

BACHELORTHESIS
Muhamad Iqbal Anshori

Wettervorhersage durch maschinelles Lernen: Regenprognose in Hamburg mittels Regressionsanalyse und modernen Deep-Learning- Ansätzen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Muhamad Iqbal Anshori

Wettervorhersage durch maschinelles Lernen:
Regenprognose in Hamburg mittels Regressionsanalyse und
modernen Deep-Learning-Ansätzen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Neitzke
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 17. Februar 2022

Muhamad Iqbal Anshori

Thema der Arbeit

Wettervorhersage durch maschinelles Lernen: Regenprognose in Hamburg mittels Regressionsanalyse und modernen Deep-Learning-Ansätzen

Stichworte

Wettervorhersage, maschinelles Lernen, Regressionsanalyse, Zeitreihenanalyse, mehrschichtiges Lernen, lineare Regression, CNN, LSTM, CNN-LSTM, Transformer, überwachtetes Lernen.

Kurzzusammenfassung

In dieser Bachelorarbeit werden verschiedene moderne Maschinelles Lernen-Verfahren zur Vorhersage der Niederschlagshöhe in Hamburg untersucht und verglichen. Anhand der vorhergesagte Niederschlagshöhe wird ermittelt, ob es morgen regnet oder nicht. Dafür werden die Wetterdaten aus Deutsche Wetterdienst (DWD) abgerufen, analysiert und für das Experiment vorbereitet. In dem Experiment werden fünf verschiedene moderne maschinelles Lernen-Verfahren, und zwar lineare Regression, LSTM, eindimensionale CNN, CNN-LSTM und Transformer anhand der Wetterdaten trainiert und anschließend evaluiert. Anhand der Ergebnisse hat sich gezeigt, dass CNN-LSTM und Transformer eine gute Vorhersage bis 75% Genauigkeit liefern bei der Vorhersage, ob es morgen regnet oder nicht.

Muhamad Iqbal Anshori

Title of Thesis

Weather forecasting through machine learning: rainfall forecast in Hamburg using regression analysis and modern deep learning approaches.

Keywords

Weather forecasting, machine learning, regression analysis, time series analysis, deep learning, linear regression, CNN, LSTM, CNN-LSTM, Transformer, supervised learning.

Abstract

In this bachelor thesis, various modern machine learning methods for predicting the amount of precipitation in Hamburg are examined and compared. The predicted amount of precipitation is used to determine whether or not it will rain tomorrow. For this purpose, the weather data from the German Weather Service (DWD) is retrieved, analysed and prepared for the experiment. In the experiment, five different modern machine learning methods, namely linear regression, LSTM, one-dimensional CNN, CNN-LSTM and Transformer, are trained using the weather data and then evaluated. The results have shown that CNN-LSTM and Transformer provide a good prediction of up to 75% accuracy in predicting whether it will rain tomorrow or not.

Danksagung

Zunächst möchte ich mich bei Luna Pujianto und Sabina Dell von HAW Hamburg (Arbeitsstelle Migration) bedanken, die meine Arbeit Korrektur gelesen haben und mir durch konstruktive Kritik geholfen haben, diese Arbeit zu verbessern.

Ganz besonderer Dank gilt an dieser Stelle Prof. Dr. Michael Neitzke, der meine Bachelorarbeit betreut und begutachtet hat. Für die Betreuung und die hilfreichen Anregungen bei der Bearbeitung dieser Arbeit möchte ich mich herzlich bedanken.

Schließlich möchte ich mich auch bei meinen Freunden, Familien und insbesondere meinem Gott für unzählige Unterstützung während der Bearbeitung dieser Arbeit bedanken

Inhaltsverzeichnis

Inhaltsverzeichnis.....	vi
Abbildungsverzeichnis.....	x
Tabellenverzeichnis.....	xii
Abkürzungen	xiii
1 Einleitung.....	1
1.1 Motivation.....	1
1.2 Zielsetzung	2
1.3 Aufbau der Arbeit	3
2 Grundlagen.....	4
2.1 Zeitreihen	4
2.1.1 Zeitreihenanalyse	5
2.1.2 Multivariate Zeitreihen.....	5
2.1.3 Ein-Schritt-Prognose	6
2.1.4 Zeitfenster	7
2.2 Regressionsanalyse	7
2.2.1 Linear Regression.....	8
2.3 Deep Learning	10
2.3.1 LSTM	10
2.3.2 Eindimensionale CNN.....	11
2.3.3 CNN-LSTM	12
2.3.4 Transformer.....	13

vi

2.4	Statistische Metriken.....	14
2.4.1	Bestimmtheitsmaß.....	14
2.4.2	Mittlere quadratische Abweichung und Mittlere absolute Abweichung.....	14
2.4.3	Pearson-Korrelation.....	16
3	Datensatz.....	17
3.1	Niederschlag.....	17
3.1.1	Entstehung.....	17
3.1.2	Niederschlagsklassifikation.....	19
3.2	Wetterdaten.....	19
3.2.1	Beschreibung.....	19
3.2.2	Ausgewählte Merkmale.....	20
4	Entwurf.....	22
4.1	Überblick.....	22
4.2	Datenabruf.....	23
4.3	Vorverarbeitung.....	24
4.3.1	Merkmalextraktion.....	24
4.3.2	Identifizierung der Anomalie.....	24
4.3.3	Kodieren der kategorialen Variabler.....	25
4.4	Datenanalyse.....	25
4.4.1	Identifizierung der Ausreißer.....	25
4.4.2	Normale Datenverteilung.....	26
4.4.3	Standarisierung.....	27
4.4.4	In Zeitfenster transformieren.....	28
4.4.5	Zusammenfügen aller Datei.....	28
4.5	Hauptverarbeitung.....	29
4.5.1	Linear Regression.....	29
4.5.2	LSTM.....	30
4.5.3	Eindimensionale CNN.....	30
4.5.4	CNN-LSTM.....	30

4.5.5	Transformer.....	31
5	Implementation	32
5.1	Auswahl der Technologie.....	32
5.2	System.....	33
5.3	Datensatz.....	33
5.4	Vorverarbeitung	34
5.4.1	Merkmalextraktion.....	34
5.4.2	Fehlende Werte imputieren	35
5.4.3	Kodieren kategorialer Variable	35
5.5	Datenanalyse	35
5.5.1	Identifizierung der Ausreißer	35
5.5.2	Histogramm-Analyse	36
5.5.3	Transformieren und Standardisierung	37
5.5.4	Zusammenfügen.....	37
5.5.5	Training-Data und Evaluation-Data	38
5.6	Hauptverarbeitung.....	38
5.6.1	Linear Regression.....	38
5.7	Deep Learning.....	39
5.7.1	Vorverarbeitung	39
5.7.2	Verarbeitung.....	39
5.7.3	Evaluation während des Trainierens	40
5.8	Evaluation	41
6	Ergebnisse	42
6.1	Linear Regression.....	42
6.1.1	Ergebnis.....	42
6.1.2	Auswertung	43
6.2	LSTM.....	44
6.2.1	Architektur	44
6.2.2	Ergebnis.....	45

6.2.3	Auswertung	46
6.3	Eindimensionale CNN	47
6.3.1	Architektur	47
6.3.2	Ergebnis.....	48
6.3.3	Auswertung	49
6.4	CNN-LSTM	50
6.4.1	Architektur	50
6.4.2	Ergebnis.....	51
6.4.3	Auswertung	52
6.5	Transformer.....	53
6.5.1	Architektur	53
6.5.2	Ergebnis.....	54
6.5.3	Auswertung	55
7.	Fazit.....	57
7.1	Zusammenfassung.....	57
7.2	Ausblick	58
8.	Urheberhinweis	59
	Quellenverzeichnis	60
A.	Anhang	63
A. 1.	Benötigte Programmbibliotheken	63
A. 2.	Kodieren kategorialer Variable	64
A. 3.	Alle verwendeten Merkmale	65

Abbildungsverzeichnis

Abbildung 2. 1: Diskrete Zeitreihe.....	4
Abbildung 2. 2: Multivariate Zeitreihe [1].....	5
Abbildung 2. 3: Ein-Schritt-Prognose.....	6
Abbildung 2. 4: Transformieren einer Zeitreihe in Zeitfenster	7
Abbildung 2. 5: Beispiel einer linearen Regression [1] (Vergleich zwischen Verdienst des Restaurants und Anzahl der Gäste).....	8
Abbildung 2. 6: formale definierte lineare Regression [5].....	9
Abbildung 2. 7: Darstellung einer LSTM-Architektur [7].....	10
Abbildung 2. 8: Verarbeitung der Zeitreihen mithilfe eindimensionales CNN [1].....	11
Abbildung 2. 9: Beispiel-Architektur einer CNN-LSTM [8].....	12
Abbildung 2. 10: Transformer-Architektur [9]	13
Abbildung 3. 1: Wasserkreislauf.....	18
Abbildung 5. 1: Ergebnis der Identifizierung der Ausreißer aller Merkmale anhand Boxplot 35	
Abbildung 5. 2: Ergebnis der Histogramm-Analyse aller Merkmalen	36
Abbildung 5. 3: Histogramm-Analyse aller Merkmale nach dem Transformieren und Standarisierung.....	37
Abbildung 5. 4: Ergebnis der Korrelationsmatrix zwischen Niederschlagshöhe und allen anderen Einflussvariablen	38

Abbildung 6. 1: Erste 500 vorhergesagten Niederschlagshöhe in lineare Regression und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data.....	42
Abbildung 6. 2: Vorhersage-Beispiele der LSTM	45
Abbildung 6. 3: Erste 500 vorhergesagten Niederschlagshöhe in LSTM und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data.....	45
Abbildung 6. 4: Architektur eindimensionaler CNN	47
Abbildung 6. 5: Vorhersage-Beispiele der eindimensionalen konvolutionalen Schicht	48
Abbildung 6. 6: Erste 500 vorhergesagten Niederschlagshöhe in eindimensionaler konvolutionen Schicht und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data ...	48
Abbildung 6. 7: Architektur der CNN-LSTM.....	50
Abbildung 6. 8: Beispiel-Vorhersage der CNN-LSTM	51
Abbildung 6. 9: Erste 500 vorhergesagten Niederschlagshöhe in CNN-LSTM und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data	51
Abbildung 6. 10: Transformer-Architektur	53
Abbildung 6. 11: Beispiel-Vorhersage der Transformer.....	54
Abbildung 6. 12: Erste 500 vorhergesagten Niederschlagshöhe in Transformer und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data.....	55

Tabellenverzeichnis

Tabelle 3. 1: Klassifikation der Niederschlagshöhe.....	19
Tabelle 3. 2: Alle ausgewählten Merkmale.....	21
Tabelle 5. 1: Eigenschaften einzelner Datensatz.....	34
Tabelle 5. 2: Konfiguration für die Implementation in Deep-Learning-Verfahren.....	39
Tabelle 5. 3: Konfiguration für die Implementation in Transformer	40
Tabelle A. 1: Kodieren kategorialer Variable.....	64
Tabelle A. 2: Alle verwendeten Merkmale	65

Abkürzungen

DWD	Deutscher Wetterdienst
MSE	<i>mean squared error</i> (dt. mittlere quadratische Abweichung)
MAE	<i>mean absolut error</i> (dt. mittlere absolute Abweichung)
CNN	<i>convolutional neural network</i> (dt. Faltungsnetz)
LSTM	<i>Long Short Term Memory</i> (dt. langes Kurzzeitgedächtnis)
NWV-Modell	Numerisches Wettervorhersagemodell
CSV	<i>comma separated value</i>
ADAM	<i>adaptive moment estimation</i>
ReLU	<i>rectified linear unit</i>

1 Einleitung

1.1 Motivation

Das Wetter ist immer in aller Munde. Ob es regnet, trocken, warm oder kalt ist, unsere alltäglichen Aktivitäten, Flug- und Seeverkehr, Fischerei bis zur Landwirtschaft hängt stark von dem Wetter ab, weshalb eine gute Wettervorhersage uns immer wichtig ist.

Wettervorhersage ist das Ergebnis einer aufwändigen Prozesskette. Am Anfang des Prozesses stehen weltweite Beobachtungen der Atmosphäre und Ozeane. Diese Beobachtungen dienen in den komplexen numerischen Wettervorhersagemodellen (NWV-Modelle) zur Bestimmung des Anfangszustandes. Es wird dann mithilfe des NWV-Modells ausgehend von einem Anfangszustand eine zukünftige Wetterentwicklung berechnet.

Gleichzeitig maschinelles Lernen zeigt eine erfolgreiche Anwendung in der Bilderkennung, Spracherkennung, Robotik, strategischen Spielen und in vielen anderen Bereichen. Hierbei beschäftigen wir uns mit der Frage, ob maschinelles Lernen-Verfahren für Wettervorhersage insbesondere die Vorhersage der Niederschlagshöhe ein gutes Ergebnis liefern kann.

Die Motivation hinter dieser Arbeit besteht darin, eine Untersuchung verschiedener maschinellen Lernen-Verfahren durchzuführen, um die Niederschlagshöhe in Hamburg vorherzusagen. Hierbei werden die Wetterdaten in verschiedenen modernen maschinellen Verfahren vorbereitet, trainiert und evaluiert. Anhand der Ergebnisse kann festgestellt werden, ob maschinelles Lernen-Verfahren ein geeignetes Hilfsmittel ist, um das Wetter vorherzusagen.

1.2 Zielsetzung

Das Ziel dieser Bachelorarbeit ist, verschiedene moderne Maschinelles Lernen-Verfahren zur Vorhersage der Niederschlagshöhe in Hamburg zu untersuchen und vergleichen. Anhand der vorhergesagte Niederschlagshöhe wird ermittelt, ob in einer bestimmten Zeit regnet oder nicht.

In der Arbeit werden fünf moderne Maschinelles Lernen-Verfahren, und zwar lineare Regression, LSTM, Eindimensionale CNN, CNN-LSTM und Transformer untersucht und anschließend analysiert und diskutiert. Außerdem sind die umfangreichen Analysen der Wetterdaten und Vorverarbeitung der Daten Bestandteile dieser Arbeit.

Die Ergebnisse der Untersuchung werden geprüft und mit bestimmten statistischen Metriken gerechnet und analysiert. Mithilfe der Ergebnisse wird festgestellt, welches maschinelle Verfahren für die Vorhersage der Zeitreihen, insbesondere für Niederschlagshöhe, das beste Ergebnis liefert.

Ein weiteres Ziel dieser Arbeit ist die Entwicklung eines Verfahrens zur Vorverarbeitung der Wetterdaten. Dafür werden zuerst die Wetterdaten analysiert und anhand der Analyse wird festgestellt, welche Schritte nötig sind, um die Wetterdaten zu verarbeiten.

1.3 Aufbau der Arbeit

Kapitel 1 Einleitung Im ersten Kapitel werden die Motivation der Arbeit und die Zielsetzung dieser Arbeit geklärt.

Kapitel 2 Grundlagen Im zweiten Kapitel werden die wichtigsten Grundbegriffe der Zeitreihenprognose geklärt. Außerdem werden die genutzten Maschinelles Lernen-Verfahren bezüglich Zeitreihenprognose dargestellt.

Kapitel 3 Analyse Im dritten Kapitel werden der Niederschlag und dessen Klassifikation dargestellt. Außerdem wird die Entstehung der Niederschlag hier geklärt und analysiert. Anhand der Analyse wird ermittelt, welche Merkmale (engl. *features*) in dieser Arbeit benötigt sind, um die Niederschlagshöhe vorherzusagen.

Kapitel 4 Entwurf Im vierten Kapitel werden der Entwurf des Experimentes detailliert diskutiert. Hier werden detailliert auf alle benötigten Teilschritte von Vorverarbeitung bis Vorhersage der Wetterdaten eingegangen.

Kapitel 5 Implementation Dieses Kapitel wird mit der Realisierung des Experiments beschäftigt. Hierbei werden verschiedene genutzten Technologien dargestellt und begründet. Außerdem werden einzelne Komponente des Verfahrens bei der Implementation hier näher auseinandergesetzt.

Kapitel 6 Ergebnisse In diesem Kapitel werden die Ergebnisse des Experiments präsentiert und mit statistischen Metriken evaluiert. Anschließend werden die Ergebnisse diskutiert und ausgewertet.

Kapitel 7 Fazit Im letzten Kapitel werden die Arbeit zusammengefasst und die Erkenntnisse dargestellt. Darüber hinaus wird im Ausblick erläutert, inwiefern die Arbeit noch optimiert bzw. erweitert werden könnte, um bessere Ergebnisse zu erzielen.

2 Grundlagen

In diesem Kapitel werden die wichtigsten Grundbegriffe der Zeitreihenprognose geklärt. Außerdem werden die genutzten Maschinelle Lernen-Verfahren bezüglich Zeitreihenprognose dargestellt.

2.1 Zeitreihen

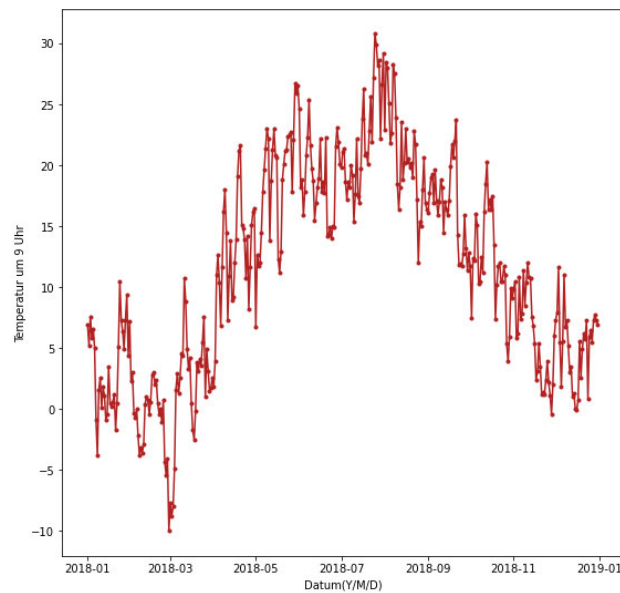


Abbildung 2. 1: Diskrete Zeitreihe

Zeitreihen sind Sammlungen von Beobachtungen, die sich über einen bestimmten Zeitraum erstrecken [1]. Bei den Zeitreihen handelt es sich um ein Strukturmerkmal von Daten, um eine Ordnung, in der Informationen vorliegen. Diese Ordnung der Daten ermöglicht es, die Veränderung des Untersuchungsgegenstandes (Satellit, Sensor, usw.) über die Zeitachse zu analysieren.

In dieser Arbeit werden ausschließlich diskrete Zeitreihen untersucht. Diskrete Zeitreihen sind Zeitreihen, in denen die Messwerte, die über einen Untersuchungsgegenstand gewonnen wurden, in regelmäßigen Intervallen organisiert sind – zum Beispiel in Sekunden-, Stunden oder Tagesintervallen [2].

2.1.1 Zeitreihenanalyse

In der Zeitreihenanalyse geht es darum, die Zeitreihen zu analysieren, und zwar, inwiefern ein zeitlich späterer Wert durch einen zeitlich früheren Wert beeinflusst wird [1]. Treten wiederholte Muster in Erscheinung, ergibt sich daraus, dass die Reihe einem bestimmten Regelsystem geregelt wird. Dessen Kenntnis kann dann dazu verwendet werden, die Historie einer spezifischen Reihe in die Zukunft hinein fortzuschreiben.

2.1.2 Multivariate Zeitreihen

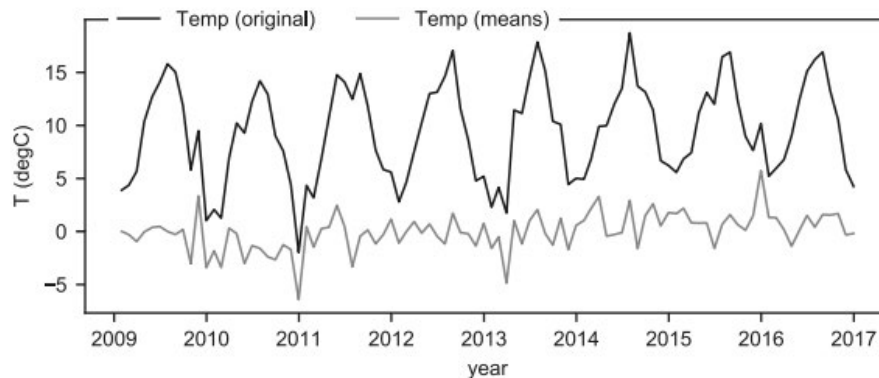


Abbildung 2. 2: Multivariate Zeitreihe [1]

In dieser Arbeit werden ausschließlich multivariate Zeitreihen untersucht. In einer multivariaten Zeitreihen werden zwei oder mehrere Messungen in einer Zeit beobachtet [3]. Jede

Messung hängt nicht nur von ihren vergangenen Werten ab, sondern ist auch von anderen Messungen abhängig. Diese Abhängigkeit wird für die Vorhersage zukünftiger Werte genutzt.

2.1.3 Ein-Schritt-Prognose

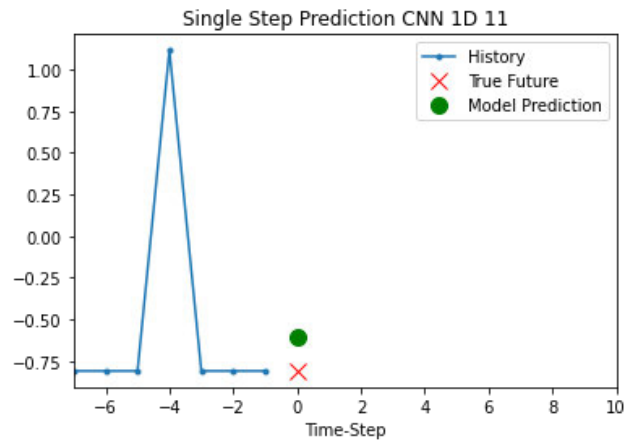


Abbildung 2. 3: Ein-Schritt-Prognose

Bei einer Ein-Schritt-Prognose (engl. *one step forecast*) in Zeitreihenanalyse wird ein nächster Zeitschritt vorhergesagt. Im Gegenteil: bei einem Mehrschritt-Prognose (engl. *multi step forecast*) werden mehrere Zeitschritte vorhergesagt [4]. In dieser Arbeit werden ausschließlich Ein-Schritt-Prognosen untersucht.

2.1.4 Zeitfenster

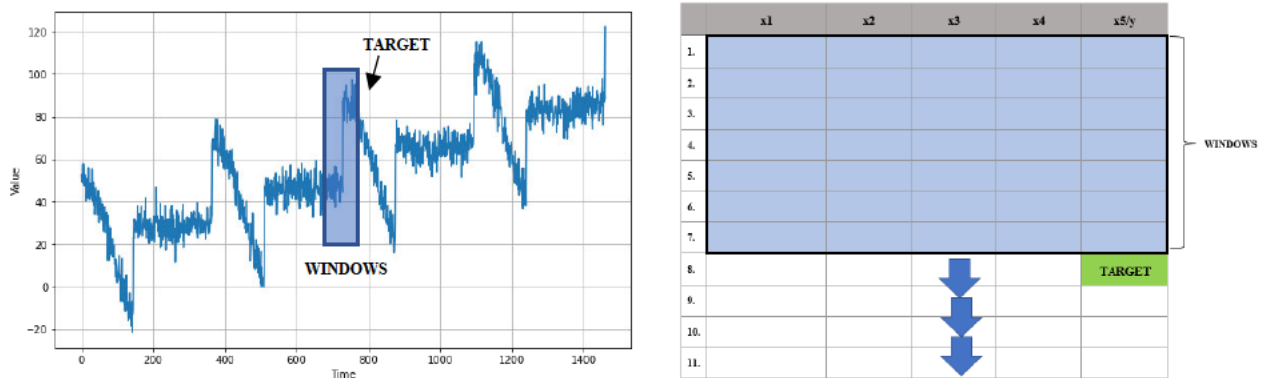


Abbildung 2. 4: Transformieren einer Zeitreihe in Zeitfenster

Um die Wetterdaten in einem Deep-Learning-Verfahren zu verarbeiten, müssen diese vorher in einem Zeitfenster transformiert werden. Unter dem Zeitfenster versteht man, dass die Zeitreihe in einer bestimmten Länge geschnitten werden soll [4], damit sie später in dem Deep-Learning-Verfahren analysiert werden kann.

Aus der Abbildung 2.4 (rechts) kann man sehen, dass die Zeitreihe mit der Länge 7 geschnitten wird. Dieses Fenster wird immer um Eins nach unten geschoben, bis man alle Daten fertig verarbeitet werden, daraus ergibt sich eine Sammlung der Zeitfenster mit der Länge 7.

2.2 Regressionsanalyse

Regressionsanalyse ist ein statisches Analyseverfahren, das zum Ziel hat, Beziehungen zwischen einer **abhängigen Variablen** (oft auch Regressand oder Zielvariable genannt) und einer oder mehreren **unabhängigen Variablen** (oft auch erklärende Variablen, Regressoren oder Einflussvariable genannt) zu modellieren [1, 5].

2.2.1 Linear Regression

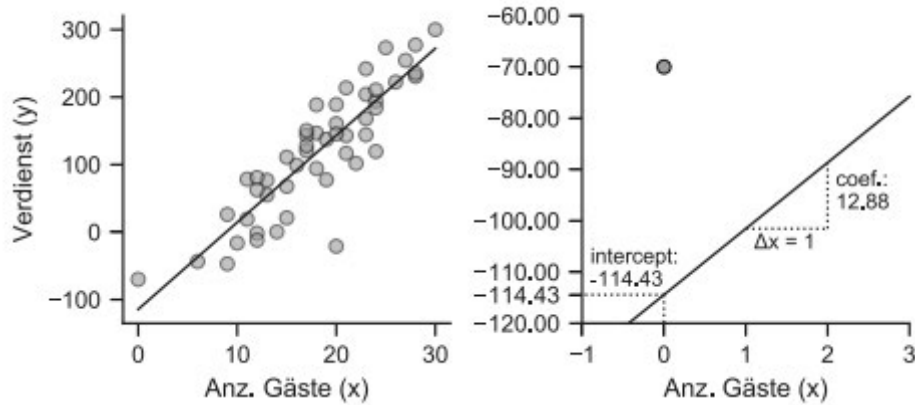


Abbildung 2. 5: Beispiel einer linearen Regression [1]
(Vergleich zwischen Verdienst des Restaurants und Anzahl der Gäste)

In linearer Regression wird versucht, lineare Beziehungen zwischen einer abhängigen Variablen und einer oder mehreren unabhängigen Variablen zu modellieren [5, 6]. Aus der Abbildung wird verdeutlicht, dass je mehr Anzahl der Gäste es gibt, desto mehr den Verdienst eines Restaurants. Hier kann man sagen, dass die beiden Variablen eine **positive lineare Beziehung** haben.

Es kann mehr als ein unabhängiger Variable gefordert werden. Außer der Anzahl der Gäste kann auch z.B. die Anzahl der Menüs, der Preis, oder die Anzahl der positiven Bewertung betrachtet werden. Wenn es mehr als eine unabhängige Variable gibt, dann wird als **multiple linear Regression** genannt [5]. Die lineare Beziehung könnte auch **negativ** sein, z.B. den Verdienst eines Restaurants und die Anzahl der negativen Bewertung.

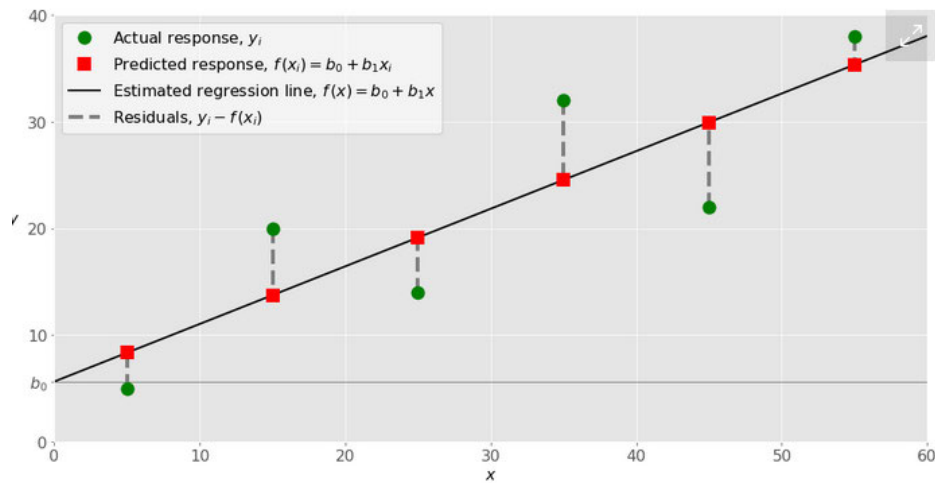


Abbildung 2. 6: formale definierte lineare Regression [5]

In Linear Regression wird angenommen, dass die abhängigen Variable und die unabhängige Variable eine lineare Beziehung haben. Daher wird hier versucht, eine passende Linie zu finden, die eine abhängigen Variable und alle unabhängigen Variablen berücksichtigen.

Linear Regression ist formal definiert wie folgt:

$$f(x_i) = b_0 + b_1x_1 + \dots + b_r x_r \quad (1)$$

- i : Anzahl der Beobachtung
- r : Anzahl der unabhängiger Variable
- b_0 : intercept
- $b_1 \dots b_r$: Regressionskoeffizient,
- $f(x_i)$: geschätzter Output
- y_i : aktuelle/tatsächlicher Output

Diese Formel sollte die linearen Abhängigkeiten zwischen die abhängigen Variable und die unabhängigen Variable möglichst gut erfassen [6]. Beim Trainieren werden die beste Regressionskoeffizienten, die kleinsten Residuen ($y_i - f(x_i)$) ergeben, gesucht und ermittelt.

2.3 Deep Learning

In diesem Abschnitt werden die in dieser Arbeit verwendete Deep-Learning-Verfahren geklärt, Die Erklärung wird auf die Verwendung des Deep-Learning-Verfahrens für Zeitreihe fokussiert.

2.3.1 LSTM

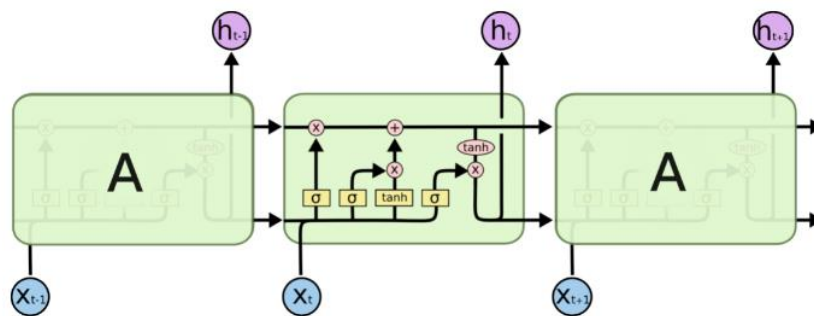


Abbildung 2. 7: Darstellung einer LSTM-Architektur [7]

LSTM (*Long-Short-Term-Memory*) ist eine bekannte Variante der rekurrente Netze und ist eine Kombination aus Lang- und Kurzzeitgedächtnis, in die wichtigen Messwerte memorisiert werden, unabhängig davon, ob sie am Anfang, in der Mitte oder am Ende einer langen Sequenz auftreten [1].

LSTM arbeitet bei Zeitreihenanalyse im Prinzip ähnlich wie bei der Verarbeitung des Textes. Es wird eine Folge von Zeitreihen (Zeitfenster) als Input übergeben, dann wird das Input anhand verschiedener Tore (engl. *gates*) analysiert und memorisiert bzw. verlernt.

2.3.2 Eindimensionale CNN

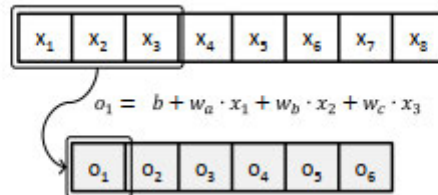


Abbildung 2. 8: Verarbeitung der Zeitreihen mithilfe eindimensionales CNN [1]

Konvolutionales Netz kann für Verarbeitung der Zeitreihen eingesetzt werden. Es arbeitet im Prinzip ähnlich wie bei der Verarbeitung der Bilder [1]. Aus der Abbildung oben wird verdeutlicht, wenn eine univariate Zeitreihe verarbeitet wird. Es wird als Länge der Zeitfenster 8 genommen, dann wird die Zeitfenster in einem eindimensionalen Array vorgelegt.

Über diesen Array wird dann ein Kernel mit zum Beispiel drei Gewichten, der einen Ausschnitt aus der Zeitreihe der Länge drei analysiert. Der Unterschied zu der Bildverarbeitung besteht darin, dass mit dem Kernel kein zweidimensionaler Ausschnitt aus einer Bildmatrix angelernt wird, sondern ein eindimensionaler Ausschnitt aus einer Zeitreihe. Dann wird für jeden Ausschnitt ein Schätzwert ergeben, der die in diesem Ausschnitt vorgefundenen Muster beschreibt.

Um verschiedene Verlaufsstrukturen wie Anstieg, Abfall, konstante Phase und Ausreißer zu identifizieren, wird in der Regel pro Ausschnitt mehr als einen Filter verwendet.

Dann kann wie bei der Bildverarbeitung ein Pooling-Schicht eingesetzt werden, um die Merkmale zu selektieren. Am Ende kann eine Dense-Schicht eingesetzt werden, um die finale Ausgabe zu produzieren.

Wie bei der Bildverarbeitung ist eindimensionale CNN ist für Identifikation der Verlaufsmuster geeignet. Wenn zum Beispiel der vorhergesagte Wert von den vergangenen Werten abhängig ist, dann sind rekurrente Netze hier besser geeignet.

2.3.3 CNN-LSTM

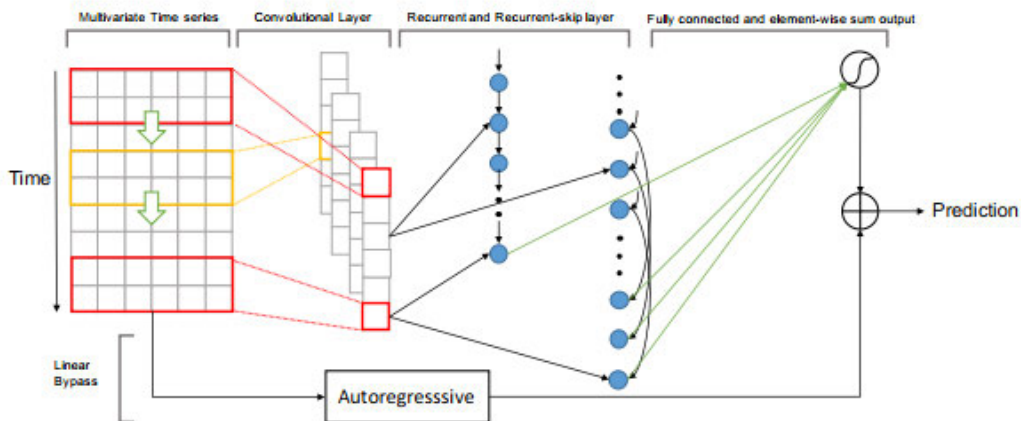


Abbildung 2. 9: Beispiel-Architektur einer CNN-LSTM [8]

CNN und LSTM werden in der Praxis manchmal auch kombiniert eingesetzt [8]. Dabei wird zuerst eine konvolutionale Schicht eingesetzt, die kleineren Ausschnitte des Zeitfensters im Hinblick auf bestimmte Verlaufsformen voruntersucht. Da die resultierende Response-Map nichts anderes als eine Sequenz ist, kann man sie einfach einer rekurrenten Schicht als Input übergeben. Die rekurrente Schicht analysiert dann den Ablauf, in der die voranalysierten Verlaufsformen nacheinander auftreten und produziert auf dieser Grundlage ein Schätzergebnis.

Hier werden die Vorteile beider Verfahren genutzt [1]: die des konvolutionalen Verfahrens, das auf die Identifikation von Teilmustern spezialisiert ist, sowie die des rekurrenten Verfahrens, das Informationen vor dem Hintergrund vorhergehender Informationen interpretieren kann.

2.3.4 Transformer

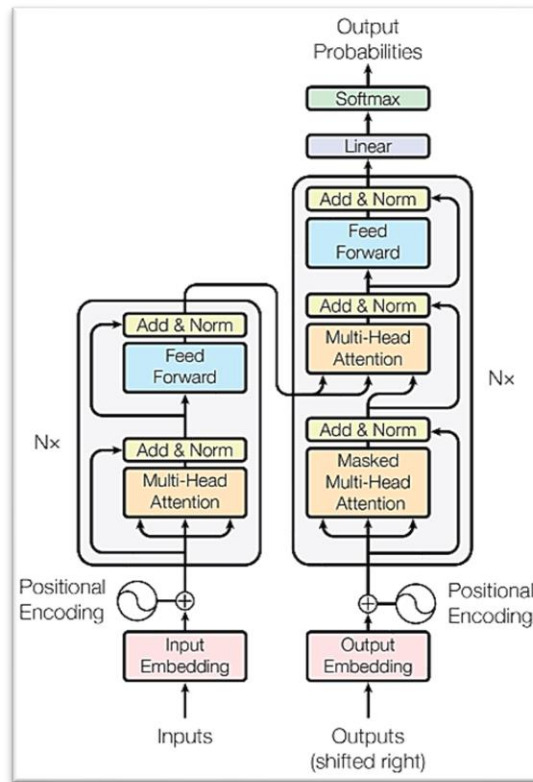


Abbildung 2. 10: Transformer-Architektur [9]

Transformer ist ein State-of-the-Art-Modell für Verarbeitung von Sequenz [9]. Es handelt sich um eine Encoder-Decoder-Architektur, deren Hauptmerkmal der "Multi-Head-Attention"-Mechanismus ist, der in der Lage ist, Abhängigkeiten innerhalb des Eingabevektors und innerhalb des Ausgabevektors (*self attention*) sowie Abhängigkeiten zwischen Eingabe- und Ausgabevektoren ("Encoder-Decoder-Attention") zu erkennen.

In dieser Arbeit wird mit der originalen Transformer-Architektur, die für Textverarbeitung verwendet wurde, versucht, die multivariate Zeitreihen vorherzusagen. Bei Zeitreihenanalyse arbeitet Transformer im Prinzip ähnlich wie bei der Textverarbeitung. Es wird eine Folge von Zeitreihen (Zeitfenster) als Input übergeben. Die Reihenfolge der Zeitreihen wird mithilfe

positional encoding geregelt, dann wird das Input anhand der "Multi-Head-Attention"-Mechanismus analysiert und memorisiert bzw. verlernt.

2.4 Statistische Metriken

In diesem Abschnitt werden die in dieser Arbeit genutzten statistische Metriken geklärt. Diese Metriken werden genutzt, um die Güte der Vorhersage später zu messen.

2.4.1 Bestimmtheitsmaß

Das Bestimmtheitsmaß (R^2) ist eine Metrik, die misst, wie gut das Ergebnis einer linearen Regression ist [10]. Das Bestimmtheitsmaß (R^2) gibt an, welcher Anteil der Variation von y durch die Abhängigkeit von \mathbf{x} unter Verwendung des jeweiligen Regressionsmodells erklärt werden kann. Ein größeres R^2 zeigt eine bessere Anpassung an.

Der Wert $R^2 = 1$ entspricht einer perfekten Anpassung, da die Werte der vorhergesagten und tatsächlichen Antworten vollständig übereinstimmen.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (2)$$

2.4.2 Mittlere quadratische Abweichung und Mittlere absolute Abweichung

Mittlere quadratische Abweichung (engl. *mean squared error (MSE)*) und mittlere absolute Abweichung (engl. *mean absolute error (MAE)*) sind Metriken, um das Ergebnis einer Vorhersage zu messen und zu bewerten [11–13].

Mittlere quadratische Abweichung und Mittlere absolute Abweichung sind zwei Hilfsmittel, um die Differenz zwischen zwei Vektoren zu messen: dem Vektor der Vorhersagen und dem Vektor der Zielwerte. Anders geschrieben wird in dieser Metrik das Error berechnet.

$$\text{MSE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^m (h(\mathbf{x}^{(i)}) - y^{(i)})^2 \quad (3)$$

$$\text{MAE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^m |(h(\mathbf{x}^{(i)}) - y^{(i)})| \quad (4)$$

m : Anzahl der Daten

$\mathbf{x}^{(i)}$: Vektor aller Merkmale in der Zeit i

\mathbf{X} : Eine Matrix, die alle Merkmale enthält.

$h^{(i)}$: Vorhersage des Modells in der Zeit i

$y^{(i)}$: tatsächlicher Wert in der Zeit i

In der mittleren quadratischen Abweichung wird das Error quadriert, damit wird ein großer Error auch groß bestraft. Diese Metrik ist optimal, wenn die Daten eine normale Datenverteilung und wenig Ausreißer verfügen [12].

In einer mittleren absoluten Abweichung wird der absolute Wert des Errors genommen und das Error nicht quadriert, damit wird ein großer Error auch nicht groß bestraft. Diese Metrik ist optimal, wenn es in dem Daten viele Ausreißer vorhanden sind, dann werden die Ausreißer nicht groß bestraft.

Sowohl in einer mittleren quadratischen Abweichung und in einer mittleren absoluten Abweichung entspricht der Wert 0 eine perfekte Vorhersage. da die Werte der vorhergesagten und die tatsächlichen Werten vollständig übereinstimmen.

2.4.3 Pearson-Korrelation

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}} \quad (5)$$

Die Pearson-Korrelation ist eine Möglichkeit, den linearen Zusammenhang zweier Variablen zu bestimmen [12, 13]. Der Korrelationskoeffizient reicht von -1 bis 1. Wenn er nahe bei 1 liegt, ist die Korrelation stark positiv. Wenn der Koeffizient nahe bei -1 liegt, besteht dann eine starke negative Korrelation. Wenn der Koeffizient nahe bei 0 liegt, bedeutet dies, dass keine Korrelation zwischen zwei Variablen besteht.

3 Datensatz

In diesem Kapitel wird der verwendete Datensatz erklärt, deren Eigenschaften und deren Analyse sind auch Bestandteil dieser Arbeit.

3.1 Niederschlag

Der Niederschlag wird in Wetterlexikon des Deutschen Wetterdienstes [14] folgendermaßen definiert:

“Unter dem Begriff Niederschlag versteht man in der Meteorologie die Ausscheidung von Wasser aus der Atmosphäre im flüssigen und/oder festen Aggregatzustand, die man am Erdboden messen oder beobachten kann. Dabei wird unterschieden zwischen fallenden (z.B. Regen), aufgewirbelten (z.B. Schneetreiben), abgelagerten (z.B. Schneedecke) und abgesetzten (z.B. Reif) Niederschlägen. Die fallenden Niederschläge sind definiert als das Ausscheiden von Wasser aus Wolken, das den Erdboden in flüssiger und/oder fester Form erreicht. „

3.1.1 Entstehung

In diesem Abschnitt wird die Entstehung der Niederschlag dargestellt [15]. Anhand der Analyse der Entstehung wird in nächsten Abschnitt entschieden, welche Merkmale für die Vorhersage der Niederschlagshöhe verwendet werden.

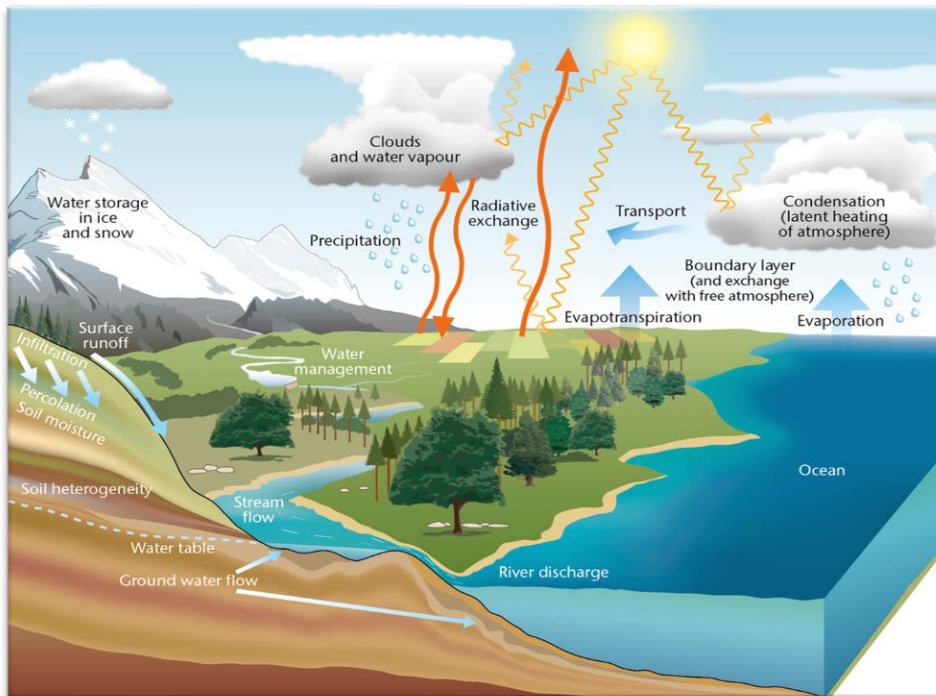


Abbildung 3. 1: Wasserkreislauf

Wasser an der Erdoberfläche, in Flüssen, Bächen, Ozeanen wird von der Sonne erwärmt. Es verdunstet dann - verwandelt sich von Flüssigkeit in Gas - und wird zu Wasserdampf, der in die Atmosphäre aufsteigt. Pflanzen und Bäume geben über ihre Blätter durch Transpiration Wasserdampf an die Atmosphäre ab. Außerdem kann Eis sich durch einen als Sublimation bekannten Prozess von fest direkt in Wasserdampf verwandeln.

Der gesamte Wasserdampf in der Atmosphäre wird schließlich gezwungen, von einem Gas zu einem flüssigen Wassertropfen zu kondensieren. In der Atmosphäre geschieht dies, wenn die Luft aufsteigt und abkühlt, wobei sich oft Wolken bilden.

Innerhalb einer Wolke werden sich die flüssigen Wassertröpfchen mit der Zeit wahrscheinlich zusammenschließen und größere Tröpfchen bilden. Irgendwann können diese Tröpfchen so groß werden, dass sie vom Himmel fallen und je nach den Bedingungen kann Hagel, Regen, Graupel oder Schnee aus den Wolken gefallen werden.

3.1.2 Niederschlagsklassifikation

Die Niederschlagshöhe wird in mm gemessen. Die Niederschlagsmessgeräte befinden sich in einer Höhe von 1 m über dem Erdboden. Eine Niederschlagshöhe von 1 mm entspricht einem Regenwasservolumen von 1 Liter pro Quadratmeter. In dieser Arbeit wird die Niederschlagshöhe in zwei Klassen klassifiziert [16].

Niederschlagshöhe	Klassifikation
< 1.0 mm	Kein Regen
>= 1.0 mm	Regen

Tabelle 3. 1: Klassifikation der Niederschlagshöhe

Mit dem Regen werden hier alle Ausscheidungsarten von Wasser gemeint, es könnte auch außer Regen, Schnee oder Reif sein.

3.2 Wetterdaten

In diesem Abschnitt werden die verwendete Wetterdaten beschrieben und die daraus ausgewählte Merkmale für diese Arbeit dargestellt

3.2.1 Beschreibung

In dieser Arbeit werden die Wetterdaten aus dem Deutschen Wetterdienst (DWD) genutzt. Genauer werden hauptsächlich die Daten aus der meteorologischen Station **Hamburg-Fuhlsbüttel** in der Zeit 01.01.1970-30.06.2021 (ca. 50 Jahre) verwendet, weil diese meteorologische Station das umfangreichste Angebot an Wetterdaten in Hamburg bietet.

3.2.2 Ausgewählte Merkmale

Nach der Analyse der Entstehung der Niederschlag aus dem Abschnitt 3.1.1 und der Analyse der vergangenen Forschung in Australien [16] wird entschieden, folgende Merkmale für diese Arbeit auszuwählen:

Merkmalsname	Einheit	Quelle	Zeit	Zeitliche Auflösung	Bemerkung
Niederschlag [17]	mm	DWD- Hamburg Fuhlsbüttel	01.01.1970- 30.06.2021	täglich	OK
Maximale Temperatur [18]	°C			täglich	OK
Minimale Temperatur [19]	°C			täglich	OK
Sonnenscheindauer [20]	Stunde			täglich	OK
Dampfdruck [21]	hPa			täglich	OK
Relative Feuchte [22]	%			täglich	OK
Taupunkttemperatur [23]	°C			stündlich	OK
Windrichtung [24]	Grad			stündlich	OK
Windgeschwindigkeit [25]	m/s			stündlich	OK

maximalen Windspitze [26]	m/s			taglich	OK
Luftdruck [27]	hPa			stundlich	OK
Wolkenbedeckung [28]	Achtel			stundlich	OK
Temperatur [29]	°C			stundlich	OK
Verdunstung	1/10 mm			taglich	nicht vorhanden

Tabelle 3. 2: Alle ausgewahlten Merkmale

Aus der Tabelle wird verdeutlicht, dass alle ausgewahlte Merkmale auer Verdunstung (engl. *evaporation*) in DWD vorhanden sind.

4 Entwurf

In diesem Kapitel wird der Entwurf des Experimentes dargestellt. Hierbei wird detailliert auf alle benötigten Teilschritte von Datenabruf bis Evaluation der Wetterdaten eingegangen.

4.1 Überblick

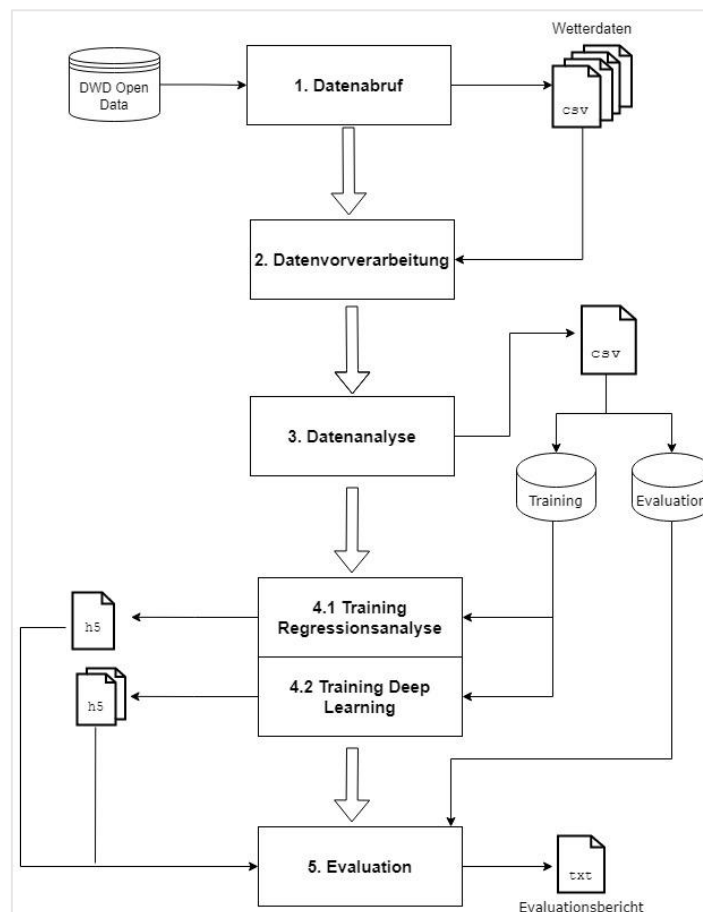


Abbildung 4. 1: Einzelner Arbeitsschritt des Verfahrens sowie entstehende Dateien

Um das Experiment besser nachzuvollziehen, zeigt Abbildung 4.1 die einzelnen Teilschritte sowie die benötigten Eingangsdaten.

Das Experiment gliedert sich in 4 Teilschritte. Zuerst werden die benötigte Wetterdaten aus dem DWD-Server abgerufen. Hierbei werden die abgerufenen Wetterdaten in der Datenvorbereitung für weitere Schritte aufbereitet. Als weitere Schritt kommt die Datenanalyse, hier werden die Daten anhand statistischer Hilfsmittel analysiert, um sie weiter zu optimieren.

Als Produkt der Datenvorbereitung und Datenanalyse entsteht der fertige Datensatz. Dieser Datensatz wird in Training-Daten und Evaluation-Daten aufgeteilt. Die Training-Daten werden sowohl im Training mit der Regressionsanalyse als auch in Training mit Deep Learning gebraucht. Die daraus entstehenden Modelle werden in Evaluation-Schritt genutzt, um die Vorhersage in Evaluation-Data zu evaluieren. Zum Schluss entsteht der Evaluationsbericht als das Ergebnis der Evaluation.

4.2 Datenabruf

In diesem Schritt werden die benötigten Wetterdaten der letzten 50 Jahre (01.01.1970 – 30.06.2021) abgerufen und heruntergeladen werden alle Merkmale aus dem Abschnitt Analyse. Jedes Merkmal hat eine eigene Datei.

Als zeitliche Auflösung wird es zwischen täglich, stündlich und 10-minütig unterschieden. Die meisten Merkmale haben täglich als zeitliche Auflösung. Einige Merkmale haben stündliche und 10-minütige zeitliche Auflösung, diese stündliche und 10-minütige Merkmale werden in der Vorverarbeitungsphase gebraucht, um weitere Merkmale zu extrahieren.

4.3 Vorverarbeitung

Ziel der Vorverarbeitung der Wetterdaten ist es, einen fertigen Datensatz zu erstellen, die für Eingabe im maschinelles-Lernen-Verfahren gebraucht werden.

Dafür werden die Wetterdaten, welche in CSV Format vorliegen, in ein geeignetes Eingabeformat transformiert. Außerdem werden die Anomalie in den Wetterdaten, wie fehlende Eintrag oder doppelte Eintrag identifiziert und behoben.

4.3.1 Merkmalextraktion

Ziel der Merkmalextraktion ist es, weitere Merkmale aus den vorhandenen Merkmalen zu extrahieren. Dieser Schritt ist erforderlich, damit wir eine bessere Repräsentation des Wetters an einem Tag erhalten.

Um weitere Merkmale zu extrahieren, werden die Merkmale, die als zeitliche Auflösung stündlich oder 10-minütig haben, genutzt. Für jedes Merkmal wird in einer 9 Uhr und 15 Uhr-Merkmal aufgeteilt. Daraus folgt, dass wir eine bessere Repräsentation des Merkmals in einem Tag haben.

Außerdem werden als zusätzliche Merkmale Tag, Monat, Jahr und Uhr extrahiert. Diese Merkmale werden gebraucht, um zeitlichen Einfluss später zu ermitteln.

4.3.2 Identifizierung der Anomalie

In diesem Schritt werden die Anomalien in den Wetterdaten identifiziert und behoben. Es wird hier geprüft, ob in den Wetterdaten fehlende Einträge (*engl, missing values*) oder doppelte Eintrag vorhanden sind.

Die fehlenden Einträge werden in diesem Schritt imputiert. Für Imputation wird der letzte gültige Eintrag genutzt, um die Lücke zu schließen. Falls der letzte gültige Eintrag nicht vorhanden ist, dann wird der nächste gültige Eintrag genutzt.

4.3.3 Kodieren der kategorialen Variabler

Die Richtung des Windes ist eine kategoriale Variable. Dieses Merkmal wird hier kodiert, damit es später im Maschinelles Lernen-Verfahren gebraucht werden kann.

4.4 Datenanalyse

Ziel der Datenanalyse ist es, statistische Eigenschaften der Wetterdaten zu erkennen. Die Analyse wird ermittelt, um sie weiter zu optimieren.

4.4.1 Identifizierung der Ausreißer

Ausreißer sind Stichproben, die außergewöhnlich weit vom Hauptstrom der Daten entfernt sind: Es könnte Inputfehler, Datenbeschädigung, oder sogar kein Fehler sein [2].

Um Ausreißer zu identifizieren, wurde der sogenannte Boxplot genutzt. Der Boxplot ist ein Diagramm, welches die übersichtliche Darstellung der Lage- und Streuungsmaße der Daten ermöglicht [4]. Es werden in Boxplot das Minimum, das untere Quartil, der Median, das obere Quartil und das Maximum abgebildet. Jedes Merkmal wird anhand Boxplot geplottet

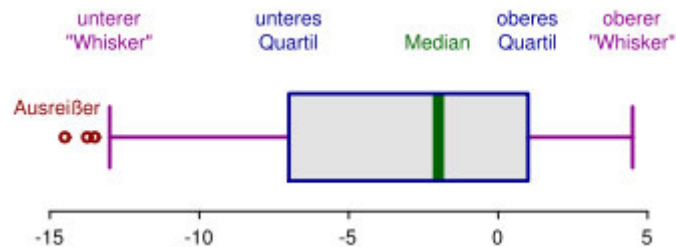


Abbildung 4. 2: Boxplot

Quelle: Wikimedia Commons¹

¹ Dieses Bild ist unter Creative-Commons-Lizenz lizenziert. Siehe: https://commons.wikimedia.org/wiki/File:Elements_of_a_boxplot.svg

Aus der Abbildung kann man sehen, dass die Ausreißer vor dem unteren „Whisker“ liegen. In der Praxis könnte auch die Ausreißer nach dem oberen „Whisker“ liegen.

4.4.2 Normale Datenverteilung

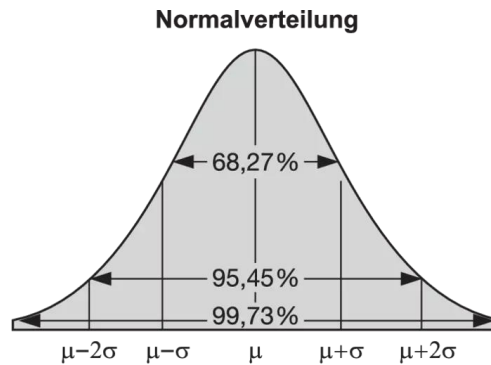


Abbildung 4. 3: Normale Datenverteilung [30]

Bei einer Normalverteilung sind die Daten symmetrisch und nicht schief verteilt [30]. In einem Diagramm sind die Daten glockenförmig dargestellt, wobei sich die meisten Werte um einen zentralen Bereich gruppieren und mit zunehmender Entfernung vom Zentrum abnehmen. Es wird gewünscht, dass die Daten über eine normale Verteilung verfügen. Dies hat zwei Vorteile, Erste ist einer schnelleren Konvergenz des Lernalgorithmus und Zweite ist einer Abmilderung anderer Konvergenzprobleme [31]. Hierbei werden die aktuellen Datenverteilungen der numerischen Merkmale dargestellt und ausgewertet, ob die Daten über eine normale Datenverteilung verfügen.

Power-Transform

In der Statistik ist Power-Transform eine Datentransformationstechnik, die zur Stabilisierung der Varianz führt, um die Daten an die Normalverteilung zu angleichen.

$$\psi(\lambda, y) = \begin{cases} ((y+1)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases} \quad (6)$$

In dieser Arbeit wird Yeo-Johnson-Power-Transform [32] genutzt. In diesem Power-Transform werden positive und negative Werte akzeptiert.

4.4.3 Standardisierung

Zur Optimierung des Trainings werden alle Merkmale standardisiert. In der Statistik ist die Standardisierung der Prozess, bei dem verschiedene Variablen auf dieselbe Skala gesetzt werden [1, 12]. Um Variablen zu standardisieren, werden in der Regel der Mittelwert und die Standardabweichung für eine Variable berechnet. Anschließend wird für jeden beobachteten Wert der Variable der Mittelwert abgezogen und durch die Standardabweichung geteilt.

$$z = (x - u) / s \tag{7}$$

4.4.4 In Zeitfenster transformieren

Es ist erforderlich, die Wetterdaten zu transformieren, bevor diese im Deep-Learning-Verfahren eingegeben werden. Die Wetterdaten sollte in genannten Zeitfenster transformiert werden.

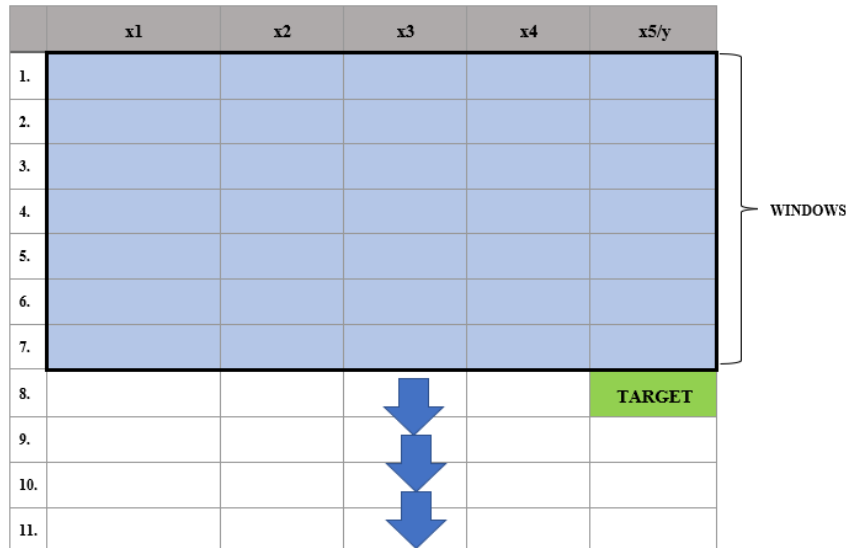


Abbildung 4. 4: Transformieren multivariater Zeitreihe in einer Zeitfenster

In diesem Zeitfenster wurde 7 oder eine Woche als Länge des Fensters (engl. *windows*) gewählt, also, dass die Merkmale die Merkmale in den 7 Tagen betrachtet werden, um den Regenfall in Hamburg für den nächsten Tag zu prognostizieren. Zur Evaluation werden die Niederschlagshöhe der 8. Tag als Bezeichnung (engl. *label*) gespeichert.

4.4.5 Zusammenfügen aller Datei

Am Ende werden alle Merkmale konkateniert, in eine CSV-Datensatz umgewandelt und bereit zu verwenden. Eine Liste der allen in dem Experiment genutzten Merkmale kann man in Anhang A.3 ansehen

4.5 Hauptverarbeitung

Nach der Vorverarbeitung sind die Daten bereit in maschinelles Verfahren verarbeitet zu werden. In diesem Abschnitt werden deshalb der in dieser Arbeit verwendeten Experiment-Ablauf der Hauptverarbeitung dargestellt und erläutert. Die Hauptverarbeitung besteht aus der Regressionsanalyse und dem Deep-Learning-Verfahren.

4.5.1 Linear Regression

Vorverarbeitung

Es ist eine zusätzliche Vorverarbeitung bei Linear Regression erforderlich. Im Kapitel 2 wurde geklärt, dass lineare Regression nur lineare Beziehungen erfasst, deshalb werden hier die Merkmale, die wenig oder gar keine linearen Beziehungen mit Niederschlag haben, entfernt. Um die lineare Beziehung zu messen, wird die sogenannte Pearson-Korrelation, die in dem Abschnitt 2.4.3 vorgestellt.

Verarbeitung

Mit folgender Formel wird die lineare Beziehung zwischen Niederschlagshöhe und allen Merkmalen ermittelt.

$$\begin{aligned} \mathbf{y}_{\text{precipitation}} = & \alpha + \beta_{\text{sunshinesum}} \cdot X_{\text{sunshinesum}} + \beta_{\text{mintemp}} \cdot X_{\text{mintemp}} + \beta_{\text{relativhum}} \cdot X_{\text{relativhum}} + \beta_{\text{vaporpres}} \cdot X_{\text{vaporpres}} + \beta_{\text{wingustMaxSpeed}} \cdot X_{\text{wingustMaxSpeed}} + \beta_{\text{windSpeed9am}} \cdot X_{\text{windSpeed9am}} + \beta_{\text{windSpeed3pm}} \cdot X_{\text{windSpeed3pm}} + \beta_{\text{airpressure9am}} \cdot X_{\text{airpressure9am}} + \beta_{\text{airpressure3pm}} \cdot X_{\text{airpressure3pm}} + \beta_{\text{cloudCov9am}} \cdot X_{\text{cloudCov9am}} + \beta_{\text{cloudCov3pm}} \cdot X_{\text{cloudCov3pm}} + \beta_{\text{dewTemperature}} \cdot X_{\text{dewTemperature}} + \beta_{\text{winDir9am}} \cdot X_{\text{winDir9am}} + \beta_{\text{winDir3pm}} \cdot X_{\text{winDir3pm}} \end{aligned} \quad (8)$$

Beim Trainieren wird der beste Koeffizient für jedes Merkmal gerechnet. Am Ende wird das entstehende Model für alle Input ausführbar sein.

4.5.2 LSTM

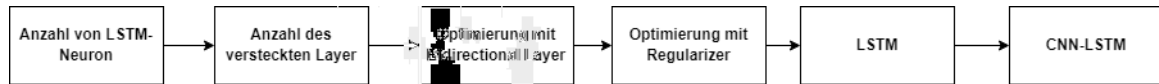


Abbildung 4. 5: Experiment-Ablauf der LSTM

Um die beste LSTM-Architektur zu finden, wird beim Experiment ein Ablauf geplant. Zuerst wird die beste Anzahl von LSTM-Neuronen gesucht. Dafür werden 4 LSTM mit folgender Anzahl von LSTM-Neuron 8, 32,64,100 trainiert und evaluiert. Danach wird die beste Anzahl der versteckten Schicht gesucht. Dafür wird das beste LSTM aus dem ersten Schritt genutzt und mit folgender Anzahl von versteckter Schicht 2,3,4,5 erweitert, trainiert und evaluiert. Danach wird das beste LSTM aus dem zweiten Schritt mit bidirektionaler Schicht und *regularizer* optimiert und wieder trainiert und evaluiert. Nach der Optimierung wird die beste LSTM-Architektur in dieser Arbeit für Niederschlagshöhe-Vorhersage entsteht.

4.5.3 Eindimensionale CNN

Im Rahmen dieser Arbeit werden mehrere eindimensionale CNN-Architekturen trainiert. Es wird vor dem Experiment ein Ablauf geplant. Hier wird wie bei dem LSTM-Experiment die beste Anzahl von eindimensionale CNN mit bestimmter Anzahl von Neuronen gesucht. Anschließend werden alle trainierten CNN-Architekturen evaluiert, um das beste Modell zu ermitteln.

4.5.4 CNN-LSTM

Für das Einbauen der CNN-LSTM-Architektur werden die beste LSTM-Architektur aus dem Abschnitt 4.5.2 und die beste eindimensionale CNN-Architektur aus dem Abschnitt 4.5.3 kombiniert, Außerdem wird für das Einbauen der CNN-LSTM-Architektur verschiedene Anzahl

der Schicht der eindimensionale CNN-Schicht untersucht und vergleicht, um optimale Architektur zu erzielen.

4.5.5 Transformer

In dieser Arbeit wird das originelle Transformer-Architektur [9] für das Experiment verwendet. Anhand dieses Ergebnisses wird entschieden, welche Hyperparameter angepasst werden soll, um besseres Ergebnis zu erzielen.

5 Implementation

Das vorgestellte Verfahren im Kapitel 4 wird in dieser Arbeit als Python-Programm durchgeführt. In diesem Kapitel wird zuerst die Auswahl der Technologie ausführlich erläutert und begründet. Danach wird die Implementation des Python-Programms geklärt.

5.1 Auswahl der Technologie

Im Rahmen dieser Arbeit wird das Experiment in *Google Colaboratory* durchgeführt, *Google Colaboratory* ist ein Forschungsprodukt von Google, das heutzutage für maschinelles Lernen und Datenanalyse hauptsächlich genutzt wird. In *Google Colaboratory* wird das Jupyter-Notebook-Dienst gehostet. Der Vorteil von *Google Colaboratory* besteht darin, dass es keine Einrichtung gefordert und gleichzeitig kostenlosen Zugang zu den Computerressourcen einschließlich GPUs angeboten werden. Außerdem ist *Google Colaboratory* kostenlos zu verwenden.

Das gesamte Experiment wird mithilfe des Python der Version 3.7.12 durchgeführt. Python ist eine universelle, interpretierte, höhere Programmiersprache, die im Bereich maschinelles Lernen weit verbreitet und eine große geeignete Bibliothek für maschinelles Lernen bietet.

Um die Vorverarbeitungsteil zu implementieren, wurde die Programmbibliothek von Python nämlich `pandas` und `numpy` genutzt. `pandas` ist ein Hilfsmittel, das für Verwaltung von Daten und deren Analyse bietet. Auch `numpy` wird verwendet, um die Daten zu verwalten und analysieren, zusätzlich dazu bietet `numpy` verschiedene Hilfsmittel für numerische Operation.

Die Hauptverarbeitungsteil wird mithilfe von `tensorflow`, `scikit-learn` und `unit8.darts` implementiert. `scikit-learn` und `tensorflow` sind Programmbibliotheken, die für maschinelles Lernen spezialisiert sind. Diese stellen viele Algorithmen des maschinellen Lernens zur Verfügung.

`scikit-learn` wird verwendet, um Regressionsanalyse zu implementieren und bewerten. Für Deep-Learning-Verfahren wird `tensorflow` und seine Schnittstelle `keras` verwendet. Mit dieser Programm-Bibliothek wird die Architektur der Deep-Learning-Verfahren relativ gut modelliert. `unit8.darts` ist Programm-Bibliothek, die für Vorhersage der Zeitreihen spezialisiert ist. Diese wird für das Einbauen der Architektur und das Experiment des Transformers verwendet. Auch wird die Programm-Bibliothek `seaborn` verwendet, um den Datenanalyseteil zu visualisieren.

Außerdem werden die `weights & bias` verwendet, um alle Logdateien des Trainierens und Evaluations und auch erzeugte Modelle zu überwachen, zu visualisieren und zu speichern.

5.2 System

Im Rahmen dieser Arbeit wird für jeden Arbeitsschritt ein eigenes Notebook erstellt. In den folgenden Abschnitten wird jeweils auf die Implementationsdetails der einzelnen Arbeitsschritte eingegangen.

5.3 Datensatz

Die Wetterdaten werden aus *DWD-Opendata* heruntergeladen und haben als Format CSV (*comma separated value*). Jedes Merkmal hat eigene CSV-Datei. Im Rahmen dieser Arbeit werden die täglichen Wetterdaten der letzten 50 Jahre heruntergeladen. Jede Datensatz beinhaltet mehrere Eigenschaften, die nicht alle benötigt werden.

Bezeichnung	Beschreibung
Produkt_Code	ID für das heruntergeladene Merkmal
SDO_ID	ID für das meteorologische Station
Zeitstempel	Das Datum des Wertes
Wert	Der Wert
Qualitaet_byte	Zeigt an, ob der Wert beanstandet, und/oder korrigiert wurde

Qualitaet_ niveau	beschreibt das Verfahren der Qualitäts- prüfung
----------------------	--

Tabelle 5. 1: Eigenschaften einzelner Datensatz

Die Eigenschaften `Produkt_Code`, `SDO_ID`, `Qualitaet_byte` und `Qualitaet_niveau` werden nicht genutzt, deshalb werden sie im nächsten Arbeitsschritt entfernt.

5.4 Vorverarbeitung

Das Notebook „*PreprocessingData.ipynb*“ dient zur Vorverarbeitung der Wetterdaten. Das Laden der Wetterdaten sowie das Verwalten und die Analyse der Wetterdaten ist Bestandteil dieses Notebooks.

5.4.1 Merkmalextraktion

Aus den stündlichen Merkmale `winddir`, `windspeed`, `airpressure`, `cloudcoverage`, und `temperature` werden weitere Merkmale extrahiert. Jedes Merkmal wird als 9 Uhr-Merkmal und 15 Uhr-Merkmal extrahiert. Dazu wird das Zeitstempel des Merkmals genutzt. Es wird hier allerdings festgestellt, dass fehlende Werte bei einige Merkmale um 9 Uhr oder um 15 Uhr vorhanden sind. Um die Lücke zu schließen, werden die Merkmale um 8 Uhr oder um 10 Uhr genutzt, um das fehlende Merkmal um 9 Uhr zu ersetzen. Die Merkmale um 14 Uhr oder um 16 Uhr werden auch genutzt, um das fehlende Merkmal um 15 Uhr zu ersetzen.

Außerdem werden als zusätzliche Merkmale `day`, `month`, `year` und `time` aus dem Zeitstempel extrahiert. Diese Merkmale sind erforderlich, um den zeitlichen Einfluss später zu ermitteln.

5.4.2 Fehlende Werte imputieren

In diesem Schritt werden fehlende Werte in allen Merkmalen imputiert. Es wird hier der letzte vorhandene Wert genutzt, um den fehlenden Wert zu imputieren. Dazu wird die Funktion `fillna` von `pandas` mit dem Parameter `method="pad"` genutzt.

5.4.3 Kodieren kategorialer Variable

Um die kategoriale Variable `winddir` zu kodieren, wird `LabelEncoder` von `sklearn` genutzt. Es ist eine Methode, um die kategoriale Variable mit einer bestimmten Zahl zu ersetzen. Die komplette Konvertierung zur numerischen Nummer kann man in A.2 ansehen.

5.5 Datenanalyse

5.5.1 Identifizierung der Ausreißer

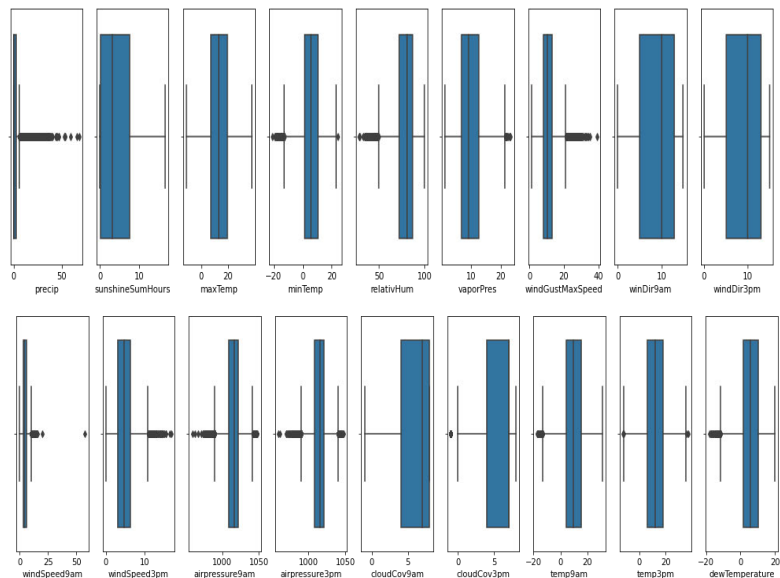


Abbildung 5. 1: Ergebnis der Identifizierung der Ausreißer aller Merkmale anhand Boxplot

Aus dem Ergebnis kann man deutlich sehen, dass Ausreißer in fast allen Merkmalen vorhanden sind. Bei Regenfall ist die Anzahl der Ausreißer deutlich höher als bei den anderen. Dies wird

hier entfernt, da diese Werte Geräuschdaten in den Daten sind. Anhand dieser Analyse wird entschieden, dass als Metrik die mittlere absolute Abweichung (engl. MAE) genutzt wird, da die mittlere absolute Abweichung für die Daten mit vielen Ausreißern mehr geeignet ist [12].

5.5.2 Histogramm-Analyse

Für die Histogramm-Analyse wird *distplot* von *seaborn* genutzt. Die Datenaufteilung jedes numerischen Merkmals wird anhand dieser Hilfsmittel angezeigt. Die folgende Abbildung zeigt das Ergebnis der Datenaufteilung.

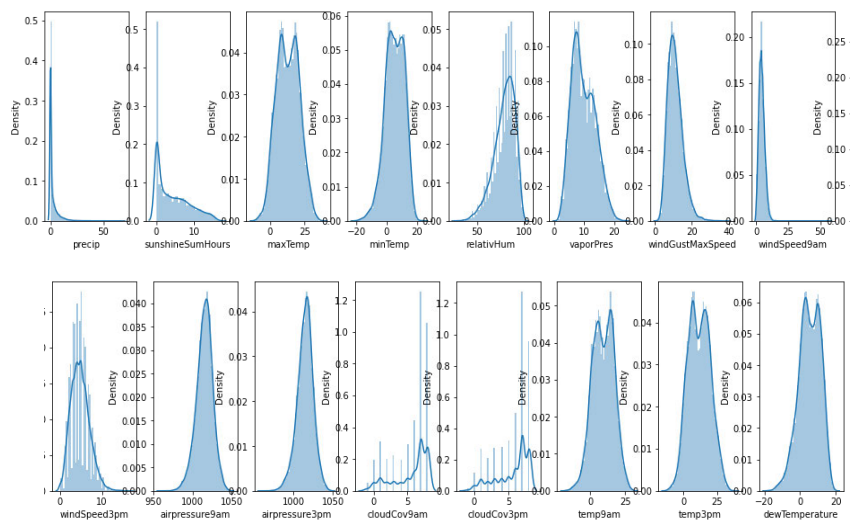


Abbildung 5. 2: Ergebnis der Histogramm-Analyse aller Merkmalen

Aus der Abbildung kann man sehen, dass das Merkmal “Luftdruck“ (engl. *air pressure*) über eine normale Verteilung verfügen. Die anderen Merkmale haben fast oder gar keine normale Verteilung. Um diese zu fixieren, werden alle Merkmale in nächsten Schritt mit der sogenannten Power-Transform in einer eher normalen Verteilung transformiert.

5.5.3 Transformieren und Standardisierung

Um die Merkmale in eine normale Datenverteilung zu bringen, werden die Merkmale transformiert. Dazu wird das `PowerTransformer` von `scikit-learn` genutzt. Zur Standardisierung wird keine zusätzliche Funktion gebraucht. Nach dem Transformieren wird standardmäßig eine Standardisierung durchgeführt.

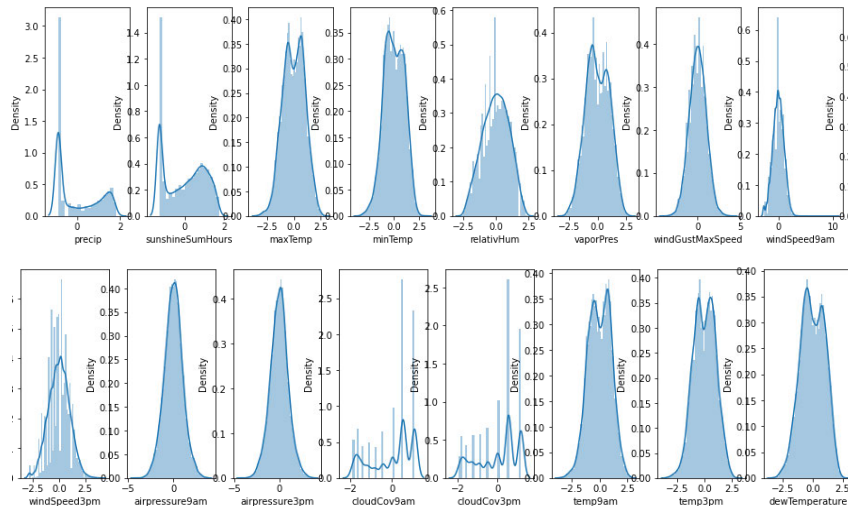


Abbildung 5. 3: Histogramm-Analyse aller Merkmale nach dem Transformieren und Standardisierung

Aus der Histogramm-Analyse wird verdeutlicht, dass mehrere Merkmale schon eine normale Datenverteilung haben. Außerdem liegen jetzt alle Merkmale wegen Standardisierung in einer gemeinsamen Einheit.

5.5.4 Zusammenfügen

Nach der Vorverarbeitung werden alle Merkmale in einer `dataframe` von `pandas` zusammengefügt. Als Index wird ein Zeitstempel ausgewählt, um ein einheitliches Daten zu haben. Am Ende wird das `dataframe` zu einer CSV-Datei umgewandelt, damit die Datei im Verarbeitungsteil wieder genutzt wird.

5.5.5 Training-Data und Evaluation-Data

Die Daten werden in einer Training-Data und Evaluation-Data mit dem Ratio 70:30 aufgeteilt. Daraus ergibt sich 15.047 Datensätze für die Training-Data und 3.762 Datensätze für die Evaluation-Data. Die Reihenfolge der Datensätze wird nicht geändert, da beim Trainieren und Evaluieren bereits auf die Reihenfolge geachtet wird.

5.6 Hauptverarbeitung

5.6.1 Linear Regression

Vorverarbeitung

Nach dem Laden wird die lineare Beziehung jeder Einflussvariablen zur Zielvariable ermittelt. Dafür wird die Methode `corr()` von `pandas` genutzt. Um das Ergebnis darzustellen, wird die Funktion von `seaborn` genutzt.

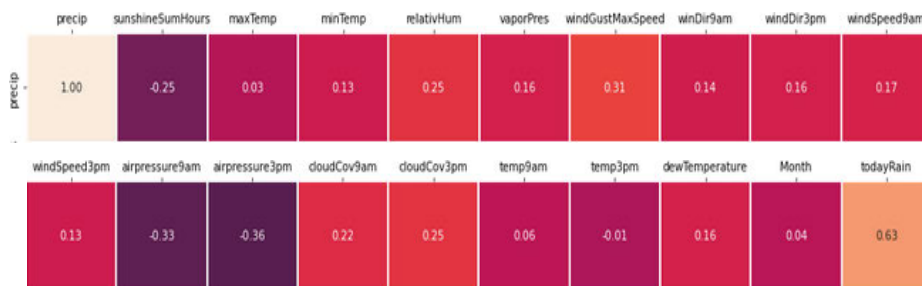


Abbildung 5. 4: Ergebnis der Korrelationsmatrix zwischen Niederschlagshöhe und allen anderen Einflussvariablen

Aus dem Korrelationsmatrix wird verdeutlicht, dass `month`, `temp 3pm`, `temp9am` nur sehr schwache lineare Beziehung mit Niederschlagshöhe haben, deshalb werden diese Merkmale in der Verarbeitungsteil nicht genutzt.

Verarbeitung

Um die lineare Regression in diesen Daten durchzuführen, wird `LinearRegression` von `skit-learn` genutzt. Mit dieser Funktion werden die Trainingsdaten trainiert.

5.7 Deep Learning

5.7.1 Vorverarbeitung

Es wird hier die Funktion `restructure_data` aus `numpy` verwendet, um die Daten in einer gewünschten Struktur zu aufzuteilen. Als Länge der Fenster wird `7` angegeben. Als Daten `training_data` und `evaluation_data` werden hier gemeint.

5.7.2 Verarbeitung

Bei der Verarbeitung wird für das Einbauen der Architektur LSTM und CNN `tensorflow.keras` verwendet. Für alle Verarbeitungsteile in Deep-Learning-Verfahren wird folgende Konfiguration verwendet:

Eigenschaften	Wert
<code>learning_rate</code>	<code>1e-8</code>
<code>optimizer</code>	<code>adam</code>
<code>loss</code>	<code>mae</code>
<code>monitor</code>	<code>val_loss</code>
<code>activation</code>	<code>ReLU</code>
<code>epochs</code>	<code>500</code>
<code>batch_size</code>	<code>32</code>

Tabelle 5. 2: Konfiguration für die Implementation in Deep-Learning-Verfahren

Bei jedem Trainieren wird die sogenannte „early_stopping“ und „model_checkpoint“ verwendet. Mit „early_stopping“ wird das Trainieren möglichst früh beendet, wenn es innerhalb 20 Epochs keine Verbesserung in der Verlustfunktion gibt, dann wird mit Hilfe von „model_checkpoint“ nur das beste Modell gespeichert.

Für das Einbauen der Transformer-Architektur wird *unit8.darts* verwendet. Darüber hinaus wird folgende Konfiguration für das Trainieren der Transformer zusätzlich verwendet:

Eigenschaften	Wert
epochs	100
dropout	0.3

Tabelle 5. 3: Konfiguration für die Implementation in Transformer

5.7.3 Evaluation während des Trainierens

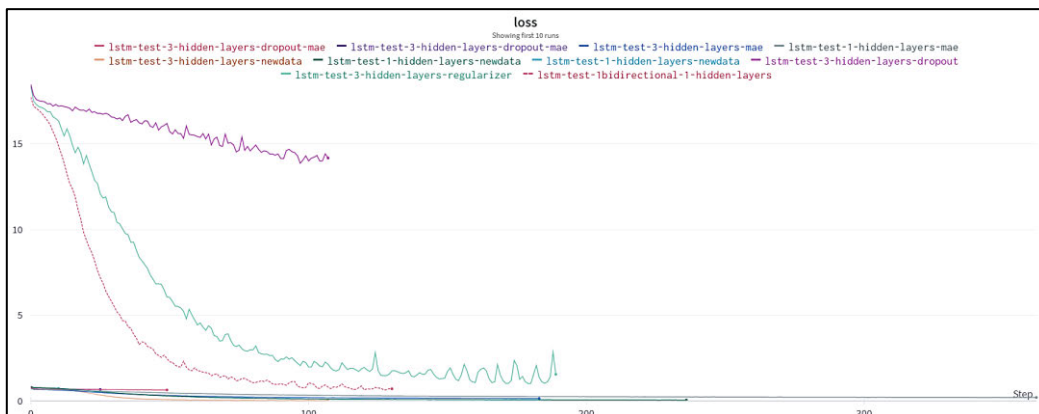


Abbildung 4. 6: Ergebnis der Verlustfunktion des LSTM-Experiments in weight&bias

Zur Evaluierung während des Trainierens wird *Weight&Biases* genutzt. Mithilfe dieses Hilfsmittels werden die Modelle parallel mit dem Trainieren evaluiert. Während des Trainierens wird das Modell mit den statistischen Metriken nach jedem Epoch evaluiert. Alle

Modelle und deren Evaluierung sind in `Weight&Biases` gespeichert, sowie alle Logdateien des Trainierens. Anhand der Evaluierung wird das beste Modell für jede Forschungsmethode ermittelt.

5.8 Evaluation

Das Notebook „Evaluation-Precipitation Forecasting with ML.ipynb“ dient zur Evaluation aller entstehender Modelle. Hier werden alle Modelle mit `evaluation-data` als Input übergeben. Anschließend wird die Vorhersage des Modells mit statistischer Metrik nämlich mittlerer absoluter Abweichung (engl. *MAE*) evaluiert.

6 Ergebnisse

6.1 Linear Regression

6.1.1 Ergebnis

Linear Regression hat kein Bezug auf Sequenz oder Zeitreihen. Die Reihenfolge wird hier nicht beachtet, deshalb ist das Transformieren der Daten zu Sequenz nicht benötigt.

Aus der Formel (8) werden die besten Koeffizienten während des Trainierens gesucht. Nach dem Trainieren wird das Modell evaluiert. Bei der Evaluation wird das trainierte Modell in Evaluation-Datensatz evaluiert und daraus ergeben sich folgende Ergebnisse:

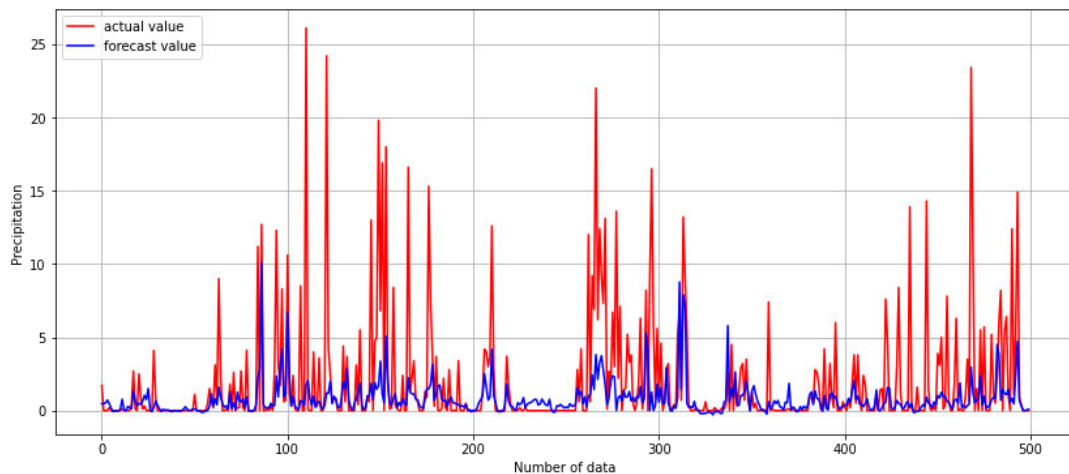


Abbildung 6. 1: Erste 500 vorhergesagten Niederschlagshöhe in lineare Regression und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data

Nach der Evaluation wird die Vorhersage anhand statistischer Hilfsmittel evaluiert:

- Bestimmtheitsmaß: 0.37 (bestes Ergebnis: 1.0)
- Mittlere Absolute Abweichung: 1.9 (bestes Ergebnis: 0.0)
- Vorhersagen, ob es regnet in Prozent: 78%

Aus dem Ergebnis wird verdeutlicht, dass der Vorhersagewert des Modells noch nicht besonders optimal ist. Besonders bei sehr hohem Niederschlagshöhe kann das Modell noch nicht so gut den Wert vorhersagen. Bei niedriger Niederschlagshöhe kann das Modell besser den Wert vorhersagen. Das Bestimmtheitsmaß liegt bei dem Wert 0.37 und die mittlere absolute Abweichung liegt bei dem Wert 1.9, ein befriedigendes Ergebnis für ein simpler Algorithmus, wie lineare Regression.

Wenn das Ergebnis in zwei Klassen klassifiziert wird, und zwar, ob es regnet oder nicht regnet, dann hat dieses Model ein gutes Ergebnis liefert. Die Genauigkeit des Modells liegt bei 78%.

6.1.2 Auswertung

Es konnte gezeigt werden, dass lineare Regression ein befriedigendes Ergebnis für eine einfache Vorhersage liefert. Es ist aber nicht überraschend, da es sich in dem Abschnitt 5.6.1 gezeigt hat, dass die genutzten Einflussvariable nicht so besonders starke lineare Beziehung mit Niederschlagshöhe haben, deshalb wird empfohlen, lineare Regression zu nutzen, wenn die Einflussvariablen mit starker linearen Beziehung zur Verfügung gestellt.

Der Nachteil von linearer Regression ist, dass sie keine Sequenz verarbeiten kann, deshalb kann hier der zeitliche Einfluss nicht betrachtet werden. Daher ist diese Forschungsmethode für eine Vorhersage des zukünftigen Wertes nicht so besonders geeignet.

6.2 LSTM

6.2.1 Architektur

Im Rahmen dieses Versuches wurde verschiedene LSTM-Architekturen trainiert und evaluiert. Der in Abbildung 6.2 dargestellte LSTM hat sich in diesen Tests das beste Ergebnis ergeben.

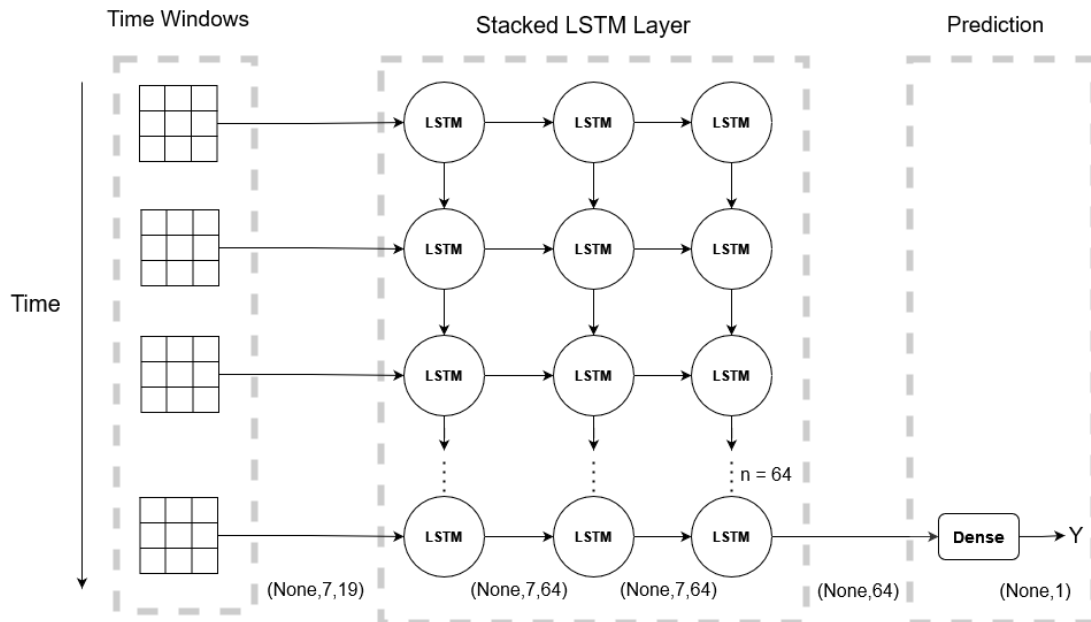


Abbildung 4. 7: Architektur der LSTM

Die Architektur besteht aus 3 LSTM versteckten Schichten (engl. *hidden layer*). Jede versteckte Schicht hat 64 LSTM-Neuronen. Die erste und zweite Schicht hat Sequenz als Output und die letzte Schicht hat Matrix als Output. Das Ergebnis der Verarbeitung der LSTM-Schicht wird in einer Dense-Schicht überführt. Diese Architektur garantiert, dass eine genügende Anzahl der Neuronen zur Verfügung gestellt wird, damit ein komplexe Zeitreihe verarbeitet wird. Als Aktivierungsfunktion kommt *Rectified Linear Unit* (ReLU) zum Einsatz. Nur in Dense-Schicht wird keine Aktivierungsfunktion (linear) genutzt. Als Optimizer wird *adaptive moment estimation* (ADAM) genutzt. Als Verlustfunktion wird die mittlere absolut Abweichung (engl. MAE) genutzt.

6.2.2 Ergebnis

Hier wird das erstellte Modell in Evaluation Dataset evaluiert. Anhand der Evaluation kann man sehen, ob das Modell ein gutes Ergebnis liefert.

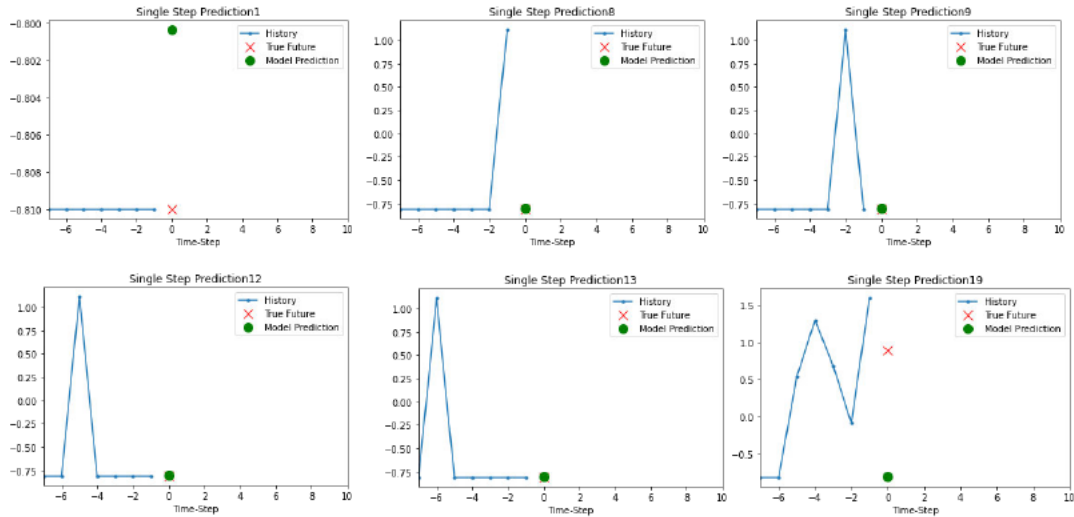


Abbildung 6. 2: Vorhersage-Beispiele der LSTM

Hier kann man sehen, dass das Modell als Input 7 Tage Einflussvariablen (engl. *features*) hat und die Niederschlagshöhe des nächsten Tages (8. Tag) vorhergesagt werden soll. Aus dem Ergebnis wird verdeutlicht, dass das Modell ein ziemlich gutes Ergebnis liefert. Das Modell kann einige Male die Niederschlagshöhe korrekt oder fast korrekt vorhersagen. Es hat aber auch gezeigt, dass das Modell auch manchmal eine schlechte Vorhersage liefert.

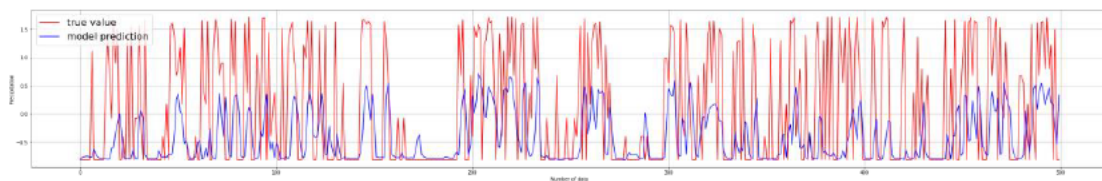


Abbildung 6. 3: Erste 500 vorhergesagten Niederschlagshöhe in LSTM und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data

Aus der Abbildung 6.3 kann man sehen, dass das Modell zwar das Muster des tatsächlichen Wertes folgen kann, aber Schwierigkeiten hat, bei sehr hohen Werten genau vorherzusagen.

Um das gesamte Ergebnis zu ermitteln, wird die Mittlere Absolute Abweichung für alle Vorhersagen in Evaluation Data gerechnet. Daraus ergibt sich folgendes Ergebnis.

- Mittlere Absolute Abweichung: 0.87 (bestes Ergebnis: 0.0)
- Vorhersagen, ob morgen regnet in Prozent: 74%

Die Mittlere Absolute Abweichung liegt bei **0,87**. Wenn das Ergebnis in zwei Klassen klassifiziert wird, und zwar, ob es morgen regnet, dann liegt die Genauigkeit dieses Modells bei **74%**.

6.2.3 Auswertung

Es konnte gezeigt werden, dass die Architektur der LSTM in der Lage ist, eine komplexe Zeitreihe zu analysieren und eine eher gute Vorhersage liefern. Insbesondere bei der Klassifikation, ob es morgen regnet, zeigt das Modell ein gutes Ergebnis. Nichtsdestotrotz wurde ebenfalls gezeigt, dass noch Bedarf an der Optimierung besteht, um die Vorhersage weiter zu verbessern.

6.3 Eindimensionale CNN

6.3.1 Architektur

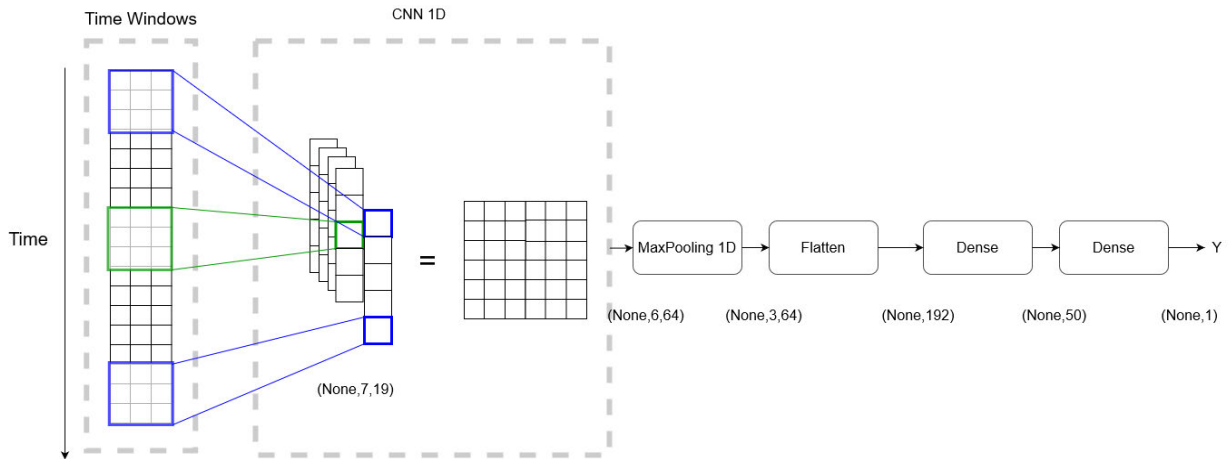


Abbildung 6. 4: Architektur eindimensionaler CNN

Im Rahmen dieses Versuchs werden mehrere Architekturen trainiert und evaluiert. Der in Abbildung 6.4 dargestellte eindimensionale CNN hat sich in diesen Tests als das beste Ergebnis ergeben. Die Architektur besteht aus eindimensionaler CNN-Schicht mit Anzahl des Kernels 2 und Anzahl der Filter 100 gefolgt durch eine MaxPooling-Schicht. Danach wird das Output von MaxPooling-Schicht in Flatten-Sicht flach gemacht. Danach werden zwei Dense-Schicht verwendet, um das Ergebnis zu vorhersagen.

6.3.2 Ergebnis

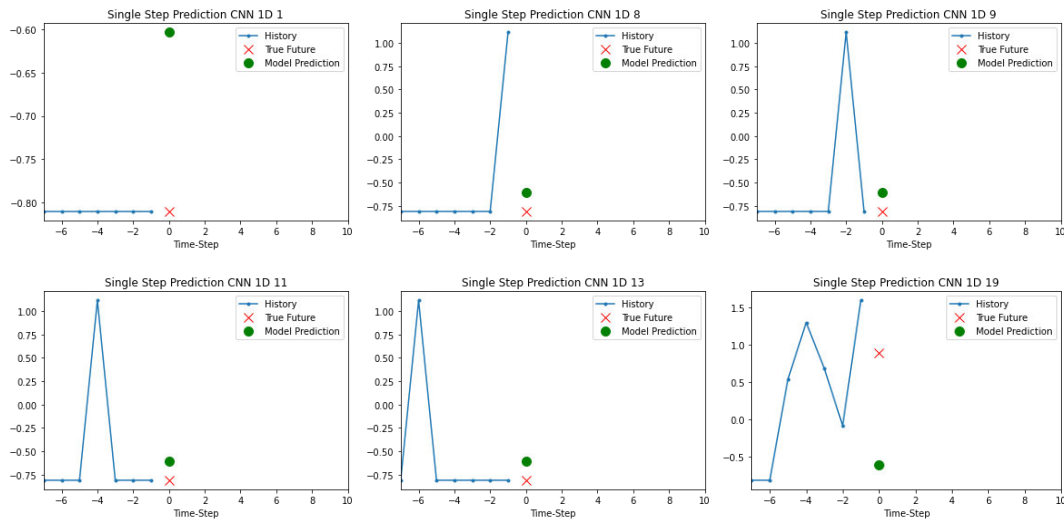


Abbildung 6. 5: Vorhersage-Beispiele der eindimensionalen konvolutionalen Schicht

Aus der Abbildung 6.5 kann man sehen, dass das Modell eher schlechte Vorhersagen liefert. Das Modell kann einige Male fast korrekt vorhergesagt werden. Es hat aber auch gezeigt, dass das Modell auch oft eine schlechte Vorhersage liefert.

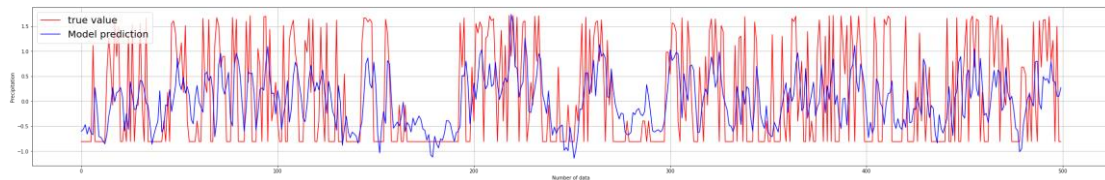


Abbildung 6. 6: Erste 500 vorhergesagten Niederschlagshöhe in eindimensionaler konvolutionen Schicht und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data

Aus der Abbildung 6.6 kann man sehen, dass das Modell das Muster des tatsächlichen Wertes nicht so besonders gut folgen kann. Bei niedrigen tatsächlichen Werten liefert oft das Modell falsche bzw. zu hohe Vorhersage.

Um das gesamte Ergebnis zu ermitteln, wird die Mittlere Absolute Abweichung für alle Vorhersagen in Evaluation Data gerechnet. Daraus ergibt sich folgendes Ergebnis.

- Mittlere Absolute Abweichung: 0.92 (bestes Ergebnis: 0.0)
- Vorhersagen, ob morgen regnet in Prozent: 67%

Die Mittlere Absolute Abweichung liegt bei **0.923**. Wenn das Ergebnis in zwei Klassen klassifiziert wird, und zwar, ob es morgen regnet, dann liegt die Genauigkeit dieses Modells bei **67%**.

6.3.3 Auswertung

Es hat gezeigt, dass eindimensionale CNN nicht optimal ist, um Niederschlagshöhe zu vorherhersagen. Deshalb wird empfohlen, bessere und komplexe Architektur zu versuchen, damit bessere Ergebnisse geliefert werden können.

6.4 CNN-LSTM

6.4.1 Architektur

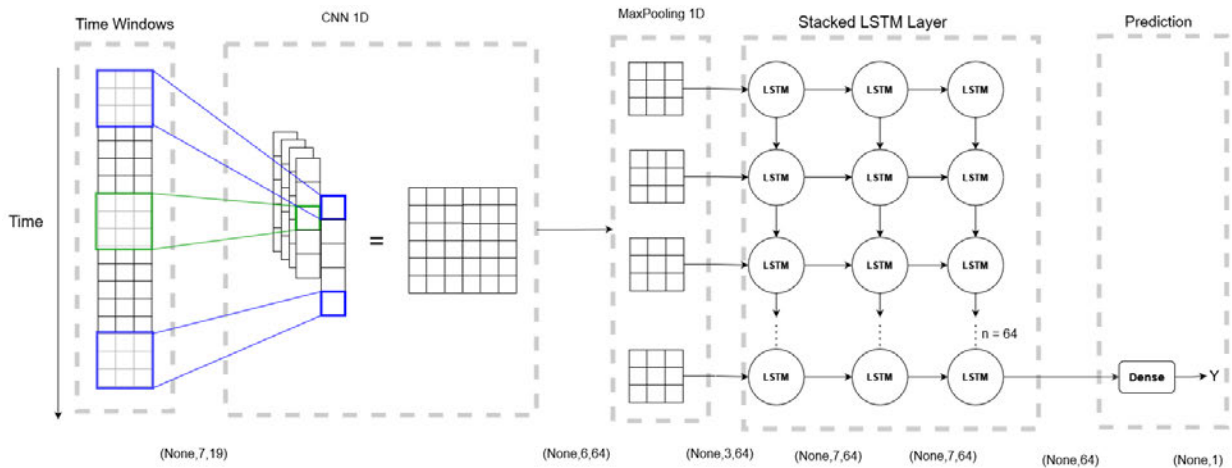


Abbildung 6. 7: Architektur der CNN-LSTM

Die Architektur besteht aus 1 CNN-Schicht und 3 LSTM versteckten Ebenen mit jeweils 64 LSTM-Neuron. Die Architektur der LSTM-Anteil ist identisch wie LSTM-Architektur in Abschnitt LSTM. Die konvolutionale Schicht hat die Aufgabe, die kleineren Ausschnitte des Zeitfensters in Hinblick auf bestimmte Verlaufsformen im Voraus zu untersuchen. Der Output des konvuntionaler Schicht ist eine Sequenz, die man einfach einer rekurrenten Schicht als Input übergeben kann. Die rekurrente Schicht analysiert dann den Ablauf, in der die voranalysierten Verlaufsformen nacheinander auftreten und produziert auf dieser Grundlage ein Schätzergebnis.

6.4.2 Ergebnis

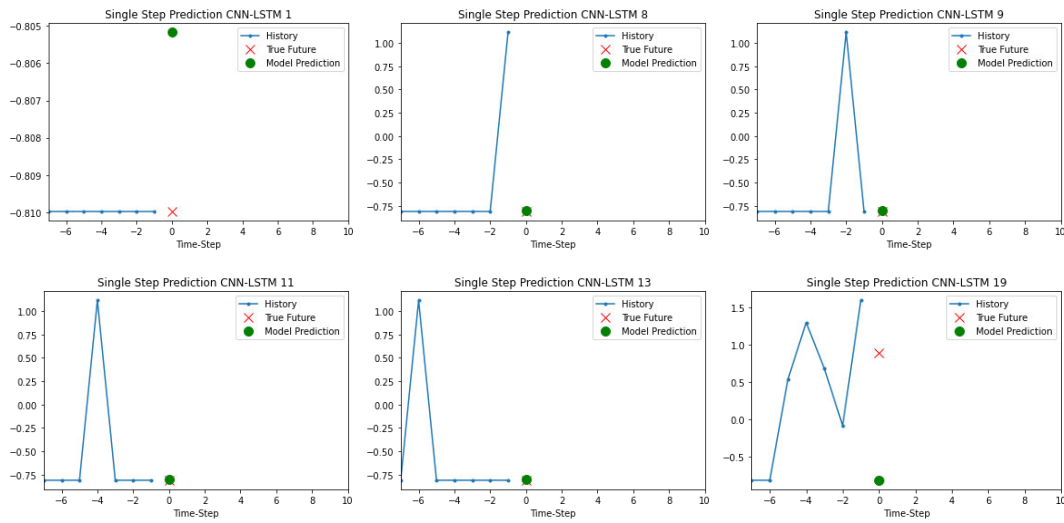


Abbildung 6. 8: Beispiel-Vorhersage der CNN-LSTM

Aus der Abbildung 6.8 wird verdeutlicht, dass das Modell ein ziemlich gutes Ergebnis liefert. Das Modell kann einige Male das Niederschlagshöhe korrekt oder fast korrekt vorhersagen. Es hat aber auch gezeigt, dass das Modell auch manchmal eine schlechte Vorhersage liefert.

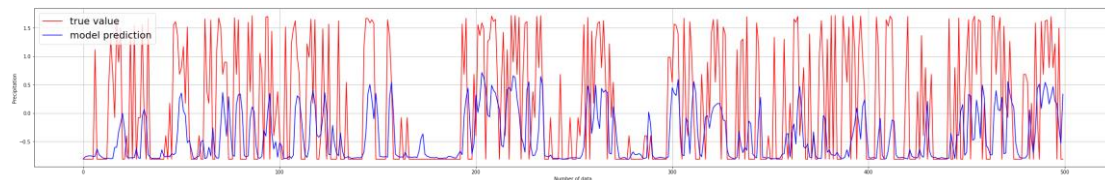


Abbildung 6. 9: Erste 500 vorhergesagten Niederschlagshöhe in CNN-LSTM und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data

Aus der Abbildung 6.9 kann man sehen, dass die Vorhersage des Modells das Muster des tatsächlichen Wertes besser folgen kann, insbesondere bei sehr niedrigen Niederschlagshöhe kann das Model bessere Vorhersagen liefern. Wenn die Niederschlagshöhe hoch ist, dann kann das Model zwar bessere Vorhersagen als LSTM oder CNN liefern, aber bei sehr hohe Niederschlagshöhe kann dies noch nicht erreicht werden.

Um das gesamte Ergebnis zu ermitteln, wird die Mittlere Absolute Abweichung für alle Vorhersagen in Evaluation Data gerechnet.

- Mittlere Absolute Abweichung: 0.86 (bestes Ergebnis: 0.0)
- Vorhersagen, ob morgen regnet in Prozent: 75%

Die Mittlere Absolute Abweichung liegt bei **0.86**. Wenn das Ergebnis in zwei Klassen klassifiziert wird, und zwar, ob es morgen regnet, Dann hat dieses Model gute Ergebnis liefert. Die Genauigkeit des Models liegt bei **75%**.

6.4.3 Auswertung

Das CNN-LSTM liefert ein etwas besseres Ergebnis als CNN oder LSTM. Bei sehr niedriger Niederschlagshöhe kann er eine gute Vorhersage liefern. Bei sehr hohe Niederschlagshöhe zwar bessere Ergebnisse als CNN oder LSTM aber noch nicht ganz optimal. Es besteht Bedarf an der Optimierung, um die Vorhersage weiter zu verbessern.

6.5 Transformer

6.5.1 Architektur

Im Rahmen dieses Versuchs werden die originale Transformer-Architektur verwendet.

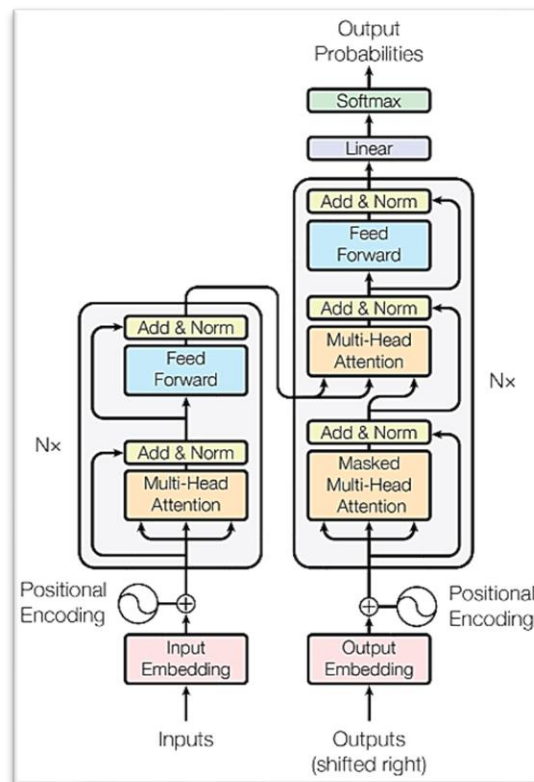


Abbildung 6. 10: Transformer-Architektur

Das originelle Transformer-Architektur mit originaler Hyperparameter, die für Textverarbeitung verwendet wird, wurde in dieser Arbeit trainiert und evaluiert, liefert aber schlechte Vorhersage. Das Modell kann das komplexe Muster der Evaluation-Daten gar nicht gefolgt werden, deshalb wird nach mehreren Versuchen entschieden, das originelle Transformer-Architektur zu vereinfachen.

Statt 6 Encoder und 6 Decoder wird 2 Encoder und 2 Decoder in dieser Architektur verwendet. Auch die Dimension für *feed forward* wird verringert, statt 2048 wird 128 genutzt. Außerdem wird auch die Anzahl der erwarteten Merkmale verkleinert, statt 512 wird 52 verwendet. Die Anzahl der Köpfe (englisch *head*) in *multi head attention* bleibt aber dasselbe, und zwar 8 Köpfe.

6.5.2 Ergebnis

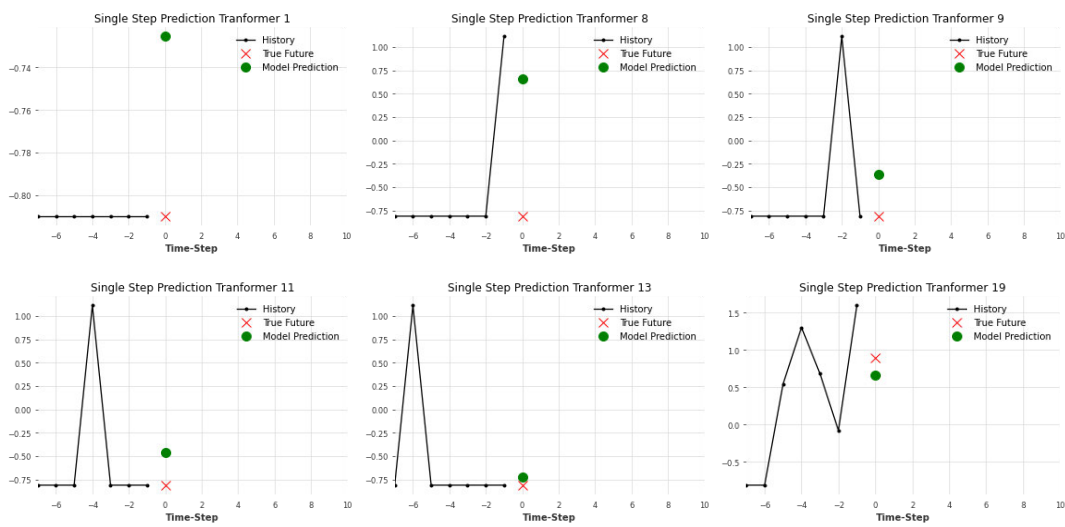


Abbildung 6. 11: Beispiel-Vorhersage der Transformer

Aus der Abbildung 6.5.2 wird verdeutlicht, dass das Modell ein ziemlich gutes Ergebnis liefert. Das Modell kann einige Male das Niederschlagshöhe fast korrekt vorhersagen. Es hat aber auch gezeigt, dass das Modell auch manchmal eine schlechte Vorhersage liefert.

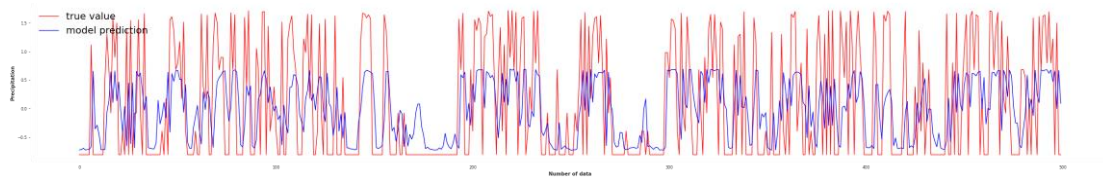


Abbildung 6. 12: Erste 500 vorhergesagten Niederschlagshöhe in Transformer und erste 500 tatsächlichen Niederschlagshöhe in Evaluation-Data

Aus der Abbildung 6.12 kann man sehen, dass die Vorhersage des Modells das Muster des tatsächlichen Wertes gut folgen kann. Das Modell kann bei sehr niedrigen Niederschlagshöhe kann ziemlich gute Vorhersage liefern. Wenn die Niederschlagshöhe hoch ist, dann kann das Modell noch nicht eine gute Vorhersage liefern, insbesondere bei sehr hohe Niederschlagshöhe kann dies noch nicht erreicht werden.

Um das gesamte Ergebnis zu ermitteln, wird die Mittlere Absolute Abweichung für alle Vorhersagen in Evaluation Data gerechnet.

- Mittlere Absolute Abweichung: 0.76 (bestes Ergebnis: 0.0)
- Vorhersagen, ob morgen regnet in Prozent: 75%

Die Mittlere Absolut Abweichung liegt bei **0.76**. Wenn das Ergebnis in zwei Klassen klassifiziert wird, und zwar, ob es morgen regnet, Dann hat dieses Model gute Ergebnis liefert. Die Genauigkeit des Models liegt bei **75%**.

6.5.3 Auswertung

Das Transformer-Modell liefert dasselbe Leistung wie CNN-LSTM. Das Modell kann im Vergleich zu CNN-LSTM bei niedriger und sehr niedriger genauer Vorhersage liefert, deshalb ist die mittlere absolute Abweichung bei diesem Model besser als andere Forschungsmethoden.

Bei hohe und sehr hohe Niederschlagshöhe kann das Modell leider wie anderen Forschungsmethoden noch nicht eine optimale Vorhersage geliefert werden. Es besteht Bedarf an der

Optimierung, um die Vorhersage weiter zu verbessern. Hierbei kann man das Hyperparameter weiter optimieren, diese beinhalten unter anderem *optimizer*, *learning rate* und *activation function*.

7. Fazit

7.1 Zusammenfassung

Ziel dieser Arbeit war die Untersuchung verschiedener maschinellen Verfahren zur Wettervorhersage insbesondere die Vorhersage der Niederschlagshöhe. Anhand der Ergebnisse werden gezeigt, ob maschinelles Lernen-Verfahren eine gute Vorhersage liefern kann.

Anhand der Experimente kann gezeigt werden, dass das maschinelle Lernen-Verfahren für Regenprognose ein gutes Ergebnis liefern kann. Sowohl LSTM, CNN-LSTM als auch Transformer hat gezeigt, dass sie in der Lage sind, komplexe Muster der Einflussvariable zu lernen. Daraus können CNN-LSTM und Transformer ein gutes Ergebnis bis **75%** Genauigkeit liefern bei der Vorhersage, ob es morgen regnet oder nicht.

Es konnte auch gezeigt werden, dass das Modell aber oft Schwierigkeiten hat, hohe und sehr hohe Niederschlagshöhen korrekt vorherzusagen. Es besteht hier der Bedarf an Optimierung.

Außerdem hat sich auch hier gezeigt, dass eindimensionale CNN kein geeignetes Hilfsmittel für die Wettervorhersage ist, da die Genauigkeit bei dieser Methode in diesem Experiment nur maximal bis **67%** liegt. Lineare Regression könnte auch ein gutes Ergebnis liefern (**78%**). Es konnte aber hier gezeigt werden, dass Lineare Regression keine gute Methode zur Vorhersage ist, da sie keine zeitliche Abhängigkeit zwischen Variablen analysieren bzw. lernen kann, deshalb wird für zukünftige Arbeit empfohlen, die lineare Regression nur für Vorverarbeitung der Wetterdaten verwendet werden.

Zusammengefasst kann in dieser Arbeit gezeigt werden, dass das maschinelle Verfahren eine potentielle Lösung für eine Wettervorhersage ist, jedoch besteht noch Bedarf an der Optimierung.

7.2 Ausblick

Um die Vorhersage weiter zu verbessern, kann man hierbei das Hyperparameter weiter optimieren, diese beinhalten unter anderem *optimizer (dropout)*, *learning rate* und *activation function* und die Länge der Zeitfenster. Darüber hinaus, um die Vorhersage hoher Werte zu verbessern, kann man statt tägliche Wetterdaten stündliche oder sogar 10-minütige Wetterdaten verwenden. Es könnte daraus folgen, dass die Anzahl der hohen Niederschlagshöhe als Training- und Evaluation-Daten vermehrt wird, daraus folgt, dass das Modell die hohe Niederschlagshöhe besser trainiert und vorhergesagt werden kann.

Darüber hinaus kann man die Vorhersage weiter verbessern, indem man neue Einflussvariablen hinzufügen bzw. alte Einflussvariablen entfernen, da es trotz der Analyse der Entstehung der Niederschlag in Kapitel 3 einige wichtige Einflussvariablen fehlen könnte. Ferner hat sich In Kapitel 3 gezeigt, dass die wichtige Einflussvariable nämlich Verdunstung fehlt, hier wird empfohlen, dass man andere Datenquelle für die Verdunstung nimmt. Des Weiteren kann man auch mit der Architektur weiter erweitern bzw. reduzieren. Die in dieser Arbeit verwendete Architektur kann für die Optimierung als Baseline genutzt werden

Für die Erweiterung der Forschung sind viele Optionen möglich. Hierbei kann man statt Ein-Schritt-Prognose Mehr-Schritte-Prognosen durchführen und analysieren. Außerdem man kann die Niederschlagshöhe weiter nach Niederschlagsintensität zum Beispiel in leichter, mäßiger oder starker Regen klassifizieren.

Darüber hinaus kann die Forschung erweitert werden, indem man andere Zielvariable nimmt. Hier sind interessant zum Beispiel die Vorhersage der Temperatur in Hamburg. Die in dieser Arbeit verwendete Vorverarbeitung- und Verarbeitung-Prozesse können wieder für diese Vorhersage verwendet werden.

8. Urheberechthinweis

Die in dieser Arbeit verwendete Wetterdaten stammen aus meteorologischer Station des Deutschen Wetterdienstes (DWD)² und dürfen laut seiner Verordnung³ unter Beigabe eines Quellenvermerks ohne Einschränkungen weiterverwendet werden.

² siehe für Details: <https://cdc.dwd.de/portal/>

³ siehe für Details: https://opendata.dwd.de/climate_environment/CDC/Nutzungsbedingungen_German.pdf

Quellenverzeichnis

- [1] J. Hirschle, *Machine Learning für Zeitreihen : Einstieg in Regressions-, ARIMA- und Deep Learning-Verfahren mit Python*. München: Hanser, 2021.
- [2] C. Chatfield, *The Analysis of Time Series*. USA: CHAPMAN&HALL/CRC, 2004.
- [3] J.-P. Kreiß und G. Neuhaus, *Einführung in die Zeitreihenanalyse*. Berlin: Springer, 2006.
- [4] J. Brownlee, *Deep Learning for Time Series Forecasting*. Melbourne: Machine Learning Mastery, 2020.
- [5] M. Stojiljković, *Linear Regression in Python*. [Online]. Verfügbar unter: <https://realpython.com/linear-regression-in-python/> (Zugriff am: 15. Februar 2022).
- [6] T. Patil und K. Shah, „Weather Forecasting Analysis using Linear and Logistic Regression Algorithm“, *International Research Journal of Engineering and Technology*, 2021.
- [7] Christopher Olah, *Understanding LSTM Networks*. Verfügbar unter: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Zugriff am: 16. Februar 2022.
- [8] Guokun Lai, Wei-Cheng Chang, Yiming Yang und Hanxiao Liu, „Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks“, *Association for Computing Machinery*, S. 95–104, 2018.
- [9] A. Vaswani *et al.*, „Attention Is All You Need“, *Curran Associates Inc*, S. 6000–6010, 2017.

- [10] L. Fahrmeir, C. Heumann, R. Künstler, I. Pigeot und G. Tutz, *Statistik: Der Weg zur Datenanalyse*, 8. Aufl. Berlin, Heidelberg: Springer Spektrum, 2016. [Online]. Verfügbar unter: <http://www.springer.com/>
- [11] J. Vogel, *Prognose von Zeitreihen : eine Einführung für Wirtschaftswissenschaftler*. Wiesbaden: Springer Gabler, 2015.
- [12] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*, 2. Aufl. Kanada: O'Reilly Media, 2019.
- [13] M. Feindt und U. Kerzel, *Prognosen bewerten : statistische Grundlagen und praktische Tipps*. Berlin: Springer Gabler, 2015.
- [14] Deutscher Wetterdienst, *Niederschlag*. [Online]. Verfügbar unter: <https://www.dwd.de/DE/service/lexikon/begriffe/N/Niederschlag.html> (Zugriff am: 14. Februar 2022).
- [15] United Kingdom National Meteorological Service, *The water cycle*. [Online]. Verfügbar unter: <https://www.metoffice.gov.uk/weather/learn-about/weather/how-weather-works/water-cycle> (Zugriff am: 15. Februar 2022).
- [16] Australia Bureau of Meteorology, *Rain in Australia (Dataset)*. [Online]. Verfügbar unter: <https://www.kaggle.com/jsphyg/weather-dataset-rattle-package/version/2> (Zugriff am: 15. Februar 2022).
- [17] DWD Climate Data Center, *Tägliche Stationsmessungen Niederschlagshöhe in mm*. Zugriff am: 8. Oktober 2021.
- [18] DWD Climate Data Center, *Tägliche Stationsmessungen des Maximums der Lufttemperatur in 2 m*. Zugriff am: 8. Oktober 2021.
- [19] DWD Climate Data Center, *Tägliche Stationsmessungen des Minimums der Lufttemperatur in 2 m*. Zugriff am: 8. Oktober 2021.
- [20] DWD Climate Data Center, *Tägliche Stationsmessungen der Sonnenscheindauer in Stunden*. Zugriff am: 8. Oktober 2021.

- [21] DWD Climate Data Center, *Tägliche Stationsmessungen des mittleren Dampfdrucks in hPa*. Zugriff am: 8. Oktober 2021.
- [22] DWD Climate Data Center, *Tagesmittel der relativen Feuchte in 2 m Höhe in %*. Zugriff am: 8. Oktober 2021.
- [23] DWD Climate Data Center, *Stündliche Stationsmessungen der Taupunkttemperatur in 2 m Höhe in °C*. Zugriff am: 8. Oktober 2021.
- [24] DWD Climate Data Center, *Stündliche Stationsmessungen der Windrichtung in ca. 10 m Höhe in Grad*. Zugriff am: 8. Oktober 2021.
- [25] DWD Climate Data Center, *Stündliche Stationsmessungen der Windgeschwindigkeit in ca. 10 m Höhe in m/s*. Zugriff am: 8. Oktober 2021.
- [26] DWD Climate Data Center, *Tägliche Stationsmessungen der maximalen Windspitze in ca. 10 m Höhe in m/s*. Zugriff am: 8. Oktober 2021.
- [27] DWD Climate Data Center, *Stündliche Stationsmessungen des Luftdruckes reduziert auf Meereshöhe*. Zugriff am: 8. Oktober 2021.
- [28] DWD Climate Data Center, *Stündliche Stationsmessungen des Bedeckungsgrades in Achtel*. Zugriff am: 8. Oktober 2021.
- [29] DWD Climate Data Center, *Stündliche Stationsmessungen der Lufttemperatur in 2 m Höhe*. Zugriff am: 8. Oktober 2021.
- [30] U. Kamps, *Normalverteilung*. [Online]. Verfügbar unter: <https://wirtschaftslexikon.gabler.de/definition/normalverteilung-39769> (Zugriff am: 16. Februar 2022).
- [31] Y. LeCun, L. Bottou, G. B. Orr und K.-R. Müller, „Efficient BackProp“ in *Neural Networks: Tricks of the Trade (2nd ed.)*, G. Montavon, G. B. Orr und K.-R. Müller, Hg., Springer, 2012, S. 9–48. [Online]. Verfügbar unter: <http://dblp.uni-trier.de/db/series/lncs/lncs7700.html#LeCunBOM12>
- [32] I.-K. Yeo und R. A. Johnson, „A New Family of Power Transformations to Improve Normality or Symmetry“, *Biometrika*, Jg. 87, Nr. 4, S. 954–959, 2000. [Online]. Verfügbar unter: <http://www.jstor.org/stable/2673623>

A. Anhang

A. 1. Benötigte Programmbibliotheken

Hier sind die Liste aller verwendete Programmbibliotheken, die in dieser Arbeit verwendet wurden (*requirement.txt*).

```
numpy==1.21.5
pandas==1.3.5
matplotlib==3.5.1
scipy==1.4.1
seaborn==0.11.2
xarray==0.18.2
darts==0.16.1
tensorflow==2.8.0
matplotlib==3.5.1
scikit-learn==1.0.2
wandb==0.12.10
keras==2.8.0
Keras-Preprocessing==1.1.2
```

A. 2. Kodieren kategorialer Variable

Kategoriale Variable der Windrichtung	Kodierte kategoriale Variable der Windrichtung
E	0
ENE	1
ESE	2
N	3
NE	4
NNE	5
NNW	6
NW	7
S	8
SE	9
SSE	10
SSW	11
SW	12
W	13
WNW	14
WSW	15

Tabelle A. 1: Kodieren kategorialer Variable

A. 3. Alle verwendeten Merkmale

Nach Merkmalextraktion werden mehrere neue Merkmale aufgetaucht. Hier sind die Liste aller verwendeten Merkmalen in dieser Arbeit.

Num.	Merkmal
1.	Niederschlagshöhe
2.	Sonnenscheindauer
3.	Maximale Temperatur
4.	Minimale Temperatur
5.	Relative Feuchtigkeit
6.	Dampfdruck
7.	Maximale Windspitze
8.	Windrichtung um 9
9.	Windrichtung um 15
10.	Windgeschwindigkeit um 9
11.	Windgeschwindigkeit um 15
12.	Luftdruck um 9
13.	Luftdruck um 15
14.	Bewölkung um 9
15.	Bewölkung um 15
16.	Temperatur um 9
17.	Temperatur um 15
18.	Tautemperatur
19.	Monat
20.	Tag

Tabelle A. 2: Alle verwendeten Merkmale

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original