstellt.

## 2.3. Bezeichnung

In der vorliegenden Arbeit werden nachfolgenden Bezeichnungen oft benutzt:

- Zeit:  $t \in T = \{1, 2, \dots, n\}$
- Zeitintervall:  $\Delta T$
- Eingangsparmeter (Zwei-dimensional:  $p$  Eingänge über alle Zeitpunkte  $t \in T$ )

$$X = (X_t)_{t \in T} = (x_{i,t})_{i \in \text{Eingänge}, t \in T}$$

- Referenzzeitreihe (Kältelast):  $y = (y_t)_{t \in T}$
- Prognose:  $\bar{y}$  oder  $f(X)$
- Prognosefehler (nicht prognostizierbares Rauschen):  $\epsilon$
- Koeffizienten:  $c = (c_i)_{i \in I}$  und  $s = (s_i)_{i \in I}$
- Optimale variable werden mit \* gekennzeichnet:  $c^*, X^*, \dots$

Dazu werden Bekannte Funktionen oft verwendet: Mittelwert, Standard Abweichung, Varianz, Exponentiell: *mean, std, var, exp*

## 2.4. Bewertungsverfahren

Schwerpunkt dieser Arbeit ist es, verschiedene Modelle zu erstellen und zu vergleichen, um die zukünftige Fernkältelast für München zu prognostizieren. Für die Messung der Prognosegenauigkeit ist eine Metrik notwendig.

Eine naheliegende Wahl bietet die Methode RMSE („Root Mean Square Error“, die Quadratwurzel des mittleren quadratischen Fehlers).

$$RMSE = \sqrt{\frac{1}{n-1} \sum_{1 \leq i \leq n} (y_i - \bar{y}_i)^2} \quad (1)$$

$y$  stellt hier die Referenzzeitreihe und  $\bar{y}$  die Prognose dar.

Solange alle Modelle auf derselben Referenzzeitreihe bewertet werden, ist ein prozentualer (oder skalenunabhängiger) Fehler zwar nicht notwendig, aber hilfreich, um die Prognosegüte zu messen.

In [14] werden unterschiedliche Metriken vorgeschlagen. Es wird empfohlen, MASE („Mean Absolute Scaled Error“) zu benutzen. Die Idee ist, die jetzige Prognose mit einer einfachen Referenzprognose zu vergleichen, wobei man die Methode MAE (Mittlere Absolute Fehler) für beide Prognosen nutzt:

$$MASE(\bar{y}) = \frac{MAE(\bar{y})}{MAE(\bar{y}_{\text{einfach}})} \quad (2)$$

Damit es kompatibel mit RMSE bleibt, wird eine ähnliche Metrik verwendet, die auf MSE statt MAE basiert. Als Referenzprognose wird der mittlere Wert der Last benutzt, deren RMSE die Standardabweichung ist (STD).

$$\text{Relativer Fehler} = \frac{RMSE(\bar{y})}{RMSE(\bar{y}_{\text{einfach}})} = \frac{RMSE(\bar{y})}{std(y)} \quad (3)$$

Die Modelle werden auf Basis historischer Daten bewertet. Für das Training und die Bewertung der Prognosegüte (Validierung) können nicht dieselben Datensätze verwendet werden.

Unter diese Betrachtung stehen verschiedene Verfahren zu Verfügung, wie:

- Kreuzvalidierung („cross-validation“)

Der Datensatz wird in  $n$  Intervalle geteilt.

Für jedes Intervall gilt:

- auf allen anderen Intervallen trainieren
- auf dem Intervall selbst validieren



Abbildung 2: Kreuzvalidierung

- Zufällige Kreuzvalidierung („random cross-validation“)

Anstelle von Intervallen werden zufällige Datenpunkte für die Validierung gewählt.



Abbildung 3: Zufällige Kreuzvalidierung

- Gleitender Horizont („rolling horizon“)

Diese Methode ist der Kreuzvalidierung ähnlich, es werden jedoch nur Daten für das Training benutzt, die vor der Validierung liegen (blau).



Abbildung 4: Gleitender Horizont

Bei einem statischen Netz ohne Netzzuwachs und ohne langfristige Veränderung des Verbrauchsmusters wären die beiden ersten Verfahren geeignet. Bei dynamischen Netzen muss die Methode Gleitender Horizont gewählt werden, um unter realen Bedingungen zu arbeiten. Aus diesem Grund wird nachfolgendes Vorgehen angewendet.

Gleitender Horizont mit RMSE und Relativer Fehler.



## 3. Explorative Datenanalyse

In der Explorative Datenanalyse werden Verbrauchsmuster erkannt und Zusammenhänge zwischen die Eingänge erkannt.

### 3.1. Datenbeschreibung

Für die Prognose stehen folgende Zeitreihen zu Verfügung:

- Historische Lastdaten
  - ca. drei Jahre: 15. September 2015 bis 22. Oktober 2018
  - Erzeugung für das gesamte Netz und Verbrauch der einzelnen Kunden
- Wetterdaten
  - Prognose und Ist-Werte von 01. Januar 2015 bis 22. Oktober 2018
  - Temperatur, Sonnenscheindauer, globale Strahlung, Niederschlag, Wind
  - Luftfeuchte gibt es nur als historische Werte, aber nicht als Prognose
- Kalendarische Daten
  - Ferientage, Wochentage, Brückentage

In Abbildung 5 ist die Last über den gesamten Betrachtungszeitraum dargestellt. Wie zu erwarten, gibt es im Sommer einen höheren Kältebedarf.

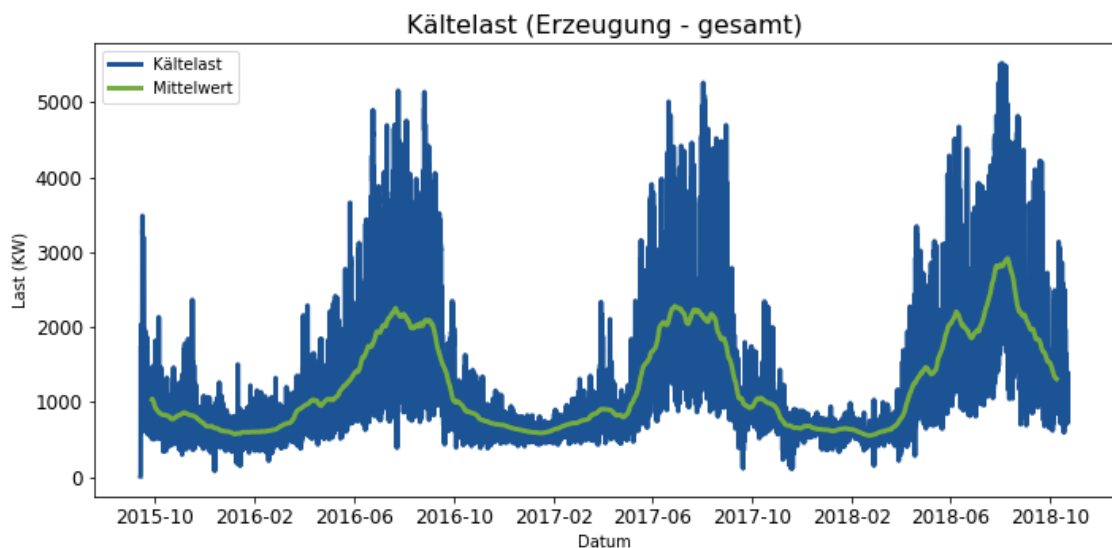


Abbildung 5: Kältelast gesamt

Es gibt ein leichtes Wachstum des Netzes während der ausgewählten Periode sowie eine Mustersänderung. Im Winter 2015-2016 war die Nachfrage wesentlich höher als in 2017-2018. Das liegt an individuellen Kundenverhalten. Wegen der relativ geringen Anzahl an Kunden können nämlich individuelle Kunden eine spürbare Auswirkung auf der Gesamtlast haben.

Da die meisten Kunden Bürogebäude oder Geschäfte sind, fällt der Verbrauch an den Sonntagen stark ab. Man merkt auch, dass sich an Feiertagen eine ähnliche Last ergibt wie sonntags. Samstag hingegen ähnelt stark dem Verlauf der Wochentage, wie in Abbildung 6 zu erkennen.

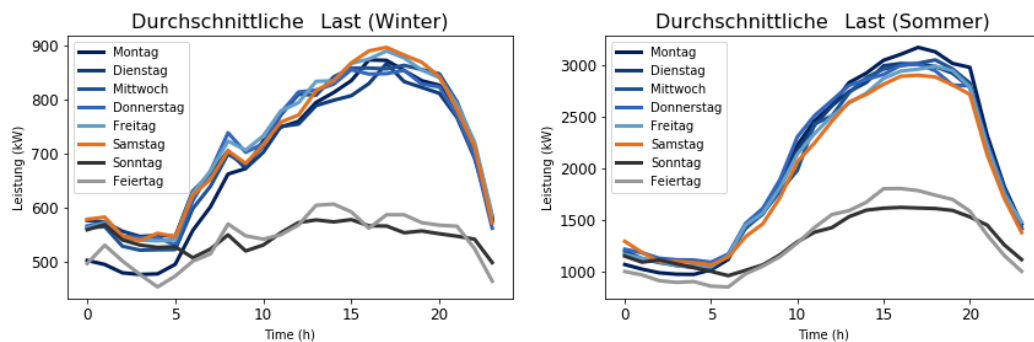


Abbildung 6: Durchschnittliche Last im Winter (rechts) und im Sommer (Links)

## 3.2. Clustering von einzelnen Kunden

Die einzelnen Kunden weisen ein unterschiedliches Verbrauchsmuster auf. Um deren Verbrauchsverhalten besser zu verstehen, werden diese Kunden in Gruppen geordnet, basierend auf deren Verbrauch.

So kann das Verhalten eines neuen Kunden besser prognostiziert werden (siehe Kapitel 7).

### 3.2.1. Clustering Verfahren

Beim sogenannten Clustering wird eine Distanz zwischen Zeitreihen festgestellt und ein bewährter Clustering Algorithmus angewendet.

In [15] werden verschiedene Vorgehensweise vorgeschlagen. So können entweder die ganze Zeitreihen benutzt oder lediglich Merkmale wie Autokorrelation oder ARIMA (Autoregressive Integrated Moving Average) Koeffizienten als Basis für die Algorithmen genommen werden.

Wenn die ganze Zeitreihen betrachtet werden, können auch andere Distanzen als die Euklidische Distanz verwendet werden. Beispielsweise bieten sich hier die Hausdorff Distanz „Dynamic Time Warping“ [16], „Shape Extraction“ oder „Shape-based distance“ [17] an.

Dazu werden in [18] verschiedene Clustering Algorithmen beschrieben.

Nach diesen Betrachtungen empfiehlt sich aber, den K-means Algorithmus anzuwenden (K steht hier für den Anzahl von Gruppen). Dieser leicht einsetzbare Ansatz kommt am häufigsten vor und hat in den Fall befriedigenden Ergebnisse geliefert.

#### **K-means Algorithmus:**

1. Man wählt zufälligerweise K Punkte (in unseren Fall K Zeitreihen). Diese Punkten werden „Zentrum“ genannt.
2. Jeder andere Punkt wird dem nächsten Zentrum zugeordnet.
3. Für jede Gruppe (Zentrum + zugehörige Punkte) wird ein Neues Zentrum berechnet. Normalerweise ist dies der Mittelwert von allen Punkten der Gruppe.
4. Solange sich das Zentrum der Gruppe ändert, werden die Schritte 2 und 3 wiederholt.

Im Anwendungsfall des Fernkältenetzes München liegen die Lastdaten von drei Jahren und insgesamt 25 Kunden vor.

Zuerst wurden als Basis für den K-means Algorithmus fünf Merkmale eingeführt:

- Mittelwert
- Durchschnittlicher Feiertag
- Durchschnittlicher Werktag
- Durchschnittliche Woche
- Anzahl von Null-Werten

Diese Merkmal-basierte Ansatz bietet den Vorteil, historische und neue Kunden ohne Bias vergleichen zu können. Problematisch ist aber, dass diese Merkmale willkürlich ausgewählt und gewichtet sind.

Zum Schluss eignet es sich ganze Zeitreihen zu benutzen.

Daher wurde die Euklidische Distanz zwischen normierten Zeitreihen benutzt, wodurch die fehlenden Werte systematisch entfernt wurden.

Für die Zeitreihen  $w$  und  $z$  ergibt sich:

$$Dist(w, z) = \sqrt{\frac{1}{n} \sum_{1 \leq t \leq n} \left( \frac{w_t}{mean(w)} - \frac{z_t}{mean(z)} \right)^2} \quad (4)$$

Wegen der Zufälligkeit der Zentren am Anfang des K-means Algorithmus ergeben sich unterschiedliche Ergebnisse, abhängig von der Güte der Anfangspunkte.

Dieses Problem wird gelöst, indem das beste Clustering von mehrere (hier: Tausend) Durchläufe ausgewählt wird, jedes Mal mit zufällig ausgewählten Startpunkten.

Um das beste Clustering bestimmen zu können, wird die Güte einer Gruppe in Form der Intra-Gruppe Distanz bemessen, verglichen mit der Varianz der normierten Kunden. („Wie nah zu einander sind die Zeitreihen, die in selben Gruppen eingeteilt werden im Vergleich zu allgemeine Zeitreihen“)

$$Güte(Cluster) = \frac{\sum_{Gruppe \in Cluster} \sum_{X_i \in Gruppe} Dist(X_i, mean(Gruppe))}{\sum_{X_i \in AlleKunde} Dist(X_i, mean(AlleKunde))} \quad (5)$$

Letztlich musste die Anzahl von Gruppen festgelegt werden. Dies kann nicht durch den Algorithmus bestimmt werden. Hierfür hat die Visualisierung der Ergebnisse für verschiedene K-Werte und der Güte der Ergebnisse geholfen. So wurden zum Schluss sechs Gruppen erstellt.

### 3.2.2. Clustering Ergebnisse

Die Hauptunterschiede lassen sich wie folgt beschreiben und sind in Abbildung 7 und Abbildung 8 optisch nachvollziehbar.

- Gruppe 1 und 2 haben eine Grundlast in der Nacht und an den Sonntagen.
- Gruppe 1 und 3 haben ein ausgeprägtes tägliches Muster im Winter, während Gruppe 2 und 4,5,6 einen flachen Verlauf im Winter haben.
- Gruppe 4 verbraucht weniger Kälte am Samstag während die anderen fast dieselben Last am Samstag haben wie in den anderen Wochentagen.
- Gruppe 4,5,6 beschreiben drei Ausreißer mit nur einem Mitglied.

Die Gruppenzentrum sind im Blau abgebildet, und die einzelnen Kunden in Grau (alle Zeitreihen sind normiert, wie oben beschrieben).

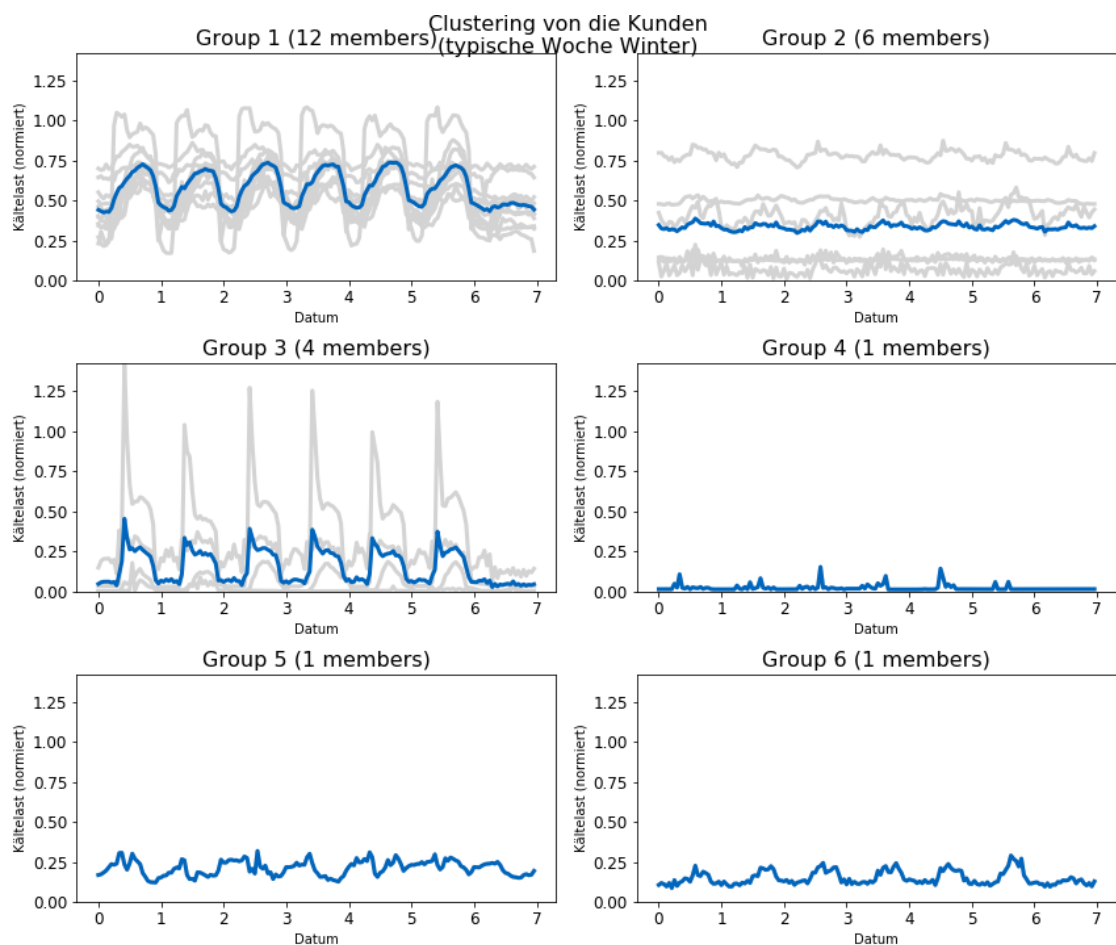


Abbildung 7: Clustering Ergebnisse, wöchentliche Last im Winter (normierte Zeitreihe)

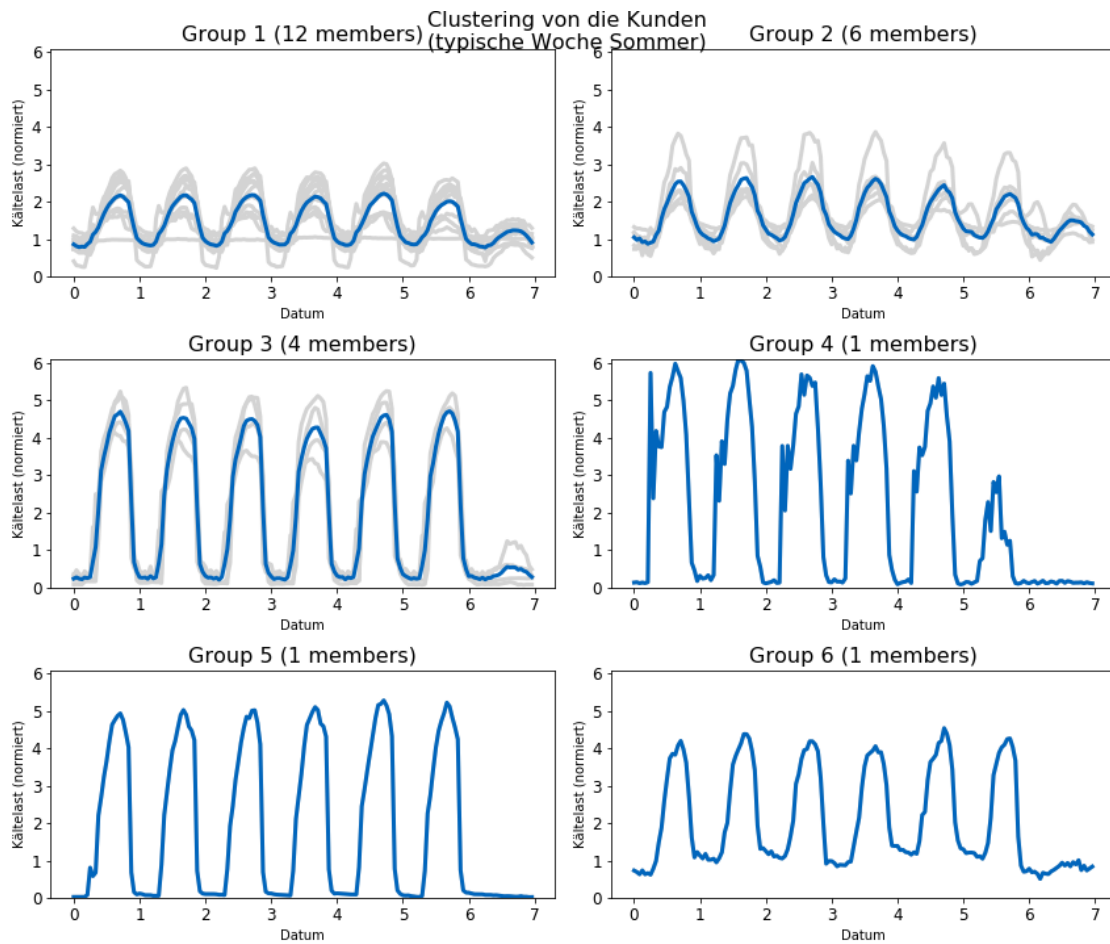


Abbildung 8: Clustering Ergebnisse, wöchentliche Last im Sommer (normierte Zeitreihe)

### 3.3. Korrelationsanalyse

Die Sensitivitätsanalyse beschäftigt sich primär mit der Frage, welche der eingangs genannten Größen die größte Relevanz für die Kältelastprognose hat. Im vorliegenden Fall sind drei Arten von Eingangsparametern vorhanden:

- Wetterdaten:  
Temperatur, Sonnenscheindauer, globale Strahlung, Luftfeuchte, Wind, Niederschlag
- Historische Daten:  
Last, Temperatur
- Kalendarische Daten:  
Wochentag, Stunde, Monat, etc...

Es wurde schon in 3.1 festgestellt, dass die Last stark von den kalendarischen Einflüssen abhängt. Einige Plots in Abbildung 9, Abbildung 10 und Abbildung 11 können helfen, die Korrelation mit den Wetterdaten und den historischen Daten zu verstehen:

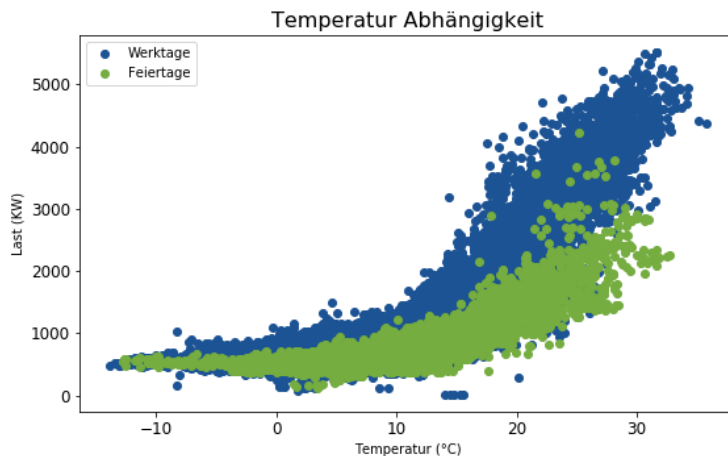


Abbildung 9: Temperatur Abhängigkeit der Last

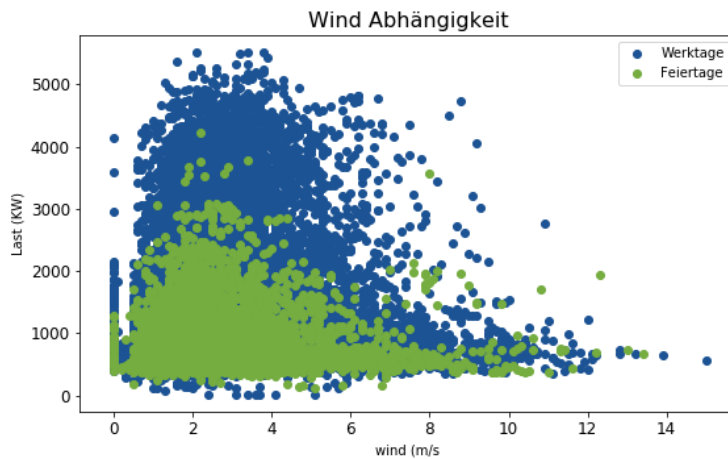


Abbildung 10: Wind Abhängigkeit der Last

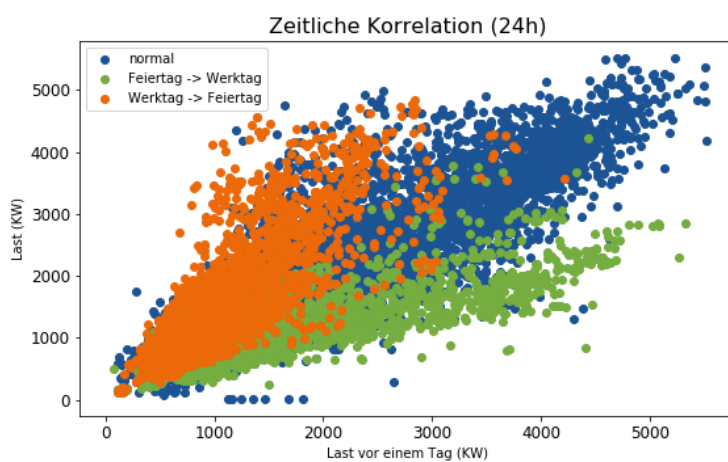


Abbildung 11: Zeitliche Korrelation der Last

Diese Abbildungen liefern folgende relevante Informationen:

- Es gibt eine klare Korrelation der Last mit der Temperatur und der historischen Last.
- Der Wind hat keinen großen Einfluss.
- Kalendarische Daten spielen immer eine Rolle und die Eingangsparameter sind abhängig voneinander, so dass es schwierig ist, nur einen Einflussfaktor einzeln zu betrachten.



Nun folgt die eigentliche Korrelationsanalyse.

Die (Pearson) Korrelation misst die lineare Abhängigkeit von den Variablen. Für die Zeitreihen  $w_t$  und  $z_t$  lässt sich berechnen:

$$\text{Korrelation}(w, z) = \frac{\sum_t (w_t - \text{mean}(w)) * (z_t - \text{mean}(z))}{\text{std}(w) * \text{std}(z)} \quad (6)$$

Mit dieser Definition kann man die Autokorrelation einer Zeitreihe bestimmen.

$$\text{Autokorrelation}(w, \text{lag}) = \text{Korrelation}(w, w_{-\text{lag}}) \quad (7)$$

Wobei  $w_{-\text{lag}}$  dieselbe Zeitreihe wie  $w$  ist, nur mit einem Zeitversatz von „lag“.

Eine andere interessante Größe ist die partielle Autokorrelation. Wenn  $w_t$  von  $w_{t-1}$  abhängt, entsteht automatisch eine Korrelation zwischen  $w_t$  und  $w_{t-2}$ . Die partielle Autokorrelation „bereinigt“ die Autokorrelation von diesem Effekt, um den tatsächlichen Einfluss von  $w_{t-2}$  zu berechnen [19] (siehe Abbildung 12).

Mit dieser Abbildung lassen sich die Zeitversätze identifizieren, welche mit der aktuellen Last korrelieren. Relevant für die Prognose sind also lediglich die Zeitversätze innerhalb eines Tages, und Tagesversätze (24h, 48h, ...)

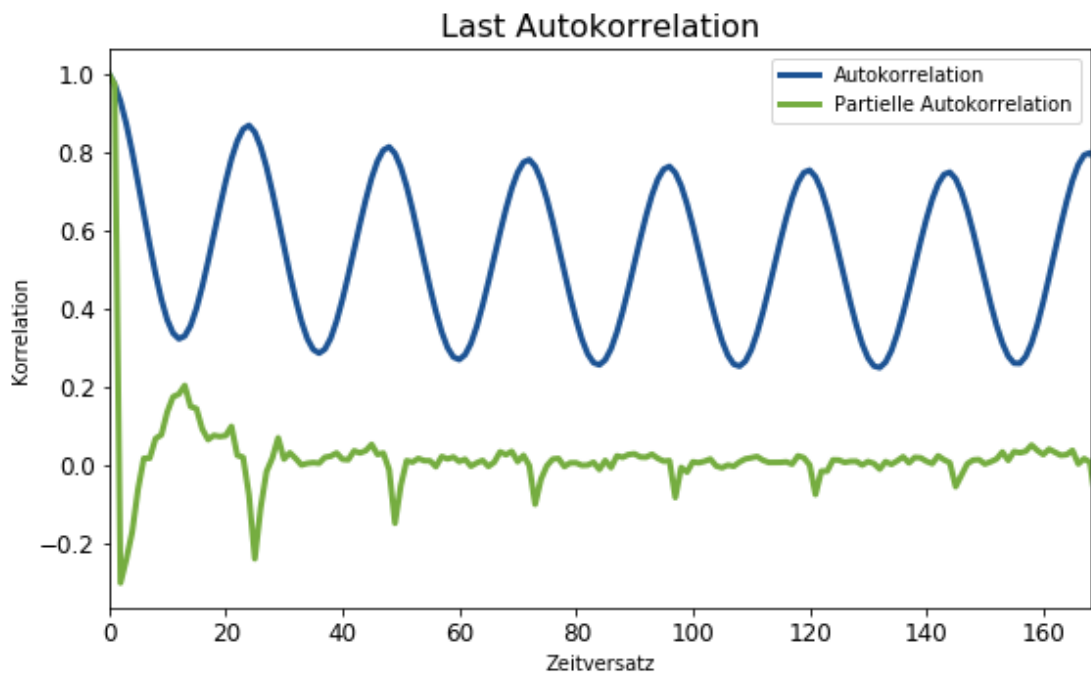


Abbildung 12: Autokorrelation der Kältelast

Ebenso kann die (lineare) Korrelation mit den Wettervariablen betrachtet werden (siehe Abbildung 13):

	Kältelast	Kältelast-24h	Temperatur	Niederschlag	Sonne	Global	Wind	Luftfeuchte	Feiertag
Kältelast	1,00	0,87	0,79	0,01	0,48	0,52	0,01	-0,50	-0,08
Kältelast-24h	0,87	1,00	0,74	0,05	0,40	0,46	0,02	-0,42	-0,05
Temperatur	0,79	0,74	1,00	-0,01	0,46	0,48	0,02	-0,57	-0,02
Niederschlag	0,01	0,05	-0,01	1,00	-0,10	-0,06	0,06	0,19	0,00
Sonne	0,48	0,40	0,46	-0,10	1,00	0,77	0,06	-0,53	0,00
Global	0,52	0,46	0,48	-0,06	0,77	1,00	0,08	-0,46	0,00
Wind	0,01	0,02	0,02	0,06	0,06	0,08	1,00	-0,16	0,01
Luftfeuchte	-0,50	-0,42	-0,57	0,19	-0,53	-0,46	-0,16	1,00	0,00
Feiertag	-0,08	-0,05	-0,02	0,00	0,00	0,00	0,01	0,00	1,00

Abbildung 13: Korrelation mit Wetterdaten

Es wird anhand dieser Abbildung festgestellt, dass die Last am stärksten mit der Kältelast vor 24h und der Temperatur korreliert.

Sonnenscheindauer, globale Strahlung und Luftfeuchte weisen auch eine hohe Korrelation mit der Last auf. Da diese Größen aber mit der Temperatur korrelieren und ein täglich wiederkehrendes Lastmuster aufweisen, kann vermutet werden, dass deren Korrelation mit der Kältelast sich hauptsächlich durch diese zwei Effekte erklären lässt (also, dass sie wenig Mehrwert für Prognose bringen).

Aus diesen Daten lassen sich auch andere Zeitreihen wie die Enthalpie berechnen, das mittlere Lastprofil oder Basisfunktionen für eine Regression (Splines, siehe 4.1.2), wie nachfolgend in Tabelle 1 dargestellt.

Tabelle 1: Korrelation mit berechneten Inputs

Haupteingang	Funktion	Fernkälte
Fernkälte	-	1,00
Jahr	Cosinus	-0,62
Jahr	Sinus	-0,19
Temperatur	-	0,80
Temperatur	Mittelwert vor 2 Tage	0,64
Temperatur	Vor 2 Tage	0,74
Temperatur	S-Spline (-10 ,10)	0,83
Temperatur	S-Spline (10,20)	0,90
Temperatur	S-Spline (20,30)	0,67
Enthalpie	-	0,79
Globale Strahlung	-	0,52
Globale Strahlung	S-Spline (0,333)	0,49
Globale Strahlung	S-Spline (333,667)	0,41
Globale Strahlung	S-Spline (667,1000)	0,28

## 4. Prognoseverfahren

Es werden vier verschiedene Prognoseverfahren eingeführt:

- Multivariate Regression
- Support Vektor Regression
- K-nächste Nachbarn
- Neuronale netze

Auf Basis diesen Verfahren werden verschiedene Prognosemodelle aufgebaut und getestet.

### 4.1. Allgemeine Methodik

Die Referenzvariable (hier die Kältelast)  $y$  wird als eine Funktion der Eingangsparameter  $X$  berechnet. Aus dem Unterschied zwischen dieser Referenz und der Prognose ergibt sich der Prognosefehler  $\epsilon$ :

$$y = f(X) + \epsilon \quad (8)$$

Für die Erstellung von Prognosemodellen eignet sich die folgende Vorgehensweise:

- Auf Basis der Hauptverfahren (Regression, neuronale Netze...) und unter Betrachtung von der Korrelationsstruktur des Problems werden verschiedene Modelle aufgebaut.
- Die Prognosen werden mit historische Lastdaten verglichen (siehe 2.3).
- Modelle, die einen niedrigen Prognosefehler aufweisen, werden weiter verbessert (bessere Parameterauswahl, Vor- und Nachbearbeitung der Prognose).

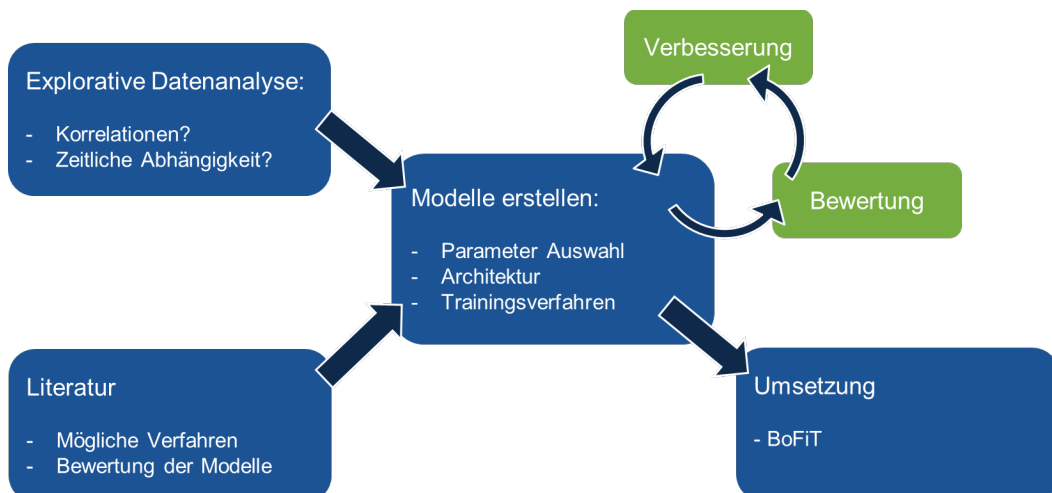


Abbildung 14: Erstellung eines Prognosemodells

### 4.1.1. Zusätzliche Eingangsparameter

Zusätzliche Eingangsparameter werden im Vorfeld berechnet, um die Prognosegüte zu verbessern. Das ist besonders nötig mit multivariater Regression, um nichtlineare Abhängigkeiten schätzen zu können, aber es wird auch für andere Methoden sinnvoll. Für diesen Anwendungsfall wurden deshalb vier Arten von Zeitreihen berechnet:

- Basis Interpolierungsfunktionen
- Zeitlich verschobene Zeitreihen
- Lastmuster und Mittelwerte
- Bekannte physische Funktion

Die Basis Interpolierungsfunktionen erlauben es, nichtlineare Abhängigkeiten nachvollziehen zu können, und die Information anders darzustellen: ein Cosinus mit Periode ein Tag entspricht besser die Periodizität den Tagen als die ursprüngliche Zeitreihe mit 24 Stunden, wie in Abbildung 15 dargestellt.

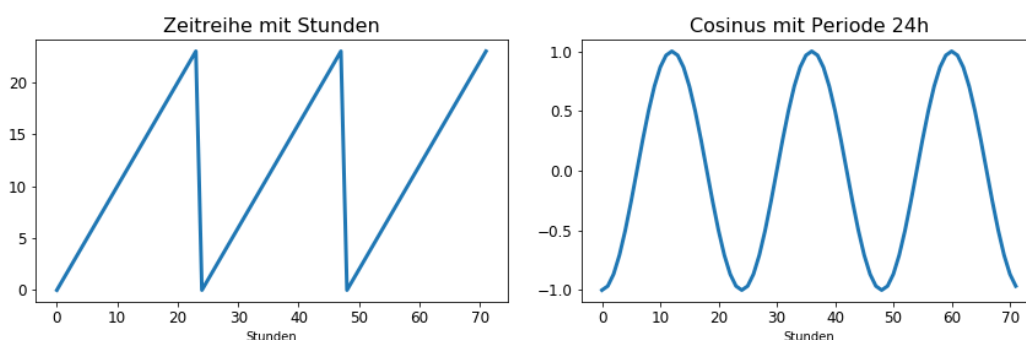


Abbildung 15: Cosinus statt Stundenzeitreihe

Mit zeitlich verschobenen Zeitreihen wird das Model auch historische Daten berücksichtigen. Es wird also „ARX“ benannt („Autoregressive“ mit externen Eingängen).

Eine Lastmuster und Mittelwerte der Last und der Temperatur vor einigen Tagen fassen einige Information zusammen für die Prognose.

Zusätzlich können andere physische Funktionen berechnet werden, wenn es festgestellt wurde, dass diese Funktionen einen direkten Einfluss auf die Last haben. In dem Fall wird vermutet, dass die Enthalpie eine wichtige Rolle spielt [1], so dass es aus der Temperatur und der Luftfeuchte ermittelt wurde

## 4.1.2. Basis Interpolierungsfunktionen

Die Basisinterpolierungsfunktionen sollten zwei Zwecke dienen. Zum einen können sie besser mit der Referenz korrelieren als die ursprünglichen Daten (z.B. den Cosinus mit Periode 24h statt Stundenwerte). Zweitens erlauben sie es, nichtlineare Abhängigkeiten besser abzubilden, ins Besondere mit multivariater Regression, die sonst nur lineare Abhängigkeiten abbildet.

Als Basis Interpolierungsfunktion eignen sich trigonometrische Funktionen (Cosinus und Sinus) für periodische Zeitreihen und „Splines“ für die übrigen Zeitreihen.

### Trigonometrische Funktion

Die Dirichlet-Bedingung besagt, dass jede periodische Funktion durch eine Summe von trigonometrischen Funktionen approximiert werden kann [20]. Dadurch wird die Grundlage für die Fourier Transformation gelegt.

#### **Satz von Dirichlet:**

Sollte die Funktion  $f: \mathbb{C} \rightarrow \mathbb{R}$   $\lambda$ -periodisch und stückweise  $C^1$  sein, gilt:

$\forall x \in \mathbb{C}$ , wenn  $f$  von  $x$  kontinuierlich ist:

**(9)**

Wo  $c_0, (c_n)_{n \geq 1}, (s_n)_{n \geq 1}$  die entsprechenden Fourier Koeffizienten sind.

Zusätzlich erhält man die  $L_2$ -Konvergenz für  $f \in L_2(\mathbb{R})$ , so dass die mittlere quadratische Abweichung (MSE) gegen Null geht, wenn die Anzahl von Fourier Frequenzen (d.h. die Anzahl von Cosinus und Sinus) wächst [21].

Anhand diesen Ergebnissen sind die trigonometrischen Funktionen eine naheliegende Wahl als Basisfunktionen für die Darstellung periodischen Eingänge.

## Splines

Im Falle von nicht-periodischen Zeitreihen empfiehlt sich, Splines anstatt trigonometrischer Funktionen anzuwenden. In [22] werden z.B. Splines ausführlicher eingeführt, und dann in Regressionsmodellen benutzt.

Es sind stückweise polynomiale Kurven von einem gewissen Grad  $q$ , die  $q - 1$  mal stetig sind. Das Raum  $\mathcal{S}_q(t_1, \dots, t_r)$  den Splines vom Grad  $q$  auf die Stützpunkte  $t_1 \leq t_2 \dots \leq t_r$  wird so beschrieben:

Die Funktion  $f \in \mathcal{S}_n(t_1, \dots, t_q)$  wenn:

- $\forall i \in [0, r], f|_{(t_i, t_{i+1})}(x) = \sum_{j=0}^q c_{i,j} x^j$   
(Polynomial von Grad  $q$  mit der Konvention  $t_0 = -\infty$  und  $t_{r+1} = +\infty$ )
- $f \in \mathcal{C}^{r-1}(\mathbb{R})$   
( $q - 1$  mal stetig)

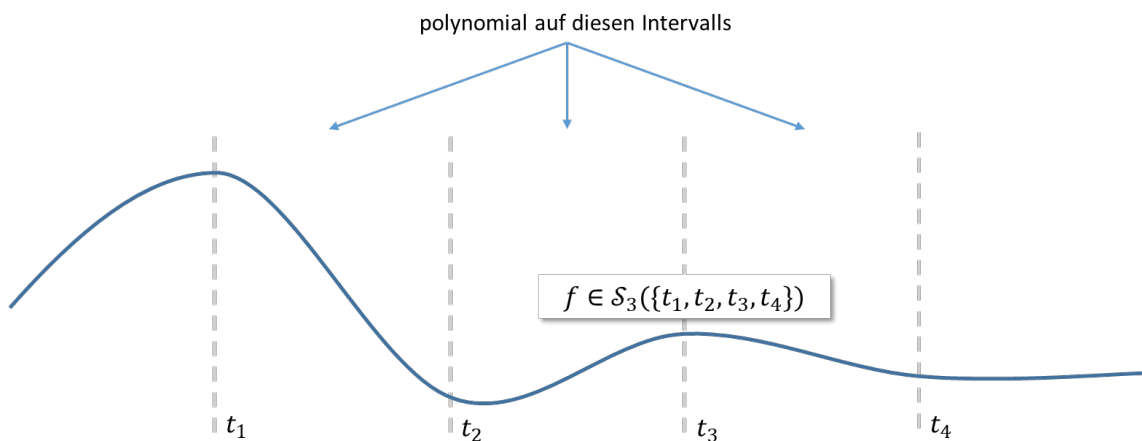


Abbildung 16: Beispiel von kubischem Spline (Spline mit Grad  $q = 3$ )

Im Rahmen dieser Arbeit werden zwei Familien von Splines betrachtet:

- B-Splines (für Basis Spline) bilden eine Basis für Splines, die null sind Außerhalb dem Intervall  $(t_1, t_r)$
- „S-Splines“ (für „Steigende Splines“) bilden eine Basis für Splines, die auf  $(-\infty, t_1]$  null sind und linear auf dem Intervall  $[t_r, +\infty)$

B-Splines kommen oft in der Literatur vor (z.B. in [23]), während der Name „S-Splines“ lediglich im Rahmen dieser Arbeit vorkommt. Allerdings sind „S-Splines“ ähnlich zu „natürliche Splines“ (siehe [22]).

Um eine Kurve zu schätzen oder Punkte zu interpolieren, sind B-Splines gut geeignet. Außerdem sind B-Splines vom Grad  $q$  nur auf  $q$  Intervalle nicht null. Dies vereinfacht die Regression.

Allerdings wurden S-Splines bevorzugt, um multivariater Regression aufzuführen. Wie natürliche Splines (siehe [22]) sind S-Splines basierten Prognose robuster für extreme Werte.

## B-Spline

Es gibt  $r - q$  B-Splines von „Grad“<sup>1</sup>  $q$ . Sie werden  $B_{q,1}, \dots, B_{q,r-q}$  gekennzeichnet und können durch die Cox-de Boor Rekursionsformel berechnet werden:

$$\forall k \in [r - q], \begin{cases} B_{1,k}(x) = \mathbf{1}_{\{t_k \leq x \leq t_{k+1}\}} \\ B_{i+1,k}(x) = \frac{x - t_k}{t_{k+i} - t_k} B_{i,k}(x) + \frac{t_{k+i+1} - x}{t_{k+i+1} - t_{k+1}} B_{i,k+1}(x) \end{cases} \quad (10)$$

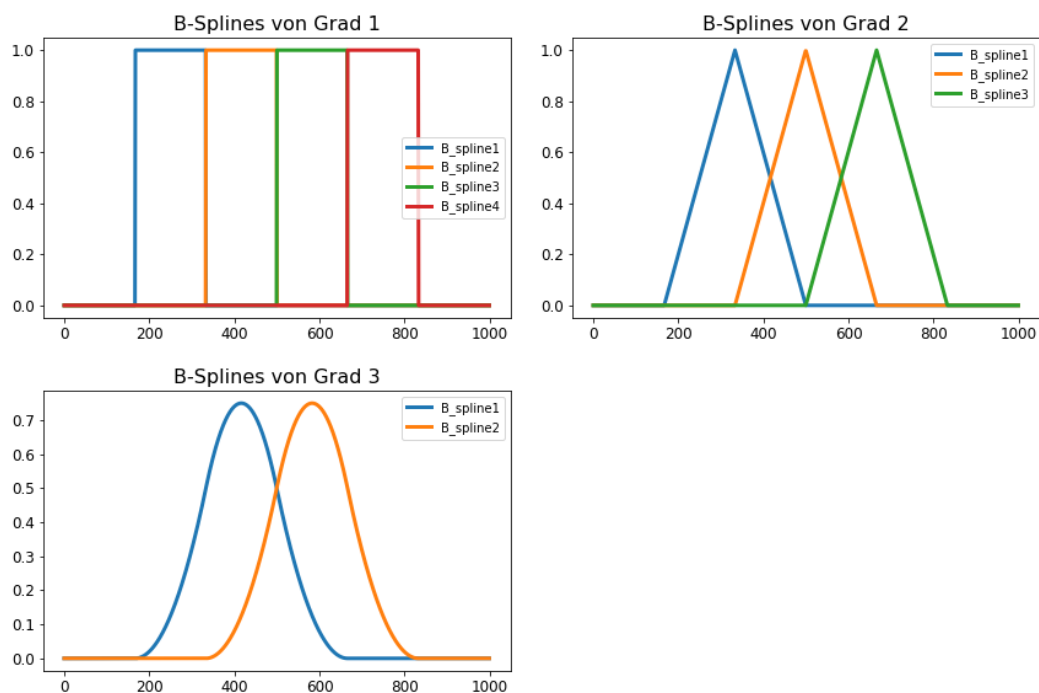


Abbildung 17: B-Splines von Grad 1,2 und 3 auf 5 Stützpunkte (167,333,450,567,833)

<sup>1</sup> Die Konvention für B-Spline ist verwirrend: sie bedeutet, dass Grad  $q$  - B-Splines stückweise polynomial von Grad  $q - 1$  sind.

## S-Splines

Die „Steigende Splines“ liegen bei null vor dem ersten Stützpunkt und sind linear nach dem letzten Stützpunkt. Der Spline  $S_{i,q}$  (von Grad  $q$ , auf Stützpunkte  $t_i$  und  $t_{i+1}$ ) wird wie folgt berechnet:

$$S_{i,q}(x) = \begin{cases} 0 & \text{wenn } x \leq t_i \\ \frac{1}{q} * \left( \frac{x - t_i}{t_{i+1} - t_i} \right)^q & \text{wenn } x \in (t_i, t_{i+1}) \\ \frac{x - t_{i+1}}{t_{i+1} - t_i} & \text{wenn } x \geq t_{i+1} \end{cases} \quad (11)$$

Sie sind in Abbildung 19 zu sehen.

## Regression mit B-Splines und S-Splines:

Die folgenden Abbildungen zeigen, wie nichtlineare Abhängigkeiten mit Hilfe von Splines nachgebildet werden in einer multivariaten Regression.

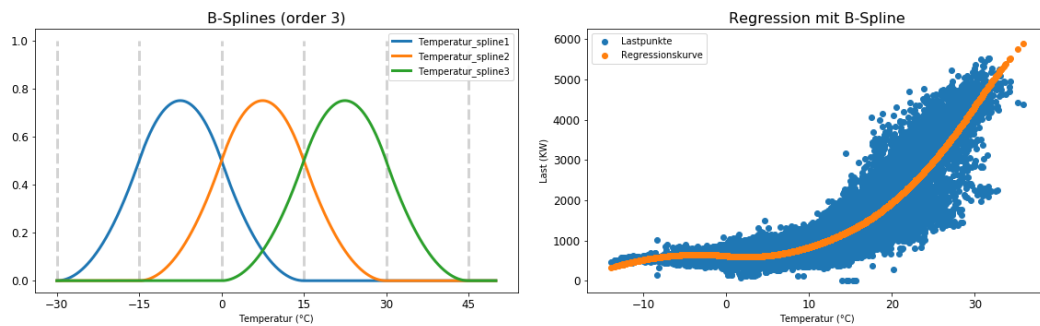


Abbildung 18: B-Splines für multivariate Regression

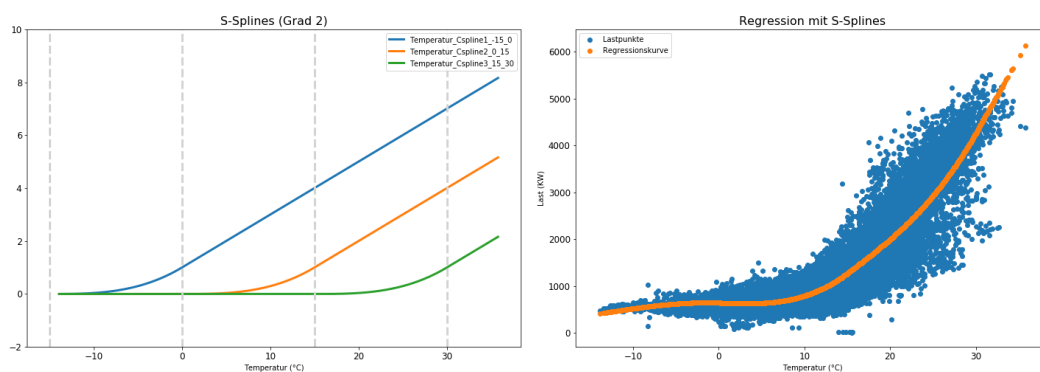


Abbildung 19: S-Splines für multivariate Regression



### 4.1.3. Lastprofil

Das Lastprofil ist eine Prognose, die nur auf Basis von kalendarischen Daten erstellt ist, und als Eingang für andere Modelle dienen sollte.

In dieser Arbeit wird die Mittlere Last für jede Stunde und Tag des Jahres berechnet, um dieses Lastprofil zu erstellen. Damit genug Datenpunkte vorliegen, ergibt sich diese mittlere Last aus einem gewichteten gleitenden Mittelwert (Konvolution) der historischen Last von „benachbarten“ Tagen ( $T_{ag} - B$  bis  $T_{ag} + B$ ), an einer bestimmten Tagesstunde und auf alle verfügbare Jahren.

Mit folgenden Gewichten  $g_t$

$$g_t = \exp\left(\frac{t}{C}\right) \times \mathbf{1}_{\{-B \leq t \leq B\}} \quad (12)$$

(C ist die Breite der „Konvolutionsfilter“ und B bestimmt die Anzahl von benachbarten Tagen)

Wird das Lastprofil  $\tilde{y}$  wie folgt berechnet:

$$\tilde{y}_t = (\mathbf{g} * \mathbf{y})_t = \frac{1}{\sum_{i=-B}^B g_i} \times \sum_{i=-B}^B g_{t-i} \times y_{t+i \times 24} \quad (13)$$

## 4.2. Multivariate Regression

### 4.2.1. Allgemeine Verfahren

Die Multivariate Regression ist ein bekanntes und einfaches Prognoseverfahren. Allerdings kann es präzise Prognose liefern, wenn geeignete Eingänge im Vorfeld ausgewählt und berechnet werden.

In diesem Verfahren wird die Referenz  $y$  mit einer gewichteten Summe der Eingangsparameter  $X_i$  dargestellt:

$$y = \sum_i c_i X_i + \epsilon \quad (14)$$

Wobei  $\epsilon$  der Prognosefehler ist.

Die optimalen Koeffizienten  $c^*$  minimieren der quadratische Fehler:

$$\begin{aligned} c^* &= \mathit{argmin}(RMSE) \\ &= \mathit{argmin}(\sum_i \epsilon_i^2) \\ &= \mathit{argmin}(\sum_t (y_t - \sum_i c_i X_i)^2) \end{aligned} \quad (15)$$

Die Darstellung kann auch in Matrixform erfolgen:

$$c^* = \mathit{argmin} \left( (Y - Xc)^T (Y - Xc) \right) \quad (16)$$

Setzt man den Gradienten Null, erhält man folgenden Zusammenhang:

$$\nabla_c RMSE = 2X^T Xc^* - 2Y^T X = 0 \quad (17)$$

Wenn  $X^T X$  invertierbar ist<sup>2</sup>, folgt:

$$c^* = (X^T X)^{-1} Y^T X \quad (18)$$

---

<sup>2</sup> Wenn nicht, kann man eine Submatrix von  $X^T X$  von maximalen Grad nutzen.

## 4.2.2. Kalendarische „Ebene“

Statt ein Model auf den gesamten Datensatz zu trainieren, ist es zu empfehlen, zwischen kalendarischen Merkmalen wie Ferientage, Stunde, und Jahreszeiten zu unterscheiden. Diesen kalendarischen Merkmalen werden hier „Ebene“ genannt. Für jede einzelne Ebene wird ein Regressionsmodel trainiert.

Dies führt zu einer deutlichen Verbesserung der Prognose.

Zum Beispiel wird mit Feiertage Ebene die Datensatz in Feiertage und Wochentage geteilt. Zwei Modellen werden dann auf die entstehenden Datensätze trainiert.

## 4.2.3. Modelle

Im Rahmen dieser Arbeit wurden verschiedene Modelle getestet, die auf multivariater Regression basieren. Der Unterschied zwischen diese Modelle liegt in der Auswahl von Eingängen und von „Ebene“.

Es werden hier nur fünf Modellen vorgestellt, die nacheinander erstellt wurden. Das sollte die Vorgehensweise widerspiegeln. Nachdem ein Model erstellt und bewertet wurde, stellt sich die Frage, welche Verbesserungen nötig wären. In dem nächsten Schritt führt diese Überlegung zu einem anderen Model.

(siehe Tabelle 2 für eine Kurzfassung den Modellen mit deren Prognosefehler)

### **Model 1:**

In der Explorative Datenanalyse hat sich gezeigt, dass die Temperaturabhängigkeit ab 15 Grad zunimmt. In den ersten Model werden also drei Eingänge ausgewählt, um die nichtlineare Abhängigkeit zwischen Last und Temperatur abzubilden: *Temperatur* , *Temperatur<sup>2</sup>* und  $\max(T - 15, 0)$ .

### **Model 2:**

Zusätzliche Inputs werden hinzugefügt im Vergleich zu Model1: Enthalpie, Luftfeuchte, Wind, Globalstrahlung.

Außer der Temperatur, Luftfeuchte und Enthalpie haben allerdings die anderen Wetterdaten die Prognose nicht verbessert, so dass sie nicht in den nächsten Modellen vorkommen.

**Model 3:**

In diesem Model werden sechs „Ebene“ eingeführt. Es wird zwischen drei Jahreszeiten und zwischen Feiertage und Wochentage unterschieden.

(Um die Grenzen dieser Jahreszeiten zu bestimmen, wurden über 150 Wochen auf Basis der Ist-Last geclustert).

Diese Ebene führen zu einer deutlichen Verbesserung.

**Model 4:**

Zusätzlich zu Wetter basierten Eingänge verfügt dieses Model über andere berechnete Eingangsparameter:

- Ein Lastprofil (siehe Abschnitt 4.1.3)
- Cosinus und Sinus mit Perioden von einem Tag und einem Jahr. Damit kann ein Lastprofil vom Model selbst berechnet werden (siehe Abschnitt 4.1.2).
- Die Mitteltemperatur vor zwei, drei, vier und sieben Tagen

Die Lastprofil bringt eigentlich keine Verbesserung, so dass es nicht weiterverwendet wird, im Gegenteil zu historischen Werten.

**Model 5**

Dieses Model benutzt andere Ebenen. 168 statt sechs Regressionen werden berechnet, für jeden Wochentag und jede Stunde.

Das führte zu einer deutlichen Verbesserung der Prognose. Leider werden die Regressionen dadurch anfälliger für extremen Werte, da jede einzelne Regression weniger Daten zum Trainieren hat.

## 4.2.4. Ergebnisse

Die Modelle werden mit Gleitender Horizont Methode, bezogen auf ein Jahr (11.2017 bis 10.2018) und mit Perioden von je zwei Wochen, bewertet (siehe 2.4).

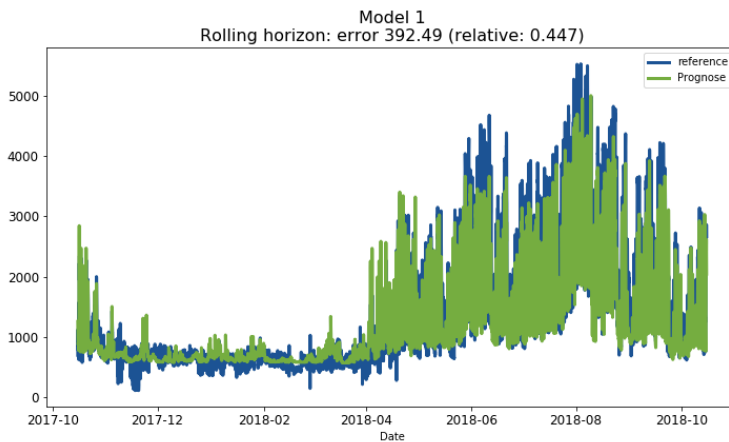


Abbildung 20: Prognose von Model 1

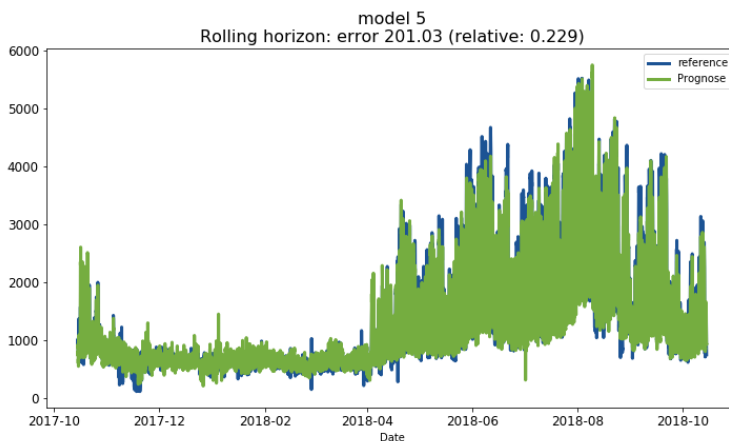


Abbildung 21: Lastprognose von Modelle 5

Mit dieser schrittweisen Erstellung und Analyse den Modellen wird erkannt, welche Eingänge und Ebene geeignet sind. Dadurch verbessert sich der Prognosefehler. Bei der Betrachtung von historischen Werten und bei der Unterscheidung zwischen den einzelnen Wochentagen und Stunden, erfolgt die größte Verbesserung der Prognosegüte.

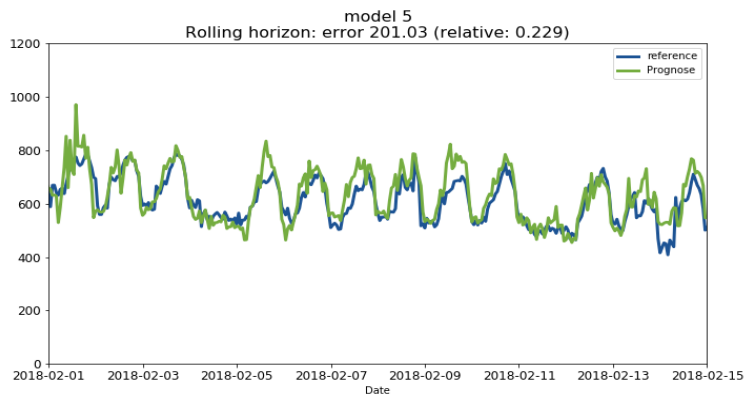
In der folgenden Tabelle werden die Hauptresultate dargestellt.

**Tabelle 2: Multivariate Regression - Ergebnisse**

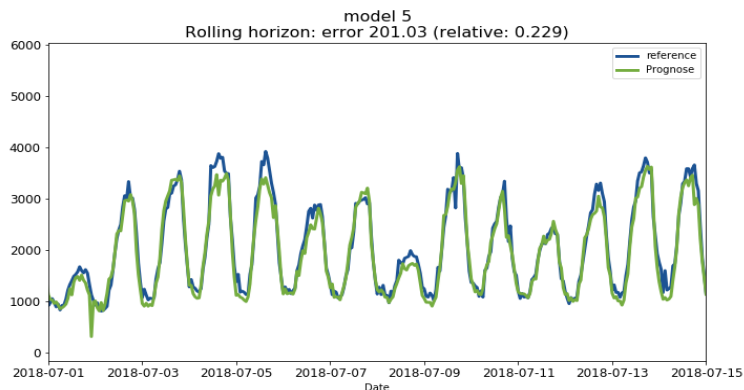
		Model 1	Model 2	Model 3	Model 4	Model 5
Eingangsparameter	Temperatur	✓	✓	✓	✓	✓
	Luftfeuchte		✓	✓	✓	✓
	Andere Wetterdaten		✓			
	Lastprofil				✓	
	Daten der vorherigen Tage				✓	✓
Ebene	Jahreszeiten			✓	✓	
	Wochentag vs. Sonntag/Feiertag			✓	✓	
	7 Wochentage					✓
	Stunden					✓
Fehler	Absolute (RMSE in kW)	392,49	385,33	323,19	258,78	201,03
	Relative	44,7 %	43,9 %	36,8 %	29,5 %	22,9 %

Die folgenden Abbildungen der Prognose von Model 5 deuten an weitere Verbesserungsmöglichkeiten, wie zum Beispiel eine Glättung der Prognose.

Solche Verbesserungen werden im Abschnitt 5.1 eingeführt.



**Abbildung 22: Lastprognose Model 5 (Winter)**



**Abbildung 23: Lastprognose Model 5 (Sommer)**

## 4.3. Support Vektor Regression

### 4.3.1. Allgemeines Verfahren

Das Prinzip der Support Vektor Regression ist wie folgt beschreibbar: Ziel ist es, ein Regressionsverfahren zu generieren, welches eine  $\epsilon$  - Abweichung zulässt. Zudem sollen nichtlineare Abhängigkeiten berechnet werden können, ohne im Vorfeld Basisinterpolierungsfunktionen erstellen zu müssen.

Die folgende Einführung bezieht sich auf [24].

Dafür wird mit einem linearen Regressionsproblem begonnen:

$$f(x) = \sum_{i \in I} c_i x_i + b \quad (19)$$

Mit  $f(x)$  die Prognose,  $c_i$  die Gewichte und  $b$  eine Konstante.

Die Funktion kann ebenso mit einem Skalarprodukt beschrieben werden:

$$f(x) = \langle c, x \rangle + b \quad (20)$$

Das führt zu folgender Problemstellung:

$$\begin{aligned} \min \quad & \frac{1}{2} \|c\|^2 \\ & \begin{cases} y_i - \langle c_i, x_i \rangle - b \leq \epsilon \\ -y_i + \langle c_i, x_i \rangle + b \leq \epsilon \end{cases} \end{aligned} \quad (21)$$

Die quadratische Summe der Gewichte wird minimiert, sodass alle Beobachtungen nicht weiter als im Abstand von  $\epsilon$  von der Prognose entfernt liegen.

Die folgenden Berechnungsschritte werden hier nur grob beschrieben. Siehe Anhang I für die ausführliche Berechnung.

Um Unzulässigkeiten zu vermeiden, können die Bedingungen von Formel 21 aufgelockert werden (Häufig können nicht alle Punkte innerhalb des  $\epsilon$ -Korridor liegen).

Wenn das Dualproblem formuliert wird, fällt auf, dass es lediglich vom Skalarprodukt der Eingänge abhängig ist.

Mit einer Änderung des Skalarprodukts, kann das Dual in einen viel größeren Vektorraum umformuliert werden. Das ist äquivalent zur Berechnung von zusätzlichen Eingängen, ohne den rechnerischen Aufwand.

Das neue Skalarprodukt  $\langle \cdot, \cdot \rangle_\phi$  wird Kernel genannt. Dieser Ansatz wird oft als „Kernel Trick“ beschrieben. Am Ende erhält man den modifizierten Dual:

$$\max -\frac{1}{2} \sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle_\phi - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \quad (22)$$

$$\begin{cases} \sum_i (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

Wobei  $\alpha_i$  und  $\alpha_i^*$  die Dualvariablen sind.

Die Prognose  $f(x)$  lautet dann:

$$f(x) = \sum_{x_i \in \text{Training}} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle_\phi + b \quad (23)$$

Wobei  $\alpha_i - \alpha_i^* \neq 0$  lediglich gilt, wenn die Beobachtung  $x_i$  außerhalb des  $\epsilon$ -Korridors um die Prognose liegt. Diese Punkte werden „Support Vektor“ benannt.

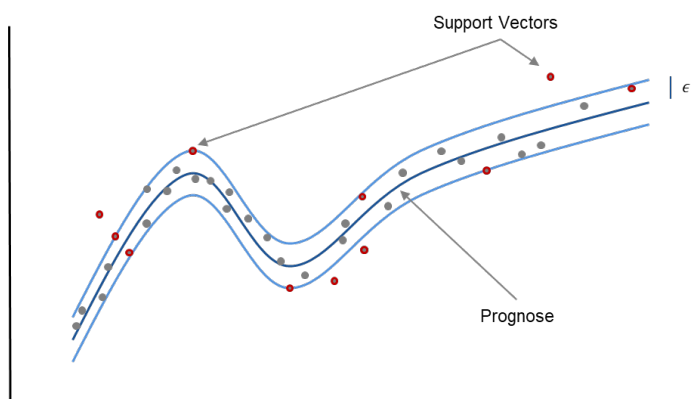


Figure 1:  $\epsilon$ -Korridor und Support Vektoren bei SVR



### 4.3.2. Implementierung

Es wird zuerst versucht, das SVR-Verfahren in Python selbst zu implementieren.

In [24] wird ein Interior-Point Algorithmus beschrieben. Die Implementierung in Python hat demzufolge fehlerhafte Ergebnisse ergeben. Der Algorithmus selbst war sehr aufwendig und unklar. Das Prinzip des Interior-Point Algorithmus ist, dass versucht wird, gleichzeitig das Problem und sein Dual zu optimieren, indem die Dualitätslücke (Dual Gap) reduziert wird.

Anschließend wird ein Newton Minimierungsalgorithmus implementiert, mit schrittweiser Projektion auf den zulässigen Bereich. Hier war die Projektion problematisch.

In dem Newton Algorithmus wird an jedem Schritt eine Richtung  $dx$  berechnet und die beste Lösung  $x$  entsprechend aktualisiert:

$$x_{t+1} = x_t + dx \quad (24)$$

Damit  $x_{t+1}$  noch zulässig ist sollte die Richtung  $dx$  innerhalb eines polyedrischen Kegels  $K$  liegen. Also wird der Gradient der zu minimierenden Funktion  $\Delta f$  auf  $K$  projiziert.

$$dx = \arg \min_{y \in K} |\Delta f - y|^2 \quad (25)$$

Was zu folgendem äquivalentem Problem führt:

$$\tilde{dx} = \arg \max_{\substack{y \in K \\ |y| \leq 1}} \langle \Delta f, y \rangle \quad (26)$$

Wobei  $\tilde{dx}$  die normierte Richtung ist.

Ein Simplex Algorithmus wurde implementiert, um dieses Problem zu lösen. Die Bedingung  $|y| \leq 1$  wurde iterativ durch lineare Bedingungen approximiert.

Aufgrund der langen Berechnungszeit wurde zunächst beschlossen, die Projektion durch zwei Projektionen zu ersetzen. Die daraus entstandenen Ergebnisse waren aber stark abhängig vom Startpunkt des Newton Algorithmus:

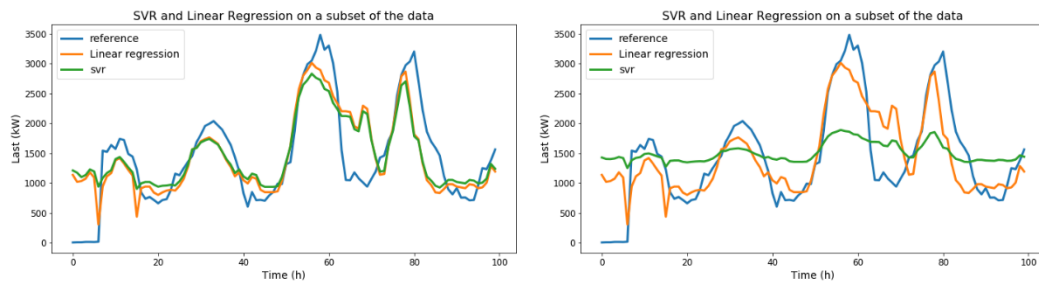


Abbildung 24: SVR Algorithmus mit gutem und falschem Startpunkt

Letztendlich wurde eine Python Funktion benutzt (`sklearn.svm.SVR`).

### 4.3.3. Modelle

Es konnten drei Kernels getestet werden, mit den selben Eingangsparametern wie im Model 4:

- Polynomial Kernel (mit Exponent 3): äquivalent zu einer Regression auf alle Multinome dritten Grades als Eingangsparameter.

$$\langle \mathbf{x}, \mathbf{y} \rangle_{poly} = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}^2, \mathbf{y}^2 \rangle + \langle \mathbf{x}^3, \mathbf{y}^3 \rangle \quad (27)$$

- RBF Kernel (Radial Basis Funktion): äquivalent zu einem Vektorprodukt in einem unendlich-dimensionalen Raum [25].

$$\langle \mathbf{x}, \mathbf{y} \rangle_{rbf} = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \quad (28)$$

- Sigmoid Kernel:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{sigmoid} = \tanh(\gamma \langle \mathbf{x}, \mathbf{y} \rangle + r) \quad (29)$$

### 4.3.4. Ergebnisse

Die Ergebnisse waren nicht zufriedenstellend, die Prognosen erfuhren keine deutliche Verbesserung im Vergleich zur multivariaten Regression. Oftmals waren die Ergebnisse schlechter. Hinzu kam der höhere Rechneraufwand.

		Linear Model 4	SVR Polynomial	SVR Sigmoid	SVR RBF
Fehler	Absolut (kW)	258,78	283,25	2735,62	253,82
	Relativ	29,5%	32,3%	311,7%	28,9%

Tabelle 3: Support Vektor Regression - Ergebnisse

Diese vergleichswisen schlechten Ergebnisse lassen sich vermutlich durch folgende Punkte erklären:

- Die Ist-Last selbst ist nicht sonderlich glatt Daher macht es wenig Sinn, einen  $\epsilon$ -Korridor zu erlauben. Die meisten Punkte werden außerhalb des Korridors liegen, somit ist keine Verbesserung zu erwarten.
- Es gibt wenige Rohdaten-Eingänge, so dass es möglich ist, alle benötigten zusätzlichen Eingangsparameter im Voraus zu berechnen.
- Möglicherweise gibt es Overfitting. Das Modell ist zu gut an die Trainingsdaten angepasst, der Verallgemeinerungsfehler wird erhöht.

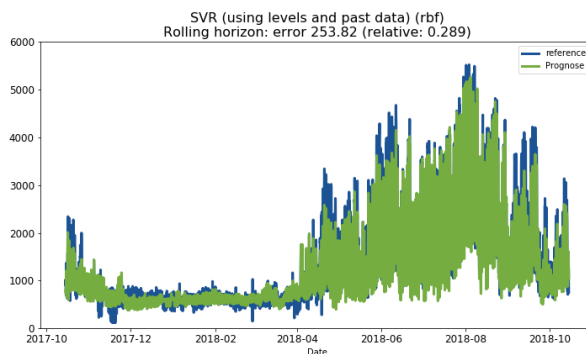


Abbildung 25: SVR Prognose mit RBF Kernel

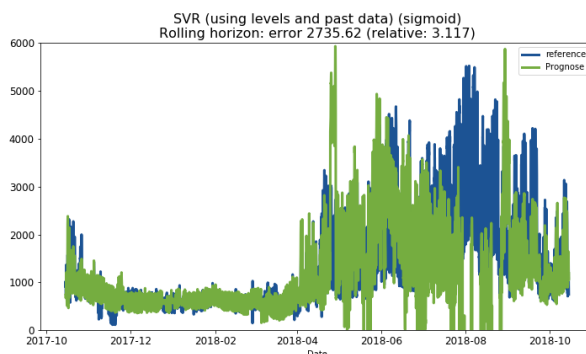


Abbildung 26: SVR Prognose mit Sigmoid Kernel

## 4.4. K-nächste Nachbarn

### 4.4.1. Allgemeine Verfahren

In K-nächste Nachbarn Algorithmen werden  $K$  historische Datenpunkte ausgewählt, die ähnlich den zu prognostizierenden Punkten sind. Es wird eine Prognose erstellt, üblicherweise auf Basis der Mittelwerte dieser ähnlichen Punkte.

Lediglich die Messung der Ähnlichkeit tritt hier als einzige Schwierigkeit auf. Diese Distanz sollte sich der Entfernung zwischen den Referenzwerten (die Kältelast) annähern.

Eine Zeitreihe, wie beispielsweise die Temperatur, oder eine gewichtete Summe der Eingangszeitreihen, kann als Ähnlichkeitsmaß verwendet werden. Es ist auch möglich, die Distanz anhand einer vorliegenden Prognose zu messen.

### 4.4.2. Modelle

Zunächst gilt es, ein Ähnlichkeitsmaß zu finden, welches den Lastunterschied zweier Tage ähnelt:

$$\mathit{dist}(\mathit{Tag}_i, \mathit{Tag}_j) \sim f(\mathit{Last}_i - \mathit{Last}_j) \quad (30)$$

Es werden drei Ansätze näher untersucht, die eine zunehmende Komplexität aufweisen:

- **Model 1:** Die Distanz beruht auf nur einer Zeitreihe, in diesem Fall der Temperatur.

$$\mathit{dist}(\mathit{Tag}_i - \mathit{Tag}_j) = \left\| \mathit{Temperatur}_i - \mathit{Temperatur}_j \right\|^2 \quad (31)$$

- **Model 2:** Die Distanz basiert auf der Prognose eines multivariaten Regressionsmodells.

$$\mathit{dist}(\mathit{Tag}_i - \mathit{Tag}_j) = \left\| \mathit{Prognose}_i - \mathit{Prognose}_j \right\|^2 \quad (32)$$

- **Model 3:** Die Distanz wird von einem neuronalen Netz gelernt.

$$\mathit{dist}(\mathit{Tag}_i, \mathit{Tag}_j) = \left\| \mathit{y}_i - \mathit{y}_j \right\|^2 \quad (33)$$

Mit  $y_i = \text{model}(\text{Tag}_i)$

Das Netz soll ein sogenanntes „Embedding“ lernen: die Eingangszeitreihen werden in einen anderen Vektorraum umwandelt, wo das Skalarprodukt die Lastunterschied entspricht.

Mit  $p$  Eingänge lässt es sich do beschreiben:

$$\text{Netz} : \mathbb{R}^p \rightarrow \mathbb{R}^m \quad \text{so, dass } \forall i, j \|\text{Netz}(X_i) - \text{Netz}(X_j)\| \approx |y_i - y_j| \quad (34)$$

$$X \rightarrow Z$$

Wobei  $p$  die Anzahl von Eingängen ist,  $m$  die Dimension der Embedding Raum,  $X_i, X_j$  die Eingänge an den Zeitpunkten  $i$  und  $j$  mit jeweiligen last  $y_i$  und  $y_j$ .

Um das Netz zu trainieren, wird mini-Batching benutzt.

- Es werden Paaren von Tagen  $(i_1, j_1), (i_2, j_2), (i_3, j_3) \dots$  betrachtet.
- Die jeweiligen Ausgänge  $(Z_i, Z_j) = (\text{Netz}(X_i), \text{Netz}(X_j))$  werden berechnet.
- Der Unterschied zwischen diese Ausgänge wird mit der Lastunterschied verglichen:

$$\text{Fehler} = \sum_k (|y_{i_k} - y_{j_k}| - \|Z_{i_k} - Z_{j_k}\|)^2 \quad (35)$$

- Zum Schluss wird diese Fehler durch Backpropagation minimiert

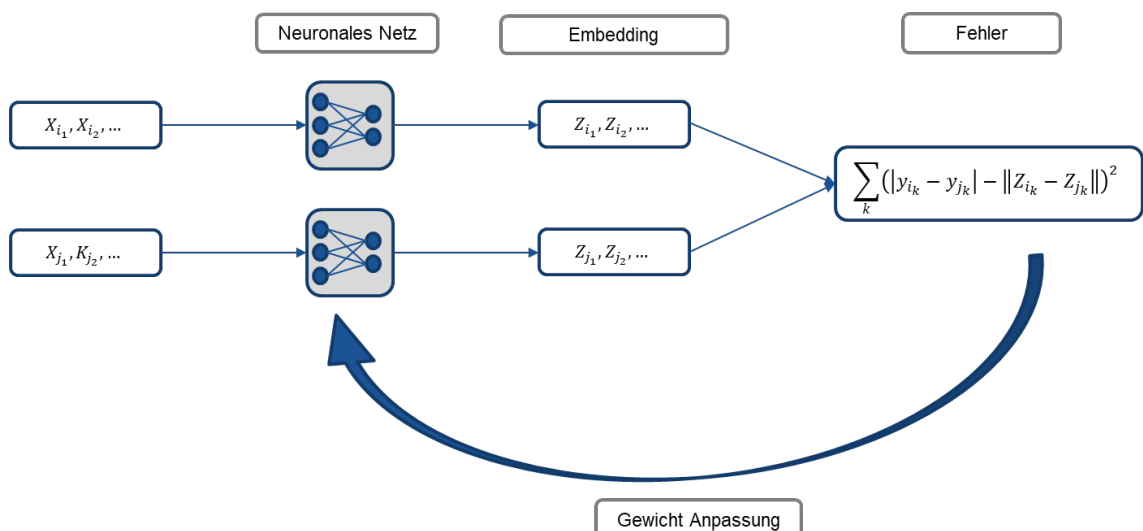


Abbildung 27: Neuronales Netz Ansatz für K-nächst-Nachbarn



### 4.4.3. Ergebnisse

Unter den Daten werden die 10 Tage gesucht, die dem zu prognostizierenden Tag am ähnlichsten sind, und anschließend ein Mittelwert als Prognose genommen.

Die letzte Möglichkeit gab zuerst bessere Prognosen mit Cross-Validation.

		Model 1 Temperatur	Model 2 Prognose	Model 3 ANN
Cross-Validation Fehler	Absoluter (kW)	313,54	308,27	242,40
	Relativer	35,7%	35,1%	27,6%

Tabelle 4: K-nächste Nachbarn - erste Ergebnisse mit Cross-Validation

Mit dem besseren linearen Model 5 und der Gleitender Horizont Bewertung hat das Model 2 eine bessere Prognose erstellt.

Für die Analyse ist es auch interessant, den Algorithmus auf Basis der Ist-Last laufen zu lassen („Model 0“). Damit ergibt sich eine untere Grenze für die Prognosefehler von K-nächste Nachbarn Modelle (im besten Fall wird diese Fehler erreicht).

In dem Anwendungsfall ist diese Grenze niedrig genug, dass es Sinn macht, K-nächste Nachbarn zu implementieren.

		Model 0 Ist-Last	Model 1 Temperatur	Model 2 Prognose	Model 3 ANN
Fehler	Absoluter (kW)	130,76	346,54	208,99	235,03
	Relativer	14,9%	39,5%	23,8%	26,8%

Tabelle 5: K-nächste Nachbarn - Ergebnisse

Diese Ergebnisse sind schlechter, als denen von Multivariate Regression. Wie bereits dargestellt, weist das lineare Model 5 einen Fehler von 22,9 % auf.

Für den Einsatz von K-nächste Nachbarn sprechen trotzdem zwei Gründe. Zuerst gibt es keine Ausreißer, da es sich nur um eine Summe von historischen Werten handelt. Zweitens sind diese Modelle komplementär. Dies bedeutet, dass bei der Bildung der Mittelwerte beider Modelle eine Verbesserung zu erwarten ist (siehe 5.1.6).

Man kann sich allerdings vorstellen, dass solche Verfahren nicht sehr gut mit dem Wachstum der Kältenachfrage umgehen können. Daher wäre es sinnvoll, eine wachstumsbereinigte Ist-Last als Eingang zu nutzen. Dieser Versuch ist nicht implementiert worden.

## 4.5. Neuronale Netze

### 4.5.1. Allgemeine Verfahren

Heutzutage sind neuronale Netze ein häufig diskutiertes Thema. Die Hauptidee selbst ist jedoch nicht revolutionär. Es werden einfache, bekannte Funktionen verwendet, um eine komplexere Funktion aufzubauen. Mit den Gradienten der Teilfunktionen, den sogenannten Neuronen, kann der Gradient zügig berechnet werden, um die Kostenfunktion effizient zu minimieren und angepasste Gewichte zu berechnen.

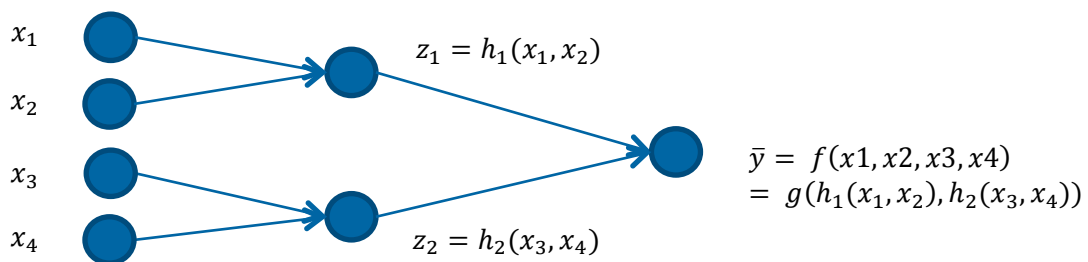


Abbildung 28: Ein „einfaches neuronales Netz“

In dem Beispiel  $\frac{\partial \bar{y}}{\partial x_1} = \frac{\partial z_1}{\partial x_1} \frac{\partial \bar{y}}{\partial z_1}$  wird effizient mit den Gradienten von den Teilfunktionen  $h_1$  und  $g$  berechnet.

Die neuronalen Netze folgen üblicherweise einem ähnlichen Aufbau. Für jedes Neuron wird zuerst eine gewichtete Summe der Eingänge berechnet. Anschließend wird das Ergebnis in eine Aktivierungsfunktion gegeben, die den Ausgang des Neurons berechnet.

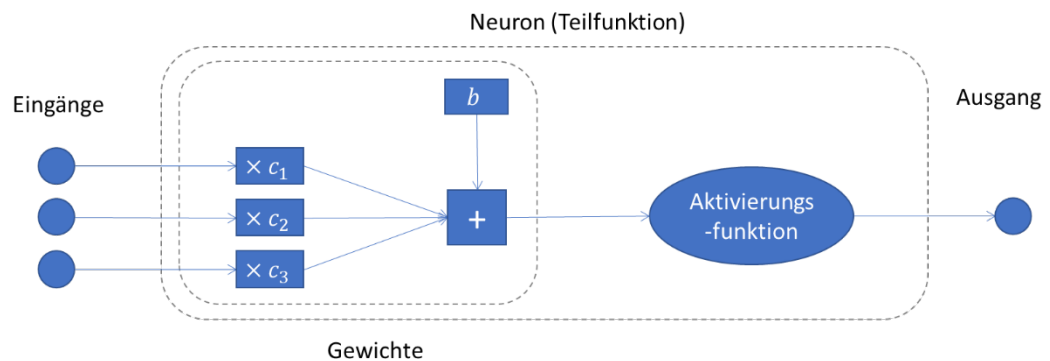


Abbildung 29: Beispiel eines künstlichen Neurons mit Gewichten  $c_1, c_2, c_3$  und  $b$



Als Aktivierungsfunktionen werden in der Regel eine der folgenden eingesetzt:

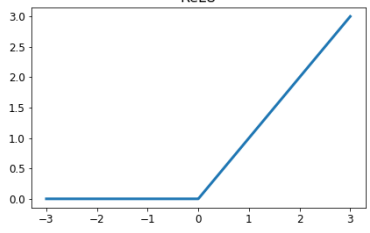
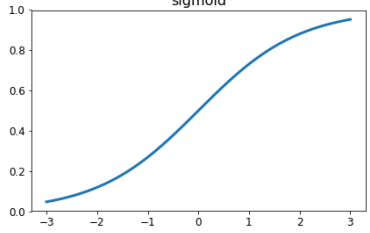
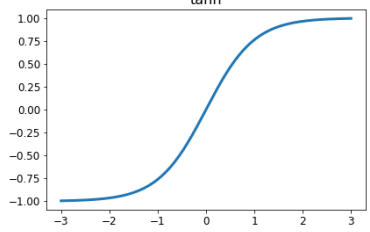
<ul style="list-style-type: none"> <li>ReLU (Rectified Linear Unit)</li> </ul>	$x \rightarrow \max(x, 0)$	
<ul style="list-style-type: none"> <li>Sigmoid</li> </ul>	$x \rightarrow \frac{1}{1 + e^{-x}}$	
<ul style="list-style-type: none"> <li>Tanh (Tangens hyperbolicus)</li> </ul>	$x \rightarrow \frac{e^x - e^{-x}}{e^x + e^{-x}}$	

Tabelle 6: Aktivierungsfunktionen

In der Literatur wird zwischen Feedforward und rekursiven Netzen unterschieden.

Letztgenannte speichern Informationen über die Vergangenheit, um eine bessere Prognose erstellen zu können.

Für diesen Anwendungsfall werden alle Modelle mit dem Python Package „PyTorch“ entwickelt und getestet.

## 4.5.2. Modelle

Für diesen Anwendungsfall werden vier Modelle erstellt:

- **Model 1:** Einfaches Feedforward Netz mit zwei versteckten Schichten.
- **Model 2:** Long Short-Term Memory Netz (LSTM).
- **Model 3:** Netz mit „Zeitfaktoren“. Unterscheidung zwischen historischen, kalendari-schen und wetterabhängigen Daten.
- **Model 4:** Physisches Netz, das das Verhalten eines Kunden besser abbildet.

**Model 1:** Das einfache Feedforward Netz (auch Perzeptron genannt) sollte als Referenzmodell dienen. Jedoch ist es nicht sonderlich gut an das Problem angepasst. Aus diesem Grunde wird folgende Architektur gewählt:

- Erste versteckte Schicht: Zehn Neuronen mit „ReLU“ (Rectified Linear Units) als Aktivierungsfunktionen.
- Zweite versteckte Schicht: Fünf Neuronen mit ReLU.

Das bedeutet, dass für jedes Neuron in den versteckten Schichten die Ausgangswerte wie folgt berechnet werden:

$$Ausgang_j = \max\left(0, c_{j,0} + \sum_i c_{i,j} Eingang_i\right) \quad (36)$$

Wobei die Gewichte  $c_{j,0}$  und  $c_{j,i}$  während des Trainingsprozesses angepasst werden.

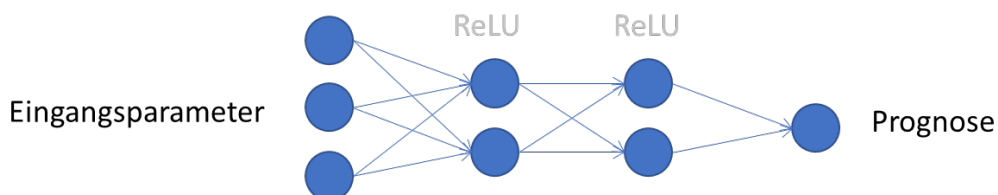


Abbildung 30: Einfaches Feedforward Netz (Perzeptron)

Die Eingänge bestehen sowohl aus Wetterdaten sowie historischen und kalendari-schen Daten.

Es werden Jahreszeiten Ebene benutzt. Wegen der hohen Zahl von Koeffizienten soll-te es nämlich vermeiden werden, zu viele Ebene anzuwenden, um Overfitting einiger-maßen zu beschränken

Verschiedene Zahlen von Neuronen in den versteckten Schichten sind auch getestet worden.

**Model 2:** LSTM neuronale Netze sind rekursive Netze (siehe Abbildung), die oft verwendet werden, um Sequenzen zu prognostizieren.

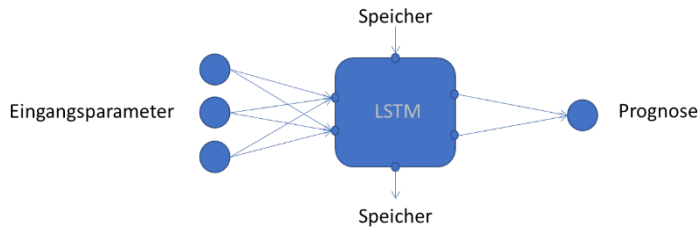


Abbildung 31: LSTM Netz

Diese Netze sind schwer zu trainieren, was oft problematisch ist.

- Statt Prognosen für jeden Zeitpunkt parallel zu berechnen, muss die Prognose nacheinander erstellt werden. Nur so kann die gespeicherte Information genutzt werden.
- Da der erste Input das Ergebnis des letzten Tages durch diese gespeicherte Information beeinflusst, ist der Gradient aufwändig zu berechnen.

Um das erste Hindernis zu umgehen, werden die Zeitreihen für das Training mithilfe des sogenannten „Batching“ Prozesses in kleinere Sequenzen geteilt.

LSTM Netze bieten eine Antwort auf das zweite beschriebene Problem. Sie liefern einen Mechanismus, der überflüssige Informationen löscht [26] [27].

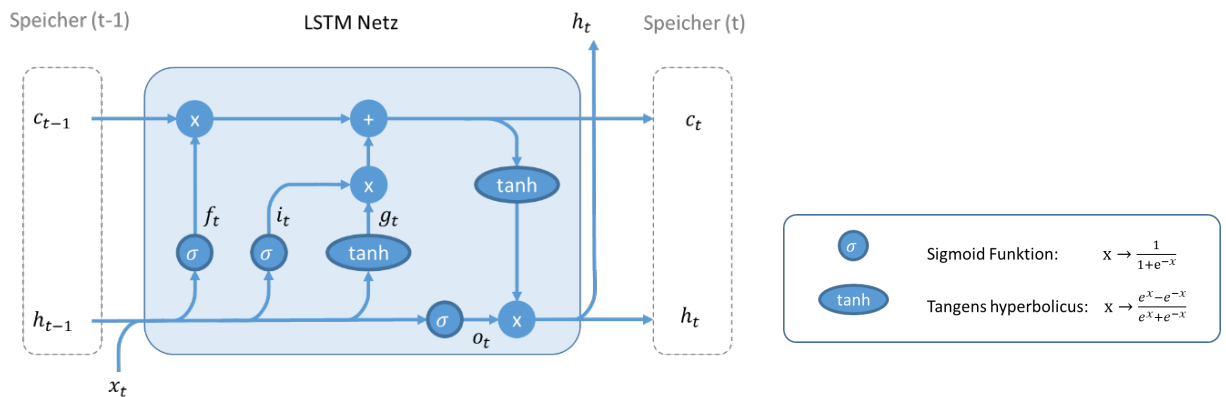


Abbildung 32: LSTM Architektur

$x_t$ : Eingang

$h_t$ : Ausgang

$c_t$ : Zellenzustand

$f_t$ : Forget Tor: Ob der vorherige Zellenzustand vergessen werden sollte

$i_t$ : Input Tor: Ob der Zellenzustand aktualisiert werden sollte

$g_t$ : "Content" Tor: Aktualisierung des Zellenzustandes

$o_t$ : Output Tor: Ausgangsentwurf

Für den Anwendungsfall wird ein zehndimensionales LSTM getestet

### Model 3: Netz mit Zeitfaktoren

Anhand der bisherigen Ergebnisse wird eine angepasste Netzarchitektur vorgeschlagen, bei der Ausgangsentwürfe durch kalendarische Daten („Faktoren“) selektiert werden:

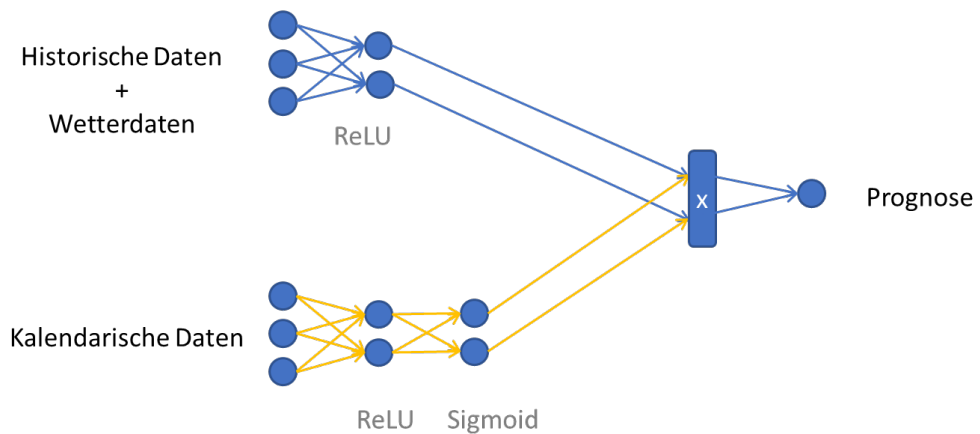


Abbildung 33: Netz mit Zeitfaktoren

Statt für jede Stunde und jeden Tag ein Model zu bauen, wie im Linear Model 5, werden die kalendarischen Daten in das Netz integriert: Die historischen Daten und wetterbasierten Prognoseentwürfe werden durch kalendarische Faktoren multipliziert und ausgewählt.

Grundsätzlich werden ReLU Aktivierungsfunktionen benutzt, um nichtlineare Funktionen abbilden zu können. Die Ausgänge von Sigmoids liegen zwischen 0 und 1, deshalb werden sie benutzt, um multiplikative Zeitfaktoren zu bilden.

Verschiedene Varianten des Netzes mit Zeitfaktoren wurden getestet, mit z.B. getrennten historischen und Wetterdaten oder mit einem LSTM Neuron am Ende des Netzes. Diese Netze sind in den folgenden Abbildungen dargestellt. Sie haben aber schlechtere Prognose berechnet.

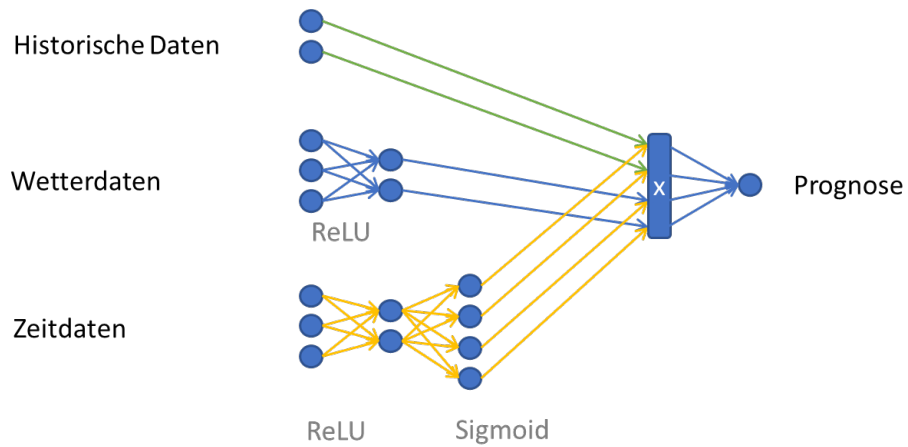


Abbildung 34: Netz mit Zeitfaktoren und getrennten historischen und Wetterdaten

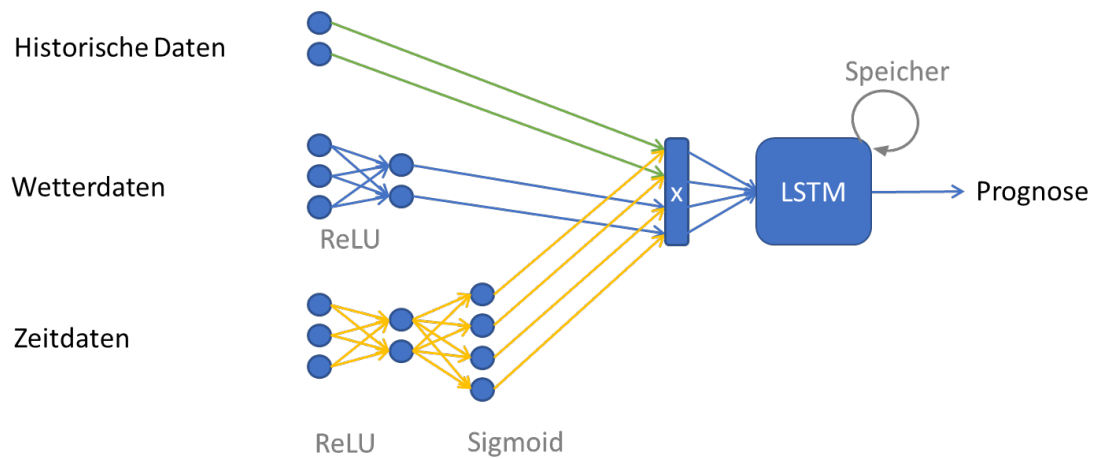


Abbildung 35: Netz mit Zeitfaktoren und LSTM

### 4.5.3. Training und gezielte Verbesserungen

Das Training ist bei neuronalen Netzen komplizierter als bei linearer Regression und besonders bei rekursiven Netzen. Deswegen wird das LSTM Netz anders als die anderen beiden Netze trainiert.

Neuronale Netze werden am häufigsten mit Gradientenverfahren trainiert. Es wird zuerst der Gradient des Fehlers durch Backpropagation ermittelt, dann werden die Parameter angepasst, um diese Fehler zu reduzieren:

$$\mathbf{Parameter}_{n+1} = \mathbf{Parameter}_n - \lambda \cdot \nabla \mathbf{Fehler} \quad (37)$$

Mit  $\lambda$  als Lernrate.

Besonders wichtig ist es, eine angepasste Schrittweite für den Gradientenverfahren Algorithmus zu wählen. Die sogenannte Lernrate wird mit dem Adam (Adaptive Moment Estimation) Verfahren bestimmt, in dem die Lernrate an jeden Schritt angepasst wird, um den Bias (systematischer Fehler) und Varianz des Gradienten zu berücksichtigen [28]. Diesen Verfahren war schon in dem PyTorch Package implementiert.

Das LSTM wird mit „Mini-Batching“ trainiert. Für jeden Schritt wird nur ein Teil des Datensatzes benutzt, um den Gradienten zu berechnen. Es ist für rekursive Netze besonders gut geeignet, anderweitig ist die Berechnung des Gradienten über die gesamte Periode sehr aufwendig. Um Rechenzeit zu sparen, ohne die Trainingseffizienz zu beeinträchtigen, werden am Anfang viele kleine Datensätze benutzt und am Ende wenige länger dauernde.

Für das Netz mit Zeitfaktor werden zwei Netze parallel trainiert und anschließend das beste der beiden für die Prognose benutzt. Zudem wird der Verallgemeinerungsfehler mit Cross-Validation berechnet und als Abbruchparameter benutzt.

#### 4.5.4. Ergebnisse

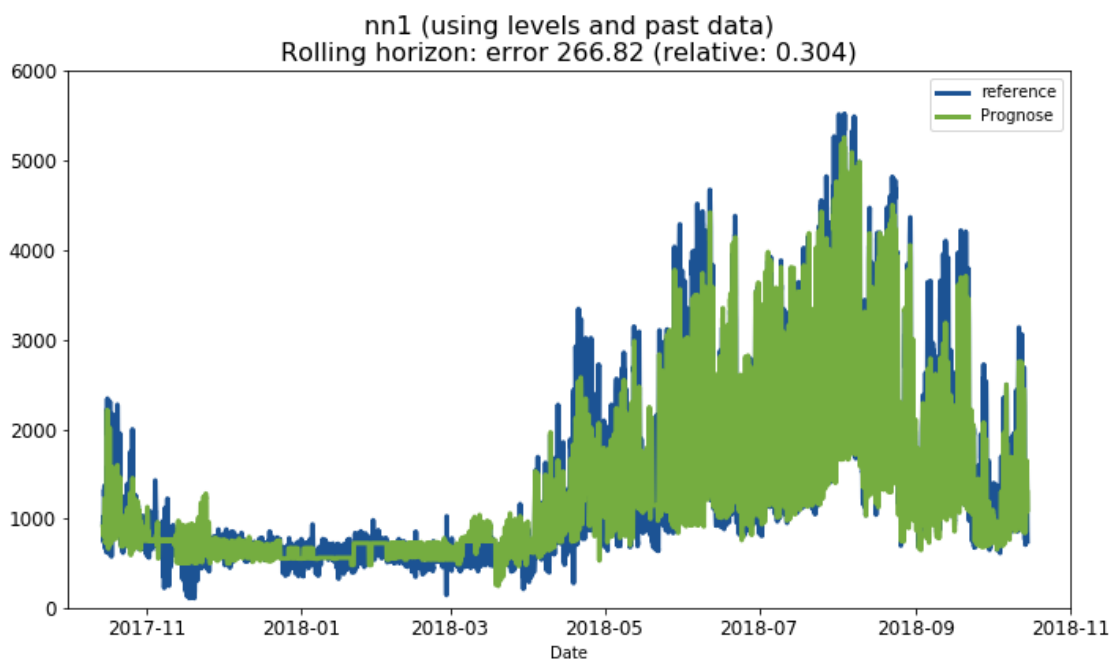
Die drei Netze werden mit den üblichen Verfahren getestet.

Die Ergebnisse beim LSTM Netz waren nicht so gut wie erwartet. Das Netz mit Zeitfaktoren hat besonders erfolgreich die kalendarischen Abhängigkeiten nachvollzogen.

Diese Ergebnisse werden in den folgenden Figuren vorgestellt.

		Model 1	Model 2	Model 3
		Perzeptron	LSTM	Zeitfaktoren
Fehler	Absoluter (kW)	266,82	400,31	197,27
	Relativer	30,4%	45,6%	22,5%

**Tabelle 7: Neuronale Netze - Ergebnisse**



**Abbildung 36: Prognose mit einfachem Feedforward Netz**

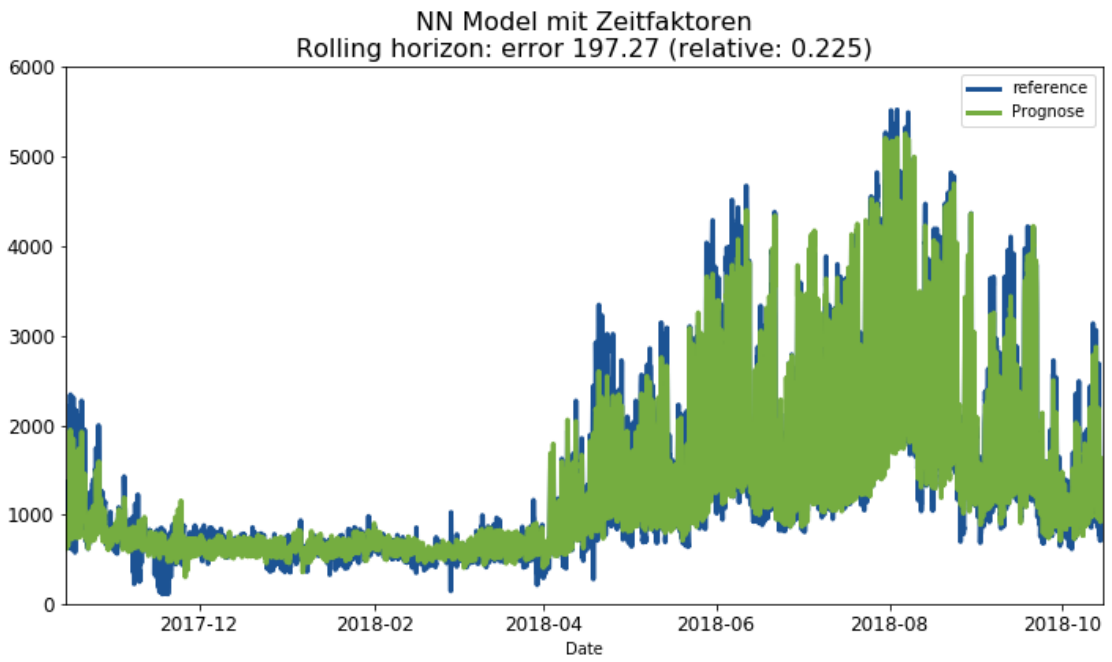


Abbildung 37: Prognose mit Zeitfaktor Netz

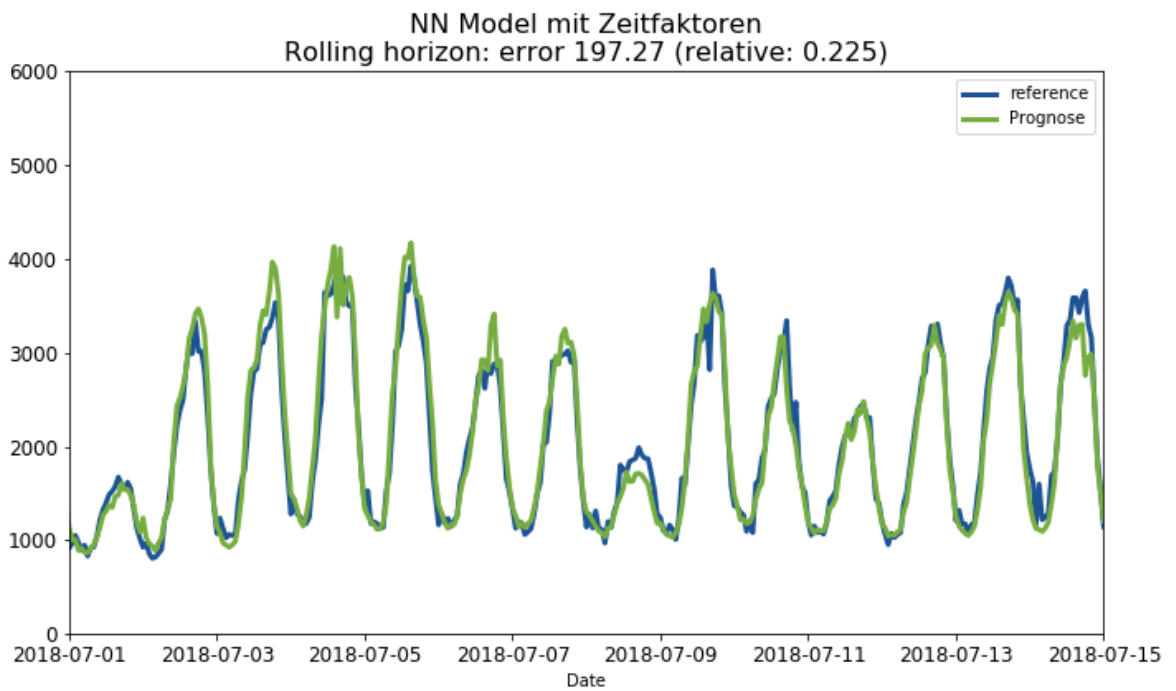


Abbildung 38: Prognose des Netzes mit Zeitfaktoren (Sommer)



## 5. Verbesserung der Modelle und Nachbearbeitung den Prognosen

Vor- und Nachbearbeitungsschritte werden eingeführt, um die Prognosegüte den vorgestellten Modellen zu verbessern.

### 5.1. Zusätzliche Verbesserungen

Zusätzlich werden verschiedene Verbesserungen eingeführt:

- Schätzung des Wachstums
- Berechnung einer unteren und oberen Grenze
- „Weiche“ Ebene
- Anpassung der Prognose an die zwei letzten Wochen
- Glättung der Prognose
- Mittelwert von verschiedenen Modellen

#### 5.1.1. Schätzung des Wachstums

In den letzten Jahren ist das Fernkälte Netz München gewachsen und es sollte in den kommenden Jahren noch stärker erweitert werden. Folglich lohnt es sich, das Wachstum zu schätzen und miteinfließen zu lassen.

Es wird die folgende Behauptung gemacht:

$$\mathbf{Last}(t) = \mathbf{Last}_{statisch}(t) \times \mathbf{Faktor}(t) \quad (38)$$

Als statische Last wird zuerst eine Prognose erstellt, die lediglich Wettergrößen als Eingang aufweist.

Der Wachstumsfaktor wird als eine Kombination von Grad 2 S-Splines beschrieben:

$$\mathbf{Faktor}(t) = \mathbf{a} + \mathbf{b} \times t + \sum_{j \in \text{Jahr}} \mathbf{c}_t \times \mathbf{S}_{2,j}(t) \quad (39)$$

Man kann die Koeffizienten dieses Faktors durch eine Regression bestimmen. So erhält man einen Wachstumsfaktor, wodurch die Ist-Last für das Training normiert wird.

## 5.1.2. Untere und obere Grenze

Wenn viele Ebene für multivariate Regression eingeführt werden und dazu viele zusätzliche Eingangsparameter, werden die Modelle anfälliger für extreme Werten. Um dieses Problem zu lösen, wird nach der Prognose eine untere und obere Grenze ermittelt, um mögliche Ausreißer zu entfernen.

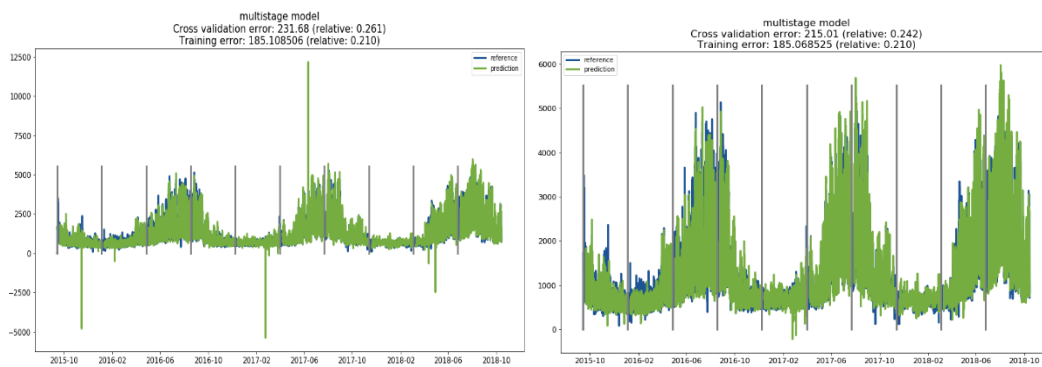


Abbildung 39: Original Prognose mit Ausreißer (links) und Verbesserung (rechts)

Für die untere Grenze wird die Minimumlast für den Tag berechnet:

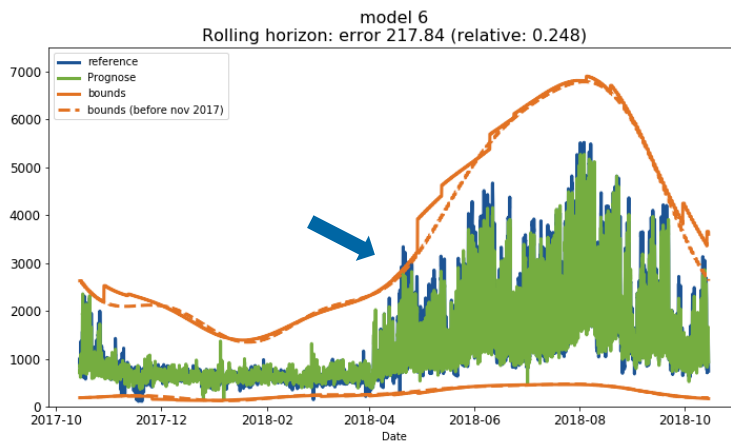
$$\mathit{Minimumlast} = \min_{j \in \text{Jahre}} \min_{\text{Tag} - \Delta T \leq t \leq \text{Tag} + \Delta T} \mathit{Last}(j, t) \quad (40)$$

Dann wird eine Interpolation durch trigonometrische Funktionen durchgeführt, um die Kurve zu glätten:

$$\mathit{untere\ Grenze}(t) = c_0 + \sum_{i=1}^n c_i \cos\left(\frac{2\pi t \times i}{\text{Jahr}}\right) + s_i \sin\left(\frac{2\pi t \times i}{\text{Jahr}}\right) \quad (41)$$

Die obere Grenze wird ähnlich berechnet.

Leider kann die Glättung die Prognose unter Umständen verschlechtern. In Abbildung 40 wird beispielsweise auf das Linear Model 5 getestet, mit  $\Delta T = 256 \text{ Stunden}$  ( $\sim 10,5 \text{ Tage}$ ).



**Abbildung 40: Die ist-Last schneidet die obere Grenze**

Wegen des Wachstums des Netzes kann es vorkommen, dass die Methode zu hohe Prognosewerten, die richtig waren, abschneidet. Dieses Problem könnte man lösen, wenn das Wachstum des Netzes im Voraus berücksichtigt wird oder mit anderen Methoden versucht wird, Ausreißer zu entfernen (etwa den Vergleich mit robusterer Prognose).

### 5.1.3. „Weiche“ Ebene

Mit dem Linear Model 3 (mit 6 Ebenen: Jahreszeiten x Feiertage) wurde bemerkt, dass die Prognose an den Übergangsbereichen der Jahreszeiten nicht gut war. Daher wird versucht, diese harte Ebene aufzulockern. Zwei unterschiedliche Ansätze wurden ausprobiert:

- An den Jahreszeitgrenzen wird die Prognose als gewichteter Mittelwert von beide Modellen berechnet (z.B. Sommer und Übergangszeit).
- Statt einer Ebene werden mehrere Ebenen mit Koeffizienten jedem Zeitpunkt zugewiesen. Dann wird ein gewichtetes anstatt eines normalen Regressionsverfahren angewandt, um die Modelle zu trainieren. Zum Schluss wird für jeden Zeitpunkt eine gewichtete Kombination der entsprechenden Prognose berechnet.

Im zweiten Ansatz werden diese „weiche“ Ebene auch im Training beachtet.

Für die Jahreszeitebene haben die beiden Ansätze eine geringe Verbesserung erbracht. Für stündliche Ebene erbringt der zweite Ansatz eine Verbesserung. Der Fehler wird von 22,9% auf 22,3% reduziert. Man merkt diese Verbesserung hauptsächlich im Winter, da hier die Prognose glatter wird (siehe Abbildung 41).

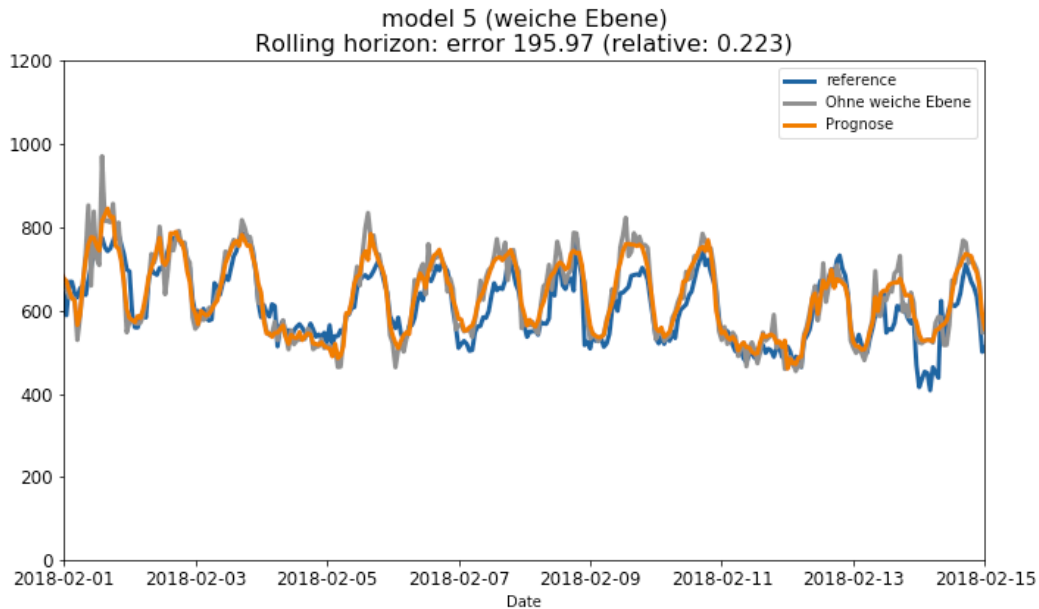


Abbildung 41: Prognose mit weichen Ebenen (Winter)

#### 5.1.4. Anpassung der Prognose an die zwei letzten Wochen

Beim Fernkältenetz München gibt es zurzeit noch nicht so viele Kunden, sodass ein neu verbundener Kunde kann ein spürbares Wachstum verursachen. Deshalb wird das Model an kurzfristiges Wachstum angepasst.

Zuerst wird eine Prognose auf Basis der zwei vergangenen Wochen erstellt.

Dann wird diese mit Hilfe einer linearen Regression angepasst:

$$y = a + b\bar{y} \quad (42)$$

Diese Koeffizienten werden benutzt, um die Prognose für die nächste Tagen anzupassen.

### 5.1.5. Glättung der Prognose

Die meisten Prognosen waren sehr spitzig. Wenn diese geglättet werden, konnte auch eine spürbare Verbesserung ausgemacht werden.

In diesem Beispiel wird die Prognose abhängig von einem Koeffizienten  $\lambda$  geglättet:

$$\bar{y}_t^{new} = \lambda * (\bar{y}_{t-1} + \bar{y}_{t+1}) + (1 - 2\lambda) * \bar{y}_t \quad (43)$$

Bemerkenswert ist, dass die Glättung der Prognose auch bei Modellen mit „weicher Ebene“ eine Verbesserung bringt. Das optimale Glättungsparameter liegt etwa bei 0,25.

In den folgenden Abbildungen wird der relative Fehler abhängig von dem Glättungsparameter dargestellt:

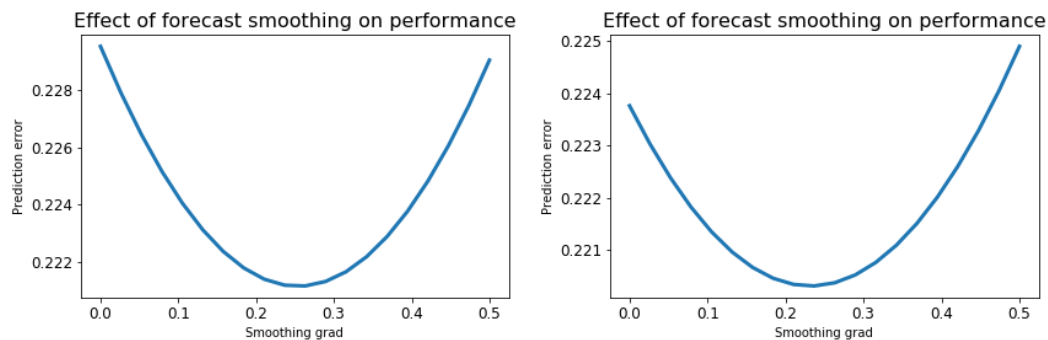


Abbildung 42: Glättung der Prognose für das Linear Model 5 (rechts mit weichen Ebenen)

## 5.1.6. Mittelwert von verschiedenen Modellen

Wenn verschiedene Prognosen mit ähnlicher Genauigkeit zusammengefasst werden, kann auch dadurch eine Verbesserung eintreten. Wichtig dafür ist, dass die Prognosen komplementär sind. Deswegen werden Prognosen bevorzugt, die durch unterschiedliche Verfahren erstellt werden.

In den folgenden Abbildungen wird die Güte der Prognosesumme abhängig von der Gewichtung abgebildet.

Mit 41% des Netzes, 33% des linearen Modells und 27% des K-nächste Nachbarn Model erreicht man sogar eine Prognosegüte von 20,7% (Zum Vergleich: 22,9% mit nur einem Model).

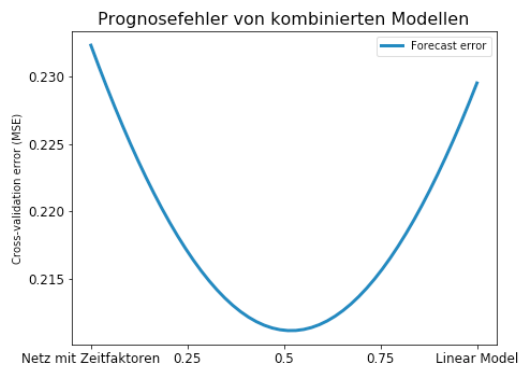


Abbildung 43: Mischung von 2 Prognose (Netz mit Zeitfaktoren und linear Model 5)

$$y = \text{Fehler}((1 - x) \times \overline{\text{Prognose}}_{\text{netz}} + x \times \overline{\text{Prognose}}_{\text{linear}})$$

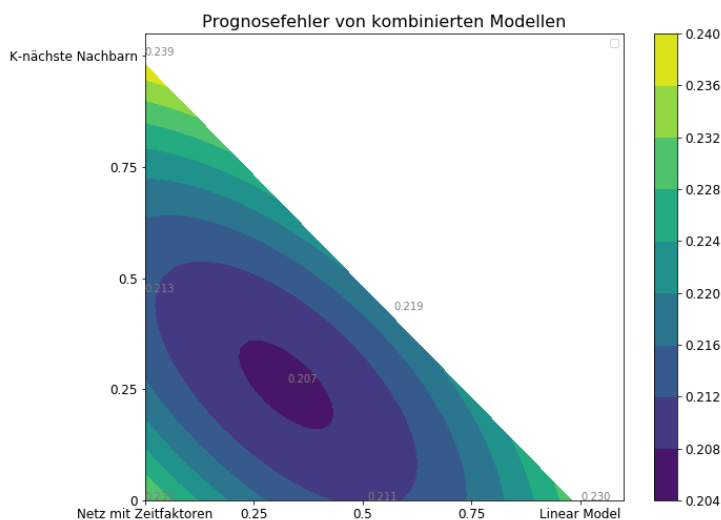


Abbildung 44: Mischung von 3 Prognose (Netz mit Zeitfaktoren, linear Model 5 und K-nächste Nachbarn 2)

$$z = \text{Fehler}((1 - y) \times [(1 - x) \times \overline{\text{Prognose}}_{\text{netz}} + x \times \overline{\text{Prognose}}_{\text{linear}}] + y \times \overline{\text{Prognose}}_{\text{K-nächste Nachbarn}})$$

### 5.1.7. Ergebnisse

Diese verschiedene Verbesserung werden auf dem Linear Model 5 getestet.

Die Glättung der Prognose ist also diejenige, die zu der größten Verbesserung führte.

		Normal	2 Wochen	Glättung	Wachstum	weiche	alle 4
			Anpassung			Ebene	
Fehler	Absoluter (kW)	201,03	199,23	193,5	195,47	195,32	187,82
	Relativer	22,9%	22,7%	22,0%	22,6%	22,3%	21,40%

## 5.2. Summe von Einzelprognosen

Das Fernkältenetz Münchens ist noch klein genug, um jeden neuen Kunden einzeln prognostizieren zu können. Daher wird versucht, die Kunden einzeln zu prognostizieren und die Prognose danach zu addieren.

Was sollte man daraus erwarten? Es ist bekannt, dass die Summe von Multivariater Regression Prognose gleich der Prognose der Summe ist:

Mit  $\bar{y}_i = \text{Prognose}(y_i)$  und  $\bar{y} = \text{Prognose}(\sum_i y_i)$  erhält man für multivariate Regression die Identität:

$$\bar{y} = \sum_i \bar{y}_i \quad (44)$$

Es können verschiedene Ansätze angewandt werden, um die zusätzlichen Informationen zu nutzen:

- Bestimmung des Anfangsdatums für das Training (neu angebundene Kunden)
- Selektion der besten Modelle für jeden einzelnen Kunden
- Wenn ein Kunde kurzfristig angebounden wurde, Anpassung der Gesamtprognose an die Last des Kunden in den zwei letzten Wochen
- Unterschiedliche Parameterauswahl für die jeweiligen Kunden

Diese drei Punkte werden zusammen mit zwei Basis Modellen eingeführt: Multivariates lineares Netz und neuronales Netz mit Zeitfaktoren.



### Summe der Einzelprognosen:

- Berechnung einer Prognose für die gesamte Last
- Für jeden Kunden:
  - Wenn der Kunde nicht genug Zeit am Netz geschlossen war:

Skalierung der Prognose der gesamten Last basierend auf den zwei letzten Wochen, um den Kunden zu prognostizieren.

$$\mathit{Prognose}_{\text{Kunde}} = a + b * \mathit{Prognose}_{\text{gesamt}} \quad (45)$$

- Anderweitig:

Auswahl der besten Prognose zwischen Prognose des einzelnen Kunden und der skalierten Gesamtprognose.

Das führte zu einer deutlichen Verbesserung. Allerdings gibt es zwei praktische Probleme. Zuerst ist die Summe des Kundenverbrauchs leicht unterschiedlich von der Erzeugung und zweitens werden in der Zukunft andere Messgeräte bei den neuen Kunden installiert, womit deren Daten nicht sofort an die SWM übermittelt werden.

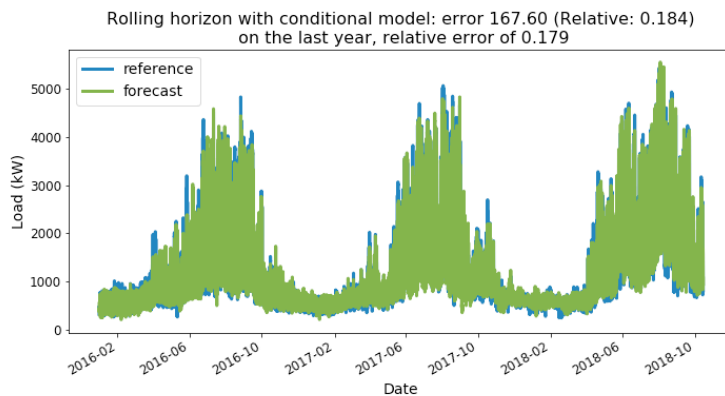


Abbildung 45: Summe von Einzelprognose verglichen mit dem gesamten Verbrauch

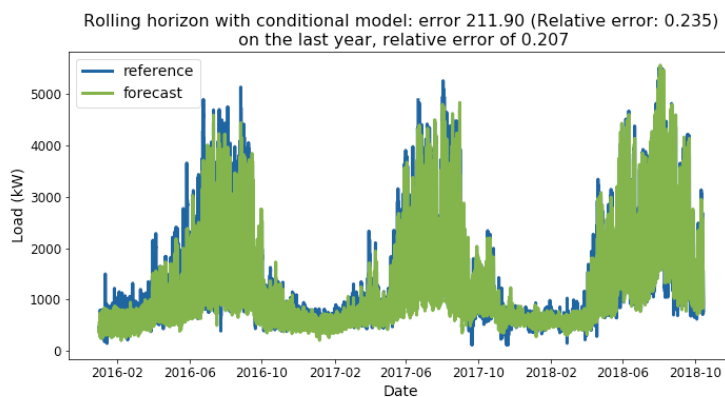


Abbildung 46: Summe von Einzelprognose verglichen mit der Erzeugungslast

Es wird einen Fehler von 20,7% im Vergleich zu der Erzeugungslast erreicht, und sogar 17,9% verglichen mit dem gesamten Verbrauch.

Dieser Unterschied zwischen Erzeugung und Verbrauch lässt sich durch zwei Punkte erklären. Der Hauptfaktor ist die Zeitversatz zwischen Erzeugung und Verbrauch. Zusätzlich fehlen im Jahr 2016 die Lastdaten für drei Kunden teilweise, daher ist die Abweichung zwischen Erzeugung und Verbrauch besonders groß in diesem Zeitraum.

## 5.3. Parameterauswahl und Sensitivität der einzelnen Parameter

Die Frage, ob ein Parameter überflüssig ist, ist nicht einfach zu beantworten. In dem aktuellen Fall hängen Enthalpie und Temperatur stark zusammen, sodass es möglich ist, die Prognose auf Basis nur eines Parameters zu erstellen. Wenn die Temperatur als Eingangsparameter festgelegt wird, ist die Enthalpie fast überflüssig. Wenn die Temperatur gar nicht im Model vorkommt, wird die Enthalpie einer der Hauptparameter.

Anhand dieses Beispiels lässt sich erklären, warum die Parametersensitivität, also die Bedeutung der Parameter, häufig verzerrt ist. Eigentlich muss das gesamte Parameterset betrachtet werden, um eine klare Aussage treffen zu können. Trotzdem ist es interessant, ob ein Parameter unnötig ist oder nicht, wofür verschiedene Sensitivitätsansätze implementiert werden.

### 5.3.1. Parameterauswahl Verfahren

Hier wird das folgende Problem angegangen:

Es stehen  $n$  Parameter zu Verfügung (Wetterdaten, Splines von Wetterdaten, vergangene Last, kalendarische Daten), von denen die beste Auswahl  $\{par_1, par_2, \dots, par_k\}$  gefunden werden soll.

Man könnte glauben, dass sich eine große Anzahl an Parametern als vorteilhaft erweist. Das trifft jedoch nicht zu. Je mehr Parameter es gibt, desto wahrscheinlicher ist es, ein Overfitting zu generieren: Das Model wird derart stark an die Trainingsdaten angepasst, dass der Verallgemeinerungsfehler zunimmt. Das Modell hat die Trainingsdaten sozusagen auswendig gelernt, die wichtigen Zusammenhänge jedoch nicht erkannt. Zusätzlich wird das Modell anfälliger für extreme Werte.

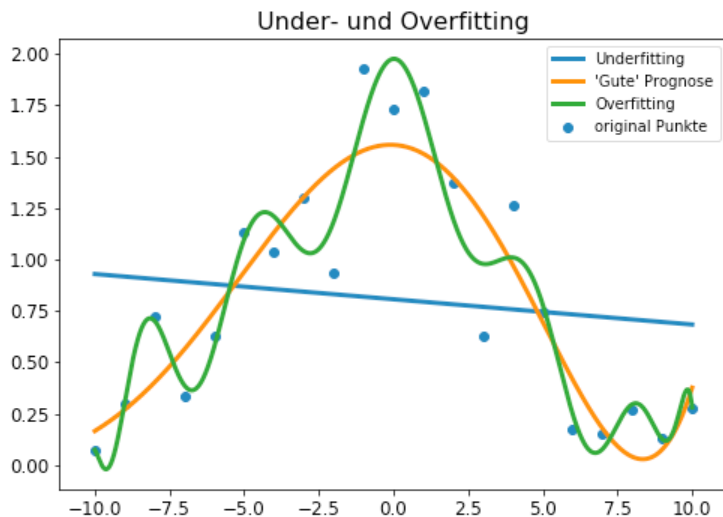


Abbildung 47: Beispiel von Under- und Overfitting

Daher gilt es, eine beschränkte Auswahl an Parametern ausfindig zu machen.

Insgesamt gibt es  $2^n$  Kombinationen (mit  $\sim 50$  Parameter sind es fast 1 125 900 Milliarden Kombinationen), so dass es höchst unpraktikabel wäre, sie alle zu testen. Deshalb gibt es auch in der Literatur einige Methoden, um die beste Auswahl zu treffen, siehe z.B. [5] und [10].

Drei Ansätze werden implementiert und verglichen:

- Schrittweise Entfernung von Eingangsparametern
- Genetischer Algorithmus mit „Simulated Annealing“ (Simulierte Ausglühen)
- Lokale Suche mit Bayesscher Inferenz

Diese Ansätze werden nun erläutert.

## Schrittweise Entfernung von Eingangsparametern

Begonnen wird mit allen Parametern.

An jedem Schritt wird bestimmt, welcher einzelne Parameter die kleinste Verbesserung bringt. Dieser Parameter wird dann entfernt.

### Implementierung

- Berechnung der aktuellen Parameterauswahl  $S = \{x_1, x_2, \dots, x_k\}$
- An jedem Schritt:
  - Für jeden Parameter  $x_i$ :
    - Berechnung des Prognosefehlers ohne  $x_i$
  - $x_i^* = \arg \min_i(\text{Fehler}_i)$
  - $S_{\text{Schritt}} = S_{\text{Schritt-1}} \setminus \{x_i^*\}$
- Ergebnis der besten Parameterauswahl  $S_{\text{Schritt}}^*$

## Genetischer Algorithmus mit „Simulated Annealing“ (Simulierte Ausglühen)

Dieser Algorithmus wird in [10] beschrieben:

Man beginnt mit einer Liste (Bevölkerung) von Sets von Parametern (Individuen). In jedem Schritt werden zwei Dinge geschehen: die „Individuen“ werden „Kinder“ haben, also werden aus zwei Sets ein neues erstellt. Diese „Kinder“ werden dann zufällig mutieren, es werden Inputs addiert, getauscht oder entfernt.

Bisher handelt es sich noch um einen konventionellen Genetischen Algorithmus. Anstatt nur die gesunden Individuen (mit geringerem Fehler) zu behalten, werden aber die „überlebenden“ Individuen zufällig gewählt: ein nicht gesundes Individuum hat zwar weniger Chancen gewählt zu werden, aber trotzdem wird es nicht kategorisch ausgeschlossen. Zu Beginn haben alle Individuen fast die gleichen Chancen ausgewählt zu sein, während sie am Ende fast nur ausgewählt werden, wenn sie gesund sind. Auf diese Weise kann zuerst einer groben Suche aufgeführt werden und diese allmählich verfeinern.

Der Algorithmus wurde vom oben erwähnten Ausglühen Prozess inspiriert: mit sinkender Temperatur finden die Molekülen Kombinationen mit niedriger Energie.

Die Überlebenschance eines Individuums wird wie folgt berechnet:

$$p_i = \frac{\exp\left(-\frac{\text{Fehler}_i}{\text{Temperatur}}\right)}{\sum_{j \in \text{Bevölkerung}} \exp\left(-\frac{\text{Fehler}_j}{\text{Temperatur}}\right)} \quad (46)$$

Bei jedem Berechnungsschritt wird dieselbe Anzahl von Individuen anhand deren Überlebenschancen ausgewählt. Außerdem wird der Algorithmus insofern modifiziert, dass immer die zwei besten Sets beibehalten werden.

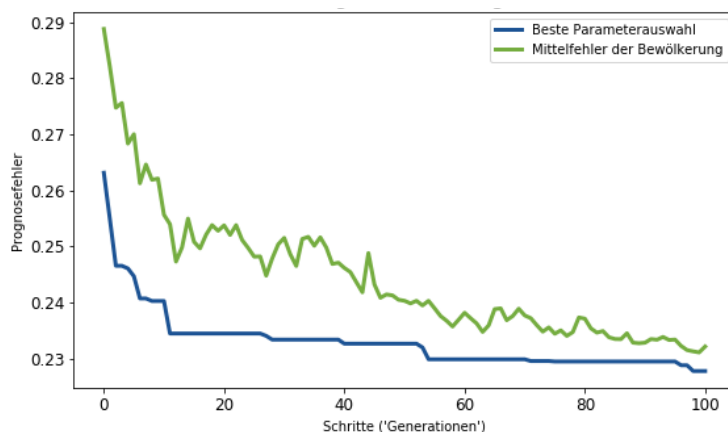


Abbildung 48: Verlauf des genetischen Algorithmus

## Lokale Suche mit Bayesscher Inferenz

Man startet hier mit nur einem Set und gibt allen Parametern das Rating 0.

An jedem Schritt werden  $m = 5$  Parameter addiert oder entfernt: Für diese  $m$  Möglichkeiten wird dann der Fehler berechnet. Liefert die neu berechnete Auswahl bessere Ergebnisse wie die vorherige, wird das neue Set für die nächsten Schritte behalten.

Das Rating dieser  $m$  Parameter wird, abhängig von deren Erfolg, aktualisiert:

Versuch	Verbesserung	Verschlechterung
Hinzufügen	+1	-1
Entfernen	-1	+1

Tabelle 8: Aktualisierung der Bayesscher Inferenz

Mit diesen Ratings lassen sich dann die „Chancen“ berechnen, sich im optimalen Set zu befinden. Durch den Parameter  $\lambda$  wird die Geschwindigkeit des Algorithmus gesteuert:

$$p_i = \frac{1}{1 + \exp(-\lambda \times \text{rating}_i)} \quad (47)$$

Mit diesen Wahrscheinlichkeiten werden im nächsten Schritt  $m$  Kandidaten ausgewählt ( $\text{Chancen} \sim p_i$  für einen Hinzufügen und  $\text{Chancen} \sim 1 - p_i$  für eine Entfernung)

Diese Art von Bayesscher Schlussfolgerung könnte besser mit folgendem Modell begründet werden:

Der Input ist Bernoulli verteilt, mit Parameter  $\theta$ .

Im Voraus glaubt man, dass der Parameter  $\theta$  eine gewisse Verteilung zwischen 0 und 1 haben:  $\theta \sim \mathcal{L}$

Anschließend wird die Wahrscheinlichkeit berechnet, dass  $\theta$  größer  $\frac{1}{2}$  sei, wenn man die Beobachtungen  $X_{i,1}, X_{i,2}, \dots, X_{i,3}$  kennt:

$$p_i = \mathbb{P}\left(\theta > \frac{1}{2} \mid X_{i,1}, X_{i,2}, \dots, X_{i,3}\right) \quad (48)$$

Im Rahmen dieser Arbeit wurde keine leicht berechenbare analytische Formel für  $p_i$  gefunden, für den Fall, dass  $\mathcal{L}$  die uniforme Verteilung darstellt.

In [29] wird auf das gleiche Problem mit einer Beta Verteilung anstatt einer uniformen Verteilung eingegangen.

## **Ergebnisse**

Für diese Algorithmen wird der relative Fehler ebenfalls mit der Methode Gleitender Horizont erfasst. Jedoch werden Perioden von zehn Wochen anstatt zwei Wochen verwendet. Somit wird die Bewertung schneller durchgeführt, der Fehler hingegen etwas ungenauer. Beispielsweise erhält man mit dem Linear Model 5 einen relativen Fehler von 22,59% bei zwei Wochen und 22,29% bei zehn Wochen.

Es werden jedes Mal mehr als 50 Parameter getestet:

- Wetterdaten
- Splines von Wetterdaten
- Historische Last
- Kalendarische Daten
- Lastprofil

Zusätzlich werden vier Fälle unterschieden:

Mit und ohne Luftfeuchte und mit und ohne historische Last.

Hintergrund dafür ist, dass keine belastbaren kurzfristige Prognosen für die Luftfeuchtigkeit vorliegen, sodass die Luftfeuchte nur für langfristige Prognosen benutzt werden könnte. Im Gegensatz dazu kann die historische Last nur für die kurzfristige Prognose benutzt werden. Im praktischen Anwendungsfall einer mittelfristigen Prognose (über 168 Stunden) sollte diese ohne Luftfeuchte und ohne historische Last erstellt werden.



Diese Verfahren werden auf „Linear Model 6“ angewandt. Es handelt sich um eine Kombination von „Linear Model 5“ und Wachstum Entfernung (siehe 5.1.1).

Die folgenden Ergebnisse werden erzeugt:

Model	Linear Model 6	Linear Model 6
mit historischer Last	✓	✗
mit Luftfeuchte	✗	✓
Methode	genetischer Algorithmus + Lokal	Lokal
Fehler (relativ)	22,56%	25,34%
Zeit	00:48:15	01:06:14
	Feiertag Global Strahlung Temperatur Temperatur_Sspline2_10_20 Temperatur_index-48_Sspline2_10_20 reference_copy_index-168 reference_copy_index-48 reference_copy_mean-48	Enthalpie Enthalpy_Sspline1_-10_20 Feiertag Luftfeuchte_Sspline1_0_33 Temperatur Temperatur_index-168 Temperatur_index-1_Sspline1_10_20 Temperatur_index-48_Sspline2_10_20 Temperatur_mean-96 Temperatur_mean0_Sspline2_10_20 yearCos3

Tabelle 9: Optimale Parameterauswahl

Mit folgender Bezeichnung:

Reference_copy	Kältelast
XXX_SsplineN1_N2_N3	S-Spline von Grad 2 mit Stützpunkten N2 und N3 bezogen auf Parameter XXX
XXX_index-N	Zeitreihe XXX mit N Stunden Versatz
XXX_mean-N	Tagesmittelwerte der Zeitreihe XXX mit N Stunden Versatz (zwischen N - 12h und N + 12h)
yearCosN	$\cos\left(\frac{Zeit}{Jahr} \times 2\pi \times N\right)$

Tabelle 10: Parameterbezeichnung

(Eigentlich gibt es hier Mittelwerte zwischen -50h und -36h, so dass das Model lediglich für einen Zeithorizont von 36h geeignet ist. Das wurde in der BoFiT Implementierung korrigiert)

Die **Parameterentfernung Algorithmus** hat insgesamt ein wenig schlechtere Ergebnisse geliefert mit längerer Laufzeit. Diese betrug etwa acht Stunden auf einem normalen Rechner für das Linear Model 6.

Es müssen für die 50 Parameter  $\frac{n(n-1)}{2}$  Kombinationen, in Summe 1275, getestet werden. Jedoch ist diese Variante einfach zu implementieren und nachvollziehbar. Es kann auch in einen zweiten Schritt dazu verwendet werden, um die Anzahl von Eingangsparametern zu reduzieren.

Für den Fall „Linear Model 6, mit historischer Last und ohne Luftfeuchte“, wird ein Fehler von 22,95% erreicht, während die lokale Suche auf 22,61% beschränkt werden konnte.

Der **genetische Algorithmus** hat weniger Zeit gebraucht. So lag die Berechnungszeit zwischen 4,75 und 7,00 Stunden für den Linear Model 6. Die Ergebnisse waren ähnlich derer der **Lokalen Suche**. Hier werden 1010 Kombinationen getestet (10+10\*100 Schritte).

Für die lokale Suche konnten hiermit gute Startpunkte gefunden werden.

Die **lokale Suche** ist mit Abstand der schnellste Algorithmus, er benötigt lediglich zwischen einer und zwei Stunden Laufzeit für das Linear Model 6. Zusätzlich werden hiermit bessere Ergebnisse geliefert. Außerdem werden so nur 500 (5\*100 Schritte) Kombinationen überprüft.

Für die weiteren Berechnungen wird deshalb die lokale Suche empfohlen.

Diese Algorithmen werden auch mit neuronalen Netzen getestet. Als problematisch zeigt sich hierbei, dass es Zufälligkeiten im Training gibt, sodass die beste Parameterauswahl nicht eindeutig zu erkennen ist. Die beiden letzten Algorithmen konnten damit gut umgehen, wobei sich die Laufzeit deutlich auf 29 und 4 Stunden erhöht.

Außerdem konnten die Neuronalen Netze besser mit vielen Parametern und Over Fitting umgehen. Im Training wird der Verallgemeinerungsfehler berechnet und als Abbruchkriterium benutzt. Daher werden mehr der 22 möglichen Parameter akzeptiert, obwohl nicht alle nötig wären.

Langfristige Prognose werden auch für jeder Kundengruppe berechnet. Daher macht es Sinn, die besten Parameterauswahl für jede Kundengruppe zu bestimmen.

Die Gruppen weisen große Unterschiede auf. Sie benutzten zwischen 9 und 23 Eingangparameter und der relative Fehler lag zwischen 16,7% bis zu 58,1%.

Weitere Ergebnisse werden im Anhang II vorgestellt.

### 5.3.2. Sensitivitätsanalyse

Ziel ist es, die Bedeutung eines einzelnen Eingangsparameter zu messen. Im Gegenteil zu das Parametersauswahl werden individuellen Eingänge bewertet. Für diesen Zweck werden verschiedene Ansätze erprobt.

- Korrelation
- Standardisierte Regression Koeffizienten
- Reihenfolge der Parameter Entfernung (Auswahlalgorithmus eins)
- Bayessche Inferenz (Wahrscheinlichkeit, dass der Parameter im optimalen Set enthalten ist, ergeben von Auswahlalgorithmus drei)
- Marginale Verbesserung im Vergleich zum optimalen Set

#### Korrelation

Die (Pearson) Korrelation misst die lineare Abhängigkeit der Daten. Die Information ist aber begrenzt, weil eine Variable wie Sonnenscheindauer lediglich mit der Last korreliert, weil es mit der Temperatur korreliert. Die Betrachtung von Sonnenscheindauer führt aber zu keiner Verbesserung der Prognosegüte.

Es ist also hauptsächlich in der erste Datenanalyse interessant (siehe Kapitel 3.3).

#### Standardisierte Regression Koeffizienten

Wenn die Last durch ein Regressionsverfahren prognostiziert wird, messen diese Koeffizienten, inwiefern sich die Varianz der Last durch die Varianz den einzelnen Inputs erklären lässt. Sie werden wie folgt berechnet:

$$\bar{c}_i = c_i \times \frac{\text{std}(X_i)}{\text{std}(y)} = \frac{\text{std}(c_i X_i)}{\text{std}(y)} \quad (49)$$

Mit  $c_i$  als Regressionskoeffizienten und  $\text{std}(y)$ ,  $\text{std}(X_i)$  als Standardabweichung der Last und des Eingangs.

Dieser Ansatz kann verallgemeinert werden, um Gruppen von Inputs, z.B. alle Temperatur Splines, anstelle einzelner Inputs umgehen zu können.

$$\bar{c}_{\{Inputs\}} = \frac{\text{std}(\bar{y}_{\{Inputs\}})}{\text{std}(y)} \quad (50)$$

Hier bezeichnet  $\bar{y}_{set}$  die Prognose, die sich ergibt, wenn alle Inputs außer diejenigen des aktuellen Sets zu Null gesetzt werden.

In den folgenden Abbildungen werden die Ergebnisse abgebildet (siehe Tabelle 10 für die Bezeichnung der Zeitreihen)

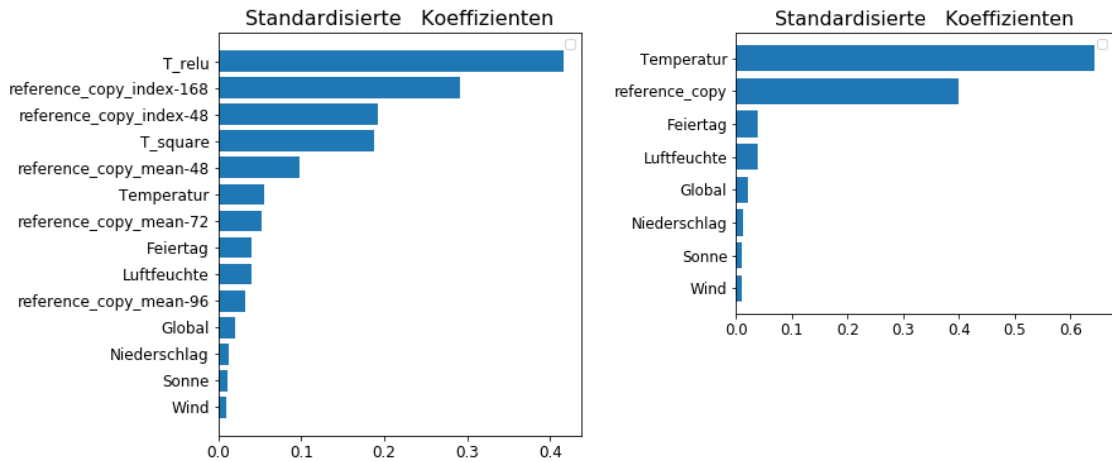


Abbildung 49: Standardisierte Koeffizienten (links: einzelne Inputs, rechts: gruppiert)

### Reihenfolge der Parameterentfernung

Beim schrittweisen Parameterentfernungs-Algorithmus wird der Koeffizient mit dem schlechtesten Beitrag zur Prognose entfernt. So lassen sich die einzelnen Parameter einordnen. Die Sensitivität ergibt sich aus der Reihenfolge der Parameterentfernung.

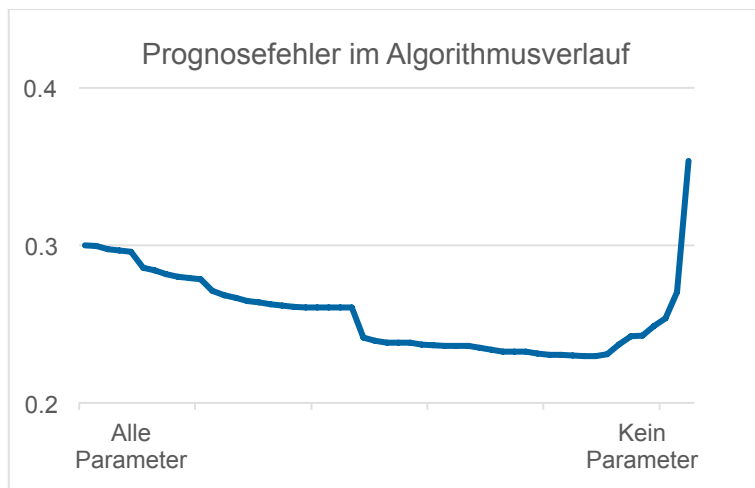


Abbildung 50: Prognosefehler (Schrittweise Parameterentfernung)

## Bayessche Inferenz

Im Laufe des Algorithmus Lokale Suche wird die Wahrscheinlichkeit berechnet, mit der sich der Parameter im optimalen Parameterset befindet. Obwohl diese Wahrscheinlichkeit von der zufälligen Reihenfolge von ausgetestetem Parameter abhängt, liefert es eine interessante Aussage über den individuellen Beitrag der Parameter.

## Marginale Verbesserung im Vergleich zum optimalen Set

Durch diesen bevorzugten Ansatz wird der individuelle Beitrag der Parameter zum optimalen Parameterset berechnet. Jeder Parameter wird entweder zum optimalen Set hinzugefügt oder entfernt, abhängig davon, ob er sich bereits im Set befindet oder nicht. Wenn das optimale Set nicht vorhanden ist, wird stattdessen das Ergebnis eines Parametersauswahl Algorithmus benutzt. In diesem Fall wird auch gleichzeitig geprüft, ob das Set zumindest ein lokales Minimum ist.

## Ergebnisse

Mit einigen Methoden konnten auch Gruppen von Eingangsparametern betrachtet werden. Damit sind die Ergebnisse besser nachvollziehbar.

So wird eine grobe Sensitivitätsanalyse durchgeführt, in der alle Eingangsparameter vom Linear Model 5, die von denselben Haupteingangsparametern (Temperatur, Luftfeuchte, ...) abhängen, zusammengefasst. Für die Korrelation wird nur der Hauptkoeffizient betrachtet:

	Korrelation	Standardisierten Koeffizienten	Reihenfolge der Parameter Entfernung	Marginale Verbesserung
Temperatur	0,79	0,53	8	13,69%
Reference_vor_2T	0,81	0,32	7	9,54%
Feiertag	0,08	0,00	6	0,25%
Luftfeuchte	0,50	0,08	5	0,12%
Global	0,52	0,00	4	0,02%
Wind	0,01	0,01	3	0,01%
Sonne	0,48	0,00	2	0,01%
Niederschlag	0,01	0,00	1	0,02%

Tabelle 11: Grobe Sensitivitätsanalyse (Die drei wichtigsten Parameter werden hervorheben)

Die Korrelation betrachtet nur die Abhängigkeit zwischen zwei Parametern. Die standardisierten Koeffizienten zeigen, wie die verschiedenen Eingänge in einer multivariaten Regression benutzt werden. Die zwei letzten Ansätze messen hingegen die marginale Verbesserung des Modells, wenn der Parameter hinzugefügt wird. Der erste Ansatz bezieht sich auf eine Parameterauswahl, die kleiner wird, während die zweite sich auf einer festgelegten Auswahl bezieht.

Daher werden die zwei letzten Ansätze bevorzugt, um die richtigen Parameter für ein Prognosemodell zu wählen.

Schließlich wird die Sensitivitätsanalyse für jeden einzelnen Eingangsparameter durchgeführt. In nachfolgender Tabelle sind die wichtigsten Parameter nach der marginalen Verbesserung sortiert. Für jeden Ansatz werden die neun besten Parameter hervorgehoben. Dadurch kann leicht erkannt werden, welche Eingänge am besten geeignet sind, um die Prognose zu erstellen. Es wird ebenfalls deutlich, dass die ersten Ansätze oft mit der marginalen Verbesserung übereinstimmen. Allerdings werden einige Parameter übersehen, die zu einer höheren Prognosegüte beitragen können (z.B. Feiertage).

	Korrelation	Standardisierten Koeffizienten	Reihenfolge der Parameter Entfernung	Bayessche Inferenz	Marginale Verbesserung
Temperatur_Sspline2_10_20	0,898	1,869	54	0,711	11,52%
reference_copy_index-48	0,814	0,051	53	0,668	0,57%
Feiertag	0,076	0,042	50	0,690	0,55%
Enthalpy	0,787	0,725	51	0,731	0,54%
reference_copy_mean-48	0,673	0,054	49	0,690	0,33%
Temperatur_index-48_Sspline2_10_20	0,786	0,015	48	0,711	0,29%
Temperatur_Sspline1_-10_10	0,831	5,460	47	0,574	0,26%
reference_copy_mean-168	0,653	0,091	52	0,646	0,18%
Luftfeuchte_Sspline2_33_67	0,467	0,054	46	0,450	0,09%
Global_Sspline3_667_1000	0,279	0,004	37	0,289	-0,01%
Temperatur	0,793	3,715	43	0,550	-0,01%
Temperatur_mean-72	0,624	0,013	33	0,310	-0,03%
reference_copy_index-168	0,806	0,282	8	0,289	-0,04%
Temperatur_mean-48	0,642	0,059	45	0,269	-0,04%
...	...	...	...	...	...
...	...	...	...	...	...

Tabelle 12: Sensitivitätsanalyse – die wichtigsten Parameter

(siehe Tabelle 10 für die Bezeichnung der Zeitreihen)

Lediglich die Parameter mit eine positive Marginale Verbesserung gehen in den Model ein.

## 5.4. Fehleranalyse und Konfidenzintervall

Um ein Konfidenzintervall zu erstellen, muss die Prognosefehler analysiert werden. Für das Linear Model 5 wird zum Beispiel bemerkt, dass die normale Verteilung nicht geeignet ist (Abbildung 51).

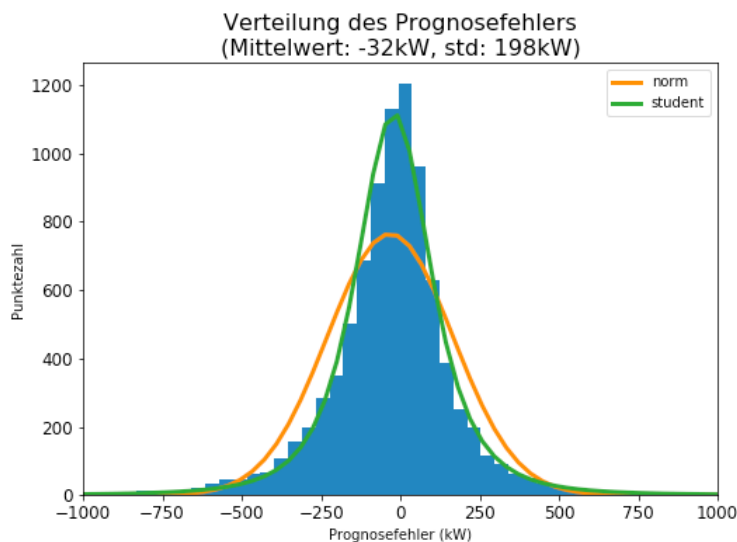


Abbildung 51: Verteilung des Prognosefehlers

Daher wird für das Konfidenzintervall das empirischen Quantils benutzt.

Für eine Konfidenz von  $\alpha \in (0,1)$  werden die folgenden Schritte verfolgt:

- An jedem Zeitpunkt wird der Prognosefehler berechnet
- Es werden Ebene ausgewählt: z.B. Monate und Stunde
- Die empirischen Quantils  $Q_{\frac{\alpha}{2}}$ ,  $Q_{1-\frac{\alpha}{2}}$  des Prognosefehlers wird für jede Ebene ermittelt
- Zu jedem Zeitpunkt addiert man das entsprechende untere und obere Quantil zur Prognose

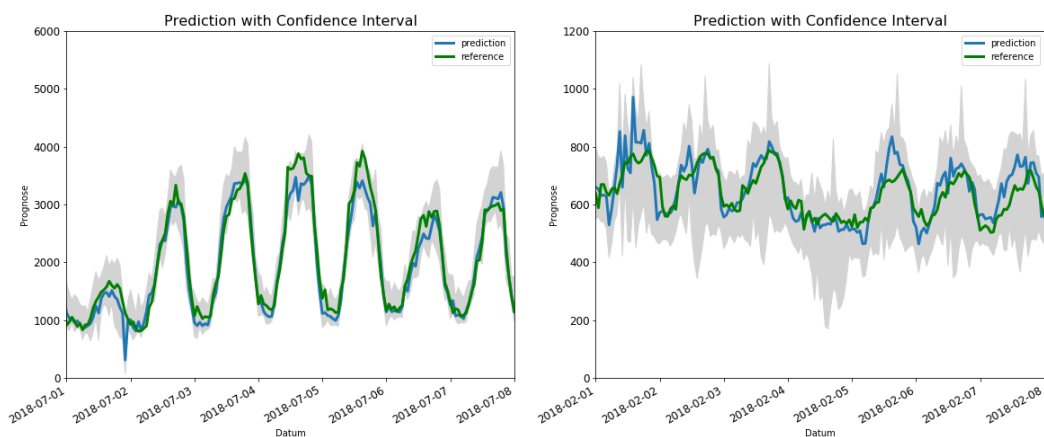


Abbildung 52: Konfidenzintervall (Sommer und Winter)



## 6. Vergleich mit Fernwärmenetz-Prognose

Für die Einsatzplanung werden bereits intern Prognosen für die Fernwärme erstellt. Daher war es interessant, die kurzfristige Kältelastprognose mit den Wärmelastprognose zu vergleichen.

Die Hauptfragestellungen lautet, ob man bei der Fernkälte dieselbe Prognosegüte erwarten kann und falls nicht, warum es zu Unterschieden kommt.

Die Wärmelast korreliert hauptsächlich mit der Temperatur und der Enthalpie.

	Temperatur	Luftfeuchte	Niederschlag	Sonne	Global	Wind	Feiersonntag	Enthalpie	Wärmelast
Temperatur	1,00	0,56	0,01	0,43	0,48	0,07	0,01	0,97	0,89
Luftfeuchte	0,56	1,00	0,21	0,52	0,47	0,16	0,01	0,39	0,38
Niederschlag	0,01	0,21	1,00	0,11	0,07	0,06	0,01	0,03	0,03
Sonne	0,43	0,52	0,11	1,00	0,78	0,04	0,01	0,35	0,25
Global	0,48	0,47	0,07	0,78	1,00	0,08	0,01	0,42	0,31
Wind	0,07	0,16	0,06	0,04	0,08	1,00	0,02	0,03	0,03
Feiersonntag	0,01	0,01	0,01	0,01	0,01	0,02	1,00	0,01	0,04
Enthalpie	0,97	0,39	0,03	0,35	0,42	0,03	0,01	1,00	0,89
Wärmelast	0,89	0,38	0,03	0,25	0,31	0,03	0,04	0,89	1,00

Tabelle 13: Korrelationen zwischen Wetterdaten und Wärmelast

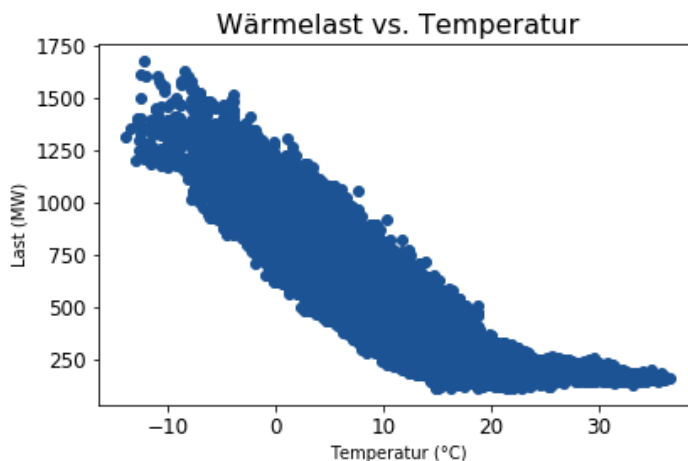


Abbildung 53: Wärmelast - Abhängigkeit von der Temperatur

Bemerkenswert ist, dass die Fernwärmelast etwa 250-mal größer ist als die Fernkältelast. Tatsächlich gibt es deutlich mehr Kunden beim Fernwärmenetz, so dass die individuellen Schwankungen besser ausgeglichen werden als bei der Fernkälte. Man merkt ebenfalls, dass das Netzwachstum geringer ist.

Diese beiden Punkte erleichtern das Prognostizieren bei Fernwärme im Vergleich zu Fernkälte. Um diese Überlegung zu überprüfen, werden drei lineare Modelle auf beide Zeitreihen getestet:

- **Model 1:** Einfaches wetterbasiertes Modell
- **Model 2:** Besseres Regressionsmodell für Fernkälte
- **Model 3:** Besseres Regressionsmodell für Fernwärme

Model	Fernwärme (Relativer Fehler)	Fernkälte (Relativer Fehler)
Einfaches Modell	38,4%	42,7%
Bestes Modell Fernkälte	13,6%	23,5%
Bestes Modell Fernwärme	11,8%	25,9%

Tabelle 14: Prognosegüte für Fernwärme und Fernkälte

Die Prognosegüte ist wesentlich besser bei der Fernwärme.

Dies kann auch mathematisch mithilfe der sogenannten „Gesetz der großen Zahlen“ formuliert werden:

Betrachtet wird ein Modell von  $n$  Kunden, mit identischem Verhalten. Deren Last wird in drei Komponenten aufgeteilt:

$$\forall k \in \mathbf{Kunden} = [1, n], \quad y_k = f_k(x) + g + w_k \quad (51)$$

- $f_k(x)$  prognostizierte Last
- $g$  Prognosefehler, der für alle Kunden gleich ist. (mögliche Prognoseverbesserung)
- $w_k$  Weißes Rauschen: individuelle Lastschwankungen, unabhängig und identisch verteilt (unabhängig von  $g$ ).

Es wird angenommen, dass die Mittelwerte von  $g$  und  $w_k$  bei null liegen.

Man kann den Prognosefehler (RMSE) für  $n$  Kunden berechnen:

$$\begin{aligned}
 \mathbf{MSE} &= \mathbb{E} \left[ \left( \sum_{k=1}^n y_k - f_k(x) \right)^2 \right] & (52) \\
 &= \mathbb{E} \left[ \left( \sum_{k=1}^n g + w_k \right)^2 \right] \\
 &= \mathbb{E} \left[ \left( \sum_{k=1}^n g \right)^2 + \sum_{k=1}^n w_k^2 + 2 \left( \sum_{k=1}^n g w_k + \sum_{j \neq k} w_k \times w_j \right) \right] \\
 &= n^2 \mathbb{E}[g^2] + n \mathbb{E}[w_k^2] \\
 &= n^2 \mathit{var}(g) + n \mathit{var}(w)
 \end{aligned}$$

$$\mathbf{RMSE} = n \times \mathit{std}(g) + \sqrt{n} \times \mathit{std}(w) \quad (53)$$

Sollte man diese Fehler durch die Mittellast teilen, erhält man einen absteigenden relativen Fehler. Um es leichter zu haben, wird der Fehler durch die Mittellast anstatt der Standardabweichung der Last geteilt:

Man nimmt an, dass alle Kunden sich identisch verhalten:  $f_k = f$

$$\mathbf{Mittellast} = n \mathbb{E}(f(x)) \quad (54)$$

$$\mathbf{normierte\ RMSE} = \frac{\mathit{std}(g)}{\mathbb{E}(f)} + \frac{1}{\sqrt{n}} \times \frac{\mathit{std}(w)}{\mathbb{E}(f)} \quad (55)$$

Der normierte Prognosefehler nimmt mit der Zahl der Kunden ab.

Es ist zu erwarten, dass die Prognosegüte bei der Fernwärme niedriger als bei der Fernwärme ist.

## 7. Langfristige Szenarien

Um Investitionen und Netzerweiterungen zu planen, braucht die SWM langfristige Prognosen, am besten für ganze Jahre in der Zukunft. Es ist auch interessant, „extreme“ Szenarien zu erstellen und das Wachstum der Nachfrage vorhersagen zu können.

Das Wachstum erfolgt maßgeblich durch zwei Effekte: zusätzliche Kunden werden jedes Jahr an dem Netz angeschlossen. Wegen des globalen Temperaturanstieges ist ein erhöhter Konsum pro Kunde zu erwarten.

Hilfreich können hier die Kundengruppen sein (siehe Kapitel 3.2). Damit kann das Verbrauchsverhalten von künftigen Kunden feiner prognostiziert werden.

Um Prognosen zu erstellen, kann entweder die Lastverteilung direkt geschätzt werden oder es können künstliche Wetterzeitreihen benutzt werden, um ein Prognosemodell im zweiten Schritt laufen zu lassen. Der zweite Ansatz wird bevorzugt, da bereits präzise Prognosemodelle zu Verfügung stehen und einige Referenzzeitreihen für Wetterdaten bereits vorhanden sind, welche vom Wetterdienst erstellt worden sind.

Als Referenzzeitreihen hat der Wetterdienst sechs Zeitreihen im Stundenraster erstellt. Für 2015 und 2045 gibt es jeweils eine normale Zeitreihe, eine mit heißerem Sommer und eine mit kälterem Winter. Bei der Betrachtung der Zeitreihe für die Stadt München im Jahr 2045 wird aber festgestellt, dass die Zeitreihe mit angeblich heißem Sommer kälter als die normale Zeitreihe ist (siehe Abbildung 54). Deswegen werden auch künstliche Zeitreihen simuliert.

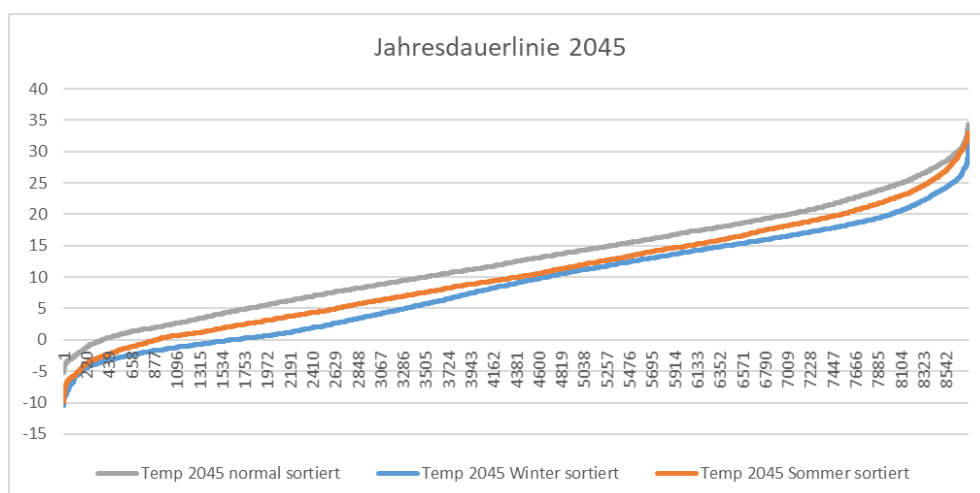


Abbildung 54: Testreferenzzeitreihe 2045 - Jahresdauerlinie der Temperatur

## 7.1. Netzzuwachs

Das Netzzuwachs stellt zwei Herausforderungen.

- Die Prognosemodellen sollten auf vergleichbare Daten getestet werden, so dass das Netzzuwachs im Vorfeld entfernt werden soll.
- In den langfristigen Prognosen sollten neue Kunden betrachtet werden.

Statt das Wachstum zu berechnen wird für langfristige Prognose der Last im Vorfeld durch die mittlere Last den gebundenen Kunden normiert. (Dafür werden die Lastdaten den individuellen Kunden benötigt, so dass es für kurzfristigen Prognosen nicht möglich war.)

### 7.1.1. Normierung der historischen Last

Für jeden Kundengruppe  $g$  wird die normierte Last („ $\overline{Last}$ “) wie folgt berechnet:

$$\forall g \in \text{Kundencluster}, \quad \overline{Last}(t) = \sum_{k \in g} \frac{Last_k(t)}{Faktor(g, t)} \quad (56)$$

$$Faktor(g, t) = \sum_{k \in g} \text{mean}(Last_k) \times \mathbf{1}_{\{k \text{ ist am Netz gebunden}\}}(t) \quad (57)$$

Mit  $\text{mean}(Last_k)$  die mittlere Last der Kunde  $k$  über die gesamte Trainingsperiode.

Das ist also die Gesamtlast der Gruppe  $g$  dividiert durch die mittlere Last der Kunden, die an diesem Zeitpunkt am Netz geschlossen sind, wie in der folgenden Figur abgebildet.

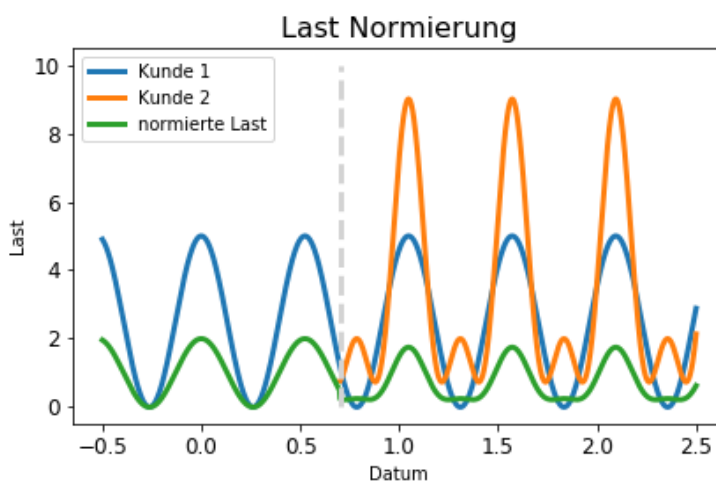


Abbildung 55: Last Normierung

Durch diese Normierung können für jeden Kundengruppe, Modellen auf einem „statische“ Last trainiert werden.

Dazu müssen die Modell Ergebnisse lediglich durch die mittlere Last den aktuellen Kunden multipliziert werden, um Lastprognose für den Kundengruppe zu erstellen.

### 7.1.2. Künftige Kunden

In dem Clustering den Kunden (Kapitel 3.2) werden verschiedene Kundenverhalten erkannt. Dazu ergeben sich drei Ausreißer (Gruppe 4 bis 6).

Um das zusätzlicher Verbrauch den neuen Kunden zu prognostizieren, empfiehlt sich also, ein Prognosemodel für jeder Gruppe zu erstellen.

Wenn das Verhalten von künftigen Kunden schon bekannt ist, können dann deren Last besser prognostiziert werden.

In dem Fall, wenn die neuen Kunden unbekannt sind, kann eine passende Summe den Prognosen für Gruppe 1 bis 3 berechnet werden (Ausreißer Entfernung).

## 7.2. Künstliche Zeitreihen

Daher werden parallel künstliche Zeitreihen für die Temperatur und die Luftfeuchte erstellt, mit der Unterstützung meiner Betreuerin, Dipl. Ing. Inga Müller.

### 7.2.1. Allgemeine Verfahren

Die Zeitreihen werden wie folgt erstellt:

- Anhand von historischen Daten zwischen 1998 und 2018 wird die Verteilung der Luftfeuchte und Temperatur für jede Stunde und jeden Tag des Jahres geschätzt.
- Die Wetterzeitreihen werden mit Hilfe dieser Verteilung in normalverteilte Zeitreihen umgerechnet
- Auf diese normierten Daten wird ein AR (Autoregressive) Model angepasst.
- Mit dem AR Model werden künstliche Zeitreihen für jede Wettergröße erstellt.
- Diese künstlichen Zeitreihen werden in die Originalverteilung umgewandelt.

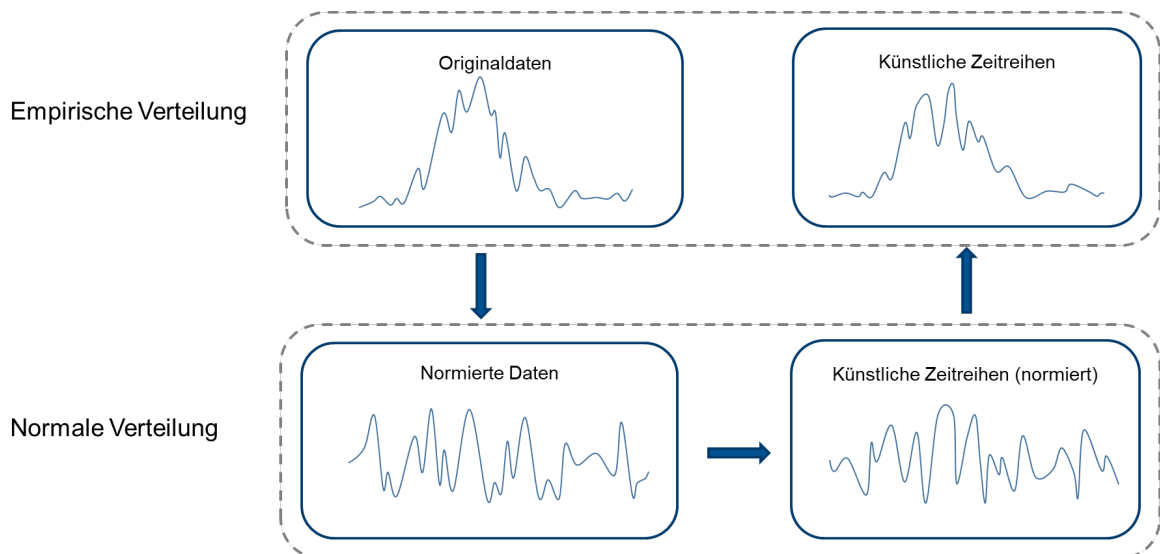


Abbildung 56: Vorgehensweise für die Erzeugung künstlichen Wetterzeitreihen

## 7.2.2. Schätzung der Verteilung

Es ist bekannt, dass Kernel-Methoden die höchste Log-Wahrscheinlichkeit erbringen [30]. Für die empirische Schätzung der Verteilung werden also Cosinus Kernels benutzt. Gaussian Kernels wurden ebenfalls ausprobiert, aber Kernel mit begrenzter Domäne waren bevorzugt, um die Stabilität der Ergebnisse und die Geschwindigkeit des Berechnens zu gewährleisten.

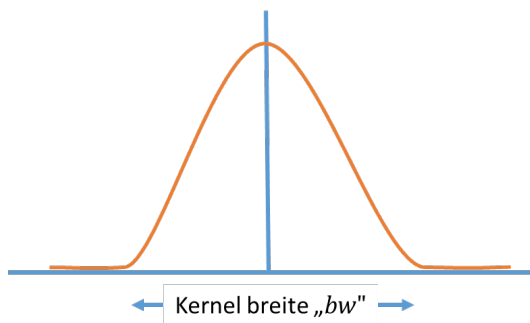


Abbildung 57: Gaussian Kernel

Das Cosinus Kernel  $ker_x$  (siehe Abbildung 57) wird wie folgt berechnet.

$$ker_x(\delta x) = \begin{cases} \frac{1}{bw_x} \left( 1 + \cos\left(\frac{2\pi \delta x}{bw_x}\right) \right) & \text{wenn } \delta x \in \left[-\frac{bw_x}{2}, \frac{bw_x}{2}\right] \\ 0 & \text{sonst} \end{cases} \quad (58)$$

Mit  $bw_x$  die Bandbreite des Kernels

Die Folgende Abbildung zeigt, wie sich die Verteilungsdichte durch Kernels schätzen lässt.

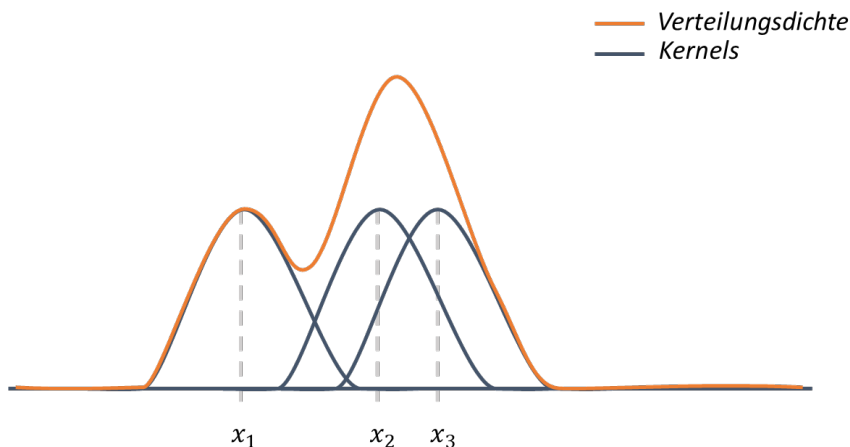


Abbildung 58: Verteilungsschätzung mit Kernels



In dem Beispiel sind es aber nur 3 Punkte, so dass die Schätzung nicht zuverlässig ist. Damit genug Datenpunkte für die Verteilungsschätzung vorliegen, werden für jede Stunde und jeder Tag auch benachbarten Punkten benutzt, aber mit abnehmenden Gewichten:

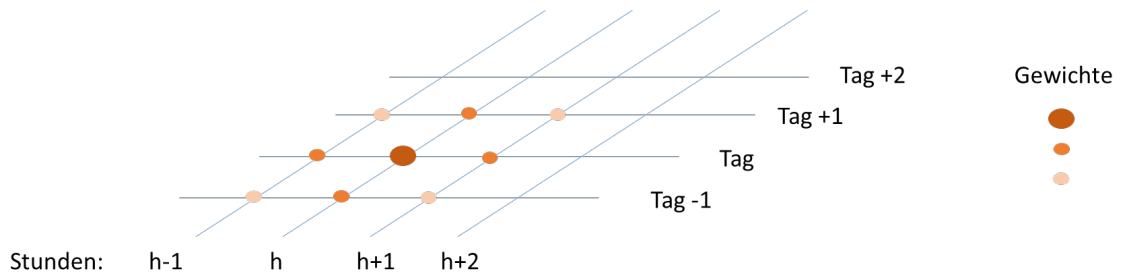


Abbildung 59: die benachbarten Punkte werden auch für die Verteilungsschätzung benutzt

Diesen abnehmenden Gewichten werden auch mit Cosinus Kernels berechnet ( $ker_t, ker_s$  mit Bandbreite  $bw_t, bw_s$  jeweils für Tag und Stunde).

Schließlich ergibt sich die Formel für die empirische Verteilungsdichte:

$$pdf_{Tag,Stunde}(x) = C \sum_{0 \leq s < 24} \sum_{0 \leq t < 366} \sum_{y \in Werte_{s,t}} ker_x(y - x) \times ker_t(Tag - t) \times ker_s(Stunde - s) \quad (59)$$

Wobei die Verteilungsdichte „pdf“ genannt wird („probability density function“)

Die Konstante  $C$  wird so ausgewählt, dass die Verteilungsdichte zu 1 integriert.

Die empirische (kumulierte) Verteilungsfunktion  $F_{Tag,Stunde}$  ergibt sich in ähnliche Weise.

Dafür wird die Cosinus Kernel  $ker_x$  durch das integrierte Kernel ersetzt.

### 7.2.3. Normierung den Zeitreihen

Mit der empirische und Gaussian kumulative Verteilungsfunktionen  $F_{empirisch}$  und  $F_{Gaussian}$  werden die Zeitreihen  $T$  und  $L$  (Temperatur und Luftfeuchte) umverteilt. Die resultierenden Zeitreihen  $\bar{T}$  und  $\bar{L}$  sind dann normalverteilt.

$$\begin{aligned}\bar{T} &= F_{normal}^{-1} \circ F_{empirisch}(T) \\ \bar{L} &= F_{normal}^{-1} \circ F_{empirisch}(L)\end{aligned}\tag{60}$$

Hier ist zu berücksichtigen, dass die empirische Verteilung abhängig der Stunde und des Tages ist.

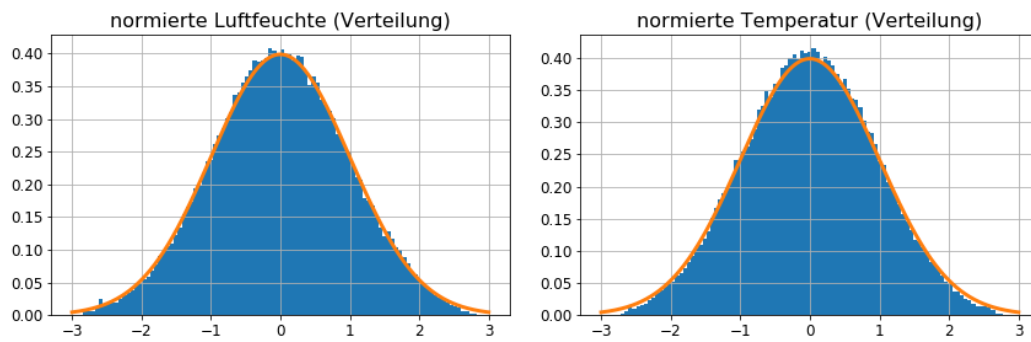


Abbildung 60: Normierte Zeitreihen (Luftfeuchte links und Temperatur rechts)

## 7.2.4. AR Model

Anhand der Autokorrelationsfunktionen und der partiellen Autokorrelationsfunktionen (Abbildung 61) wird ein AR(2) Modell ausgewählt:

$$T_t = \phi_1 T_{t-1} + \phi_2 T_{t-2} + \psi_1 L_{t-1} + \psi_2 L_{t-2} \quad (61)$$

$$\begin{cases} T_t = \phi_1 T_{t-1} + \phi_2 T_{t-2} + \psi_1 L_{t-1} + \psi_2 L_{t-2} \\ L_t = \phi_1^* T_{t-1} + \phi_2^* T_{t-2} + \psi_1^* L_{t-1} + \psi_2^* L_{t-2} \end{cases}$$

In dieser zwei-dimensionalen AR(2) Model wird auch die Korrelation zwischen Temperatur und Luftfeuchte berücksichtigt.

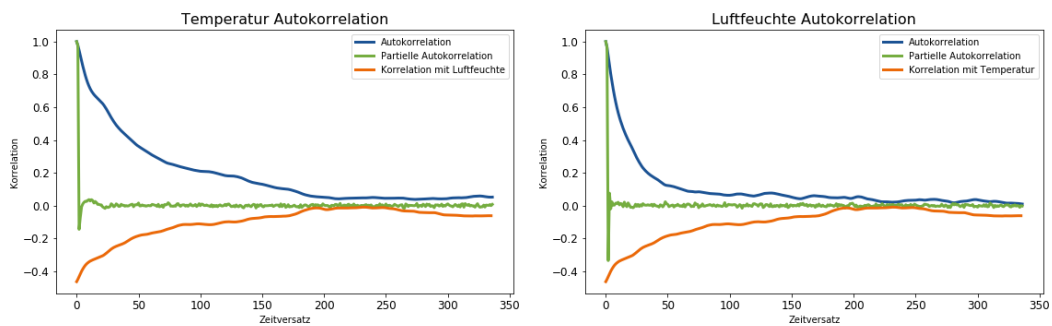


Abbildung 61: Autokorrelation von Wetterdaten

## 7.2.5. Erste Ergebnisse

Die ersten Ergebnisse wurden mit lediglich vier Jahren (2015 bis 2018) berechnet.

Es wird trotzdem festgestellt, dass die künstlichen Zeitreihen die Originaldaten ähneln:

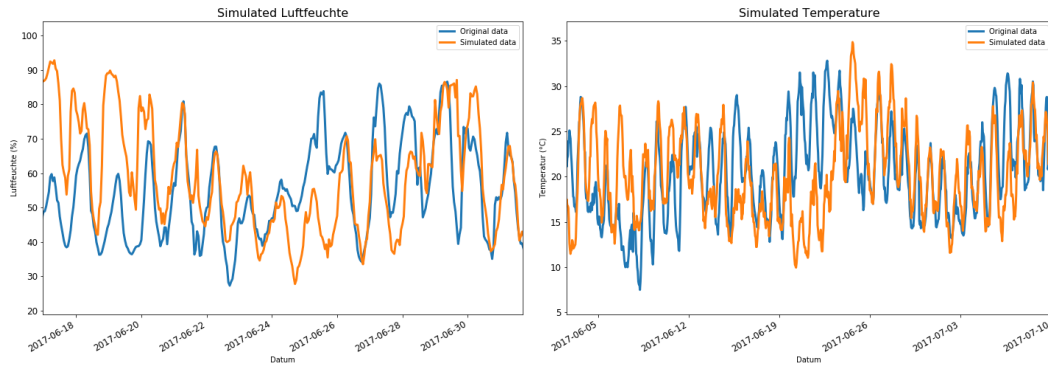


Abbildung 62: Vergleich von die künstlichen Zeitreihen

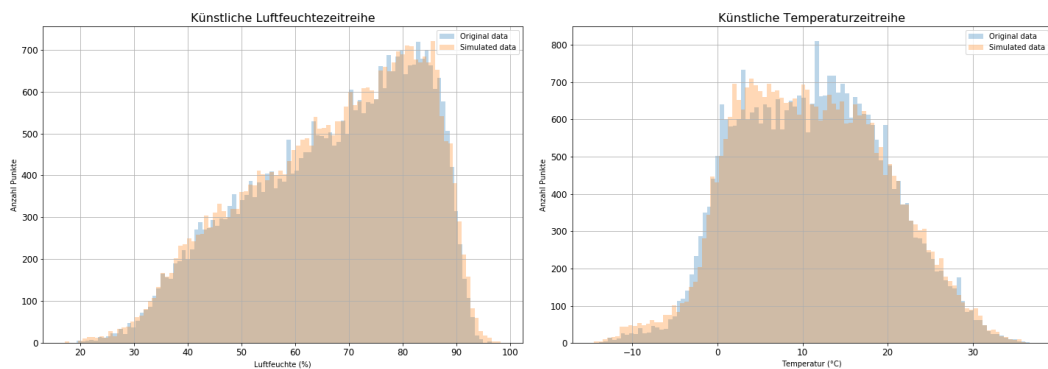


Abbildung 63: Histogramm von künstlichen Zeitreihen

Drei Verbesserungen wären aber noch wünschenswert.

- Der Temperaturanstieg sollte berücksichtigt werden.
- Es sollten auch extremen Jahren berechnet werden
- In der Realität gibt es große Temperaturunterschiede, wenn das Himmel klar ist, und niedrige Temperaturunterschiede, wenn es bewölkt ist. Mathematisch gesehen steht die Varianz der Temperatur nicht fest, sondern es ist auch zufällig. Deswegen sollte der Model „Heteroskedastik“ sein.

## 7.2.6. Verbesserungen

### Temperaturanstieg

Um der Temperaturanstieg zu berücksichtigen, wird eine Vorgehensweise in vier Schritte vorgeschlagen:

- Der Temperaturanstieg wird in den Originaldaten gemessen und entfernt
- Mit diesen Wachstum bereinigten Daten werden Künstliche Zeitreihen erstellt
- Anhand von Klimastudien werden Temperaturanstieg Annahme festgestellt
- Der Temperaturanstieg wird zu den erzeugten Zeitreihen addiert

Nach eine kleine Literaturrecherche ( [31] [32]) werden die Folgenden Annahme benutzt:

- Extrem heiß: +1,7°C in die nächsten 50 Jahren
- Normal: +1,1°C in die nächsten 50 Jahren
- Kühl: +0.9°C in die nächsten 50 Jahren

### Extreme Szenarien

Eine Zeitreihe mit kühlem Sommer und eine Zeitreihe mit heißen Sommer wird berechnet.

Zu diesen Zweck werden die 2 extreme Jahre in den Vergangenen Jahren betrachtet, 2003 (heiß) und 2005 (kühl).

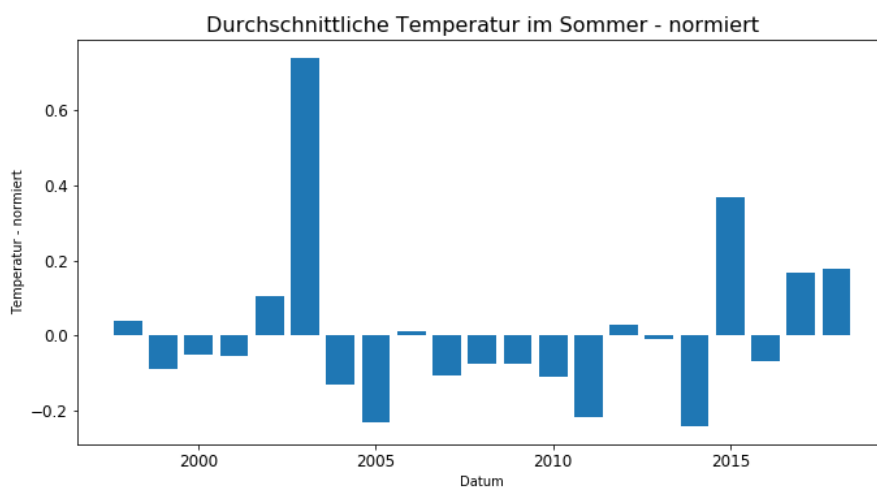
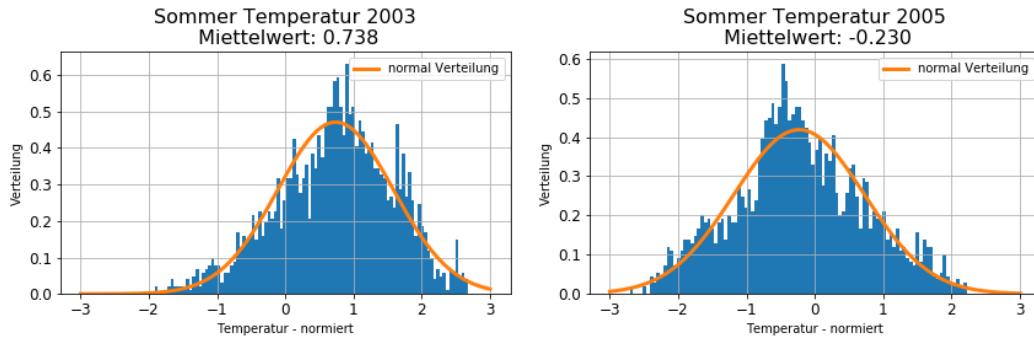


Abbildung 64: Durchschnittliche Temperatur im Sommer (Normierte Zeitreihe)

Es wird dann gemessen, wie viel die normierten Zeitreihen im Sommer 2003 und im Sommer 2005 von der normalen Verteilung abweichen, wie in Abbildung 65 abgebildet. (Für normalen Jahren liegt der Mittelwert der normierte Temperatur nahe Null)



**Abbildung 65: Temperaturabweichung im Sommer 2003 und 2005 (normierte Zeitreihe)**

Um extremen Jahren zu simulieren, wird die simulierte normierte Zeitreihe im Sommer entsprechend skaliert, um dieselbe Varianz und Mittelwert wie Sommer 2003 oder Sommer 2005 aufzuweisen. Nur danach wird die Zeitreihe in Realen Temperatur- oder Luftfeuchtbereich umwandelt.

Wegen der Korrelation zwischen Luftfeuchte und Temperatur, wird nicht nur die Temperaturzeitreihe, sondern auch die Luftfeuchtezeitreihe skaliert.

## Heteroskedastizität

Ziel ist es, dass die erzeugten normierten Zeitreihen eine zufällige Varianz aufweisen, mit demselben Verhalten wie die normierten Originaldaten.

Die folgende Vorgehensweise wird implementiert:

- Nachdem der AR Model kalibriert wird, können die Residuen berechnet werden (Prognosefehler der AR Model).
- Dann wird die gleitende Varianz auf einem Tag berechnet.
- In den nächsten Schritten wird eine künstliche Varianzzeitreihe wie bei Wetterzeitreihen erstellt.
  - Die Varianzverteilung der Residuen wird geschätzt.
  - Es wird ein AR(1) Model auf den normierte Residuen kalibriert
  - Eine künstliche normierte Varianzzeitreihe wird berechnet
  - Die künstliche Zeitreihe wird in der Originalverteilung umwandelt
- Anhand dieser künstliche Varianzzeitreihe wird eine heteroskedastisches Rauschen erstellt
- Das weißes Rauschen in den AR(2) Model (für normierte Wetterdaten) wird durch dieses Rauschen ersetzt

Ein zusätzlicher Schritt wird hinzugefügt, um die Stabilität dieser Ansatz zu gewährleisten.

- Die künstliche Zeitreihe wird umverteilt, um eine normale Verteilung zu entsprechen.

## 7.2.7. Zusammenfassung und Ergebnisse

In der folgenden Abbildung wird die gesamte Vorgehensweise dargestellt, mit den zusätzlichen Verbesserungen. In Grün ist der Ansatz, um extreme Sommer zu berechnen, und in Hellblau die Ansatz, um Heteroskedastizität zu simulieren.

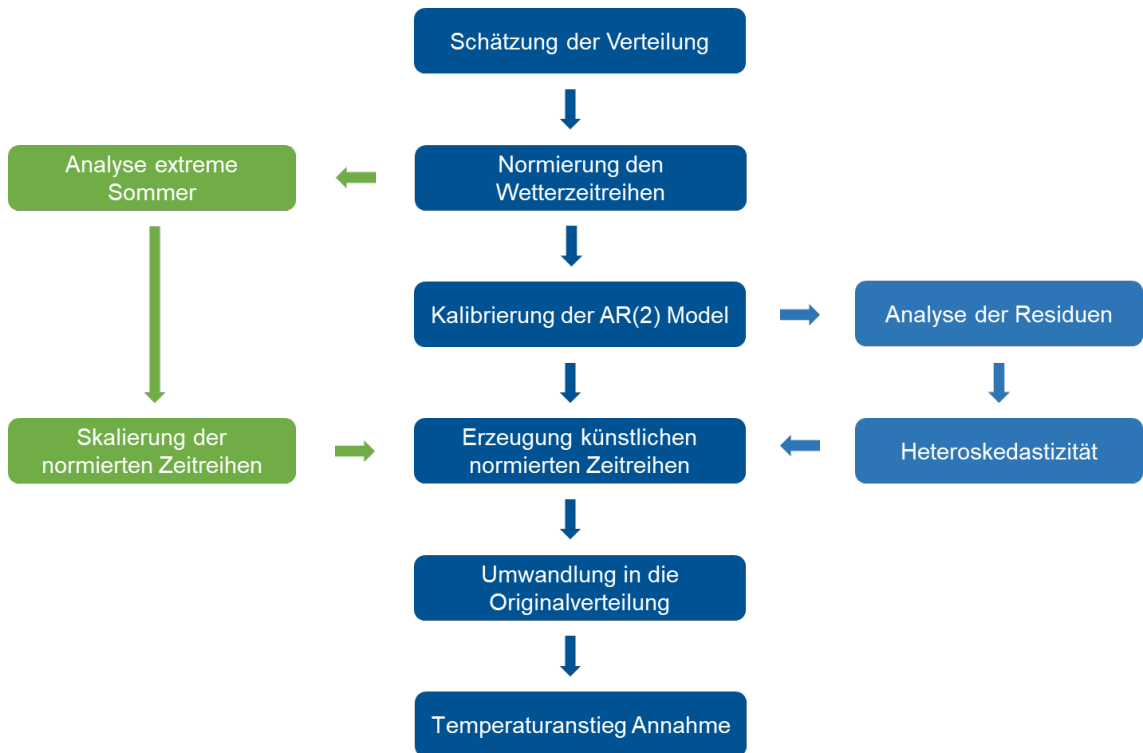
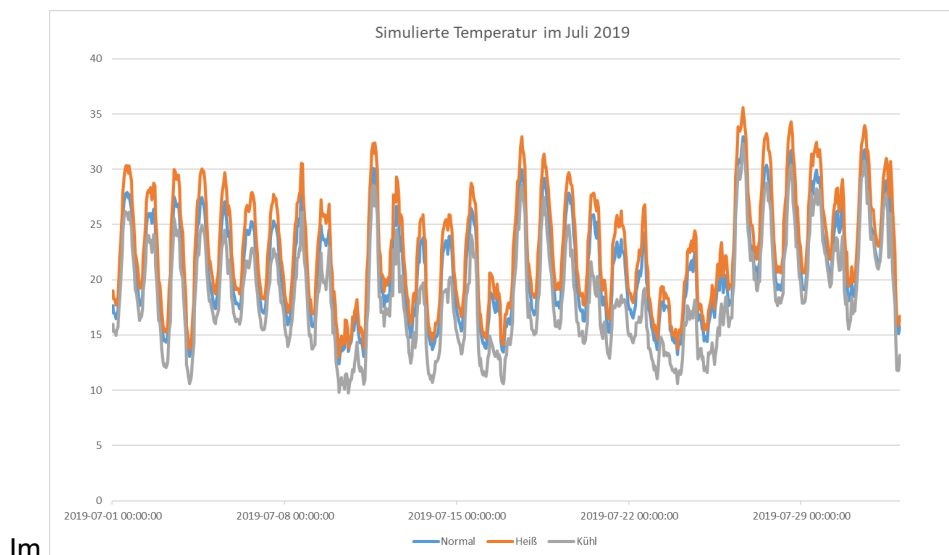


Abbildung 66: Vorgehensweise für die Erzeugung von künstlichen Wetterzeitreihen



Im Abbildung 67 werden die Ergebnisse dargestellt. Leider konnte in dem Zeitraum keine ausführliche Bewertung der Ergebnisse aufgeführt werden.



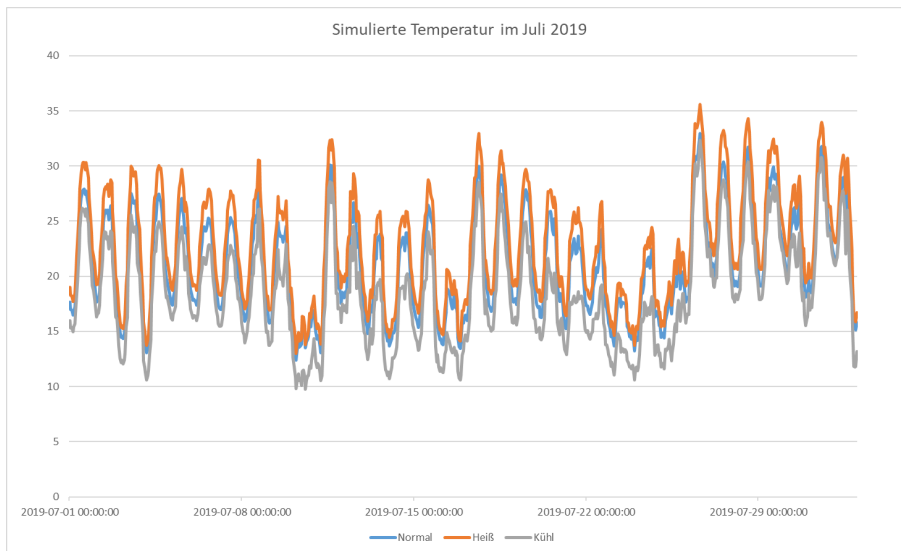


Abbildung 67: Simulierte Temperatur (normal, heiß und kühl)

### 7.2.8. Beschränkungen

Die Abhängigkeit zwischen Temperatur und Luftfeuchte wird nicht gut berücksichtigt (siehe Abbildung 68 und Abbildung 69).

Dazu ergeben sich Ausreißer bei der Luftfeuchte (bis zu 102% Feuchtigkeit!). Die Verteilungsschätzung mit Kernel sollte nämlich ausgeführt werden, wenn sie Verteilung begrenzt ist.

Um langfristige Prognose zu erstellen wird also empfohlen, nur die simulierte Temperaturzeitreihe zu nützen.

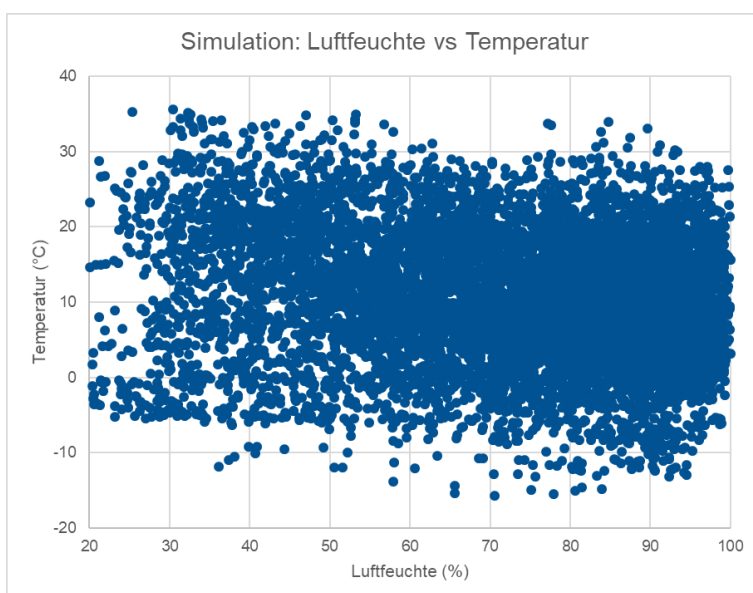
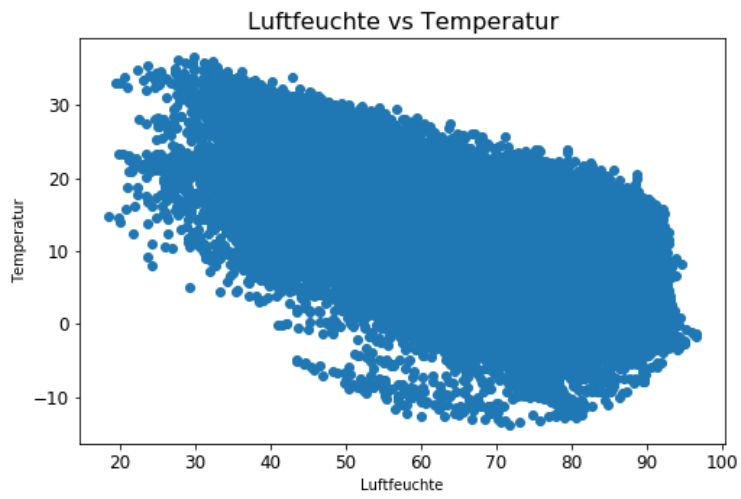


Abbildung 68: Luftfeuchte versus Temperatur (simulierte Daten)



**Abbildung 69: Luftfeuchte versus Temperatur (ist-Daten)**

## 8. Umsetzung im BoFiT

Das vom Unternehmen ProCom entwickelte Programm BoFiT dient hauptsächlich dem Zeitreihenmanagement, der Prognoseerstellung und der Modellierung von Energiesysteme. Für die Einsatzplanung bei den SWM ist es ein effizientes Tool im täglichen Einsatz. Ziel ist es, die Optimierungsmodelle für die Fernkälte-Erzeugungsanlagen auch in BoFiT zu programmieren.

Die besten Prognosemodelle (für kurz- und langfristige Prognose) werden in BoFiT umgesetzt, so weit wie es möglich ist. Es gibt nämlich Restriktionen, die berücksichtigt werden müssen.

### 8.1. Zusammenfassung BoFiT

In BoFiT wird in Form von Zeitreihen gerechnet. Zu jeder dieser Zeitreihen gehört ein gewisses Zeitraster sowie eine Einheit wie Grad Celsius oder Megawatt. Dadurch werden Zeitrasterumwandlungen und fehlende Werte automatisch berücksichtigt. Leider ist dies gleichzeitig eine Restriktion, es gibt keine Variable im engeren Sinn.

Folgende Objekte können erstellt werden:

- Formeln: Um neue Zeitreihen zu berechnen oder mit mehreren Zeitreihen zu rechnen.
- Zeitreihen
- Transfergruppen: Datenimport und Export.
- Abläufe: Legt eine Reihenfolge von Berechnungen fest.
- Modelle: Beispielweise von Energiesystemen.
- Prognosen

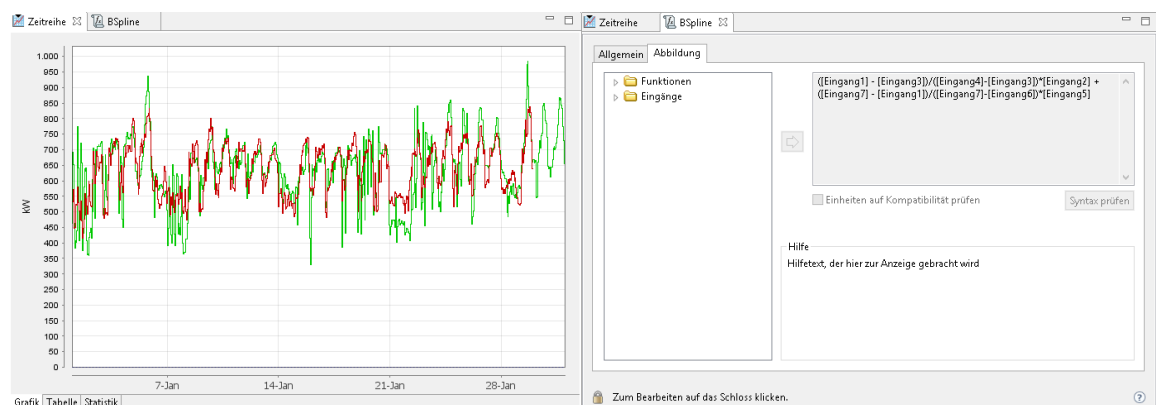


Abbildung 70: BoFiT - Visualisierung einer Zeitreihe und Erstellung einer Formel

## 8.2. Mögliche Verfahren

Um eine Prognose zu erstellen, stehen verschiedenen Verfahren zu Verfügung:

- Multivariate Regression
- SARIMAX (Saisonal AutoRegressive Integrated Moving Average with Exogenous inputs)
- Künstliches neuronales Netz
- Lastprofile und andere festgelegte Modelle

Die Multivariate Regression ist immer das gleiche Verfahren, so dass alle Regression basierten Modellen im BoFiT übertragbar sind. (Im BoFiT können zusätzlich Ausreißer vor der Regression entfernt werden)

Den SARIMAX Model ist eine Kombination von Regression und ARIMA Verfahren (mit Berücksichtigung von historischen Daten) mit Saisonale Ebene. BoFiT kann auch selbst die Metaparameter des ARIMA Model bestimmen (z.B. die Zahl von Zeitverzögerungen). Zwei Restriktionen behindern die Anwendung dieses Verfahren. Es gibt eine beschränkte Auswahl von Ebene, und das Model kann nicht auf Daten trainiert werden, die Lücke am Anfang und am Ende des Zeitraums aufweisen.

Das Künstliches Neuronales Netz ist vielversprechend, und der Benutzer kann die Trainingsparameter selbst festlegen, aber die Struktur des neuronalen Netzes ist unbekannt und kann nicht am Problem gepasst werden.

Das Regression basiertes Modell mit optimalen Eingängen und 168 Ebene (Wochentage und Stunden) wurde mit den künstlichen Netz im BoFiT verglichen. Mit eine Trainingszeitraum von 2015 bis 2017 und eine Validierungszeitraum von Januar 2018 bis Oktober 2018 ergeben sich die Folgenden Prognosefehler (für die Wachstum bereinigten Last).

		Linear	Neuronales Netz	Neuronales Netz	Neuronales Netz
Ebene	Stunden	✓			✓
	Wochentage	✓		✓	✓
Prognosefehler	RMSE (kW)	169,73	283,71	279,71	211,33

Daher wird beschlossen, Regression basierten Modelle mit optimierten Eingängen und Ebene und mit zusätzlicher Vor- und Nachbearbeitung, im BoFiT zu implementieren.

## 8.3. Prognosemodelle in BoFiT

Am Ende sollen zwei Modelle in BoFiT zur Verfügung stehen: Eines für die kurzfristige und eines für die langfristige Prognose. Grund dafür sind zwei wichtige Unterschiede:

- Für die kurzfristige Prognose kann man die Ist-Last der vorherigen Tage als Eingangsparmeter nutzen.
- Für langfristige Szenarien sollte das Netzwachstum wegen künftiger Kunden und Temperaturen Anstieg berücksichtigt werden.

### 8.3.1. Kurzfristige Prognose

In dieser Arbeit liegt der Fokus auf die Prognose vor zwei Tage. In der praktischen Anwendung sollten aber Prognose für die nächsten 2 Wochen erstellt werden.

Um die Genauigkeit der Prognose für die nächsten Tagen zu bewahren werden 6 Zeithorizont betrachtet: 24h, 48h, 72h, 96h, 168h und größer 168h. Für jeder Zeithorizont wird eine multivariate Regression individuell trainiert. Am Ende wird die Prognose der Modelle benutzt, die den geringsten Zeithorizont aufweisen.

Der Model besteht also aus 6 multivariaten Regressionen, jeder mit 168 Ebene, und dazu Vor- und Nachbearbeitungsschritte wie im Abbildung 71 dargestellt.

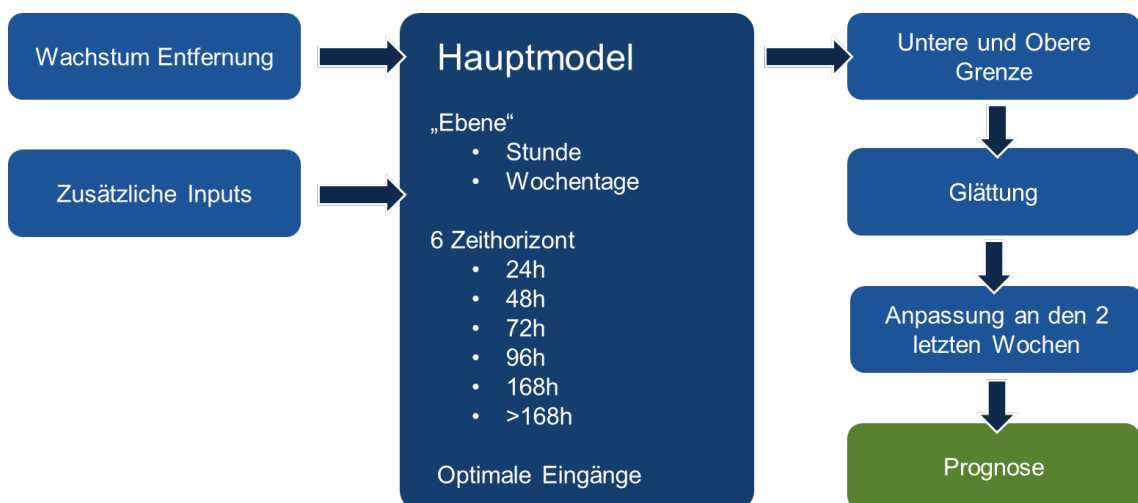


Abbildung 71: Kurzfristiges Model im BoFiT

Diese Bearbeitungsschritte (Schätzung des Wachstums, untere und obere Grenze, Glättung der Prognose und Anpassung an die zwei letzten Wochen) sind im Kapitel 5 beschrieben.

Die Eingänge der Regression mit 48h Zeithorizont ergeben sich aus die Parameterauswahl Verfahren (siehe 5.3.1). Es sind:

- Die Temperatur
- Der S-Spline der Temperatur mit Stützpunkten 10°C und 20°C und polynomial Grad 2
- Der S-Spline der Temperatur vor 48h mit Stützpunkten 10°C und 20°C und polynomial Grad 2
- Die ist-Last vor 48h (zwei Tage)
- Die ist-Last vor 168h (sieben Tage)
- Die Tagesmittelwert der ist-Last vor 2 Tage
- Die Feiertage
- Die Globalstrahlung

Für andere Zeithorizont werden dieselben Eingänge benutzt außer für die ist-Last.

Abhängig von die Zeithorizont werden die entsprechend verfügbaren Lastdaten benutzt.

Im Abbildung 72: Struktur des kurzfristigen Models im BoFiT Abbildung 72 wird die Struktur des kurzfristigen Models im BoFiT dargestellt. Links sind die unterschiedlichen Zeitreihen und rechts die entsprechenden Regressionsmodellen.



Abbildung 72: Struktur des kurzfristigen Models im BoFiT

### 8.3.2. Langfristige Szenarien

Für langfristige Szenarien sollten folgende Punkte berücksichtigt werden. Neue Kunden können zum Netz hinzukommen (Netzzuwachs). Auch das Wetter kann sich langfristig ändern.

Das langfristige Prognosemodell weist eine ähnliche Struktur wie den kurzfristigen Modell aus. Der Hauptteil des Modells ist eine multivariate Regression mit 168 Ebene (Wochentage und Stunde). Dazu wird die Prognosegüte durch Vor- und Nachbearbeitung verbessert. Das ist in der folgenden Abbildung dargestellt.

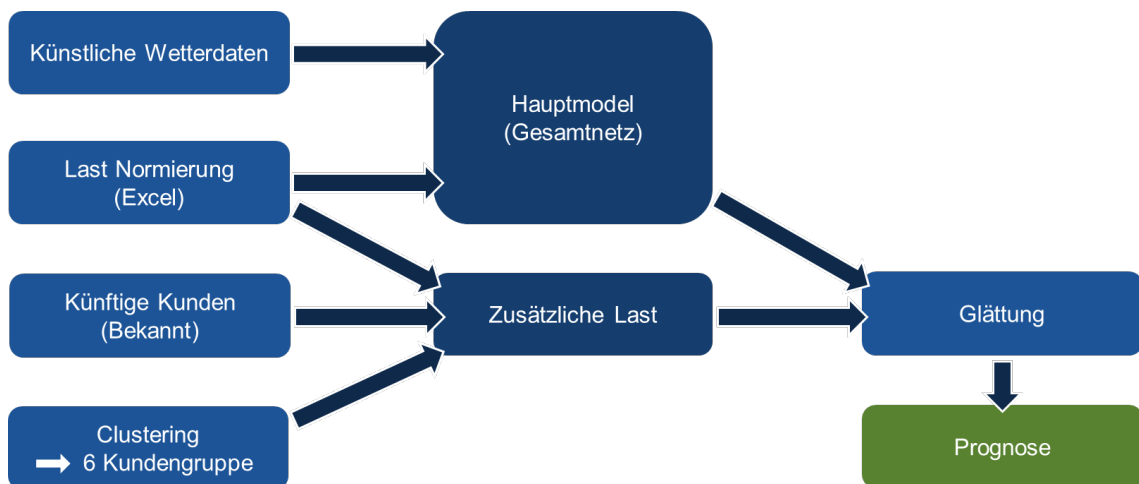


Abbildung 73: Langfristiges Netz im BoFIT

Anstelle der Wachstumsberechnung wird hier die Last im Vorfeld normiert, im Excel. Dieser Schritt ist in Abschnitt 7.1.1 beschrieben.

Dazu wird die Ergebnisse des Kundenclustering benutzt (siehe 3.2.2), um die Last von künftigen Kunden zu prognostizieren. (Das ist wiederum in Abschnitt 7.1.2 erläutert).

Um eine mögliche Wetteränderung in das Modell zu implementieren, können entweder die Testreferenzzeitreihen des Deutschen Wetterdienstes genutzt werden oder simulierten Wetterdaten (siehe 7.2).

Am Anfang der Berechnung sollte der Benutzer zwei Punkte entscheiden:

- Welche Wetterzeitreihen (normal oder extrem) benutzt wird
- Was ist das erwartete Wachstum der jeweiligen Kundengruppe

Zum Beispiel werden in der folgenden Abbildung drei Szenarien dargestellt.

- In dem normalen Szenario wird kein Netzzuwachs angenommen.
- In dem Szenario mit heißem Sommer wird die extreme Temperatur Zeitreihe benutzt.
- In dem Szenario mit zusätzlicher Last wird angenommen, dass 2 neue Kunde von Kundengruppe 1 pro Jahr neu gebunden sind, sowie 1/2 Kunde von Kundengruppe 2.

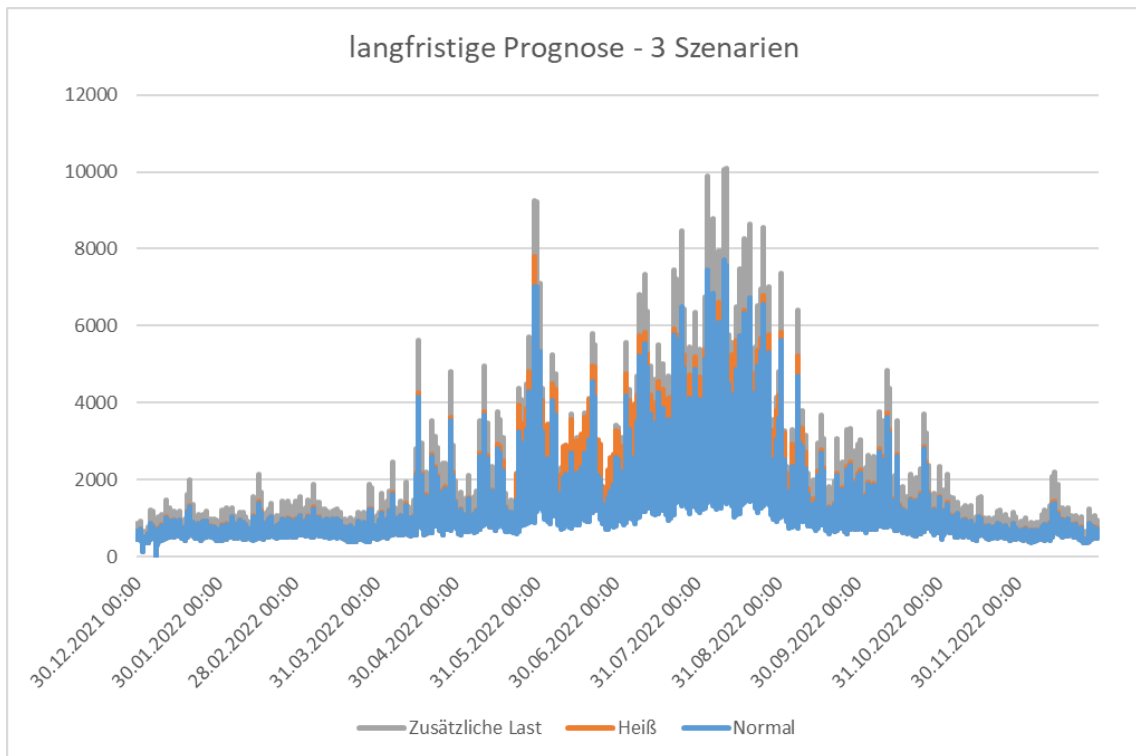


Abbildung 74: Langfristige Prognose - 3 Szenarien



## 8.4. Ergebnisse

Es wurden Prognosen für die zwei ersten Wochen von Februar bis Juli 2018 erstellt und mit der Ist-Last verglichen. Für jede Prognose wurden nur vorherige Datenpunkte benutzt.

Das Gleitender Horizont Verfahren ist mit der Python Prognoseaufwertung vergleichbar. Hierbei ist auffällig, dass die Last im Winter ungleichmäßiger wird, was das Prognostizieren erschwert. Allerdings ist die Last auch geringer, so dass der absolute Fehler im Winter trotzdem kleiner ist. Für die dargestellte 24h Prognose beträgt die RMSE 37,14kW im Winter und 182,88kW im Sommer.

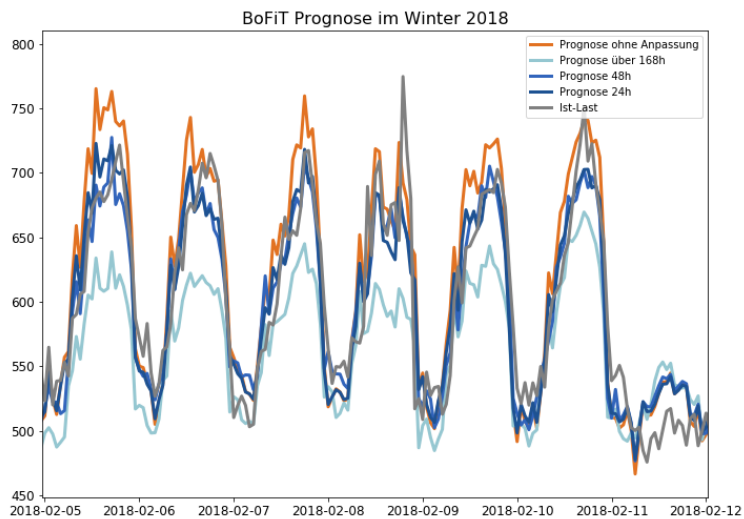


Abbildung 75: BoFiT Prognose im Winter 2018

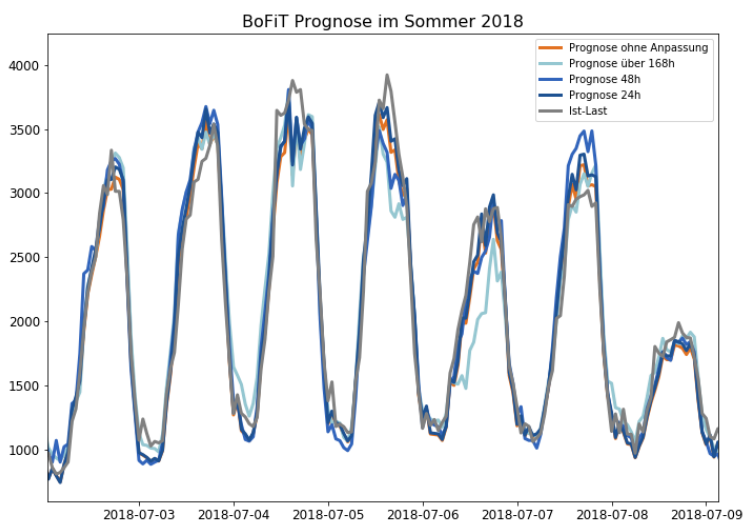


Abbildung 76: BoFiT Prognose im Sommer 2018

In BoFiT erhält man eine ähnliche Prognosegüte wie im Python mit den Regressionsmodellen. Interessant ist es auch, verschiedene Zeithorizonte miteinander zu vergleichen. Auffällig ist, dass der letzte Schritt, die Anpassung an die zwei letzten Wochen, eine deutliche Verbesserung bringt.

Beschreibung	RMSE (kW)	relativer RMSE
Prognose ohne Anpassung (24h)	179,29	20,37%
Prognose 24h	162,90	18,51%
Prognose 48h	178,58	20,29%
Prognose 72h	189,34	21,52%
Prognose 96h	186,75	21,22%
Prognose 168h	196,14	22,29%
Prognose > 168h (ohne ist-Last)	241,37	27,43%

Tabelle 15: Ergebnisse der Modelle für unterschiedliche Zeithorizonte

Mit Hilfe von Normal Kerndichteschätzern kann man die Fehlerverteilung analysieren:

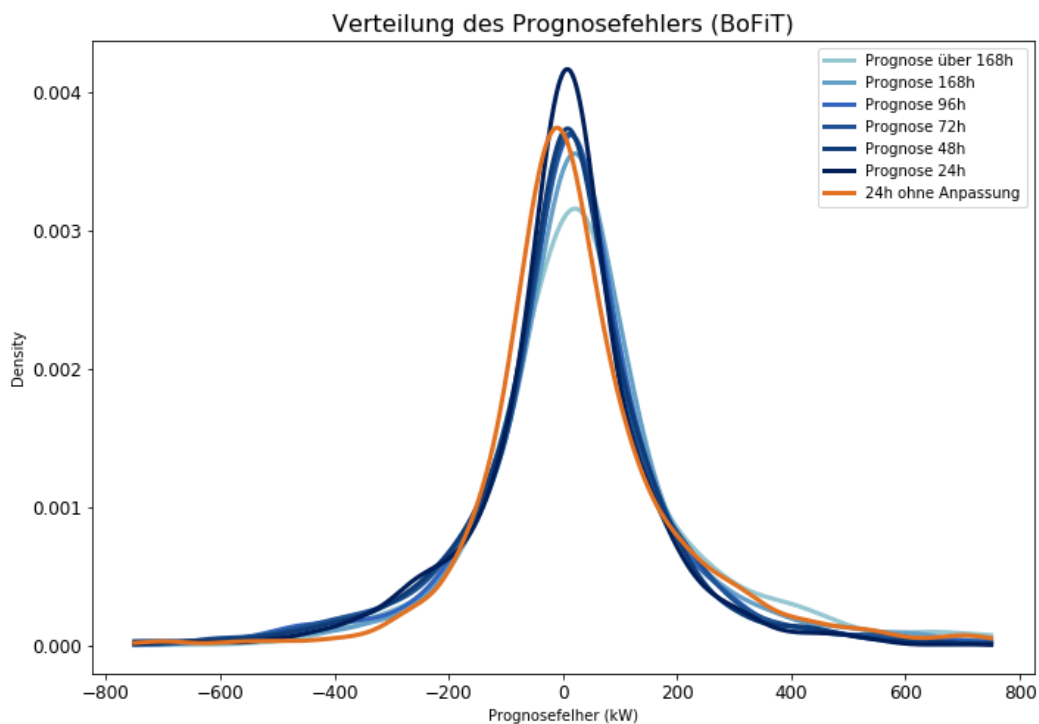


Abbildung 77: Verteilung des Prognosefehlers der BoFiT Berechnung

## 9. Zusammenfassung und Ausblick

Nach der explorativen Datenanalyse und auf Basis der Literatur werden vier verschiedene Ansätze mit verschiedenen Varianten implementiert.

Durch ein Gleitender Horizont Verfahren werden den folgenden Ergebnissen ermittelt:

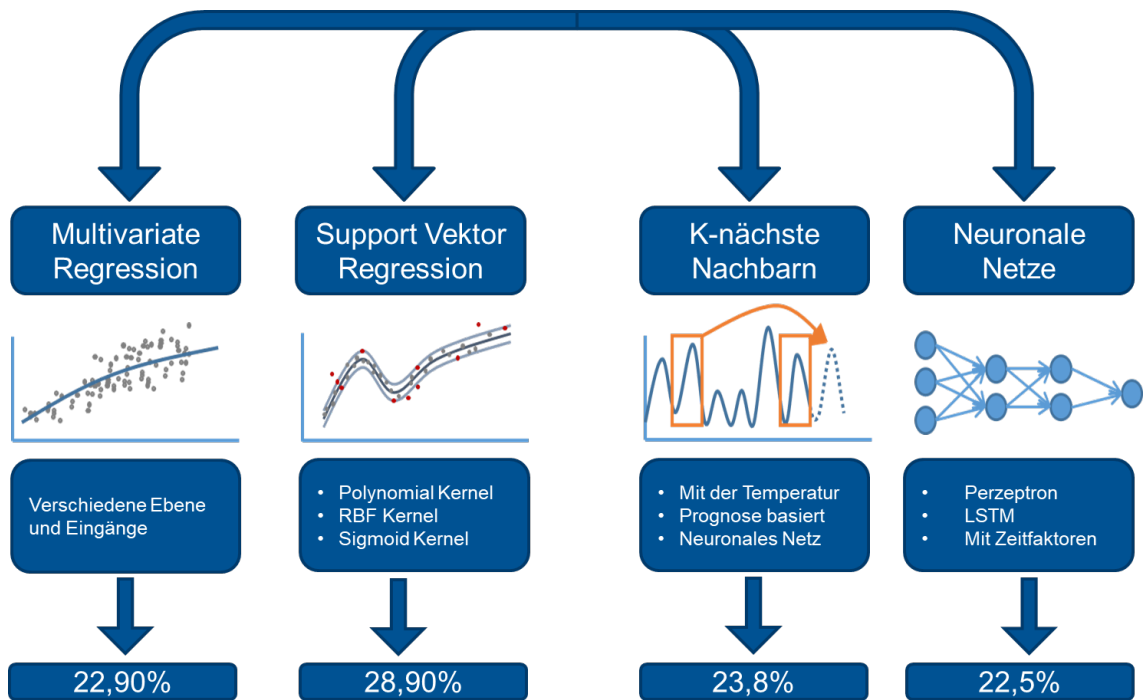


Abbildung 78: Ergebnisse für die verschiedenen Ansätze

Diese Prognosefehler werden mit den jeweiligen besten Modellen erzeugt:

- Linear Model 5: Multivariate Regression mit 168 Ebenen (Stunden und Wochentage)
- Support Vektor Regression mit RBF („Radial Basis Function“) Kernel und 6 Ebene (Jahreszeiten und Feiertage)
- K-nächste Nachbarn auf Basis der Prognose von Linear Model 5
- Neuronale Netz mit Zeitfaktoren

Es werden in einem zweiten Schritt verschiedene Verbesserungen vorgeschlagen:

- Schätzung des Wachstums
- Entfernung von Ausreißern mithilfe einer unteren und oberen Grenze
- Anpassung der Prognose an die zwei letzten Wochen
- Glättung der Prognose
- Mittelwert von verschiedenen Prognosen

Diese zusätzlichen Schritte haben zu unterschiedlich starken Verbesserungen geführt. Lediglich die Entfernung von Ausreißern kann zu einer Verschlechterung der Prognose führen, zeitgleich wird aber eine gewisse Robustheit gewährleistet.

Es wurde auch gezeigt, dass die differenzierte Betrachtung von individuellen Kunden zu einer deutlichen Verbesserung der Prognose führen kann.

Verschiedene Algorithmen wurden implementiert und zum Teil neu entwickelt, um das optimale Eingangsparameterset zu bestimmen. Für diesen Anwendungsfall war die Lokale Suche mit Bayesscher Inferenz besonders effizient. Diese optimalen Sets wurden für verschiedene Szenarien berechnet, abhängig von der Verfügbarkeit einigen Daten wie Luftfeuchte oder der historischen Last.

Anhand dieser optimierten Parametersets konnte eine Sensitivitätsanalyse durchgeführt werden.

Eine Vorgehensweise in drei Schritten wird vorgeschlagen, um langfristige Prognose zu erstellen, mit Berücksichtigung von zwei Wachstumsfaktoren (Temperaturanstieg und Netzzuwachs)

Auf Basis dieser Studie wurden zwei Modelle in BoFiT implementiert: Eines für die kurzfristige Prognose mit unterschiedlichen Zeithorizonten und eines für die langfristige Prognose. Zusätzlich wurden die Kunden in sechs Gruppen unterteilt, um diese langfristige Prognose verfeinern zu können.

Obwohl verschiedene Prognosemodelle und Ansätze in dieser Studie untersucht und verglichen werden, sind noch weitere Verbesserungen möglich. Bevorzugt könnten neue Vorgehensweisen ausgetestet werden, um mögliche Prognoseausreißer zu entfernen. Beispielsweise könnten Prognosen mit den Ergebnissen eines robusteren Verfahrens verglichen werden.

Dazu könnten auch weitere neuronale Netze aufgebaut werden und andere bekannte Ansätze (GRU statt LSTM) angewendet werden. Es ist auch nicht auszuschließen, dass verfeinerte Trainingsverfahren zu einer präziseren Prognose führen.

Für K-nächste Nachbarn Algorithmen sollte das Wachstum möglicherweise im Vorfeld entfernt werden. Dazu wären andere neuronal-basierten Embeddings interessant zu betrachten.

Support Vektor Regression könnte mit zusätzlichen Ebenen implementiert werden, wenn der Rechneraufwand kein Kriterium ist.

Dazu könnten andere Verfahren betrachtet werden, wie Entscheidungsbaum basiertes Boosting oder das Selektionsverfahren (Lasso). Normale Ridge Regression oder Ridge Regression mit Kernels sind auch üblich, um Overfitting zu vermeiden.

# Literaturverzeichnis

- [1] M. Durst, „Erfassen und Analysieren von Optimierungspotenzialen für die nächste Kältebedarfsperiode unter Betrachtung der bisherigen Leistungsdaten der Kälteinsel-Stachus.“ 2015.
- [2] D. Vengertsev, „Deep Learning Architecture for Univariate Time Series Forecasting,“ Stanford University, 2014.
- [3] K. M. Powell, A. Sriprasad, W. J. Cole und T. F. Edgar, „Heating, cooling, and electrical load forecasting for a large-scale district energy system,“ *Energy*, 2014.
- [4] E. Busseti, I. Osband and S. Wong, „Deep Learning for Time Series Modeling,“ 2012.
- [5] S. Bouktif, A. Fiaz, A. Ouni und M. A. Serhani, „Optimal Deep Learning LSTM Model for Electric Load Forecasting Using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches,“ *energies*, 2018.
- [6] S. Du, M. Pandey und C. Xing, „Modeling Approaches for Time Series Forecasting and Anomaly Detection,“ 2017.
- [7] T. Faber, J. Groß und M. Finkenrath, „Innovative Lastprognosen mit »Deep Learning«-Methoden,“ *EuroHeat&Power*, 2018.
- [8] J. Chung, C. Gulcehre und Y. B. Kyung Hyun Cho, „Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,“ *arXiv*, 2014.
- [9] M. Q. Raza und A. Khosravi, „A review on artificial intelligence based load demand forecasting techniques for smart grid and building,“ *Renewable and Sustainable Energy Reviews*, 2015.
- [10] H. H. Örkücü, „Subset selection in multiple linear regression models: A hybrid of genetic and simulated annealing algorithms,“ *Applied Mathematics and Computation*, 2013.
- [11] T. Mehmood, K. H. Liland, L. Snipen und S. Sæbø, „A review of variable selection methods in Partial Least Squares Regression,“ *Chemometrics and Intelligent Laboratory Systems*, 2012.
- [12] I. Dimoukias und M. Amelin, „Medium and Long-term Probabilistic Heat Demand Forecasting in District Heating,“ *EuroHeat&Power*, 2018.
- [13] T. H. Gobmaier, „Entwicklung und Anwendung einer Methodik zur Synthese zukünftiger Verbraucherlastgänge,“ 2013.
- [14] R. J. Hyndman, „Another look at forecast-accuracy metrics for intermittent demand,“ *Foresight - International Journal of Applied Forecasting*, 2006.
- [15] A. M. Alonso, „Time Series Clustering,“ UC3M, 2018.
- [16] S. Aghabozorgi, S. S. Ali und Y. W. Teh, „Time series clustering – A decade review,“ *Information Systems*, 2014.
- [17] J. Paparrizos und L. Gravano, „Fast and Accurate Time-Series Clustering,“ *ACM Transactions on Database Systems*, 2017.
- [18] G. Seif, „The 5 Clustering Algorithms Data Scientists Need to Know,“ 2018. [Online]. Available: <https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68>.
- [19] C. Klüppelberg, Time Series Analysis (Vorlesungskript), 2018.
- [20] J. M. Monier, „Théorème de Dirichlet,“ in *Analyse MP*, p. 451.
- [21] Warwick University, „L2 convergence of Fourier series,“ in *Mathematics Department Undergraduate Handbook*, 2014.
- [22] J. Taylor und S. Bacallado, „Lecture 17: Smoothing splines, local Regression, und GAMs,“ in *data mining and analysis*, 2018.

- [23] A. Ratnani und E. Sonnendrücker, „B-Splines and IsoGeometric Analysis,“ TUM, 2016.
- [24] A. J. Smola, „A tutorial on support vector regression,“ *Statistics and Computing*, 2004.
- [25] A. Shashua, *Introduction to Machine Learning*, 2008.
- [26] „[https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory),“ Wikipedia. [Online]. [Zugriff am 2019].
- [27] F.-F. Li, J. Johnson und S. Yeung, „Recurrent Neural Networks,“ in *Convolutional Neural Networks for Visual Recognition*, 2017.
- [28] M. N. Laura Leal-Taixé, „Course: Introduction to deep Learning,“ 2018.
- [29] QuantStart Team, „Bayesian inference of a Binomial Proportion - The analytical Approach,“ [Online]. Available: <https://www.quantstart.com/articles/Bayesian-Inference-of-a-Binomial-Proportion-The-Analytical-Approach>. [Zugriff am 2019].
- [30] E. I. Pennec, „Course: statistical learning,“ Ecole Polytechnique, 2015.
- [31] Bayerisches Landesamt für Umwelt, „Auswertung regionaler Klimaprojektionen - Klimabericht Bayern,“ 2012.
- [32] Umweltbundesamt, „Klimafolgen Anpassung - Bundesland Bayern,“ Umweltbundesamt, 2018. [Online]. Available: <https://www.umweltbundesamt.de/themen/klima-energie/klimafolgen-anpassung/bundesland-bayern#textpart-1>.
- [33] M. Ulbrich, *Restringierte Nichtlineare Optimierung*, München, 2018.
- [34] TUM, „TUM Corporate Design,“ München, 2016.
- [35] A. Ng, „Support Vector Machines (CS229 Lecture Notes, Part V)“.
- [36] P.-Y. Lagrée, „Résolution de l'Équation de la chaleur instationnaire,“ in *Résolution de l'Équation de la Chaleur*, 2010.
- [37] L. Suganthi und A. A. Samuel, „Energy models for demand forecasting - A review,“ *Renewable and Sustainable Energy Reviews*, 2012.
- [38] C. Czado, E. Brechmann und N. Krämer, *Generalized linear models with applications*, 2018.
- [39] S. Robertson, „Classifying names with a character level RNN,“ 2017. [Online]. Available: [https://pytorch.org/tutorials/intermediate/char\\_rnn\\_classification\\_tutorial.html](https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html).
- [40] A. Baliyan, K. Gaurav und S. K. Mishra, „A Review of Short Term Load Forecasting using Artificial Neural Network Models,“ *International Conference on Intelligent Computing, Communication & Convergence*, 2015.

# Abbildungsverzeichnis

Abbildung 1: Feedforward NN (Links) vs Rekursive NN (Rechts) .....	11
Abbildung 2: Kreuzvalidierung .....	16
Abbildung 3: Zufällige Kreuzvalidierung .....	16
Abbildung 4: Gleitender Horizont .....	16
Abbildung 5: Kältelast gesamt .....	17
Abbildung 6: Durchschnittliche Last im Winter (rechts) und im Sommer (Links) .....	18
Abbildung 7: Clustering Ergebnisse, wöchentliche Last im Winter (normierte Zeitreihe) .....	22
Abbildung 8: Clustering Ergebnisse, wöchentliche Last im Sommer (normierte Zeitreihe) .....	22
Abbildung 9: Temperatur Abhängigkeit der Last .....	23
Abbildung 10: Wind Abhängigkeit der Last .....	24
Abbildung 11: Zeitliche Korrelation der Last .....	24
Abbildung 12: Autokorrelation der Kältelast .....	25
Abbildung 13: Korrelation mit Wetterdaten .....	26
Abbildung 14: Erstellung eines Prognosemodells .....	27
Abbildung 15: Cosinus statt Stundenzeitreihe .....	28
Abbildung 16: Beispiel von kubischem Spline (Spline mit Grad $q = 3$ ) .....	30
Abbildung 17: B-Spline von Grad 1,2 und 3 auf 5 Stützpunkte (167,333,450,567,833) ...	31
Abbildung 18: B-Splines für multivariate Regression .....	32
Abbildung 19: S-Splines für multivariate Regression .....	32
Abbildung 20: Prognose von Model 1 .....	37
Abbildung 21: Lastprognose von Modelle 5 .....	37
Abbildung 22: Lastprognose Model 5 (Winter) .....	38
Abbildung 23: Lastprognose Model 5 (Sommer) .....	38
Abbildung 24: SVR Algorithmus mit gutem und falschem Startpunkt .....	42
Abbildung 25: SVR Prognose mit RBF Kernel .....	43
Abbildung 26: SVR Prognose mit Sigmoid Kernel .....	43
Abbildung 27: Neuronal Netz Ansatz für K-nächst Nachbarn .....	45
Abbildung 28: Ein „einfaches neuronalen Netz“ .....	48
Abbildung 29: Beispiel eins künstlichen Neurons mit Gewichten $c_1, c_2, c_3$ und $b$ .....	48
Abbildung 30: Einfaches Feedforward Netz (Perzeptron) .....	50
Abbildung 31: LSTM Netz .....	51
Abbildung 32: LSTM Architektur .....	51
Abbildung 33: Netz mit Zeitfaktoren .....	52
Abbildung 34: Netz mit Zeitfaktoren und getrennten historischen und Wetterdaten .....	53



Abbildung 35: Netz mit Zeitfaktoren und LSTM .....	53
Abbildung 36: Prognose mit einfachem Feedforward Netz .....	55
Abbildung 37: Prognose mit Zeitfaktor Netz .....	56
Abbildung 38: Prognose des Netzes mit Zeitfaktoren (Sommer).....	56
Abbildung 39: Original Prognose mit Ausreißer (links) und Verbesserung (rechts) .....	58
Abbildung 40: Die ist-Last schneidet die obere Grenze.....	59
Abbildung 41: Prognose mit weichen Ebenen (Winter) .....	60
Abbildung 42: Glättung der Prognose für das Linear Model 5 (rechts mit weichen Ebenen) .....	61
Abbildung 43: Mischung von 2 Prognose (Netz mit Zeitfaktoren und linear Model 5) .....	62
Abbildung 44: Mischung von 3 Prognose (Netz mit Zeitfaktoren, linear Model 5 und K- nächste Nachbarn 2).....	62
Abbildung 45: Summe von Einzelprognose verglichen mit dem gesamten Verbrauch ....	65
Abbildung 46: Summe von Einzelprognose verglichen mit der Erzeugungslast.....	65
Abbildung 47: Beispiel von Under- und Overfitting .....	68
Abbildung 48: Verlauf des genetischen Algorithmus .....	70
Abbildung 49: Standardisierte Koeffizienten (links: einzelne Inputs, rechts: gruppiert) ....	77
Abbildung 50: Prognosefehler (Schrittweise Parameterentfernung).....	77
Abbildung 51: Verteilung des Prognosefehlers.....	80
Abbildung 52: Konfidenzintervall (Sommer und Winter) .....	80
Abbildung 53: Wärmelast - Abhängigkeit von der Temperatur .....	81
Abbildung 54: Testreferenzzeitreihe 2045 - Jahresdauerlinie der Temperatur.....	84
Abbildung 55: Last Normierung .....	85
Abbildung 56: Vorgehensweise für die Erzeugung künstlichen Wetterzeitreihen.....	87
Abbildung 57: Gaussian Kernel .....	88
Abbildung 58: Verteilungsschätzung mit Kernels .....	88
Abbildung 59: die benachbarten Punkte werden auch für die Verteilungsschätzung benutzt .....	89
Abbildung 60: Normierte Zeitreihen (Luftfeuchte links und Temperatur rechts) .....	90
Abbildung 61: Autokorrelation von Wetterdaten .....	91
Abbildung 62: Vergleich von die künstlichen Zeitreihen .....	92
Abbildung 63: Histogramm von künstlichen Zeitreihen.....	92
Abbildung 64: Durchschnittliche Temperatur im Sommer (Normierte Zeitreihe) .....	93
Abbildung 65: Temperaturabweichung im Sommer 2003 und 2005 (normierte Zeitreihe) .....	94
Abbildung 66: Vorgehensweise für die Erzeugung von künstlichen Wetterzeitreihen.....	96
Abbildung 67: Simulierte Temperatur (normal, heiß und kühl) .....	97
Abbildung 68: Luftfeuchte versus Temperatur (simulierte Daten) .....	97

Abbildung 69: Luftfeuchte versus Temperatur (ist-Daten) .....	98
Abbildung 70: BoFiT - Visualisierung einer Zeitreihe und Erstellung einer Formel .....	99
Abbildung 71: Kurzfristiges Model im BoFiT .....	101
Abbildung 72: Struktur des kurzfristigen Models im BoFiT .....	102
Abbildung 73: Langfristiges Netz im BoFiT .....	103
Abbildung 74: Langfristige Prognose - 3 Szenarien .....	104
Abbildung 75: BoFiT Prognose im Winter 2018 .....	105
Abbildung 76: BoFiT Prognose im Sommer 2018 .....	105
Abbildung 77: Verteilung des Prognosefehlers der BoFiT Berechnung .....	106
Abbildung 78: Ergebnisse für die verschiedenen Ansätze .....	107

# Abkürzungsverzeichnis

## A

Adam.....	<i>Adaptative Moment Estimation</i>
AR.....	<i>Autoregressive Model</i>
ARIMA.....	<i>Autoregressive Integrated Moving Average</i>
ARMA.....	<i>Autoregressive Moving Average</i>

## C

CNN.....	<i>Convolutional Neuronal Net</i>
----------	-----------------------------------

## G

GRU.....	<i>Gated Recurrent Units</i>
----------	------------------------------

## L

LSTM.....	<i>Long Short Term Memory neuronal net, Long Short Term Memory</i>
-----------	--

## M

MAE.....	<i>Mean Absolute Error</i>
MASE.....	<i>Mean Absolute Scaled Error</i>

## N

NN.....	<i>Künstliche Neurale Netze</i>
---------	---------------------------------

## P

pdf.....	<i>Probability Density Function</i>
PLS.....	<i>Partial Least Square</i>

## R

RBF.....	<i>Radial Basis Funktion</i>
<b>ReLU</b> .....	<b><i>Rectified Linear Unit</i></b>
RMSE.....	<i>Root Mean Squared Error</i>

## S

SARIMAX.....	<i>Saisonal AutoRegressiv Integrated Moving Average</i>
STD.....	<i>Standard deviation</i>
SVR.....	<i>Support Vektor Regression</i>
SWM.....	<i>Stadtwerke München</i>

## T

tanh.....	<i>Tangens hyperbolicus</i>
-----------	-----------------------------

# Tabellenverzeichnis

Tabelle 1: Korrelation mit berechneten Inputs .....	26
Tabelle 2: Multivariate Regression - Ergebnisse .....	38
Tabelle 3: Support Vektor Regression - Ergebnisse.....	43
Tabelle 4: K-nächste Nachbarn - erste Ergebnisse mit Cross-Validation .....	47
Tabelle 5: K-nächste Nachbarn - Ergebnisse .....	47
Tabelle 6: Aktivierungsfunktionen .....	49
Tabelle 7: Neuronale Netze - Ergebnisse .....	55
Tabelle 8: Aktualisierung der Bayesscher Inferenz .....	71
Tabelle 9: Optimale Parameterauswahl.....	73
Tabelle 10: Parameterbezeichnung .....	73
Tabelle 11: Grobe Sensitivitätsanalyse (Die drei wichtigsten Parameter werden hervorheben).....	78
Tabelle 12: Sensitivitätsanalyse – die wichtigsten Parameter .....	79
Tabelle 13: Korrelationen zwischen Wetterdaten und Wärmelast .....	81
Tabelle 14: Prognosegüte für Fernwärme und Fernkälte .....	82
Tabelle 15: Ergebnisse der Modelle für unterschiedliche Zeithorizonte .....	106
Tabelle 18: Zeitreihenbezeichnungen.....	119
Tabelle 19: Bestes Parameterset für jede Kundengruppen.....	120
Tabelle 20: Ergebnisse der Lokalen Suche für jede Kundengruppen .....	121
Tabelle 21: Ergebnisse der Parameterauswahl Algorithmen für das Netz mit Zeitfaktoren .....	122
Tabelle 22: Sensitivitätsanalyse - Ergebnisse .....	124

# Anhänge

## I. Support Vektor Regression

Gestartet wird mit einem quadratischen Problem:

$$\min \frac{1}{2} \|c\|^2 \quad (62)$$

$$\begin{cases} y_i - \langle c, x_i \rangle - b \leq \epsilon \\ -y_i + \langle c, x_i \rangle + b \leq \epsilon \end{cases}$$

Dies kann zu unzulässigen Problemen führen. Deshalb werden auch Schlupfvariablen („*slack variable*“) eingeführt: wenn eine Beobachtung außerhalb dieser  $\epsilon$ -Bereich liegt, wird bestraft.

$$\min \frac{1}{2} \|c\|^2 + C \sum_i (\xi_i + \xi_i^*) \quad (63)$$

$$\begin{cases} y_i - \langle c, x_i \rangle - b \leq \epsilon + \xi_i \\ -y_i + \langle c, x_i \rangle + b \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Dieses Problem ist konvex, so dass eine starke Dualität vorliegt [33] [24]. Daher wird das Dual Problem betrachtet. Es kann mit Hilfe der Lagrange Funktion  $\mathcal{L}$  formuliert werden:

$$\mathcal{L}(c, b, \xi_i, \xi_i^*, \alpha_i, \alpha_i^*, \mu_i, \mu_i^*) = \frac{1}{2} \|c\|^2 + C \sum_i (\xi_i + \xi_i^*) \quad (64)$$

$$+ \sum_i \alpha_i (y_i - \langle c, x_i \rangle - b - \epsilon - \xi_i)$$

$$+ \sum_i \alpha_i^* (-y_i + \langle c, x_i \rangle + b - \epsilon - \xi_i^*)$$

$$- \sum_i \mu_i \xi_i + \mu_i^* \xi_i^*$$

Der Dual-Problem lautet:

$$\max -\frac{1}{2} \sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*)$$

$$\begin{cases} \sum_i (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

Interessant ist, dass man implizit zusätzliche Funktionen der Eingangsparameter auch berücksichtigen kann:

Es wird angenommen, dass aus den Beobachtungen  $X = (x_i)_{1 \leq i \leq n} \in R^{n \times p}$  die Werte  $\Phi(X) = (\phi(x_i))_{1 \leq i \leq n} \in R^{n \times q}$  berechnet werden. Danach kann das vorherige Problem in  $\mathbb{R}^p$  statt  $\mathbb{R}^n$  formuliert werden:

$$\begin{aligned} \max & -\frac{1}{2} \sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \phi(x_i), \phi(x_j) \rangle & (65) \\ & -\epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \\ & \begin{cases} \sum_i (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned}$$

Hier ist eine Schreibweise mit einem geänderten Skalarprodukt möglich, welches man Kernel genannt wird. Bisher wurde nur der Standard, euklidisches Skalarprodukt, benutzt:

$$\mathbf{ker}_\phi(x, y) = \langle x, y \rangle_\phi = \langle \phi(x), \phi(y) \rangle \quad (66)$$

So lautet das Problem:

$$\begin{aligned} \max & -\frac{1}{2} \sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle_\phi - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \\ & \begin{cases} \sum_i (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned}$$

## II. Parameterauswahl: Weitere Ergebnisse

Es werden die folgenden Bezeichnungen benutzt:

<b>Reference_copy</b>	Kältelast
<b>XXX_SsplineN1_N2_N3</b>	S-Spline von Grad 2 mit Stützpunkten $N2$ und $N3$ bezogen auf Parameter $XXX$
<b>XXX_index-N</b>	Zeitreihe $XXX$ mit $N$ Stunden Versatz
<b>XXX_mean-N</b>	Tagesmittelwert der Zeitreihe $XXX$ mit $N$ Stunden Versatz (zwischen $N - 12h$ und $N + 12h$ )
<b>yearCosN</b>	$\cos\left(\frac{Zeit}{Jahr} \times 2\pi \times N\right)$
<b>yearTrend</b>	Zeit in Jahren
<b>T_square</b>	Temperatur Quadrat
<b>T_relu</b>	$\max(Temperatur - 15,0)$
<b>T_XXX</b>	Temperatur_XXX

Tabelle 16: Zeitreihenbezeichnungen

Modell	Variante 6	Variante 6	Variante 6	Variante 6	Variante 6	Variante 6	Variante 6
Beschreibung	AufGruppe1	AufGruppe2	AufGruppe3	AufGruppe4	AufGruppe5	AufGruppe6	
Mit Historische Last	Nein	Nein	Nein	Nein	Nein	Nein	
mit Luftfeuchte	Ja	Ja	Ja	Ja	Ja	Ja	
Mit S-Splines	Ja	Ja	Ja	Ja	Ja	Ja	
Methode	Lokal	Lokal	Lokal	Lokal	Lokal	Lokal	
Fehler (relative)	0,2436	0,5811	0,1666	0,4661	0,5491	0,5052	
Zeit	01:25:41	01:33:01	01:31:42	01:24:43	01:28:48	01:07:56	
	Enthalpy_Sspline2_20_40	Enthalpy	Enthalpy	Feiertag	Feiertag	Enthalpy_Sspline1_-10_20	
	Feiertag	Feiertag	Enthalpy_Sspline1_-10_20	Luftfeuchte_Sspline3_67_100	Luftfeuchte	Feiertag	
	Luftfeuchte_Sspline2_33_67	Luftfeuchte_Sspline1_0_33	Enthalpy_Sspline2_20_40	T_index-1_Sspline1_10_20	Luftfeuchte_Sspline3_67_100	Feiertag	
	Sonne_Sspline3_0_1	Luftfeuchte_Sspline2_33_67	Feiertag	T_index-24	Luftfeuchte_Sspline3_67_100	T_index-1	
	Temperatur	Luftfeuchte_Sspline3_67_100	T_relu	T_index-24_Sspline1_-10_10	T_square	T_index-48_Sspline2_10_20	
	T_index-1	Temperatur	Temperatur_Sspline3_20_30	T_index-72	Temperatur_Sspline1_-10_10	T_index-168	
	T_index-2_Sspline1_10_20	Temperatur_Sspline2_10_20	T_index-1	T_index-96	T_index-168	T_index-168	
	T_index-48_Sspline2_10_20	T_index-168	T_index-168	T_index-24	T_index-24_Sspline1_-10_10	T_index-168	
	T_index-168	T_index-24	T_index-1_Sspline1_10_20	yearCos2	T_index-24_Sspline2_10_20	T_index-168	
	T_index-72	T_index-24_Sspline1_-10_10	T_index-48	yearTrend	T_index-48	yearSin1	
	T_index0_Sspline1_-10_10	T_index-4	T_index-48_Sspline2_10_20		T_index-48_Sspline1_-10_10	yearSin2	
		T_index-48_Sspline1_-10_10	T_index-72		T_index-48_Sspline2_10_20	yearSin3	
		T_index-48_Sspline2_10_20	T_index-168		T_index-96	yearTrend	
		T_index-48_Sspline2_10_20	T_index-48		T_index-96		
		T_index-96	Wind_Sspline3_10_15		Wind		
		T_index-96			Wind		
		T_index0_Sspline2_10_20			yearCos3		
		Wind			yearCos4		
		Wind_Sspline2_5_10			yearCos4		
		yearCos2			yearSin1		
		yearCos3			yearSin2		
		yearTrend			yearSin3		
					yearTrend		

Tabelle 17: Bestes Parameterset für jede Kundengruppen



Ergebnisse der lokalen Suche für jede Kundengruppen:

Links ist die Wahrscheinlichkeit, sich im optimalen Set zu befinden dargestellt.

Rechts lässt sich erkennen, ob es im finalen Set aufgenommen wurde. Es wird ein „Score“ zum Ranking eingeführt, der die Summe der Wahrscheinlichkeiten darstellt.

	Gruppe 1	Gruppe 2	Gruppe 3	Gruppe 4	Gruppe 5	Gruppe 6	Score
Temperatur_index-48_Cspline2_10_20	0,7311 WAHR	0,8581 WAHR	0,7685 WAHR	0,2315 FALSCH	0,6457 WAHR	0,832 WAHR	4,0669
Feiertag	0,69 WAHR	0,8022 WAHR	0,5987 WAHR	0,5 WAHR	0,5987 WAHR	0,7685 WAHR	3,9581
Temperatur_mean0	0,2689 FALSCH	0,69 WAHR	0,8022 WAHR	0,2689 FALSCH	0,7685 WAHR	0,6457 WAHR	3,4442
Temperatur_mean-168	0,7311 WAHR	0,7685 WAHR	0,31 FALSCH	0,168 FALSCH	0,7311 WAHR	0,7311 WAHR	3,4398
yeartrend	0,2689 FALSCH	0,7685 WAHR	0,31 FALSCH	0,8808 WAHR	0,8581 WAHR	0,2315 FALSCH	3,3178
Temperatur_mean-96	0,1978 FALSCH	0,8022 WAHR	0,1978 FALSCH	0,8581 WAHR	0,7685 WAHR	0,4502 FALSCH	3,2746
Temperatur_index-24_Cspline1_-10_10	0,3543 FALSCH	0,8022 WAHR	0,4502 FALSCH	0,5987 WAHR	0,5987 WAHR	0,1192 FALSCH	2,9233
Temperatur_index-168	0,168 FALSCH	0,7311 WAHR	0,8022 WAHR	0,1419 FALSCH	0,7685 WAHR	0,2315 FALSCH	2,8432
Temperatur_mean0_Cspline1_-10_10	0,8581 WAHR	0,3543 FALSCH	0,8581 WAHR	0,1419 FALSCH	0,31 FALSCH	0,31 FALSCH	2,8324
Temperatur_mean-72	0,6457 WAHR	0,2689 FALSCH	0,7685 WAHR	0,5498 WAHR	0,3543 FALSCH	0,2315 FALSCH	2,8187
Temperatur_index-24	0,168 FALSCH	0,8808 WAHR	0,4502 FALSCH	0,5987 WAHR	0,4013 WAHR	0,31 FALSCH	2,809
yearCos3	0,2315 FALSCH	0,8022 WAHR	0,31 FALSCH	0,1419 FALSCH	0,7685 WAHR	0,3543 FALSCH	2,6084
Wind	0,5 FALSCH	0,69 WAHR	0,1978 FALSCH	0,2315 FALSCH	0,69 WAHR	0,2689 FALSCH	2,5782
Temperatur_index-1	0,5 WAHR	0,2315 FALSCH	0,7311 WAHR	0,1419 FALSCH	0,2315 FALSCH	0,7311 WAHR	2,5671
Temperatur_index-48_Cspline1_-10_10	0,31 FALSCH	0,8022 WAHR	0,168 FALSCH	0,2689 FALSCH	0,8022 WAHR	0,168 FALSCH	2,5193
T_square	0,69 FALSCH	0,1192 FALSCH	0,6457 FALSCH	0,1419 FALSCH	0,6457 WAHR	0,2315 FALSCH	2,474
yearsIn1	0,2315 FALSCH	0,1419 FALSCH	0,168 FALSCH	0,2689 FALSCH	0,8581 WAHR	0,8022 WAHR	2,4706
Luftfeuchte_Cspline3_67_100	0,1419 FALSCH	0,69 WAHR	0,5498 FALSCH	0,31 WAHR	0,5 WAHR	0,2689 FALSCH	2,4606
Temperatur	0,5987 WAHR	0,7685 WAHR	0,1419 FALSCH	0,31 FALSCH	0,3543 FALSCH	0,2689 FALSCH	2,4423
Temperatur_mean0_Cspline2_10_20	0,1419 FALSCH	0,7311 WAHR	0,31 FALSCH	0,31 FALSCH	0,168 FALSCH	0,7685 WAHR	2,4295
Luftfeuchte_Cspline2_33_67	0,69 WAHR	0,7685 WAHR	0,2689 FALSCH	0,31 FALSCH	0,2315 FALSCH	0,1419 FALSCH	2,4108
yearSin2	0,1419 FALSCH	0,1978 FALSCH	0,2315 FALSCH	0,168 FALSCH	0,8022 WAHR	0,8022 WAHR	2,3436
yearCos2	0,1192 FALSCH	0,7685 WAHR	0,168 FALSCH	0,8581 WAHR	0,1978 FALSCH	0,2315 FALSCH	2,3431
Temperatur_index-2_Cspline1_10_20	0,8022 WAHR	0,1978 FALSCH	0,7685 FALSCH	0,2315 FALSCH	0,0998 FALSCH	0,2315 FALSCH	2,3313
Enthalpy	0,1978 FALSCH	0,8581 WAHR	0,7685 WAHR	0,1978 FALSCH	0,1419 FALSCH	0,1192 FALSCH	2,2833
Enthalpy_Cspline1_-10_20	0,2689 FALSCH	0,2689 FALSCH	0,69 WAHR	0,1192 FALSCH	0,168 FALSCH	0,6457 WAHR	2,1607
Luftfeuchte_Cspline1_0_33	0,31 FALSCH	0,832 WAHR	0,2315 FALSCH	0,2315 FALSCH	0,3543 FALSCH	0,1978 FALSCH	2,1571
Enthalpy_Cspline2_20_40	0,8022 WAHR	0,0998 FALSCH	0,69 WAHR	0,168 FALSCH	0,1978 FALSCH	0,1978 FALSCH	2,1556
Temperatur_index-48	0,1978 FALSCH	0,2315 FALSCH	0,69 WAHR	0,168 FALSCH	0,5987 WAHR	0,2689 FALSCH	2,1549
T_relu	0,31 FALSCH	0,2315 FALSCH	0,7311 WAHR	0,1978 FALSCH	0,2689 FALSCH	0,2315 FALSCH	1,9708
Temperatur_index-1_Cspline1_10_20	0,0998 FALSCH	0,2315 FALSCH	0,168 WAHR	0,832 WAHR	0,3543 FALSCH	0,1978 FALSCH	1,8834
Luftfeuchte	0,2689 FALSCH	0,1978 FALSCH	0,4013 FALSCH	0,168 FALSCH	0,69 WAHR	0,1419 FALSCH	1,8679
Temperatur_mean-48	0,2689 FALSCH	0,7311 WAHR	0,2689 FALSCH	0,168 FALSCH	0,2315 FALSCH	0,168 FALSCH	1,8364
Temperatur_index-4_Cspline1_-10_10	0,1419 FALSCH	0,4013 FALSCH	0,2689 FALSCH	0,2315 FALSCH	0,5987 WAHR	0,168 FALSCH	1,8103
yearSin3	0,2315 FALSCH	0,1978 FALSCH	0,168 FALSCH	0,1978 FALSCH	0,8022 WAHR	0,168 FALSCH	1,7653
Temperatur_index-24_Cspline2_10_20	0,2315 FALSCH	0,1192 FALSCH	0,2689 FALSCH	0,1419 FALSCH	0,7685 WAHR	0,2315 FALSCH	1,7615
Temperatur_Cspline2_10_20	0,1419 FALSCH	0,7685 WAHR	0,2689 FALSCH	0,168 FALSCH	0,1419 FALSCH	0,2689 FALSCH	1,7581
Temperatur_Cspline3_20_30	0,2689 FALSCH	0,1978 FALSCH	0,69 WAHR	0,2315 FALSCH	0,1978 FALSCH	0,168 FALSCH	1,754
Temperatur_index-4	0,2315 FALSCH	0,6457 WAHR	0,2689 FALSCH	0,2315 FALSCH	0,1978 FALSCH	0,168 FALSCH	1,7434
yearCos4	0,1978 FALSCH	0,2315 FALSCH	0,2689 FALSCH	0,0998 FALSCH	0,69 WAHR	0,2315 FALSCH	1,7195
Wind_Cspline2_5_10	0,1978 FALSCH	0,8022 WAHR	0,168 FALSCH	0,2689 FALSCH	0,1192 FALSCH	0,1419 FALSCH	1,698
Sonne_Cspline1_0_0	0,31 FALSCH	0,31 FALSCH	0,5 FALSCH	0,1978 FALSCH	0,168 FALSCH	0,168 FALSCH	1,6538
Temperatur_mean-24	0,5 FALSCH	0,2315 FALSCH	0,31 FALSCH	0,168 FALSCH	0,1978 FALSCH	0,1419 FALSCH	1,5492
Sonne	0,2689 FALSCH	0,2689 FALSCH	0,5 FALSCH	0,168 FALSCH	0,168 FALSCH	0,168 FALSCH	1,5418
yearSin4	0,168 FALSCH	0,168 FALSCH	0,1978 FALSCH	0,2315 FALSCH	0,2689 FALSCH	0,5 FALSCH	1,5342
Sonne_Cspline3_0_1	0,5498 WAHR	0,1978 FALSCH	0,2315 FALSCH	0,1978 FALSCH	0,168 FALSCH	0,168 FALSCH	1,5129
Niederschlag	0,168 FALSCH	0,168 FALSCH	0,4502 FALSCH	0,1978 FALSCH	0,2689 FALSCH	0,1978 FALSCH	1,4507
Wind_Cspline3_10_15	0,1978 FALSCH	0,2315 FALSCH	0,5 WAHR	0,1192 FALSCH	0,1419 FALSCH	0,2315 FALSCH	1,4219
Temperatur_index-2	0,2689 FALSCH	0,2315 FALSCH	0,4013 FALSCH	0,1978 FALSCH	0,168 FALSCH	0,1192 FALSCH	1,3867
Temperatur_index-4_Cspline1_10_20	0,2689 FALSCH	0,2689 FALSCH	0,1978 FALSCH	0,3543 FALSCH	0,1192 FALSCH	0,1192 FALSCH	1,3283
Enthalpy_Cspline3_40_60	0,2315 FALSCH	0,2315 FALSCH	0,2315 FALSCH	0,1419 FALSCH	0,168 FALSCH	0,31 FALSCH	1,3144
Temperatur_index-3	0,2689 FALSCH	0,168 FALSCH	0,2689 FALSCH	0,168 FALSCH	0,1419 FALSCH	0,2689 FALSCH	1,2846
yearCos1	0,1978 FALSCH	0,31 FALSCH	0,168 FALSCH	0,31 FALSCH	0,1419 FALSCH	0,1419 FALSCH	1,2696
Wind_Cspline1_0_5	0,2315 FALSCH	0,168 FALSCH	0,2689 FALSCH	0,1419 FALSCH	0,1978 FALSCH	0,2315 FALSCH	1,2396
Global	0,168 FALSCH	0,168 FALSCH	0,2315 FALSCH	0,2315 FALSCH	0,2689 FALSCH	0,168 FALSCH	1,2359
Global_Cspline3_667_1000	0,1192 FALSCH	0,1419 FALSCH	0,1978 FALSCH	0,31 FALSCH	0,2315 FALSCH	0,1978 FALSCH	1,1982
Temperatur_index-72	0,1978 FALSCH	0,2689 FALSCH	0,168 FALSCH	0,1419 FALSCH	0,168 FALSCH	0,2315 FALSCH	1,1761
Global_Cspline1_0_333	0,1419 FALSCH	0,2315 FALSCH	0,2315 FALSCH	0,1978 FALSCH	0,2315 FALSCH	0,1419 FALSCH	1,1761
Temperatur_index-3_Cspline1_10_20	0,0998 FALSCH	0,2315 FALSCH	0,2315 FALSCH	0,168 FALSCH	0,2315 FALSCH	0,1978 FALSCH	1,1601
Sonne_Cspline2_0_0	0,0998 FALSCH	0,0998 FALSCH	0,2689 FALSCH	0,1978 FALSCH	0,31 FALSCH	0,168 FALSCH	1,1443
Temperatur_mean0_Cspline3_20_30	0,168 FALSCH	0,2315 FALSCH	0,2689 FALSCH	0,1978 FALSCH	0,1192 FALSCH	0,1419 FALSCH	1,1273
Temperatur_index-24_Cspline3_20_30	0,1419 FALSCH	0,1192 FALSCH	0,168 FALSCH	0,2315 FALSCH	0,2315 FALSCH	0,2315 FALSCH	1,1236
Temperatur_index-48_Cspline3_20_30	0,1419 FALSCH	0,168 FALSCH	0,1978 FALSCH	0,1419 FALSCH	0,2315 FALSCH	0,168 FALSCH	1,0491
Global_Cspline2_333_667	0,1419 FALSCH	0,168 FALSCH	0,168 FALSCH	0,1419 FALSCH	0,2315 FALSCH	0,1419 FALSCH	0,9932

Tabelle 18: Ergebnisse der Lokalen Suche für jede Kundengruppen

Für neuronale Netze:

Model	Zeitfaktor Netz	Zeitfaktor Netz
<b>Mit Historische Last mit Luftfeuchte Mit S-Splines</b>	Ja Nein Nein	Ja Nein Nein
<b>Methode</b>	genetischer Algorithmus	genetischer Algorithmus + Lokal
<b>Fehler (relative)</b>	22,83%	22,26%
<b>Zeit</b>	29:05:50	03:56:52
	Feiertag Global_spline1 Global_spline4 Niederschlag Sonne_spline2 Sonne_spline3 Sonne_spline4 T_relu T_square Temperatur Temperatur_index-168 Temperatur_index-48 Temperatur_index-48_spline2 Temperatur_index-48_spline3 Temperatur_index-48_spline4 Temperatur_index-72 Temperatur_spline2_2 Temperatur_spline2_3 Temperatur_spline2_4 Temperatur_spline3_5 dayCos1 dayCos2 daySin1 hour reference_copy_index-48 reference_copy_index-48_spline2 reference_copy_mean-168 reference_copy_mean-48 reference_copy_mean-72 weekday	Feiertag Global_spline1 Global_spline4 Niederschlag Sonne_spline2 Sonne_spline3 Sonne_spline4 T_relu T_square Temperatur Temperatur_index-168 Temperatur_index-48 Temperatur_index-48_spline2 Temperatur_index-48_spline3 Temperatur_index-48_spline4 Temperatur_index-72 Temperatur_spline2_2 Temperatur_spline2_3 Temperatur_spline2_4 Temperatur_spline3_5 Wind_spline2 dayCos1 dayCos2 dayCos3 daySin1 hour reference_copy_index-48 reference_copy_index-48_spline2 reference_copy_mean-168 reference_copy_mean-48 reference_copy_mean-72 weekday

Tabelle 19: Ergebnisse der Parameterauswahl Algorithmen für das Netz mit Zeitfaktoren

### III. Parametersensitivität

Für die Sensitivitätsanalyse wurden 54 verschiedene Zeitreihe berechnet und analysiert. Im Standardmodell wird ein Regressionsverfahren angewandt, welches zwischen Tagen und Stunden unterscheidet.

In der ersten Methode (standardisierte Koeffizienten) wurde keine Ebene verwendet.

	Korrelation	Standardisierte Koeffizienten	Reihenfolge der Parameter Entfernung	Bayessche Inferenz	Marginale Verbesserung
Temperatur_Cspline2_10_20	0,898156	1,8685		0,7109	11,52%
reference_copy_index-48	0,813546	0,0507		0,6682	0,57%
Feiertag	0,07593	0,0417		0,69	0,55%
Enthalpy	0,787143	0,7253		0,7311	0,54%
reference_copy_mean-48	0,673384	0,0543		0,69	0,33%
Temperatur_index-48_Cspline2_10_20	0,78649	0,0152		0,7109	0,29%
Temperatur_Cspline1_-10_10	0,830882	5,46		0,5744	0,26%
reference_copy_mean-168	0,653471	0,0911		0,6457	0,18%
Luftfeuchte_Cspline2_33_67	0,466847	0,0539		0,4502	0,09%
Global_Cspline3_667_1000	0,279366	0,004		0,2891	-0,01%
Temperatur	0,792564	3,715		0,5498	-0,01%
Temperatur_mean-72	0,623621	0,013		0,31	-0,03%
reference_copy_index-168	0,806382	0,2818		0,2891	-0,04%
Temperatur_mean-48	0,642277	0,0587		0,2689	-0,04%
yearCos3	0,016787	0,0018		0,31	-0,04%
yearSin2	0,147097	0,0633		0,3318	-0,04%
Temperatur_index-72	0,714354	0,0084		0,31	-0,05%
Global	0,521967	0,0402		0,31	-0,06%
Temperatur_index-48	0,735326	0,0001		0,5	-0,06%
Wind_Cspline3_10_15	0,015187	0,0015		0,2315	-0,06%
yearCos2	0,145148	0,0532		0,31	-0,06%
Global_Cspline1_0_333	0,489365	0,104		0,2497	-0,07%
Temperatur_index-48_Cspline1_-10_10	0,766528	0,0202		0,4256	-0,07%
Temperatur_mean-96	0,611479	0,0506		0,4013	-0,07%
Niederschlag	0,008888	0,0094		0,2315	-0,08%
Luftfeuchte_Cspline3_67_100	0,279808	0,0229		0,2891	-0,09%
Wind	0,010559	0,068		0,31	-0,09%
reference_copy_index-72	0,781156	0,0128		0,2689	-0,10%
Wind_Cspline1_0_5	0,021533	0,0689		0,31	-0,10%
T_relu	0,887263	0,0459		0,2497	-0,11%
Temperatur_index-168	0,681969	0,1329		0,2689	-0,11%
Global_Cspline2_333_667	0,412269	0,0386		0,2891	-0,13%
reference_copy_mean-72	0,643663	0,0166		0,31	-0,13%
T_square	0,895431	3,9275		0,2891	-0,13%
Temperatur_mean-168	0,591402	0,0261		0,2497	-0,13%
Sonne_Cspline2_0_0	0,428452	0,0375		0,3318	-0,14%
Sonne_Cspline3_0_1	0,373956	0,0008		0,2142	-0,14%
Luftfeuchte	0,501611	0,2488		0,3318	-0,15%
Luftfeuchte_Cspline1_0_33	0,50177	0,3468		0,31	-0,15%
Sonne	0,479274	0,131		0,3318	-0,15%
yearSin1	0,188473	0,0573		0,2891	-0,15%
Sonne_Cspline1_0_0	0,465269	0,1533		0,3318	-0,16%
Wind_Cspline2_5_10	0,053601	0,0154		0,2891	-0,17%
reference_copy_mean-96	0,626111	0,0237		0,475	-0,19%

Temperatur_Cspline3_20_30	0,67213	0,3174	0,2497	-0,19%
yearSin4	0,004198	0,0154	0,2315	-0,20%
yearCos4	0,039003	0,0134	0,2497	-0,24%
yearCos1	0,613813	0,1268	0,31	-0,32%
yearSin3	0,073337	0,0113	0,2891	-0,45%
Temperatur_index-48_Cspline3_20_30	0,552849	0,0527	0,31	-0,90%
reference_copy_profile	0,858699	0,1625	0,3318	-0,96%
yearTrend	0,247491	0,0145	0,3318	-1,10%
reference_copy_index-48_spline1	0,271681	0,0969	0,3318	-1,33%
reference_copy_index-48_spline2	0,658914	0,1447	0,3543	-1,37%

**Tabelle 20: Sensitivitätsanalyse - Ergebnisse**