



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Lucas Anders

Deep Learning zur Vorhersage des Energiebedarfs
der antarktischen Forschungsstation Neumayer III

Lucas Anders

Deep Learning zur Vorhersage des Energiebedarfs der
antarktischen Forschungsstation Neumayer III

Bachelorarbeit eingereicht im Rahmen des Studiums

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Kai v. Luck
Zweitgutachter : Prof. Dr. Tim Tiedemann

Abgegeben am 29.10.2018

Lucas Anders

Thema der Arbeit

Deep Learning zur Vorhersage des Energiebedarfs der antarktischen Forschungsstation Neumayer III

Stichworte

Deep-Learning, Machine-Learning, Zeitreihendaten, Sensordaten, Long-Short-Term-Memory-Networks

Kurzzusammenfassung

Die kurzfristige Prognose des Energiebedarfs ist für Energieversorger ein interessanter Anwendungsfall und kritischer Bestandteil eines effizienten Energiemanagementsystems. In dieser Arbeit wird ein intelligentes Vorhersagemodell entwickelt, welches den Gesamtenergiebedarf eines abgelegenen Forschungsgebäudes auf der Basis von realen und multidimensionalen Sensordaten prognostizieren kann. Die verwendeten Daten entstammen der Polarforschungsstation Neumayer III. Dabei werden verschiedene zeitreihenbasierte Deep-Learning-Verfahren eingesetzt und evaluiert.

Lucas Anders

Title of the paper

Deep Learning for energy consumption prediction of the antarctic research facility Neumayer III

Keywords

Deep learning, Machine Learning, Time-Series Data, Sensor Data, Long-Short-Term-Memory-Networks

Abstract

Predicting the short-term energy consumption is an interesting use case for energy providers and crucial for an efficient energy management system. This thesis implements an intelligent model which predicts short-term energy demands based on real and multivariate sensor data. The dataset is taken from the antarctic research facility Neumayer III. Different deep learning models for time series prediction are applied and evaluated.

Danksagung

An dieser Stelle möchte ich mich zuerst bei Prof. Kai v. Luck und Prof. Tim Tiedemann für die Betreuung dieser Arbeit und die hilfreichen Anregungen bedanken. Besonderer Dank gilt außerdem Paul Assendorp für die umfangreiche Unterstützung und konstruktive Kritik. Weiterhin danke ich dem Alfred-Wegener-Institut für die Bereitstellung eines sehr interessanten Anwendungsfalls und des Datensatzes. Zudem möchte ich mich bei der Firma Werum Software & Systems und damit vor allem Hendrik Bohlen und Uwe Zenker für das Ermöglichen dieser Arbeit bedanken.

Natürlich möchte ich mich abschließend auch herzlich bei meinen Freunden und meiner Familie bedanken, die mich während des gesamten Studiums begleitet und unterstützt haben.

Inhaltsverzeichnis

1	Einleitung	1
2	Analyse	3
2.1	Problemstellung	3
2.2	Machine Learning.....	5
2.3	Deep Learning	5
2.3.1	Deep Learning in Zeitreihendaten.....	6
2.4	Entwicklung von intelligenten Systemen	7
2.4.1	Datenselektion	9
2.4.2	Datenbereinigung.....	9
2.4.3	Feature Selection	11
2.4.4	Datentransformation	12
2.4.5	Data Mining.....	13
2.4.6	Evaluation.....	13
2.5	Zielsetzung	14
3	Praktische Durchführung	16
3.1	Dataset	16
3.2	Datenselektion	18
3.3	Data Cleaning	19
3.4	Feature Selection	21
3.5	Datentransformation	23
3.6	Data Mining.....	24
3.6.1	DNN-Modell	25
3.6.2	LSTM-Modell	27
3.7	Versuchsaufbau.....	28
4	Auswertung.....	31

4.1	Entwicklung des Vorhersagemodells	31
4.2	Herausforderungen bei der Entwicklung intelligenter Vorhersagemodelle	34
4.3	Bedeutung fachbezogener Intuition in der Entwicklung von DL-Systemen	36
4.4	Bewertung des Vorhersagemodells	37
5	Fazit	39
5.1	Ausblick	40
	Literaturverzeichnis	41

1 Einleitung

In modernen Big-Data-Systemen werden häufig Prognosesysteme eingesetzt, um neue Erkenntnisse zur Optimierung vorhandener Geschäftsprozesse zu erlangen. Ein solches Vorgehen ist auch in der Energieversorgung zu beobachten. Netzbetreiber nutzen Vorhersagesysteme, um den erwarteten Energiebedarf zu berechnen und somit das Gleichgewicht zwischen eingespeister und verbrauchter Energie gewährleisten zu können. Diese Methodik lässt sich ebenfalls auf kleine, abgeschlossene Systeme übertragen, in denen Energie selbstständig erzeugt werden muss. Im Zuge der Digitalisierung werden Gebäude und Infrastruktur zunehmend mit Sensoren ausgestattet, welche Werte sowie Statusmeldungen zu den verbauten Geräten erfassen. Deren Daten können ausgewertet werden, um neue Erkenntnisse zu erlangen und vorhandene Arbeitsprozesse zu unterstützen.

Außerdem ist durch neue Entwicklungen in den vergangenen Jahren die Speicherung und Verarbeitung von multidimensionalen, sensorbasierten Datenbeständen ermöglicht worden. Dennoch stellt die Entwicklung von zuverlässigen Vorhersagesystemen auf der Grundlage solcher Datenbestände eine Herausforderung dar. In der Praxis werden dazu häufig regelbasierte Expertensysteme eingesetzt, welche jedoch aufwändig zu konstruieren und zu pflegen sind.

Der Einsatz von Machine Learning hingegen erlaubt es, Muster und Gesetzmäßigkeiten aus den Datenbeständen selbstständig zu extrahieren und diese auf diverse Probleme, z.B. eine Energieprognose anzuwenden. Die Entwicklung der zugrundeliegenden Algorithmen hat sich mittlerweile zu einem wichtigen Forschungsgebiet entwickelt, welches regelmäßig neue Erfolge verzeichnet und wertvolle Erkenntnisse hervorbringt. In verwandten Arbeiten konnte bereits gezeigt werden, dass der Einsatz von Deep-Learning-Algorithmen die zuverlässige Prognose eines Energiebedarfs ermöglicht [1]–[3]. Dabei wurden die aufgezeichneten Daten meist unter Berücksichtigung temporaler Abhängigkeiten analysiert. Gegenstand dieser Arbeit ist die Entwicklung eines Deep-Learning-Verfahrens zur Prognose eines Energieverbrauchs auf der Basis von hochdimensionalen und zeitreihenbasierten Sensordaten. Zu diesem Zweck sollen Long-Short-Term-Memory-Networks eingesetzt werden. Die Grundlage dafür bilden reale Daten eines Forschungsgebäudes in der Antarktis,

der *Neumayer Station III*. Weiterhin soll evaluiert werden, ob der vorliegende Datenbestand für die Entwicklung eines Deep-Learning-basierten Prognosemodells geeignet ist. In diesem Kontext wird außerdem der Einfluss domänenbezogener Annahmen auf die Genauigkeit eines solchen Modells untersucht.

Diese Arbeit ist in fünf Kapitel unterteilt. In Kapitel 2 erfolgt zunächst eine Beschreibung des vorliegenden Problems. Weiterhin wird ein Einblick in die Grundlagen der Verarbeitung und Auswertung von Zeitreihendaten gegeben. Anschließend werden verschiedene Machine-Learning-Ansätze aus ähnlichen Arbeiten diskutiert und die Ziele dieser Arbeit genannt.

Kapitel 3 widmet sich der praktischen Umsetzung der genannten Techniken und Algorithmen. Dabei wird besonders auf die Herausforderungen des vorliegenden Anwendungsfalls und die daraus resultierenden Entscheidungen eingegangen.

In Kapitel 4 werden die Ergebnisse und Beobachtungen aus den zuvor beschriebenen Experimenten ausgewertet. Weiterhin werden Probleme genannt, welche in dieser Arbeit identifiziert und beseitigt worden sind. Schließlich wird die Relevanz von menschlichen Erfahrungswerten im Kontext von Deep Learning evaluiert.

Kapitel 5 schließt die Arbeit mit einem Fazit ab. Zusätzlich wird ein Ausblick zu möglichen Maßnahmen gegeben, welche im Anschluss an diese Arbeit ergriffen werden könnten.

2 Analyse

Algorithmen des maschinellen Lernens sind in der Vergangenheit bereits häufig in multidimensionalen und zeitreihenbasierten Datenbeständen eingesetzt worden. Das folgende Kapitel beschäftigt sich mit der Analyse und Bewertung geeigneter Verfahren. Dazu werden verwandte Arbeiten herangezogen, in denen ähnliche Probleme erfolgreich behandelt worden sind.

Der Teilabschnitt 2.1 enthält eine detaillierte Beschreibung der Problemstellung, welche in dieser Thesis gelöst werden soll. Anschließend erfolgt in Kapitel 2.2 und 2.3 eine Einführung in den aktuellen Forschungsstand von Machine Learning und Deep Learning. Der Entwicklungsprozess eines intelligenten Vorhersagemodells wird in Kapitel 2.4 beschrieben. Abschließend wird in Kapitel 2.5 das Ziel dieser Arbeit präzisiert.

2.1 Problemstellung

Die *Neumayer Station III* ist eine deutsche Polarforschungsstation des *Alfred-Wegener-Instituts* (AWI). Das Gebäude befindet sich in der Atka-Bucht auf dem Ekström-Schelfeis der Antarktis und wird ganzjährig bewohnt. Seit der Stationseröffnung im Jahr 2009 zeichnet die *Neumayer III* Sensordaten zu diversen Geräten auf. Diese Sensordaten werden von wissenschaftlichen Messinstrumenten sowie von betriebstechnisch relevanten Geräten, wie etwa den Generatoren, erfasst. Die Daten der *Neumayer III* dienen den Wissenschaftlern als Grundlage für den Forschungsbetrieb, welcher unter anderem die Fachbereiche Klimaforschung, Geophysik, Glaziologie und Luftchemie umfasst.

Die Energieversorgung der *Neumayer III* erfolgt über vier Blockheizkraftwerke (BHKW), von denen eines für Notfälle zurückgehalten wird. Um die BHKWs und den Fuhrpark der Forschungsstation zu betreiben, wird die *Neumayer III* von dem Polarforschungsschiff *Polarstern* einmal jährlich mit Treibstoff beliefert. Um diese Lieferungen effizient zu planen, ist ein gutes Energiemanagementsystem notwendig. Zu diesem Zweck sollen in dieser Arbeit die verfügbaren Sensordaten der *Neumayer III* genutzt werden, um ein intelligentes Vorhersagemodell für den Gesamtenergiebedarf des Stationsgebäudes zu konstruieren. Dieses Modell soll in der Lage sein, relevante Zusammenhänge mit Bezug auf den Energiebedarf aus den vorliegenden Daten zu extrahieren. Die gewonnenen Erkenntnisse können anschließend auf neue Beobachtungen angewendet werden, um eine kurzfristige Prognose zu erzeugen.

Aufgrund der Komplexität der Problemstellung sowie der Verfügbarkeit von hochauflösenden Sensordaten ist die Implementation eines regelbasierten

Expertensystems für die Vorhersage des Energiebedarfs enorm aufwändig und somit auch teuer. Allerdings eignet sich der vorliegende Anwendungsfall für die Anwendung eines maschinell lernenden Systems, welche für die Verarbeitung großer Datenbestände optimiert sind und außerdem keine bzw. nur wenige Kenntnisse über die Domäne des Gebäudemanagements erfordern. Voraussetzung für den erfolgreichen Einsatz eines solchen Systems ist jedoch, dass tatsächlich ein Zusammenhang zwischen den aufgezeichneten Sensordaten und dem Energieverbrauch des Stationsgebäudes besteht. Diese These wird in Kapitel 3 dieser Arbeit untersucht. Weiterhin sind maschinenlernende Systeme nicht in der Lage, unstetige Funktionen abzubilden. Sollte der vorliegende Datenbestand einschneidende Ereignisse beinhalten, leidet daher die Qualität des Vorhersagemodells. Bisher existiert für den Energiebedarf der *Neumayer III* noch kein Prognosemodell, welches zur Evaluation der entwickelten Modelle herangezogen werden könnte. Die Beurteilung der Ergebnisse erfolgt daher durch die zuständigen Experten des *Alfred-Wegener-Instituts*.

Die Sensoren der *Neumayer III* zeichnen in regelmäßigen Intervallen Daten auf. Da anzunehmen ist, dass der aktuelle Energiebedarf der Forschungsstation durch vorherige Ereignisse beeinflusst wird, handelt es sich bei diesem Anwendungsfall um ein Zeitreihenproblem. Zeitreihenprobleme zeichnen sich dadurch aus, dass einzelne Datenpunkte semantisch verknüpft sind und eine genaue Vorhersage dementsprechend die Konstruktion eines zeitlichen Kontexts erfordert. Die Herausforderungen, die daraus für die Entwicklung eines Vorhersagemodells entstehen, werden in den folgenden Kapiteln besonders hervorgehoben.

Abbildung 1 zeigt den Gesamtenergiebedarf der *Neumayer III* beispielhaft für das Jahr 2014. Das entwickelte Vorhersagemodell soll in der Lage sein, diesen Wert angemessen zu prognostizieren, um das Energiemanagement der Forschungsstation zu unterstützen. Die Entwicklung eines solchen Modells für eine reale Problemstellung stellt einige Herausforderungen dar, die in den folgenden Teilabschnitten diskutiert werden.

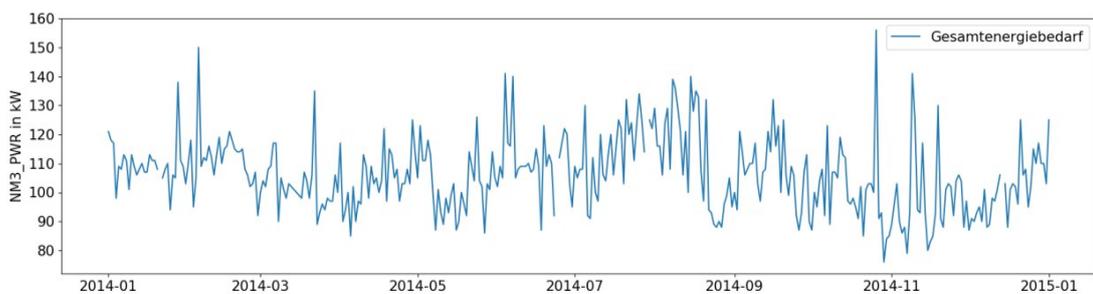


Abbildung 1: Gesamtenergiebedarf der Neumayer Station III im Jahr 2014

2.2 Machine Learning

Machine Learning (ML) ist ein Teilgebiet der künstlichen Intelligenz und beschäftigt sich mit der maschinell unterstützten Generierung von Wissen aus Daten sowie der Anwendung dieses Wissens zur Lösung von datenbasierten Problemen. In der Wissenschaft wird der Begriff ML häufig synonym zum Erkenntnisgewinn aus Datenbeständen verwendet. In dieser Thesis wird ML jedoch als Begriff für eine Gruppe von Algorithmen genutzt. Diese Algorithmen basieren meist auf einer statistischen Auswertung der Datenbestände und sind in der Lage, komplexe Zusammenhänge selbstständig zu erlernen. Darin unterscheiden sie sich grundlegend von herkömmlichen Verfahren der Datenverarbeitung, welche explizit für ein spezifisches Problem angepasst werden müssen.

Maschinenlernende Modelle eignen sich besonders, um auf einer großen Menge an Daten anhand einer spezifischen Problemstellung trainiert zu werden. Die erfassten Zusammenhänge können anschließend auf neue Beobachtungen aus der zuvor behandelten Domäne angewandt werden, um Vorhersagen zu treffen. Die Vorhersagen eines maschinenlernenden Systems sind jedoch, in Abhängigkeit des gewählten Algorithmus, möglicherweise nicht nachvollziehbar.

Dieser Umstand birgt Risiken für die Integration eines ML-Verfahrens in produktive Systeme. Sollte sich die Datenverteilung des betrachteten Problems gravierend verändern, leidet darunter die Qualität der Modellvorhersagen. Hängen kritische Entscheidungen von einer solchen Vorhersage ab, kann dies zu Fehlern im System führen. Dennoch sind maschinenlernende Algorithmen in den letzten Jahren in viele Bereiche des alltäglichen Lebens eingebunden worden. In kritischen Produktiv-Umgebungen werden die entwickelten ML-Systeme aufgrund der zuvor genannten Problematik häufig unterstützend eingesetzt.

Ein weiterer Kritikpunkt an maschinell lernenden Verfahren ist die Verdrängung von menschlicher Intuition durch statistische Werkzeuge. Moderne Algorithmen sind in der Lage, aus riesigen Datenmengen wesentliche Informationen zu extrahieren. In vielen Arbeiten ist bereits zu beobachten, dass die entwickelten Systeme die Fähigkeiten menschlicher Arbeitskräfte übersteigen. Eine Erkenntnis dieser Arbeit soll daher sein, wie stark maschinenlernende Systeme von der Intuition menschlicher Experten profitieren bzw. auf diese angewiesen sind. Dazu werden unterschiedliche Versuche durchgeführt, deren Aufbau in Teilen durch die Einbindung von domänenspezifischem Vorwissen beeinflusst wird.

2.3 Deep Learning

Deep Learning (DL) ist eine Teildisziplin des ML. Dieser Begriff beschreibt den Einsatz von künstlichen neuronalen Netzen (KNN) zur Lösung datenbasierter Probleme. KNN sind mehrschichtige, verknüpfte Perzeptrone [4]. Deren Funktionsweise stellt eine enorm vereinfachte Abstraktion der namensgebenden Nervenzellen dar. KNN sind in der Lage,

eine kompakte Darstellung von Datenpunkten selbstständig zu abstrahieren und diese auf komplexe Probleme anzuwenden. Dieser Prozess wird im DL-Kontext als Training bzw. Lernen bezeichnet. Weiterhin können neuronale Netze nicht-lineare Zusammenhänge erlernen und somit, bei angemessener Konfiguration, jede beliebige Funktion abbilden. Dadurch sind KNN einfachen Verfahren des maschinellen Lernens, welche der aufwändigen Konstruktion von Merkmalsvektoren bedürfen, überlegen. DL-Verfahren eignen sich daher besonders für komplexe Probleme, wie Bild- und Spracherkennung [5].

In einem KNN ist jeder Verbindung zwischen einzelnen Neuronen eine Gewichtung und ein bias zugeordnet. Diese bilden jeweils eine Matrix aus variablen Parametern, welche angepasst werden können, um eine abstrakte Darstellung eines spezifischen Anwendungsfalls zu erlernen. Abhängig von der Tiefe des Netzes sowie der Anzahl der darin enthaltenen Neuronen können diese Matrizen mehrere Millionen modifizierbare Parameter beinhalten.

Für die Konfiguration der Parameter in KNN existieren drei Lernmethoden, das überwachte Lernen (engl. supervised learning), das unüberwachte Lernen (engl. unsupervised learning) und das verstärkende Lernen (engl. reinforcement learning). Die Modelle in dieser Arbeit werden auf der Grundlage des überwachten Lernens trainiert, da in diesem Fall die erwarteten Prognosen zum Zeitpunkt des Trainings bereits bekannt sind. Die Korrektur der Netzparameter erfolgt basierend auf einer Fehlerfunktion, welche die Distanz zwischen der Vorhersage des Modells und dem erwarteten Wert abbildet.

Die Optimierung der Parameter eines neuronalen Netzes erfolgt anschließend anhand des Backpropagation-Algorithmus [6]. Der berechnete Fehler wird dabei als Fehlersignal rückwärts durch alle Schichten des KNN propagiert. Währenddessen werden die Gewichte des neuronalen Netzes angepasst, sodass das Fehlersignal gegen null konvergiert.

Da die Berechnung eines absoluten Minimums in dem hochgradig mehrdimensionalen Parameterraum des Modells jedoch mit einem enorm hohen Rechenaufwand verbunden ist, wird beim Training von neuronalen Netzen das Gradientenverfahren eingesetzt. Der Gradient beschreibt die Richtung sowie Stärke, nach der die Parameter des KNN angepasst werden, um den Fehler zu optimieren. Die Optimierung der Gewichte erfolgt iterativ in kleinen Schritten, bis der Gradient ein lokales bzw. globales Minimum erreicht und der Lernerfolg stagniert.

Weiterhin wird das Training eines neuronalen Netzes durch Hyperparameter, wie etwa die Lernrate, beeinflusst. Diese Hyperparameter werden vom Anwender bestimmt und sind während des Trainings konstant. Die Wahl von geeigneten Hyperparametern ermöglicht es, ein neuronales Netz noch besser an einen spezifischen Anwendungsfall anzupassen.

2.3.1 Deep Learning in Zeitreihendaten

Recurrent Neural Networks (RNN) stellen eine komplexe Architektur von KNN dar, welche sich besonders gut für die Verarbeitung sequenzieller Daten eignet. Rekurrente Netze können Informationen aus vorherigen Beobachtungen in einem internen Speicher ablegen

und daraus während des Trainings Informationen abrufen. Dies ermöglicht es dem Netzwerk, Zeitreihendaten effizient zu verarbeiten und in diesen zeitliche Zusammenhänge zu erkennen. Beliebte Anwendungsfälle von RNN sind Natural Language Processing und Time Series Prediction.

Durch die zusätzliche Komplexität bei der Verarbeitung von sequenziellen Daten enthalten RNN ggü. herkömmlichen Netzen sehr viele Parameter, welche im Gradientenverfahren optimiert werden müssen. Dies fügt dem Trainingsprozess von RNN eine weitere Komplexitätsstufe hinzu. Besonders in tiefen RNN oder bei der Verwendung von vielen Zeitschritten ist das Gradientenverfahren daher sehr anfällig [7], [8]. Wenn der Gradient in einem KNN zu klein (engl. vanishing) bzw. zu groß (engl. exploding) wird, wird das Training des KNN instabil und der Lernerfolg bleibt aus. Um diesem Problem entgegenzutreten, wurden von Hochreiter und Schmidhuber spezielle RNN entwickelt, die Long-Short-Term-Memory-Networks [9]. Diese führen zusätzliche Gatter in die Architektur rekurrenter Schichten ein, welche das Netzwerk weniger anfällig ggü. extremen Gradienten gestalten. Deshalb sind LSTM in der Lage, deutlich längere Sequenzen, bis zu einer Größe von 1000 Zeitschritten, effektiv zu verarbeiten [10]. Eine alternative Architektur zu den LSTM stellen Gated Recurrent Units (GRU) dar [11]. Diese sind ebenfalls in der Lage, Zusammenhänge über viele Zeitschritte zu erkennen und zugleich weniger komplex als LSTM, resultierend in einer kürzeren Trainingsdauer.

2.4 Entwicklung von intelligenten Systemen

In den vergangenen Jahrzehnten hat sich der Umgang mit Daten grundlegend geändert. Dies ist vor allem in der Aufzeichnung von Informationen zu beobachten. Durch die zunehmende Digitalisierung und den Vormarsch der Industrie 4.0 werden immer mehr Daten mit zunehmender Frequenz und Dimension aufgezeichnet, sodass der gesamte Datenumfang enorm angestiegen ist. Dieses Phänomen wird begleitet von der Entwicklung neuer und performanterer Hardware, welche die Speicherung und Verarbeitung von riesigen Datenmengen erlaubt. Die Verfügbarkeit von Cloud-Systemen ermöglicht zudem einen einfacheren Zugriff auf diese Datenbanken.

Die Aufzeichnung großer Datenmengen birgt allerdings auch große Herausforderungen für die Auswertung eben solcher. Daher existieren einige Prozesse, welche das allgemeine Vorgehen bei der Entwicklung von intelligenten Systemen beschreiben. In Fayyad et al. wird ein allgemeingültiger Prozess beschrieben, dessen Ziel der Erkenntnisgewinn aus einer Datenmenge ist [12].

„KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data“

[12]

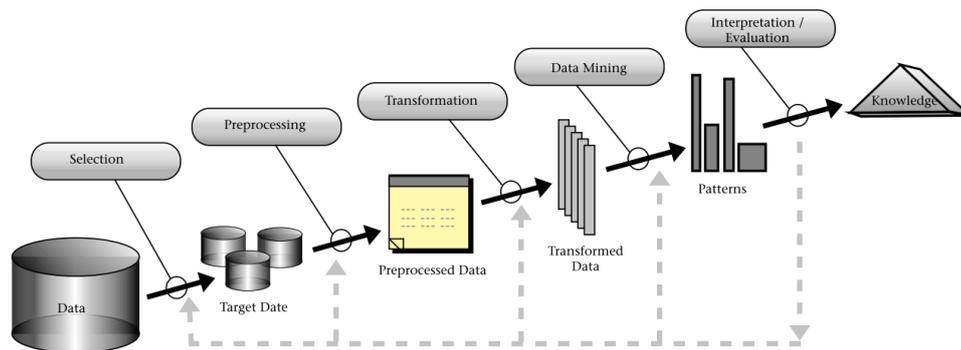


Abbildung 2: KDD-Prozess nach Fayyad. Abbildung S. 41 Fayyad et al.(1996)

Der Knowledge Discovery in Databases (KDD) - Prozess sieht insgesamt fünf Phasen vor (siehe Abbildung 2). In der ersten Phase erfolgt die Selektion der Rohdaten aus einem verfügbaren Gesamtdatenbestand. Diese Rohdaten werden anschließend in der zweiten Phase, der Datenvorverarbeitung, von Fehlern bereinigt. In der dritten Phase werden diese bereinigten Daten zu einer geeigneten Darstellung transformiert, bevor sie in der vierten Phase durch ein Data-Mining-Verfahren ausgewertet werden. Schließlich werden die Ergebnisse des Data-Mining-Verfahrens im fünften und letzten Schritt hinsichtlich der Ziele des KDD bewertet.

Dieses Verfahren ist allerdings in Hinblick auf große, manuell gepflegte Datenbanken entwickelt worden und eignet sich daher nur bedingt für die Anwendung in modernen Systemumgebungen. Dennoch orientiert sich das Verfahren, welches in dieser Arbeit zum Einsatz kommt, an den Grundlagen des KDD.

Bei der Entwicklung eines intelligenten, selbstlernenden Systems auf der Grundlage von Daten handelt es sich um einen iterativen und interaktiven Prozess, der maßgeblich durch die Entscheidungen des Anwenders bestimmt wird. In jeder Phase des Prozesses müssen Entscheidungen getroffen werden, welche den Erkenntnisgewinn des Modells beeinflussen. Die Bewertung der Ergebnisse zum Ende einer Iteration bestimmt anschließend darüber, welche Maßnahmen in der nächsten Iteration ergriffen werden.

In den folgenden Teilabschnitten werden die einzelnen Phasen erläutert, welche bei der Entwicklung dieser Arbeit durchlaufen wurden. Dabei werden Probleme beschrieben, die häufig in Verbindung mit realen Datenbeständen auftreten. Dies betrifft insbesondere die Sicherung der Datenqualität. Da sich diese Arbeit mit der Verarbeitung von Zeitreihendaten durch ein DL-Verfahren beschäftigt, erfolgt die Beschreibung der einzelnen Phasen außerdem mit einem Fokus auf Deep Learning und Zeitreihenproblemen.

2.4.1 Datenselektion

Die Datenselektion stellt den ersten Arbeitsschritt des Entwicklungsprozesses dar. Bei der Selektion eines geeigneten Datenbestandes gilt es, den Datenumfang durch eine geeignete Auswahl einzuschränken und gleichzeitig einen hohen Informationsgehalt zu gewährleisten. Zu diesem Zweck wird eine Teilmenge der verfügbaren Datenpunkte ausgewählt, welche anschließend dem Modell zur Verfügung gestellt werden.

Data-Mining-Verfahren basieren häufig auf der Annahme, dass die einzelnen Datenpunkte des zugrunde liegenden Datenbestands voneinander unabhängig und zudem gleichmäßig verteilt sind. Bei der Verwendung von Zeitreihendaten besteht jedoch ein zeitlicher Kontext, sodass die These hier nicht zutrifft. Dieser Umstand fügt der Datenselektion eine weitere Dimension hinzu und erhöht somit die Komplexität des Entscheidungsprozesses. Dabei werden keine Datenpunkte selektiert, sondern zusammenhängende Betrachtungszeiträume. Sollten diese zu groß gewählt werden, könnten trendbedingte Entwicklungen in den Datenpunkten die Vorhersagen eines Modells beeinträchtigen. Sollten die Zeiträume, aus denen die Daten stammen, jedoch zu klein gewählt werden, stehen dem Modell möglicherweise nicht genügend Informationen zur Verfügung. In diesem Fall kann ebenfalls keine genaue Prognose entwickelt werden. Daher ist es erforderlich, einen optimalen Kompromiss bezüglich des Datenumfangs zu finden.

Die Datenselektion beinhaltet außerdem die Aggregation von Informationen aus verschiedenen Datenquellen. Bei der Aggregation von Zeitreihendaten ist es erforderlich, die sequenzielle Natur des Datenbestandes zu erhalten. Datenpunkte aus unterschiedlichen Quellen können nur dann zusammengeführt werden, wenn diese zur gleichen Zeit aufgezeichnet worden sind oder sich auf den gleichen Zeitpunkt beziehen.

Einige Data-Mining-Verfahren neigen dazu, eine Überanpassung (engl. Overfitting) zu entwickeln. Bei einer Überanpassung lernen die Modelle Zusammenhänge in den bekannten Daten auswendig. Dieses Verhalten führt zwar zu sehr guten Ergebnissen auf den bekannten Datenbeständen, überangepasste Modelle generieren jedoch schlechte Prognosen für neue, unbekannte Beobachtungen aus der gleichen Domäne. Um dieses Phänomen frühzeitig zu erkennen und gegebenenfalls geeignete Maßnahmen ergreifen zu können, wird während der Datenselektion zusätzlich ein Testdatensatz ausgewählt. Dieser Testdatenbestand enthält eine Teilmenge aller verfügbaren Datenpunkte und ist dem Modell zum Zeitpunkt des Trainings nicht bekannt. Diese Datenpunkte werden anschließend genutzt, um die Anwendung des Modells auf neue Daten zu simulieren und die Qualität des Modells zu bewerten.

2.4.2 Datenbereinigung

Im Anschluss an die Datenselektion erfolgt die Bereinigung der Daten. Bei der Aufzeichnung von Daten in produktiv eingesetzten Systemen treten in der Regel eine Vielzahl von Fehlern auf, die zu unreinen Datenbeständen führen können. Die Qualität jener Datenbestände beeinflusst den Erkenntnisgewinn wesentlich, daher sollten Unreinheiten in den

selektierten Daten bereits frühzeitig entfernt werden. Bei der Behandlung von unreinen Daten wird zwischen zwei Fällen unterschieden. Unreinheiten entstehen entweder durch unvollständige Datenbestände oder durch eine fehlerhafte Aufzeichnung der Daten.

Unvollständige Datenbestände sind verhältnismäßig leicht zu identifizieren. Da viele gängige ML-Verfahren keine unvollständigen Daten verarbeiten können, müssen diese zwangsläufig behandelt werden. Abhängig vom Anwendungsfall kann es jedoch vorkommen, dass das Fehlen einer Information ebenfalls eine relevante Information darstellt (z.B. der Defekt eines Geräts). In diesem Fall ist es üblich, alle fehlenden Werte durch einen konstanten Wert zu ersetzen, der diese Information abbildet. In vielen Datenbeständen, insbesondere bei der Verwendung von Sensordaten, deuten fehlende Informationen jedoch auf Lücken in der Aufzeichnung hin (z.B. durch unterschiedliche Frequenzen der aufzeichnenden Geräte oder ausgefallene Sensoren). Daher wird häufig versucht, diese Informationen aus den vorhandenen Beobachtungen zu rekonstruieren. Wenn ein Kanal eine Korrelation zu anderen Kanälen aufweist, ist es möglich, die betroffenen Daten aus den Informationen der abhängigen Kanäle abzuleiten. Alternativ können auch die vorhandenen Informationen aus dem fehlerbehafteten Kanal hinzugezogen werden. Bei numerischen Werten ist es üblich, das arithmetische Mittel der übrigen vorliegenden Daten zu errechnen und daraus die fehlenden Werte zu interpolieren. Bei Zeitreihendaten wird dies häufig mit einem Zeitfenster kombiniert, sodass bei der Rekonstruktion ausschließlich Werte berücksichtigt werden, welche in unmittelbarer zeitlicher Nähe zu den fehlenden Informationen aufgezeichnet worden sind.

Die Substitution fehlender Informationen ermöglicht es dem System, verunreinigte Daten während des Trainings eines ML-Algorithmus dennoch zu berücksichtigen. Bei der Rekonstruktion von fehlenden Daten gehen jedoch häufig Informationen verloren oder werden verfälscht. Sollte der Informationsgehalt der rekonstruierten Daten stark von den übrigen Daten abweichen, wird das Ergebnis von ML-Algorithmen dadurch beeinträchtigt. In Grenzfällen sollten die betroffenen Datenpunkte daher nicht im Training berücksichtigt werden.

Fehlerhafte Daten hingegen entstehen durch defekte Sensoren oder Bedienfehler während der Aufzeichnung. Die Detektion von fehlerhaften Daten stellt eine größere Herausforderung dar, weil die beobachteten Daten zunächst in einen fachlichen Kontext gebracht werden müssen. Die Konstruktion eines solchen Kontexts erfordert entsprechendes Wissen über die fachliche Domäne des konkreten Anwendungsfalls. Die Identifikation und Aufbereitung fehlerhafter Daten setzt dementsprechend den Einsatz von Experten voraus und ist verhältnismäßig teuer.

Die betroffenen Datenpunkte sind in vielen Fällen nicht mehr eindeutig rekonstruierbar und können folglich nicht beim Training eines Modells berücksichtigt werden. In wenigen Fällen kann es sinnvoll sein, fehlende Daten als unvollständig zu betrachten und die betroffenen Datenpunkte nach dem oben genannten Verfahren zu behandeln.

2.4.3 Feature Selection

In Teilabschnitt 2.4.1 ist bereits die Selektion geeigneter Datenpunkte beschrieben worden. Bei der Selektion von Trainingsdaten gilt es außerdem, den Informationsgehalt in Bezug auf die Problemstellung durch eine geeignete Auswahl zu maximieren. Dafür können zu den bereits selektierten Datenpunkten einzelne Merkmale (engl. „features“) ausgewählt werden, um den Datenumfang weiter zu reduzieren.

Um eine geeignete Auswahl zu treffen, ist es zunächst erforderlich, das Problem aus der Sicht des Kunden zu definieren. Die Featureselektion wird meist durch fachlich bedingte Entscheidungen beeinflusst, sodass der Anwender des KDD zuerst ein Verständnis des zu lösenden Problems erlangen muss. Abhängig vom vorliegenden Datenbestand kann es sinnvoll sein, keine Featureselektion durchzuführen und stattdessen alle verfügbaren Informationen zu nutzen. Erfahrungswerte weisen allerdings darauf hin, dass die Selektion besonders relevanter Informationen Verbesserungen bezüglich der Genauigkeit und der Laufzeit vieler Algorithmen bietet [13]. Kürzere Entwicklungsiterationen ermöglichen es dem Anwender, Erkenntnisse aus den vorangegangenen Experimenten schneller in den Prozess einzuarbeiten und ggf. Änderungen daran vorzunehmen.

Da die Aufzeichnung von Daten in industriellen Umgebungen häufig nicht auf ein einzelnes spezifisches Problem abgestimmt ist, enthalten diese Datenbestände oft auch Informationen, die in keinem Zusammenhang zu dem spezifizierten Problem stehen. Daten mit einem geringen Informationsgehalt sollten nicht in den selektierten Datenbestand eingehen, um die algorithmische Komplexität möglichst gering zu halten.

Wesentlicher Bestandteil der Datenselektion ist es daher, einen Kompromiss aus Informationsgehalt und Datenumfang bezüglich der Datenmerkmale zu bilden. Je mehr Merkmale bei der Analyse betrachtet werden, desto größer sind die potenziellen Erkenntnisse des Systems, da komplexere Zusammenhänge abgebildet werden können. Allerdings steigt mit zunehmender Anzahl von Merkmalen auch die algorithmische Komplexität und die damit verbundene Laufzeit der meisten Data Mining-Verfahren an. Dies hat zur Folge, dass die eingesetzten Verfahren möglicherweise nicht mehr in der Lage sind, nützliche Erkenntnisse zu erlangen oder mehr Trainingsdaten benötigen.

Einige Verfahren der Merkmalsselektion bilden das statistische Verhalten einzelner Features zueinander ab, um die Relevanz dieser Merkmale zu beurteilen. Anhand der Auswertung können anschließend solche Merkmale ausgewählt werden, die besonders stark mit der gewünschten Vorhersage korrelieren. Einige Beispiele für solche Verfahren sind Random Forest [14] oder der Select-K-Best Algorithmus. Mit diesen Verfahren können zwar keine besonders komplexen Zusammenhänge abgebildet werden, aber die Korrelationsanalyse ist ein guter Indikator für die Relevanz einzelner Merkmale.

Ergänzend zu der automatisierten Selektion einiger Features besteht die Möglichkeit, die Auswahl von Merkmalen auf der Basis von domänenspezifischen Kenntnissen zu treffen. Diese haben keinen Bezug zu Data-Mining-Verfahren, können aber wertvolle Ergänzungen zur fachlichen Domäne der Problemstellung in den Prozess einbringen. Solche Betrachtungen erlauben es dem Anwender Merkmale auszuwählen, die

anwendungsbezogen besonders geeignet erscheinen. Ein solches Verfahren setzt allerdings die Verfügbarkeit von Domänenwissen bzw. von beratenden Experten voraus, welche in einigen Domänen nicht vorhanden sind.

Zusätzlich zur Selektion einiger Merkmale aus den gegebenen Datenbeständen kann es sinnvoll sein, einige Kontextinformationen einzubeziehen. Kontextinformationen sind Beobachtungen, welche in Bezug auf die Problemstellung interessant sind, aber nicht in den Datenbeständen abgebildet sind und oft aus externen Quellen eingebunden werden müssen. Die Ergänzung gut gewählter Kontextinformationen ist häufig mit Verbesserungen in der Genauigkeit eines selbstlernenden Systems verknüpft.

2.4.4 Datentransformation

Im Anschluss an die Datenvorverarbeitung sieht der KDD-Prozess die Transformation der bereinigten Daten vor. Dies beinhaltet die Abbildung der Daten auf eine einheitliche und geeignete Darstellung, sodass die Daten anschließend durch ein Data Mining-Verfahren analysiert werden können.

Eine wesentliche Rolle spielt dabei der Wertebereich der einzelnen Kanäle. Da sich die Messgrößen und der Wertebereich verschiedener Sensoren in realen Anwendungen häufig stark unterscheiden, kann Werten mit einem besonders hohen Wertebereich fälschlicherweise eine höhere Bedeutung angerechnet werden, als den Kanälen mit einem niedrigen Wertebereich. Um dies zu vermeiden, werden die Daten auf einen gemeinsamen Wertebereich fest definierter Länge abgebildet. Dazu wird zunächst die individuelle Verteilung des zu skalierenden Kanals betrachtet. Die einzelnen Beobachtungen werden anschließend so skaliert, dass das Verhältnis der Datenpunkte zueinander erhalten bleibt und keine relevanten Informationen verloren gehen.

Wie bereits in Abschnitt 2.4.1 erwähnt, kann der Umfang des Datenbestands ein Hindernis bei der Entwicklung eines geeigneten Modells darstellen. Sollten beim Training eines Modells Schwierigkeiten auftreten, die auf den Umfang der Trainingsdaten zurückzuführen sind, kann die Dimension der Eingabedaten in diesem Schritt weiter reduziert werden.

Für die Dimensionsreduktion existieren Algorithmen, welche diesen Prozess automatisieren. Besonders verbreitet ist die Verwendung einer Principal-Component-Analysis (PCA) [15]. Durch eine PCA kann die algorithmische Komplexität und somit die Anforderungen an die eingesetzte Hardware sowie die Trainingsdauer erheblich reduziert werden.

Bei der Verwendung von Zeitreihendaten erfolgt in diesem Arbeitsschritt abschließend die Transformation einzelner Datenpunkte in zusammenhängende, sich ggf. überlappende Sequenzen. Zu diesem Zweck wird eine gleitende Fensterfunktion auf die Daten angewandt, die aufeinanderfolgende Datenpunkte zu einem Sample aggregiert. Dabei gilt es, die Länge des Fensters sowie die Schrittweite, in Abhängigkeit des Problems, geeignet zu wählen.

2.4.5 Data Mining

Daraufhin beinhaltet der Prozess der Informationsgewinnung die Extraktion von Informationen aus den zuvor aufbereiteten Daten. Für die Auswertung der Daten bieten sich viele verschiedene Algorithmen an. Die Auswahl eines Modells hängt im Wesentlichen von der Problemstellung sowie den Charakteristika der Datenbestände ab. Um die Qualität der Datenvorverarbeitung beurteilen zu können, werden Baseline-Verfahren eingesetzt. Diese Verfahren sind meist nicht sehr komplex und bieten daher Vorteile in Hinblick auf die Laufzeit der Algorithmen. Als Baseline werden häufig lineare Verfahren eingesetzt. Diese erreichen zwar nicht die Genauigkeit komplexer Verfahren, ermöglichen aber bereits mit wenig Konfigurationsaufwand akzeptable Ergebnisse. Sollten mit einem Baseline-Verfahren keine akzeptablen Ergebnisse erzielt werden, existieren zwei mögliche Ursachen. Entweder enthalten die Daten nicht genügend Informationen, um das vorliegende Problem angemessen zu beschreiben. In diesem Fall ist der Datenbestand und somit auch der Anwendungsfall ungeeignet für ML-Algorithmen. Alternativ kann das verwendete Modell die komplexen Zusammenhänge der Daten nicht abbilden, in diesem Fall kann mit der Implementierung eines komplexeren Modells begonnen werden.

Als Vorhersagemodell werden in der aktuellen Forschung häufig KNN eingesetzt, aber auch traditionelle Algorithmen wie etwa Support Vector Machines [16] oder lineare Regressoren können neue Erkenntnisse generieren. Um einen optimal geeigneten Algorithmus für eine spezifische Problemstellung auszuwählen, kommen häufig mehrere unterschiedliche Verfahren zum Einsatz, deren Komplexität sukzessive ansteigt.

Durch die Entwicklung erfolgreicher Systeme ist die Erwartungshaltung gegenüber den ML-Methoden in jüngster Vergangenheit stark gestiegen. Dies stellt eine besondere Herausforderung für die Anwender und Entwickler solcher Systeme dar. In der Praxis gestaltet sich die erfolgreiche Anwendung von Algorithmen häufig schwieriger als in verwandten Forschungsarbeiten. Die Genauigkeit der Systeme ist in der Regel stark abhängig von der Qualität des Datenbestands und damit auch von den Entscheidungen aus den vorherigen Phasen des KDD-Prozesses. Häufig fließt bei der Entwicklung eines selbstlernenden Systems ein Großteil der Arbeitszeit in die Aufbereitung des Datenbestands. Dennoch kann die Wahl eines geeigneten Algorithmus erheblichen Einfluss auf die finale Ergebnisoptimierung nehmen.

2.4.6 Evaluation

Der letzte Arbeitsschritt, die Evaluation der in Teilabschnitt 2.4.5 erlangten Erkenntnisse, entspricht vollständig der fünften Phase des KDD-Prozesses. Zu diesem Zweck werden alle Entscheidungen aus den vorangegangenen Schritten berücksichtigt und bewertet. Nach jeder Iteration des Verfahrens müssen die Ergebnisse neu bewertet und die einzelnen Arbeitsschritte gegebenenfalls angepasst werden, sodass das Ergebnis inkrementell verbessert werden kann. Die Bewertung der Ergebnisse erfolgt durch Experten der betreffenden Domäne anhand von folgenden Kriterien.

Die gewonnenen Erkenntnisse sollten auf neue Beobachtungen in der zuvor behandelten Domäne anwendbar sein und dementsprechend gut generalisierbar sein. Dies wird häufig unter Verwendung eines Testdatensatzes (siehe Kapitel 2.4.1), welcher zum Zeitpunkt des Trainings nicht bekannt ist, evaluiert. Weiterhin wird die Neuartigkeit der Erkenntnisse sowie deren Nützlichkeit und Anwendbarkeit in bestehenden Systemen beurteilt. Zuletzt sollten die Ergebnisse des KDD allgemein verständlich sein. Dies erfordert in einigen Fällen noch die Aufarbeitung der generierten Daten in visueller oder textueller Form.

Zwischen den vorangegangenen Phasen treten häufig Wechselwirkungen auf, sodass es sinnvoll ist, in jeder Iteration nur kleinere Änderungen vorzunehmen, um die Auswirkungen einzelner Entscheidungen isoliert betrachten zu können. Diese Entscheidungen betreffen die Wahl der eingesetzten Verfahren, aber auch die Optimierung von Hyperparametern. Da jede Problemstellung und jeder Datenbestand individuell zu betrachten ist, wird der Entscheidungsprozess häufig durch eine Vielzahl von Experimenten unterstützt. Der gesamte Entscheidungsraum ist bei vielen Problemen jedoch sehr groß, sodass unmöglich alle möglichen Parameterkombinationen getestet werden können [17]. Deshalb fließen Erfahrung und Intuition zu einem beträchtlichen Anteil in den Entscheidungsprozess mit ein.

Sobald die wesentlichen Entscheidungen bezüglich der Architektur und der Vorverarbeitung getroffen sind, kann die Wahl der Hyperparameter teilweise automatisiert werden. Häufig kommen Werkzeuge zum Einsatz, die die optimale Kombination von Hyperparametern, wie z.B. die Lernrate des neuronalen Netzes, mithilfe von wiederholten Experimenten in einem vordefinierten Suchraum finden.

Der Prozess des Erkenntnisgewinns ist abgeschlossen, sobald die oben genannten Kriterien erfüllt sind oder sich die Problemstellung aufgrund der Beobachtungen als ungeeignet erweist.

2.5 Zielsetzung

Die in dieser Arbeit verwendeten Daten entstammen dem DSHIP-System¹, einem Datenmanagement- und Verteilsystem der Firma Werum Software & Systems. Die Daten wurden während des regulären Forschungsbetriebs auf der Neumayer III aufgezeichnet. Die Aufzeichnung der Daten begann kurz nach der Inbetriebnahme der Station im Jahr 2009 und erfolgt seitdem kontinuierlich. Die verwendeten Daten wurden über ein Webinterface des DSHIP-Systems abgerufen, das auch von den Mitarbeitern der Forschungsstation genutzt wird. Bisher werden die aufgezeichneten Daten primär von Wissenschaftlern genutzt, um ihre Arbeit in verschiedenen Bereichen der Polarforschung zu unterstützen. In dieser Arbeit werden die Daten der Neumayer Station III erstmalig für die Entwicklung eines selbstlernenden Systems verwendet.

Die Integration von Echtzeiten ist üblicherweise mit zusätzlichen Schwierigkeiten verbunden, da diese zunächst aufbereitet und interpretiert werden müssen. Diese Arbeit

¹ https://www.werum.de/de/fachartikel.html?file=files/werum/PDF/Werum_BR_DSHIP_e.pdf

dient somit auch der Untersuchung des Datenbestandes auf dessen Qualität und Eignung im Rahmen eines maschinellen Lernverfahrens.

Zu diesem Zweck sind unterschiedliche Algorithmen für die Entwicklung eines Vorhersagemodells zum Einsatz gekommen. Zunächst wurde ein einfaches Baseline-Verfahren etabliert, welches anschließend als Referenz für die Bewertung komplexerer Verfahren eingesetzt wurde. In dieser Arbeit wurde ein lineares Regressionsverfahren als Baseline-Verfahren ausgewählt. Anschließend wurden komplexere neuronale Netze untersucht, welche performante Prognosen generieren können. Sowohl Dense Neural Networks (DNN) als auch Recurrent Neural Networks (RNN) sind in den Experimenten verwendet worden.

3 Praktische Durchführung

Das folgende Kapitel beschäftigt sich mit der Entwicklung eines Vorhersagemodells für den Energiebedarf der *Neumayer III*. Strukturell orientiert es sich an den in Teilabschnitt 2.4 beschriebenen Arbeitsschritten. Zunächst erfolgen eine detaillierte Beschreibung des vorliegenden Datenbestandes und eine erste Beurteilung bzgl. der Umsetzbarkeit einer DL-Lösung. Anschließend wird in Kapitel 3.2 die Vorauswahl der zu betrachtenden Datenpunkte beschrieben, welche in Abschnitt 3.3 von Unreinheiten bereinigt werden. Daraufhin werden in Kapitel 3.4 einige relevante Kanäle ausgewählt. Die Daten aus diesen Kanälen werden in Kapitel 3.5 in eine KNN-geeignete Darstellung transformiert. Abschließend erfolgt in Teilabschnitt 3.6 eine Beschreibung der eingesetzten Modelle und der Maßnahmen, welche zur Optimierung dieser Modelle ergriffen worden sind. In Kapitel 3.7 erfolgt abschließend eine Beschreibung des Entwicklungsprozesses aus technischer Sicht.

3.1 Dataset

Der Energiebedarf eines Gebäudes hängt von vielen unterschiedlichen Faktoren wie dem Wetter, den Bewohnern und den aktiven Geräten ab. Im Falle der *Neumayer III* nehmen außerdem wissenschaftliche Aktivitäten eine wesentliche Rolle bzgl. des Energiebedarfs des Forschungsgebäudes ein. Einige der besagten Informationen werden vom *DSHIP-System* der *Neumayer III* erfasst. Es wurde daher angenommen, dass dieser Datenbestand sich gut zur Entwicklung eines DL-basierten Vorhersagemodells für den Energiebedarf eignet. Diese Annahme soll im folgenden Kapitel evaluiert werden.

Die Sensordaten aus dem Bestand der *Neumayer III* werden vollautomatisch durch die Geräte der Forschungsstation aufgezeichnet. Diese Geräte umfassen Instrumente, welche für den Forschungsbetrieb eingesetzt werden sowie gebäudetechnisch relevante Geräte, z.B. die Generatoren. Sämtliche Daten des Systems liegen in einer Frequenz von einem Datenpunkt pro Minute vor. Der Datenbestand beinhaltet ausschließlich numerische sowie kategorische Kanäle, wobei die kategorischen Informationen numerisch kodiert worden sind. Aufgrund ihrer Beschaffenheit sind rein numerische Beobachtungen besonders schwierig zu interpretieren und erfordern weitere Informationen, wie z.B. die zugehörige physikalische Größe und eine Messeinheit. Das System erfasst insgesamt 614 Kanäle, welche 29 verschiedenen Gerätegruppen zuzuordnen sind (siehe

Tabelle 1). Jedem dieser Kanäle ist ein eindeutiger Identifikator zugeordnet, der häufig Rückschlüsse auf den Ursprung des Kanals zulässt. Weiterhin ist jeder Datenpunkt mit einem Zeitstempel verknüpft, sodass zeitliche Zusammenhänge zwischen den Informationen erhalten bleiben. 4 der insgesamt 19 Gerätegruppen beinhalten allerdings Geräte, welche thematisch vermutlich nicht mit einer Energieprognose verwandt sind, somit verbleiben 323 zu betrachtende Kanäle. Die Relevanz einzelner Gerätegruppen ging aus Gesprächen mit den zuständigen Mitarbeitern des AWI hervor.

Weiterhin ist zu erwähnen, dass im betrachteten Zeitraum Änderungen an der Infrastruktur des Stationsgebäudes vorgenommen worden sind und neue Laboratorien ergänzt worden sind. Diese Entwicklungen sind im Datenbestand jedoch nicht vermerkt und verzerren möglicherweise die Erkenntnisse. Dies stellt eine weitere Herausforderung für den Anwendungsfall dar, da ein Vorhersagemodell an solche Änderungen adaptiert werden muss.

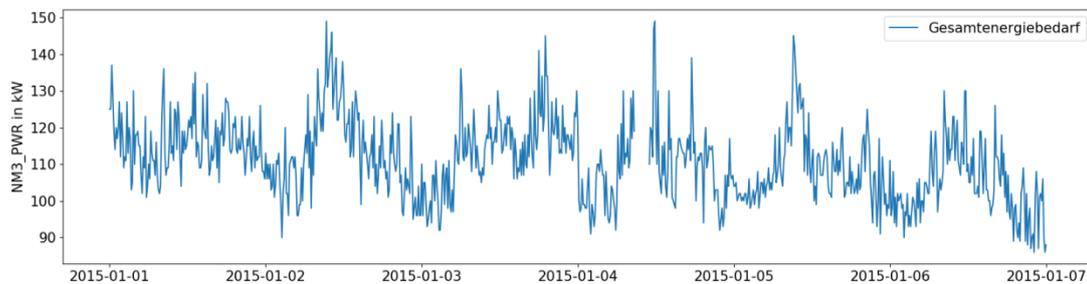


Abbildung 3: Gesamtenergiebedarf der Neumayer III über eine Woche im Januar 2015

Um die Genauigkeit der Vorhersagemodelle zu beurteilen, wurde zu Beginn des Entwicklungsprozesses eine Baseline auf der Grundlage von fachlichen Entscheidungen etabliert. In den Trainingsdaten ist zu beobachten, dass der Energiebedarf der Forschungsstation in den Nächten abnimmt und tagsüber, meist um die Mittagstunden herum, den Tageshöchstwert erreicht (siehe Abbildung 3). Es wurde daher angenommen, dass der Energiebedarf der Station an zwei aufeinanderfolgenden Tagen nicht allzu stark variiert. Folglich wurden als Baseline die Werte des Vortags zur jeweils gleichen Zeit eingesetzt.

Tabelle 1: Relevanz der Gerätegruppen

GERÄTEGRUPPE	ANZAHL DER KANÄLE	RELEVANT
400/401 BHKW	13	Ja
Anlage 500/600	2	Ja
Anlage 100 Containerlüftung	8	Ja
Anlage 200 Containerlüftung	8	Ja
BHKW und Notstromanlage	39	Ja
BHKW1	53	Ja
BHKW2	53	Ja
BHKW3	53	Ja
Belüftung Werkstatt	5	Ja
Energiemanagement	21	Ja
Frischwasser	4	Ja
GPS	3	Nein
Hüllenumluft	10	Ja
Hydraulik	154	Nein
Hydraulikanlage	6	Nein
Kraftstoffversorgungsanlage	1	Ja
Ladespeicher WWB	3	Ja
Lastkreise	12	Ja
Meteorologie	27	Ja
RKW BHKW	4	Ja
RKW Not	4	Ja
Schneesmelze	10	Ja
Serverraum	3	Ja
System	72	Nein
Temperatursensoren	10	Ja
USV Anlage	6	Ja
Wärmengenerfassung	27	Ja
Zusatzmelder Hülle	3	Ja

3.2 Datenselektion

Im Anschluss an eine erste Analyse der Daten erfolgte die Auswahl geeigneter Datenpunkte aus dem Gesamtdatenbestand. Da es sich bei der Energieprognose um ein Zeitreihenproblem handelt, wurden nur Datenpunkte aus einem eingeschränkten Zeitraum betrachtet. Ein Großteil der Kanäle der Forschungsstation erfassen Sensordaten ab Mitte des Jahres 2010. Außerdem weisen einige Kanäle ab Beginn des Jahres 2017 einige Anomalien auf. Als Anomalien wurden solche Datenpunkte und Muster bewertet, welche sich signifikant von den übrigen Beobachtungen des jeweiligen Kanals unterscheiden haben. Um diese zu identifizieren, sind die Werte der einzelnen Kanäle graphisch dargestellt worden. Einige Anomalien konnten durch eine domänenbezogene Betrachtung erklärt werden, andere wurden jedoch als Fehler identifiziert. Auf der Grundlage dieser Beobachtungen wurden im Folgenden nur Daten berücksichtigt, welche zwischen Beginn 2011 und Ende 2016 aufgezeichnet worden sind. In diesem Zeitraum sind keine erkennbaren Anomalien aufgetreten. Die Betrachtung eines kürzeren Zeitraums würde

zwar die Genauigkeit der Vorhersagen verbessern, beeinträchtigt jedoch die Generalisierbarkeit des Modells.

Um die Ergebnisse des entwickelten Prognosemodells zu evaluieren und eine Überanpassung zu erkennen, ist es erforderlich einen Testdatenbestand auszuwählen. In dieser Arbeit wurden 20 % des selektierten Datenbestands als Testdatenbestand ausgewählt. Trotz des geringen Anteils kann dieser Bestand die Fähigkeit des Modells zu generalisieren hinreichend bemessen.

Aus dem betrachteten Zeitraum sind Zeitscheiben extrahiert worden, welche anschließend als Trainingsamples eingesetzt worden sind. Diese sind in jeder Trainingsiteration zufällig dem Trainings- oder Testdatensatz zugeordnet worden. Dies hat zur Folge, dass die Beurteilung eines trainierten Modells zwischen zwei Trainingsiterationen auf der Grundlage von unterschiedlichen Datenpunkten geschieht. Weiterhin wird das Modell in diesem Fall mit Beobachtungen trainiert, die möglicherweise nach dem vorherzusagenden Ereignis auftreten.

Weitere Experimente wurden daher unter Verwendung eines Testsets durchgeführt, dessen Selektion anhand einer zeitlichen Trennlinie erfolgt. Zu diesem Zweck wurden die Datenpunkte aus dem Jahr 2016 als Testdaten gewählt, da diese Daten die höchste Aktualität aufweisen und am meisten Parallelen zu einem möglichen Produktiv-Einsatz des Vorhersagesystems aufweisen. Dieser Testdatensatz nimmt einen Anteil von 16.7 % der zuvor selektierten Daten ein. Bei dieser Strategie ist jedoch zu beachten, dass der vorliegende Datenbestand möglicherweise zeitbedingte Entwicklungen beinhaltet. Diese Entwicklungen entstehen z.B. durch die Erweiterung der Forschungsstation um ein neues Laboratorium. Falls solche Entwicklungen erst nach Ende des Trainingszeitraums auftreten, ist ein vollständig trainiertes Modell eventuell nicht in der Lage, diese bei der Prognose zu berücksichtigen. Dies kann zu ungenauen Prognosen führen.

3.3 Data Cleaning

Wie bereits in Teilabschnitt 2.4.2 erwähnt, enthalten reale Datenbestände häufig Unreinheiten. Diese sind zurückzuführen auf technische Defekte, den Ausfall von Sensoren, oder Bedienerfehler bei der Pflege eines Systems. Solche Unreinheiten können die Genauigkeit eines intelligenten Vorhersagesystems erheblich einschränken, falls sie nicht angemessen behandelt werden. Auch in den Datenbeständen der *Neumayer III* konnten einige Unreinheiten identifiziert werden. Im Folgenden wird der verwendete Datenbestand unter Berücksichtigung dieser Problematik untersucht und es werden die ergriffenen Maßnahmen zur Bereinigung des Datenbestandes erläutert. Dabei wird unterschieden zwischen fehlenden und fehlerhaften Daten.

Die Problematik der fehlenden Daten ist in Kapitel 3.2 durch die Einschränkung des Betrachtungszeitraums bereits erheblich reduziert worden. Bei einer Frequenz von einem Datenpunkt pro Minute ergeben sich bei 323 selektierten Kanälen insgesamt etwa 1,2 Mrd. Sensorwerte. Von diesen Sensorwerten sind jedoch etwa 10,9 Prozent, vermutlich aufgrund eines Defekts oder einer Fehlkonfiguration nicht erfasst worden, sodass Lücken in der

Aufzeichnung entstanden sind. In diesem konkreten Anwendungsfall stellt die Abwesenheit von Datenpunkten daher keine relevante Information dar. Weil der Defekt eines Geräts zunächst bemerkt werden muss, treten diese Lücken häufig über einen längeren und zusammenhängenden Zeitraum auf

Bei der Bereinigung von unvollständigen Datenpunkten ist weiterhin die Verteilung der betroffenen Datenpunkte über unterschiedliche Gerätegruppen zu beachten. In der Gruppe „Lastkreise“ sind insgesamt 20,2% der Datenpunkte nicht erfasst worden, in der Gruppe „Anlage 200 Containerlüftung“ beträgt der Anteil sogar 88,9%.

Die Rekonstruktion der fehlenden Datenpunkte in dieser Arbeit wurde mit dem Einsatz eines gleitenden Zeitfensters realisiert. Dazu wurden die verfügbaren Informationen um die betroffenen Beobachtungen ausgewertet, um diese zu interpolieren. Je nach Dauer des Ausfalls wurden Zeitfenster der Länge 1, 3, 7 sowie 28 Tage eingesetzt. Aus den verbliebenen, korrekten Informationen dieser Zeitfenster wurde der Median berechnet und für die fehlenden Werte eingesetzt. An dieser Stelle gilt es zu erwähnen, dass Kanälen, deren Werte oszillieren, durch die Berechnung des Medians nur relativ unpräzise rekonstruiert werden können. Weitere Experimente wurden daher unter Einsatz des arithmetischen Mittels anstelle des Medians durchgeführt. Es hat sich jedoch herausgestellt, dass unregelmäßig auftretende, häufig durch Messfehler bedingte Extremwerte in den Daten das arithmetische Mittel zu sehr verzerren und es dadurch zu einer Beeinträchtigung der Vorhersage kommt. Daher wurde in den folgenden Experimenten der Median verwendet.

Im Folgenden wird exemplarisch eine Anomalie beschrieben, die bei der Untersuchung der Kanäle auf fehlerhafte Daten identifiziert worden ist. Im vorliegenden Datenbestand konnten mehrere solcher Anomalien anhand der graphischen Darstellung von Kanalinformationen identifiziert werden.

Abbildung 4 zeigt die Werte des Kanals NM3_PWR, welcher den Gesamtenergiebedarf erfasst, im Zeitraum von 2015 bis 2018. In diesem Diagramm ist zu beobachten, dass die aufgezeichneten Werte im April 2017 enorm ansteigen. Der Mittelwert sowie die Varianz der Werte in diesem Zeitraum weichen stark von den übrigen Beobachtungen ab. Aus diesen Beobachtungen lässt sich ableiten, dass an dieser Stelle ein Fehler vorliegt. Der betroffene Zeitpunkt konnte in Zusammenarbeit mit einem Domänenexperten auf einen Bedienerfehler bei der Datenaufzeichnung zurückgeführt werden. Der Gesamtenergiebedarf der *Neumayer III* ergibt sich aus der Kombination mehrerer Kanäle, welche den Energieverbrauch einzelner Gerätegruppen abbilden. In diesem konkreten Fall ist der Kanal „WKA_PWR“ mit dem Faktor 10 in die Berechnung des Gesamtbedarfs eingeflossen. Die betroffenen Informationen konnten daher durch Anwendung der Gleichung 1 verlustfrei rekonstruiert werden.

$$PWR_{neu}(t) = PWR_{alt}(t) - 9 * WKA_PWR(t)$$

Gleichung 1: Formel zur Rekonstruktion des Energiebedarfs im Zeitraum

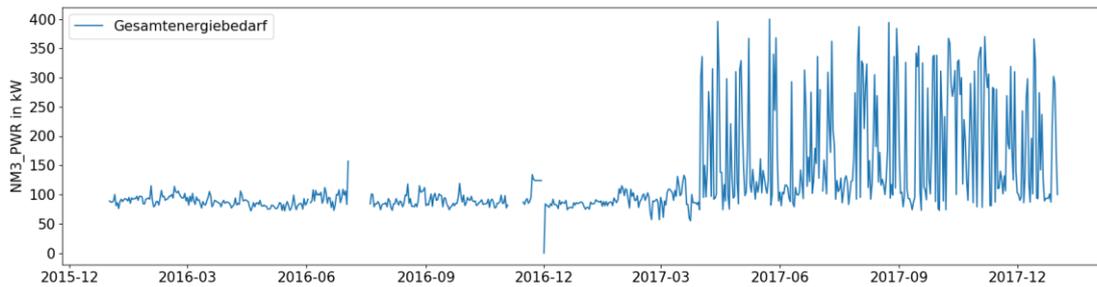


Abbildung 4: Anomalie im Gesamtenergiebedarf im Zeitraum 2016-2018

Die Anzahl fehlender Datenpunkte im Datenbestand der *Neumayer III* ist für die Entwicklung eines Prognosemodells als unkritisch eingestuft worden. Betroffene Datenpunkte nehmen nur einen geringen Anteil am Gesamtbestand ein und konnten durch geeignete Maßnahmen rekonstruiert werden. Dadurch ist der Informationsgehalt hinreichend groß, um eine performante Prognose zu gewährleisten. Einzig die Kanäle der Gerätegruppe „Anlage200 Containerlüftung“ sind aufgrund der hohen Fehlerrate als ungeeignet bewertet worden und wurden im Folgenden nicht weiter betrachtet.

Es ist jedoch anzunehmen, dass der übrige Datenbestand einige fehlerhafte Datenpunkte beinhaltet, welche in diesem Arbeitsschritt nicht identifiziert werden konnten. Weitere Maßnahmen sind aus zeitlichen Gründen aber nicht ergriffen worden.

3.4 Feature Selection

Nachdem die Daten selektiert und bereinigt worden sind, wurden anschließend besonders relevante Merkmale ausgewählt. Die Reduktion der Trainingsdaten auf wenige Merkmale reduziert ggf. die Anzahl der erlernbaren Parameter und erlaubt somit einen schnelleren Trainingsprozess (siehe Teilabschnitt 2.4.3).

Die erste Strategie zur Merkmalsselektion, die in dieser Thesis angewendet wurde, sieht die fachliche Bewertung des Anwendungsfalls vor. Zu diesem Zweck wurden Empfehlungen der Experten des Alfred-Wegener-Instituts berücksichtigt. Diese wählten insgesamt 34 Sensoren aus.

Bei einer Reduktion der Trainingsdaten von 323 verfügbaren auf 34 tatsächlich genutzte Kanäle gehen jedoch viele Informationen verloren, welche möglicherweise relevant für die Energieprognose sind. Unter Berücksichtigung dieser Annahme wurden in einem alternativen Versuchsaufbau sämtliche verfügbaren Kanäle für das Training des Modells genutzt. In diesem Fall wird die Bewertung der einzelnen Merkmale von dem eingesetzten Modell übernommen. Dies setzt jedoch ein Modell voraus, welches als Featureextraktor fungieren kann. In anderen Arbeiten konnte gezeigt werden, dass neuronale Netze diese Funktion gut ausfüllen können [18]. Diese Strategie kommt vollständig ohne den Einsatz

von domänenspezifischen Experten aus und ist somit auf jede beliebige Problemstellung übertragbar.

Die statistische Auswertung aller verfügbaren Kanäle ermöglicht ebenfalls die Auswahl ohne Einbindung von domänenspezifischem Wissen. Bei diesem Ansatz wurden zunächst alle Kanäle entfernt, die keinerlei statistische Varianz aufweisen, sondern über den gesamten betrachteten Zeitraum einen konstanten Wert aufzeichnen und somit keinen Einfluss auf die Energieprognose nehmen können. Dies betrifft insgesamt 62 von 323 Kanälen. Weiterhin wurde eine Korrelationsanalyse durchgeführt, um besonders aussagekräftige Merkmale zu identifizieren.

Alternativ zu diesem Verfahren wurde ein Random Forest-Algorithmus eingesetzt. Dieses Modell dient jedoch nicht als Regressionsmodell, sondern als Instrument, welches die Relevanz einzelner Merkmale beurteilt. Zu diesem Zweck wurde der Algorithmus mit sämtlichen verfügbaren Merkmalen trainiert. Jeder Knoten des resultierenden Entscheidungsbaums stellt eine Entscheidung in Abhängigkeit eines Merkmals dar. In Annahme haben jene Entscheidungen, welche sich nah an der Wurzel des Entscheidungsbaums befinden, einen besonders hohen Einfluss auf die Zielvariable. In dieser Arbeit wurden die Trainingsdaten in einigen Experimenten auf solche Merkmale beschränkt, die vom RandomForest-Algorithmus als besonders relevant bewertet worden sind.

In diesem Kapitel wurden drei Strategien zur Selektion geeigneter Merkmale genannt, welche im experimentellen Teil der Arbeit untersucht wurden. In Kapitel 4 erfolgen eine umfangreiche Auswertung der Ergebnisse und eine Bewertung der genannten Strategien.

Zusätzlich zu den Sensordaten der *Neumayer III* wurde der Datenbestand um einige Informationen ergänzt, die vom *DSHIP-System* der Forschungsstation nicht erfasst werden, aber vermutlich einen wesentlichen Einfluss auf den Energiebedarf des Forschungsgebäudes nehmen. Es wurde daher angenommen, dass die Einbindung dieser Merkmale die Genauigkeit eines Vorhersagesystems verbessert. Folgende Merkmale wurden daher den Trainingsdaten hinzugefügt.

- Anzahl der Stationsbewohner
- Wochentag
- Monat
- Tag des Monats

Im antarktischen Sommer kann das Stationsgebäude der *Neumayer III* bis zu 50 Personen aufnehmen, im Winter wird die Station aufgrund der unwirtlichen Wetterbedingungen jedoch nur von 9 Wissenschaftlern bewohnt. Weiterhin dient die Forschungsstation im Sommer als Ausgangspunkt für Expeditionen in die Antarktis, daher variiert die Anzahl der wissenschaftlichen Mitarbeiter ständig. Da jeder der Stationsbewohner Geräte betreibt, welche Energie benötigen und zum Gesamtenergieverbrauch des Stationsgebäudes beitragen, ist die Anzahl der Bewohner von hoher Bedeutung für die Prognose des Energiebedarfs.

Die übrigen Kontextinformationen bilden zeitliche Zusammenhänge ab und verbessern somit die Fähigkeit des Modells, wiederkehrende Muster zu erkennen. Diese Informationen sind bereits im Zeitstempel enthalten. Da sie in Form eines Zeitstempels jedoch nicht verarbeitet werden können, ist dieser in mehrere Merkmale aufgeteilt worden. Ein Beispiel für solche wiederkehrenden Muster stellen die Außentemperaturen dar, welche in den Wintermonaten stark abnehmen, sodass mehr Energie für die Beheizung des Stationsgebäudes benötigt wird. Dieser Zusammenhang wird durch die Ergänzung des laufenden Monats in den Daten abgebildet.

Die Wahl nützlicher Kontextinformationen wurde unterstützt durch mehrere Experimente mit unterschiedlichen bzw. keinen ergänzten Informationen. Dabei konnte beobachtet werden, dass die Prognose durch das Einbinden der oben genannten Kontextinformationen an Genauigkeit zunimmt.

In Teilabschnitt 2.4.4 wurde bereits erwähnt, dass in verwandten Arbeiten häufig eine Principal Component Analysis durchgeführt wird. In dieser Arbeit wurde ebenfalls eine PCA vorgenommen, um die Auswirkungen des Verfahrens auf den Informationsgehalt der Trainingsdaten zu untersuchen. Dabei wurden verschiedene Hauptkomponenten gebildet, um ein Modell zu trainieren. Es hat sich jedoch herausgestellt, dass die Verwendung einer PCA in diesem konkreten Anwendungsfall die Qualität der Energieprognose wesentlich beeinträchtigt.

3.5 Datentransformation

Im Folgenden werden die Arbeitsschritte beschrieben, welche für die allgemeine Transformation der Daten erforderlich waren. Anschließend wird die Transformation in sequenzielle Daten erläutert, welche für die Verwendung von RNN benötigt wird.

Um den Effekt unterschiedlicher Wertebereiche (siehe Teilabschnitt 2.4.4) zu eliminieren, wurde eine Normalisierung der zuvor ausgewählten Trainingsdaten durchgeführt. Dabei wurden die Kanäle individuell skaliert, sodass die resultierenden Werte eines Kanals normalverteilt sind. Dies ist Voraussetzung für einige Verfahren des maschinellen Lernens und wirkt sich zudem häufig positiv auf die Qualität eines Prognosemodells aus.

Alternativ zur Normalisierung erfolgte in einigen Experimenten eine Skalierung der Trainingsdaten. Dabei wurden sämtliche Kanäle auf einen Wertebereich von $-1 \leq x \leq 1$ abgebildet. Es konnten jedoch keine Verbesserungen gegenüber der zuvor benannten Transformation festgestellt werden.

Im Anschluss an die Normalisierung bzw. Skalierung ist ein Downsampling durchgeführt worden. Zu diesem Zweck wurde aus mehreren Datenpunkten das arithmetische Mittel gebildet, um die Frequenz der Daten zu reduzieren. Diese Technik ist möglicherweise nicht optimal, da Kanäle, deren Werte oszillieren, durch das arithmetische Mittel verzerrt werden können.

Danach wurden die Trainingsdaten sequenzialisiert, um die Vorzüge eines RNN optimal nutzen zu können. Die Verwendung von sequenziellen Trainingsdaten ermöglicht es dem

RNN, wiederkehrende und zeitlich bedingte Muster zu erkennen und diese bei einer Prognose zu berücksichtigen.

Dazu wurde eine gleitende Fensterfunktion auf die Daten angewandt, sodass aufeinanderfolgende Datenpunkte zu einer Sequenz aggregiert werden (siehe Abbildung 5). Bei einer solchen Fensterfunktion gilt es zwei Parameter zu beachten, die Länge des Fensters und die Schrittweite. Aus der Kombination dieser Parameter ergibt sich die Anzahl der Zeitschritte in den resultierenden Sequenzen. In verwandten Arbeiten konnte gezeigt werden, dass LSTM-basierte RNN in Sequenzen mit mehr als 1000 Zeitschritten keine Zusammenhänge erkannt werden können [17].

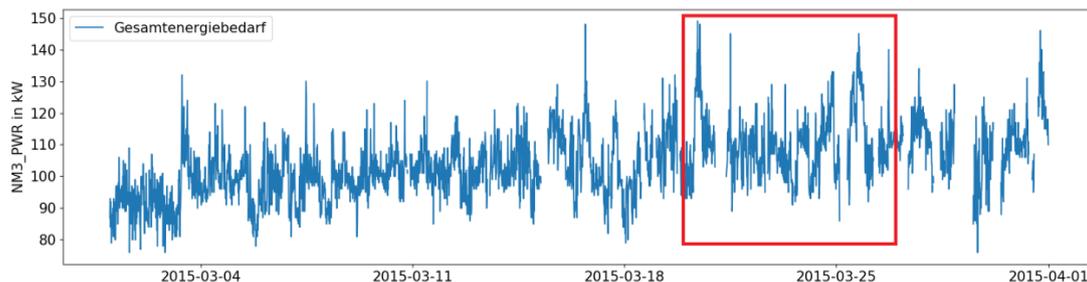


Abbildung 5: Fensterfunktion für die Sequenzialisierung der Trainingsdaten

Die Länge des Fensters beschreibt, wie weit die betrachteten Informationen in einer Sequenz zurückliegen sollen. Die konkrete Wahl dieses Parameters ist abhängig von den Charakteristika der betrachteten Problemstellung und dementsprechend individuell zu treffen. In dieser Arbeit wurde mit Sequenzlängen zwischen einem Tag und zwei Wochen experimentiert.

In dem vorliegenden Datenbestand repräsentiert jeder verfügbare Datenpunkt einen fixen Zeitpunkt. Es ist prinzipiell vorteilhaft, kurze Schrittweiten zu wählen, da diese einen geringen Informationsverlust durch das Downsampling gewährleisten. Dabei gilt es jedoch, die Komplexität des Trainingsprozesses und die Obergrenze von 1000 Zeitschritten (siehe Teilabschnitt 2.3.1) zu beachten. Im experimentellen Teil dieser Arbeit wurde die Schrittweite zwischen 5 und 60 Minuten variiert. Da die Vorhersage des Energiebedarfs in stündlichen Intervallen erfolgt, ist eine Schrittweite von mehr als 60 Minuten für diesen Anwendungsfall nicht praktikabel. Kürzere Schrittweiten hingegen sind aufgrund der rechnerischen Komplexität nicht möglich gewesen.

3.6 Data Mining

In den folgenden Teilabschnitten werden Architektur und Konfiguration der eingesetzten Modelle detailliert beschrieben. Weiterhin werden diese Modelle anhand ihrer Anwendbarkeit auf den vorliegenden Datenbestand untersucht. Die zuvor beschriebene Datenvorverarbeitung ist dabei unabhängig von der Wahl des Modells zu betrachten. Die

im vorigen Teilabschnitt beschriebene Transformation der Trainingsdaten in sequenzielle Form bezieht sich jedoch nur auf Experimente mit rekurrenten Netzen.

Die in dieser Arbeit entwickelten Modelle sollen nach Abschluss des Trainings in der Lage sein, einen skalaren Wert für den Energiebedarf der *Neumayer III* vorherzusagen. Um alle entwickelten Modelle gegeneinander evaluieren zu können, wurden als Metriken die *mittlere quadratische Abweichung* (MSE) sowie der *Mittlere absolute Fehler* (MAE) eingesetzt. Diese sind in Gleichung 2 und Gleichung 3 dargestellt. Der MSE ist außerdem als Fehlerfunktion im Trainingsprozess aller neuronalen Netze verwendet worden, da dieser in der Forschung als beliebte Fehlerfunktion in Regressionsproblemen gilt. Diese Funktion bewirkt, dass extreme Abweichungen in den vorhergesagten Werten besonders schlecht bewertet werden und das Modell motiviert wird, diese schneller zu korrigieren.

$$MSE = \frac{1}{n} * \sum_{i=1}^n (Y_i - X_i)^2$$

Gleichung 2: Mittlere quadratische Abweichung (MSE)

$$MAE = \frac{1}{n} * \sum_{i=1}^n (Y_i - X_i)$$

Gleichung 3: Mittlerer absoluter Fehler (MAE)

In weiteren Experimenten konnte beobachtet werden, dass sich die Verwendung des MAE als Fehlerfunktion negativ auf die Qualität der Prognose auswirkt.

Zunächst ist in dieser Arbeit ein lineares Regressionsverfahren eingesetzt worden, um eine fortgeschrittene Bewertungsgrundlage für die übrigen Modelle zu schaffen. Die Vorverarbeitung ist in allen Experimenten und unabhängig von den konkreten Modellen identisch konfiguriert worden.

3.6.1 DNN-Modell

Aufgrund der zuvor bereits beschriebenen Charakteristiken des vorliegenden Datenbestandes wurden in dieser Thesis KNN eingesetzt, welche die Komplexität und Genauigkeit eines Linearen Regressors übersteigen.

Beim Training eines neuronalen Netzes stellt die optimale Wahl der Hyperparameter, wie etwa der Lernrate, eine große Herausforderung dar und kann gravierende Auswirkungen auf die Ergebnisse eines Modells haben. Zudem ist die Optimierung der Hyperparameter enorm zeitaufwändig. Unter Verwendung von Erfahrungswerten und unter Berücksichtigung verwandter Arbeiten konnten die kritischen Hyperparameter bereits auf einen geeigneten Bereich eingeschränkt werden. Anschließend wurden viele Modelle mit verschiedenen Parameterkombinationen im zuvor festgesetzten Bereich trainiert, um eine

hinreichend performante Konfiguration experimentell zu identifizieren. Die Vollständigkeit der Parametersuche ist jedoch abhängig von der verfügbaren Rechenkapazität bzw. -zeit, die gefundenen Konfigurationen sind also nicht zwangsläufig optimal.

Um die Stabilität des Trainingsprozesses für alle Konfigurationen zu beurteilen, sind die zugehörigen Modelle in jeweils drei Iterationen trainiert worden. Konfigurationen, die in den Experimenten eine hohe Varianz aufwiesen, wurden in dieser Arbeit schlechter bewertet, da die Stabilität des Trainingsprozess eine wichtige Anforderung für den Produktiv-Einsatz eines intelligenten Vorhersagesystems darstellt.

Sämtliche neuronalen Netze in dieser Arbeit wurden unter Verwendung der Optimierungsfunktion *adam* [19] trainiert, da diese in verwandten Arbeiten häufig eingesetzt worden ist.

In der Literatur zu Deep Learning werden häufig variable Lernraten beim Training eines Modells verwendet, um eine Oszillation im Gradienten-Abstieg zu verhindern und damit die Genauigkeit des Systems zu verbessern. Smith variiert die Lernrate der Optimierungsfunktion innerhalb eines zuvor definierten Bereichs, ähnlich der Sinusfunktion [20]. Dabei konnte gezeigt werden, dass eine kurzzeitige Erhöhung der Lernrate die Überbrückung von lokalen Minima im Gradientenverfahren unterstützt. Zudem beschleunigt die Verwendung einer variablen Lernrate die Konvergenz des neuronalen Netzes. In Folge dessen wurden die Modelle in dieser Arbeit ebenfalls unter Verwendung einer zyklischen Lernrate trainiert. Bei der Verwendung einer zyklischen Lernrate in Kombination mit dem adaptiven Optimierungsalgorithmus *adam* konnten die oben genannten Vorteile beobachtet werden.

In den ersten Iterationen der Modellentwicklung sind Dense Neural Networks zum Einsatz gekommen, da diese die einfachste Form eines KNN darstellen. In dieser Arbeit wurden verschiedene Varianten eines DNN getestet und evaluiert, im Folgenden werden die wesentlichen Entscheidungen bezüglich der Architektur benannt.

Die eingesetzten neuronalen Netze bestehen aus einer bzw. mehreren aufeinanderfolgenden Schichten, gefolgt von einer weiteren Ausgabeschicht mit 24 Neuronen. Jedes dieser Neuronen gibt einen skalaren Wert aus, die Kombination dieser Werte bildet die Vorhersage des Energiebedarfs für den Folgetag in Abständen von jeweils einer Stunde. Als Aktivierungsfunktion der Neuronen wurde *Rectified Linear Unit* (ReLU) gewählt, da diese das Training gegenüber anderen üblichen Aktivierungsfunktionen beschleunigen konnte.

Während des Trainings der DNN-Netzwerke konnte beobachtet werden, dass der Test-Error nach einiger Zeit stagniert und anschließend wieder ansteigt, während der Trainings-Error weiterhin abnimmt. Dies deutet auf ein Overfitting des Modells hin. Um diesem entgegenzuwirken, wurden in dieser Arbeit zwei Maßnahmen ergriffen, Dropout [21] und eine Regularisierung der trainierbaren Gewichte.

Des Weiteren wurde der Trainingsprozess eines Modells vorzeitig unterbrochen, wenn sich der Test-Error über viele Epochen nicht verbessert hat. Dies verhindert ein übermäßiges Training des Modells, welches häufig in Verbindung mit einer starken Überanpassung auftritt [22]. In dieser Arbeit wurde der Trainingsprozess abgebrochen, sobald nach 100

Epochen keine Verbesserung zu beobachten war. In den Experimenten konnte beobachtet werden, dass das Gradientenverfahren, welches im Training der Netze eingesetzt wird, bis zu 100 Epochen benötigt, um ein lokales Minimum bzw. ein Plateau zu verlassen.

3.6.2 LSTM-Modell

Aufgrund der Einschränkungen der DNN aus Teilabschnitt 3.6.2 in Bezug auf die Verarbeitung von Zeitreihendaten wurden im Folgenden Recurrent Neural Networks untersucht. Diese eignen sich besonders gut für die Erkennung von zeitlichen Abhängigkeiten und wurden bereits erfolgreich in ähnlichen Arbeiten eingesetzt. LSTM-basierte Netze gelten als State-of-the-Art bei der Lösung von Zeitreihenproblemen und wurden deshalb auch in dieser Arbeit verwendet. Alternativ zu LSTM-basierten Netzwerken werden oft Gated Recurrent Units (GRU) eingesetzt, da diese effizienter bzw. schneller zu trainieren sind.

Die im Rahmen dieser Arbeit eingesetzten RNN setzen sich zusammen aus einem bzw. mehreren LSTM-Schichten. Die jeweils letzte LSTM-Schicht gibt dabei lediglich die letzte Ausgabe einer Sequenz weiter, während die übrigen LSTM-Schichten die gesamte Sequenz ausgeben. Im Anschluss an die rekurrenten Schichten des RNN folgen einige Dense Layer, welche analog zum zuvor beschriebenen DNN aufgebaut sind. Die LSTM-Schichten der Modelle nutzen als Aktivierungsfunktion \tanh , da diese in Keras standardmäßig für LSTM verwendet wird und das Problem von zu großen Gradienten in LSTM-Netzen nicht auftritt. Die Aktivierung des letzten Dense-Layer hingegen erfolgt durch die $ReLU$ Funktion. Abbildung 6 enthält die schematische Darstellung der Netzwerkarchitektur eines LSTM-basierten RNN, wie sie in dieser Arbeit eingesetzt worden sind.

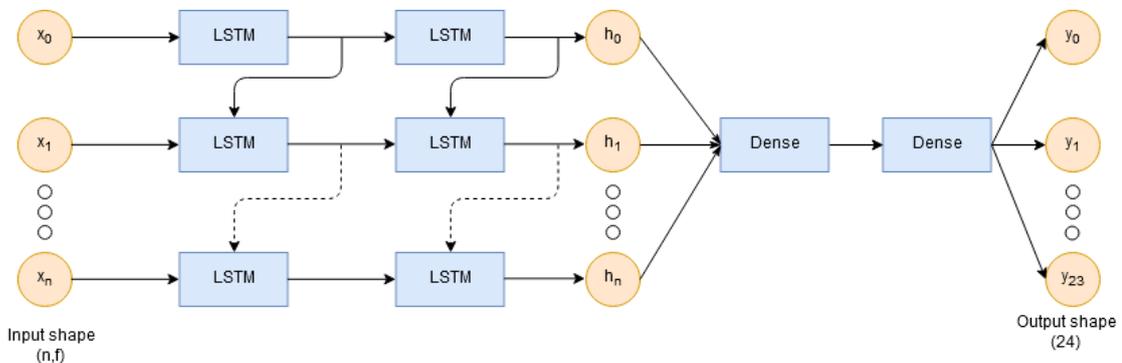


Abbildung 6: Schematische Darstellung eines LSTM-basierten RNN mit zwei rekurrenten Schichten

Auch bei den LSTM-Netzen ist während des Trainings eine Überanpassung an die Trainingsdaten aufgetreten. Um diesem Problem entgegenzuwirken, ist in den RNN *recurrent dropout* eingesetzt worden, eine Variation der Standardimplementierung von Dropout [23]. In dieser Variante wird ein Teil der internen Verbindungen, welche für die

Verarbeitung von Sequenzen in LSTM-Schichten entscheidend sind, entfernt. Im Gegensatz zur herkömmlichen Implementierung werden dabei während einer Batch immer dieselben Verbindungen deaktiviert. In dieser Thesis konnte nachgewiesen werden, dass die Verwendung von *recurrent dropout* sich positiv auf die Genauigkeit eines Prognosemodells auswirkt und einer Überanpassung der Modelle entgegenwirkt.

Trotz des Einsatzes von *recurrent dropout* weisen die trainierten Modelle eine Überanpassung zu den Trainingsdaten auf. Eine weitere Maßnahme gegen die Überanpassung neuronaler Netze stellt die Regularisierung der trainierbaren Parameter dar. Diese Regularisierung bewirkt, dass die Gewichte des neuronalen Netzes gewissen Einschränkungen unterliegen und die Ergebnisse des Netzes somit besser generalisierbar sind. In dieser Arbeit wurden unterschiedliche Regularisierungsmethoden getestet. Eine Verbesserung der Qualität des Vorhersagemodells konnte jedoch nur bei Verwendung einer bias-Regularisierung festgestellt werden.

Weiterhin wurden in dieser Arbeit bidirektionale rekurrente Netze genutzt [24]. Es konnte beobachtet werden, dass diese Netzwerke schneller zu einer guten Lösung konvergieren und der Trainingsprozess daher beschleunigt werden kann. Weiterhin konnten dort leichte Verbesserungen der Genauigkeit nachgewiesen werden.

In Produktiv-Umgebungen werden intelligente Prognosemodelle häufig eingesetzt, um Entscheidungsprozesse zu unterstützen und Empfehlungen auszusprechen. Wenn kritische Entscheidungen davon abhängen, ist es notwendig die Unsicherheit eines solchen Systems abzubilden und ggf. fehlerhafte Prognosen als solche zu erkennen. In [25] wurde dazu eine Technik mit dem Namen Monte-Carlo-Dropout vorgestellt. Diese Technik erlaubt es, in Dropout-basierten Netzwerken präzise Aussagen über die Unsicherheit des Modells zu treffen.

Sollten dem Modell Samples übergeben werden, welche sich stark von den bekannten Trainingsdaten unterscheiden, kann das Modell diese nicht zuverlässig vorhersagen. Demnach sollte für solche Samples die Prognoseunsicherheit sehr hoch sein. In dieser Arbeit sind einige Experimente mit *Monte-Carlo-Dropout* durchgeführt worden, es konnte jedoch keine Korrelation zwischen der berechneten Unsicherheit und den Abweichungen in der Prognose beobachtet werden.

3.7 Versuchsaufbau

Sämtliche Bibliotheken und Frameworks, die in dieser Arbeit eingesetzt worden sind, unterliegen einer Open-Source-Lizenz und sind somit jedem Entwickler frei zugänglich. Dank dieser Softwarelösungen liegt die Einstiegshürde für die Entwicklung von intelligenten und selbstlernenden Systemen enorm niedrig. Die Anwendung von ML-Algorithmen steht nahezu jedem offen, lediglich das Training besonders komplexer Modelle erfordert performante und teure Hardware. Die Verwendung von GPU-basierten Cloud-Lösungen relativiert jedoch auch dieses Problem. Mittlerweile ist es jedem Data Scientist möglich, den

Datensatz *ImageNet*², welcher häufig als Benchmark für Bilderkennungsalgorithmen in wissenschaftlichen Artikeln genutzt wird, mit einer Investition von 40 US-Dollar bis zu einer Genauigkeit von 93% zu trainieren³.

Die zuvor beschriebenen neuronalen Netze wurden entwickelt unter Verwendung der Deep-Learning-Bibliothek *Keras*. *Keras* stellt eine High-Level-API zur Verfügung und ermöglicht dem Anwender einen simplen Einstieg in die Welt des DL. Die Bibliothek ist in *Python* implementiert und nutzt als Backend das *Tensorflow* Framework von *Google*, welches die effiziente Berechnung von matrizenbasierten Algorithmen implementiert. *Tensorflow* ermöglicht außerdem die Ausführung auf Grafikprozessoren, die für Matrizenberechnung optimiert sind [26].

Für die Vorverarbeitung der verwendeten Daten wurde die Python-Bibliothek *scikit-learn* [27] eingebunden. Diese implementiert bereits viele Verfahren, die in verwandten Arbeiten üblicherweise Verwendung finden.

Weiterhin wurde die Python-Bibliothek *matplotlib* eingesetzt, um experimentelle Ergebnisse zu visualisieren. Die in dieser Arbeit abgebildeten Diagramme sind ebenfalls mit *matplotlib* generiert worden.

Um eine hinreichend performantes Vorhersagemodell zu entwickeln, sind in dieser Thesis zahlreiche Experimente durchgeführt worden. Um diese Experimente sowie deren Ergebnisse zu organisieren wurde die Bibliothek *Sacred*⁴ eingesetzt. Diese ermöglicht es, Konfiguration und Ergebnis eines Experiments in einer Datenbank abzulegen und zu verwalten. Um die Suche der Hyperparameter zu automatisieren, wurde *labwatch*⁵ eingesetzt, eine Erweiterung zu *Sacred*. Diese ermöglicht es, geeignete Hyperparameter in einem vordefinierten Suchraum anhand von wiederholten Experimenten zu finden und somit den Trainingsprozess der Modelle zu optimieren.

Ein Großteil der Experimente in dieser Arbeit ist unter Verwendung einer *Nvidia GTX1080Ti* mit 11GB Grafikspeicher und 3584 *CUDA* Cores sowie einem *Intel Core i7-8700K* durchgeführt worden.

Während dem Training der Modelle konnte beobachtet werden, dass der Einsatz einer GPU eine signifikante Beschleunigung des Trainingsprozesses gegenüber einer CPU bewirkt. Die Laufzeit des Trainingsprozesses variiert zudem stark in Abhängigkeit von der Architektur des Modells und den gewählten Hyperparametern. Eine gesamte Iteration des Trainingsprozess benötigt bei Verwendung eines DNN zwischen 15 Minuten und einer Stunde, bei der Verwendung eines LSTM erhöht sich diese Laufzeit auf 60 Minuten bis 6 Stunden.

Weiterhin wurde in einigen Experimenten ein GPU-Cluster eingesetzt, welches sich aus zehn *Nvidia Tesla P100* Grafikprozessoren zusammensetzt. *Tensorflow* ermöglicht dem Anwender, die Matrizenberechnungen im Trainingsprozess eines neuronalen Netzes auf mehrere GPUS zu verteilen, um den Prozess zu beschleunigen. Dieses Vorgehen konnte für

² <http://www.image-net.org/>

³ <http://www.fast.ai/2018/08/10/fastai-diu-imagenet/>

⁴ <https://github.com/IDSIA/sacred>

⁵ <https://github.com/automl/labwatch>

rekurrente Netze in dieser Arbeit jedoch keine Verbesserungen bezüglich der Laufzeit bewirken. Stattdessen wurden mehrere Modelle mit unterschiedlichen Konfigurationen gleichzeitig trainiert, sodass die effektive Laufzeit des Entwicklungsprozesses um den Faktor 10 reduziert werden konnte.

In Teilabschnitt 2.4.5 wurde bereits erläutert, dass die Konfiguration eines neuronalen Netzes enorm aufwändig ist. Aus diesem Grund wurden in dieser Arbeit sehr viele Modelle trainiert, um eine geeignete Kombination aus Modellparametern und Hyperparametern zu erhalten. Änderungen an diesen Parametern haben teilweise signifikanten Einfluss auf die Qualität der Prognose genommen. Der Prozess der Modelloptimierung ist maßgeblich durch *labwatch* unterstützt worden.

4 Auswertung

Im folgenden Kapitel werden die experimentellen Ergebnisse dieser Arbeit ausgewertet und beurteilt. Dazu werden zunächst die wesentlichen Entscheidungen erläutert, welche aus den Beobachtungen resultierten. Anschließend werden einige Probleme beschrieben, die in der experimentellen Phase aufgetreten sind und charakteristisch für Anwendungsfälle mit Echtzeiten sind. Abschließend erfolgt eine Beurteilung des Entwicklungsprozesses für diese konkrete Problemstellung. Dabei wird besonders die Berücksichtigung von domänenbezogenen Entscheidungen thematisiert.

4.1 Entwicklung des Vorhersagemodells

In Kapitel 3.4 sind mehrere Strategien zur Selektion geeigneter Merkmale beschrieben worden. Einige dieser Ansätze sind in dieser Arbeit experimentell evaluiert worden, die Ergebnisse dieser Untersuchung sind in Tabelle 2 dargestellt. Es wurde festgestellt, dass eine Beschränkung der verfügbaren Merkmale die Eingabedimension der Trainingsdaten reduziert und demnach zu kürzeren Trainingszeiten führt. Weiterhin konnte jedoch beobachtet werden, dass Vorhersagemodelle eine höhere Genauigkeit erreichen, wenn diese mit allen verfügbaren Merkmalen trainiert worden sind. Daraus ist zu entnehmen, dass die eingesetzten KNN durchaus in der Lage sind, als Featureextraktor zu fungieren und selbstständig eine angemessene Repräsentation der Trainingsdaten abzuleiten.

Weitere Maßnahmen, um die Dimension der Trainingsdaten zu verringern, wie etwa eine Principal Component Analysis, bewirken eine Beschleunigung des Trainingsprozesses. Allerdings verschlechtern diese Verfahren die Genauigkeit des Vorhersagemodells und eignen sich daher nicht für die in dieser Arbeit betrachtete Energieprognose.

Tabelle 2: Vergleich verschiedener Strategien zur Selektion von Merkmalen anhand der mittleren quadratischen Abweichung (MSE). Es ist jeweils ein LSTM-basiertes RNN mit einer LSTM-Schicht mit 1024 Neuronen bis zur Konvergenz bzw. max. 1000 Epochen trainiert worden. Die Daten zur Trainingsdauer beziehen sich auf den Versuchsaufbau aus Kapitel 3.7.

SELEKTIONSSTRATEGIE	FEATURES	TRAININGSDAUER	MSE
Ohne Vorauswahl	269	2h 13 min	43,277
Vorauswahl der Ingenieure	39	2h 23min	48,541
Principal Component Analysis	10	1h 44min	340,185
Random Forest	50	2h 28min	146,563

Aufgrund der Beschaffenheit des Datenbestands der *Neumayer III* wurde bereits im Vorfeld angenommen, dass rekurrente Netze sich besonders gut zur Entwicklung eines Prognosemodells eignen. Diese Annahme konnte in einer Vielzahl von Experimenten empirisch nachgewiesen werden. Tabelle 3 stellt die Resultate unterschiedlicher Modelltypen dar, welche in der Entwicklung dieser Thesis zum Einsatz gekommen sind. Darin ist zu beobachten, dass die besten Ergebnisse unter Verwendung von LSTM-basierten Netzen erzielt wurden. Die übrigen Verfahren erzielen zwar ebenfalls bessere Ergebnisse als die zuvor definierte Baseline und als ein Linearer Regressor, reichen aber nicht an die Qualität eines LSTM-basierten Modells heran. Dies deutet darauf hin, dass zeitliche Zusammenhänge und wiederkehrende Muster tatsächlich einen wesentlichen Einfluss auf den Energiebedarf nehmen. Weiterhin wurde beobachtet, dass rekurrente Netze, in denen GRU-Zellen zum Einsatz kommen, deutlich schneller konvergieren als solche mit LSTM-Zellen. Die Qualität der Prognosen ist allerdings etwas schwächer als in den LSTM-basierten RNN.

Tabelle 3: Vergleich verschiedener ML-Modelle anhand der mittleren quadratischen Abweichung. Die Modelle sind mit identischen Daten und Hyperparametern bis zur Konvergenz bzw. max. 1000 Epochen trainiert worden. Die Werte zur Dauer des Trainings beziehen sich auf den in Kapitel 3.7 beschriebenen Versuchsaufbau.

MODELLTYP	TRAININGSDAUER	MSE
Baseline	-	154,491
Linearer Regressor	1min 32s	115,971
Dense Neural Network	1min 42s	53,793
LSTM-basiertes RNN	2h 13min	43,277
GRU-basiertes RNN	56min	45,287

Die Konfiguration des Modells und dessen Hyperparameter sind in dieser Thesis experimentell bestimmt worden. Zu diesem Zweck sind etwa 1000 Experimente mit unterschiedlichen Konfigurationen durchgeführt worden. Diese Versuche haben gezeigt, dass Netzwerke mit einer LSTM-Schicht bereits ausreichend tief sind, um gute Prognosen zu generieren. Die Einbindung weiterer Schichten verlängert zwar den Trainingsprozess, bringt aber keine Vorteile bezüglich der Genauigkeit. Bei Netzwerken mit mehr als drei Schichten sind zudem deutlich schlechtere Prognosen zu beobachten. Dies ist vermutlich auf die Anzahl trainierbarer Parameter zurückzuführen. Dadurch brauchten komplexe Netzwerke mehr als 1000 Epochen, um zu einer angemessenen Lösung zu konvergieren.

In den Experimenten konnte weiterhin festgestellt werden, dass die Konfigurationen mit steigender Anzahl der Hidden Units in den rekurrenten Schichten zunehmend genauere Vorhersagen erzielt haben. Die Zahl der Neuronen hat demnach einen größeren Einfluss auf die Qualität eines trainierten Modells als die Anzahl der Schichten. Die optimale Konfiguration eines LSTM enthält nur eine einzige LSTM-Schicht mit 1024 Neuronen. Die Ergebnisse bezüglich einiger exemplarischer Konfiguration eines LSTM-basierten RNN sind

in Tabelle 4 dargestellt. Diese Modelle sind mit einer zyklischen Lernrate trainiert worden, die Lernrate wurde dabei zwischen 0,004 und 0,0002 variiert.

Tabelle 4.: Vergleich verschiedener RNN-Konfigurationen anhand der mittleren quadratischen Abweichung. Jedes dieser Modelle ist mit identischen Hyperparametern bis zur Konvergenz bzw. max. 1000 Epochen trainiert worden und enthält jeweils ein bidirektionales LSTM-Layer. Die besten Konfiguration ist hervorgehoben worden.

ANZAHL DER LAYER	ANZAHL DER HIDDEN UNITS	MSE
1	128	850,73
1	256	175,532
1	512	71,301
1	1024	43,277
2	128	224,019
2	256	61,958
2	512	58,019
3	128	147,698
3	256	58,742
3	512	49,9032

In den zuvor aufgeführten Modellen ist *Recurrent Dropout* und eine Regularisierung der trainierbaren Parameter eingesetzt worden, um dem Overfitting des Modells entgegenzuwirken. Die optimale Dropout-Rate wurde dabei experimentell auf 0,5 festgesetzt.

Die Länge des gleitenden Zeitfensters zur Sequentialisierung wurde ebenfalls experimentell auf 7 Tage bestimmt. Zuvor wurden die Datenpunkte der Sequenzen einem Downsampling unterzogen. Dazu wurde das arithmetische Mittel der Beobachtungen aus 15-minütigen Zeitfenstern berechnet. Eine gesamte Sequenz enthielt somit insgesamt 672 Zeitschritte. Während des Trainings konnte beobachtet werden, dass längere bzw. kürzere Sequenzen die Genauigkeit der Prognose teils signifikant beeinträchtigen.

In den Ergebnissen der besten Modellkonfigurationen ist zu beobachten, dass langfristige Trends, wie etwa monatliche Abweichungen, präzise vorhergesagt werden können (siehe Abbildung 7). Mittelfristige Trends, z.B der Tag-Nacht-Zyklus (siehe Abbildung 3) konnten bis auf wenige Ausnahmen ebenfalls zuverlässig abgebildet werden.

Aus diesen Beobachtungen lässt sich schließen, dass ein hinreichend genaues Prognosemodell entwickelt wurde. Ferner ist das Modell in der Lage, komplexe Zusammenhänge in den vorliegenden Daten, in Form von mittel- und langfristigen Trends abzubilden.

Allerdings treten im Energiebedarf mitunter extreme kurzzeitige Lastspitzen auf, welche von dem Modell nicht abgebildet werden konnten. Da der Einsatz von komplexeren Netzwerken jedoch keine Verbesserung bewirkt hat, ist die Ursache für die Abweichungen in kurzfristigen Prognosen vermutlich im Datenbestand zu finden. Es wird angenommen, dass extreme Lastspitzen des Energiebedarfs auf komplexe Zusammenhänge

zurückzuführen sind, welche in den verfügbaren Trainingsdaten nicht abgebildet sind und somit nicht durch ein neuronales Netz erlernt werden können.

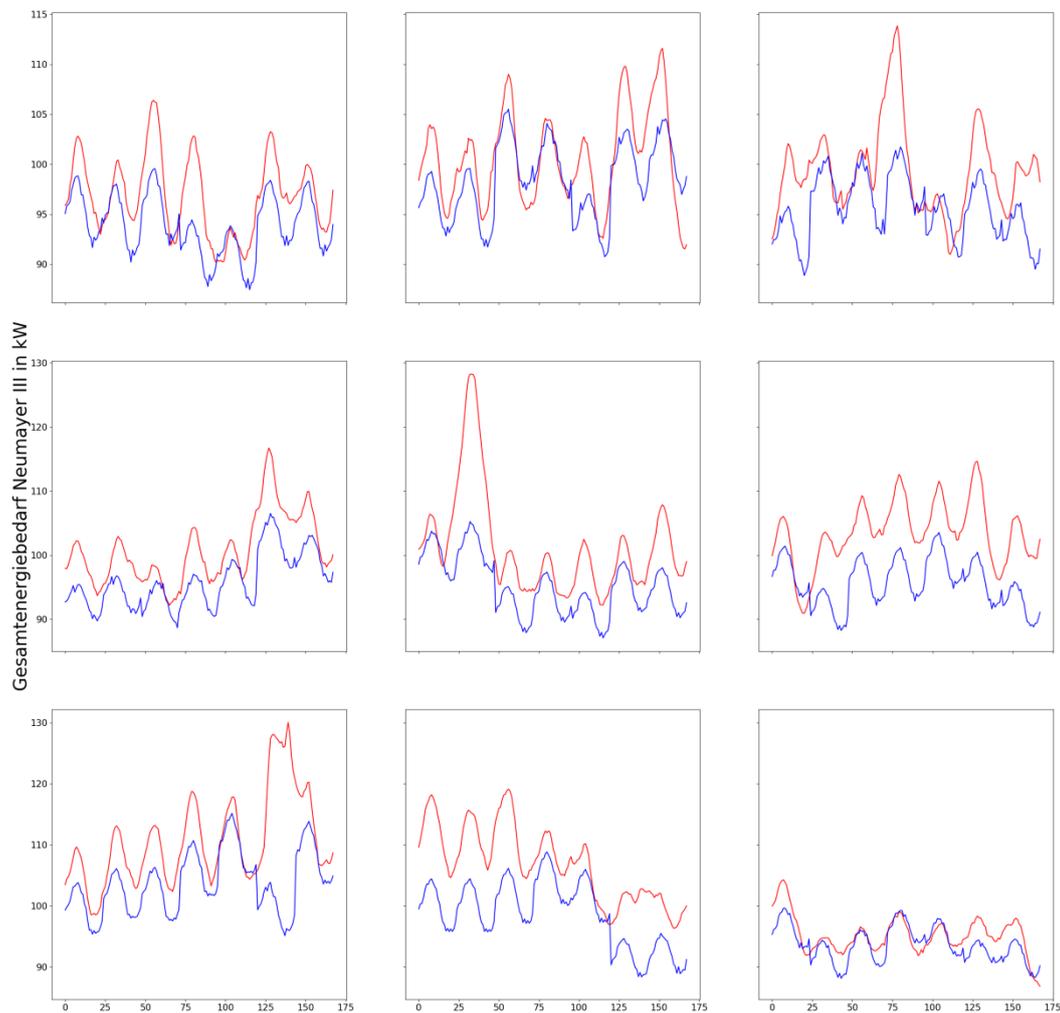


Abbildung 7: Prognosen eines LSTM-basierten RNN. Dargestellt sind prognostizierte und reale Werte des Gesamtenergiebedarfs über einen Zeitraum von jeweils einer Woche bzw. 176 Stunden.

4.2 Herausforderungen bei der Entwicklung intelligenter Vorhersagemodelle

Die Verwendung von realen Daten, insbesondere von multidimensionalen Zeitreihendaten, ist häufig mit einer Reihe von Herausforderungen verbunden. Auch in dieser Arbeit konnten einige Hindernisse identifiziert werden, welche im Wesentlichen auf den verwendeten

Datenbestand zurückzuführen sind. Diese werden im Folgenden erläutert, anschließend werden geeignete Lösungsansätze genannt.

Die Infrastruktur der Neumayer III unterliegt stetigen Veränderungen, welche sich wiederum auf die Verteilung der aufgezeichneten Daten auswirkt. Die zugrundeliegenden externer Ereignisse, werden in den Daten jedoch nicht erfasst. Dadurch sind die trainierten Modelle nicht in der Lage, diese Trends abzubilden und generieren möglicherweise schlechtere Prognosen. Besonders kritisch ist dieser Sachverhalt bei der Evaluation eines Modells. Da die Modelle ausschließlich auf historischen Daten bis Ende 2015 trainiert worden sind, konnten nachfolgende Trends nicht zuverlässig prognostiziert werden. In den Ergebnissen ist daher zu beobachten, dass die Genauigkeit des trainierten Vorhersagemodells für solche Sequenzen, welche am Ende des Trainingszeitraums liegen, abnimmt. Darunter leidet die Qualität eines Prognosemodells. Ein kürzerer Testzeitraum ist jedoch möglicherweise nicht repräsentativ für den Gesamtdatenbestand und ermöglicht somit keine angemessene Beurteilung der Modellqualität.

Eine weitere Herausforderung stellten fehlerhafte Sensorpunkte im Datenbestand dar (siehe Kapitel 3.3). Bei der Rekonstruktion der betroffenen Informationen gehen zwangsläufig einige Informationen verloren, sodass Zusammenhänge zwischen den einzelnen Kanälen verzerrt werden können und. Dies wirkt sich ebenfalls negativ auf die Qualität des entwickelten Modells aus. Zudem ist anzunehmen, dass trotz der ergriffenen Maßnahmen zur Datenbereinigung noch einige fehlerhafte Werte enthalten sind, welche im Rahmen dieser Arbeit nicht identifiziert werden konnten.

Zudem sind bei der während des Modelltrainings einige Schwierigkeiten aufgetreten. Simple Modelle neigen in zeitreihenbasierten Daten dazu, die vorangegangenen Werte der Zielvariablen vorherzusagen, da diese in den Trainingsdaten bereits enthalten sind. In einigen Modellen konnte außerdem festgestellt werden, dass diese bei der Prognose auf Berechnung eines Mittelwerts zurückgreifen. Diese beiden Prognosestrategien können jedoch keine neuartigen und somit auch keine interessanten Erkenntnisse generieren. In dieser Arbeit sind einige Modelle trainiert worden, welche bei einer ersten Betrachtung der Metriken MSE und MAE gute Prognosen erzielen konnten. Bei einer näheren Begutachtung konnte jedoch festgestellt werden, dass diese auf die zuvor genannten Strategien zurückgreifen und somit unbrauchbar sind.

Um eine fundierte Beurteilung der trainierten Modelle zu gewährleisten, wurden daher zusätzliche Ergebnisdiagramme in den Entwicklungsprozess integriert. Diese bildeten sowohl die prognostizierten Werte als auch die erwarteten, realen Beobachtungen ab. Ein entsprechendes Diagramm ist in Abbildung 7 dargestellt. Der direkte Vergleich von Realität und Prognose bietet tiefere Einblicke in die Ergebnisse der Algorithmen.

4.3 Bedeutung fachbezogener Intuition in der Entwicklung von DL-Systemen

Künstliche Intelligenz allgemein und intelligente Vorhersagesysteme im Speziellen stellen eine Möglichkeit dar, um Arbeitsprozesse, welche zuvor manuell durchgeführt worden sind, zu vereinfachen oder zu ersetzen. In Folge dessen wird häufig argumentiert, dass es gefährlich sei, die Intuition menschlicher Experten und deren Erfahrungswerte aus den Entscheidungsprozessen zu verbannen. Im Folgenden wird diese Fragestellung anhand der Erkenntnisse dieser Thesis diskutiert.

Für die Entwicklung eines DL-Verfahrens sind in dieser Arbeit einige Transferleistungen zwischen den fachbezogenen Aussagen der Ingenieure und der Anwendung von DL-Verfahren erforderlich gewesen. Der Datenbestand der Neumayer III ist in dieser Thesis zum ersten Mal im Kontext von DL-Verfahren genutzt worden. Dies stellte eine Herausforderung für die Modellentwicklung, aber auch für die Domänenexperten dar.

Die Konstruktion einer angemessenen und lösbaren Problemstellung hat in dieser Hinsicht die erste Herausforderung dargestellt. Dazu sind von den Domänenexperten zunächst potenzielle Anwendungsfälle identifiziert worden. Diese sind anschließend unter Berücksichtigung des aktuellen Forschungsstands auf ihre Lösbarkeit mit DL-Verfahren untersucht worden. Daraufhin wurde die Vorhersage des Gesamtenergiebedarfs als Problemstellung zur weiteren Bearbeitung ausgewählt. Diese Problemstellung ist zunächst auf rein fachlicher Ebene definiert worden. Dabei sind bereits technische Details und Hintergrundinformationen der zuständigen Ingenieure aufgenommen worden, welche schließlich in die Entwicklung des Vorhersagemodells eingeflossen sind.

Weiterhin sind die Domänenexperten an der Bereinigung des Datenbestandes beteiligt gewesen. In den aufgezeichneten Daten sind einige ungewöhnliche Muster identifiziert worden, die möglicherweise auf einen Fehler in der Aufzeichnung hinweisen. Da die Unterscheidung von fehlerhaften bzw. plausiblen Anomalien häufig Fachwissen voraussetzt, wurden in diesem Arbeitsschritt ebenfalls Domänenexperten eingebunden.

In Teilabschnitt 4.1 konnte gezeigt werden, dass Modelle, welche mit allen verfügbaren Merkmalen trainiert worden sind, die besten Ergebnisse erzielen konnten. Dies deutet darauf hin, dass der gesamte Datenbestand einige Zusammenhänge enthält, welche in fachlichen Betrachtungen nicht als relevant eingestuft worden sind. Die Ergebnisse des Modells eröffnen den Domänenexperten daher neue Erkenntnisse über das Verhalten der Forschungsstation und stellen somit eine wertvolle Ergänzung dar.

Weiterhin konnte im vorigen Teilabschnitt gezeigt werden, dass Fachkenntnisse erforderlich sind, um die Ergebnisse eines Modells angemessen beurteilen zu können. Dadurch lässt sich z.B. die Prognose eines Mittelwerts verhindern.

Die übrigen Arbeitsschritte, die Konstruktion sowie Konfiguration eines geeigneten Vorhersagemodells, sind hingegen vollkommen ohne Bezug zur fachlichen Domäne erfolgt. Dabei sind jedoch Erfahrungswerte und DL-spezifisches Fachwissen eingeflossen.

Aus den zuvor genannten Beobachtungen geht hervor, dass die Einbindung von Domänenexperten entgegen der anfangs aufgestellten These eine Voraussetzung für die

Entwicklung eines intelligenten Prognosemodells darstellt. Zwar sind einige Teilarbeitsschritte ohne fachlichen Bezug realisierbar, der Entwicklungsprozess profitiert jedoch immens von dem Austausch fachbezogener und DL-spezifischer Hintergrundinformationen. Die Existenzberechtigung von Spezialisten wird durch den Einsatz von DL-Verfahren folglich keineswegs aufgehoben. Im Gegenteil, die Kombination aus herkömmlichen Entscheidungsprozessen und intelligenten Systemen kann auch für die Domänenexperten einen signifikanten Erkenntnisgewinn darstellen.

4.4 Bewertung des Vorhersagemodells

In den vorangegangenen Kapiteln sind verschiedene Ansätze aus verwandten Arbeiten vorgestellt und angewandt worden, um eine Energieprognose für die Polarforschungsstation Neumayer III zu entwickeln.

Schließlich wurde ein LSTM-basiertes Regressionsmodell ausgewählt. Es konnte gezeigt werden, dass dieses Modell von allen getesteten Architekturen die besten Ergebnisse erzeugt. Zudem ermöglicht es eine hinreichend genaue Prognose für den vorliegenden Anwendungsfall.

Weiterhin wurde festgestellt, dass der zugrundeliegende Datenbestand der Neumayer III einige Fehler beinhaltet, welche die Entwicklung eines intelligenten Vorhersagemodells erschwerten. Es konnten jedoch geeignete Maßnahmen ergriffen werden, um den Datenbestand von Fehlern zu bereinigen und diesen für die Anwendung eines DL-Verfahrens aufzubereiten. Währenddessen wurde beobachtet, dass die korrekte Vorverarbeitung der Trainingsdaten maßgeblichen Einfluss auf die Genauigkeit der Vorhersage genommen hat.

Im Rahmen dieser Arbeit sind außerdem einige Möglichkeiten zur weiteren Optimierung des entwickelten Modells identifiziert worden, welche jedoch aus zeitlichen Gründen nicht mehr ausgeschöpft worden sind. Dies betrifft u.a. die Architektur sowie die Konfiguration des eingesetzten DL-Modells. Da diese zu großen Teilen experimentell und mit eingeschränkten Ressourcen bestimmt worden ist, sind die gewählten Parameter möglicherweise nicht optimal. Es ist daher anzunehmen, dass eine ausgiebige Parametersuche mit leistungsstärkerer Hardware bessere Konfigurationen generieren würde.

Ein weiteres Optimierungspotenzial stellt die Vorverarbeitung der Trainingsdaten für das Modell dar. Trotz der in Teilabschnitt 3.3 erwähnten Maßnahmen wurde vermutet, dass der vorliegende Datenbestand weitere Unreinheiten enthält. Diese Unreinheiten könnten durch eine detaillierte Datenanalyse in Zusammenarbeit mit den Domänenexperten erkannt und bereinigt werden. Ein solches Vorgehen verspricht weitere Verbesserungen in der Qualität des Vorhersagemodells.

In Teilabschnitt 4.1 wurde angenommen, dass die Abweichungen in der kurzfristigen Prognose des Energiebedarfs auf Zusammenhänge zurückzuführen sind, welche in den verfügbaren Trainingsdaten nicht abgebildet sind. In einer genaueren Betrachtung der Lastspitzen könnten weitere relevante Informationen gemeinsam mit den

Domänenexperten identifiziert werden und ggf. in den Datenbestand aufgenommen werden. Es ist anzunehmen, dass diese Informationen die Fähigkeit des Modells, kurzzeitige, scheinbar zufällige Fluktuationen des Energiebedarfs zuverlässig vorherzusagen, weiter verbessern würde.

5 Fazit

Das Ziel dieser Arbeit war die Konstruktion eines Prognosemodells anhand des Energiebedarfs der Neumayer III. Dazu wurden unterschiedliche Verfahren zur Verarbeitung von multidimensionalen und zeitreihenbasierten Sensordaten aus der Literatur vorgestellt. Anschließend ist in iterativer Vorgehensweise ein Deep-Learning-Verfahren entwickelt worden. Dabei ist der zugrundeliegende Datenbestand zunächst einer Folge von aufwändigen Vorverarbeitungsschritten unterzogen worden. Währenddessen konnte beobachtet werden, dass die Datenqualität maßgeblichen Einfluss auf die Qualität des Prognosemodells genommen hat.

Anschließend ist die Relevanz einzelner Datenmerkmale untersucht worden. Es konnte empirisch nachgewiesen werden, dass eine Einschränkung der verfügbaren Merkmale keine Vorteile bezüglich der Genauigkeit des Modells bewirkt. Ferner konnte gezeigt werden, dass neuronale Netze die Funktion eines Featureextraktors angemessen ausfüllen können. Während der Entwicklung der Modellarchitektur konnte beobachtet werden, dass die trainierten Modelle immens von der Berücksichtigung temporaler Abhängigkeiten profitieren, aus diesem Grund ist ein rekurrentes neuronales Netz ausgewählt worden. Insbesondere konnte gezeigt werden, dass Long-Short-Term-Memory-Networks eine geeignete Architektur darstellen, um zuverlässige Prognosen auf der Grundlage von hochdimensionalen Zeitreihendaten zu ermöglichen.

Es ist anzunehmen, dass sich die Erkenntnisse dieser Thesis auf andere, ähnlich charakterisierte Probleme aus industriellen Umgebungen übertragen lassen. Es sind jedoch einige Anpassungen vorgenommen worden, welche speziell auf den vorliegenden Datenbestand der Neumayer III und den Anwendungsfall Energieprognose abzielen. Dies betrifft einerseits Entscheidungen bezüglich der Datenaufbereitung, aber auch die Wahl sowie Konfiguration des Modells. Die Übertragbarkeit auf alternative Problemstellungen sollte daher in weiteren Untersuchungen evaluiert werden.

Zudem sind während des gesamten Entwicklungsprozesses immer wieder Empfehlungen von Domänenexperten berücksichtigt worden. In den vorangegangenen Kapiteln konnte gezeigt werden, dass der Transfer von fachbezogenen Hintergrundinformationen essentiell für die Entwicklung eines intelligenten Vorhersagemodells ist.

5.1 Ausblick

Die Experimente in dieser Arbeit konnten viele wertvolle Erkenntnisse zur Entwicklung eines intelligenten Vorhersagemodells hervorbringen. Diese Erkenntnisse dienen als Grundlage für weiterführende Schritte. Im Folgenden werden einige Optionen genannt, welche eine wertvolle Ergänzung zu dieser Arbeit darstellen.

Das Modell in dieser Arbeit ist für eine Energieprognose der Forschungsstation Neumayer III entwickelt worden. Um ein solches Modell produktiv einzusetzen, sind jedoch weitere Anpassungen erforderlich. Da sich Änderungen an der Infrastruktur der Forschungsstation auch auf die Verteilung der Daten auswirken können, sind historische Trainingsdaten möglicherweise nicht aussagekräftig, um neue Vorhersagen zu treffen. Um diesem Problem entgegenzuwirken könnte eine Integration der Daten, welche weiterhin von der Neumayer III aufgezeichnet werden, implementiert werden. Diese Daten könnten eingesetzt werden, um das Prognosemodell kontinuierlich an aktuelle Änderungen anzupassen.

Weiterhin könnte ein Metaprozess zur kontinuierlichen Adaption des Vorhersagemodells etabliert werden, um ggf. auftretende Trends zu relativieren und somit eine genauere Prognose zu ermöglichen. Einige interessante Ansätze zu dieser Thematik werden in [28] und in [29] vorgestellt.

Eine weitere wertvolle Erweiterung des entwickelten Modells würde die Einschätzung der Prognoseunsicherheit darstellen. In Teilabschnitt 3.6.2 sind diesbezüglich bereits erste Schritte eingeleitet worden. Die Verwendung von Monte-Carlo-Dropout hat jedoch keine zufriedenstellenden Ergebnisse bezüglich der Modellunsicherheit erzeugt, sodass im Anschluss an diese Arbeit dieser und ggf. alternative Ansätze weiter verfolgt werden sollten.

Die zuvor genannten Schritte ermöglichen einen produktiven Einsatz des entwickelten Prognosemodells auf der Neumayer III. Ein solches System wäre in der Lage, neuartige und nützliche Erkenntnisse bezüglich des Energiebedarfs zu generieren. Diese Erkenntnisse könnten weiterführend im Rahmen eines Energiemanagementsystems eingesetzt werden, um besagten Energieverbrauch zu optimieren. In einem solchen Szenario bietet sich die Implementation eines Metaprozesses an, welcher Empfehlungen aus den Prognosen des trainierten Modells generiert. Solche Empfehlungen können genutzt werden, um Diskrepanzen zwischen erzeugter und verbrauchter Energie zu minimieren. Dieses Verfahren würde sowohl aus ökologischer als auch aus ökonomischer Sicht eine wertvolle Ergänzung für die Stationsbewohner darstellen.

Literaturverzeichnis

- [1] D. Liu, Q. Chen, and K. Mori, "Time series forecasting method of building energy consumption using support vector regression," *Int. Conf. Inf. Autom.*, no. August, pp. 1628–1632, 2015.
- [2] R. Zhou, Y. Pan, Z. Huang, and Q. Wang, "Building Energy Use Prediction Using Time Series Analysis," in *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, 2013, pp. 309–313.
- [3] J. Gao and R. Jamidar, "Machine Learning Applications for Data Center Optimization," *Google White Pap.*, pp. 1–13, 2014.
- [4] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain," *Psychol. Rev.*, pp. 65–386, 1958.
- [5] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *CoRR*, vol. abs/1404.7, 2014.
- [6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Neurocomputing: Foundations of Research," J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA, USA: MIT Press, 1988, pp. 696–699.
- [7] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," 1991.
- [8] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-term Dependencies with Gradient Descent is Difficult," *Trans. Neur. Netw.*, vol. 5, no. 2, pp. 157–166, 1994.
- [9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [10] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *CoRR*, vol. abs/1412.3, 2014.
- [11] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using {RNN} Encoder-Decoder for Statistical Machine Translation," *CoRR*, vol. abs/1406.1, 2014.

-
- [12] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Advances in Knowledge Discovery and Data Mining," U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996, pp. 1–34.
- [13] P. Domingos, "A Few Useful Things to Know About Machine Learning," *Commun. ACM*, vol. 55, pp. 78–87, 2012.
- [14] L. Breiman, "Random Forests," 2001.
- [15] I. T. Jolliffe, "Principal Component Analysis, Second Edition," 2002.
- [16] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000.
- [17] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [19] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014.
- [20] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks."
- [21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
- [22] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, 2000, pp. 381–387.
- [23] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," Sep. 2014.
- [24] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [25] Y. Gal and Z. Ghahramani, "Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, pp.

1050–1059.

- [26] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *CoRR*, vol. abs/1603.0, 2016.
- [27] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
- [28] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, “Lifelong Learning with Dynamically Expandable Networks,” 2017.
- [29] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, “Online Deep Learning: Learning Deep Neural Networks on the Fly,” Nov. 2017.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 29. Oktober 2018

Lucas Anders