

Hochschule Darmstadt

Fachbereiche Mathematik und Naturwissenschaften & Informatik

Anwendung von Machine Learning zur Prognose von Erdbebenereignissen

Abschlussarbeit zur Erlangung des akademischen Grades

Master of Science (M. Sc.)
im Studiengang Data Science

vorgelegt von

Leonard Conrath

Referent : Prof. Dr. Andreas Weinmann

Korreferent : Prof. Dr. Arnim Malcherek

Ausgabedatum : 1. November 2023

Abgabedatum : 02. April 2024

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 02. April 2024

Leonard Conrath

ABSTRACT

This paper sheds light on the use of Machine Learning (ML) in earthquake prediction, a research area of significant importance for safety, infrastructure, and the economy. Given the challenges associated with accurate earthquake forecasting and the identification of precursor patterns, this study focuses on evaluating the effectiveness and accuracy of various ML methods compared to traditional seismological approaches. This is achieved through a comprehensive meta-analysis of existing literature and the development and validation of a proprietary ML model for predicting the magnitude and occurrence of major seismic events.

At the heart of this work's methodological approach is the integration of multidisciplinary techniques that combine ML algorithms with seismological data to gain comprehensive insights into seismic patterns and their predictive potential. The investigation includes a wide range of data sources, including seismic waveform data, geological and tectonic information, and historical earthquake records. A key scientific finding of the study is the significant role of feature engineering and the integration of new data parameters using advanced seismological computation methods in improving prediction accuracy. This underscores that, alongside advanced ML models, especially those based on Deep Learning and Ensemble methods, the quality of data preparation is crucial for successful forecasting.

Moreover, the development and implementation of a proprietary ML model in the Turkey-Syria region demonstrate that such models can predict the magnitude and occurrence of major earthquake events with remarkable precision. An LSTM model achieved excellent accuracy in magnitude prediction with a MAE of 0.6679 and an MSE of 1.2887. The prediction of future major earthquakes was precisely achieved with the help of an Ensemble model, which boasts an accuracy of 0.873 and a precision of 0.5496, as well as another LSTM model distinguished by a sensitivity of 0.8830.

These findings underscore the scientific relevance of ML in seismology and its ability to significantly enhance the accuracy and reliability of earthquake forecasts. They highlight the need for continuous improvement of ML algorithms, optimization of data quality and diversity, and the establishment of uniform evaluation criteria to increase the precision of prediction models. Thus, the integration of ML into seismological research is crucial for effective risk reduction and preparedness for earthquakes.

ZUSAMMENFASSUNG

Diese Arbeit beleuchtet den Einsatz von Machine Learning (ML) in der Erdbebenvorhersage, einem Forschungsfeld von großer Tragweite für Sicherheit, Infrastruktur und Wirtschaft. Vor dem Hintergrund der Schwierigkeiten, die mit der genauen Vorhersage von Erdbeben und der Identifikation von Vorläufermustern verbunden sind, fokussiert sich diese Studie auf die Untersuchung der Effektivität und Genauigkeit verschiedener ML-Methoden im Vergleich zu traditionellen seismologischen Ansätzen. Dies erfolgt einerseits durch eine umfassende Meta-Analyse bestehender Literatur und andererseits durch die Entwicklung und Validierung eines eigenen ML-Modells zur Vorhersage der Magnitude und des Auftretens starker Erdbebenereignisse.

Im Zentrum des methodischen Ansatzes dieser Arbeit steht die Integration multidisziplinärer Techniken, welche ML-Algorithmen mit seismologischen Daten verbinden, um umfassende Einblicke in seismische Muster und deren Vorhersagepotenzial zu erlangen. Die Untersuchung umfasst eine breite Palette an Datenquellen, darunter seismische Wellenformdaten, geologische und tektonische Informationen sowie historische Erdbeben Datensätze. Eine zentrale wissenschaftliche Erkenntnis der Studie ist die signifikante Rolle des Feature Engineerings, der Integration neuer Datenparameter mittels fortschrittlicher seismologischer Berechnungsmethoden, bei der Verbesserung der Vorhersagegenauigkeit. Dies unterstreicht, dass neben fortschrittlichen ML-Modellen, insbesondere jenen, die auf Deep Learning und Ensemble-Methoden basieren, die Qualität der Datenaufbereitung entscheidend für den Erfolg der Vorhersage ist.

Darüber hinaus demonstriert die Entwicklung und Umsetzung eines eigenen ML-Modells in der Türkei-Syrien-Region, dass solche Modelle die Magnitude und das Auftreten starker Erdbebenereignisse mit bemerkenswerter Präzision vorhersagen können. Hierbei erreichte ein LSTM-Modell mit einem MAE von 0.6679 und einem MSE von 1.2887 eine exzellente Genauigkeit in der Magnitudenvorhersage. Die Vorhersage zukünftiger starker Erdbeben gelang präzise mit Hilfe eines Ensemble-Modells, das eine Genauigkeit von 0.873 und eine Präzision von 0.5496 aufweist, sowie eines weiteren LSTM-Modells, das sich durch eine Sensitivität von 0.8830 auszeichnet.

Diese Erkenntnisse unterstreichen die wissenschaftliche Relevanz von ML in der Seismologie und dessen Fähigkeit, die Genauigkeit und Verlässlichkeit von Erdbebenvorhersagen wesentlich zu erhöhen. Sie betonen die Notwendigkeit, ML-Algorithmen fortlaufend zu verbessern, Datenqualität und -diversität zu optimieren und einheitliche Bewertungskriterien zu etablieren, um die Präzision der Vorhersagemodelle zu steigern. Der Einsatz von ML in der seismologischen Forschung ist damit entscheidend für eine effektive Risikominderung und Vorbereitung auf Erdbeben.

INHALTSVERZEICHNIS

I	Thesis	
1	Einleitung	2
1.1	Hintergrund	2
1.2	Forschungsfrage	3
1.3	Zielsetzung	4
1.4	Methodik und Aufbau der Arbeit	4
2	Seismische Grundlagen und Erdbebenerkennung	6
2.1	Seismische Grundlagen und die Mechanismen von Erdbeben	6
2.1.1	Forschungsfeld der Seismologie	6
2.1.2	Erdbeben: Definition, Ursachen und Messmethoden	6
2.1.3	Auswirkungen, Bedeutung und Herausforderungen der Erdbebenvorhersage	7
2.2	Anwendung gesetzlicher Grundlagen und seismologischer Techniken	9
3	Maschinelles Lernen in der Erdbebenvorhersage	11
3.1	Präzisierung des Begriffs Machine Learning	11
3.2	Potenziale und Herausforderungen in der Erdbebenvorhersage	12
3.3	Indikatoren für die Erdbebenvorhersage durch Machine Learning	14
3.3.1	Grundlegende seismische Daten	14
3.3.2	Fortgeschrittene seismologische Berechnungen	14
3.4	Machine Learning Methoden in der Erdbebenvorhersage	17
3.4.1	Überwachte Lernmethoden	17
3.4.2	Unüberwachte Lernmethoden	24
3.4.3	Hybrid- und Ensemble-Methoden	25
3.5	Modellbewertungsmetriken	27
3.5.1	Klassifikationsmetriken	27
3.5.2	Regressionsmetriken	28
4	Aktueller Stand der Forschung zum Einsatz von Machine Learning in der Erdbebenvorhersage	30
4.1	Methodik der Analyse und Strukturierung der Studien	30
4.2	Spezifische Algorithmen	33
4.3	Vergleichende Studien	39
4.4	Fortgeschrittene Hybrid- und Ensemble-Modelle	45
4.5	Erkenntnisse	54
4.5.1	Effizienzvergleich der verschiedenen Modelle	54
4.5.2	Verbesserung der Datenaufbereitung und -nutzung	55
4.5.3	Herausforderungen bei der Implementierung	55
4.5.4	Zukunftsstrategien für Machine Learning in der Erdbebenvorhersage	56
5	Modellentwicklung	58

5.1	Methodische Grundlagen und Zielsetzung	58
5.2	Beschreibung, Preprocessing und Analyse des Datensatzes . . .	59
5.2.1	Beschreibung des Datensatzes	59
5.2.2	Preprocessing	59
5.2.3	Explorative Datenanalyse	59
5.3	Modellauswahl und Festlegung von Bewertungskriterien	62
5.4	Datenaufbereitung	64
5.4.1	Aufteilung in Trainings- und Testdatensätze	66
5.4.2	Optimierung der Modelle	66
5.5	Lineare und logistische Regression	68
5.5.1	Vorhersage der Magnituden	68
5.5.2	Vorhersage starker Erdbebenereignisse	68
5.6	Random Forest	70
5.6.1	Vorhersage der Magnituden	70
5.6.2	Vorhersage starker Erdbebenereignisse	70
5.7	Künstliches Neuronales Netz	72
5.7.1	Vorhersage der Magnituden	72
5.7.2	Vorhersage starker Erdbebenereignisse	72
5.8	Long Short-Term Memory	74
5.8.1	Vorhersage der Magnituden	74
5.8.2	Vorhersage starker Erdbebenereignisse	74
5.9	Ensemble-Modelle	76
5.9.1	Vorhersage der Magnituden	76
5.9.2	Vorhersage starker Erdbebenereignisse	76
5.10	Analyse und Reflexion der Studienergebnisse	78
5.10.1	Ergebnisse der Modelle im Detail	78
5.10.2	Tiefgreifende Diskussion der Studienbefunde	78
6	Schlussfolgerungen und Ausblick	82
6.1	Zusammenfassung der Ergebnisse	82
6.2	Limitationen der Studie	82
6.3	Praktische Implikationen und Empfehlungen für zukünftige Forschung	83
II Appendix		
A	Jupyter Notebook zur Erdbebenvorhersage	85
Literatur		121

ABBILDUNGSVERZEICHNIS

Abbildung 3.1	SVM-Klassifikationsverfahren	17
Abbildung 3.2	Illustration des KNN-Klassifikationsverfahrens mit drei Erdbebenklassen	20
Abbildung 3.3	Schematische Darstellung der Architektur eines ANNs	22
Abbildung 3.4	Visualisierung seismischer Cluster mittels K-Means Clustering Algorithmus	24
Abbildung 4.1	Verteilung der Forschungsansätze	32
Abbildung 5.1	Magnitudenentwicklung in der Türkei-Syrien Region .	60
Abbildung 5.2	Lokalisierung der Erbeben in der Türkei-Syrien Region	60

TABELLENVERZEICHNIS

Tabelle 2.1	Kritische Bewertung traditioneller Erdbebenvorhersagemethoden	8
Tabelle 4.1	Zusammenfassung der spezifischen Algorithmen . . .	38
Tabelle 4.2	Zusammenfassung der vergleichenden Studien	44
Tabelle 4.3	Zusammenfassung der fortgeschrittenen Hybrid- und Ensemble-Modelle	53
Tabelle 5.1	Evaluationsergebnisse für die Vorhersage der Erdbebenmagnitudo	78
Tabelle 5.2	Evaluationsergebnisse für die Vorhersage des Auftretens starker Erdbebenereignisse	78

ABKÜRZUNGSVERZEICHNIS

ADABOOST	Adaptive Boosting
ANN	Künstliche Neuronale Netze
ANFIS	Adaptive Neuro-Fuzzy Inference System
Bagging	Bootstrap Aggregating
CART	Classification and Regression Trees
CHAID	Chi-squared Automatic Interaction Detector
CNN	Convolutional Neural Networks
DL	Deep Learning
DT	Decision Trees
ELM	Extreme Learning Machines
EPNN	Evolutionary Polynomial Neural Network
ERT	Extremely Randomized Trees
FFNN	Feedforward-Neuronale Netzwerke
FLANN	Functional Link Artificial Neural Network
FN	False Negatives
FP	False Positives
GBM	Gradient Boosting Machine
GLM	Generalisierte Lineare Modelle
GP	Genetische Programmierung
HNN	Hybrid Neural Networks
KNN	K-Nearest Neighbors-Algorithmus
LinR	Lineare Regression
LogR	Logistische Regression
LPBoost	Linear Programming Boosting
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error

MCC	Matthew's Korrelationskoeffizient
ML	Machine Learning
MSE	Mean Squared Error
NB	Naive Bayes-Algorithmus
NDC	Neuronale Dynamikklassifizierung
NSE	Nash-Sutcliffe-Effizienzkoeffizient
PNN	Probabilistische Neuronale Netzwerke
RF	Random Forest
RNN	Rekurrente Neuronale Netzwerke
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
SVR	Support Vector Regression
TN	True Negatives
TP	True Positives
USGS	United States Geological Survey

Teil I

THESIS

EINLEITUNG

1.1 HINTERGRUND

Erdbeben stellen eine der verheerendsten Formen natürlicher Katastrophen dar, mit gravierenden Folgen für das menschliche Leben, die Infrastruktur und die Wirtschaft. Der United States Geological Survey (USGS) schätzt, dass jährlich etwa 500.000 Erdbeben stattfinden, von denen rund 100.000 von Menschen wahrgenommen werden können und etwa 100 ernsthafte Zerstörungen verursachen [63]. Angesichts dieser Zahlen ist die Bedeutung einer präzisen Erdbebenvorhersage sowohl aus wissenschaftlicher als auch aus gesellschaftlicher Perspektive immens. Trotz intensiver Forschungsbemühungen bleibt die exakte Vorhersage von Erdbeben eine der größten ungelösten Fragen in der Geophysik. Traditionelle Ansätze, die auf die Analyse von Vorbebenaktivitäten und tektonischen Spannungen basieren, haben sich als informativ erwiesen, weisen jedoch signifikante Einschränkungen auf, insbesondere bei der Vorhersage des genauen Zeitpunkts und Ortes eines bevorstehenden Erdbebens [25].

In diesem Kontext eröffnet Machine Learning (ML) neue Wege für die Erdbebenvorhersage. ML-Algorithmen sind in der Lage, komplexe Muster in seismischen Daten zu erkennen, die für menschliche Beobachter oder herkömmliche Analysemethoden schwer zugänglich sind. Insbesondere Deep-Learning-Algorithmen werden zunehmend für die Erdbebenvorhersage eingesetzt, da sie große Datensätze verarbeiten und aus Fehlern lernen können [28]. Die Anwendung von ML in der Erdbebenvorhersage beinhaltet eine Vielzahl von Datentypen. Neben seismischen Daten, die Amplitude, Frequenz und Dauer seismischer Wellen abbilden, werden auch geografische und tektonische Daten einbezogen, die Informationen über Plattengrenzen, Bruchzonen und historische Erdbebenaktivitäten in bestimmten Regionen liefern können [43, 50].

Die vorliegende Thesis konzentriert sich auf die Analyse und Anwendung von ML-Techniken zur Erdbebenvorhersage. Die immense Bedeutung einer genauen Erdbebenvorhersage für die öffentliche Sicherheit macht diese Forschungsrichtung hochrelevant. Sie verbindet theoretische und praktische Aspekte und zielt darauf ab, das Verständnis und die Effektivität von ML-gestützten Vorhersagemodellen zu verbessern.

1.2 FORSCHUNGSFRAGE

Die zentrale Ambition dieser Thesis ist es, die Effektivität und Präzision von ML-Techniken in der Vorhersage von Erdbeben zu evaluieren und zu optimieren. Um eine tiefergehende Untersuchung zu ermöglichen, werden spezifische und detaillierte Forschungsfragen formuliert, die auf die Erkennung relevanter seismischer Muster, die präzise Prognose von Erdbebenparametern sowie die Verbesserung und praktische Anwendung dieser Technologien abzielen. Die überarbeiteten Forschungsfragen lauten wie folgt:

- 1.) **Vergleichende Analyse der Effektivität von ML-Modellen gegenüber traditionellen seismologischen Methoden:** Welche ML-Modelle übertreffen traditionelle seismologische Vorhersagemethoden in Bezug auf die Erkennung seismischer Muster und die Vorhersage von Zeitpunkt, Ort und Stärke von Erdbeben? Diese Frage soll durch einen systematischen Vergleich der Vorhersagegenauigkeit, -zuverlässigkeit und -effizienz beider Ansätze beantwortet werden.
- 2.) **Identifikation und Evaluation der robustesten ML-Modelle für spezifische Vorhersageparameter:** Welche ML-Modelle erweisen sich als besonders effektiv für die Vorhersage spezifischer Erdbebenparameter (Zeitpunkt, Ort, Magnitude) und unter welchen Bedingungen (z.B. Datenverfügbarkeit, geografische Regionen)? Eine detaillierte Untersuchung soll Aufschluss darüber geben, wie Modellarchitekturen und -parameter an verschiedene Vorhersageaufgaben angepasst werden können.
- 3.) **Optimierung der Datennutzung für präzisere ML-basierte Erdbebenvorhersagen:** Welche spezifischen Datentypen (seismische Wellenformdaten, geologische und tektonische Daten, historische Erdbebendatensätze) tragen am meisten zur Genauigkeit von ML-Vorhersagen bei? Diese Frage zielt darauf ab, die Effektivität verschiedener Datenquellen zu bewerten und Methoden zur Überwindung von Herausforderungen wie unvollständigen oder verrauschten Daten zu entwickeln.
- 4.) **Optimierung von ML-Methoden für präzisere Erdbebenvorhersagen:** Inwieweit lassen sich durch den Einsatz neuer ML-Techniken, wie Deep Learning und Ensemble-Methoden, die Genauigkeit und Zuverlässigkeit von Vorhersagemodellen für Erdbeben steigern? Ziel ist die Untersuchung, wie innovative Algorithmen die Limitationen existierender Ansätze überwinden und signifikant zur Verbesserung der Vorhersageleistung beitragen können, um letztendlich die Effektivität von Katastrophenmanagement-Strategien zu erhöhen.

Durch die Beantwortung dieser präzisierten Forschungsfragen soll ein tieferes Verständnis der Potenziale und Herausforderungen von ML in der Erdbebenvorhersage erlangt werden. Ziel ist es, die wissenschaftliche Grundlage für die Weiterentwicklung von Vorhersagemodellen

zu schaffen und zugleich konkrete Ansätze für die Anwendung dieser Technologien im Bereich des Katastrophenmanagements zu entwickeln.

1.3 ZIELSETZUNG

Diese Thesis zielt darauf ab, die Potenziale und Grenzen von ML-Technologien in der Erdbebenvorhersage umfassend zu untersuchen. Basierend auf aktuellen Forschungsergebnissen wird analysiert, inwiefern moderne ML-Methoden in der Lage sind, seismische Muster effektiver als traditionelle Prognoseansätze zu identifizieren und vorherzusagen. Ein kritisches Augenmerk liegt auf der Bewertung von Leistung und Zuverlässigkeit existierender ML-Modelle unter realen Bedingungen.

Ein weiteres Anliegen ist die Entwicklung und empirische Validierung neuer ML-Modelle, die auf echten seismischen Daten basieren, um ihre Vorhersagegenauigkeit und Anwendbarkeit in der Praxis zu überprüfen.

Darüber hinaus werden konkrete Empfehlungen für die Integration von ML-Modellen in seismologische Vorhersagesysteme erarbeitet, um sowohl die öffentliche Sicherheit als auch die Risikobewertung zu verbessern. Der wissenschaftliche Beitrag dieser Arbeit besteht darin, das Verständnis und Management von Erdbebenrisiken durch die Einführung präziserer Vorhersagemethoden voranzutreiben.

1.4 METHODIK UND AUFBAU DER ARBEIT

Diese Thesis verfolgt einen multidisziplinären Ansatz, der die Grundlagen der Seismologie mit fortschrittlichen ML-Techniken verknüpft, um die Möglichkeiten und Grenzen der Erdbebenvorhersage zu erkunden. Die Struktur der Arbeit ist wie folgt gegliedert:

1. Einleitung: Stellt den Forschungskontext, die Motivation und die Zielsetzung der Studie vor.
2. Seismische Grundlagen und Erdbebenerkennung: Erläutert seismologische Konzepte und traditionelle Vorhersagemethoden.
3. ML in der Erdbebenvorhersage: Behandelt die grundlegenden Prinzipien des ML und erörtert dessen Rolle bei der Vorhersage seismischer Ereignisse. Dabei wird auf die Anwendung verschiedener ML-Methoden eingegangen.
4. Stand der Forschung: Untersucht aktuelle Studien zum Einsatz von maschinellen Lernverfahren in der Erdbebenvorhersage. Die gewonnenen Erkenntnisse werden systematisch aufbereitet und zusammengefasst.
5. Implementierung maschineller Lernverfahren: Erörtert die methodologischen Grundlagen einschließlich Datenaufbereitung und Modellkonzeption. Des Weiteren wird die Implementierung und Evaluierung der

Modelle detailliert dargestellt. Abschließend erfolgt die Präsentation der Forschungsergebnisse, die im Licht der Zielsetzung und des aktuellen Forschungsstands diskutiert werden.

6. Schlussfolgerungen und Ausblick: Fasst die wichtigsten Erkenntnisse zusammen, reflektiert die Limitationen der Studie und skizziert zukünftige Forschungsrichtungen.

Jedes Kapitel baut auf dem vorherigen auf, um ein umfassendes Bild der aktuellen Herausforderungen und Möglichkeiten in der Erdbebenvorhersage zu zeichnen, mit einem besonderen Fokus auf die Anwendung und Evaluation von ML-Modellen.

SEISMISCHE GRUNDLAGEN UND ERDBEBENERKENNUNG

2.1 SEISMISCHE GRUNDLAGEN UND DIE MECHANISMEN VON ERDBEBEN

2.1.1 *Forschungsfeld der Seismologie*

Seismologie, die Wissenschaft von Erdbeben und der Ausbreitung elastischer Wellen durch die Erde, erforscht die treibenden Kräfte tektonischer Bewegungen, vulkanischer Aktivitäten und anderer geophysikalischer Prozesse, die Erdbeben auslösen. Diese Disziplin konzentriert sich auf das Verständnis der Zusammenhänge zwischen den Ursachen und Auswirkungen seismischer Ereignisse. Fortgeschrittene Instrumente wie Seismographen und Seismometer ermöglichen die präzise Messung von Erdbebenvorgängen. Seit dem 19. Jahrhundert hat die seismologische Forschung, begünstigt durch wissenschaftliche und technologische Fortschritte, ein umfassendes Verständnis der Erdstrukturen und der Dynamik von Erdbeben erlangt. Die heutige Seismologie profitiert enorm von modernsten Technologien und der Verfügbarkeit umfangreicher digitaler Datensätze, wodurch die Datenanalyse präziser denn je ist. Diese Entwicklungen tragen entscheidend zur Verfeinerung von Methoden für die Erdbebenvorhersage und zur Risikominderung bei [44].

2.1.2 *Erdbeben: Definition, Ursachen und Messmethoden*

Erdbeben werden durch plötzliche tektonische Bewegungen entlang von Verwerfungslinien in der Erdkruste verursacht, die gespeicherte elastische Spannungsenergie als seismische Wellen freisetzen. Das Ursprungsgebiet dieser Freisetzung, bekannt als Hypozentrum, und die resultierenden Bodenerschütterungen am Epizentrum an der Erdoberfläche, charakterisieren ein Erdbeben. Es werden primär zwei Arten seismischer Wellen unterschieden: Körperwellen, die sich durch das Erdinnere bewegen, und Oberflächenwellen, die entlang der Erdkruste propagieren. Körperwellen bestehen aus schnelleren Primärwellen (P-Wellen) und Sekundärwellen (S-Wellen), die maßgeblich für die Verursachung von Schäden sind, während Oberflächenwellen längere und an der Oberfläche spürbare Bewegungen verursachen [13].

Die Richterskala, entwickelt in den 1930er Jahren von Charles F. Richter, ist eine logarithmische Skala zur Messung der Magnitude von Erdbeben. Jede Einheit auf dieser Skala repräsentiert eine zehnfache Erhöhung der Amplitude der seismischen Wellen und eine ungefähr 31,6-fache Zunahme der

freigesetzten Energie. Dieses Messinstrument erlaubt es, Erdbeben nach ihrer Stärke zu klassifizieren: Beben mit einer Magnitude von weniger als 3,0 werden allgemein als kaum spürbar eingestuft, solche zwischen 3,0 und 4,0 als leicht, zwischen 4,0 und 5,5 als moderat, und Beben mit einer Magnitude von 5,5 oder höher gelten als stark.

Neben der Magnitude spielt die Intensität eine wesentliche Rolle bei der Beurteilung der Auswirkungen eines Erdbebens. Während die Magnitude die gesamte freigesetzte Energie eines Bebens misst, beschreibt die Intensität dessen Effekte an einem bestimmten Standort, basierend auf den beobachteten Schäden, Veränderungen an der Erdoberfläche und den Reaktionen der Menschen. Diese Differenzierung ermöglicht es, die physischen und gesellschaftlichen Folgen von Erdbeben genauer zu erfassen und zu verstehen.

Die fortschrittliche Entwicklung seismischer Instrumentierung und Methoden, einschließlich der Anwendung von Magnituden- und Intensitätsskalen, ermöglicht eine präzise Lokalisierung von Erdbebenherden und eine detaillierte Bewertung der Bebenstärke. Dies trägt zu einem tieferen Verständnis der tektonischen Prozesse bei und verbessert die Genauigkeit der Erdbebenvorhersage, was für die Risikominderung und Notfallplanung von entscheidender Bedeutung ist [27, 61].

2.1.3 *Auswirkungen, Bedeutung und Herausforderungen der Erdbebenvorhersage*

Erdbeben generieren erhebliche Risiken, vor allem in städtischen Ballungsräumen, mit möglichen Konsequenzen, die von Bodenverformungen bis hin zu Tsunamis und Erdrutschen reichen. Die Vielfalt der möglichen Schäden unterstreicht die Notwendigkeit eines umfassenden Ansatzes für die Erdbebenvorsorge. Dieser Ansatz sollte neben der Verstärkung der baulichen Infrastruktur zur Erhöhung der seismischen Widerstandsfähigkeit auch die Einführung von Frühwarnsystemen beinhalten. Interdisziplinäre Forschung, die technische und sozialwissenschaftliche Perspektiven vereint, ist entscheidend für die Entwicklung im Bereich der Frühwarnung. Zur Vorhersage von Erdbeben konzentriert sich die Forschung auf die Verfeinerung von Frühwarnsystemen durch die Integration seismologischer und geophysikalischer Techniken. Diese Methoden streben danach, die Vorhersagepräzision seismischer Aktivitäten zu verbessern und die Zeitfenster für Warnungen zu maximieren. Solche Fortschritte sollen nicht nur die Genauigkeit erhöhen, sondern auch den Gemeinden mehr Vorbereitungszeit bieten, um die potenziellen Schäden und Risiken zu minimieren [56].

Die Prognose von Erdbeben über längere Zeiträume hinweg bleibt indessen eine große Herausforderung, hauptsächlich aufgrund der Komplexität der unterliegenden geophysikalischen Prozesse und der derzeitigen Unzulänglichkeit in der Identifizierung verlässlicher präseismischer Phänomene. Obschon historische Erdbebendatensätze eine probabilistische Risikoabschätzung erlauben, stellt diese nur einen Teilaspekt der erforderlichen Planungsgrundlage dar [57].

Traditionelle Vorhersagemethoden, die ein Spektrum von Anzeichen umfassen – von anomalem Tierverhalten über präseismische seismische Aktivität bis hin zu geophysikalischen und gaschemischen Indikationen –, leiden unter eingeschränkter Zuverlässigkeit und Präzision. Die methodischen Schwächen liegen in der variablen Effektivität und der schwierigen Reproduzierbarkeit der Vorhersageindikatoren. Das Fehlen eines einheitlichen, wissenschaftlich validierten Rahmens zur Integration dieser Vorläufer in präzise Prognosemodelle unterstreicht die Limitationen der traditionellen Ansätze [25].

Die im Folgenden aufgeführte Tabelle bietet eine systematische Übersicht der herkömmlichen Methoden zur Erdbebenvorhersage, einschließlich einer kritischen Bewertung ihrer Effektivität:

Tabelle 2.1: Kritische Bewertung traditioneller Erdbebenvorhersagemethoden

Methode	Beschreibung	Effektivität
Tierverhalten	Beobachtungen von Verhaltensänderungen bei Tieren als mögliche präseismische Indikatoren.	Variabel: Unterschiedliche Interpretationen und fehlende Konsistenz in den Ergebnissen
Seismische Aktivität	Untersuchung kleiner seismischer Ereignisse, die als Vorbeben einem Hauptbeben vorausgehen können.	Mäßig: Gute Vorhersage bei einigen Gebieten, aber begrenzte Reichweite und schwierige Auswertung
Geophysikalische Änderungen	Erfassung von Anomalien in geophysikalischen Parametern, wie magnetischen Feldern und Bodenleitfähigkeit, die auf bevorstehende seismische Aktivitäten hinweisen könnten.	Niedrig bis mäßig: Mangelnde Konsistenz in den Daten
Gasemissionen	Messung von Anomalien in der Freisetzung von Gasen, insbesondere Radon, aus der Erdoberfläche, die mit tektonischen Bewegungen in Verbindung stehen.	Niedrig: Schwache Korrelation zwischen Gasemissionen und seismischer Aktivität

Die traditionellen Methoden der Erdbebenvorhersage basieren auf der Annahme, dass bestimmte physikalische, chemische oder biologische Veränderungen in der Umwelt auf bevorstehende seismische Ereignisse hinweisen können. Jedoch ist die Effektivität dieser Methoden durch die hohe Variabilität und die oft fehlende Reproduzierbarkeit der Vorhersagesignale be-

grenzt. Darüber hinaus ist die kausale Verbindung zwischen diesen Anzeichen und den darauffolgenden Erdbeben nicht immer eindeutig oder wissenschaftlich fundiert, was ihre Zuverlässigkeit als Vorhersageinstrumente weiter einschränkt [25].

Die Herausforderungen bei der traditionellen Erdbebenvorhersage unterstreichen die Notwendigkeit, fortgeschrittene wissenschaftliche Methoden und Technologien zu entwickeln und einzusetzen. Insbesondere bieten moderne Ansätze, die auf der Analyse großer Datenmengen und der Anwendung von ML basieren, das Potenzial, die Genauigkeit und Zuverlässigkeit von Erdbebenvorhersagen zu verbessern. Diese innovativen Methoden ermöglichen eine tiefere Einsicht in die komplexen geophysikalischen Prozesse, die Erdbeben zugrunde liegen, und tragen zur Entwicklung präziserer und zuverlässigerer Vorhersagemodelle bei.

2.2 ANWENDUNG GESETZLICHER GRUNDLAGEN UND SEISMOLOGISCHER TECHNIKEN

Die Seismologie nutzt Theorien wie das Gutenberg-Richter-Gesetz, Quieszenz und GMPEs, um Erdbeben zu verstehen und ihre Auswirkungen vorherzusagen. Diese Konzepte sind essentiell für die Risikoabschätzung und Erdbebenvorhersage.

2.2.0.1 *Das Gutenberg-Richter-Gesetz*

Das Gutenberg-Richter-Gesetz ist ein fundamentales Konzept in der Seismologie, das eine wichtige Rolle bei der Vorhersage von Erdbeben spielt. Es beschreibt die Beziehung zwischen der Anzahl von Erdbeben und deren Magnituden in einem bestimmten Gebiet. Die Grundidee ist, dass kleinere Erdbeben in einem Gebiet häufiger auftreten als größere. Diese Beziehung wird mathematisch ausgedrückt durch die Formel:

$$\log N_i = a - bM_i$$

Hierbei ist N_i die Anzahl der Erdbeben mit einer Magnitude gleich oder größer als M_i , a ein Maß für die seismische Aktivität in einem Gebiet und b ein Indikator für das Verhältnis zwischen der Häufigkeit von kleinen und großen Erdbeben.

2.2.0.2 *Quieszenz*

Quieszenz, eine Phase der verringerten seismischen Aktivität, kann oftmals als Vorläufer größerer Erdbeben auftreten. Diese Ruheperiode lässt sich durch das Omori-Gesetz, welches die Abklingrate von Nachbeben beschreibt, im Kontext der seismischen Energieverteilung verstehen. Eine solche Phase deutet möglicherweise auf die anstehende Freisetzung akkumulierter seismischer Energie in Form eines signifikanten seismischen Ereignisses hin. Die

Fähigkeit, Quieszenzphasen präzise zu identifizieren, ist daher ein kritischer Aspekt der Erdbebenvorhersage, da sie als potentielle Warnsignale für bevorstehende seismische Ereignisse fungieren können [29].

2.2.0.3 *Ground Motion Prediction Equations (GMPEs)*

Ground Motion Prediction Equations (GMPEs) sind fundamentale Werkzeuge in der Seismologie, die zur Abschätzung der Bodenbewegung an einem bestimmten Standort als Funktion der Magnitude eines Erdbebens, der Entfernung zum Epizentrum und gegebenenfalls weiterer lokaler geologischer und seismologischer Bedingungen verwendet werden. GMPEs basieren typischerweise auf regressionsanalytischen Ansätzen, die historische Daten von Erdbeben und den damit verbundenen Bodenbewegungen auswerten, um Beziehungen zwischen den Eingangsparametern und der Bodenbewegung herzustellen. Die allgemeine Form einer GMPE kann wie folgt ausgedrückt werden:

$$\ln(Y) = f(M, R, \theta) + \epsilon,$$

wobei Y die Bodenbewegung repräsentiert, definiert durch Indikatoren wie Spitzenbodenbeschleunigung, die die höchste Rate der Bodenbewegung angibt, und spektrale Beschleunigung, die die Reaktion von Strukturen verschiedener Eigenfrequenzen auf Erdbebenbewegungen beschreibt. M steht für die Magnitude des Erdbebens, R für die Distanz zum Epizentrum und θ für zusätzliche lokale geologische und seismische Faktoren. Der Term ϵ repräsentiert den zufälligen Fehler der Schätzung, der die natürliche Variabilität der Daten und die Unvollkommenheiten des Modells widerspiegelt. GMPEs ermöglichen es Ingenieuren und Wissenschaftlern, die erwartete Bodenbewegung für neue oder hypothetische Erdbeben vorherzusagen.

MASCHINELLES LERNEN IN DER ERDBEBENVORHERSAGE

3.1 PRÄZISIERUNG DES BEGRIFFS MACHINE LEARNING

Machine Learning bildet einen integralen Bestandteil der Informatik und widmet sich der Entwicklung von Algorithmen, die das automatisierte Lösen komplexer Probleme ermöglichen, welche traditionelle Programmiermethoden nicht effizient bewältigen können. Im Gegensatz zu herkömmlichen Ansätzen, die auf der expliziten Kodierung von Lösungen basieren, zeichnen sich ML-Algorithmen durch ihre Fähigkeit aus, aus Daten zu lernen und Muster eigenständig zu erkennen. Diese Fähigkeit zur Adaption und zum Lernen aus Erfahrung macht ML besonders wertvoll für Anwendungen, bei denen direkte Programmierlösungen schwer zu definieren sind.

ML-Techniken lassen sich grundsätzlich in zwei Hauptkategorien einteilen: überwachtes und unüberwachtes Lernen. **Überwachtes Lernen** bezieht sich auf den Prozess, bei dem Algorithmen anhand von Beispieldatensätzen trainiert werden, die sowohl Eingabedaten als auch die zugehörigen Ausgaben enthalten. Das Ziel ist die Entwicklung eines Modells, das in der Lage ist, korrekte Ausgaben für neue, unbekannte Eingaben vorherzusagen. Diese Methode ist besonders effektiv für Klassifikations- und Regressionsaufgaben, bei denen die Ausgabe entweder eine Kategorie (Klassifikation) oder ein kontinuierlicher Wert (Regression) ist. **Unüberwachtes Lernen** hingegen operiert ohne vorher festgelegte Ausgaben und konzentriert sich darauf, in den Daten inhärente Strukturen oder Muster zu entdecken. Techniken des unüberwachten Lernens wie Clustering oder Dimensionsreduktion werden eingesetzt, um Zusammenhänge und Gruppierungen in den Daten zu identifizieren, die nicht sofort ersichtlich sind. Dieser Ansatz eignet sich besonders für explorative Datenanalyse und das Entdecken von Datenstrukturen ohne vorherige Kenntnisse.

In der seismologischen Forschung und Erdbebenvorhersage spielen ML-Methoden eine entscheidende Rolle bei der Identifizierung von Mustern und Korrelationen innerhalb umfangreicher seismischer Datensätze. Überwachtes Lernen erlaubt die genaue Prognose von Erdbebenmerkmalen, einschließlich Zeit, Ort und Magnitude, unter Verwendung historischer Datenmuster. Unüberwachtes Lernen trägt zur Erkennung von bisher unbekanntem Zusammenhängen und potenziellen Vorläufern seismischer Aktivitäten bei. Zusammen bieten diese ML-Techniken die Möglichkeit, die Genauigkeit der Erdbebenvorhersage signifikant zu verbessern und tragen so zur Risikominderung und effektiveren Vorbereitung auf seismische Ereignisse bei [35, 48].

3.2 POTENZIALE UND HERAUSFORDERUNGEN IN DER ERDBEBENVORHERSAGE

POTENZIALE

- **Multidimensionale Datensynthese für erweiterte Vorhersagemodelle:** ML revolutioniert die seismologische Forschung durch die gründliche Analyse einer Vielzahl von Datenquellen. Indem seismische Wellenformdaten, geologische und tektonische Informationen sowie historische Erdbebendatensätze integriert werden, ermöglicht ML die Entwicklung von Vorhersagemodellen, die durch ihre fortschrittliche Genauigkeit bestechen. Die Kapazität des ML, historische und aktuelle seismische Daten mit außerordentlicher Präzision zu analysieren, ist von entscheidender Bedeutung für die Steigerung der Vorhersagequalität und die Ausgestaltung wirksamer Erdbebenfrühwarnsysteme. Insbesondere verbessert ML die Fähigkeit zur automatischen Erkennung und präzisen Lokalisierung von seismischen Ereignissen erheblich – selbst in Umgebungen, die durch starkes Rauschen oder andere störende Faktoren gekennzeichnet sind. Durch die Einbettung von ML in die seismologische Datenanalyse werden nicht nur die Möglichkeiten der Datenauswertung signifikant erweitert, sondern auch die Effizienz und Effektivität von Erdbebenvorhersagen und Frühwarnmechanismen erheblich verbessert. Dies verdeutlicht das transformative Potenzial des ML in der seismologischen Forschung und in der Anwendung von Erdbebenvorhersage- und Frühwarnsystemen [46, 55].
- **Integration seismologischer Prinzipien in ML für präzisere Vorhersagen:** Die Integration fundamentaler seismologischer Konzepte, einschließlich des Gutenberg-Richter-Gesetzes und des Phänomens der Quieszenz, in ML-Modelle markiert einen innovativen Ansatz zur Verfeinerung der Erdbebenvorhersage. Diese Modelle nutzen umfangreiche Datensätze vergangener seismischer Ereignisse, um die komplexen Beziehungen zwischen verschiedenen seismologischen Parametern zu entschlüsseln. Ein tieferes Verständnis für diese grundlegenden seismologischen Theorien ermöglicht es, durch ML präzise Vorhersagemodelle zu entwickeln, die subtile Muster und Anomalien in den Daten erkennen. Die Kombination von ML mit seismologischem Fachwissen eröffnet neue Wege, bisher unentdeckte Zusammenhänge innerhalb der seismischen Aktivitätsdaten zu identifizieren, was zu einer signifikanten Verbesserung der Vorhersagegenauigkeit führt. Insbesondere die Fokussierung auf spezifische seismologische Phänomene wie die Quieszenzphasen und die Anwendung des Gutenberg-Richter-Gesetzes innerhalb von ML-Modellen verspricht ein erhöhtes Potenzial zur Identifikation von Mustern, die für präzisere und zuverlässigere Erdbebenvorhersagen entscheidend sind [53].

HERAUSFORDERUNGEN

- **Datenqualität und Modellinterpretation** Eine zentrale Herausforderung beim Einsatz von ML für die Erdbebenvorhersage ist die Gewährleistung der Datenqualität. Die Präzision und Zuverlässigkeit von ML-Modellen sind direkt abhängig von der Vollständigkeit und Genauigkeit der zugrunde liegenden Datensätze. Historische Erdbebenaufzeichnungen, die oft lückenhaft oder inkonsistent sind, begrenzen die Fähigkeit der Modelle, verlässliche Vorhersagen zu treffen. Eine weitere Herausforderung besteht in der Überanpassung, bei der Modelle zu spezifisch auf die Trainingsdaten zugeschnitten sind und somit ihre Vorhersagekraft für unbekannte Daten verlieren. Effektive Strategien zur Überwindung dieser Probleme beinhalten die sorgfältige Validierung und Erweiterung von Erdbebenkatalogen sowie den Einsatz unüberwachter Lernmethoden, die fähig sind, auch ohne explizite Vorgaben tieferliegende Datenmuster zu erkennen und zu nutzen [10].
- **Einfluss geophysikalischer Prozesse** Eine weitere Herausforderung für ML in der Erdbebenvorhersage besteht darin, die komplexe Dynamik geophysikalischer Prozesse adäquat zu modellieren. Beispielsweise erfordern Konzepte wie das Gutenberg-Richter-Gesetz, das Verständnis von Quieszenzperioden oder die Anwendung von GMPEs ein tiefes Verständnis seismologischer Zusammenhänge, das schwer in ML-Modelle zu integrieren ist. Die Grenzen aktueller Instrumentierung und Datenerfassungsmethoden erschweren zusätzlich die präzise Erfassung und Interpretation der notwendigen Daten. Während ML das Potenzial hat, neue Muster in den Daten zu erkennen, bleibt die Herausforderung, diese komplexen und oft nicht-linearen geophysikalischen Prozesse in ML-Modellen zu kodifizieren und präzise Vorhersagen zu generieren, ohne die realen physikalischen Prinzipien zu übersehen [33].
- **Die inhärente Unvorhersehbarkeit seismischer Ereignisse** Trotz erheblicher Fortschritte in der seismologischen Forschung bleibt die präzise Vorhersage von Erdbebenereignissen eine der größten wissenschaftlichen Herausforderungen. ML-basierte Modelle haben zwar Potenzial gezeigt, die Genauigkeit von Erdbebenvorhersagen zu verbessern, jedoch steht der definitive Beweis ihrer Effektivität über verschiedene geografische Regionen hinweg noch aus. Derzeit gibt es keine schlüssigen Belege dafür, dass ML-Modelle die Fähigkeit besitzen, Erdbebenereignisse zuverlässig vorherzusagen. Diese Unsicherheit spiegelt die fundamentale Komplexität und Unvorhersehbarkeit seismischer Prozesse wider, die von zahlreichen variablen geophysikalischen Bedingungen beeinflusst werden. Angesichts des frühen Stadiums der Forschung in diesem Bereich eröffnet sich ein breites Feld für Innovationen, um die Grenzen der Vorhersagegenauigkeit zu erweitern und die Methoden kontinuierlich zu verfeinern [46].

3.3 INDIKATOREN FÜR DIE ERDBEBENVORHERSAGE DURCH MACHINE LEARNING

Die Genauigkeit der Erdbebenvorhersage mittels ML hängt entscheidend von der umfassenden Analyse seismologischer Daten ab. ML-Modelle ziehen dazu ein breites Spektrum an Informationen heran – angefangen bei Basisdaten der Seismologie über komplexe Berechnungsverfahren bis hin zu detaillierten Modellierungen. ML-Modelle verarbeiten und interpretieren unterschiedlichste Datenpunkte, um sowohl die Wahrscheinlichkeit als auch die möglichen Konsequenzen von Erdbeben vorherzusagen. Indem die Modelle vielschichtige Datenquellen einbeziehen, erlangen sie ein vertieftes Verständnis für seismische Risiken, was die Vorhersagepräzision signifikant erhöht und zu fundierteren Prognosen führt [34].

3.3.1 Grundlegende seismische Daten

Die Genauigkeit von Erdbeben-Vorhersagemodellen beruht auf der systematischen Erfassung und eingehenden Analyse essenzieller seismischer Daten. Wesentliche Informationen, die für die Modellbildung benötigt werden, umfassen:

- **Zeitstempel des Ereignisses:** Essentiell für die Erkennung zeitlicher Muster in der Sequenz seismischer Aktivitäten.
- **Lokalisierung des Epizentrums:** Fundamental zur Bestimmung des Erdbebenursprungs und zur Untersuchung tektonischer Dynamiken.
- **Magnitude:** Zentral zur Bewertung der bei einem Erdbeben freigesetzten Energie und zur Abschätzung des Schadenspotenzials.
- **Epizentrale Distanz:** Misst die Entfernung zwischen dem Epizentrum und einer Beobachtungsstation, entscheidend für die Abschätzung der Bebenauswirkungen.
- **Tiefe des Hypozentrums:** Bestimmt die Position des Bebenherds unterhalb der Erdoberfläche, wichtig für das Verständnis der Bebenursache.

3.3.2 Fortgeschrittene seismologische Berechnungen

Zur Verfeinerung der Vorhersagefähigkeit von ML-Modellen werden fortgeschrittene seismologische Berechnungen herangezogen, darunter:

GUTENBERG-RICHTERS A- UND B-WERTE Die Parameter a und b des Gutenberg-Richter-Gesetzes (siehe Abschnitt 2.2.0.1) sind fundamentale metrische Inputs, die Einblicke in die Häufigkeitsverteilung von Erdbeben unterschiedlicher Magnituden bieten und somit zur Vorhersage der Wahrscheinlichkeit größerer seismischer Ereignisse beitragen.

ABWEICHUNG VOM GUTENBERG-RICHTER-GESETZ, η Die Abweichung η von den tatsächlichen Daten im Vergleich zum inversen Gutenberg-Richter-Gesetz zeigt auf, inwiefern das Gesetz die reale Verteilung der Erdbebenmagnituden abbildet.

$$\eta = \frac{\sum_{i=1}^n (\log N - a - bM)^2}{n-1} \quad (5) \quad (3.1)$$

STANDARDABWEICHUNG DES b -WERTES, σ_b Die Standardabweichung des b -Wertes (σ_b) gibt an, wie stark die individuellen b -Werte um den Mittelwert streuen.

$$\sigma_b = 2.3 \cdot b^2 \cdot \sqrt{\frac{\sum_{i=1}^n (M_i - M_{\text{mean}})^2}{n(n-1)}} \quad (6) \quad (3.2)$$

SEISMISCHE ENERGIEABGABE, dE Die seismische Energie (dE) ist die Energie, die in Form kleiner Erdbeben aus dem Boden freigesetzt wird. Wenn die Energieabgabe stoppt, wird dieses Phänomen als Ruhephase bezeichnet, die in Form eines größeren Erdbebens auftreten kann. Der Zustand der Ruhe kann zu einer Reduktion der seismischen Rate der Region führen und einen abnehmenden b -Wert verursachen.

$$dE^{\frac{1}{2}} = \left(\frac{1}{T} \sum \left(10^{11.8+1.5M} \right) \right)^{\frac{1}{2}} \quad (2) \quad (3.3)$$

ZEIT DER n EREIGNISSE, T Zeit (T) in Tagen, während der n seismische Ereignisse vor dem Erdbeben aufgetreten sind.

$$T = t_n - t_i \quad (3) \quad (3.4)$$

MITTLERE MAGNITUDE, M_{MEAN} Die mittlere Magnitude (M_{mean}) bezieht sich auf den Mittelwert von n Ereignissen.

$$M_{\text{mean}} = \frac{\sum_{i=1}^n M_i}{n} \quad (4) \quad (3.5)$$

MAGNITUDE-DEFIZIT, ΔM Das Magnitude-Defizit (ΔM) ist die Differenz zwischen der maximal beobachteten und der maximal erwarteten Erdbebenmagnitude.

$$M_{\text{max.expected}} = \frac{a}{b} \quad (7) \quad (3.6)$$

$$\Delta M = M_{\text{max.actual}} - M_{\text{max.expected}} \quad (8) \quad (3.7)$$

WIEDERKEHRPERIODE, T_r Die Wiederkehrperiode (T_r) beschreibt die durchschnittliche Zeit zwischen zwei Ereignissen. Dieser Parameter ist entscheidend für die Bewertung der Erdbebengefährdung in einem bestimmten Gebiet.

AUSBREITUNGSZONE, z Die Ausbreitungszone (z) gibt die geografische Region an, in der seismische Wellen aus einem Erdbebenereignis empfangen werden können.

EFFIZIENTE ERDBEBENERKENNUNG DURCH P- UND S-WELLEN Die Analyse von P- und S-Wellen spielt eine zentrale Rolle in der Erdbebenerkennung. Wesentlich hierbei ist die Differenz ihrer Ankunftszeiten ($\Delta t = t_s - t_p$), die entscheidend für die Ermittlung der Distanz zum Epizentrum ist:

$$\Delta t = t_s - t_p \quad (3.8)$$

Die zeitliche Differenz zwischen dem Eintreffen von P- und S-Wellen ist entscheidend für die Lokalisierung des Erdbebenhypozen-trums und bietet Einsichten in die geophysikalischen Zustände unterhalb der Erdoberfläche. Variationen in Amplitudenverhältnissen und Frequenzspek-tren dieser Wellen deuten auf Änderungen im Untergrund hin, die mögliche Vorboten von Erdbeben sein können.

Die detaillierte Analyse von Wellenformmerkmalen wie Amplitude, Frequenz und Dauer ist für die Beurteilung von Erdbeben wesentlich. Wichtige Parameter wie die dominante Periode (τ_p), maximale dominante Periode (τ_{pmax}) und durchschnittliche Periode (τ_c) liefern essenzielle Informationen für Frühwarnungen und die Intensitätsabschätzung von Beben. Moderne seismische Technologien und Algorithmen ermöglichen die genaue Bestimmung der Ankunftszeiten von P- und S-Wellen, wodurch zuverlässige Frühwarnsysteme unterstützt werden. Die Anwendung fortschrittlicher Analysemethoden auf umfassende seismische Datensätze erlaubt eine noch nie dagewesene Präzision in der Erfassung und Bewertung seismischer Ereignisse.

INTEGRATION VON GMPES In ML-Modellen können GMPES (siehe Abschnitt 2.2.0.3) genutzt werden, um die Auswirkungen eines prognostizierten Erdbebens auf verschiedene Standorte zu schätzen. Die Parameter der GMPES, insbesondere die Regressionskoeffizienten der Funktion $f(M, R, \theta)$, werden als Inputvariablen in ML-Modelle integriert, um die Vorhersagegenauigkeit bezüglich der Bodenbewegung und daraus resultierender potenzieller Schäden zu erhöhen.

3.4 MACHINE LEARNING METHODEN IN DER ERDBEBENVORHERSAGE

3.4.1 Überwachte Lernmethoden

3.4.1.1 Support Vector Machine und Support Vector Regression

SUPPORT VECTOR MACHINE Support Vector Machine (SVM) ist eine überwachte Lernmethode, die durch die Konstruktion eines Hyperplans zur Klassifizierung von Daten in einem hochdimensionalen Raum arbeitet. Der Hyperplan wird so gewählt, dass er den größten Abstand (Margin) zwischen den Datenpunkten unterschiedlicher Klassen aufweist. Für nicht linear separierbare Daten werden Kernelfunktionen eingesetzt, um die Daten in einen Raum höherer Dimension zu transformieren, wo eine lineare Trennung möglich ist. Die Effizienz von SVMs in der Erdbebenvorhersage hängt von der Auswahl der Kernelfunktionen und Hyperparameter ab. SVMs bieten eine hohe Generalisierungsfähigkeit, sind aber bei großen Datensätzen rechenintensiv [42].

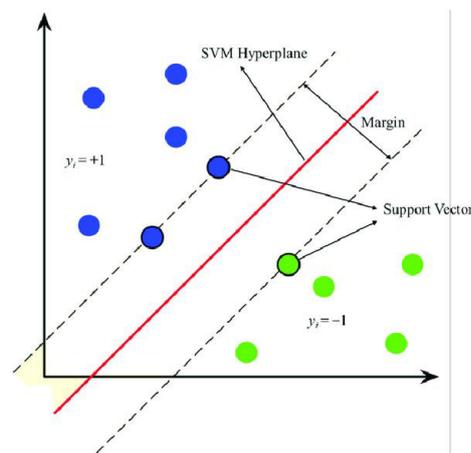


Abbildung 3.1: SVM-Klassifikationsverfahren

SUPPORT VECTOR REGRESSION Support Vector Regression (SVR) adaptiert das SVM-Prinzip für Regressionsaufgaben, indem es sich auf Vorhersagen innerhalb einer bestimmten Fehlertoleranz (ϵ -Insensitivitätszone) konzentriert. Diese Methode eignet sich besonders für die präzise Vorhersage von Erdbebenparametern wie Zeitpunkt, Magnitude und Epizentrum. SVR nutzt Kernelfunktionen, um mit nichtlinearen Daten umzugehen, was die Methode effektiv für die Analyse komplexer seismischer Muster macht.

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\zeta_i + \zeta_i^*) \quad (3.9)$$

SVR zeigt sich als effektives Instrument zur Modellierung der nichtlinearen Beziehungen zwischen seismischen Variablen, wodurch es für die präzise Prognose von Erdbebencharakteristika unverzichtbar wird. Ihre Einsatz-

möglichkeiten umfassen unter anderem die Abschätzung der Erdbebenmagnitude und die Vorhersage von Nachbebensequenzen [9].

3.4.1.2 Diskriminanzanalyse

Die Diskriminanzanalyse ist eine traditionelle statistische Methode zur Klassifizierung von Gruppen durch Muster in ihren Variablen. Im Gegensatz zu SVM und SVR, die auf der Maximierung des Abstands zwischen den Klassen durch einen Hyperplan basieren, fokussiert die Diskriminanzanalyse auf die direkte Trennung von Gruppen anhand charakteristischer Variablen. Dies geschieht mittels Diskriminanzfunktionen, die eine effektive Gruppentrennung im Merkmalsraum erzielen. Während SVM und SVR für komplexe, nichtlineare Daten und hohe Generalisierungsansprüche geeignet sind, zeichnet sich die Diskriminanzanalyse durch ihre Einfachheit und geringere Rechenintensität aus, was sie für klar strukturierte Gruppendaten prädestiniert. Sie bietet einen direkten, weniger rechenintensiven Zugang zur Unterscheidung zwischen Gruppen und ist besonders nützlich, wenn die Gruppenzugehörigkeit durch deutliche Variablenmuster definiert wird [19].

3.4.1.3 Lineare, logistische und generalisierte lineare Modelle

LINEARE REGRESSION Die lineare Regression (LinR) ist ein fundamentales statistisches Verfahren zur Modellierung der Beziehung zwischen einer abhängigen Variablen y und einer oder mehreren unabhängigen Variablen x_i . Sie ist definiert durch die Gleichung:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon, \quad (3.10)$$

wobei β_0 den Achsenabschnitt, β_i die Steigungskoeffizienten, die die Stärke des Einflusses der jeweiligen unabhängigen Variablen auf y anzeigen, und ϵ den Fehlerterm darstellen. In der Seismologie ermöglicht die lineare Regression die Quantifizierung des Einflusses geophysikalischer und historischer Variablen auf kontinuierliche Zielgrößen wie die Erdbebenstärke [31].

LOGISTISCHE REGRESSION Die logistische Regression (LogR) wird zur Vorhersage der Wahrscheinlichkeit des Eintretens eines Ereignisses verwendet und ist besonders nützlich für binäre Klassifikationsprobleme. Im Gegensatz zur linearen Regression, die kontinuierliche Ausgaben liefert, modelliert die logistische Regression die Wahrscheinlichkeit $P(Y = 1)$ eines binären Ausgangs Y mittels der logistischen Funktion:

$$P(Y = 1) = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad (3.11)$$

mit $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$, wobei z die lineare Kombination der Eingangsvariablen repräsentiert. Diese Methode ist effektiv für die Be-

stimmung der Eintrittswahrscheinlichkeit seismischer Ereignisse basierend auf prädisponierenden Faktoren [31].

GENERALISIERTE LINEARE MODELLE UND RIDGE REGRESSION Generalisierte Lineare Modelle (GLM) bieten eine flexible Erweiterung der linearer und logistischer Regression, indem sie verschiedene Verteilungen für die Zielvariablen und eine Link-Funktion zur Verknüpfung der Vorhersage mit diesen Verteilungen zulassen. Diese Anpassungen ermöglichen es GLMs, eine breitere Palette realer Daten effektiv zu modellieren, einschließlich der Verteilung seismischer Ereignisse.

Ein kritischer Aspekt der Modellierung, insbesondere in der Vorhersage von Erdbeben, ist die Vermeidung von Überanpassung, bei der Modelle zu spezifisch auf Trainingsdaten zugeschnitten werden und dadurch ihre Vorhersagekraft für neue Daten verlieren. Hier kommt die Ridge Regression ins Spiel. Sie erweitert lineare Modelle durch die Einführung eines Regularisierungsterms, der die Größe der Regressionskoeffizienten begrenzt. Dieser Term wird direkt in die Verlustfunktion des Modells integriert, die das Maß des Fehlers zwischen den prognostizierten und tatsächlichen Werten quantifiziert. Die Ridge Regression wirkt der Überanpassung entgegen, indem sie die Komplexität des Modells durch diesen Regularisierungsterm steuert. Der Regularisierungsparameter α spielt eine entscheidende Rolle in diesem Prozess. Ein höherer Wert von α erhöht die Stärke der Regularisierung, was zu einer stärkeren Glättung der Modellkoeffizienten führt. Dies hilft, das Modell zu vereinfachen und die Generalisierungsfähigkeit zu verbessern, allerdings auf Kosten der Passgenauigkeit auf den Trainingsdaten [26].

3.4.1.4 *K-Nearest Neighbors*

Der K-Nearest Neighbors (KNN)-Algorithmus ist ein intuitives und effektives Modell, das in der seismologischen Forschung zur Klassifizierung und Prognose von Erdbeben durch Analyse der Ähnlichkeit zu historischen Ereignissen eingesetzt wird. KNN beruht auf der Prämisse, dass seismische Ereignisse mit ähnlichen Merkmalen im Merkmalsraum nah beieinander gruppiert sind.

Die Implementierung von KNN umfasst mehrere Schlüsselschritte:

1. Wählen der Anzahl k der nächsten Nachbarn.
2. Messen der Distanz zum neuen Datenpunkt von allen Trainingsdaten, typisch mit der Euklidischen Distanz.
3. Identifizieren der k nächsten Nachbarn.
4. Zuweisen des neuen Datenpunkts zur häufigsten Klasse seiner k Nachbarn oder Berechnen des Durchschnitts für Regressionsziele.

KNN klassifiziert neue seismische Ereignisse durch Vergleich mit bekannten, wobei die Auswahl von k und der Distanzmetrik entscheidend ist, um

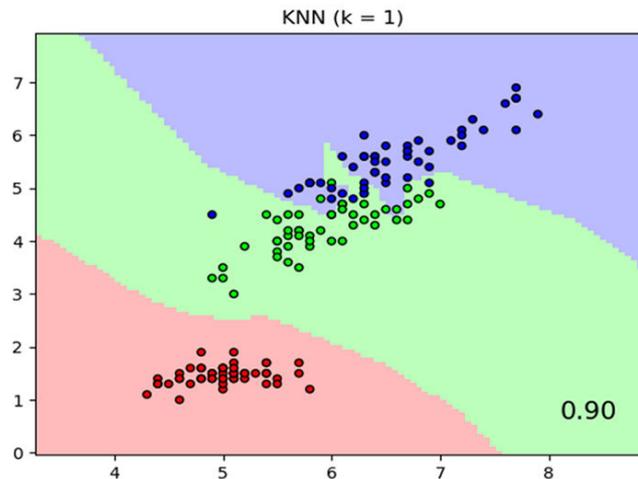


Abbildung 3.2: Illustration des KNN-Klassifikationsverfahrens mit drei Erdbebenklassen

Überanpassung zu minimieren und die Vorhersagepräzision zu maximieren [38].

3.4.1.5 Entscheidungsbäume

Entscheidungsbäume (Decision Trees, DT) sind fundamentale Modelle, die komplexe Entscheidungsprozesse in einer klar strukturierten Baumform visualisieren. Diese Modelle finden hauptsächlich Anwendung in Klassifikationsaufgaben, können aber auch für Regression genutzt werden. In der Seismologie ermöglichen Entscheidungsbäume die effiziente Analyse und Klassifikation von Daten, basierend auf charakteristischen Merkmalen wie tektonischen Bewegungen oder geologischen Eigenschaften. Durch die schrittweise Aufteilung der Daten an Entscheidungsknoten zielen sie auf eine optimale Trennung der Klassen im Merkmalsraum ab. Das Ziel ist die Maximierung der Klarheit und Eindeutigkeit in den Blattknoten [59].

ALGORITHMEN FÜR ENTSCHEIDUNGSBÄUME Für die Entwicklung von Entscheidungsbäumen werden diverse Algorithmen eingesetzt:

- **ID3/C4.5:** Fokussieren auf Maximierung des Informationsgewinns, optimiert für kategorische Daten.
- **CART:** Universell einsetzbar für Klassifikation und Regression, verwendet den Gini-Index zur Bewertung von Datenteilungen.
- **CHAID:** Setzt Chi-Quadrat-Tests ein, um wesentliche Trennpunkte in kategorialen Daten zu finden, geeignet für komplexe Mustererkennung.
- **Tree-AS:** Ermöglicht durch seine iterative Datenpartitionierung eine flexible und präzise Analyse, besonders geeignet für die Klassifizierung in der Seismologie [16].

Mithilfe des Pruning-Verfahrens adressieren DT das Risiko der Überanpassung, indem es irrelevante Teile des Baums entfernt. Dies verbessert die Generalisierbarkeit und Anwendbarkeit des Modells [20].

3.4.1.6 Naive Bayes

Der Naive Bayes (NB)-Algorithmus, basierend auf dem Bayesschen Theorem, ist effektiv in der Klassifikation und zeichnet sich durch die Annahme der Merkmalsunabhängigkeit aus. Das Bayessche Theorem formuliert die posteriori Wahrscheinlichkeit einer Klassenzugehörigkeit $P(C_k|x_1, \dots, x_n)$, basierend auf einem Satz von Prädiktoren x_1, \dots, x_n :

$$P(C_k|x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n|C_k)P(C_k)}{P(x_1, \dots, x_n)}, \quad (3.12)$$

wobei $P(C_k)$ die Vorab-Wahrscheinlichkeit der Klasse C_k und $P(x_1, \dots, x_n|C_k)$ die Likelihood der Merkmale gegeben die Klasse C_k repräsentiert. Der Nenner $P(x_1, \dots, x_n)$ agiert als Normalisierungskonstante.

Die Annahme der Unabhängigkeit zwischen den Merkmalen führt zur Vereinfachung der Likelihood-Berechnung zu:

$$P(x_1, \dots, x_n|C_k) = \prod_{i=1}^n P(x_i|C_k), \quad (3.13)$$

wobei $P(x_i|C_k)$ die Wahrscheinlichkeit des Auftretens des Merkmals x_i unter der Bedingung der Klasse C_k angibt.

Für die finale Klassifizierungsentscheidung wählt NB die Klasse mit der höchsten posteriori Wahrscheinlichkeit. Diese Methode eignet sich vor allem für die schnelle Verarbeitung großer Datenvolumina, wie sie in der Seismologie vorkommen, wo NB zur effizienten Analyse und Mustererkennung in umfangreichen seismischen Datensätzen eingesetzt wird [67].

3.4.1.7 Künstliche Neuronale Netze

Künstliche Neuronale Netze (ANNs) simulieren die Informationsverarbeitung des menschlichen Gehirns und sind effektiv in der Erkennung komplexer Muster innerhalb großer Datenmengen. Sie werden erfolgreich in der Vorhersage seismischer Ereignisse eingesetzt, indem sie eine Vielzahl von Daten, einschließlich historischer Erdbebendaten und tektonischer Informationen, verarbeiten.

Ein ANN setzt sich aus einer Eingabeschicht, mehreren verborgenen Schichten und einer Ausgabeschicht zusammen. Diese Schichten sind durch Neuronen vernetzt, die Inputsignale mittels gewichteter Verbindungen und Biases bearbeiten. Der Lernprozess eines ANNs basiert auf der Anpassung dieser Gewichte und Biases durch den Backpropagation-Algorithmus, um die Abweichung zwischen den Ausgaben des Netzes und den tatsächlichen Zielwerten zu minimieren.

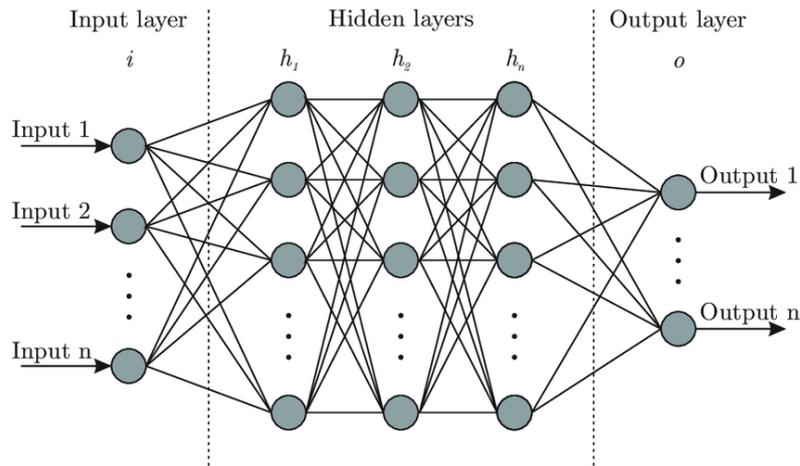


Abbildung 3.3: Schematische Darstellung der Architektur eines ANNs

$$y = b + \sum_{i=1}^m x_i v_i \quad (3.14)$$

Hierbei steht y für die Ausgaben, x_i für die Eingaben, v_i für die Gewichte und b für die Bias-Gewichte. Die Anzahl der Eingabeeinheiten wird durch m repräsentiert.

Die Optimierung des Netzwerks, inklusive der Minimierung der Fehlerfunktion und der Auswahl der Netzwerkarchitektur, ist entscheidend für die Balance zwischen Anpassungsfähigkeit und Generalisierbarkeit, um Überanpassung zu vermeiden. Methoden wie Kreuzvalidierung und Regularisierung, unterstützt durch Hyperparameter-Tuning, sind unerlässlich, um die Leistungsfähigkeit des ANNs zu maximieren [69].

3.4.1.8 Erweiterte neuronale Netzwerkmodelle

Erweiterte neuronale Netzwerkmodelle haben das Potenzial, die Genauigkeit und Zuverlässigkeit der Erdbebenvorhersage erheblich zu verbessern. Durch die Nutzung von Deep Learning (DL) Architekturen können diese Modelle komplexe Muster in seismologischen Daten erkennen, die für herkömmliche analytische Methoden unzugänglich sind. Der spezifische Nutzen dieser fortgeschrittenen Modelle für die Erdbebenvorhersage umfasst: [58]

- **Convolutional Neural Networks (CNNs):** Ihre Fähigkeit, Merkmale aus seismischen Bildern und Karten zu extrahieren, kann für die Analyse von Bodenbewegungen und Bruchlinien verwendet werden. Dies ermöglicht eine präzisere Lokalisierung von potenziellen Erdbebengebieten und eine verbesserte Abschätzung der Bebenstärke.
- **Rekurrente Neuronale Netzwerke (RNNs) und Long Short-Term Memory (LSTM) Modelle:** Diese sind ideal, um zeitliche Abhängigkeiten

in seismischen Zeitreihendaten zu modellieren. Sie können zur Vorhersage von Erdbebenmustern über Zeit verwendet werden, indem sie langfristige Trends und zyklische Muster in den seismischen Aktivitäten erfassen.

- **Feedforward-Neuronale Netzwerke (FFNNs):** Sie dienen als Grundgerüst für komplexe Modelle und können für grundlegende Vorhersagen von Erdbebenereignissen basierend auf einer Reihe von Eingabeparametern eingesetzt werden.
- **Functional Link Artificial Neural Network (FLANN):** Durch ihre nicht-lineare Datenverarbeitung können FLANNs komplexe Beziehungen zwischen seismologischen Datenpunkten erkennen und tragen somit zur Verbesserung der Vorhersagegenauigkeit bei.
- **Hybrid Neural Networks (HNN):** Die Kombination verschiedener Netzwerktypen ermöglicht es, die spezifischen Vorteile jedes Modells zu nutzen. Beispielsweise könnte die Kombination von CNNs für Bilderkennung und RNNs für zeitliche Datenanalyse eine umfassende Analyse seismischer Daten bieten.
- **Adaptive Neuro-Fuzzy Inference System (ANFIS):** ANFIS kann für die Modellierung der Unsicherheit in seismischen Daten verwendet werden, indem es neuronale Netzwerkflexibilität mit Fuzzy-Logik kombiniert. Dies hilft, vage und unsichere Informationen in den Vorhersageprozess einzubeziehen.
- **Extreme Learning Machines (ELM):** Ihre effiziente Verarbeitung großer Datensätze kann genutzt werden, um schnell Veränderungen in seismischen Mustern zu erkennen, was für Frühwarnsysteme von Bedeutung ist.
- **Genetische Programmierung (GP):** GP kann zur Entwicklung von Algorithmen verwendet werden, die spezifische Muster in seismologischen Daten erkennen, was die Identifizierung potenzieller Erdbebenvorläufer erleichtert.
- **Neuronale Dynamikklassifizierung (NDC) und Probabilistische Neuronale Netzwerke (PNN):** Beide Ansätze bieten fortschrittliche Methoden zur Klassifizierung und Vorhersage von Erdbebenereignissen, indem sie die Wahrscheinlichkeitsdichtefunktionen seismischer Aktivitäten effektiv modellieren.
- **Evolutionary Polynomial Neural Network (EPNN):** Die Nutzung evolutionärer Algorithmen zur Optimierung von Netzwerkstrukturen kann dazu beitragen, verborgene Muster in seismischen Daten zu entdecken, die herkömmliche Modelle möglicherweise übersehen.

3.4.2 Unüberwachte Lernmethoden

3.4.2.1 K-Means Clustering

K-Means Clustering ist ein essenzielles Verfahren des unüberwachten Lernens, das darauf ausgerichtet ist, umfangreiche Datensätze in kohärente Untergruppen (Cluster) zu organisieren. Wegen ihrer Einfachheit und Effizienz wird die Methode oft genutzt, um verborgene Muster in seismologischen Daten zu finden, etwa durch das Identifizieren ähnlicher seismischer Aktivitäten in Clustern, was zukünftige Ereignisse in diesen Gebieten andeuten könnte. Die Bestimmung der optimalen Anzahl der Cluster (k) ist ein entscheidender Schritt in der Anwendung von K-Means und wird häufig mittels der Elbow-Methode durchgeführt, um den Punkt zu identifizieren, an dem zusätzliche Cluster keinen signifikanten Mehrwert mehr bieten [2].

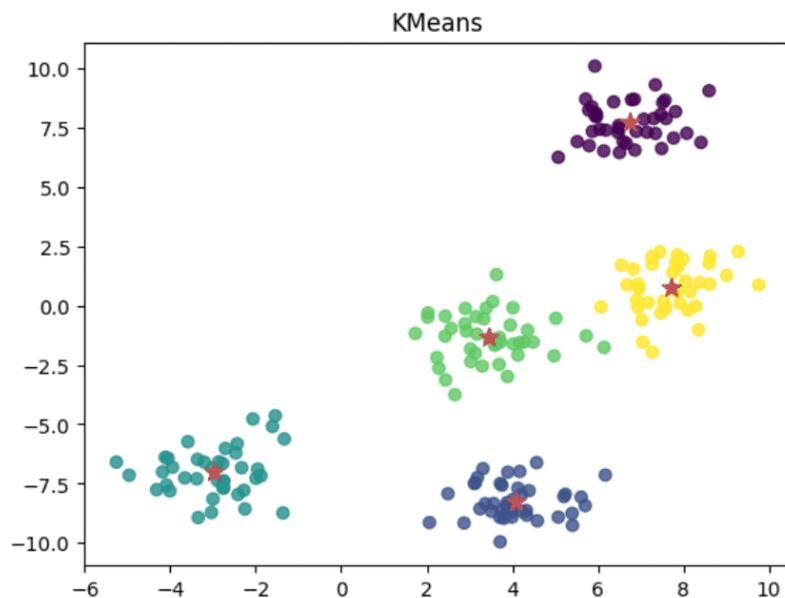


Abbildung 3.4: Visualisierung seismischer Cluster mittels K-Means Clustering Algorithmus

Der K-Means-Algorithmus implementiert einen iterativen Ansatz:

1. Initialisierung von k Zentroiden innerhalb des Datensatzes, oft zufällig gewählt.
2. Zuordnung jedes Datenpunktes zum geografisch nächstgelegenen Zentroid, wobei die Distanz durch gängige Metriken wie die Euklidische Distanz gemessen wird.
3. Aktualisierung der Zentroidpositionen, indem der Durchschnitt aller Punkte, die einem Cluster zugeordnet sind, berechnet wird.
4. Wiederholung der Zuordnung und Aktualisierung der Zentroide, bis keine wesentlichen Veränderungen in der Position der Zentroide mehr auftreten.

3.4.3 Hybrid- und Ensemble-Methoden

Ensemble-Methoden bieten einen fortschrittlichen Ansatz zur Verbesserung der Genauigkeit von Vorhersagemodellen durch die Kombination mehrerer Lernmodelle. Diese Methoden nutzen die Diversität der Vorhersagen, um die Gesamtleistung zu steigern und gleichzeitig das Überanpassungsrisiko zu minimieren. Zu den Kernensemble-Techniken zählen Bootstrap Aggregating (Bagging), Boosting und Stacking [49].

Hybrid-Methoden erweitern das Konzept der Ensemble-Modelle durch die Integration unterschiedlicher maschineller Lernverfahren. Diese Ansätze zielen darauf ab, die spezifischen Stärken einzelner Modelle zu nutzen und somit die Effektivität der Erdbebenvorhersage signifikant zu erhöhen. Ein Beispiel hierfür ist die Kombination von unüberwachten Lernmodellen zur Mustererkennung in seismischen Daten mit überwachten Modellen für die Vorhersage spezifischer Ereignismerkmale [3].

3.4.3.1 Bagging-Methoden

Bagging, eine Abkürzung für Bootstrap Aggregating, verbessert die Stabilität und Genauigkeit von ML-Modellen durch die Erstellung mehrerer Trainingssets aus Bootstrap-Samples des ursprünglichen Datensatzes. Auf jedem dieser Datensätze wird unabhängig ein Modell (Basislerner) trainiert, und die endgültige Vorhersage ergibt sich aus der Aggregation der Vorhersagen dieser Modelle [49].

RANDOM FOREST Random Forest (RF), ein bekanntes Beispiel für Bagging-Techniken, setzt auf DT als Basislerner. Durch das Trainieren jedes Baumes auf einer zufälligen Unterstichprobe der Daten und die mittlere Aggregation ihrer Vorhersagen minimiert RF Überanpassung und verbessert die Prognosegenauigkeit. Diese Technik ist für eine Vielzahl von Aufgaben, einschließlich Klassifikation und Regression, einsetzbar und gilt als eines der robustesten und vielseitigsten Modelle [31].

EXTRA TREES Extra Trees (Extremely Randomized Trees, ERT) erweitern das Konzept von RF durch eine zusätzliche Einführung von Zufälligkeit bei der Auswahl der Trennschwellen für die Features sowie bei der Auswahl der Features für jede Teilung in den Bäumen. Diese Strategie erhöht die Diversifizierung unter den Bäumen und kann zu einer verbesserten Modellgenauigkeit führen, insbesondere in komplexen Datensätzen.

3.4.3.2 Boosting-Methoden

Boosting-Verfahren erhöhen die Vorhersageleistung durch sequenzielles Trainieren schwacher Lerner, wobei jeder folgende Lerner die Fehler des vorherigen korrigiert. Das Ziel ist es, eine starke Vorhersageleistung durch die Kombination der Stärken mehrerer einfacher Modelle zu erreichen [23].

ADABOOST UND LPBOOST AdaBoost (Adaptive Boosting) passt die Gewichte von fehlerhaft klassifizierten Beispielen an, um nachfolgende Lerner stärker auf diese Beispiele zu fokussieren. LPBoost (Linear Programming Boosting) verwendet lineare Programmierung, um die Gewichte der Lerner so anzupassen, dass der Gesamtverlust minimiert wird, wobei beide Ansätze iterativ die Genauigkeit der Gesamtvorhersage verbessern.

GRADIENT BOOSTING-METHODEN Zu den fortgeschrittenen Boosting-Modellen zählen Gradient Boosting Machine (GBM) und ihre Variationen wie XGBoost, LightGBM und CatBoost, die iterative Korrekturen auf den Residuen der Vorhersagen vornehmen. Diese Modelle zeichnen sich durch ihre Effizienz und Genauigkeit aus und finden breite Anwendung in der Praxis.

3.4.3.3 *Stacking-Methoden*

Stacking verwendet die Vorhersagen mehrerer Modelle als Eingangsvariablen für ein Meta-Modell, das die endgültige Vorhersage trifft. Diese Methode kombiniert die Stärken verschiedener Modelle, um die Gesamtgenauigkeit zu verbessern, und ist besonders nützlich in Wettbewerben und komplexen Vorhersageproblemen, wo die Diversität der Modellansätze zu besseren Ergebnissen führen kann [49].

3.5 MODELLBEWERTUNGSMETRIKEN

Die Bewertung von ML-Modellen in der Erdbebenvorhersage erfordert den Einsatz präziser Evaluationsmetriken, die über die bloße Genauigkeit hinausgehen und die Fähigkeit des Modells zur Unterscheidung zwischen tatsächlich stattfindenden und ausbleibenden seismischen Ereignissen erfassen. Solche Metriken sind unerlässlich, um die Effektivität der Modelle zu beurteilen und leiten die Feinabstimmung der Vorhersagealgorithmen an. Die sorgfältige Auswahl dieser Metriken ist fundamental für das Fortschreiten und die Verfeinerung von Erdbebenvorhersagemodellen, wobei Klassifikationsmetriken Einblicke in die Verlässlichkeit und Regressionsmetriken detaillierte Bewertungen der Vorhersagepräzision ermöglichen.

3.5.1 Klassifikationsmetriken

Klassifikationsmetriken bewerten, wie gut ein Modell das Auftreten oder Nichtauftreten von Erdbeben vorhersagen kann. Sie basieren auf der Analyse von True Positives (TP), True Negatives (TN), False Positives (FP) und False Negatives (FN). Aus diesen Metriken lassen sich weitere wichtige Maße berechnen [22]:

$$\text{Genauigkeit} = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.15)$$

$$\text{Sensitivität} = \frac{TP}{TP + FN} \quad (3.16)$$

$$\text{Präzision} = \frac{TP}{TP + FP} \quad (3.17)$$

$$\text{F1-Score} = \frac{2 \times \text{Präzision} \times \text{Sensitivität}}{\text{Präzision} + \text{Sensitivität}} \quad (3.18)$$

$$\text{Spezifität} = \frac{TN}{TN + FP} \quad (3.19)$$

$$\text{Positiver Vorhersagewert} = \frac{TP}{TP + FP} \quad (3.20)$$

$$\text{Negativer Vorhersagewert} = \frac{TN}{TN + FN} \quad (3.21)$$

$$\text{Matthew's Korrelationskoeffizient (MCC)} = \frac{Z}{N}, \quad (3.22)$$

wobei $Z = TP \times TN - FP \times FN$,

$$N = \sqrt{(TN + FN)(TP + FP)(TP + FN)(TN + FP)}.$$

$$\text{R-Wert} = \frac{(TP \times TN) - (FP \times FN)}{(TP + FN) \times (TN + FP)} \quad (3.23)$$

Die detaillierte Berechnung dieser Metriken ermöglicht es, das Gleichgewicht zwischen der Erkennung seismischer Ereignisse und der Vermeidung falscher Alarme zu bewerten, was für die praktische Anwendung von Erdbebenvorhersagemodellen von entscheidender Bedeutung ist.

3.5.2 Regressionsmetriken

Zur Bewertung der Vorhersagegenauigkeit konkreter Werte von Erdbeben, wie der exakten Bestimmung des Erdbebenzentrums, des Zeitpunkts des Auftretens und der Magnitude, kommen Regressionsmetriken zum Einsatz. Der Mean Absolute Error (MAE), der Mean Squared Error (MSE), das Bestimmtheitsmaß R^2 und der Nash-Sutcliffe-Effizienzkoeffizient (NSE) sind hierbei von zentraler Bedeutung. Diese Metriken messen die Abweichung zwischen den vorhergesagten und den tatsächlichen Werten und bieten somit eine quantitative Bewertung der Präzision des Modells in Bezug auf die Vorhersage spezifischer Eigenschaften eines Erdbebens [14]. Die Formeln für diese Metriken sind:

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (3.24)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3.25)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (3.26)$$

$$NSE = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (3.27)$$

Dabei repräsentiert Y_i die tatsächlichen Beobachtungen, \hat{Y}_i die Vorhersagen des Modells, \bar{Y} den Mittelwert der tatsächlichen Beobachtungen, und n die Anzahl der Beobachtungen.

AKTUELLER STAND DER FORSCHUNG ZUM EINSATZ VON MACHINE LEARNING IN DER ERDBEBENVORHERSAGE

4.1 METHODIK DER ANALYSE UND STRUKTURIERUNG DER STUDIEN

SUCHSTRATEGIE Um eine fundierte Auswahl an Forschungsarbeiten zur Erdbebenvorhersage mittels ML zu gewährleisten, wurde eine umfassende Literatursuche durchgeführt. Diese Suche erstreckte sich über anerkannte wissenschaftliche Datenbanken wie PubMed, IEEE Xplore, Scopus und Google Scholar und wurde bis zum 29. Februar 2024 ausgedehnt, um die Aktualität der Ergebnisse sicherzustellen.

Die Suche basierte auf einer Kombination aus Schlüsselbegriffen des ML ("Machine Learning", "Artificial Intelligence", "Deep Learning", "Neural Networks") und erdbebenspezifischen Termini ("Earthquake Prediction", "Seismic Forecasting", "Earthquake Early Warning Systems", "Seismology and AI"). Diese strategisch ausgewählten Suchbegriffe wurden für jede Datenbank individuell angepasst, um eine möglichst umfassende Abdeckung relevanter Publikationen zu erreichen.

EIN- UND AUSSCHLUSSKRITERIEN Die Studienauswahl folgte einem präzisen Satz von Kriterien, um die Eignung und Qualität der identifizierten Literatur zu bewerten.

Einschlusskriterien:

1. Studien, die ML-Methoden zur Erdbebenvorhersage anwenden.
2. Forschungsarbeiten, veröffentlicht in Englisch oder Deutsch.
3. Peer-Review Artikel und Konferenzbeiträge.
4. Veröffentlichungen bis zum 31. Februar 2024.

Ausschlusskriterien:

1. Arbeiten ohne Anwendung von ML in der Erdbebenvorhersage.
2. Übersichtsarbeiten oder Meta-Analysen.
3. Studien in anderen Sprachen als Englisch oder Deutsch.
4. Unveröffentlichte Dokumente oder Preprints.

STUDIENAUSWAHL UND ANALYSE Die Selektion und tiefgehende Analyse der Studien erfolgte nach dem Ausschluss von Duplikaten streng nach definierten Kriterien. Dieser Prozess, inklusive der Qualitätsbewertung, wurde eigenständig vorgenommen.

Die Qualität wurde nach einem Schema beurteilt, das auf wissenschaftliche Rigorosität, die Bedeutung der eingesetzten Methodik für die Erdbebenvorhersage, Klarheit in der Darstellung der Ergebnisse sowie auf Nachvollziehbarkeit und Reproduzierbarkeit der Forschung abzielte. Ein besonderes Augenmerk lag dabei auf der Validität der Daten, der Eignung der angewendeten ML-Modelle und der statistischen Bedeutsamkeit der Ergebnisse. Die Transparenz in Bezug auf die Beschreibung der Datensätze und die methodische Vorgehensweise wurde ebenfalls hoch bewertet. Eine Präferenz erhielten Studien, die durch eine umfassende Methodik, klare und verständliche Ergebnisse sowie durch eine sorgfältige Diskussion eventueller Einschränkungen überzeugten. Dieser individuell angelegte Bewertungsansatz führte zur Auswahl hochwertiger Studien, die wesentlich zum Verständnis und Fortschritt in der Erdbebenvorhersage mittels ML beitragen.

Die schlussendlich ausgewählten Studien unterzogen sich einer ausführlichen Untersuchung, in welcher wichtige Informationen bezüglich der verwendeten ML-Methodiken, der analysierten Datensätze, der Zielvariablen und der erzielten Ergebnisse extrahiert wurden. Dieser methodische Ansatz garantiert eine detaillierte und exakte Aufarbeitung des gegenwärtigen Forschungsstands.

STRUKTURIERUNG DER FORSCHUNGSANSÄTZE Um die methodischen Ansätze und deren Effizienz vollständig zu erfassen, wurden die Forschungsarbeiten in drei Kategorien unterteilt: Studien zu spezifischen Algorithmen, vergleichende Analysen von Algorithmen sowie Studien mit fortgeschrittenen Hybrid- und Ensemble-Methoden. Diese Kategorisierung erlaubt eine tiefgehende Betrachtung der Entwicklungen, aktuellen Herausforderungen und zukünftigen Perspektiven in diesem Forschungsfeld. Sie dient als Basis, um die methodische Diversität und die Wirksamkeit von ML-Techniken in der Seismologie zu bewerten.

Spezifische Algorithmen Diese Kategorie umfasst Studien, die sich auf die detaillierte Untersuchung einzelner maschineller Lernalgorithmen konzentrieren. Sie zielen darauf ab, die spezifischen Fähigkeiten und Grenzen dieser Modelle bei der Analyse und Vorhersage seismischer Aktivitäten zu evaluieren. Solche Untersuchungen sind essenziell, um die Anwendbarkeit einzelner Algorithmen auf konkrete Problemstellungen in der Erdbebenvorhersage zu bestimmen.

Vergleichende Studien Hier werden Arbeiten zusammengefasst, die verschiedene Algorithmen gegenüberstellen, um ihre relativen Vor- und Nachteile zu beleuchten. Diese vergleichenden Studien liefern wertvolle Erkenntnisse für die Auswahl des am besten geeigneten maschinellen Lernalgorithmus, basierend auf der Art der seismischen Daten und den spezifischen Vorhersagezielen.

Fortgeschrittene Hybrid- und Ensemble-Modelle In dieser Kategorie finden sich Forschungsarbeiten, die durch die Kombination mehrerer Algorithmen die Genauigkeit und Zuverlässigkeit von Erdbebenvorhersagen verbessern. Während Hybridmodelle unterschiedliche methodische Ansätze integrieren, um ein umfassenderes Bild seismischer Daten zu gewinnen, nutzen Ensemble-Methoden die Stärken mehrerer Modelle, um die Gesamtgenauigkeit der Vorhersagen zu erhöhen. Diese Ansätze zeigen das Bestreben, durch die Nutzung der Vielfalt an maschinellen Lernmethoden die Prognosequalität kontinuierlich zu optimieren.

Bei der Analyse wurden 9 Studien mit spezifischen Algorithmen, 8 vergleichende Studien und 14 Studien mit fortgeschrittenen Hybrid- und Ensemble-Modellen identifiziert.

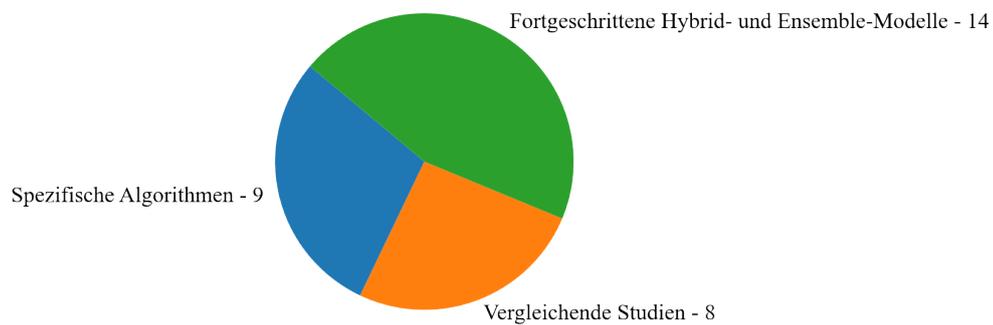


Abbildung 4.1: Verteilung der Forschungsansätze

4.2 SPEZIFISCHE ALGORITHMEN

ROUET-LEDUC ET AL., 2017 Rouet-Leduc et al. untersuchten 2017 die Anwendbarkeit eines RF-Modells zur Vorhersage der Zeit bis zum nächsten Scherfehler in einem kontrollierten Laborumfeld. Diese Laborexperimente simulieren die komplexen physikalischen Bedingungen, die zu seismischen Ereignissen führen, wobei ein besonderer Fokus auf der Rolle akustischer Emissionen als Indikatoren für bevorstehende Scherfehler gelegt wird. **Methodik:** Das RF-Modell nutzt kontinuierlich erfasste akustische Zeitreihendaten, die aus mikroseismischen Aktivitäten in Gesteinsproben unter Stressbedingungen stammen. Durch die Integration eines gewichteten Durchschnittsverfahrens über mehrere Entscheidungsbäume prognostiziert das Modell die verbleibende Zeit bis zum nächsten Scherbruch. Eine wichtige Komponente der Studie ist das rekursive Feature-Selection-Verfahren, das aus einer Vielzahl statistischer Maße (z.B. Mittelwert, Varianz, Kurtosis, Autokorrelation) die prädiktivsten Merkmale isoliert. **Dateninput:** Die Eingabedaten umfassen akustische Emissionsdaten, die während der Laborversuche gesammelt wurden, einschließlich der Zeitreihen von Schallwellen, die durch die mikroseismische Aktivität der Gesteinsproben unter Druck entstanden sind. Diese Daten spiegeln die physischen Prozesse wider, die einem Scherfehler vorausgehen, und bieten damit eine prädiktive Basis für das Modell. **Ergebnisse:** Das entwickelte RF-Modell zeichnete sich durch eine hohe Vorhersagegenauigkeit aus, was durch einen R^2 -Wert von 0,89 bestätigt wurde. Dies verdeutlicht die Effektivität des Modells bei der präzisen Prognose der Zeitspanne bis zum nächsten Scherfehler. **Schlussfolgerung:** Die Ergebnisse der Studie belegen die Machbarkeit der Erdbebenvorhersage durch die Analyse akustischer Emissionen mittels RF-Modellen. Die erfolgreiche Identifikation und Auswahl relevanter Merkmale ist dabei von zentraler Bedeutung für die Prognoseleistung des Modells. [51]

HAJIKHODAVERDIKHAN ET AL., 2018 Hajikhodaverdikhan et al. entwickelten 2018 ein fortschrittliches Modell, das SVR mit Partikelfilter-Techniken kombiniert, um Erdbebenmagnitude und -häufigkeit im folgenden Monat in Tabriz, Iran, vorherzusagen. Die Partikelfilter-Technik ermöglichen die dynamische Anpassung von Modellparametern durch die Simulation und Auswahl der wahrscheinlichsten Parameterwerte basierend auf ihren Vorhersageergebnissen. Diese innovative Methode zieht sowohl meteorologische als auch seismische Daten heran, um präzise Prognosen zu ermöglichen. **Methodik:** Die Studie verwendet meteorologische Daten und seismische Daten, um die mittlere Magnitude und Erdbebenhäufigkeit innerhalb eines Monats mit einem SVR-Modell vorherzusagen. Zur Optimierung der SVR-Parameter, einschließlich des Skalierungsfaktors des Radial Basis Function Kernels, wird das Verfahren der Partikelfilter eingesetzt. Diese Partikelfilter, eine spezielle Form von Sequential-Monte-Carlo-Methoden, passen die Parameter an, indem sie eine Reihe von Stichproben (Partikel) generieren und iterativ anhand der Wahrscheinlichkeitsdichtefunktion bewerten. Diese

Funktion bestimmt die Gewichtung jedes Partikels, wodurch die Parameter schrittweise verfeinert werden, um eine optimale Lösung zu finden. **Dateninput:** Die Analyse integriert meteorologische (z.B. Temperatur, Windgeschwindigkeit, Niederschlag) und seismische Daten als Eingabemerkmale für das SVR-Modell. **Ergebnisse:** Die methodische Erweiterung des SVR-Modells durch Partikelfilter erzielte eine bemerkenswerte Ergebnisse in der Vorhersage, mit einem R^2 -Wert von 0,96 für die mittlere Magnitude und Genauigkeit von 0,78 für die Erdbebenhäufigkeit. **Schlussfolgerung:** Durch die Kombination von SVR mit Partikelfilter-Techniken gelingt es, die Modellparameter präzise anzupassen, was zu einer signifikanten Steigerung der Vorhersagegenauigkeit führt. Diese Studie betont das Potenzial, meteorologische und seismische Daten effektiv für die Erdbebenvorhersage zu nutzen. [30]

SHODIQ ET AL., 2018 Im Jahr 2018 entwickelten Shodiq et al. ein ANN, das zum Ziel hatte, die Wahrscheinlichkeit von Nachbeben in Indonesien nach einem Hauptbeben innerhalb 5 Tage zu bestimmen. Für diese Analyse wurden historische seismische Daten aus mehr als einem Jahrhundert herangezogen. **Methodik:** Als Eingabeparameter für das ANN dienten sieben spezifische seismische Indikatoren. Vor der Modellierung führten die Forschenden eine automatische Clusterbildung durch, um eine optimale Datenorganisation zu gewährleisten, wobei eine hierarchische K-Means-Clustering-Methode zum Einsatz kam. Die Optimierung des ANN-Modells erfolgte experimentell durch Anpassung der Anzahl der versteckten Schichten und Neuronen. **Dateninput:** Das Modell verarbeitete Daten zu 82.850 Erdbeben, die zwischen 1910 und 2017 in Indonesien registriert wurden. Diese Daten beinhalteten Indikatoren wie die Wiederkehrperiode T , die mittlere Magnitude M_{mean} , den Energieunterschied $dE_{1,2}$, den Koeffizienten β , die viskose Dämpfung η , die erwartete Magnitude M_{expected} , die Reibung μ , fünf Inkremente des b -Werts, die maximale Magnitude und das dynamische Gutenberg-Richter Gesetz. **Ergebnisse:** Das ANN-Modell erzielte eine Vorhersagegenauigkeit von 56% bis 72% für Erdbeben mit einer Magnitude über 6, bei einer Präzision von von 86% bis 96% und einer Sensitivität von 0% bis 38%, was die Effektivität dieser Methode für die Vorhersage von Nachbeben demonstriert. **Schlussfolgerung:** Diese Untersuchung hebt die Bedeutung einer sorgfältigen Datenpräparation und der spezifischen Anpassung der Architektur des neuronalen Netzwerks hervor. Die Ergebnisse belegen zudem das Potenzial von ANNs in der Analyse seismischer Daten und der Prognose von Nachbebenereignissen. [60]

LIN ET AL., 2018 In ihrer Studie aus dem Jahr 2018 untersuchten Lin et al. die Anwendung eines ANNs mit Backpropagation zur Prognose von Erdbeben. Durch die Integration von realen Daten und Informationen über tektonische Verschiebungsraten strebte die Forschungsgruppe danach, die Tiefe als auch die Magnitude zukünftiger Erdbeben abzuschätzen. **Methodik:** Ein wesentlicher Schwerpunkt der Studie lag auf der Feinabstimmung der An-

zahl von Neuronen in den verborgenen Schichten, um die Genauigkeit der Vorhersagen zu optimieren. Die Leistung des Modells wurde anhand von Kriterien wie niedrigen Standardabweichungen, minimalen mittleren MSE und hohen Korrelationskoeffizienten für die Magnitude und Tiefe von zukünftigen Erdbeben mit einer Magnitude größer als 5 und einer Tiefe von 300 km. **Dateninput:** Die verwendeten Daten umfassten Zeitreihen und tektonische Verschiebungsraten aus dem Zeitraum von 2000 bis 2011. **Ergebnisse:** Optimale Modellkonfigurationen führten zu MSE-Werten zwischen 0,01 und 0,09, was die Wirksamkeit des vorgeschlagenen Ansatzes für die Vorhersage von Erdbeben unterstreicht. **Schlussfolgerung:** Diese Forschung betont die Wichtigkeit einer sorgfältigen Modellkonfiguration, insbesondere bezüglich der Anzahl der Neuronen in den verborgenen Schichten, für die Verbesserung der Vorhersagegenauigkeit bei der Nutzung neuronaler Netzwerke in der Seismologie. [40]

HUANG ET AL., 2018 Huang et al. erforschten 2018 den Einsatz eines CNN zur Vorhersage von Erbebenereignissen in Taiwan für die nächsten 30 Tage. Die Studie nutzt umfassende historische seismische Datensätze, um die Eignung von CNNs für die Erdbebenvorhersage zu bewerten. **Methodik:** Entwickelt wurde das Modell mit einem Fokus auf Erdbeben mit einer Magnitude von $M=6$ oder höher. Um Überanpassung entgegenzuwirken, integrierte das Modell zwei Dropout-Schichten und verwendete die ReLU-Aktivierungsfunktion. Die Modelloptimierung basierte auf einer Lernrate von 0.001 und einem Momentum von 0.9. **Dateninput:** Die Analyse umfasste einen umfassenden Datensatz historischer seismischer Ereignisse in Taiwan, einschließlich Zeitpunkt, geografische Koordinaten sowie Magnitude, und deckte den Zeitraum vom 1. Januar 1970 bis zum 25. Mai 2016 ab. **Ergebnisse:** Trotz des relativ niedrigen R-Wertes von 0,303 verdeutlicht das CNN-Modell sowohl die Herausforderungen bei der Vorhersage von Erdbebenmagnituden als auch das Potenzial von CNNs in diesem Forschungsbereich. **Schlussfolgerung:** Die Untersuchung illustriert, wie CNNs zur Analyse seismischer Daten genutzt werden können, und unterstreicht die Bedeutung der Anwendung von Techniken zur Vermeidung von Überanpassung. Obwohl der erzielte R-Wert die Schwierigkeiten bei der Magnitudenvorhersage aufzeigt, bietet der Ansatz einen innovativen Pfad für zukünftige Forschungen in der Erdbebenvorhersage. [32]

FLORIDO ET AL., 2018 Im Jahr 2018 führten Florido et al. eine innovative Studie durch, die sich der Entwicklung einer Methodik widmete, um seismische Vorläufermuster mittels DT zu identifizieren. Ziel war es, auf der Grundlage historischer Seismizitätsdaten aus der Iberischen Halbinsel, Chile und Japan Muster zu entdecken, die auf mittlere bis große Erdbeben hinweisen. **Methodik:** Die Forschenden setzten Clustering-Verfahren und den Aufbau von Vorläuferbäumen ein, um signifikante Muster zu erkennen. Diese Methodik zielte darauf ab, die Genauigkeit der Erdbebenvorhersagen zu erhöhen und gleichzeitig die Rate falsch positiver Vorhersagen zu mini-

mieren. **Dateninput:** Analysiert wurden historische Seismizitätsdaten, einschließlich einer Reihe von Parametern wie Wiederkehrperiode T , mittlere Magnitude M_{mean} , Energieunterschied $dE_{1,2}$, viskose Dämpfung η , Reibung μ , Koeffizient c , definierte Magnitude M_{def} , die a und b Werte, die maximale Magnitude, die Erdbebenwahrscheinlichkeit und fünf Inkremente des b -Werts. **Ergebnisse:** Die Identifizierung signifikanter seismischer Vorläufermuster ermöglichte die Vorhersage eines Erdbebens am folgenden Tag mit einer maximalen Genauigkeit von 93,59% für Erdbeben mit einer Magnitude größer als 4,5, bei einer Sensitivität von 78,57% und einer Spezifität von 95,07%. **Schlussfolgerung:** Diese Forschungsarbeit demonstriert das große Potenzial baumbasierter Algorithmen zur Verfeinerung der Erdbebenvorhersage durch detaillierte Analyse historischer Seismizitätsdaten. Der Ansatz von Florido et al. könnte einen bedeutenden Beitrag zur seismischen Risikobewertung leisten. [24]

WANG ET AL., 2020 Wang et al. untersuchten 2020 die Anwendung von LSTM Netzwerken zur Erforschung raum-zeitlicher Korrelationen zwischen Erdbebenereignissen in China. Basierend auf einem umfangreichen Erdbebenkatalog, der Magnitude 4,5 und höher von 1966 bis 2016 umfasst, zielte die Studie darauf ab, Muster zu identifizieren, die Erdbeben vorhersagen können. **Methoden:** Zur Darstellung seismischer Aktivitäten entwickelte das Team eine innovative zweidimensionale Matrix, die Erdbebenereignisse und deren Zeitstempel aus verschiedenen Regionen abbildet. Diese Matrix, ergänzt durch ein komplexes Netzwerkdesign mit 128 LSTM-Neuronen, einem mehrschichtigen Dichtennetzwerk, SoftMax-Aktivierungsfunktion und RMSprop-Optimierung, bildete den Kern der Datenanalyse. Eine spezielle Datenzerlegungsmethode gruppierte die Erdbebenereignisse nach geologischen Störungzonen, was die Analyse der raum-zeitlichen Muster verbesserte. **Dateninput:** Die Analyse nutzte seismische Daten, strukturiert in einer zweidimensionalen Matrix, die Magnitude und Zeitstempel von Erdbeben in China über fünf Jahrzehnte umfasst, aufgeteilt nach geologischen Störungzonen. **Ergebnisse:** Die Leistung des LSTM-Modells bei der Vorhersage von Erdbeben mit einer Magnitude über 4,5 zeigte bei eindimensionalen Dateninputs eine Genauigkeit von 63,50%, eine Sensitivität von 46,83% und eine Präzision von 79,6%. Eine deutliche Verbesserung wurde bei der Verwendung zweidimensionaler Dateninputs festgestellt, wo die Genauigkeit auf 87,59%, die Sensitivität auf 86,48% und die Präzision auf 87,87% anstieg. Diese Ergebnisse verdeutlichen die Fähigkeit des LSTM-Netzwerks, komplexe raum-zeitliche Muster in den seismischen Daten zu erkennen und daraus präzise Vorhersagen abzuleiten. **Schlussfolgerung:** Wang et al. betonen das Potenzial von LSTM-Netzwerken, die inhärente raum-zeitliche Struktur in seismischen Daten effektiv für präzise Erdbebenvorhersagen zu nutzen. Besonders hervorgehoben wird die Bedeutung einer methodischen Datenstrukturierung und -gruppierung, die essentiell für die Optimierung der Vorhersageleistung ist. [64]

SAAD ET AL., 2022 Saad et al. präsentierten 2022 einen RF-Algorithmus für die schnelle Lokalisierung von Erdbebenepizentren. Dieses Modell nutzt die differentialen Ankunftszeiten der P-Wellen an den ersten fünf seismologischen Stationen, um die Epizentralorte präzise zu schätzen und somit Frühwarnsysteme für Erdbeben bei der Entscheidungsfindung zu unterstützen. **Methodik:** Der RF-Algorithmus klassifiziert die differentiellen Ankunftszeiten der P-Wellen zusammen mit den geografischen Positionen der Stationen, wobei die Ankunftszeit an der ersten registrierenden Station als Referenzwert dient. **Dateninput:** Als Eingabe dienen die differentiellen Ankunftszeiten von P-Wellen sowie die Standorte der ersten fünf Stationen, die ein Erdbeben registriert haben. **Ergebnisse:** Der Algorithmus liefert Schätzungen für die Epizentralorte der Erdbeben. Die Evaluation des RF-Modells ergab einen mittleren absoluten Fehler (MAE) von 2,88 km, was die hohe Genauigkeit des Modells bei der Vorhersage der Epizentralorte unterstreicht. Zudem zeigt das Modell, dass selbst mit einer begrenzten Datenmenge (nur 10 % des gesamten Datensatzes) und weniger Stationen (drei) zufriedenstellende Ergebnisse ($MAE < 5$ km) erzielt werden können. **Schlussfolgerung:** Das RF-basierte Modell von Saad et al. unterstreicht die kritische Rolle der schnellen und präzisen Bestimmung von Epizentralorten für Erdbebenfrühwarnsysteme. Es belegt effektiv, dass der RF-Algorithmus eine vielversprechende Methode für die schnelle Lokalisierung von Erdbebenquellen bietet. [52]

WANG ET AL., 2023 Wang et al. entwickelten 2023 das Modell EEWNet, ein hochentwickeltes CCN, das gezielt darauf ausgerichtet ist, aus den initialen Sekunden der an einer einzelnen seismischen Station erfassten P-Wellen eine schnelle und genaue Magnitudenschätzung zu ermöglichen. **Methodik:** EEWNet implementiert tiefe Lernstrategien, um aus den anfänglichen P-Wellen-Signalen automatisch relevante Merkmale für die Magnitudenvorhersage zu extrahieren. Das Training, die Validierung und das Testen des Modells erfolgten anhand umfangreicher Datensätze von Bohrloch-Untergrundstarkbewegungen. **Dateninput:** Der Ansatz verwendet Parameter der ersten Sekunden der P-Wellen, aufgezeichnet von einer einzelnen seismischen Station, ohne die Notwendigkeit einer Vorverarbeitung. **Ergebnisse:** EEWNet konnte die Magnitude von Erdbeben im Bereich von 4,0 bis 5,9 bereits innerhalb der ersten 0,5 Sekunden nach der Ankunft der P-Welle mit einem mittleren MAE von 0,28 vorhersagen. **Schlussfolgerung:** Die Ergebnisse von Wang et al. betonen das enorme Potenzial tiefer Lernmethoden zur Steigerung der Präzision und Geschwindigkeit in Erdbebenfrühwarnsystemen. Mit EEWNet wird ein signifikanter Schritt nach vorne gemacht, indem schnelle und genaue Magnitudenvorhersagen ermöglicht werden, was essentiell für die Minimierung von Erdbebenschäden ist. [65]

Tabelle 4.1: Zusammenfassung der spezifischen Algorithmen

Algorithmus	Eingabe (Merkmale)	Ausgabe (Prognose)	Metrik	Ref.
RF	Akustische Emissionsdaten	Zeit bis zum Scherbruch	R^2 : 0,89	[51]
SVR	Temperatur, Windgeschwindigkeit, Niederschlag	Magnitude und Häufigkeit	R^2 : 0,96 (Magnitude), Genauigkeit: 0,78 (Häufigkeit)	[30]
ANN	$T, M_{\text{mean}}, dE_{1,2}, \beta, \eta, M_{\text{expected}}, \mu, 5 \times b$ -Wert, maximale Magnitude	Erdbebenereignis	Genauigkeit: 56%-72%, Präzision: 86%-96%, Sensitivität 0%-38%	[60]
ANN	Zeitreihendaten, tektonische Verschiebungsraten	Magnitude und Tiefe	MSE: 0,01-0,09	[40]
CNN	Zeitpunkt, geografische Koordinaten, Magnitude	Erdbebenereignis	R-Wert: 0,303	[32]
DT	$T, M_{\text{mean}}, dE_{1,2}, \eta, \mu, c, M_{\text{def}}, a$ und b Wert, maximale Magnitude, Wahrscheinlichkeit eines Erdbebenereignisses, $5 \times b$ -Wert	Erdbebenereignis	Genauigkeit: 93,59%, Sensitivität: 78,57%, Spezifität: 95,07%	[24]
LSTM	Magnitude, Zeitpunkt, Störungszone	Erdbebenereignis	Genauigkeit: 87,59%, Sensitivität: 86,48%, Präzision: 79,6%, 87,87%	[64]
RF	P-Wellen Ankunftszeiten, Stationenstandorte	Epizentralort	MAE: 2,88 km	[52]
CCN	Erste Sekunden der P-Wellen	Magnitude	MAE: 0,28	[65]

4.3 VERGLEICHENDE STUDIEN

BHANDARKAR ET AL., 2019 Bhandarkar et al. führten 2019 eine vergleichende Untersuchung der Leistungsfähigkeit eines FFNN gegenüber einem LSTM Netzwerk durch, um Erdbebetrends (Magnitude, Tiefe, den genauen Zeitpunkt des Eintretens sowie geografische Koordinaten) in verschiedenen geographischen Gebieten, darunter der indische Subkontinent und mehrere asiatische Länder, vorherzusagen. **Methodik:** Das FFNN-Modell umfasste zwei versteckte Schichten mit jeweils 20 und 60 Neuronen und setzte die Sigmoid-Funktion als Aktivierungsmechanismus ein. Demgegenüber verfügte das LSTM-Modell über zwei verborgene Schichten mit 40 Einheiten und verwendete den Adagrad-Algorithmus zur Minimierung des RMS-Fehlers, ergänzt durch eine Dropout-Schicht zur Vermeidung von Overfitting. **Dateninput:** Grundlage der Analyse bildeten detaillierte historische Aufzeichnungen zu Erdbebenereignissen in den untersuchten Regionen, darunter spezifische Daten wie Magnitude, geografische Lage und Zeitpunkt jedes Ereignisses. Diese Daten wurden gezielt herangezogen, um Muster und Trends in der Häufigkeit sowie der Intensität der Erdbeben über bestimmte Zeiträume hinweg zu identifizieren und vorherzusagen. **Ergebnisse:** Die Vorhersagen zielten auf die Ermittlung von Trends bei Erdbeben ab, insbesondere bezüglich allgemeiner Richtung sowie Muster in Frequenz und Stärke über Zeit. Hierbei erwies sich das LSTM-Modell als deutlich leistungsfähiger und erzielte einen um 59% höheren durchschnittlichen R^2 -Wert von -0.252 über alle Trendvariablen im Vergleich zum FFNN. **Schlussfolgerung:** Diese Studie unterstreicht die Überlegenheit von LSTM-Netzwerken gegenüber FFNNs bei der Vorhersage von Erdbebetrends. Die Fähigkeit von LSTM, langfristige zeitliche Abhängigkeiten in den Daten zu erfassen, macht es besonders geeignet für die Analyse seismischer Aktivitäten. [11]

KARIMZADEH ET AL., 2019 Karimzadeh et al. realisierten 2019 eine vergleichende Analyse der Algorithmen NB, KNN, SVM und RF durch, um Muster von Nachbeben des Kermanshah-Erdbebens in Iran von 2017 vorherzusagen, das schwerste Erdbeben des Landes in jenem Jahr. Die Daten hierzu stammen vom Iranischen Seismologischen Netzwerk, erfasst zwischen dem 12. November 2017 und dem Ende September 2018, mit dem spezifischen Ziel, die Position von Nachbeben mit einer Magnitude von über 2,5 vorherzusagen. **Methodik:** Im Zentrum der Analyse standen die Slip-Distribution (Bodenbewegungsanalyse), die Coulomb-Spannungsänderung (Änderungen der Erdspannung) und die Positionen benachbarter aktiver Verwerfungen. Zur Ermittlung der Slip-Distribution kamen Berechnungen zum Einsatz, die sowohl nichtlineare als auch lineare Umkehrmethoden umfassten. Mit diesen Informationen sowie den geografischen Koordinaten der Nachbeben erstellten die Forschenden eine binäre Rasterkarte, auf deren Basis die genannten Klassifikatoren die Wahrscheinlichkeit von Nachbeben für jedes Rasterfeld vorhersagten. **Dateninput:** Erfasst wurden Daten zu Nachbeben, einschließlich geografischer Koordinaten und Tiefe, die

Slip-Distribution, Coulomb-Spannungsänderungen und geografische Koordinaten. **Ergebnisse:** Als Output generierten die Algorithmen binäre Vorhersagen (0 oder 1) bezüglich der Wahrscheinlichkeit von Nachbeben in den Rasterfeldern. Der RF-Algorithmus zeichnete sich durch die höchste Vorhersagegenauigkeit von 75% aus und erwies sich damit als das effektivste Werkzeug zur Prognose von Nachbebenmustern. **Schlussfolgerung:** Diese Studie verdeutlicht den Mehrwert eines vergleichenden Ansatzes verschiedener Algorithmen für die Erdbebenvorhersage und hebt besonders die Eignung des RF-Algorithmus hervor. Die sorgfältige Integration geologischer und seismologischer Datenquellen trägt entscheidend zur Steigerung der Vorhersagequalität bei. [36]

KHAWAJA M. ASIM ET AL., 2020 Khawaja M. Asim et al. evaluierten 2018 die Fähigkeiten kurzfristiger Vorhersagen seismischer Aktivitäten niedriger Magnitude in der Zypern-Region unter Verwendung von Algorithmen wie ANN, SVM und RF. **Methodik:** Die Forschenden wendeten diese drei Algorithmen auf einen sorgfältig bereinigten Erdbebenkatalog an, aus dem Störungen entfernt und sechzig seismische Merkmale extrahiert wurden. Diese dienen dazu, die seismischen Zustände der Region zu modellieren und Vorhersagen über verschiedene Momenten-Magnitudenschwellen und Zeiträume zu treffen. **Dateninput:** Der bereinigte Erdbebenkatalog enthielt sechzig seismische Merkmale, darunter geografische Koordinaten T , mittlere Magnitude M_{mean} , Tiefenfehler $dE_{1,2}$, Bodenverschiebung β , Ankunftszeitdifferenz η , Streuung der Bruchstärke σ_b , definierte Magnitude M_{def} , Wiederkehrperiode T_r , Ausbreitungszone z , Messfehler δm , die a und b Werte der Gutenberg-Richter-Beziehung sowie die Wahrscheinlichkeit eines Erdbebenereignisses. **Ergebnisse:** Für die Vorhersagen über verschiedene Momenten-Magnitudenschwellen und Zeiträume zeigte der RF-Algorithmus die höchste Genauigkeit bei Schwellenwerten von 3,0 und 3,5, während die SVM bei Schwellenwerten von 4,0 und 4,5 mit einem MCC von 0.861 die besten Ergebnisse lieferte. **Schlussfolgerung:** Diese Studie beleuchtet die Effektivität spezifischer Algorithmen bei der Prognose seismischer Aktivitäten verschiedener Magnituden und Zeiträume. Besonders hervorgehoben werden SVM und RF für ihre herausragende Leistung, was ihre Eignung für die Anwendung in der seismologischen Vorhersage unterstreicht. [8]

MURWANTARA ET AL., 2020 Im Jahr 2020 untersuchten Murwantara et al. die Leistungsfähigkeit der Algorithmen LogR, SVM und NB, die für die kontinuierliche Wertevorhersage angepasst wurden, für mittel- bis langfristige Erdbebenvorhersagen. Sie konzentrierten sich dabei auf die Analyse umfangreicher seismischer Datensätze, die Erdbeben über zwei unterschiedliche Zeiträume (10 und 30 Jahre) klassifizierten. **Methodik:** Die Studie bediente sich eines zweistufigen Vorhersageansatzes, bei dem zunächst die Erdbebenenergie aus geografischen Koordinaten extrahiert wurde. Daraufhin wurde die Magnitude der Erdbeben basierend auf den geografischen Koordinaten und der Tiefe der seismischen Ereignisse bestimmt. **Dateninput:**

Es wurden umfangreiche Datensätze herangezogen, die Erdbeben von 1900 bis 2018 dokumentieren, einschließlich Uhrzeit, Datum, geografische Koordinaten, Magnitude und Tiefe. **Ergebnisse:** Die Untersuchung erbrachte Vorhersagen hinsichtlich des Orts, der Tiefe und der Magnitude von Erdbeben. Insbesondere bei der Vorhersage der Magnitude erwies sich SVM mit dem niedrigsten MAE von 0,598 als am präzisesten im Vergleich zu den anderen getesteten Algorithmen. **Schlussfolgerung:** Die Ergebnisse von Murwantara et al. betonen die überlegene Genauigkeit von SVM bei der Vorhersage spezifischer Erdbebencharakteristika für mittel- bis langfristige Zeiträume. [45]

ESSAM ET AL., 2021 Essam et al. erforschten 2021 die Anwendung von ANN und RF für die Vorhersage von Erdbebenparametern, wie Beschleunigung, Tiefe und Bewegung der Erdoberfläche (Bodengeschwindigkeit), in Terengganu, Malaysia. Diese Modelle wurden mit Daten von sechs seismologischen Stationen trainiert. **Methodik:** Die Forscher stellten fest, dass die Tiefe von Erdbeben effizient mit einem univariaten Modell prognostiziert werden kann, das nur die Tiefe als Eingabeparameter verwendet. Um jedoch eine genaue Vorhersage der Beschleunigung und Geschwindigkeit zu erreichen, war es notwendig, die Tiefe als zusätzlichen Eingabeparameter in die Modelle zu integrieren. **Dateninput:** Die Analyse basierte auf Daten von seismologischen Stationen, die Informationen über die Tiefe, Beschleunigung und Geschwindigkeit von Erdbeben enthielten. **Ergebnisse:** Die ANN- und RF-Modelle erbrachten Vorhersagen für mehrere Erdbebenparameter. Das ANN-Modell zeigte dabei eine besonders hohe Leistungsfähigkeit, mit einem NSE von 0,8815 für die Beschleunigung, 0,9985 für die Tiefe und 0,9933 für die Geschwindigkeit. **Schlussfolgerung:** Die Studie betont die signifikante Bedeutung der Tiefe bei der Vorhersage von Erdbebenparametern und bestätigt das Potenzial von ANN und RF, komplexe seismologische Daten effektiv zu analysieren und präzise Vorhersagen zu ermöglichen. [21]

KHOSRAVIKIA & CLAYTON, 2021 Im Jahr 2021 evaluierten Khosravikia und Clayton die Effizienz verschiedener Algorithmen – speziell ANNs, RF und SVM – bei der Prognose der Bodenbewegungsintensität, ausgedrückt durch Spitzenbodenbeschleunigung (PGA) und spektrale Beschleunigung (PSA). **Methodik:** Die Untersuchung bezog sich auf eine detaillierte Datenbank mit Informationen zu 374 Erdbeben, die seit 2005 in Texas, Oklahoma und Kansas verzeichnet wurden. Als Merkmale für die Modelle dienten die Momentmagnitude, die hypozentrale Distanz und die durchschnittliche Scherwellengeschwindigkeit. **Dateninput:** Grundlage für die Vorhersagen waren Aufzeichnungen von Bodenbewegungen dieser Erdbeben, einschließlich der Momenten-Magnituden-Größe, der Entfernung zum Epizentrum und der Geschwindigkeit der S-Welle. **Ergebnisse:** Die Modelle prognostizierten die Intensität der Bodenbewegung. Unter den getesteten Algorithmen erzielte SVM mit einem R-Wert von 0,739 die genauesten Ergebnisse, was es sowohl ANN als auch RF überlegen machte. **Schlussfolgerung:** Die-

se Studie hebt die Überlegenheit von SVM bei der Vorhersage von Bodenbewegungsintensitäten hervor und verweist auf die kritische Bedeutung der gezielten Auswahl und Anwendung maschineller Lernverfahren für seismische Analysen. [37]

PAPIYA DEBNATH ET AL., 2021 Debnath et al. untersuchten 2021 die Wirksamkeit verschiedener Algorithmen - DT, NB, LogR und RF - für die Prognose von klassierten Erdbebenmagnituden in sechs indischen Staaten mit hohem Erdbebenrisiko von 1900 bis 2000. **Methodik:** Die Studie zielte darauf ab, Erdbeben in drei Magnitudenkategorien einzuteilen: leicht, moderat und schwer. Dabei wurden historische Erdbebendaten mit 20 Attributen, einschließlich Zeit, geografische Koordinaten und Tiefe, als Grundlage verwendet. **Dateninput:** Es wurden umfangreiche historische Erdbebendatenätze für Indien mit Details wie Zeitpunkt, geografischen Koordinaten und Tiefe analysiert. **Ergebnisse:** Als Ergebnis generierten die Modelle Vorhersagen über die Zugehörigkeit von Erdbeben zu den definierten Magnitudenkategorien (Milde Erdbeben: (weniger als 4,5), Moderate Erdbeben: (4,5 bis 5,5), Schwere (Fatal) Erdbeben: (mehr als 5,5)). Unter den getesteten Algorithmen zeichnete sich die logistische Regression mit einer herausragenden Genauigkeit von 99,94% aus, was ihre Eignung für die Erdbebenvorhersage unterstreicht. **Schlussfolgerung:** Debnath et al. demonstrieren mit ihrer Forschung, dass die sorgfältige Auswahl und Anwendung maschineller Lernverfahren entscheidend für die Genauigkeit von Erdbebenvorhersagen ist. Insbesondere wird die Fähigkeit einfacher Modelle wie der logistischen Regression hervorgehoben, bei korrekter Implementierung außergewöhnliche Ergebnisse zu liefern. [18]

YOUSEFZADEH ET AL., 2021 Im Jahr 2021 evaluierten Yousefzadeh et al. die Anwendung verschiedener Algorithmen, einschließlich SVM, DT, ein einfaches ANN und ein tiefes ANN, um klassierten Erdbebenmagnituden in Iran für die nachfolgende Woche vorherzusagen. **Methodik:** Die Forschungsarbeit stützt sich auf Daten zwischen Januar 1973 und Juli 2019, die ausschließlich Erdbeben mit einer Magnitude über 3 betrachten. Ein neuartiger Parameter zur Erfassung der Fehlerdichte wurde in die Analyse integriert. Zur Minimierung der Überanpassung setzten die Autoren eine Reihe spezifischer Feinabstimmungstechniken ein, um die Modelle zu optimieren. **Dateninput:** Die Analyse basierte auf seismischen Daten, die 16 Parameter umfassten, darunter die Wiederkehrperiode T , die mittlere Magnitude M_{mean} , den Energieunterschied $dE_{1,2}$, die viskose Dämpfung η , die Reibung μ , die Kohäsion c , die Magnitudendifferenz ΔM , die maximale Magnitude, die Wahrscheinlichkeit eines Erdbebenereignisses, fünf Inkremente des b -Werts, die a und b Werte, geografische Koordinaten, Tiefe und Störungsdichte. **Ergebnisse:** Als Ergebnis wurden Magnitudenvorhersagen für die kommende Woche in vier Klassen (Klasse 1: (3 bis 3,7), Klasse 2: (3,7 bis 4,5), Klasse 3: (4,5 bis 5,1), Klasse 4: (5,1 bis 6,4)) generiert. Dabei erzielten die DT eine Vorhersagegenauigkeit von 100%, während SVM in Bezug auf die Sensitivi-

tätsbewertung mit 97,7% herausragte. **Schlussfolgerung:** Die Untersuchung unterstreicht das Potenzial der Kombination aus etablierten maschinellen Lernverfahren und der Einführung neuer Datenparameter, um die Vorhersagegenauigkeit von Erdbebenmagnituden signifikant zu verbessern. [68]

Tabelle 4.2: Zusammenfassung der vergleichenden Studien

Algorithmen	Eingabe (Merkmale)	Ausgabe (Prognose)	Metrik	Ref.
FFNN, LSTM	Magnitude, geografische Koordinaten, Zeitpunkt	Trend	R^2 : -0.252 (LSTM)	[11]
NB, KNN, SVM, RF	Daten der Nachbeben, Slip-Distribution, Coulomb-Spannungsänderung	Erdbebenereignis	Genauigkeit: 75% (RF)	[36]
ANN, SVM, RF	$T, M_{\text{mean}}, dE_{1,2}, \beta, \eta, \sigma_b, M_{\text{def}}, T_r, z, \delta m, a, b$	Magnitude (klassiert)	MCC: 0,861 (SVM)	[8]
LogR, SVM, NB	Zeitpunkt, geografische Koordinaten, Magnitude, Tiefe	Ort, Tiefe, Magnitude	MAE: 0,598 (Magnitude mit SVM)	[45]
ANN, RF	Beschleunigung, Tiefe, Geschwindigkeit	Tiefe, Beschleunigung, Geschwindigkeit	NSE (ANN): 0,8815 (Beschleunigung), 0,9985 (Tiefe), 0,9933 (Geschwindigkeit)	[21]
ANN, RF, SVM	Magnitude, Entfernung zum Epizentrum, Geschwindigkeit der S-Welle	Bodenbewegungsintensität	R-Wert: 0,739 (SVM)	[37]
DT, NB, LogR, RF	20 Attribute einschließlich geografische Koordinaten, Zeitpunkt, Tiefe	Magnitude (Klassiert)	Genauigkeit: 99,94 %	[18]
SVM, DT, ANN	$T, M_{\text{mean}}, dE_{1,2}, \eta, \mu, c, \Delta M$, maximale Magnitude, Wahrscheinlichkeit eines Erdbebenereignisses, $5 \times b$ -Wert, a, b , geografische Koordinaten, Tiefe, Störungsdichte	Magnitude (Klassiert)	Genauigkeit: 100 % (DT), Sensitivität: 97,7 (SVM)	[68]

4.4 FORTGESCHRITTENE HYBRID- UND ENSEMBLE-MODELLE

ASIM ET AL., 2016 Asim et al. untersuchten 2016 die Vorhersage von Erdbeben in der Hindukush-Region mit Magnituden $\geq 5,5$ unter Verwendung von vier maschinellen Lernverfahren und acht mathematisch berechneten seismischen Indikatoren, basierend auf historischen Erdbebendaten von Januar 1976 bis Dezember 2013. **Methodik:** Es wurden vier Modelle eingesetzt: PRNN, RNN, RF und LPBoost. Jedes Modell wurde separat angewendet, um die Beziehungen zwischen den seismischen Parametern und zukünftigen Erdbebenereignissen zu modellieren. Die Vorhersage wurde als binäre Klassifikationsaufgabe formuliert, wobei Erdbeben mit Magnituden $\geq 5,5$ über einen Zeitraum von einem Monat vorhergesagt wurden. **Dateninput:** Die Analyse umfasste 11.137 Erdbebenereignisse und acht seismische Indikatoren, die auf der Gutenberg-Richter-Inversionsregel, der Verteilung charakteristischer Erdbebenmagnituden und seismischer Ruhephasen basieren. Zu den Indikatoren gehörten M_{mean} , T , dE , η , σ_b und ΔM . **Ergebnisse:** Die Modelle prognostizierten das Auftreten von Erdbeben mit Magnituden $\geq 5,5$ für den nächsten Monat. Dabei zeigte LPBoost mit einer Genauigkeit von 65% und einer Sensitivität von 91% die beste Leistung unter den getesteten Methoden. Die Studie bestätigte das Potenzial von Ensemble-Methoden zur Verbesserung der Genauigkeit in der Erdbebenvorhersage. **Schlussfolgerung:** Die Ergebnisse unterstreichen den Nutzen von Ensemble-Methoden für präzisere Erdbebenvorhersagen in der Seismologie. Die Integration verschiedener Ansätze kann zur Zuverlässigkeit der Vorhersagen beitragen. [7]

RAFEI & ADELI, 2017 Rafei und Adeli entwickelten 2017 ein fortschrittliches Erdbeben-Frühwarnsystem (NEEWS), das auf der Kombination aus ML und mathematischer Optimierung basiert. Ihr Ansatz nutzt vier Klassifizierungsalgorithmen - die drei speziellen neuronalen Netzwerkmodelle NDC, PNN, EPNN und SVM - zusammen mit einem Optimierungsalgorithmus, dem Neural Dynamic Optimization Algorithm (NDAP), zur Vorhersage von Erdbeben in Südkalifornien. **Methodik:** Das System nutzt acht seismische Indikatoren als Eingabedaten, um die Magnitude und den Ort zukünftiger Erdbeben zu prognostizieren. Die Klassifizierungsalgorithmen beurteilen die Wahrscheinlichkeit des Auftretens eines Erdbebens über einem festgelegten Schwellenwert innerhalb eines bestimmten Zeitraums. Der Optimierungsalgorithmus NDAP spielt eine Schlüsselrolle bei der Präzisierung der Vorhersage, indem er den Ort des Erdbebens mit der höchsten Wahrscheinlichkeit identifiziert. NDAP verwendet eine fortgeschrittene Form des maschinellen Lernens, um komplexe, nichtlineare Optimierungsprobleme zu lösen, die bei der Vorhersage seismischer Ereignisse auftreten. Durch das Lernen aus historischen Daten identifiziert NDAP Muster in den seismischen Aktivitäten und optimiert die Berechnung des wahrscheinlichsten Erdbebenzentrums, indem es die geografischen Koordinaten und die erwartete Magnitude in die Prognose einbezieht. **Dateninput:** Verwendet wurden Daten aus Erdbebenkatalogen für Südkalifornien von 1932 bis 2016, einschließlich $dE^{\frac{1}{2}}$, η , σ_b und

ΔM. Ergebnisse: Das NDC-Modell erzielte eine herausragende Genauigkeit von 99,4% bei der Vorhersage der vier Magnitudenklasse (Schwellwerte: 4,5; 5,0; 5,5; 6,0 und 6,5) des nächsten Erdbebens innerhalb eines halben Monats, was die Effektivität des kombinierten Ansatzes von Klassifizierung und Optimierung unterstreicht. **Schlussfolgerung:** Die Studie zeigt das Potenzial der Integration aus maschinellem Lernen und Optimierungsalgorithmen für die Entwicklung von Erdbeben-Frühwarnsystemen. Die Kombination verschiedener Algorithmen ermöglicht eine effektive Vorhersage der Erdbebenmagnitude und -ortung und stellt einen bedeutenden Fortschritt in der seismologischen Forschung dar. [47]

ASENCIO-CORTÉS ET AL., 2018 Asencio-Cortés et al. stellten 2018 eine fortschrittliche Ensemble-basierte Regressionsmethode vor, die auf die Vorhersage der Erdbebenmagnitude innerhalb eines Zeitraums von sieben Tagen abzielt. Grundlage ihrer Methode war die Nutzung eines umfassenden Erdbebendatenkatalogs aus Kalifornien verarbeitet wurde. **Methodik:** Die Autoren nutzten eine fortschrittliche Ensemble-Lernmethode zur Magnitudenprognose, indem sie vier Algorithmen – GLM, GBM, DL und RF – kombinierten. Diese individuellen Modelle wurden zunächst unabhängig auf den Datensatz angewandt, um separate Vorhersagen zu generieren. Die resultierenden Vorhersagen flossen anschließend als Eingaben in ein Stacking-Ensemble, das ebenfalls auf dem RF-Algorithmus basiert. Dieses Meta-Modell wurde trainiert, um die Vorhersagen der einzelnen Algorithmen effektiv zu integrieren und so die Gesamtgenauigkeit der Magnitudenprognose zu verbessern. Diese Kombination wurde auf 16 seismischen Merkmalen und 27 Regressionsdatensätzen angewendet. **Dateninput:** Die verwendeten Erdbebendaten umfassten Breitengrad, Längengrad und Magnitude von Beben in Kalifornien, aufgeteilt in eine Gittermatrix mit Zellengröße 0,5x0,5 Grad. **Ergebnisse:** Der Fokus lag auf der Vorhersage der Magnitude innerhalb der nächsten sieben Tage. Das RF-basierte Stacking-Ensemble erzielte durchschnittlich die besten Ergebnisse, mit einem mittleren MAE von 0,74. **Schlussfolgerung:** Die Studie demonstriert das Potenzial von Ensemble-Methoden, insbesondere von Stacking-Ensembles, zur Verbesserung der Präzision in der Erdbebenvorhersage. Die Anwendung dieser Methoden auf Cloud-Plattformen wie AWS bietet zudem neue Möglichkeiten für die effiziente Verarbeitung und Analyse seismischer Daten. [4]

ASIM ET AL., 2018A Khawaja M. Asim et al. entwickelten 2018 ein fortschrittliches Klassifikationssystem, das auf einer Kombination von SVR und HNN basiert, um Erdbeben der Magnitude 5.0 und darüber bis zu 15 Tage im Voraus vorherzusagen. Der Fokus der Studie lag auf den Regionen Hindukusch, Südkalifornien und Chile im Zeitraum von Januar 1980 bis Dezember 2016. **Methodik:** Durch ein zweistufiges Verfahren der Merkmalsauswahl, das auf der Methode der maximalen Relevanz und minimalen Redundanz (MRMR) basiert, wurden sechzig aussagekräftige seismische Merkmale als Prädiktoren für seismische Ereignisse identifiziert. Diese ausge-

wählten Merkmale wurden zunächst der SVR zugeführt, deren Prognosen anschließend als Eingabedaten für das HNN dienen. Diese Prozesse wurden durch eine optimierte Partikelschwarmoptimierung (EPSO) unterstützt, um die Gewichtungen der Modelle zu optimieren. **Dateninput:** Die Analyse berücksichtigte seismische Daten, darunter die Zeit zwischen Ereignissen (T), durchschnittliche Magnitude (M_{mean}), seismische Energieabgabe (dE), das Verhältnis von kleinen zu großen Beben (β), die Abweichung vom Gutenberg-Richter-Gesetz (η), die Standardabweichung des b -Wertes (σ_b), das Magnitudedefizit (ΔM), die Wiederkehrperiode (T_r), die maximale Magnitude (z), die Wahrscheinlichkeit eines Erdbebenereignisses (P) sowie die a - und b -Werte nach der Gutenberg-Richter-Beziehung. **Ergebnisse:** Das umgesetzte Vorhersagemodell erreichte für die Region Südkalifornien die größte Genauigkeit mit 90,6%, bei einer Sensitivität von 63,5% und einer Präzision von 93,8%. **Schlussfolgerung:** Das vorgestellte Klassifikationssystem, welches SVR und HNN miteinander verbindet, demonstriert das Potenzial dieser Methodik für die Vorhersage von Erdbeben. Besonders hervorgehoben wird die Wichtigkeit einer sorgfältigen Auswahl und Optimierung der Merkmale.[5]

ASIM ET AL., 2018B Khawaja M. Asim et al. entwickelten analog zu ihrer Studie [5] 2018 ein weiteres ML-Modell, welches auch darauf abzielt, Erdbeben der Magnitude 5.0 und darüber in den Regionen Hindukusch, Chile und Südkalifornien bis zu 15 Tage im Voraus von Januar 1980 bis Dezember 2016 zu prognostizieren. Dabei entwickelten sie EPGPBoost, ein zukunftsweisendes Erdbebenvorhersagesystem, das GP mit AdaBoost vereint. Dieser Ansatz **Methodik:** EPGPBoost nutzt die Kombination aus den explorativen Kapazitäten der GP und der Verfeinerung durch AdaBoost, um die Vorhersagepräzision zu erhöhen und die Anzahl der Falschalarme zu reduzieren. **Dateninput:** Verwendet werden seismische Daten einschließlich der Zeit zwischen Ereignissen (T), durchschnittlicher Magnitude (M_{mean}), seismischer Energieabgabe (dE), dem Verhältnis von kleinen zu großen Beben (β), Abweichungen vom Gutenberg-Richter-Gesetz (η), der Standardabweichung des b -Werts (σ_b), Magnitudedefizit (ΔM), Wiederkehrperiode (T_r), maximaler Magnitude (z), der Wahrscheinlichkeit eines Erdbebenereignisses (P) und den a - und b -Werten der Gutenberg-Richter-Beziehung. **Ergebnisse:** In der Region Südkalifornien wurde eine Genauigkeit von 86,6%, eine Sensitivität von 68,7% und Präzision von 84,2% erreicht, was die Effektivität von EPGPBoost bei der Erdbebenvorhersage belegt. **Schlussfolgerung:** Der innovative Einsatz von GP und AdaBoost in EPGPBoost illustriert das Potenzial von Ensemble-Methoden zur Steigerung der Vorhersagegenauigkeit und bietet einen vielversprechenden Weg zur Minimierung von Falschalarmen in der Erdbebenvorhersage.[6]

CORBI ET AL., 2019 In ihrer Studie aus dem Jahr 2019 untersuchten Corbi et al. die Anwendbarkeit von GBRT zur Vorhersage des Zeitpunkts und der Magnitude von in einer Labor-Subduktionszone simulierten Erdbeben. Die-

se Simulationen deckten ein Magnitudenspektrum von 6,2 bis 8,3 ab und imitierten damit realistische seismische Ereignisse. **Methodik:** Als physisches Modell diente eine Gelatinemasse, die auf einer flachen, rigiden Platte platziert wurde, um seismische Zyklen zu emulieren. GBRT kam zum Einsatz, um die Zeit bis zum nächsten Bruch zu prognostizieren, basierend auf 94 Variablen, die die Oberflächenverformungen erfassen. **Dateninput:** Grundlage der Untersuchung waren die aus den Laborversuchen gewonnenen Daten, die simulierte seismische Zyklen innerhalb einer künstlich erstellten Subduktionszone abbildeten. **Ergebnisse:** Die Vorhersagemethode erreichte einen Korrelationskoeffizienten R^2 zwischen 0,7 und 0,8, was eine hohe Zuverlässigkeit in der Vorhersagekraft der Methode signalisiert. **Schlussfolgerung:** Corbi et al. demonstrieren das Potential von GBRT zur Erdbebenvorhersage in kontrollierten Laborumgebungen und legen nahe, dass solche Machine Learning-Methoden wesentlich zur Verbesserung der Genauigkeit von Echtzeit-Erdbebenvorhersagemodellen beitragen könnten. [17]

MAJHI ET AL., 2019 Majhi et al. präsentierten 2019 ein innovatives Vorhersagemodell, das auf einem FLANN basiert und durch einen Moth Flame Optimization (MFO) verbessert wird. Ziel war es, die Magnitude von Erdbeben mit einer Magnitude von 5,5 oder höher vorherzusagen. **Methodik:** Das vorgestellte FLANN-Modell wurde mit dem MFO-Algorithmus kombiniert, um die optimalen Gewichte des Modells zu bestimmen. Dieser Ansatz zielt darauf ab, die Genauigkeit der Erdbebenvorhersage durch Optimierung der Modellparameter zu verbessern. **Dateninput:** Als Eingabe dienten seismische Indikatoren wie Datum, Zeit, geografische Koordinaten und Tiefe der Erdbeben. **Ergebnisse:** Das Modell erzielte einen Root MSE von 0,0565, was seine herausragende Vorhersageleistung unterstreicht. **Schlussfolgerung:** Diese Studie demonstriert das Potenzial hybrider Vorhersagemodelle, insbesondere die Kombination aus FLANN und MFO, zur Steigerung der Vorhersagegenauigkeit in der Seismologie. Die Arbeit betont die Wichtigkeit der Erweiterung von Eingangsmustern und der präzisen Optimierung der Modellgewichte, um komplexe Nichtlinearitäten in Erdbeben-daten zu erfassen und zuverlässige Vorhersagen zu ermöglichen. [41]

KUBO ET AL., 2020 Kubo et al. präsentierten 2020 eine fortschrittliche hybride Methode mit GMPes, um die Intensität von durch Erdbeben ausgelösten Bodenbewegungen in Japan zu schätzen. Diese Methode nutzt Daten von 1997 bis 2017 und kombiniert Beobachtungsdaten zu Bodenbewegungen mit geographischen und seismischen Quellinformationen. **Methodik:** Die Hybridmethode zielt darauf ab, durch die Integration von GMPes mit ERT die präzisen Vorhersagen von GMPes und die flexible Datenanpassung durch ERT zu vereinen. **Dateninput:** Verwendet wurden seismische Daten aus Japan, darunter die epizentrale Distanz, Momentmagnitude, Tiefe, obere Tiefe zur Schicht mit einer S-Wellengeschwindigkeit von 1.400 m/s am Standort und die durchschnittliche S-Wellengeschwindigkeit bis zu einer Tiefe von 30 m am Standort. **Ergebnisse:** Die innovative Hybridmethode

erzielte einen R-Wert von 0.619 und zeigte damit eine höhere Vorhersagepräzision im Vergleich zu den Einzelmethoden. **Schlussfolgerung:** Diese Untersuchung unterstreicht das Verbesserungspotenzial der Vorhersagegenauigkeit durch den Einsatz hybrider Modelle, die verschiedene methodologische Ansätze kombinieren, besonders in Bereichen mit komplexen geologischen Bedingungen und limitierten Daten. [39]

CHANDA ET AL., 2021 Chanda et al. führten 2021 eine umfassende Untersuchung der Vorhersagefähigkeit von sechs Algorithmen - KNN, SVM, RF, ANN, DT und AdaBoost - für die Prognose der Bodenbewegung in unterschiedlichen tektonischen Umgebungen Chiles durch. Die Studie analysierte Daten von 10.713 Erdbeben, unterteilt in die Hauptkategorien inslab, interface und crustale Beben. **Methodik:** Unter Einsatz einer zehnfachen Kreuzvalidierung und einer gewichteten Anzahl von 30 Nachbarn für KNN evaluierten die Forscher die Effektivität jedes Algorithmus in Bezug auf die Vorhersagegenauigkeit von Bodenbewegungen. **Dateninput:** Die Analyse bezog sich auf Erdbebendaten aus Chile, die nach ihrer tektonischen Beschaffenheit klassifiziert waren. **Ergebnisse:** AdaBoost zeigte sich als überlegenster Algorithmus mit einem perfekten R-Wert von 1.00, was seine exzellente Fähigkeit zur Vorhersage der Bodenbewegung unterstreicht. **Schlussfolgerung:** Diese Studie verdeutlicht die Überlegenheit von AdaBoost im Vergleich zu anderen untersuchten Algorithmen bei der Prognose von Bodenbewegungen, abhängig von tektonischen Bedingungen. Sie betont zudem die Wichtigkeit der sorgfältigen Auswahl spezifischer Algorithmen basierend auf den Charakteristiken der Daten und den speziellen Anforderungen des Untersuchungsgegenstandes. [15]

SALAM ET AL., 2021 Salam et al. präsentierten 2021 zwei innovative hybride Modelle, um die Magnitude von Erdbeben in Südkalifornien für die nächsten 15 Tage zu prognostizieren. **Methodik:** Die Modelle FPA-ELM und FPA-LS-SVM zielen darauf ab, die Vorhersagegenauigkeit zu verbessern, indem sie die Modellparameter mithilfe des Fower Pollination Algorithm (FPA) optimieren. Dieser Algorithmus wird sowohl mit einem ELM als auch mit einer Least Square-SVM (LS-SVM) verwendet, um die internen Gewichtungen und Bias-Werte der ELM anzupassen und die optimalen Parameter für die LS-SVM zu finden. Die LS-SVM ist eine Variante der SVM, die auf der Minimierung von Quadratsummen basiert. **Dateninput:** Die Analyse verwendete Erdbebendaten aus Südkalifornien, die verschiedene Merkmale umfassten: Zeitintervalle zwischen den Ereignissen (T), durchschnittliche Magnitude (M_{mean}), Energieabgabe (dE), das Verhältnis von Erdbebenhäufigkeit zu ihrer Magnitude (β), Vorhersageabweichung (η), erwartete Magnitude ($M_{expected}$) und der Reibungskoeffizient oder die Scherrate (μ). **Ergebnisse:** Das FPA-LS-SVM Modell erzielte einen RMSE-Wert von 0.565476, was die Wirksamkeit dieses hybriden Ansatzes bei der Vorhersage von Erdbebenmagnituden belegt. **Schlussfolgerung:** Die Integration von Optimierungsalgorithmen wie dem FPA in maschinelle Lernverfahren verbessert signifikant

die Vorhersagepräzision, indem sie die Identifikation optimaler Parameterwerte ermöglicht. [54]

XIONG ET AL., 2021 Xiong et al. präsentierten 2021 das fortschrittliche "Inverse Boosting Pruning Trees" (IBPT)-Modell, das speziell für die kurzfristige Erdbebenvorhersage (bis 24 Tage) entwickelt wurde. Das IBPT-Modell integriert eine innovative Kombination aus adaptiven Boosting-Verfahren und dem Beschneiden von Entscheidungsbäumen und nutzt dabei Daten von 1371 Erdbeben mit einer Magnitude von 6 oder höher. **Methodik:** Das IBPT-Modell verwendet eine Ensemble-Technik, um die Vorhersagegenauigkeit durch das Beschneiden von Entscheidungsbäumen und das Anpassen der Gewichte mit einer neuartigen inversen Boosting-Technik zu verbessern. **Dateninput:** Die Modellierung berücksichtigt Erdbebendaten sowie Oberflächentemperatur- und atmosphärische Messungen. **Ergebnisse:** Das IBPT-Modell erreicht in der kurzfristigen Erdbebenvorhersage durch einen R-Wert von 0.6429 und einen MCC von 0.6581, was seine präzise Prognosefähigkeit für Bebenereignisse innerhalb eines kurzfristigen Zeitrahmens unterstreicht. **Schlussfolgerung:** Die Studie betont die Vorteile von Ensemble-Methoden, insbesondere die Anwendung des inversen Boostings auf beschnittene Entscheidungsbäume, als einen fortschrittlichen Ansatz in der Erdbebenvorhersage, der sowohl Genauigkeit als auch Robustheit der Vorhersage verbessert. [66]

CHITTORA ET AL., 2022 Chittora et al. führten 2022 eine umfassende Analyse von sechs unterschiedlichen Algorithmen durch, mit dem Ziel, die Magnituden zukünftiger Erdbeben in Indien und den angrenzenden Gebieten vorherzusagen. Die Studie nutzte sechs Datensätze, die aus verschiedenen Regionen Indiens stammen, um die Magnitudenklasse zukünftiger Erdbeben in drei Kategorien einzuteilen: tödliche Erdbeben (Magnituden über 5,5), moderate Erdbeben (Magnituden zwischen 4,5 und 5,5) und leichte Erdbeben (Magnituden unter 4,5). **Methodik:** Die angewandten Methoden beinhalten die Lernalgorithmen wie das ANN, RF, Diskriminanzanalyse, XGBoost Tree (Verbindung zwischen XGBoost und DT) und die DT-Algorithmen CHAID und Tree-AS. **Dateninput:** Die analysierten Datensätze enthielten detaillierte seismische Informationen über Zeit, geografische Koordinaten, Tiefe, Magnitude, Anzahl der Messstationen, seismische Lückendaten, epizentrale Distanz, Zeitmesspräzision, spezifische Ortsangaben und Fehler in horizontaler Position. **Ergebnisse:** Besonders hervorzuheben ist die herausragende Leistung des XGBoost Tree-Algorithmus, der für den Datensatz der Andamanen und Nikobaren eine Präzision von 99,27%, eine Sensitivität von 99,31% und eine Genauigkeit von 99,10 % erreichte. **Schlussfolgerung:** Die Ergebnisse der Studie unterstreichen die Überlegenheit des XGBoost Tree-Klassifikators im Vergleich zu anderen Algorithmen bei der Erdbebenvorhersage und betonen die Wichtigkeit, den am besten geeigneten Algorithmus auf Basis der spezifischen Anforderungen und Charakteristiken der Daten auszuwählen. [16]

ABDALZAHER ET AL., 2023 Abdalzaher et al. präsentierten 2023 ein innovatives Modell, das 2S-ML-EIOS, zur Schätzung der seismischen Intensität, speziell entwickelt für Erdbebenvorwarnsysteme. Dieser Ansatz vereint lineare und nicht-lineare Modelle zur Verbesserung der Vorwarnkapazitäten. **Methodik:** Das 2S-ML-EIOS-Modell nutzt die ersten zwei Sekunden nach der Ankunft der P-Welle, um die seismische Intensität zu schätzen. Das Ziel ist die Minimierung der Implementierungskomplexität und die Steigerung der Effektivität von Frühwarnsystemen. Verschiedene maschinelle Lernansätze, darunter sowohl lineare als auch nicht-lineare Modelle wie XGBoost und ANNs, werden zur Intensitätsschätzung eingesetzt, basierend auf den Daten der ersten zwei Sekunden nach dem P-Wellenbeginn. **Dateninput:** Die Datengrundlage besteht aus seismischen Wellenformen, die von 386 Stationen des italienischen nationalen seismischen Netzwerks aufgezeichnet wurden, basierend auf 50.000 seismischen Ereignissen. **Ergebnisse:** Das Modell erreichte eine Genauigkeit von 98,59%, mit XGBoost als einem der effektivsten verwendeten Modelle bei sechs Intensitätsklassen. **Schlussfolgerung:** Diese Untersuchung hebt das Potenzial des XGBoost-Modells, für die schnelle und präzise Schätzung der seismischen Intensität in Erdbebenvorwarnsystemen hervor. Dies ist von großer Bedeutung für die Risikominderung und das Katastrophenmanagement. [1]

BISWAS ET AL., 2023 Biswas et al. adressierten 2023 die Herausforderung der Erdbebenmagnitudenvorhersage und der seismischen Verwundbarkeitskartierung in der Türkei durch den Einsatz einer Vielfalt an Regressionsalgorithmen und ANNs. **Methodik:** In der Studie kamen verschiedene Methoden zum Einsatz, darunter DT, ET, RF, Ridge Regression und fortschrittliche Boosting-Algorithmen wie XGBoost, LightGBM und CatBoost. Zur Untersuchung der Effizienz bei der Erdbebenvorhersage und Risikokartierung wurden zudem drei unterschiedlich konfigurierte ANNs betrachtet. Das erste ANN (ANN-1) konzentrierte sich auf grundlegende Datenstrukturen, das zweite (ANN-2) auf die Optimierung der Erkennung zeitlicher Muster, und das dritte (ANN-3) nutzte eine tiefere und komplexere Netzwerkstruktur, um die Vorhersagefähigkeit zu verbessern. Diese Ansätze zielten darauf ab, ein breites Spektrum an Mustern und Beziehungen in den Daten zu erfassen und die Genauigkeit der Modellvorhersagen zu steigern. **Dateninput:** Die Analyse nutzte umfangreiche Erdbebendaten aus der Türkei vom Januar 2015 bis Februar 2023, einschließlich geografischer Koordinaten, Magnitude, Tiefe und spezifischer Messwerte wie horizontaler und Tiefenfehler. Die Kartierung der seismischen Verwundbarkeit erfolgte unter Verwendung geografischer Informationssysteme (GIS), wobei vier Risikozonen basierend auf der vorhergesagten Erdbebenmagnitude und historischen Daten abgegrenzt wurden: Rote Zone (höchstes Risiko), Orange Zone, Gelbe Zone und Grüne Zone (sicherste Region). **Ergebnisse:** Die Analyse ergab signifikante Ergebnisse für die Vorhersage der Erdbebenmagnitude und die Kartierung seismischer Risikobereiche. Die ANN-Modelle zeigten mit R^2 -Werten von 0,98012 für ANN-1, 0,98710 für ANN-2 und 0,97847 für ANN-3 herausra-

gende Vorhersageleistungen, was ihre Fähigkeit unterstreicht, die Variabilität der Erdbebenmagnitude präzise zu erfassen. Die ML-Modelle, insbesondere LightGBM mit einem MSE von 0,0607 und CatBoost mit einem MAE von 0,1984, lieferten ebenfalls starke Ergebnisse, allerdings mit einer geringeren Vorhersagegenauigkeit im Vergleich zu den ANN-Modellen. **Schlussfolgerung:** Die höheren R^2 -Werte der ANNs zeigen, wie gut sie komplexe Beziehungen in Erdbebendaten erfassen, während ihre höheren MSE- und MAE-Werte eine gesteigerte Sensibilität für spezielle Muster und Ausreißer andeuten. Ihre ausgezeichnete Generalisierungsfähigkeit und Genauigkeit machen ANNs zu effektiven Werkzeugen für Erdbebenvorhersagen und Risikokartierungen. Die Ensemble-Boosting-Modelle XGBoost, LightGBM und CatBoost zeichnen sich durch ihre Ausgewogenheit in Präzision und Widerstandsfähigkeit gegen Überanpassung aus. [12].

Tabelle 4.3: Zusammenfassung der fortgeschrittenen Hybrid- und Ensemble-Modelle

Algorithmen	Eingabe (Merkmale)	Ausgabe (Prognose)	Metrik	Ref.
PRNN, RF, RNN, LPBoost	$M_{\text{mean}}, T, dE, \eta, \sigma_b, \Delta M$	Erdbebenereignis	Genauigkeit: 65%, Sensitivität: 91% (LPBoost)	[7]
NDC, PNN, EPNN, SVM mit NDAP	$dE^{\frac{1}{2}}, \eta, \sigma_b, \Delta M$	Magnitude (Klassiert)	Genauigkeit: 99,4% (NDC mit NDAP)	[47]
GLM, GBM, DL, RF, Stacking	Magnitude, geografische Koordinaten	Magnitude	MAE: 0,74 (RF)	[4]
SVR, HNN	$T, M_{\text{mean}}, dE, \beta, \eta, \sigma_b, \Delta M, T_r, z, P, a, b$	Erdbebenereignis	Genauigkeit: 90,6%. Sensitivität: 63,5%, Präzision: 93,8%	[5]
GP, AdaBoost	$T, M_{\text{mean}}, dE, \beta, \eta, \sigma_b, \Delta M, T_r, z, P, a, b$	Erdbebenereignis	Genauigkeit: 86,6%, Sensitivität: 68,7%, Präzision: 84,2%	[6]
GBRT	94 Merkmale der Verformung	Zeit bis zum nächsten Labor-Erdbebens	R^2 : 0,7-0,8	[17]
FLANN, MFO	Datum, Zeitpunkt, geografische Koordinaten, Tiefe	Magnitude	RMSE: 0,0565	[41]
GMPE, ERT	Epizentrale Distanz, Momentmagnitude, Tiefe, S-Wellengeschwindigkeit	Intensität der Bewegungen	R-Wert: 0,619	[39]
KNN, SVM, RF, ANN, DT, AdaBoost	Historische Daten zur Dauer von Bodenbewegungen	Dauer von Bodenbewegungen	R-Wert: 1,00 (AdaBoost)	[15]
FPA-ELM, FPA-LS-SVM	$T, M_{\text{mean}}, dE, \beta, \eta, M_{\text{expected}}, \mu$	Magnitude	RMSE: 0,565476 (FPA-LS-SVM)	[54]
IBPT (DT, AdaBoost)	Temperatur, atmosphärische Daten	Erdbebenereignis	R-Wert: 0,6429, MCC: 0,6581	[66]
ANN, RF, XGBoost Tree, CHAID, Tree-AS	Zeitpunkt, geografische Koordinaten, Tiefe, Magnitude, Stationenanzahl, epizentrale Distanz, Lückendaten	Magnitude (Klassiert)	Genauigkeit: 99,10% (XGBoost Tree)	[16]
2S-ML-EIOS (XGBoost)	Seismischen Wellenformen der P-Wellen	Erdbebenintensität (Klassiert)	Genauigkeit: 98,59%	[1]
ANN, XGBoost, LightGBM, CatBoost	Magnitude, Tiefe, geografische Koordinaten, Stationenanzahl, epizentrale Distanz, Horizont- und Tiefenfehler	Magnitude (Risikokartierung)	R^2 : 0,987 (ANN), MSE: 0,061 (LightGBM), MAE: 0,198 (CatBoost)	[12]

4.5 ERKENNTNISSE

4.5.1 Effizienzvergleich der verschiedenen Modelle

Die untersuchten ML-Modelle beeindrucken durch ihre hohe Genauigkeit und geringen Fehlerquoten, wodurch sie sich klar von den herkömmlichen seismologischen Methoden absetzen. Besonders Ensemble-Methoden wie RF und AdaBoost sowie DL-Modelle wie CNN und LSTM zeigen eine bemerkenswerte Effektivität in der Erkennung und Interpretation seismologischer Datenmuster.

LSTM- UND CNN-BASIERTE ANSÄTZE LSTM-Modelle haben sich bei der Modellierung zeitabhängiger Dynamiken als effektiv erwiesen. Ein Beispiel hierfür ist die Arbeit von Wang et al. (2020), die eine Vorhersagegenauigkeit von 87,59% erzielte [64]. CNNs, bekannt für ihre Fähigkeit, räumliche Muster effizient zu erkennen, erreichten in der Studie von Huang (2018) zur Vorhersage seismischer Aktivitäten in Taiwan eine Genauigkeit von 78% [32]. EEWNet, ein speziell entwickeltes CNN-Modell, erbrachte präzise Schätzungen der Erdbebenstärke mit einem MAE von 0,28 [65].

ENSEMBLE- UND HYBRIDE MODELLE Ensemble- und Hybridmodelle heben die Erdbebenvorhersage auf ein neues Niveau, indem sie die Stärken verschiedener Methoden vereinen. Ein einfacheres Beispiel ist das RF-Modell von Saad et al. (2022), welches einen MAE von 2,88 km erreichte [52], was seine Effizienz in der Erdbebenvorhersage demonstriert. Die Kombination von GP und AdaBoost, implementiert von Asim et al. (2018b), erreichte sogar eine Vorhersagegenauigkeit von 86,6% [6]. Darüber hinaus illustrieren Biswas et al. (2023) mit LightGBM und CatBoost die Vorteile spezialisierter Algorithmen für die Magnitudenschätzung, mit einem MSE von 0,0607 und einem MAE von 0,1984 [12]. Diese Ansätze verdeutlichen, dass durch die Kombination und Integration unterschiedlicher Lernverfahren die Prognosegenauigkeit durch Reduzierung der Modellkomplexität und Minimierung der Gefahr von Überanpassung signifikant verbessert wird, was insbesondere bei der Analyse komplexer Erdbeben Daten entscheidend ist.

Die Präzision der Modelle beruht dabei auf einer sorgfältigen Selektion von Eingabedaten, Zielvariablen und der Anpassung spezifischer Algorithmen. Die Verwendung gleicher Algorithmen kann, abhängig von Datensatz und Zielsetzung, zu unterschiedlichen Ergebnissen führen, was die Suche nach einem universell überlegenen Algorithmus hinfällig macht. Der Schlüssel zur Optimierung der Prognoseleistung liegt in der Auswahl, Aufbereitung der Daten und der Feinjustierung der Algorithmen, um sie an die spezifischen Eigenschaften und Anforderungen der seismologischen Daten anzupassen .

4.5.2 Verbesserung der Datenaufbereitung und -nutzung

Die Effektivität von ML-Modellen in der Erdbebenvorhersage hängt wesentlich von der Qualität und Vielfalt der Daten sowie deren sorgfältiger Verarbeitung ab. Schlüsselaspekte der Datennutzung werden im Folgenden dargestellt:

DIVERSITÄT DER DATENQUELLEN: Eine breite Palette von Datenquellen verbessert signifikant die Modellleistung. Hajikhodaverdikhan et al. (2018) zeigen, dass die Integration meteorologischer Daten die Vorhersagegenauigkeit substantiell erhöht, mit einem R^2 -Wert von 0,96 [30].

GEZIELTE FEATURE-AUSWAHL: Die Auswahl relevanter Features ist entscheidend für die Modellpräzision. Rouet-Leduc et al. (2017) erreichten durch ein rekursives Feature-Selection-Verfahren ein R^2 von 0,89 [51].

VORVERARBEITUNG UND DATENQUALITÄT: Eine gründliche Vorverarbeitung ist für die Genauigkeit der Modelle essentiell. Shodiq et al. (2018) verbesserten mittels hierarchischem K-Means-Clustering die Vorhersagegenauigkeit deutlich [60].

EFFEKTIVE DATENAUFBEREITUNG UND FEATURE ENGINEERING: Die akribische Aufbereitung der Daten sowie das gezielte Engineering von Merkmalen stellen fundamentale Pfeiler in der Architektur präziser Vorhersagemodelle dar. Insbesondere illustriert das LSTM-Modell von Wang et al. (2020) auf exemplarische Weise, wie durch die sorgfältige Analyse und Nutzung räumlich-zeitlicher Strukturen innerhalb seismischer Daten ein signifikanter Anstieg in der Vorhersagegenauigkeit erzielt werden konnte [64]. Darüber hinaus hebt die Studie von Yousefzadeh et al. (2021) das Verbesserungspotenzial der Vorhersagegenauigkeit durch die Integration neuer Datenparameter mittels fortschrittlicher seismologischer Berechnungsmethoden, wie beispielsweise die Parameter a und b des Gutenberg-Richter-Gesetzes, hervor [68]. Diese Erkenntnisse unterstreichen die essenzielle Rolle einer methodischen Datenaufbereitung und des Feature Engineerings bei der Entwicklung fortschrittlicher maschineller Lernmodelle für die Erdbebenvorhersage.

4.5.3 Herausforderungen bei der Implementierung

ML-Modelle in der Erdbebenvorhersage stehen vor spezifischen Herausforderungen, die ihre Effizienz und Genauigkeit beeinflussen:

VERFÜGBARKEIT UND QUALITÄT DER TRAININGSDATEN: Die Leistung von ML-Modellen hängt stark von der Vollständigkeit und Genauigkeit der Trainingsdaten ab. Hajikhodaverdikhan et al. (2018) und Rouet-Leduc et al. (2017) betonen die Bedeutung umfassender Datensätze für eine effektive Modellbildung. Die Herausforderung besteht darin, umfassende Datensätze zu

sammeln, die eine breite Palette seismologischer Phänomene abdecken, um die Modelle effektiv zu trainieren und Über- bzw. Unteranpassung zu vermeiden [30, 51].

GENERALISIERBARKEIT: Ein weiteres kritisches Problem ist die Generalisierbarkeit der ML-Modelle über unterschiedliche geografische Gebiete hinweg. Wie die Untersuchungen von Chittora et al. (2022), Shodiq et al. (2018) und Wang et al. (2020) verdeutlichen, können Modelle, die in einer spezifischen Region erfolgreich sind, nicht notwendigerweise auf andere Regionen mit abweichenden seismischen Eigenschaften übertragen werden [16, 60, 64]. Insbesondere das ML-Modell von Chittora et al. (2022) zeigte eine deutliche Abweichung in der Leistung des XGBoost Tree-Algorithmus: Während er in einer Region eine herausragende Genauigkeit von 99,10% erzielte, fiel diese in einer anderen Region mit denselben Spezifikationen auf 72,96% ab [16]. Dies macht eine gezielte Anpassung und Validierung der Modelle für jede einzelne Anwendungsregion unerlässlich.

INTERPRETIERBARKEIT: Die Interpretierbarkeit komplexer ML-Modelle, insbesondere bei tiefen Lernverfahren, ist eine weitere signifikante Herausforderung. Modelle wie das EEWNet von Wang et al. (2023), die bemerkenswerte Leistungen in der Vorhersage zeigen, leiden oft unter einer begrenzten Nachvollziehbarkeit der Entscheidungsfindungsprozesse, was ihre praktische Anwendbarkeit einschränken kann [65].

VERGLEICHBARKEIT: Die Vielfalt der verwendeten Methoden und die individuelle Anpassung dieser, wie zum Beispiel unterschiedliche Klassifizierungen von Erdbeben nach ihrer Magnitude, erschweren einen direkten Vergleich der Leistungsfähigkeit verschiedener Modelle beträchtlich [18, 68]. Unterschiede in den Datensätzen, Bewertungskriterien und spezifischen Anpassungen der Algorithmen behindern eine einheitliche und objektive Beurteilung ihrer Wirksamkeit.

4.5.4 Zukunftsstrategien für Machine Learning in der Erdbebenvorhersage

Optimierungen im Bereich maschinellen Lernens sind für eine fortschrittliche Erdbebenvorhersage und effiziente Frühwarnsysteme essentiell. Zukünftige Forschung muss sich auf Schlüsselbereiche konzentrieren, um die Performance und Verlässlichkeit von ML-Modellen zu steigern:

INNOVATIVE ALGORITHMEN: Die Weiterentwicklung fortgeschrittener ML-Algorithmen, einschließlich CNNs und LSTMs, sowie der Einsatz hybriden Ansätzen, soll deren Flexibilität und Wirksamkeit über verschiedene seismische Kontexte hinweg erhöhen [32, 64].

DATENQUALITÄT UND -DIVERSITÄT: Präzise Vorhersagen setzen den Zugriff auf umfassende und vielfältige Daten voraus. Die Einbeziehung von

Satellitenbildern und geophysikalischen Messdaten, unterstützt durch eine sorgfältige Datenkuratierung, ist unabdingbar [6, 51].

STANDARDISIERTE EVALUATIONSMETRIKEN: Die Entwicklung einheitlicher Bewertungsstandards und Benchmark-Datensätze ist notwendig, um ML-Modelle objektiv zu beurteilen und die effektivsten Ansätze zu identifizieren.

Durch diese strategischen Schwerpunkte können signifikante Verbesserungen in Genauigkeit und Zuverlässigkeit der ML-basierten Seismologie erreicht werden. Kontinuierliche Innovationen, eine breite Datenbasis und standardisierte Evaluationsprozesse legen das Fundament für zukünftige Entwicklungen in diesem Forschungsfeld und in Frühwarnsystemen.

MODELLENTWICKLUNG

5.1 METHODISCHE GRUNDLAGEN UND ZIELSETZUNG

In diesem Abschnitt wird ein spezifisches ML-Modell konzipiert, das auf realen seismischen Daten basiert. Ziel ist es, die Vorhersagefähigkeit und praktische Anwendbarkeit von ML-Algorithmen in der Erdbebenvorhersage gründlich zu evaluieren. Dieser Ansatz ergänzt vorhandene Forschungen durch die Demonstration der Implementierung und Anwendung von ML-Techniken zur Minimierung seismischer Risiken und zur Förderung der öffentlichen Sicherheit. Die Implementierung wurde in einem Jupyter Notebook durchgeführt (siehe Anhang A).

Um das Problem der Generalisierbarkeit von ML-Modellen über verschiedene geografische Gebiete hinweg, wie in Abschnitt 4.5.3 beschrieben, anzugehen, wird die seismisch aktive Region der Türkei und Syrien (zwischen 36° und 42° nördlicher Breite und 26° und 45° östlicher Länge) untersucht. Dabei werden zwei spezifische Vorhersageziele fokussiert:

1. Die Ermittlung der Magnitude des stärksten Erdbebens innerhalb der nächsten fünf Tage.
2. Die Bestimmung der Wahrscheinlichkeit eines signifikanten Erdbebenereignisses mit einer Magnitude von mindestens 5.5 im gleichen fünf-tägigen Zeitraum.

Diese direkte Fokussierung auf die Modellierung der Magnitude unterstreicht das Bestreben, ein tiefgreifendes Verständnis der seismischen Dynamik zu entwickeln und die Präzision in der Risikoermittlung zu erhöhen. Der gewählte Fünf-Tage-Zeitraum reflektiert die hohe seismische Aktivität in der Region und die praktische Relevanz für präventive Sicherheitsmaßnahmen.

5.2 BESCHREIBUNG, PREPROCESSING UND ANALYSE DES DATENSATZES

5.2.1 *Beschreibung des Datensatzes*

Der verwendete Datensatz umfasst detaillierte Aufzeichnungen von rund 3,5 Millionen Erdbebenereignissen weltweit, erfasst im Zeitraum von 1990 bis 2023 und zur Verfügung gestellt durch die USGS. Er beinhaltet Informationen wie den Dauer des Ereignisses (in Millisekunden gemessen), geographische Lokalisationen, den Überprüfungsstatus der Daten (manuell überprüft oder automatisch generiert), das Vorliegen von Tsunami-Auslösungen, die Bedeutung des Ereignisses (auf einer Skala von 1 bis 3000, basierend auf einer Kombination aus Magnitude und Anzahl der Rückmeldungen), den Ereignistyp (überwiegend Erdbeben), die Magnitude, die spezifische Region sowie die Koordinaten (Längen- und Breitengrad) des Ereignisses [62].

Die umfassende Datenmenge und deren validierte Qualität machen diesen Datensatz besonders wertvoll für die Untersuchung und Bestätigung der Präzision sowie der praktischen Einsatzmöglichkeiten von ML in der Erdbebenvorhersage, wie unter 4.5.3 dargestellt.

5.2.2 *Preprocessing*

Eine vorläufige Analyse ergab keine fehlenden Werte, sodass nur minimales Preprocessing erforderlich war. Dazu gehörte das Entfernen doppelter Zeilen, das Bereinigen und Standardisieren der Spalte 'state', das Konvertieren der 'date'-Spalte in das datetime-Format, das Filtern der Daten auf Ereignisse des Typs 'Erdbeben' und das Entfernen von Einträgen mit einer Magnitude von 0. Zusätzlich wurde die 'time'-Spalte von Unix-Zeitstempeln in lesbare Datums-/Zeitangaben konvertiert und mit der 'date'-Spalte abgeglichen. Für eine fokussierte Analyse wurden die Daten anschließend gefiltert, um nur Erdbebenereignisse innerhalb der geografischen Grenzen der Türkei (zwischen 36° und 42° nördlicher Breite und 26° und 45° östlicher Länge) zu berücksichtigen. Diese Eingrenzung resultierte in einem spezifischen Datensatz von 21,760 Einträgen, ideal für die tiefgehende Untersuchung der Erdbebenvorhersagefähigkeit von ML-Modellen in dieser seismisch aktiven Region.

5.2.3 *Explorative Datenanalyse*

MAGNITUDEENTWICKLUNG Um einen Überblick über die Entwicklung der Magnituden in der Region von 1990 bis 2023 zu erhalten, wurden die Magnituden der Erdbeben in einem Scatter-Plot dargestellt. Dabei ist zu erkennen, dass die durchschnittliche Magnitude in der betrachteten Region im Zeitraum von 1990 bis 2023 eine signifikante Zunahme erfahren hat, beginnend bei knapp über 3 und ansteigend bis zu einem Wert von 4. Auffällig ist dabei, dass nur wenige Beben signifikant von der durchschnittlichen Magnitude abweichen.

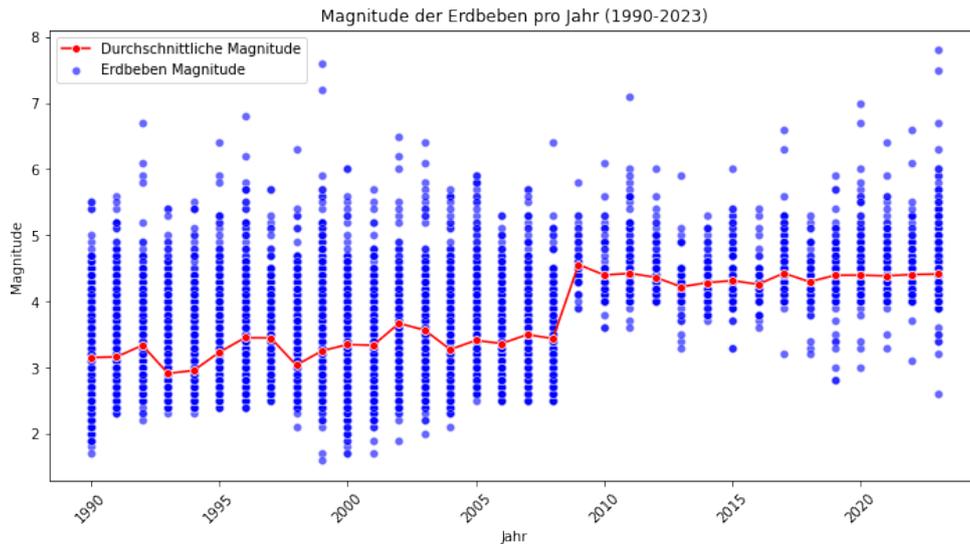


Abbildung 5.1: Magnitudenentwicklung in der Türkei-Syrien Region

Insgesamt machen starke Erdbeben nur einen geringen Anteil von 0.574% aller Ereignisse in der Region aus. Dennoch ist ein zunehmender Trend starker Beben im Laufe der Jahre zu beobachten. Dabei hat der Anteil starker Erdbeben kontinuierlich von 1990 bis 2023 von 0.260 % auf 2.773% zugenommen. Diese Daten unterstreichen eine besorgniserregende Tendenz hin zu einer größeren Häufigkeit und Intensität seismischer Aktivitäten in der Region, was die Notwendigkeit einer fortlaufenden Überwachung und angepasster Präventionsmaßnahmen unterstreicht.

LOKALISIERUNG DER ERDBEBEN Um die Verteilung von starken und schwachen Erdbeben zu visualisieren, wurde eine Kartierung der Erdbeben in der Region durchgeführt.

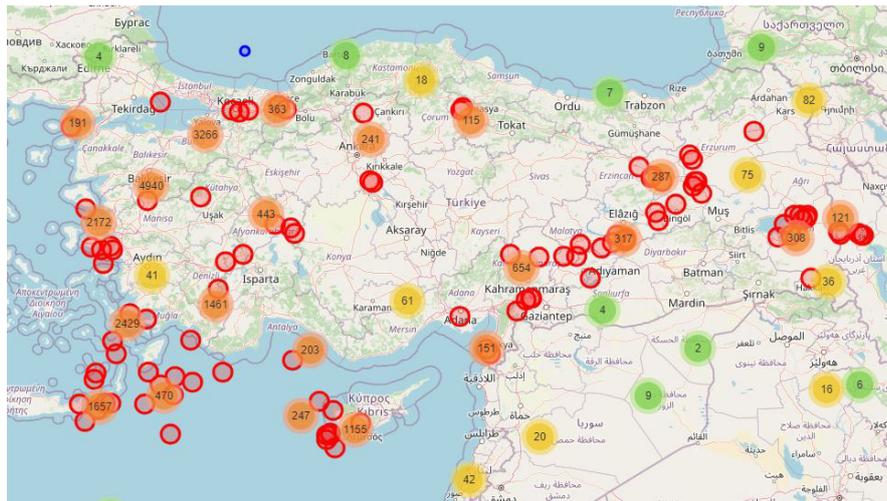


Abbildung 5.2: Lokalisierung der Erbeben in der Türkei-Syrien Region

Auf dieser Karte ist eine markante Beobachtung, dass starke Erdbeben, hervorgehoben durch rote Markierungen, tendenziell in der Nähe von Bereichen mit zahlreichen schwächeren Beben auftreten. Die Tatsache, dass starke Erdbeben weniger häufig auftreten als ihre schwächeren Gegenstücke, und ihre räumliche Nähe zu Zonen hoher seismischer Aktivität niedrigerer Intensität, bestätigt die Gültigkeit des Gutenberg-Richter-Gesetzes für die dargestellte Region und dessen Bedeutung in der Erdbebenvorhersage.

5.3 MODELLAUSWAHL UND FESTLEGUNG VON BEWERTUNGSKRITERIEN

5.3.0.1 *Modellauswahl*

Für die Modellierung wurden folgende Algorithmen ausgewählt:

- **Regression:** Einsatz der linearen Regression zur Magnitudenschätzung und logistische Regression für die Wahrscheinlichkeitsprognose signifikanter Erdbeben als Basisalgorithmen.
- **RF:** Anwendung für beide Zielsetzungen durch Aggregation von Entscheidungsbäumen zur Genauigkeitssteigerung.
- **ANN:** Verwendung für die Modellierung komplexer Muster in den Zielvorhersagen durch mehrschichtige Datenverarbeitung.
- **LSTM:** Spezialisierung auf die Analyse zeitlicher Sequenzen zur Erfassung langfristiger Muster in den Erdbebendaten.
- **Ensemble-Methoden:** Kombination verschiedener Modelle zu einem Meta-Modell für optimierte Vorhersagen beider Forschungsziele.

Die lineare und logistische Regression werden als Basisalgorithmen aufgrund ihrer Effektivität in der Darstellung von Trends und Wahrscheinlichkeiten ausgewählt. Sie bilden das Fundament für die Entwicklung komplexerer Modelle, indem sie essenzielle Fähigkeiten zur Mustererkennung und Vorhersage bieten, auf denen weiterführende Analysen aufbauen. Die spezifische Auswahl weiterer Modelle wie RF, ANN, LSTM und Ensemble-Methoden basiert auf ihrer nachgewiesenen Effizienz bei der Handhabung komplexer Datenstrukturen und -muster, wie in Abschnitt 4.5.1 detailliert beschrieben. Diese Modelle ergänzen die Regressionsanalyse optimal, indem sie tiefere Einsichten und verbesserte Vorhersagegenauigkeit für die vielschichtigen Aspekte seismischer Aktivitäten liefern. Ihre Einbeziehung spiegelt die Bestrebung wider, ein umfassendes Verständnis seismischer Phänomene zu erlangen und die Vorhersageleistung der Studie zu maximieren.

5.3.0.2 *Bewertungskriterien*

Für die Beurteilung der Modellgenauigkeit in Bezug auf die Erdbebenvorhersage werden spezifische Evaluationsmetriken angewandt:

- **Magnitude-Vorhersage:** MAE und MSE dienen zur Messung der Genauigkeit der Magnitude-Prognosen. Niedrigere Werte indizieren eine präzisere Vorhersage.
- **Wahrscheinlichkeit starker Erdbeben:** Sensitivität, Genauigkeit und Präzision werden genutzt, um die Prognosefähigkeit bezüglich starker Erdbeben zu bewerten.

- **Sensitivität** erfasst die Rate korrekt vorhergesagter starker Erdbeben an der Gesamtzahl tatsächlicher starker Erdbeben, was für Frühwarnsysteme besonders relevant ist.
- **Genauigkeit** misst den Prozentsatz aller korrekten Vorhersagen und gibt einen Gesamtüberblick über die Modellleistung.
- **Präzision** bewertet, wie verlässlich die Vorhersage eines starken Erdbebens ist.

Die Betrachtung einer einzigen Metrik ist nicht aussagekräftig, um die Modelle zu evaluieren. Mit einem Anteil von nur 7,51% starken Erdbeben innerhalb fünf Tage im Datensatz würde eine naive Vorhersage, die immer „kein starkes Erdbeben“ annimmt, eine Genauigkeit von 92,49% erreichen. Diese Beobachtung unterstreicht, dass Genauigkeit allein keine aussagekräftige Metrik ist, da sie dazu führen könnte, das seltene, aber kritische Ereignis starker Erdbeben zu übersehen.

Die Kombination dieser Metriken ermöglicht eine umfassende Bewertung der Modellperformanz für unterschiedliche Vorhersageziele und unterstützt somit die Auswahl geeigneter Modelle für den praktischen Einsatz in der Erdbebenvorhersage. Die Betonung der Sensitivität ist besonders kritisch, da sie die Fähigkeit des Modells hervorhebt, potenziell gefährliche Ereignisse zu erkennen, was für die Risikominimierung und Notfallplanung von größter Bedeutung ist.

5.4 DATENAUFBEREITUNG

5.4.0.1 Feature Engineering

Im Rahmen der Modell-Entwicklung wurden verschiedene seismologische Features konstruiert, die aus den vorhandenen Daten abgeleitet wurden. Diese Features sind, wie unter 4.5.2 festgestellt, entscheidend, um die Vorhersagegenauigkeit des Modells zu verbessern.

SEISMISCHE ENERGIEABGABE (dE) Die seismische Energieabgabe eines Erdbebens ist ein wichtiges Merkmal, das die durch ein Erdbeben freigesetzte Energiemenge quantifiziert. Diese besitzt eine empirische ermittelte Beziehung zu der Magnitude und wird aus dieser berechnet.

MITTLERE MAGNITUDE DER LETZTEN 5 TAGE (M_{MEAN}) Für die Analyse der jüngsten seismischen Aktivität wird die mittlere Magnitude (M_{mean}) der Erdbeben, die innerhalb der letzten 5 Tage vor dem aktuellen Datum aufgetreten sind, berechnet. Dieser Durchschnitt gibt Einblick in die Intensität der seismischen Aktivität in der jüngsten Vergangenheit.

GUTENBERG-RICHTER-PARAMETER Die Gutenberg-Richter-Parameter a und b werden aus den historischen Erdbebendaten berechnet, um die Beziehung zwischen der Häufigkeit und der Magnitude von Erdbeben zu quantifizieren. Diese Parameter sind entscheidend für das Verständnis der seismischen Aktivität eines Gebiets und werden durch lineare Regression auf die logarithmierte kumulative Anzahl von Erdbeben in Abhängigkeit ihrer Magnitude ermittelt.

ZEIT SEIT DEM LETZTEN GRÖßEREN ERDBEBEN Die Zeit seit dem letzten größeren Erdbeben, definiert als ein Ereignis mit einer Magnitude größer als 5,5, wird für jedes Datum berechnet. Diese Metrik dient als Indikator für die relative Aktivität oder Ruhephase eines seismischen Gebiets.

MAGNITUDE-DEFIZIT (ΔM) Das Magnitude-Defizit (ΔM) misst die Differenz zwischen der maximal beobachteten Magnitude und der basierend auf der Gutenberg-Richter-Relation erwarteten maximalen Magnitude bis zum aktuellen Datum. Ein zunehmendes Defizit kann ein Indikator für aufgestaute seismische Spannungen sein.

DATENANREICHERUNG UND IMPUTATION Nach der Berechnung dieser Features für jede Beobachtung im Datensatz wurden die resultierenden Werte dem Datensatz hinzugefügt. Für Merkmale, bei denen Daten fehlten, wurde eine Imputation durchgeführt, um mit vollständigen Daten trainieren zu können. Die ersten 10 Zeilen des Datensatzes wurden entfernt, da für diese keine vollständigen Gutenberg-Richter-Parameter berechnet werden konnten.

ERZEUGUNG DER ZIELVARIABLEN Für die Vorhersage wurden anschließend die zwei Zielvariablen definiert:

- **Zielvariable 1:** Die Magnitude des stärksten Erdbebens, das innerhalb der nächsten 5 Tage erwartet wird. Diese Zielvariable ist kontinuierlich und reflektiert die Intensität des stärksten zu erwartenden seismischen Ereignisses.
- **Zielvariable 2:** Eine binäre Variable, die anzeigt, ob innerhalb der nächsten 5 Tage ein starkes Erdbeben mit einer Magnitude von mindestens 5.5 auftreten wird oder nicht. Diese Einstufung basiert auf der konventionellen Definition eines *starken* Erdbebens und ermöglicht eine direkte binäre Klassifizierung bevorstehender seismischer Risiken [27].

Zur Berechnung und Festlegung dieser Zielvariablen im Datensatz wurde eine maßgeschneiderte Funktion entwickelt und angewandt. Diese Funktion durchläuft den Datensatz, um für jedes verzeichnete Erdbeben das größte Nachbeben innerhalb des festgelegten 5-Tage-Zeitfensters zu identifizieren. Basierend auf

5.4.1 Aufteilung in Trainings- und Testdatensätze

Für eine zuverlässige Bewertung der Modellleistung und um die zeitliche Sequenz der Daten korrekt abzubilden, erfolgte eine sorgfältige Aufteilung des Datensatzes in Trainings- und Testdatensätze mit einem Verhältnis von 80% zu 20%. Diese spezifische Aufteilung bietet eine ausgewogene Balance zwischen einer ausreichend großen Menge an Trainingsdaten, um die Modelle effektiv zu trainieren, und genügend Testdaten, um eine aussagekräftige Evaluation der Modellleistung zu ermöglichen. Die Auswahl der jüngsten 20% der Daten als Testset stellt sicher, dass die Modelle auf die Vorhersage zukünftiger Ereignisse ausgerichtet sind, indem sie die aktuellsten Informationen im Testset evaluieren. Diese zeitbasierte Segmentierung bewahrt die chronologische Reihenfolge der Daten und verhindert Target Leakage, was entscheidend ist, um die Integrität des Bewertungsprozesses zu wahren. Die daraus resultierende Trennung unterstützt eine präzise Einschätzung der Modellfähigkeiten unter realen Bedingungen und fördert eine robuste Generalisierbarkeit auf neue, unbekannte Daten.

5.4.2 Optimierung der Modelle

FEATURE-SELEKTION Um die Genauigkeit der Modelle zu steigern, erfolgte eine gründliche Auswahl der Merkmale basierend auf Korrelationsanalysen. Diese strategische Auswahl war entscheidend, um signifikante Prädiktoren wie die mittlere Magnitude der letzten 5 Tage für die Magnitude-Vorhersage und den Gutenberg-Richter a -Wert für die Prognose des Ereigniseintritts zu identifizieren. Die Einbeziehung sowohl positiver als auch negativer Korrelationen maximiert die Vorhersagepräzision durch ein breiteres Spektrum an analysierten Einflussfaktoren, was ein tiefgreifendes Verständnis der Beziehungen zwischen den Features und den Vorhersagezielen ermöglicht. Zusätzlich kam eine automatische Feature-Selektion zum Einsatz, um präzise die aussagekräftigsten Merkmale zu bestimmen und gleichzeitig die Komplexität der Modelle zu reduzieren..

ADRESSIERUNG UNAUSGEWGENER DATENSÄTZE UND DATENSTANDARDISIERUNG Die Herausforderung durch unausgewogene Datensätze, wie unter 5.3.0.2 beschrieben, wurde durch die Integration von SMOTE (Synthetic Minority Over-sampling Technique) in die Verarbeitungspipeline bewältigt. Diese Methode verbessert die Modelleffizienz, indem sie künstliche Datenpunkte für unterrepräsentierte Klassen generiert. Für eine optimale Modellperformance wurde zudem eine Datenstandardisierung mittels 'StandardScaler' vorgenommen. Diese Normalisierung ist besonders relevant für Modelle wie ANN und LSTM, die sensibel auf unterschiedliche Datenmaßstäbe reagieren, und trägt zur Beschleunigung der Konvergenz bei den linearen Modellen und RF bei. Weiterhin wurden bei LSTM-Modellen die Trainingsdaten sequenziert, um zeitliche Abhängigkeiten effektiv zu nutzen.

HYPERPARAMETER-OPTIMIERUNG Die Feinjustierung der Modelle erfolgte durch eine systematische Hyperparameter-Optimierung mittels Grid-Search und Cross-Validation, wodurch für jedes Modell die optimale Konfiguration ermittelt wurde. Diese sorgfältige Anpassung umfasste die Variation von Schlüsselparametern wie der Anzahl der Bäume im RF, der Tiefe der Entscheidungsbäume, der Lernrate bei Boosting-Methoden sowie der Anzahl der Neuronen und der Lernrate in den neuronalen Netzwerken ANN und LSTM. Durch diesen methodischen Ansatz konnte jedes Modell spezifisch auf seine Eigenschaften und Anforderungen abgestimmt werden, was eine deutliche Steigerung der Vorhersageleistung zur Folge hatte.

5.5 LINEARE UND LOGISTISCHE REGRESSION

5.5.1 Vorhersage der Magnituden

MODELLSPEZIFIKATION Das Modell zur Vorhersage der Magnitude setzt eine Ridge Regression ein, die durch Hinzufügen eines Regularisierungsterms zur Verlustfunktion MSE Überanpassung minimiert und Multikollinearität zwischen den Prädiktoren der linearen Regression reduziert. Im Kern dieses Regularisierungsansatzes steht die Modifikation der Verlustfunktion durch den Regularisierungsterm, der die Größe der Koeffizienten beschränkt, um eine zu starke Anpassung an das Trainingset zu verhindern und die Generalisierungsfähigkeit des Modells zu verbessern.

Die Architektur des Modells beinhaltet eine umfassende Pipeline, die mit einem Preprocessing-Schritt für numerische Daten beginnt. Dieser Schritt umfasst die Behandlung fehlender Werte mittels Imputation, die Erzeugung polynomialer Features zur Erfassung nicht-linearer Zusammenhänge und die Standardisierung der Daten zur Gewährleistung einer gleichmäßigen Behandlung aller Variablen. Ausgewählt wurden Features, die für die seismologische Analyse aussagekräftig sind. Diese Auswahl basiert auf der Annahme, dass diese Variablen in einer signifikanten korrelativen Beziehung zur Zielgröße stehen (siehe Abschnitt 5.4.2). Die Optimierung des Ridge Regression Modells erfolgt über Grid-Search, bei der verschiedene Werte für den Regularisierungsparameter α getestet werden, um den MSE zu minimieren. Dieser sorgfältig abgestimmte Prozess hilft, die Modellkomplexität effektiv zu steuern und bietet einen robusten Mechanismus gegen das Problem der Überanpassung.

ERGEBNISSE

- MAE: 0.6867
- MSE: 1.5077

5.5.2 Vorhersage starker Erdbebenereignisse

MODELLSPEZIFIKATION Das Modell zur Prognose des Eintretens starker Erdbeben nutzt logistische Regression, um binäre Vorhersagen zu treffen, wobei die Auswahl der Features analog zur Ridge Regression erfolgt, um konsistente und relevante Prädiktoren einzubeziehen. Zur Adressierung des unausgewogenen Datensatzes wurde die SMOTE-Technik in die Verarbeitungspipeline integriert, bevor die Daten standardisiert wurden. Die logistische Regression verwendet als Verlustfunktion den Logarithmischen Verlust (log loss), der die Unsicherheit einer Vorhersage basierend auf der Wahrscheinlichkeit des tatsächlichen Ausgangs misst. Dieser Ansatz eignet sich besonders für binäre Klassifikationsprobleme, da er die Distanz zwischen den vorhergesagten Wahrscheinlichkeiten und den tatsächlichen Klassenwerten quantifiziert. Der Regularisierungsterm C , der in der Optimierung

des Modells variiert wurde, kontrolliert die Stärke der Regularisierung, wobei niedrigere Werte von C stärkere Regularisierung bedeuten. Dies hilft, das Risiko von Überanpassung zu minimieren. Der Strafterm *penalty* bezieht sich auf die Art der Regularisierung, wobei 'l2' die Quadratsumme der Koeffizienten bestraft und somit die Modellkomplexität effektiv kontrolliert. Die Feinabstimmung dieser Hyperparameter, einschließlich des Regularisierungsterms C und des Strafterms *penalty*, wurde mittels Grid-Search durchgeführt, um die optimale Konfiguration für maximale Vorhersagegenauigkeit zu ermitteln. Diese systematische Methode zur Hyperparameter-Optimierung stellt sicher, dass das Modell sowohl robust als auch präzise in der Vorhersage des Eintretens starker Erdbeben ist.

ERGEBNISSE

- Sensitivität: 0.7006
- Genauigkeit: 0.6510
- Präzision: 0.1377

5.6 RANDOM FOREST

5.6.1 Vorhersage der Magnituden

MODELLSPEZIFIKATION Die konzipierte Pipeline zur Prognose der Magnituden seismischer Ereignisse integriert diverse Verarbeitungsschritte, um die Daten adäquat auf die Modellanwendung vorzubereiten und die prädiktive Effizienz zu maximieren. Initial werden die Feature-Daten standardisiert. Daran anschließend findet eine Selektion der Merkmale statt, basierend auf deren Relevanz, die durch den Einsatz eines ‘RandomForestRegressor’ ermittelt wird. Das Kernstück der Pipeline bildet besagter ‘RandomForestRegressor’, der primär die Aufgabe der Magnitudenvorhersage übernimmt.

Zur Feinabstimmung der Hyperparameter des Random-Forest-Regressors dient das Verfahren ‘GridSearchCV’, welches eine systematische Durchsuchung diverser Hyperparameter-Konfigurationen ermöglicht. Variiert werden dabei insbesondere die Anzahl der Entscheidungsbäume, deren maximale Tiefe, die minimale Anzahl von Stichproben, die für das Teilen eines Knotens erforderlich sind, sowie die minimale Stichprobengröße an einem Blattknoten. Ziel ist es, die Konfiguration zu identifizieren, die die höchste prädiktive Leistung aufweist, gemessen am MSE.

ERGEBNISSE

- MAE: 0.8181
- MSE: 1.6796

5.6.2 Vorhersage starker Erdbebenereignisse

MODELLSPEZIFIKATION Die für die Vorhersage des Eintretens starker seismischer Aktivitäten entwickelte Pipeline nutzt den ‘RandomForestClassifier’, um eine effiziente und präzise Klassifizierung zu ermöglichen. In enger Analogie zur Magnitudenprognose beginnt der Prozess mit einer Standardisierung der Merkmale, um eine konsistente Datenbasis für das Training zu schaffen. Eine sorgfältige Auswahl der Merkmale, basierend auf ihrer Bedeutung, die mittels ‘RandomForestClassifier’ und der ‘SelectFromModel’-Methode ermittelt wird, folgt als nächster Schritt. Diese Merkmalsauswahl gewährleistet, dass nur die informativsten Merkmale in die Modellbildung einfließen.

Für die Einstellung des ‘RandomForestClassifier’ werden optimierte Hyperparameter verwendet, die durch eine systematische Anwendung von ‘GridSearchCV’ ermittelt wurden. Dieser Prozess ähnelt der Methodik, die bei der Magnitudenprognose angewandt wurde, und beinhaltet die Anpassung von Schlüsselparametern wie der Anzahl der Entscheidungsbäume und deren maximale Tiefe, um die bestmögliche Vorhersageleistung, gemessen am F1-Score, zu erreichen.

Nach der Feinabstimmung der Hyperparameter und dem Abschluss des Trainingsprozesses generiert das Modell Wahrscheinlichkeitsvorhersagen für das Auftreten starker Beben. Die Bestimmung des optimalen Schwellenwerts, der zur finalen binären Klassifikation herangezogen wird, basiert auf der Analyse der Präzisions-Recall-Kurve und der Maximierung des F1-Scores. Diese methodische Herangehensweise stellt sicher, dass das Modell eine ausgewogene Performance hinsichtlich der Präzision und des Recalls bietet, was für die zuverlässige Vorhersage des Eintretens starker Erdbebenereignisse entscheidend ist.

ERGEBNISSE

- Sensitivität: 0.7628
- Genauigkeit: 0.5683
- Präzision: 0.2158

5.7 KÜNSTLICHES NEURONALES NETZ

5.7.1 *Vorhersage der Magnituden*

MODELLSPEZIFIKATION Das entwickelte ANN zur Vorhersage der Erdbebenmagnitude implementiert ein systematisches Design, beginnend mit einer Standardisierung der Eingabedaten. Dieses Sequential-Modell ist strategisch mit mehreren Schichten aufgebaut, darunter zwei verborgene Schichten, jede mit 128 Neuronen und aktiviert durch die ReLU-Funktion, um nicht-lineare Beziehungen effektiv zu modellieren. Zur Reduzierung von Überanpassung sind zwischen diesen Schichten Dropout-Schichten mit einer Rate von 0.2 eingebettet. Die finale Ausgabeschicht, gestaltet für die Regressionsaufgabe, besteht aus einem einzigen Neuron.

Für die Steuerung des Trainingsprozesses wird der Adam-Optimierer mit einer Lernrate von 0.001 verwendet, während der MSE als Verlustfunktion dient, um die Differenz zwischen den prognostizierten und tatsächlichen Magnituden zu quantifizieren. Ein Early-Stopping-Mechanismus, konfiguriert mit einer Geduld von 10 Epochen, überwacht den Validierungsverlust, um das Training bei fehlender Verbesserung frühzeitig zu beenden und die Modellgewichte bei bester Performance wiederherzustellen. Diese Maßnahme trägt wesentlich dazu bei, das Risiko einer Überanpassung zu minimieren und fördert die Erhaltung der Generalisierungsfähigkeit des Modells.

ERGEBNISSE

- MAE: 0.7265
- MSE: 1.4639

5.7.2 *Vorhersage starker Erdbebenereignisse*

MODELLSPEZIFIKATION Das entwickelte ANN zur Prognose des Auftretens starker Erdbebenereignisse implementiert eine systematische Herangehensweise, die die Balance der Trainingsdaten durch den Einsatz von SMOTE und die Standardisierung der Merkmale priorisiert. Das Modell besteht aus einer Eingabeschicht mit 128 Neuronen, gefolgt von einer verborgenen Schicht mit 64 Neuronen, wobei beide Schichten die ReLU-Aktivierungsfunktion nutzen. Um Überanpassung entgegenzuwirken, werden nach jeder dieser Schichten Dropout-Schichten mit einer Rate von 0.5 eingeführt. Die finale Schicht nutzt eine Sigmoid-Funktion, um die binäre Klassifikation durchzuführen.

Für den Trainingsvorgang wird der Adam-Optimierer mit einer Lernrate von 0.001 verwendet, wobei die binäre Kreuzentropie als Verlustfunktion dient, um die Genauigkeit der binären Klassifikationsvorhersagen zu messen. Die Modellperformance wird zudem durch die Präzisionsmetrik (Precision) evaluiert. Ein Early-Stopping-Mechanismus, der bei einer ausbleibenden Verbesserung der Validierungsverluste nach 10 Epochen das Training

abbricht, trägt zur Optimierung bei und verhindert eine Überanpassung des Modells an die Trainingsdaten. Dies gewährleistet, dass das Modell eine hohe Generalisierbarkeit auf neue, unbekannte Daten behält und effektiv das Auftreten starker Erdbebenereignisse vorhersagen kann.

ERGEBNISSE

- Sensitivität: 0.6731
- Genauigkeit: 0.8200
- Präzision: 0.4204

5.8 LONG SHORT-TERM MEMORY

5.8.1 *Vorhersage der Magnituden*

MODELLSPEZIFIKATION Das LSTM-Modell optimiert die Vorhersage der Erdbebenmagnitude durch eine präzise Vorbereitung und Verarbeitung der Daten. Nach der Initialisierung durch Datenstandardisierung transformiert das Modell die Daten in sequenzielle Abschnitte von jeweils 10 aufeinanderfolgenden Messungen. Diese Technik ist essenziell, um den Kontext und die zeitlichen Muster zu erfassen, die entscheidend für das Erkennen von Anzeichen bevorstehender signifikanter seismischer Ereignisse sind. Durch diese spezifische Sequenzierung wird die Fähigkeit des Modells, langfristige Abhängigkeiten zu lernen und die Dynamik seismischer Aktivitäten zu interpretieren, signifikant verbessert.

Die Architektur des LSTM-Modells umfasst eine erste Schicht mit 128 Einheiten, aktiviert durch die tanh-Funktion, und integriert Dropout-Schichten mit einer Rate von 0.2 zur Vermeidung von Overfitting. Ergänzt wird die Struktur durch eine dichte Schicht mit 128 Neuronen und ReLU-Aktivierung, gefolgt von einer weiteren Dropout-Schicht, die das Modell in der Identifikation komplexer Muster unterstützt. Die kontinuierliche Vorhersage der Magnitude wird durch eine einzelne Ausgabeschicht ermöglicht.

Techniken wie Early Stopping und Lernratenreduktion sind zentrale Bestandteile der Trainingsstrategie, um das Training zu optimieren und gleichzeitig Overfitting zu minimieren. Diese Maßnahmen unterstützen eine effiziente Anpassung des Lernprozesses und sichern die hohe Generalisierbarkeit des Modells.

Insgesamt ermöglicht diese spezialisierte Modellkonfiguration, gepaart mit der gezielten Vorbereitung der Daten, eine präzise und zuverlässige Vorhersage der Erdbebenmagnitude. Die systematische Trainingsstrategie, unterstützt durch fortschrittliche Techniken zur Optimierung und Vermeidung von Überanpassung, trägt entscheidend zur Verbesserung der Vorhersageleistung bei.

ERGEBNISSE

- MAE: 0.6679
- MSE: 1.2887

5.8.2 *Vorhersage starker Erdbebenereignisse*

MODELLSPEZIFIKATION Für die Prognose des Eintretens starker Erdbeben wurde ein spezifisch konzipiertes LSTM-Modell entwickelt, das eine sorgfältige Standardisierung und Sequenzierung der Daten analog zum Modell für die Magnitudenvorhersage durchführt. Um die Herausforderung der Datenbalance anzugehen, integriert dieses Modell SMOTE für das Trainingsset. Die Architektur dieses spezialisierten Modells umfasst zwei LSTM-

Schichten, wobei die erste Schicht sequenzielle Daten für die nachgelagerte Verarbeitung bereitstellt, während die zweite Schicht die gewonnenen Einsichten verdichtet. Dropout-Schichten mit einer hohen Rate von 0.5 folgen auf jede LSTM-Schicht, um der Überanpassung entgegenzuwirken. Die binäre Klassifikation erfolgt durch eine Output-Schicht, die mit einer Sigmoid-Funktion ausgestattet ist, optimiert für die Vorhersage der Wahrscheinlichkeit eines starken Bebens.

Zur Steuerung des Trainingsprozesses und zur Optimierung der Modellleistung wird der Adam-Optimierer mit einer feinjustierten Lernrate von 0.001 verwendet, während die binäre Kreuzentropie als Verlustfunktion dient, um die Genauigkeit der Vorhersagen zu bewerten. Die Implementierung von Präzision als Metrik und der Einsatz von Early Stopping und Lernratenreduktion dienen der weiteren Verfeinerung des Modells, minimieren das Risiko von Overfitting und unterstützen die Identifikation des optimalen Schwellenwerts basierend auf dem F1-Score. Diese umfassenden Strategien gewährleisten eine präzise und zuverlässige Prognose des Auftretens starker Erdbebenereignisse.

ERGEBNISSE

- Sensitivität: 0.8830
- Genauigkeit: 0.3882
- Präzision: 0.1758

5.9 ENSEMBLE-MODELLE

5.9.1 Vorhersage der Magnituden

MODELLSPEZIFIKATION Das Ensemble-Modell ist ein Boosting-Verfahren und basiert auf der Integration der Präzision der individuellen Modelle (LinR, RF, ANN und LSTM) zur Gewichtung ihrer Vorhersagen. Die Gewichte, abgeleitet aus den Präzisionswerten der einzelnen Modelle, werden normalisiert, sodass ihre Summe eins ergibt. Diese Methodik erlaubt eine differenzierte Berücksichtigung der Modelle basierend auf ihrer Vorhersagegenauigkeit.

Für die Synthese der Vorhersagen wird eine gewichtete Kombination der einzelnen Prognosen vorgenommen. Hierbei werden die Vorhersagen der Modelle mit ihren jeweiligen Gewichten multipliziert, um eine gewichtete Vorhersage zu erstellen. Da AdaBoost, als Meta-Modell verwendet, nicht direkt mit kontinuierlichen Werten umgehen kann, wird der gewichtete Durchschnitt der Vorhersagen als einziges Merkmal für das Training von AdaBoost herangezogen.

AdaBoost, gekoppelt mit einem DT Classifier als Basis-Lerner, wird daraufhin mit diesen gewichteten Durchschnittswerten trainiert. Diese Konfiguration zielt darauf ab, die Stärken der individuellen Modelle zu nutzen und gleichzeitig ihre Schwächen zu mildern, indem ein Meta-Modell erstellt wird, das fähig ist, eine generalisierte Vorhersage auf Basis der kombinierten Einblicke zu treffen.

ERGEBNISSE

- MAE: 0.8575
- MSE: 1.6332

5.9.2 Vorhersage starker Erdbebenereignisse

MODELLSPEZIFIKATION Auch zur Vorhersage der starken Erdbebenereignisses wird ein Boosting-Verfahren durch AdaBoost als Meta-Modell verwendet. Dieser Ansatz beginnt mit der Sammlung von Vorhersagen der individuellen Modelle (LogR, RF, ANN und LSTM).

Die Kernidee besteht darin, die Präzision jedes Modells als Grundlage für die Gewichtung seiner Vorhersagen zu verwenden. Indem die Präzisionswerte in ein Array von Gewichten umgewandelt werden, die anschließend normalisiert werden, um sicherzustellen, dass ihre Summe eins ergibt, wird eine differenzierte Betrachtung der Modellvorhersagen ermöglicht. Durch Multiplikation der Vorhersagen mit diesen Gewichten und der Berechnung eines gewichteten Durchschnitts wird eine aggregierte Vorhersage erstellt, die die Stärken der einzelnen Modelle berücksichtigt und deren Schwächen mindert.

AdaBoost nutzt einen Decision Tree Classifier als Basis-Lerner und wird mit diesen gewichteten Durchschnittswerten trainiert. AdaBoost adaptiert sich iterativ, indem es den Fokus auf jene Fälle legt, die von vorherigen Iterationen falsch vorhergesagt wurden, und verstärkt so die Vorhersagegenauigkeit des Ensembles. Die Auswahl von AdaBoost mit Entscheidungsbäumen geringer Tiefe (`max_depth=1`) und einer spezifischen Anzahl an Iterationen (`n_estimators=100`) zielt darauf ab, ein ausgewogenes Modell zu erstellen, das effektiv generalisieren kann.

ERGEBNISSE

- Sensitivität: 0.6248
- Genauigkeit: 0.8731
- Präzision: 0.5496

5.10 ANALYSE UND REFLEXION DER STUDIENERGEBNISSE

5.10.1 *Ergebnisse der Modelle im Detail*

Tabelle 5.1: Evaluationsergebnisse für die Vorhersage der Erdbebenmagnitude

Metrik	LR	RF	ANN	LSTM	Ensemble
MAE	0.6867	0.8181	0.7265	0.6679	0.8575
MSE	1.5077	1.6796	1.4639	1.2887	1.6332

Tabelle 5.2: Evaluationsergebnisse für die Vorhersage des Auftretens starker Erdbebenereignisse

Metrik	LR	RF	ANN	LSTM	Ensemble
Sensitivität	0.7006	0.7628	0.6731	0.8830	0.6248
Genauigkeit	0.6510	0.5683	0.8200	0.3882	0.8731
Präzision	0.1377	0.2158	0.4204	0.1758	0.5496

Bei der Vorhersage der Erdbebenmagnitude zeigt das LSTM-Modell die besten Ergebnisse hinsichtlich der MAE- und MSE-Metriken mit einem MAE von 0.6679 und einem MSE von 1.2887, was auf seine Fähigkeit hinweist, langfristige Abhängigkeiten in den Zeitreihendaten effektiv zu nutzen und präzise Magnitudenvorhersagen zu liefern.

Für die Prognose des Auftretens starker Erdbebenereignisse sticht das Ensemble-Modell mit der höchsten Genauigkeit von 0.8731 und Präzision von 0.5496 hervor, was seine Überlegenheit bei der Klassifizierung und die effektive Kombination der Stärken einzelner Modelle unterstreicht. Insbesondere die hohe Präzision des Ensemble-Modells zeigt, dass es bei der Vorhersage starker Erdbebenereignisse weniger falsch positive Ergebnisse liefert. Allerdings zeigt das LSTM-Modell die höchste Sensitivität mit 0.8830, was darauf hindeutet, dass es am besten geeignet ist, tatsächliche Ereignisse zu identifizieren, auch wenn dies auf Kosten einer höheren Rate an falsch positiven Ergebnissen geschieht. Diese Fähigkeit kann besonders wertvoll sein, um potenzielle Erdbebenereignisse nicht zu übersehen.

5.10.2 *Tiefgreifende Diskussion der Studienbefunde*

Das Feature Engineering erwies sich als wesentlicher Erfolgsfaktor für die in dieser Studie verfolgten Vorhersageziele, wie in Abschnitt 4.5.2 dargestellt. Durch gezielte Auswahl und methodische Aufbereitung der Features, basierend auf seismologischen Berechnungen, konnte eine signifikante Steigerung der Vorhersageeffizienz der ML-Modelle erreicht werden. Dies hebt die entscheidende Bedeutung der sorgfältigen Analyse und Vorbereitung der Eingabedaten für die Modellpräzision hervor. Insbesondere die effektive Extraktion und Anwendung relevanter Merkmale aus komplexen Datensät-

zen waren ausschlaggebend, um die Prognosegenauigkeit entscheidend zu verbessern und den spezifischen Anforderungen der Erdbebenvorhersage gerecht zu werden.

In der weiteren Diskussion werden die Leistungen der für diese Untersuchung entwickelten Modelle zur Prognose von Erdbebenstärken und der Wahrscheinlichkeit signifikanter seismischer Ereignisse eingehend analysiert. Dies geschieht durch den Vergleich mit Ergebnissen aus relevanten Studien und durch die Integration der eigenen erzielten Resultate.

DISKUSSION DER VORHERSAGE DER MAGNITUDEN Die Genauigkeit bei der Vorhersage von Erdbebenmagnituden, dargestellt durch das LSTM-Modell mit einem MAE von 0.6679 und einem MSE von 1.2887, findet Parallelen in ähnlichen Forschungsarbeiten. Beispielsweise erzielten das SVM-Modell von Murwantara et al. (2020) und das Modell von Salam et al. (2021) mit MAE-Werten von 0.598 bzw. RMSE-Werten von 0.5655 in ihren Untersuchungen zur Vorhersage von Erdbebenmagnituden vergleichbare Effektivitätsniveaus. Diese Übereinstimmung mit den Ergebnissen der aktuellen Studie betont den wesentlichen Beitrag zur Steigerung der Genauigkeit in der Erdbebenvorhersage. Es zeigt sich, dass das in dieser Forschung entwickelte Modell unter unterschiedlichen Bedingungen Leistungen erzielt, die mit denen der erwähnten Studien vergleichbar oder sogar überlegen sind. Dies bekräftigt den Wert und die Wichtigkeit der Weiterentwicklung und gezielten Anpassung von Vorhersagemodellen an die vielschichtigen Muster seismischer Daten [45, 54].

Demgegenüber erzielte das FLANN-Modell von Majhi et al. (2019) unter Verwendung vergleichbarer seismischer Inputdaten einen herausragenden RMSE von 0.0565. Diese bemerkenswerte Präzision wurde jedoch speziell für Erdbeben mit Magnituden über 5.5 erreicht, was Zweifel aufkommen lässt, ob eine derart hohe Genauigkeit über ein breiteres Spektrum von Erdbebenmagnituden durch das Modell erzielt werden könnte und inwiefern diese Genauigkeit spezifisch dem Modellansatz zuzuschreiben ist [41].

Ein weiteres beachtliches Ergebnis wurde mit dem EEWNet von Wang et al. (2023) erzielt, das einen MAE von 0.28 basierend auf der Analyse von P-Wellen-Signalen erreichte. Dieses Ergebnis unterstreicht das signifikante Potential, durch die Einbindung zusätzlicher Datenquellen die Präzision von Erdbebenvorhersagen zu steigern [65].

DISKUSSION DER VORHERSAGE STARKER ERDBEBENEREIGNISSE Die Analyse des Auftretens starker Erdbebenereignisse und deren vergleichende Betrachtung mit etablierten Forschungsarbeiten unterstreichen die Effizienz des in dieser Arbeit vorgestellten Ensemble-Modells, welches mit einer Genauigkeit von 0.8731 und einer Präzision von 0.5496 beeindruckt. Das LSTM-Modell weist mit einer Sensitivität von 0.8830 ebenfalls exzellente Leistung auf. Diese Befunde heben sich innerhalb der wissenschaftlichen Gemeinschaft ab, indem sie nicht nur außerordentliche Resultate präsentieren, sondern auch das Potenzial maßgeschneiderter Modellierungsansätze

sowie die essenzielle Rolle präziser seismischer Datenanalyse betonen. Beispielsweise erzielten Shodiq et al. (2018) im Vergleich mit ihrem ANN bei der Prognose von Nachbeben der Stärke größer als 6.0 in einer spezifischen indonesischen Region eine niedrigere Genauigkeit (zwischen 56% und 72%), hohe Präzisionswerte (zwischen 86% und 96%) sowie eine niedrigere Sensitivität (zwischen 0% und 38%) [60].

Diese Befunde unterstreichen einerseits die relative Überlegenheit der eigenen Modellergebnisse, reflektieren jedoch auch die inhärente Variabilität von Leistungsmetriken, wie sie in der eigenen Untersuchung festgestellt wurde. Die Variabilität in den Ergebnissen von Shodiq et al. illustriert die übliche Bandbreite der Modellleistung, die auf verschiedene Faktoren, einschließlich geografischer Spezifika und Zieldefinitionen, zurückzuführen ist. Dies untermauert die Bedeutung einer kontinuierlichen Bewertung und Anpassung von Vorhersagemodellen an die zugrundeliegenden Datenstrukturen und spezifischen Anwendungskontexte.

Florido et al. (2018) konnten bei der Erkennung seismischer Vorläufermuster eine beeindruckende Genauigkeit von bis zu 93,59% für Erdbeben mit einer Magnitude über 4,5 am nächsten Tag erzielen, mit Sensitivitätsraten von 78.57% und einer Spezifität von 95.07% [24]. Obwohl die Vergleichbarkeit dieser Ergebnisse durch unterschiedliche methodologische Ansätze und geografische Unterschiede eingeschränkt ist, illustrieren sie dennoch, wie die zeitliche Proximität zu einem Ereignis die Vorhersagegenauigkeit erheblich steigern kann. Für das eigene Modell deutet dies darauf hin, dass eine Fokussierung auf kurzfristigere Vorhersagezeiträume potenziell die Genauigkeit verbessern könnte, indem es die dynamischen Veränderungen in den seismischen Mustern präziser abbildet.

Chittora et al. (2022) demonstrierten mit dem XGBoost Tree-Algorithmus in den Andamanen und Nikobaren herausragende Ergebnisse mit einer Präzision von 99.27%, einer Sensitivität von 99.31% und einer Genauigkeit von 99.10%, während in Gujarat deutlich niedrigere Leistungswerte des gleichen Modells von 65.16% Präzision, 69.64% Sensitivität und 72.96% Genauigkeit beobachtet wurden [16]. Diese Diskrepanz unterstreicht die Bedeutung der geographischen Region und der Notwendigkeit, Vorhersagemodelle an lokale Besonderheiten anzupassen, um ihre Vorhersagekraft zu optimieren. Für das in dieser Arbeit entwickelte ML-Modell impliziert dies, dass eine hohe Generalisierbarkeit über verschiedene Regionen hinweg nicht vorausgesetzt werden kann, ohne gezielte Anpassungen und Validierungen vorzunehmen. Entsprechend muss die Leistungsbewertung unseres ML-Modells nicht nur im Hinblick auf seine algorithmische Effizienz, sondern auch in Bezug auf seine Fähigkeit zur regionalen Anpassung und Flexibilität betrachtet werden, wie bereits unter der Diskussion zur Generalisierbarkeit (siehe Abschnitt 4.5.3) erörtert wurde.

Des Weiteren offenbart die Studie von Wang et al. (2020) zusätzliches Optimierungspotential für das eigene Modell durch die Integration umfangreicherer Datenmengen. Durch die Einbeziehung zweidimensionaler Dateininputs, welche die raum-zeitlichen Korrelationen zwischen Erdbebenereig-

nissen in China umfassend abbilden, konnten Wang et al. eine signifikante Steigerung der Genauigkeit, Sensitivität und Präzision erreichen [64]. Dies demonstriert eindrucksvoll, wie eine erweiterte Datengrundlage die Modellleistung verbessern kann. Für das eigene Modell könnte dies bedeuten, dass durch die Einbeziehung zusätzlicher, relevanter Datenquellen, wie etwa geologischer oder klimatischer Informationen, die Präzision der Erdbebenvorhersagen ebenfalls deutlich erhöht werden könnte.

SCHLUSSFOLGERUNGEN UND AUSBLICK

6.1 ZUSAMMENFASSUNG DER ERGEBNISSE

Die Analyse von ML-Modellen für die Erdbebenvorhersage hat signifikante Fortschritte im Vergleich zu traditionellen seismologischen Methoden offenbart. ML-Modelle, insbesondere Ensemble-Ansätze und DL, haben eine erhöhte Genauigkeit und Effizienz in der Erkennung seismischer Muster sowie in der Vorhersage von Zeitpunkt, Ort und Stärke von Erdbeben gezeigt. Durch den systematischen Vergleich wurde deutlich, dass ML-Modelle durch ihre Fähigkeit, komplexe Muster in den Daten zu erkennen und aus Erfahrungen zu lernen, traditionellen Methoden überlegen sind. Die Identifikation und Evaluation der robustesten ML-Modelle für spezifische Vorhersageparameter haben LSTM und CNN als besonders effektiv in der Modellierung zeitabhängiger Dynamiken und der Erkennung räumlicher Muster offenbart. Die Präzision dieser Modelle offenbart sich nicht ausschließlich durch ihren Einsatz, sondern primär durch eine systematische Auswahl der Eingabedaten und Zielvariablen, kombiniert mit einer gezielten Anpassung spezifischer Algorithmen, wie es aktuelle Studien und die eigene Implementierung eindrücklich bewiesen haben. Die Verwendung identischer Algorithmen kann aufgrund der Eigenheiten des Datensatzes, der angestrebten Zielsetzung und geographischen Gegebenheiten zu divergenten Ergebnissen führen. Der zentrale Ansatzpunkt zur Steigerung der prognostischen Genauigkeit liegt in der präzisen Auswahl und Aufbereitung der Daten sowie der akkuraten Justierung der Algorithmen, um sie den spezifischen Merkmalen und Anforderungen der seismologischen Datensätze anzupassen. Die Optimierung der Datennutzung, insbesondere die Integration diverser Datenquellen und gezieltes Feature Engineering, um seismische Merkmale abzubilden, trägt dabei signifikant zur Verbesserung der Vorhersagegenauigkeit bei, wie die untersuchten Modelle und die eigene Implementierung gezeigt haben.

6.2 LIMITATIONEN DER STUDIE

Trotz bemerkenswerter Fortschritte weisen ML-Modelle in der Erdbebenvorhersage spezifische Limitationen auf. Die Herausforderungen in Bezug auf die Verfügbarkeit und Qualität der Trainingsdaten, die Generalisierbarkeit der Modelle über verschiedene geografische Gebiete hinweg und die Interpretierbarkeit komplexer ML-Modelle wurden identifiziert. Insbesondere beeinflusst die Vollständigkeit und Genauigkeit der Trainingsdaten die Leistungsfähigkeit der ML-Modelle erheblich. Die Übertragbarkeit der Modelle auf unterschiedliche Regionen mit abweichenden seismischen Eigenschaften

erwies sich als kritisch für die Generalisierbarkeit der Vorhersagen. Zudem stellte die begrenzte Nachvollziehbarkeit der Entscheidungsprozesse in tiefen Lernverfahren eine signifikante Herausforderung dar, welche die praktische Anwendbarkeit der Modelle einschränken kann.

Eine weitere grundlegende Limitation betrifft die intrinsische Unvorhersehbarkeit von Erdbeben und das für die Modellentwicklung erforderliche tiefgreifende seismologische Wissen. Erdbeben resultieren aus komplexen geologischen Prozessen, die durch tektonische Bewegungen angetrieben werden und durch zahlreiche, schwer zu modellierende Faktoren beeinflusst werden können. Ohne ein umfassendes Verständnis dieser Prozesse und aller relevanten Variablen bleibt die Genauigkeit von ML-Modellen in der Vorhersage von Zeitpunkt, Ort und Stärke zukünftiger Erdbeben begrenzt. Die Weiterentwicklung der Erdbebenvorhersage mittels ML hängt entscheidend von fortgesetzter seismologischer Forschung und einem verbesserten Verständnis der zugrundeliegenden geophysikalischen Dynamiken ab.

6.3 PRAKTISCHE IMPLIKATIONEN UND EMPFEHLUNGEN FÜR ZUKÜNFTIGE FORSCHUNG

Diese Studie hebt nicht nur das Potenzial von ML zur Verfeinerung der Erdbebenvorhersage hervor, sondern weist auch auf bedeutende praktische Implikationen für die Bereiche Katastrophenprävention und -management hin. Für die Zukunft der seismologischen Forschung ist es von entscheidender Bedeutung, die Entwicklung und Anwendung fortschrittlicher ML-Algorithmen voranzutreiben. Dabei sollte ein besonderer Fokus auf der Verbesserung der Qualität und Diversität der genutzten Datensätze liegen. Darüber hinaus ist die Schaffung standardisierter Evaluationskriterien unabdingbar, um eine objektive Beurteilung der Leistungsfähigkeit unterschiedlicher ML-Modelle zu ermöglichen.

Die Einbeziehung von Satellitenbildern und umfassenden geophysikalischen Daten verspricht eine signifikante Steigerung der Vorhersagepräzision. Ein systematischer Ansatz für die Erstellung und Nutzung einheitlicher Benchmark-Datensätze würde es der Forschungsgemeinschaft erlauben, verschiedene Modelle effektiv zu vergleichen und die leistungsfähigsten Methoden zu identifizieren. Diese Empfehlungen bilden einen strategischen Rahmen für zukünftige Forschungsinitiativen, die darauf abzielen, die Genauigkeit, Zuverlässigkeit und Anwendbarkeit von ML-gestützten seismologischen Vorhersagemodellen kontinuierlich zu verbessern. Die systematische Adressierung dieser Aspekte wird nicht nur die wissenschaftliche Forschung bereichern, sondern auch praktische Fortschritte im Katastrophenmanagement ermöglichen, indem effektivere Vorwarnsysteme und Risikominderungsstrategien entwickelt werden.

Teil II

APPENDIX



JUPYTER NOTEBOOK ZUR ERDBEBENVORHERSAGE

EarthquakePrediction

April 1, 2024

1 Bibliotheken importieren

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import folium
from folium.plugins import MarkerCluster

from datetime import timedelta

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLRonPlateau
from tensorflow.keras.metrics import Precision

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder,
↳PolynomialFeatures
from sklearn.impute import SimpleImputer
from sklearn.linear_model import Ridge, LogisticRegression, LinearRegression
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier,
↳AdaBoostClassifier
from sklearn.metrics import (
    mean_absolute_error, mean_squared_error, r2_score,
    accuracy_score, precision_score, recall_score, f1_score,
    precision_recall_curve, roc_auc_score
)
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.feature_selection import SelectFromModel
from sklearn.tree import DecisionTreeClassifier
```

```

from imblearn.pipeline import Pipeline as ImbPipeline
from imblearn.over_sampling import SMOTE

from skopt import gp_minimize
from skopt.space import Real, Integer

from scipy.stats import linregress

```

2 Daten importieren

```
[2]: all_data = pd.read_csv("Eartquakes-1990-2023.csv")
```

3 Data Understanding

```
[3]: all_data.head()
```

```
[3]:
```

	time	place	status	tsunami	\
0	631153353990	12 km NNW of Meadow Lakes, Alaska	reviewed	0	
1	631153491210	14 km S of Volcano, Hawaii	reviewed	0	
2	631154083450	7 km W of Cobb, California	reviewed	0	
3	631155512130	11 km E of Mammoth Lakes, California	reviewed	0	
4	631155824490	16km N of Fillmore, CA	reviewed	0	

	significance	data_type	magnitudo	state	longitude	latitude	\
0	96	earthquake	2.50	Alaska	-149.669200	61.730200	
1	31	earthquake	1.41	Hawaii	-155.212333	19.317667	
2	19	earthquake	1.11	California	-122.806167	38.821000	
3	15	earthquake	0.98	California	-118.846333	37.664333	
4	134	earthquake	2.95	California	-118.934000	34.546000	

	depth	date
0	30.100	1990-01-01 00:22:33.990000+00:00
1	6.585	1990-01-01 00:24:51.210000+00:00
2	3.220	1990-01-01 00:34:43.450000+00:00
3	-0.584	1990-01-01 00:58:32.130000+00:00
4	16.122	1990-01-01 01:03:44.490000+00:00

```
[4]: all_data.describe()
```

```
[4]:
```

	time	tsunami	significance	magnitudo	longitude	\
count	3.445751e+06	3.445751e+06	3.445751e+06	3.445751e+06	3.445751e+06	
mean	1.247124e+12	4.434447e-04	7.400973e+01	1.774076e+00	-1.012876e+02	
std	2.976292e+11	2.105346e-02	1.016364e+02	1.291055e+00	7.697416e+01	
min	6.311534e+11	0.000000e+00	0.000000e+00	-9.990000e+00	-1.799997e+02	

```

25%    1.024401e+12  0.000000e+00  1.300000e+01  9.100000e-01 -1.464274e+02
50%    1.282338e+12  0.000000e+00  3.300000e+01  1.460000e+00 -1.189538e+02
75%    1.508701e+12  0.000000e+00  8.100000e+01  2.300000e+00 -1.159277e+02
max     1.690629e+12  1.000000e+00  2.910000e+03  9.100000e+00  1.800000e+02

```

```

          latitude      depth
count  3.445751e+06  3.445751e+06
mean   3.746483e+01  2.285387e+01
std    2.041577e+01  5.484938e+01
min   -8.442200e+01 -1.000000e+01
25%    3.406400e+01  3.120000e+00
50%    3.793567e+01  7.700000e+00
75%    4.784800e+01  1.612000e+01
max    8.738600e+01  7.358000e+02

```

```
[5]: all_data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3445751 entries, 0 to 3445750
Data columns (total 12 columns):
#   Column      Dtype
---  -
0   time        int64
1   place       object
2   status      object
3   tsunami    int64
4   significance int64
5   data_type   object
6   magnitudo  float64
7   state       object
8   longitude   float64
9   latitude    float64
10  depth       float64
11  date        object
dtypes: float64(4), int64(3), object(5)
memory usage: 315.5+ MB

```

4 Preprocessing

```
[6]: all_data.isna().sum()
```

```

[6]: time          0
     place         0
     status        0
     tsunami       0
     significance  0
     data_type     0

```

```
magnitudo      0
state           0
longitude       0
latitude        0
depth           0
date            0
dtype: int64
```

Da keine Null-Values vorhanden sind ist kein Preprocessing notwendig, außer:

```
[7]: # Doppelte Zeilen entfernen
all_data = all_data.drop_duplicates()

# Bereinigen der Daten, um sicherzustellen, dass alle Einträge konsistent sind
↳ (z.B. California)
all_data['state'] = all_data['state'].str.strip()

# Konvertiere 'date'-Spalte in das datetime-Format
all_data['date'] = pd.to_datetime(all_data['date'])

# Filtere den DataFrame, um nur die Zeilen mit data_type = 'Erdbeben' zu behalten
all_data = all_data[all_data['data_type'] == 'earthquake']

# Entferne die Spalte 'data_type'
all_data = all_data.drop(columns=['data_type'])

# Zeilen löschen, bei denen Magnitudo gleich 0 ist
all_data = all_data[all_data['magnitudo'] != 0]
```

```
[8]: # Überprüfen ob Time und Date das gleiche sind - ja

# Konvertiere 'time' von Unix-Zeitstempel in Millisekunden zu 'datetime'
test_data = pd.DataFrame()
test_data['time_converted'] = pd.to_datetime(all_data['time'], unit='ms',
↳ utc=True)

# Vergleiche die konvertierte 'time' Spalte mit 'date'
# Hier wird geprüft, ob die Differenz zwischen den beiden Spalten minimal ist
test_data['time_diff'] = test_data['time_converted'] - all_data['date']

# Überprüfe die ersten paar Zeilen der Differenz
print(test_data['time_diff'].head())
```

```
0    0 days
1    0 days
2    0 days
3    0 days
4    0 days
```

```
Name: time_diff, dtype: timedelta64[ns]
```

5 Festlegen auf Gebiet der Türkei

Die Türkei liegt ungefähr zwischen 36° und 42° nördlicher Breite (Latitude) und 26° und 45° östlicher Länge (Longitude).

```
[9]: # Definiere die geografischen Grenzen der Türkei
latitude_min, latitude_max = 32, 42
longitude_min, longitude_max = 26, 45

# Filtere Daten für das Gebiet der Türkei
data = all_data[(all_data['latitude'] >= latitude_min) & (all_data['latitude']
↳<= latitude_max) &
                (all_data['longitude'] >= longitude_min) & (all_data['longitude']
↳<= longitude_max)]

# Ausgabe der gefilterten Daten
```

```
[10]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21760 entries, 19 to 3444615
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
---  -
0   time            21760 non-null  int64
1   place          21760 non-null  object
2   status         21760 non-null  object
3   tsunami        21760 non-null  int64
4   significance   21760 non-null  int64
5   magnitudo      21760 non-null  float64
6   state          21760 non-null  object
7   longitude      21760 non-null  float64
8   latitude       21760 non-null  float64
9   depth          21760 non-null  float64
10  date           21760 non-null  datetime64[ns, UTC]
dtypes: datetime64[ns, UTC](1), float64(4), int64(3), object(3)
memory usage: 2.0+ MB
```

6 EDA (Explorative Datenanalyse): Data visualisations

6.1 Magnitude

```
[11]: edadata = data.copy()
```

```
[12]: # Datumsformatierung und Extraktion der Jahre
edadata['year'] = edadata['date'].dt.year

# Durchschnittliche Magnitude pro Jahr berechnen
avg_magnitude_per_year = edadata.groupby('year')['magnitudo'].mean().
↳reset_index()

# Einzelne Erdbeben als Scatter-Plot darstellen
plt.figure(figsize=(12, 6))
sns.scatterplot(data=edadata, x='year', y='magnitudo', color='blue', alpha=0.6,↳
↳label='Erdbeben Magnitude')

# Durchschnittliche Magnitude pro Jahr als Linie darstellen
sns.lineplot(data=avg_magnitude_per_year, x='year', y='magnitudo', color='red',↳
↳marker='o', label='Durchschnittliche Magnitude')

plt.xlabel("Jahr")
plt.ylabel("Magnitude")
plt.title("Magnitude der Erdbeben pro Jahr (1990-2023)")
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

output_20_0.png

```
[13]: # Wie viel Prozent der Erdbeben sind "stark" insgesamt
percentage_of_earthquakes_with_magnitude_ge_6 = (edadata['magnitudo'] >= 5.5).
↳mean() * 100
print(percentage_of_earthquakes_with_magnitude_ge_6, '%')

# Das wird später relevant
```

0.5744485294117647 %

```
[14]: # # Wie viel Prozent der Erdbeben sind "stark" pro Jahr
edadata['year'] = pd.to_datetime(edadata['time'], unit='ms').dt.year

# Filtere die Daten für den Zeitraum von 1990 bis 2023
data_filtered = edadata[(edadata['year'] >= 1990) & (edadata['year'] <= 2023)]
```

```

# Gruppieren nach Jahr und berechne den Prozentsatz der Erdbeben mit Magnitude >=
↳ 5.5
percentage_per_year = data_filtered[data_filtered['magnitudo'] >= 5.5].
↳ groupby('year').size() / data_filtered.groupby('year').size() * 100

# Fülle Jahre ohne Erdbeben mit Magnitude >= 5.5 mit 0% auf
percentage_per_year = percentage_per_year.reindex(range(1990, 2024),
↳ fill_value=0)

print(percentage_per_year)

```

```

year
1990    0.259740
1991    0.199800
1992    0.640000
1993         NaN
1994    0.038241
1995    0.311526
1996    1.137980
1997    0.581395
1998    0.087873
1999    0.553633
2000    0.763359
2001    0.326264
2002    1.208459
2003    1.324503
2004    0.405844
2005    0.696379
2006         NaN
2007    0.438596
2008    0.117786
2009    1.639344
2010    2.702703
2011    3.717472
2012    1.360544
2013    1.000000
2014         NaN
2015    0.675676
2016         NaN
2017    1.875000
2018         NaN
2019    2.479339
2020    3.906250
2021    2.673797
2022    2.380952
2023    2.773246
dtype: float64

```

6.2 Lokalisierung der Erdbebenorte

```
[15]: # Mittelpunkt der Karte definieren (ungefähr zwischen Türkei und Syrien)
mittel_lat, mittel_lon = 36.5, 38.0

# Eine Folium-Karte mit dem definierten Mittelpunkt erstellen
karte = folium.Map(location=[mittel_lat, mittel_lon], zoom_start=5)

# MarkerCluster für eine bessere Darstellung vieler Punkte
marker_cluster = MarkerCluster().add_to(karte)

# Erdbebenstandorte auf der Karte markieren
for idx, ereignis in edadata.iterrows():
    if ereignis['magnitudo'] > 5.5:
        # Erhöhe den Radius und ändere die Farbe für starke Erdbeben
        folium.CircleMarker(
            location=[ereignis['latitude'], ereignis['longitude']],
            radius=10, # Größerer Radius für stärkere Erdbeben
            popup=f"Magnitude: {ereignis['magnitudo']}\nOrt:␣
↪{ereignis['place']}\nDatum: {ereignis['date']}",
            color='red',
            fill=True,
            fill_color='red'
        ).add_to(karte)
    else:
        # Standard-Einstellungen für weniger starke Erdbeben
        folium.CircleMarker(
            location=[ereignis['latitude'], ereignis['longitude']],
            radius=5,
            popup=f"Magnitude: {ereignis['magnitudo']}\nOrt:␣
↪{ereignis['place']}\nDatum: {ereignis['date']}",
            color='blue',
            fill=True,
            fill_color='blue'
        ).add_to(marker_cluster)

# Karte speichern als HTML
karte.save('erdbeben_karte.html')
```

7 Feature Engineering

```
[16]: data_sorted = data.sort_values(by='date').reset_index(drop=True)
```

```
[17]: from scipy.stats import linregress
```

```

[18]: # Seismische Energieabgabe dE
def calculate_seismic_energy(magnitude):
    return 10 ** (11.8 + 1.5 * magnitude)

# Mittlere Magnitude M_mean (der letzten 5 Tage)
def calculate_mean_magnitude(data, current_date):
    filtered_data = data[(data['date'] >= current_date - pd.Timedelta(days=5)) &
↳(data['date'] < current_date)]
    if not filtered_data.empty:
        # Ersetze NaN-Werte durch 0
        mean_magnitude = filtered_data['magnitudo'].mean()
        return mean_magnitude if not np.isnan(mean_magnitude) else 0
    return 0

# Gutenberg-Richter a und b Werte zu aktuellem Erdbeben berechnen
def calculate_gutenberg_richter_parameters(data, current_date):
    filtered_data = data[data['date'] < current_date]
    if len(filtered_data) >= 10: # Sicherstellen, dass genügend Daten vorhanden
↳sind
        magnitudes = filtered_data['magnitudo']
        magnitude_counts = magnitudes.value_counts().sort_index(ascending=False)
        log_N = np.log10(magnitude_counts.cumsum()) # Kumulative Summe für N
        M = magnitude_counts.index.values
        slope, intercept, _, _, _ = linregress(M, log_N)
        return intercept, slope
    return np.nan, np.nan

# Zeit nach letzten größeren Erdbeben
def calculate_time_since_last_major(data, current_date, threshold=5.5):
    past_major_quakes = data[(data['date'] < current_date) & (data['magnitudo']
↳>= threshold)]
    if not past_major_quakes.empty:
        last_major_date = past_major_quakes['date'].iloc[-1]
        return (current_date - last_major_date).days
    return np.nan

# Magnitude-Defizit deltaM
def calculate_magnitude_deficit(data, current_date):
    a, b = calculate_gutenberg_richter_parameters(data, current_date)
    if not np.isnan(a) and not np.isnan(b) and b != 0:
        M_max_expected = a / b
        M_max_actual = data[data['date'] < current_date]['magnitudo'].max()
        return M_max_actual - M_max_expected
    return np.nan

# Liste für a und b Werte
a_values = []

```

```

b_values = []

# Berechnung der neuen Features für jede Zeile
for index, row in data_sorted.iterrows():
    a, b = calculate_gutenberg_richter_parameters(data_sorted, row['date'])
    a_values.append(a)
    b_values.append(b)

# Neue Features zu DataFrame hinzufügen
data_sorted['a_value'] = a_values
data_sorted['b_value'] = b_values
data_sorted['seismic_energy'] = data_sorted['magnitudo'].
    ↪apply(calculate_seismic_energy)
data_sorted['mean_magnitude_last_5_days'] = data_sorted['date'].apply(lambda x:
    ↪calculate_mean_magnitude(data_sorted, x))
data_sorted['magnitude_deficit'] = data_sorted['date'].apply(lambda x:
    ↪calculate_magnitude_deficit(data_sorted, x))
data_sorted['time_since_last_major'] = data_sorted['date'].apply(lambda x:
    ↪calculate_time_since_last_major(data_sorted, x))

# Imputation und Bereinigung
high_value_for_imputation = 365
data_sorted['time_since_last_major'].fillna(high_value_for_imputation,
    ↪inplace=True)
data_sorted = data_sorted.iloc[10:].reset_index(drop=True)

```

8 Zielvariablen definieren

```

[19]: def calculate_largest_aftershock_within_days(data, days_after=5):
    """
    Berechnet die Magnitude des größten Nachbebens innerhalb von `days_after`
    ↪Tagen
    nach jedem Erdbeben im Datensatz. Wenn kein Nachbeben auftritt, wird ein
    ↪Standardwert
    (z.B. 0) verwendet.
    """
    # Initialisiere die Zielvariable im DataFrame
    data['largest_aftershock_magnitude'] = 0.0

    # Gehe durch jeden Eintrag im DataFrame
    for index, row in data.iterrows():
        # Bestimme den Zeitraum für die Suche nach Nachbeben
        start_date = row['date']
        end_date = start_date + timedelta(days=days_after)

        # Filtere potenzielle Nachbeben, die nach dem aktuellen Beben auftreten

```

```

    aftershocks = data[(data['date'] > start_date) & (data['date'] <=
↳end_date)]

    if not aftershocks.empty:
        # Setze die Magnitude des größten Nachbebens in diesem Zeitraum
        largest_aftershock_magnitude = aftershocks['magnitudo'].max()
        data.at[index, 'largest_aftershock_magnitude'] =
↳largest_aftershock_magnitude
    else:
        # Verwende einen Standardwert, wenn kein Nachbeben auftritt
        data.at[index, 'largest_aftershock_magnitude'] = 0.0

# Anwenden der Funktion auf den Datensatz
calculate_largest_aftershock_within_days(data_sorted)

```

```

[20]: # Definiere den Schwellenwert für "starke" Erdbeben
strong_threshold = 5.5

# Erstelle eine neue binäre Zielvariable basierend auf der Magnitude des größten
↳Nachbebens
data_sorted['strong_aftershock_occurs'] =
↳(data_sorted['largest_aftershock_magnitude'] >= strong_threshold).astype(int)

```

```

[23]: # Zähle die Anzahl der Einsen in der Spalte 'strong_aftershock_occurs'
anzahl_einsen = data_sorted['strong_aftershock_occurs'].sum()

# Zähle die Gesamtanzahl der Einträge in der Spalte 'strong_aftershock_occurs'
gesamt_anzahl_einträge = len(data_sorted['strong_aftershock_occurs'])

# Gib das Verhältnis von Einsen zu allen Einträgen aus
verhältnis = anzahl_einsen / gesamt_anzahl_einträge

print("Anzahl der Einsen:", anzahl_einsen)
print("Gesamtanzahl der Einträge:", gesamt_anzahl_einträge)
print("Prozent starke Erdbeben:", verhältnis*100)

```

Anzahl der Einsen: 1634
Gesamtanzahl der Einträge: 21750
Prozent starke Erdbeben: 7.51264367816092

9 Modelle

Korrelationsanalyse Kombinieren der Features und der Zielvariablen für die Korrelationsanalyse
 $data_{for_correlation_occurs} = pd.concat([X, y_{occurs}.rename('occurs_{target}')], axis = 1)$

Berechnung der Korrelation $correlation_{matrix_occurs} = data_{for_correlation_occurs}.corr()$

Ausgabe der Korrelation der Features mit der Zielvariablen
'occurs_{target}'`print(correlationmatrixoccurs['occurstarget'].sortvalues(ascending = False))`

10 1. Regression

10.1 1.1 Magnitude

```
[26]: # Auswahl numerischer Features
numerical_features = ['significance', 'magnitudo', 'longitude', 'latitude',
    ↳ 'depth',
    'a_value', 'b_value', 'seismic_energy',
    ↳ 'mean_magnitude_last_5_days',
    'time_since_last_major']

X = data_sorted[numerical_features]
y_magnitude = data_sorted['largest_aftershock_magnitude']

# Zeitbasierte Aufteilung der Daten in Trainings- und Testsets
split_index = int(len(X) * 0.8)
X_train, X_test = X[:split_index], X[split_index:]
y_train_magnitude, y_test_magnitude = y_magnitude[:split_index],
    ↳ y_magnitude[split_index:]

# Erweiterung der Pipeline um einen Schritt für die Imputation
numerical_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')), # Fügt den Imputer hinzu, um
    ↳ fehlende Werte zu behandeln
    ('poly_features', PolynomialFeatures(degree=2, include_bias=False)),
    ('scaler', StandardScaler())
])

preprocessor = ColumnTransformer([
    ('num', numerical_pipeline, numerical_features)
])

ridge_reg_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('ridge_reg', Ridge())
])

param_grid = {
    'ridge_reg__alpha': [0.1, 1, 10, 100]
}

grid_search = GridSearchCV(ridge_reg_pipeline, param_grid, cv=5,
    ↳ scoring='neg_mean_squared_error', verbose=1)
```

```

grid_search.fit(X_train, y_train_magnitude)

best_model = grid_search.best_estimator_

predictions_magnitude_lr = best_model.predict(X_test)

mae_lr = mean_absolute_error(y_test_magnitude, predictions_magnitude_lr)
mse_lr = mean_squared_error(y_test_magnitude, predictions_magnitude_lr)

print("Ridge Regression - Magnitude Vorhersage:")
print("MAE:", mae_lr)
print("MSE:", mse_lr)

```

Fitting 5 folds for each of 4 candidates, totalling 20 fits
 Verbesserte Ridge Regression - Magnitude Vorhersage:
 MAE: 0.6866925621652359
 MSE: 1.5077400634288955

10.2 1.2 Vorkommen

```

[27]: # Auswahl der Features
features_columns = ['significance', 'magnitudo', 'longitude', 'latitude',
                    ↪ 'depth',
                    'a_value', 'b_value', 'seismic_energy',
                    ↪ 'mean_magnitude_last_5_days',
                    'time_since_last_major']
X = data_sorted[features_columns]
y_occurs = data_sorted['strong_aftershock_occurs']

# Zeitbasierte Aufteilung der Daten in Trainings- und Testsets
X_train_occurs, X_test_occurs, y_train_occurs, y_test_occurs =
↪ train_test_split(X, y_occurs, test_size=0.2, random_state=42)

# Definition der Pipeline
pipeline = ImbPipeline([
    ('scaler', StandardScaler()),
    ('smote', SMOTE(random_state=42)), # SMOTE innerhalb der Pipeline
    ('clf', LogisticRegression())
])

# Definition des Parameterraums für die Grid-Suche
param_grid = {
    'clf__C': [0.1, 1, 10, 100], # Regulierungsterm
    'clf__penalty': ['l2'] # Nur 'l2' Strafterm verwenden
}

```

```

# Durchführen der Grid-Suche
grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='accuracy',
    ↳ verbose=1)
grid_search.fit(X_train_occurs, y_train_occurs)

print("Beste Parameter:", grid_search.best_params_)

best_model = grid_search.best_estimator_

# Vorhersagen auf dem Testset
predictions_occurs_lr = best_model.predict(X_test_occurs)

# Berechnung der Metriken
accuracy_lr = accuracy_score(y_test_occurs, predictions_occurs_lr)
precision_lr = precision_score(y_test_occurs, predictions_occurs_lr)
recall_lr = recall_score(y_test_occurs, predictions_occurs_lr)
specificity_lr = recall_score(y_test_occurs, predictions_occurs_lr, pos_label=0)

# Ausgabe der Metriken
print("Verbesserte logistische Regression - Vorkommen Vorhersage:")
print(f"Accuracy: {accuracy_lr:.4f}")
print(f"Precision: {precision_lr:.4f}")
print(f"Recall (Sensitivität): {recall_lr:.4f}")

```

Fitting 5 folds for each of 4 candidates, totalling 20 fits
Beste Parameter: {'clf__C': 0.1, 'clf__penalty': 'l2'}
Verbesserte logistische Regression - Vorkommen Vorhersage:
Accuracy: 0.6510
Precision: 0.1377
Recall (Sensitivität): 0.7006

11 2. Random Forrest

11.1 2.1 Magnitude

```

[28]: # Auswahl numerischer Features
numerical_features = ['significance', 'magnitudo', 'longitude', 'latitude',
    ↳ 'depth',
    ↳ 'a_value', 'b_value', 'seismic_energy',
    ↳ 'mean_magnitude_last_5_days',
    ↳ 'time_since_last_major']

X = data_sorted[numerical_features]
y_magnitude = data_sorted['largest_aftershock_magnitude']

# Daten in Trainings- und Testsets zeitbasiert aufteilen
split_index = int(len(X) * 0.8)

```

```

X_train = X[:split_index]
X_test = X[split_index:]
y_train_magnitude = y_magnitude[:split_index]
y_test_magnitude = y_magnitude[split_index:]

# Erstellen einer Pipeline für die Skalierung, Feature-Auswahl und das Modell
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('feature_selection',
     ↪SelectFromModel(RandomForestRegressor(random_state=42))),
    ('rf', RandomForestRegressor(random_state=42))
])

# Definition des Parameterraums für das GridSearchCV
param_grid = {
    'rf__n_estimators': [100, 200, 300],
    'rf__max_depth': [None, 10, 20, 30],
    'rf__min_samples_split': [2, 5, 10],
    'rf__min_samples_leaf': [1, 2, 4],
}

# Durchführung des GridSearchCV zur Optimierung der Hyperparameter
grid_search = GridSearchCV(pipeline, param_grid, cv=5,
     ↪scoring='neg_mean_squared_error', verbose=1, n_jobs=-1)
grid_search.fit(X_train, y_train_magnitude)

# Bester Regressor und dessen Hyperparameter
best_regressor = grid_search.best_estimator_
print("Beste Hyperparameter:", grid_search.best_params_)

# Vorhersagen auf dem Testset machen
predictions_magnitude_rf = best_regressor.predict(X_test)

# Evaluiere die Ergebnisse
mae_rf = mean_absolute_error(y_test_magnitude, predictions_magnitude_rf)
mse_rf = mean_squared_error(y_test_magnitude, predictions_magnitude_rf)

# Ausgabe der Ergebnisse
print("Random Forest - Magnitude Vorhersage:")
print("MAE:", mae_rf)
print("MSE:", mse_rf)

```

Fitting 5 folds for each of 108 candidates, totalling 540 fits
Beste Hyperparameter: {'rf__max_depth': 10, 'rf__min_samples_leaf': 1, 'rf__min_samples_split': 2, 'rf__n_estimators': 100}
Verbesserter Random Forest - Magnitude Vorhersage:
MAE: 0.8181211188889578

MSE: 1.6796032725334626

11.2 2.1 Vorkommen

```
[29]: # Auswahl der Features und Zielvariable
features_columns = ['significance', 'magnitudo', 'longitudo', 'latitude',
↳ 'depth',
                    'a_value', 'b_value', 'seismic_energy',
↳ 'mean_magnitude_last_5_days',
                    'time_since_last_major']
X = data_sorted[features_columns]
y_occurs = data_sorted['strong_aftershock_occurs']

# Daten in Trainings- und Testsets zeitbasiert aufteilen
split_index = int(len(X) * 0.8)
X_train_occurs, X_test_occurs = X.iloc[:split_index], X.iloc[split_index:]
y_train_occurs, y_test_occurs = y_occurs.iloc[:split_index], y_occurs.
↳ iloc[split_index:]

# Angepasste Scoring-Funktion
f1_scorer = make_scorer(f1_score)

# Pipeline mit Feature-Selektion
pipeline = imblearnPipeline([
    ('scaler', StandardScaler()),
    ('smote', SMOTE(random_state=42)),
    ('feature_selection',
↳ SelectFromModel(RandomForestClassifier(n_estimators=100, random_state=42))),
    ('classifier', RandomForestClassifier(random_state=42))
])

# Parameter-Raum für GridSearchCV
param_grid = {
    'classifier__n_estimators': [100, 300, 500],
    'classifier__max_depth': [None, 10, 20],
    'classifier__min_samples_split': [2, 5],
    'classifier__min_samples_leaf': [1, 2, 4],
}

# GridSearchCV
grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring=f1_scorer,
↳ n_jobs=-1, verbose=1)
grid_search.fit(X_train_occurs, y_train_occurs)

# Beste Parameter und Modell
best_model = grid_search.best_estimator_
```

```

# Wahrscheinlichkeitsbasierte Vorhersagen
predictions_proba = best_model.predict_proba(X_test_occurs)[: , 1]

# Präzision, Recall, Schwellenwerte
precision, recall, thresholds = precision_recall_curve(y_test_occurs,
↳ predictions_proba)

# F1-Scores
f1_scores = 2 * (precision * recall) / (precision + recall)
f1_scores = f1_scores[~np.isnan(f1_scores)] # Entfernen von NaN Werten aus
↳ F1-Scores
optimal_idx = np.argmax(f1_scores)
optimal_threshold = thresholds[optimal_idx]

# Anwendung des optimalen Schwellenwerts
predictions_occurs_rf = (predictions_proba >= optimal_threshold).astype(int)

# Evaluation
accuracy_rf = accuracy_score(y_test_occurs, predictions_occurs_rf)
precision_rf = precision_score(y_test_occurs, predictions_occurs_rf)
recall_rf = recall_score(y_test_occurs, predictions_occurs_rf)

# Ausgabe der Ergebnisse
print(f"Optimaler Schwellenwert: {optimal_threshold}")
print("Random Forest - Vorkommen Vorhersage:")
print(f"Accuracy: {accuracy_rf:.4f}")
print(f"Precision: {precision_rf:.4f}")
print(f"Recall: {recall_rf:.4f}")

```

Fitting 5 folds for each of 54 candidates, totalling 270 fits
Optimaler Schwellenwert: 0.05778174603174604
Verbesserter Random Forest - Vorkommen Vorhersage:
Accuracy: 0.5683
Precision: 0.2158
Recall: 0.7628

12 3. Neuronales Netz

12.1 3.1 Magnitude

```

[30]: #Auswahl der Features und Zielvariable
features_columns = ['significance', 'magnitudo', 'longitude', 'latitude',
↳ 'depth',

```

```

        'a_value', 'b_value', 'seismic_energy',
        ↪ 'mean_magnitude_last_5_days',
        'time_since_last_major']
X = data_sorted[features_columns]
y_magnitude = data_sorted['largest_aftershock_magnitude']

# Datenstandardisierung
scaler = StandardScaler()

# Daten in Trainings- und Testsets zeitbasiert aufteilen
split_index = int(len(X) * 0.8)
X_train_scaled = scaler.fit_transform(X[:split_index])
X_test_scaled = scaler.fit_transform(X[split_index:])
y_train_magnitude = y_magnitude[:split_index]
y_test_magnitude = y_magnitude[split_index:]

# Modell definieren
model_magnitude = Sequential([
    Dense(128, activation='relu', input_shape=(X_train_scaled.shape[1],)),
    Dropout(0.2),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(1)
])

model_magnitude.compile(optimizer=Adam(learning_rate=0.001),
    ↪ loss='mean_squared_error')

# Early Stopping Regel, um Overfitting zu vermeiden
early_stopping_monitor = EarlyStopping(
    monitor='val_loss',
    patience=10,
    restore_best_weights=True,
    verbose=1
)

# Modelltraining mit Validierungsdaten
history = model_magnitude.fit(
    X_train_scaled,
    y_train_magnitude,
    validation_split=0.1,
    epochs=100,
    batch_size=32,
    callbacks=[early_stopping_monitor],
    verbose=1
)

```

```

# Vorhersagen
predictions_magnitude_ann = model_magnitude.predict(X_test_scaled)

# Evaluation
mae_ann = mean_absolute_error(y_test_magnitude, predictions_magnitude_ann)
mse_ann = mean_squared_error(y_test_magnitude, predictions_magnitude_ann)

# Ausgabe der Ergebnisse
print("ANN - Magnitude Vorhersage:")
print("MAE:", mae_ann)
print("MSE:", mse_ann)

```

```

Epoch 1/100
490/490 [=====] - 2s 2ms/step - loss: 1.2871 -
val_loss: 0.4345
Epoch 2/100
490/490 [=====] - 1s 2ms/step - loss: 0.5565 -
val_loss: 0.3340
Epoch 3/100
490/490 [=====] - 1s 2ms/step - loss: 0.4897 -
val_loss: 0.3322
Epoch 4/100
490/490 [=====] - 1s 2ms/step - loss: 0.4661 -
val_loss: 0.3213
Epoch 5/100
490/490 [=====] - 1s 2ms/step - loss: 0.4342 -
val_loss: 0.3277
Epoch 6/100
490/490 [=====] - 1s 2ms/step - loss: 0.4395 -
val_loss: 0.3117
Epoch 7/100
490/490 [=====] - 1s 2ms/step - loss: 0.4714 -
val_loss: 0.3166
Epoch 8/100
490/490 [=====] - 1s 2ms/step - loss: 0.4154 -
val_loss: 0.3201
Epoch 9/100
490/490 [=====] - 1s 2ms/step - loss: 0.4099 -
val_loss: 0.3188
Epoch 10/100
490/490 [=====] - 1s 2ms/step - loss: 0.4001 -
val_loss: 0.3132
Epoch 11/100
490/490 [=====] - 1s 2ms/step - loss: 0.4147 -
val_loss: 0.3092
Epoch 12/100
490/490 [=====] - 1s 2ms/step - loss: 0.3966 -

```

```

val_loss: 0.3162
Epoch 13/100
490/490 [=====] - 1s 2ms/step - loss: 0.3904 -
val_loss: 0.3400
Epoch 14/100
490/490 [=====] - 1s 2ms/step - loss: 0.3901 -
val_loss: 0.3284
Epoch 15/100
490/490 [=====] - 1s 2ms/step - loss: 0.4013 -
val_loss: 0.3076
Epoch 16/100
490/490 [=====] - 1s 2ms/step - loss: 0.4186 -
val_loss: 0.3173
Epoch 17/100
490/490 [=====] - 1s 2ms/step - loss: 0.3889 -
val_loss: 0.3761
Epoch 18/100
490/490 [=====] - 1s 2ms/step - loss: 0.3845 -
val_loss: 0.3424
Epoch 19/100
490/490 [=====] - 1s 2ms/step - loss: 0.4091 -
val_loss: 0.3266
Epoch 20/100
490/490 [=====] - 1s 2ms/step - loss: 0.4144 -
val_loss: 0.3268
Epoch 21/100
490/490 [=====] - 1s 2ms/step - loss: 0.3752 -
val_loss: 0.3185
Epoch 22/100
490/490 [=====] - 1s 2ms/step - loss: 0.4329 -
val_loss: 0.3266
Epoch 23/100
490/490 [=====] - 1s 2ms/step - loss: 0.3691 -
val_loss: 0.3272
Epoch 24/100
490/490 [=====] - 1s 2ms/step - loss: 0.3679 -
val_loss: 0.3194
Epoch 25/100
466/490 [=====>..] - ETA: 0s - loss: 0.3575Restoring model
weights from the end of the best epoch: 15.
490/490 [=====] - 1s 2ms/step - loss: 0.3658 -
val_loss: 0.3132
Epoch 25: early stopping
136/136 [=====] - 1s 1ms/step
ANN - Magnitude Vorhersage:
MAE: 0.72654263521611
MSE: 1.4639265024234895

```

12.2 3.2 Vorkommen

```
[31]: # Auswahl der Features und Zielvariable
features_columns = ['significance', 'magnitudo', 'longitude', 'latitude',
↳ 'depth',
                    'a_value', 'b_value', 'seismic_energy',
↳ 'mean_magnitude_last_5_days',
                    'time_since_last_major']
X = data_sorted[features_columns]
y_occurs = data_sorted['strong_aftershock_occurs']

# Daten in Trainings- und Testsets zeitbasiert aufteilen
split_index = int(len(X) * 0.8)
X_train, X_test = X.iloc[:split_index], X.iloc[split_index:]
y_train_occurs, y_test_occurs = y_occurs.iloc[:split_index], y_occurs.
↳ iloc[split_index:]

# Datenstandardisierung
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# SMOTE anwenden auf Trainingsdaten
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train_scaled,
↳ y_train_occurs)

# Modell definieren
model = Sequential([
    Dense(128, activation='relu', input_dim=X_train_resampled.shape[1]),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])

# Modell kompilieren
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy',
↳ metrics=[Precision()]) # Verwende Precision()

# Callbacks definieren
early_stopping = EarlyStopping(monitor='val_loss', patience=10)

# Modell trainieren
history = model.fit(
    X_train_resampled,
    y_train_resampled,
```

```

    epochs=100,
    batch_size=32,
    validation_split=0.1,
    callbacks=[early_stopping],
    verbose=1
)

# Vorhersage-Wahrscheinlichkeiten generieren
predictions_proba = model.predict(X_test_scaled)

# Berechnung von Präzision, Recall und Schwellenwerten
precision, recall, thresholds = precision_recall_curve(y_test_occurs,
↳predictions_proba.ravel())

# Berechnung der F1-Scores für verschiedene Schwellenwerte
f1_scores = 2 * (precision * recall) / (precision + recall)
f1_scores = f1_scores[~np.isnan(f1_scores)] # Entfernen von NaN Werten aus
↳F1-Scores
optimal_idx = np.argmax(f1_scores)
optimal_threshold = thresholds[optimal_idx]

# Anwendung des optimalen Schwellenwerts für finale Vorhersagen
predictions_occurs_ann = (predictions_proba > optimal_threshold).astype(int)

# Evaluierung
accuracy_ann = accuracy_score(y_test_occurs, predictions_occurs_ann)
precision_ann = precision_score(y_test_occurs, predictions_occurs_ann)
recall_ann = recall_score(y_test_occurs, predictions_occurs_ann)

# Ausgabe der Ergebnisse
print(f"Optimaler Schwellenwert: {optimal_threshold}")
print("Verbesserte ANN Vorhersage:")
print(f"Accuracy: {accuracy_ann:.4f}")
print(f"Precision: {precision_ann:.4f}")
print(f"Recall: {recall_ann:.4f}")

```

```

Epoch 1/100
922/922 [=====] - 3s 2ms/step - loss: 0.6176 -
precision: 0.6373 - val_loss: 0.6208 - val_precision: 1.0000
Epoch 2/100
922/922 [=====] - 2s 2ms/step - loss: 0.5627 -
precision: 0.6887 - val_loss: 0.5982 - val_precision: 1.0000
Epoch 3/100
922/922 [=====] - 2s 2ms/step - loss: 0.5322 -
precision: 0.7046 - val_loss: 0.5122 - val_precision: 1.0000
Epoch 4/100
922/922 [=====] - 2s 2ms/step - loss: 0.5016 -

```

precision: 0.7173 - val_loss: 0.4847 - val_precision: 1.0000
Epoch 5/100
922/922 [=====] - 2s 2ms/step - loss: 0.4709 -
precision: 0.7268 - val_loss: 0.4182 - val_precision: 1.0000
Epoch 6/100
922/922 [=====] - 2s 2ms/step - loss: 0.4530 -
precision: 0.7333 - val_loss: 0.4036 - val_precision: 1.0000
Epoch 7/100
922/922 [=====] - 2s 2ms/step - loss: 0.4424 -
precision: 0.7321 - val_loss: 0.3999 - val_precision: 1.0000
Epoch 8/100
922/922 [=====] - 2s 2ms/step - loss: 0.4279 -
precision: 0.7402 - val_loss: 0.3868 - val_precision: 1.0000
Epoch 9/100
922/922 [=====] - 2s 2ms/step - loss: 0.4177 -
precision: 0.7345 - val_loss: 0.4024 - val_precision: 1.0000
Epoch 10/100
922/922 [=====] - 2s 2ms/step - loss: 0.4078 -
precision: 0.7317 - val_loss: 0.3867 - val_precision: 1.0000
Epoch 11/100
922/922 [=====] - 2s 2ms/step - loss: 0.4006 -
precision: 0.7372 - val_loss: 0.3451 - val_precision: 1.0000
Epoch 12/100
922/922 [=====] - 2s 2ms/step - loss: 0.4014 -
precision: 0.7370 - val_loss: 0.3647 - val_precision: 1.0000
Epoch 13/100
922/922 [=====] - 2s 2ms/step - loss: 0.3910 -
precision: 0.7376 - val_loss: 0.3386 - val_precision: 1.0000
Epoch 14/100
922/922 [=====] - 2s 2ms/step - loss: 0.3853 -
precision: 0.7364 - val_loss: 0.3314 - val_precision: 1.0000
Epoch 15/100
922/922 [=====] - 2s 2ms/step - loss: 0.3799 -
precision: 0.7427 - val_loss: 0.3144 - val_precision: 1.0000
Epoch 16/100
922/922 [=====] - 2s 2ms/step - loss: 0.3772 -
precision: 0.7405 - val_loss: 0.3245 - val_precision: 1.0000
Epoch 17/100
922/922 [=====] - 2s 2ms/step - loss: 0.3688 -
precision: 0.7565 - val_loss: 0.3089 - val_precision: 1.0000
Epoch 18/100
922/922 [=====] - 2s 2ms/step - loss: 0.3674 -
precision: 0.7618 - val_loss: 0.3104 - val_precision: 1.0000
Epoch 19/100
922/922 [=====] - 2s 2ms/step - loss: 0.3648 -
precision: 0.7736 - val_loss: 0.3005 - val_precision: 1.0000
Epoch 20/100
922/922 [=====] - 2s 2ms/step - loss: 0.3643 -

precision: 0.7754 - val_loss: 0.2935 - val_precision: 1.0000
Epoch 21/100
922/922 [=====] - 2s 2ms/step - loss: 0.3599 -
precision: 0.7790 - val_loss: 0.3053 - val_precision: 1.0000
Epoch 22/100
922/922 [=====] - 2s 2ms/step - loss: 0.3571 -
precision: 0.7814 - val_loss: 0.3042 - val_precision: 1.0000
Epoch 23/100
922/922 [=====] - 2s 2ms/step - loss: 0.3537 -
precision: 0.7832 - val_loss: 0.2960 - val_precision: 1.0000
Epoch 24/100
922/922 [=====] - 2s 2ms/step - loss: 0.3507 -
precision: 0.7834 - val_loss: 0.2850 - val_precision: 1.0000
Epoch 25/100
922/922 [=====] - 2s 2ms/step - loss: 0.3437 -
precision: 0.7879 - val_loss: 0.3179 - val_precision: 1.0000
Epoch 26/100
922/922 [=====] - 2s 2ms/step - loss: 0.3522 -
precision: 0.7897 - val_loss: 0.3040 - val_precision: 1.0000
Epoch 27/100
922/922 [=====] - 2s 2ms/step - loss: 0.3467 -
precision: 0.7884 - val_loss: 0.2598 - val_precision: 1.0000
Epoch 28/100
922/922 [=====] - 2s 2ms/step - loss: 0.3484 -
precision: 0.7890 - val_loss: 0.2994 - val_precision: 1.0000
Epoch 29/100
922/922 [=====] - 2s 2ms/step - loss: 0.3416 -
precision: 0.7938 - val_loss: 0.2833 - val_precision: 1.0000
Epoch 30/100
922/922 [=====] - 2s 2ms/step - loss: 0.3446 -
precision: 0.7959 - val_loss: 0.3041 - val_precision: 1.0000
Epoch 31/100
922/922 [=====] - 2s 2ms/step - loss: 0.3442 -
precision: 0.7965 - val_loss: 0.2991 - val_precision: 1.0000
Epoch 32/100
922/922 [=====] - 2s 2ms/step - loss: 0.3384 -
precision: 0.7946 - val_loss: 0.2885 - val_precision: 1.0000
Epoch 33/100
922/922 [=====] - 2s 2ms/step - loss: 0.3375 -
precision: 0.7982 - val_loss: 0.3158 - val_precision: 1.0000
Epoch 34/100
922/922 [=====] - 2s 2ms/step - loss: 0.3320 -
precision: 0.8009 - val_loss: 0.2929 - val_precision: 1.0000
Epoch 35/100
922/922 [=====] - 2s 2ms/step - loss: 0.3400 -
precision: 0.7962 - val_loss: 0.2850 - val_precision: 1.0000
Epoch 36/100
922/922 [=====] - 2s 2ms/step - loss: 0.3368 -

```

precision: 0.7973 - val_loss: 0.2815 - val_precision: 1.0000
Epoch 37/100
922/922 [=====] - 2s 2ms/step - loss: 0.3323
- precision: 0.8003 - val_loss: 0.2718 - val_precision: 1.0000
136/136 [=====] - 0s 1ms/step
Optimaler Schwellenwert: 0.5052958726882935
Verbesserte ANN Vorhersage:
  Accuracy: 0.8200
  Precision: 0.4204
  Recall: 0.6731

```

13 4. LSTM

13.1 4.1 Magnitude

```

[32]: # Funktion zum Erzeugen der Sequenzen
def create_sequences(X, y, sequence_length=10):
    X_seq, y_seq = [], []
    X = np.array(X)
    y = np.array(y)
    for i in range(len(X) - sequence_length):
        X_seq.append(X[i:(i+sequence_length)])
        y_seq.append(y[i + sequence_length])
    return np.array(X_seq), np.array(y_seq)

# Auswahl der Features und Zielvariable
features_columns = ['significance', 'magnitudo', 'longitude', 'latitude',
                    'depth',
                    'a_value', 'b_value', 'seismic_energy',
                    'mean_magnitude_last_5_days',
                    'time_since_last_major']
X = data_sorted[features_columns]
y_magnitude = data_sorted['largest_aftershock_magnitude']

# Datenstandardisierung
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Sequenzen erstellen
X_seq, y_seq = create_sequences(X_scaled, y_magnitude, sequence_length=10)

# Aufteilen in Trainings- und Testdaten
split_index = int(len(X_seq) * 0.8)

```

```

X_train_magnitude_lstm, X_test_magnitude_lstm = X_seq[:split_index],
↳X_seq[split_index:]
y_train_magnitude_lstm, y_test_magnitude_lstm = y_seq[:split_index],
↳y_seq[split_index:]

# Modell definieren
model = Sequential([
    LSTM(128, activation='tanh', input_shape=(X_train_magnitude_lstm.shape[1],
↳X_train_magnitude_lstm.shape[2])),
    Dropout(0.2),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(1)
])

# Modell kompilieren
model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')

# Callbacks definieren
early_stopping = EarlyStopping(monitor='val_loss', patience=20,
↳restore_best_weights=True)
reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.2, patience=10,
↳min_lr=0.00001)

# Modell trainieren
history = model.fit(
    X_train_magnitude_lstm,
    y_train_magnitude_lstm,
    validation_split=0.1,
    epochs=100,
    batch_size=32,
    callbacks=[early_stopping, reduce_lr],
    verbose=1
)

# Vorhersagen auf dem Testset
predictions_magnitude_lstm = model.predict(X_test_magnitude_lstm)
mae_lstm = mean_absolute_error(y_test_magnitude_lstm, predictions_magnitude_lstm)
mse_lstm = mean_squared_error(y_test_magnitude_lstm, predictions_magnitude_lstm)

print("LSTM - Magnitude Vorhersage:")
print("MAE:", mae_lstm)
print("MSE:", mse_lstm)

```

```

Epoch 1/100
490/490 [=====] - 7s 11ms/step - loss: 1.2110 -
val_loss: 0.3429 - lr: 0.0010

```

Epoch 2/100
490/490 [=====] - 5s 10ms/step - loss: 0.5050 -
val_loss: 0.3279 - lr: 0.0010
Epoch 3/100
490/490 [=====] - 5s 10ms/step - loss: 0.4743 -
val_loss: 0.3336 - lr: 0.0010
Epoch 4/100
490/490 [=====] - 5s 10ms/step - loss: 0.4594 -
val_loss: 0.3070 - lr: 0.0010
Epoch 5/100
490/490 [=====] - 5s 10ms/step - loss: 0.4482 -
val_loss: 0.3025 - lr: 0.0010
Epoch 6/100
490/490 [=====] - 5s 10ms/step - loss: 0.4298 -
val_loss: 0.3036 - lr: 0.0010
Epoch 7/100
490/490 [=====] - 5s 10ms/step - loss: 0.4233 -
val_loss: 0.3616 - lr: 0.0010
Epoch 8/100
490/490 [=====] - 5s 10ms/step - loss: 0.4127 -
val_loss: 0.3521 - lr: 0.0010
Epoch 9/100
490/490 [=====] - 5s 10ms/step - loss: 0.4106 -
val_loss: 0.3463 - lr: 0.0010
Epoch 10/100
490/490 [=====] - 5s 10ms/step - loss: 0.3979 -
val_loss: 0.3461 - lr: 0.0010
Epoch 11/100
490/490 [=====] - 5s 10ms/step - loss: 0.3935 -
val_loss: 0.3310 - lr: 0.0010
Epoch 12/100
490/490 [=====] - 5s 10ms/step - loss: 0.3811 -
val_loss: 0.3567 - lr: 0.0010
Epoch 13/100
490/490 [=====] - 5s 10ms/step - loss: 0.3817 -
val_loss: 0.3571 - lr: 0.0010
Epoch 14/100
490/490 [=====] - 5s 10ms/step - loss: 0.3703 -
val_loss: 0.3397 - lr: 0.0010
Epoch 15/100
490/490 [=====] - 5s 10ms/step - loss: 0.3608 -
val_loss: 0.4814 - lr: 0.0010
Epoch 16/100
490/490 [=====] - 5s 10ms/step - loss: 0.3460 -
val_loss: 0.4506 - lr: 2.0000e-04
Epoch 17/100
490/490 [=====] - 5s 10ms/step - loss: 0.3431 -
val_loss: 0.4768 - lr: 2.0000e-04

```

Epoch 18/100
490/490 [=====] - 5s 10ms/step - loss: 0.3392 -
val_loss: 0.5020 - lr: 2.0000e-04
Epoch 19/100
490/490 [=====] - 5s 10ms/step - loss: 0.3350 -
val_loss: 0.5796 - lr: 2.0000e-04
Epoch 20/100
490/490 [=====] - 5s 10ms/step - loss: 0.3339 -
val_loss: 0.4998 - lr: 2.0000e-04
Epoch 21/100
490/490 [=====] - 5s 10ms/step - loss: 0.3309 -
val_loss: 0.5092 - lr: 2.0000e-04
Epoch 22/100
490/490 [=====] - 5s 10ms/step - loss: 0.3314 -
val_loss: 0.5757 - lr: 2.0000e-04
Epoch 23/100
490/490 [=====] - 5s 10ms/step - loss: 0.3262 -
val_loss: 0.5873 - lr: 2.0000e-04
Epoch 24/100
490/490 [=====] - 5s 10ms/step - loss: 0.3242 -
val_loss: 0.4610 - lr: 2.0000e-04
Epoch 25/100
490/490 [=====] - 5s 10ms/step - loss: 0.3272 -
val_loss: 0.5012 - lr: 2.0000e-04
136/136 [=====] - 1s 5ms/step
LSTM - Magnitude Vorhersage:
MAE: 0.6678738734320752
MSE: 1.2887081619612262

```

13.2 4.2 Vorkommen

```

[33]: # Auswahl der Features und Zielvariable
features_columns = ['significance', 'magnitudo', 'longitude', 'latitude',
↳ 'depth',
                    'a_value', 'b_value', 'seismic_energy',
↳ 'mean_magnitude_last_5_days',
                    'time_since_last_major']
X = data_sorted[features_columns].values
y = data_sorted['strong_aftershock_occurs'].values

# Aufteilen in Trainings- und Testdaten (Zeitbasiert, angenommen Daten sind
↳ zeitlich sortiert)
split_index = int(len(X) * 0.8)
X_train, X_test = X[:split_index], X[split_index:]
y_train, y_test = y[:split_index], y[split_index:]

# Datenstandardisierung

```

```

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# SMOTE Anwendung nur auf das Trainingsset
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train_scaled,
↳y_train)

# Umwandeln der Daten in Sequenzen
def create_sequences(X, y, sequence_length=10):
    X_seq, y_seq = [], []
    for i in range(len(X) - sequence_length + 1):
        X_seq.append(X[i:i + sequence_length])
        y_seq.append(y[i + sequence_length - 1])
    return np.array(X_seq), np.array(y_seq)

sequence_length = 10
X_train_seq, y_train_seq = create_sequences(X_train_resampled,
↳y_train_resampled, sequence_length)
X_test_seq, y_test_seq = create_sequences(X_test_scaled, y_test, sequence_length)

# Modell definieren
model_lstm = Sequential([
    LSTM(128, return_sequences=True, input_shape=(sequence_length, X_train_seq.
↳shape[2])),
    Dropout(0.5),
    LSTM(64),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])

# Modell kompilieren
model_lstm.compile(optimizer=Adam(learning_rate=0.001),
↳loss='binary_crossentropy', metrics=['Precision'])

# Callbacks definieren
callbacks_lstm = [
    EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=0.0001)
]

# Modell trainieren
history_lstm = model_lstm.fit(X_train_seq, y_train_seq, epochs=100,
↳batch_size=64, validation_split=0.1, callbacks=callbacks_lstm, verbose=1)

# Vorhersagen auf dem Testset

```

```

predictions_proba_lstm = model_lstm.predict(X_test_seq)

# Berechnung von Präzision, Recall und Schwellenwerten
precision_lstm, recall_lstm, thresholds_lstm =
↳precision_recall_curve(y_test_seq, predictions_proba_lstm.ravel())

# Optimierung: Anpassen der Methode zur Schwellenwertfindung
# Finden des Schwellenwerts nahe einem ausgewogenen Verhältnis zwischen
↳Precision und Recall
f1_scores_lstm = 2 * (precision_lstm * recall_lstm) / (precision_lstm +
↳recall_lstm)
f1_scores_lstm = f1_scores_lstm[~np.isnan(f1_scores_lstm)] # NaN Werte entfernen
optimal_idx_lstm = np.argmax(f1_scores_lstm)
optimal_threshold_lstm = thresholds_lstm[optimal_idx_lstm]

# Anwendung des optimalen Schwellenwerts für finale Vorhersagen
predictions_occurs_lstm = (predictions_proba_lstm > optimal_threshold_lstm).
↳astype(int)

# Evaluierung
accuracy_lstm = accuracy_score(y_test_seq, predictions_occurs_lstm)
precision_lstm = precision_score(y_test_seq, predictions_occurs_lstm)
recall_lstm = recall_score(y_test_seq, predictions_occurs_lstm)

# Ausgabe der Ergebnisse
print(f"Optimaler Schwellenwert (LSTM): {optimal_threshold_lstm}")
print("Verbesserte LSTM Vorhersage:")
print(f"Accuracy: {accuracy_lstm:.4f}")
print(f"Precision: {precision_lstm:.4f}")
print(f"Recall: {recall_lstm:.4f}")

```

Epoch 1/100

461/461 [=====] - 15s 24ms/step - loss: 0.2084 -
precision: 0.9199 - val_loss: 0.0139 - val_precision: 1.0000 - lr: 0.0010

Epoch 2/100

461/461 [=====] - 10s 22ms/step - loss: 0.1214 -
precision: 0.9840 - val_loss: 0.0106 - val_precision: 1.0000 - lr: 0.0010

Epoch 3/100

461/461 [=====] - 10s 22ms/step - loss: 0.1094 -
precision: 0.9885 - val_loss: 0.0110 - val_precision: 1.0000 - lr: 0.0010

Epoch 4/100

461/461 [=====] - 10s 22ms/step - loss: 0.1003 -
precision: 0.9871 - val_loss: 0.0064 - val_precision: 1.0000 - lr: 0.0010

Epoch 5/100

461/461 [=====] - 10s 22ms/step - loss: 0.0889 -
precision: 0.9872 - val_loss: 0.0077 - val_precision: 1.0000 - lr: 0.0010

Epoch 6/100

```
461/461 [=====] - 10s 22ms/step - loss: 0.0823
- precision: 0.9867 - val_loss: 0.0031 - val_precision: 1.0000 - lr: 0.0010
Epoch 7/100
461/461 [=====] - 10s 22ms/step - loss: 0.0761
- precision: 0.9866 - val_loss: 0.0045 - val_precision: 1.0000 - lr: 0.0010
Epoch 8/100
461/461 [=====] - 10s 23ms/step - loss: 0.0692
- precision: 0.9841 - val_loss: 0.0067 - val_precision: 1.0000 - lr: 0.0010
Epoch 9/100
461/461 [=====] - 10s 22ms/step - loss: 0.0638
- precision: 0.9840 - val_loss: 0.0070 - val_precision: 1.0000 - lr: 0.0010
Epoch 10/100
461/461 [=====] - 10s 23ms/step - loss: 0.0599
- precision: 0.9867 - val_loss: 0.0059 - val_precision: 1.0000 - lr: 0.0010
Epoch 11/100
461/461 [=====] - 11s 23ms/step - loss: 0.0542
- precision: 0.9877 - val_loss: 0.0063 - val_precision: 1.0000 - lr: 0.0010
Epoch 12/100
461/461 [=====] - 10s 23ms/step - loss: 0.0377
- precision: 0.9926 - val_loss: 0.0041 - val_precision: 1.0000 - lr:
2.0000e-04 Epoch 13/100
461/461 [=====] - 10s 23ms/step - loss: 0.0335
- precision: 0.9924 - val_loss: 0.0054 - val_precision: 1.0000 - lr:
2.0000e-04 Epoch 14/100
461/461 [=====] - 10s 22ms/step - loss: 0.0300
- precision: 0.9932 - val_loss: 0.0045 - val_precision: 1.0000 - lr:
2.0000e-04 Epoch 15/100
461/461 [=====] - 10s 23ms/step - loss: 0.0292
- precision: 0.9915 - val_loss: 0.0040 - val_precision: 1.0000 - lr:
2.0000e-04 Epoch 16/100
461/461 [=====] - 10s 23ms/step - loss: 0.0270
- precision: 0.9922 - val_loss: 0.0073 - val_precision: 1.0000 - lr:
2.0000e-04 136/136 [=====] - 2s 6ms/step
Optimaler Schwellenwert (LSTM): 0.012300126254558563
Verbesserte LSTM Vorhersage:
  Accuracy: 0.3882
Precision: 0.1758
Recall: 0.8830
```

14 5. Ensemble

14.1 5.1 Magnitude

```
[39]: # Bestimmen der minimalen Länge da beim LSTM die ersten 9 Datenpunkte nicht
      ↳ beachtet werden
min_length = min(len(predictions_magnitude_lr), len(predictions_magnitude_rf),
                 len(predictions_magnitude_ann), len(predictions_magnitude_lstm))

# Schneiden der Vorhersagen auf die minimale Länge
predictions_magnitude_lr_adjusted = predictions_magnitude_lr[:min_length]
predictions_magnitude_rf_adjusted = predictions_magnitude_rf[:min_length]
predictions_magnitude_ann_adjusted = predictions_magnitude_ann[:min_length]
predictions_magnitude_lstm_adjusted = predictions_magnitude_lstm[:min_length]

# Kombinieren Sie die angepassten Vorhersagen in einem Feature-Array
X_combined_adjusted = np.column_stack((
    predictions_magnitude_lr_adjusted,
    predictions_magnitude_rf_adjusted,
    predictions_magnitude_ann_adjusted,
    predictions_magnitude_lstm_adjusted
))

# Die wahren Werte entsprechend der minimalen Länge der Vorhersagen anpassen
y_true_adjusted = y_test_magnitude[:min_length]

# Zeitbasiertes Splitting der Daten in 80% Training und 20% Test
split_index_adjusted = int(len(X_combined_adjusted) * 0.8)
X_train_adjusted, X_test_adjusted = X_combined_adjusted[:split_index_adjusted],
↳ X_combined_adjusted[split_index_adjusted:]
y_train_adjusted, y_test_adjusted = y_true_adjusted[:split_index_adjusted],
↳ y_true_adjusted[split_index_adjusted:]

# Training des Meta-Modells (Lineare Regression)
meta_model = LinearRegression()
meta_model.fit(X_train_adjusted, y_train_adjusted)

# Vorhersagen des Meta-Modells auf dem Testset
predictions_meta_adjusted = meta_model.predict(X_test_adjusted)

# Evaluation mit MSE und MAE
mse_ensemble = mean_squared_error(y_test_adjusted, predictions_meta_adjusted)
mae_ensemble = mean_absolute_error(y_test_adjusted, predictions_meta_adjusted)

print(f'Mean Absolute Error (MAE) des Ensemble-Modells: {mae_ensemble:.4f}')
print(f'Mean Squared Error (MSE) des Ensemble-Modells: {mse_ensemble:.4f}')
```

Mean Absolute Error (MAE) des Ensemble-Modells: 0.8575

Mean Squared Error (MSE) des Ensemble-Modells: 1.6332

14.2 5.2 Vorkommen

```
[40]: # Bestimmen der Gewichtung der einzelnen Vorhersagen basierend auf der Präzision
weights = np.array([precision_lr, precision_rf, precision_ann, precision_lstm])
# Normalisieren der Gewichte, damit sie sich zu 1 summieren
weights = weights / weights.sum()

# Anpassen der Länge der anderen Vorhersagen an die Länge der LSTM-Vorhersagen
min_length = len(predictions_occurs_lstm.ravel())
predictions_occurs_lr_adjusted = predictions_occurs_lr[:min_length]
predictions_occurs_rf_adjusted = predictions_occurs_rf[:min_length]
predictions_occurs_ann_adjusted = predictions_occurs_ann[:min_length]

# Gewichtete Vorhersagen berechnen
weighted_predictions = np.column_stack((
    predictions_occurs_lr_adjusted,
    predictions_occurs_rf_adjusted,
    predictions_occurs_ann_adjusted,
    predictions_occurs_lstm.ravel() # Achten Sie darauf, dass die Dimensionen
    ↳übereinstimmen
)) * weights

# Da AdaBoost nicht direkt mit kontinuierlichen gewichteten Summen umgehen kann,
# könnten wir als einfaches Beispiel den gewichteten Durchschnitt als einziges
↳Feature verwenden:
average_weighted_prediction = np.mean(weighted_predictions, axis=1).reshape(-1,
↳1)

# AdaBoost als Meta-Modell verwenden
meta_model = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=1),
    n_estimators=100,
    random_state=42
)

# Anpassen der y-Werte
y_true_occurs_adjusted = y_test_occurs[:min_length]

# Training des Meta-Modells auf der Basis der gewichteten Durchschnitte
meta_model.fit(average_weighted_prediction, y_true_occurs_adjusted)

# Vorhersagen des Meta-Modells
final_predictions = meta_model.predict(average_weighted_prediction)
```

```

# Evaluation
accuracy_ensemble = accuracy_score(y_true_occurs_adjusted, final_predictions)
precision_ensemble = precision_score(y_true_occurs_adjusted, final_predictions,
    ↪zero_division=0)
recall_ensemble = recall_score(y_true_occurs_adjusted, final_predictions)

print(f"Final Accuracy: {accuracy_ensemble:.4f}")
print(f"Final Precision: {precision_ensemble:.4f}")
print(f"Final Recall: {recall_ensemble:.4f}")

```

Final Accuracy: 0.8731
Final Precision: 0.5496
Final Recall: 0.6248

14.3 Vergleich aller finalen Ergebnisse:

```

[41]: # MAE
print(f"{'Metrik':<10} {'LR':<10} {'RF':<10} {'ANN':<10} {'LSTM':<10}␣

```

```

    ↪{'Ensemble':<10}")
print(f"{'MAE':<10} {mae_lr:<10.4f} {mae_rf:<10.4f} {mae_ann:<10.4f} {mae_lstm:
    ↪<10.4f} {mae_ensemble:<10.4f}")

# MSE
print(f"\n{'Metrik':<10} {'LR':<10} {'RF':<10} {'ANN':<10} {'LSTM':<10}␣
    ↪{'Ensemble':<10}")
print(f"{'MSE':<10} {mse_lr:<10.4f} {mse_rf:<10.4f} {mse_ann:<10.4f} {mse_lstm:
    ↪<10.4f} {mse_ensemble:<10.4f}")

# Precision
print(f"\n{'Metrik':<10} {'LR':<10} {'RF':<10} {'ANN':<10} {'LSTM':<10}␣
    ↪{'Ensemble':<10}")
print(f"{'Precision':<10} {precision_lr:<10.4f} {precision_rf:<10.4f}␣
    ↪{precision_ann:<10.4f} {precision_lstm:<10.4f} {precision_ensemble:<10.4f}")

# Accuracy
print(f"\n{'Metrik':<10} {'LR':<10} {'RF':<10} {'ANN':<10} {'LSTM':<10}␣
    ↪{'Ensemble':<10}")
print(f"{'Accuracy':<10} {accuracy_lr:<10.4f} {accuracy_rf:<10.4f} {accuracy_ann:
    ↪<10.4f} {accuracy_lstm:<10.4f} {accuracy_ensemble:<10.4f}")

# Recall

```

```

print(f"\n{'Metrik':<10} {'LR':<10} {'RF':<10} {'ANN':<10} {'LSTM':<10}␣
↳{'Ensemble':<10}")
print(f"{'Recall':<10} {recall_lr:<10.4f} {recall_rf:<10.4f} {recall_ann:<10.4f}␣
↳{recall_lstm:<10.4f} {recall_ensemble:<10.4f}")

```

Metrik	LR	RF	ANN	LSTM	Ensemble
MAE	0.6867	0.8181	0.7265	0.6679	0.8575
Metrik	LR	RF	ANN	LSTM	Ensemble
MSE	1.5077	1.6796	1.4639	1.2887	1.6332
Metrik	LR	RF	ANN	LSTM	Ensemble
Precision	0.1377	0.2158	0.4204	0.1758	0.5496
Metrik	LR	RF	ANN	LSTM	Ensemble
Accuracy	0.6510	0.5683	0.8200	0.3882	0.8731
Metrik	LR	RF	ANN	LSTM	Ensemble
Recall	0.7006	0.7628	0.6731	0.8830	0.6248

LITERATUR

- [1] M. S. Abdalzaher, M. S. Soliman und S. M. El-Hady. "Seismic Intensity Estimation for Earthquake Early Warning Using Optimized Machine Learning Model". In: *IEEE Transactions on Geoscience and Remote Sensing* 61.Art no. 5914211 (2023), S. 1–11. DOI: [10.1109/TGRS.2023.3296520](https://doi.org/10.1109/TGRS.2023.3296520).
- [2] Md. Kamrul Ahmed, Rifat Seraj und S. M. Shahriar Islam. "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation". In: *Electronics* 9.8 (2020), S. 1295. DOI: [10.3390/electronics9081295](https://doi.org/10.3390/electronics9081295). URL: <https://doi.org/10.3390/electronics9081295>.
- [3] S. Ardabili, A. Mosavi und A. R. Várkonyi-Kóczy. "Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods". In: *Engineering for Sustainable Future. INTER-ACADEMIA 2019*. Hrsg. von András Várkonyi-Kóczy. Bd. 101. Lecture Notes in Networks and Systems. Cham: Springer, 2020. DOI: [10.1007/978-3-030-36841-8_21](https://doi.org/10.1007/978-3-030-36841-8_21). URL: https://doi.org/10.1007/978-3-030-36841-8_21.
- [4] G. Asencio-Cortés, A. Morales-Esteban, X. Shang und F. Martínez-Álvarez. "Earthquake prediction in California using regression algorithms and cloud-based big data infrastructure". In: *Computers & Geosciences* 115 (2018), S. 198–210. ISSN: 0098-3004. DOI: [10.1016/j.cageo.2017.10.011](https://doi.org/10.1016/j.cageo.2017.10.011). URL: <https://doi.org/10.1016/j.cageo.2017.10.011>.
- [5] K. M. Asim, A. Idris, T. Iqbal und F. Martínez-Álvarez. "Earthquake prediction model using support vector regressor and hybrid neural networks". In: *PLOS ONE* 13.7 (2018), e0199004. DOI: [10.1371/journal.pone.0199004](https://doi.org/10.1371/journal.pone.0199004). URL: <https://doi.org/10.1371/journal.pone.0199004>.
- [6] K. M. Asim, A. Idris, T. Iqbal und F. Martínez-Álvarez. "Seismic indicators based earthquake predictor system using Genetic Programming and AdaBoost classification". In: *Soil Dynamics and Earthquake Engineering* 111 (2018), S. 1–7. DOI: [10.1016/j.soildyn.2018.04.020](https://doi.org/10.1016/j.soildyn.2018.04.020). URL: <https://doi.org/10.1016/j.soildyn.2018.04.020>.
- [7] K. M. Asim, F. Martínez-Álvarez, A. Basit und T. Iqbal. "Earthquake magnitude prediction in Hindukush region using machine learning techniques". In: *Natural Hazards* 85.1 (2016), 471–486. DOI: [10.1007/s11069-016-2579-3](https://doi.org/10.1007/s11069-016-2579-3).
- [8] K. M. Asim, S. S. R. Moustafa, I. A. Niaz, E. Elawadi, T. Iqbal und F. Martínez-Álvarez. "Seismicity analysis and machine learning models for short-term low magnitude seismic activity predictions in Cyprus". In: *Soil Dynamics and Earthquake Engineering* 130 (2020), S. 105932. DOI: [10.1016/j.soildyn.2019.105932](https://doi.org/10.1016/j.soildyn.2019.105932). URL: <https://doi.org/10.1016/j.soildyn.2019.105932>.

- [9] Mohamad Awad und Rahul Khanna. "Support Vector Regression". In: *Efficient Learning Machines*. Berkeley, CA: Apress, 2015. DOI: [10.1007/978-1-4302-5990-9_4](https://doi.org/10.1007/978-1-4302-5990-9_4). URL: https://doi.org/10.1007/978-1-4302-5990-9_4.
- [10] Gregory C. Beroza, Margarita Segou und S. Mostafa Mousavi. "Machine learning and earthquake forecasting—next steps". In: *Nature Communications* 12.1 (2021). DOI: [10.1038/s41467-021-24952-6](https://doi.org/10.1038/s41467-021-24952-6). URL: <https://doi.org/10.1038/s41467-021-24952-6>.
- [11] T. Bhandarkar, K. Vardaan, N. Satish, S. S. Sridhar, R. Sivakumar und S. Ghosh. "Earthquake trend prediction using long short-term memory RNN". In: *International Journal of Electrical and Computer Engineering* 9.2 (2019), S. 1304. DOI: [10.11591/ijece.v9i2.pp1304-1312](https://doi.org/10.11591/ijece.v9i2.pp1304-1312). URL: <https://doi.org/10.11591/ijece.v9i2.pp1304-1312>.
- [12] S. Biswas, D. Kumar und U. K. Bera. "Prediction of earthquake magnitude and seismic vulnerability mapping using artificial intelligence techniques: a case study of Turkey". In: *Research Square* (2023). Preprint version. URL: <https://doi.org/10.21203/rs.3.rs-2863887/v1>.
- [13] British Geological Survey. *What causes earthquakes? - British Geological Survey*. <https://www.bgs.ac.uk/discovering-geology/earth-hazards/earthquakes/what-causes-earthquakes/>. Juni 2021.
- [14] Tianling Chai und Roland R. Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature". In: *Geoscientific Model Development* 7 (2014), S. 1247–1250. DOI: [10.5194/gmd-7-1247-2014](https://doi.org/10.5194/gmd-7-1247-2014). URL: <https://doi.org/10.5194/gmd-7-1247-2014>.
- [15] S. Chanda, M. C. Raghucharan, K. K. Reddy, V. Chaudhari und S. Somala. "Duration prediction of Chilean strong motion data using machine learning". In: *Journal of South American Earth Sciences* 109 (2021), S. 103253. DOI: [10.1016/j.jsames.2021.103253](https://doi.org/10.1016/j.jsames.2021.103253). URL: <https://doi.org/10.1016/j.jsames.2021.103253>.
- [16] P. Chittora, T. Chakrabarti, P. Debnath, A. Gupta, P. Chakrabarti, S. P. Praveen, M. Margala und A. A. Elngar. "Experimental analysis of earthquake prediction using machine learning classifiers, curve fitting, and neural modeling". In: *Research Square* (2022). Preprint version. URL: <https://doi.org/10.21203/rs.3.rs-1896823/v2>.
- [17] F. Corbi, L. Sandri, J. Bedford, F. Funicello, S. Brizzi, M. Rosenau und S. Lallemand. "Machine Learning Can Predict the Timing and Size of Analog Earthquakes". In: *Geophysical Research Letters* 46.3 (2019). Research Letter, S. 1303–1311. DOI: [10.1029/2018GL081251](https://doi.org/10.1029/2018GL081251). URL: <https://doi.org/10.1029/2018GL081251>.
- [18] P. Debnath, P. Chittora, T. Chakrabarti, P. Chakrabarti, Z. Leonowicz, M. Jasiński, R. Goňo und E. Jasińska. "Analysis of earthquake forecasting in India using supervised machine learning classifiers". In: *Su-*

- sustainability* 13.2 (2021), S. 971. DOI: [10.3390/su13020971](https://doi.org/10.3390/su13020971). URL: <https://doi.org/10.3390/su13020971>.
- [19] R. D. Decker, S. Rašković und K. Brunsiek. "Diskriminanzanalyse". In: *Handbuch der sozialwissenschaftlichen Datenanalyse*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 495–523.
- [20] F. Esposito, D. Malerba, G. Semeraro und J. Kay. "A comparative analysis of methods for pruning decision trees". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.5 (Mai 1997), S. 476–491. DOI: [10.1109/34.589207](https://doi.org/10.1109/34.589207).
- [21] Yusuf Essam, Pavitra Kumar, Ali Najah Ahmed, Muhammad Ary Murti und Ahmed El-Shafie. "Exploring the reliability of different artificial intelligence techniques in predicting earthquake for Malaysia". In: *Soil Dynamics and Earthquake Engineering* 147 (2021). ISSN: 0267-7261. DOI: [10.1016/j.soildyn.2021.106826](https://doi.org/10.1016/j.soildyn.2021.106826). URL: <https://doi.org/10.1016/j.soildyn.2021.106826>.
- [22] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), S. 861–874. ISSN: 0167-8655. DOI: [10.1016/j.patrec.2005.10.010](https://doi.org/10.1016/j.patrec.2005.10.010). URL: <https://doi.org/10.1016/j.patrec.2005.10.010>.
- [23] Artur J. Ferreira und Mario A. Figueiredo. "Boosting algorithms: A review of methods, theory, and applications". In: *Ensemble machine learning: Methods and applications*. 2012, S. 35–85.
- [24] E. Florido, G. Asencio–Cortés, J. L. Aznarte, C. Rubio-Escudero und F. Martínez–Álvarez. "A novel tree-based algorithm to discover seismic patterns in earthquake catalogs". In: *Computers & Geosciences* 115 (2018), S. 96–104. ISSN: 0098-3004. DOI: [10.1016/j.cageo.2018.03.005](https://doi.org/10.1016/j.cageo.2018.03.005). URL: <https://doi.org/10.1016/j.cageo.2018.03.005>.
- [25] Robert J. Geller. "Earthquake Prediction: A Critical Review". In: *Geophysical Journal International* (1997).
- [26] Jan Gertheiss und Gerhard Tutz. "Generalisierte lineare und gemischte Modelle". In: *Moderne Verfahren der Angewandten Statistik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023, S. 1–37.
- [27] John P. Grotzinger und Thomas H. Jordan. "Erdbeben". In: *Springer eBooks*. 2016, S. 335–370. DOI: [10.1007/978-3-662-48342-8_13](https://doi.org/10.1007/978-3-662-48342-8_13). URL: https://doi.org/10.1007/978-3-662-48342-8_13.
- [28] H. Gursoy u. a. "Deep Learning for Earthquake Prediction". In: *Journal of Seismology* (2023).
- [29] Sebastian Hainzl, Günter Zöller und Jürgen Kurths. "SELF-ORGANIZED CRITICALITY MODEL FOR EARTHQUAKES: QUIESCENCE, FORESHOCKS AND AFTERSHOCKS". In: *International Journal Of Bifurcation And Chaos* 09.12 (1999), S. 2249–2255. DOI: [10.1142/s0218127499001711](https://doi.org/10.1142/s0218127499001711). URL: <https://doi.org/10.1142/s0218127499001711>.

- [30] Peyman Hajikhodaverdikhan, Mohammad Nazari, Mehrdad Mohsenizadeh, Shahaboddin Shamsirband und Kwok-Wing Chau. "Earthquake prediction with meteorological data by particle filter-based support vector regression". In: *Engineering Applications of Computational Fluid Mechanics* 12.1 (2018), S. 679–688. DOI: [10.1080/19942060.2018.1512010](https://doi.org/10.1080/19942060.2018.1512010).
- [31] Andrea Hildebrandt, Sebastian Jäckle, Fabian Wolf und Andreas Heindl. "Lineare und logistische Regression". In: *Springer eBooks*. 2015, S. 63–108. DOI: [10.1007/978-3-531-18993-2_4](https://doi.org/10.1007/978-3-531-18993-2_4). URL: https://doi.org/10.1007/978-3-531-18993-2_4.
- [32] J. Huang, X. Wang, Y. Zhao, X. Chen und X. Han. "Large Earthquake Magnitude Prediction in Taiwan Based on Deep Learning Neural Network". In: *Neural Network World* 28.2 (2018), S. 149–160. DOI: [10.14311/nnw.2018.28.009](https://doi.org/10.14311/nnw.2018.28.009). URL: <https://doi.org/10.14311/nnw.2018.28.009>.
- [33] Abby Hutchison. *How machine learning might unlock earthquake prediction*. Dez. 2023. URL: <https://www.technologyreview.com/2023/12/29/108>.
- [34] R. Jain, A. Nayyar, S. Arora und et al. "A comprehensive analysis and prediction of earthquake magnitude based on position and depth parameters using machine and deep learning models". In: *Multimedia Tools and Applications* 80 (2021), S. 28419–28438. DOI: [10.1007/s11042-021-11001-z](https://doi.org/10.1007/s11042-021-11001-z). URL: <https://doi.org/10.1007/s11042-021-11001-z>.
- [35] Jürgen Jünger und Christine Gärtner. "Maschinelles Lernen". In: *Computational Methods für die Sozial- und Geisteswissenschaften*. Wiesbaden: Springer VS, 2023, S. 305–354. DOI: [10.1007/978-3-658-37747-2_8](https://doi.org/10.1007/978-3-658-37747-2_8).
- [36] S. Karimzadeh, M. Matsuoka, J. Kuang und G. Liu. "Spatial prediction of aftershocks triggered by a major earthquake: a binary machine learning perspective". In: *ISPRS International Journal of Geo-information* 8.10 (2019), S. 462. DOI: [10.3390/ijgi8100462](https://doi.org/10.3390/ijgi8100462). URL: <https://doi.org/10.3390/ijgi8100462>.
- [37] F. Khosravikia und P. Clayton. "Machine learning in ground motion prediction". In: *Computers & Geosciences* 148 (2021), S. 104700. DOI: [10.1016/j.cageo.2021.104700](https://doi.org/10.1016/j.cageo.2021.104700). URL: <https://doi.org/10.1016/j.cageo.2021.104700>.
- [38] O. Kramer. "K-Nearest Neighbors". In: *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Bd. 51. Intelligent Systems Reference Library. Berlin, Heidelberg: Springer, 2013. DOI: [10.1007/978-3-642-38652-7_2](https://doi.org/10.1007/978-3-642-38652-7_2). URL: https://doi.org/10.1007/978-3-642-38652-7_2.
- [39] H. Kubo, T. Kunugi, W. Suzuki, S. Suzuki und S. Aoi. "Hybrid predictor for ground-motion intensity with machine learning and conventional ground motion prediction equation". In: *Scientific Reports* 10.1 (2020). DOI: [10.1038/s41598-020-68630-x](https://doi.org/10.1038/s41598-020-68630-x). URL: <https://doi.org/10.1038/s41598-020-68630-x>.

- [40] Jian-Hong Lin, Chien-Cheng Chao und Jyh-Cheng Chiou. "Determining neuronal number in each hidden layer using earthquake catalogues as training data in training an embedded back propagation neural network for predicting earthquake magnitude". In: *IEEE Access* 6 (2018), S. 52582–52597. DOI: [10.1109/access.2018.2870189](https://doi.org/10.1109/access.2018.2870189). URL: <https://doi.org/10.1109/access.2018.2870189>.
- [41] S. K. Majhi, S. S. Hossain und T. Padhi. "MFOFLANN: moth flame optimized functional link artificial neural network for prediction of earthquake magnitude". In: *Evolving Systems* 11.1 (2019), S. 45–63. DOI: [10.1007/s12530-019-09293-6](https://doi.org/10.1007/s12530-019-09293-6). URL: <https://doi.org/10.1007/s12530-019-09293-6>.
- [42] Michael E. Mavroforakis und Sergios Theodoridis. "A Geometric Approach to Support Vector Machine (SVM) Classification". In: *IEEE Transactions On Neural Networks* 17.3 (2006), S. 671–682. DOI: [10.1109/tnn.2006.873281](https://doi.org/10.1109/tnn.2006.873281). URL: <https://doi.org/10.1109/tnn.2006.873281>.
- [43] M. A. Meier u. a. "Machine Learning in Seismology: Turning Data into Insights". In: *Seismological Research Letters* (2017).
- [44] Ministry of Education, Culture, Sports, Science and Technology (MEXT). *Seismology Toward Research innovation with data of Earthquake (STAR-E Project)*. 2021. URL: https://www.mext.go.jp/en/policy/science_technology/research_development/title01/detail01/mext_00001.html.
- [45] I. M. Murwantara, P. Yugopuspito und R. Hermawan. "Comparison of machine learning performance for earthquake prediction in Indonesia using 30 years historical data". In: *TELKOMNIKA Telecommunication Computing Electronics and Control* 18.3 (2020), S. 1331. DOI: [10.12928/telkomnika.v18i3.14756](https://doi.org/10.12928/telkomnika.v18i3.14756). URL: <https://doi.org/10.12928/telkomnika.v18i3.14756>.
- [46] C. Panagopulos. *AI-Driven earthquake forecasting shows promise in trials*. Dez. 2023. URL: <https://news.utexas.edu/2023/10/05/ai-driven-earthquake-forecasting-shows-promise-in-trials/>.
- [47] M. H. Rafiei und H. Adeli. "NEEWS: A novel earthquake early warning model using neural dynamic classification and neural dynamic optimization". In: *Soil Dynamics and Earthquake Engineering* 100 (2017), 417–427. DOI: [10.1016/j.soildyn.2017.05.013](https://doi.org/10.1016/j.soildyn.2017.05.013).
- [48] Gururaj Rebala, Ankit Ravi und Sunita Churiwala. "Machine Learning Definition and Basics". In: *An Introduction to Machine Learning*. Cham: Springer, 2019. DOI: [10.1007/978-3-030-15729-6_1](https://doi.org/10.1007/978-3-030-15729-6_1). URL: https://doi.org/10.1007/978-3-030-15729-6_1.
- [49] M. H. D. M. Ribeiro und L. D. S. Coelho. "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series". In: *Applied Soft Computing* 86 (2020), S. 105837. DOI: [10.1016/j.asoc.2019.105837](https://doi.org/10.1016/j.asoc.2019.105837). URL: <https://doi.org/10.1016/j.asoc.2019.105837>.

- [50] Bertrand Rouet-Leduc u. a. "Machine Learning Predictions of Earthquake Timing". In: *Nature* (2017).
- [51] Bertrand Rouet-Leduc, Claudia Hulbert, Nicholas Lubbers, Kévin Barros, Colin J. Humphreys und Paul A. Johnson. "Machine learning predicts laboratory earthquakes". In: *Geophysical Research Letters* 44.18 (2017), S. 9276–9282. DOI: [10.1002/2017gl074677](https://doi.org/10.1002/2017gl074677). URL: <https://doi.org/10.1002/2017gl074677>.
- [52] O. M. Saad u. a. "Machine Learning for Fast and Reliable Source-Location Estimation in Earthquake Early Warning". In: *IEEE Geoscience and Remote Sensing Letters* 19.Art no. 8025705 (2022), S. 1–5. DOI: [10.1109/LGRS.2022.3142714](https://doi.org/10.1109/LGRS.2022.3142714).
- [53] B. Sadhukhan, S. Chakraborty, S. Mukherjee und R. K. Samanta. "Climatic and seismic data-driven deep learning model for earthquake magnitude prediction". In: *Frontiers in Earth Science* 11 (2023). DOI: [10.3389/feart.2023.1082832](https://doi.org/10.3389/feart.2023.1082832). URL: <https://doi.org/10.3389/feart.2023.1082832>.
- [54] M. A. Salam, L. Ibrahim und D. S. Abdelminaam. "Earthquake Prediction using Hybrid Machine Learning Techniques". In: *International Journal of Advanced Computer Science and Applications* 12.5 (2021). DOI: [10.14569/ijacsa.2021.0120578](https://doi.org/10.14569/ijacsa.2021.0120578). URL: <https://doi.org/10.14569/ijacsa.2021.0120578>.
- [55] A. K. Saleem und A. N. Rashid. "Applications of machine learning for earthquake prediction: A review". In: *AIP Conference Proceedings* (2023). DOI: [10.1063/5.0119623](https://doi.org/10.1063/5.0119623). URL: <https://doi.org/10.1063/5.0119623>.
- [56] Jesús Santos-Reyes. "Introductory Chapter: Earthquakes - Impact, Community Vulnerability, and Resilience". In: *IntechOpen eBooks*. 2019. DOI: [10.5772/intechopen.86284](https://doi.org/10.5772/intechopen.86284). URL: <https://doi.org/10.5772/intechopen.86284>.
- [57] Kenji Satake. "Advances in earthquake and tsunami sciences and disaster risk reduction since the 2004 Indian ocean tsunami". In: *Geosci. Lett.* 1 (2014), S. 15. DOI: [10.1186/s40562-014-0015-7](https://doi.org/10.1186/s40562-014-0015-7). URL: <https://doi.org/10.1186/s40562-014-0015-7>.
- [58] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), S. 85–117. ISSN: 0893-6080. DOI: [10.1016/j.neunet.2014.09.003](https://doi.org/10.1016/j.neunet.2014.09.003).
- [59] Javed Mehedi Shamrat, Ravi Ranjan, Kazi Md. Hasib, Abhishek Yadav und A. H. Siddique. "Performance Evaluation Among ID3, C4.5, and CART Decision Tree Algorithm". In: *Pervasive Computing and Social Networking*. Hrsg. von Geetha Ranganathan, Robert Bestak, Ramalingam Palanisamy und Álvaro Rocha. Bd. 317. Lecture Notes in Networks and Systems. Singapore: Springer, 2022. DOI: [10.1007/978-981-16-5640-8_11](https://doi.org/10.1007/978-981-16-5640-8_11). URL: https://doi.org/10.1007/978-981-16-5640-8_11.

- [60] Muhammad Nizam Shodiq, Dendi Hendrianto Kusuma und Muhammad Ghifari Rifqi. "Neural Network for Earthquake Prediction Based on Automatic Clustering in Indonesia". In: *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION* 2 (2018), S. 37–43.
- [61] U.S. Geological Survey. *Earthquake Magnitude, Energy Release, and Shaking Intensity*. URL: <https://www.usgs.gov/programs/earthquake-hazards/earthquake-magnitude-energy-release-and-shaking-intensity>.
- [62] *USGS Earthquake Hazards Program: Significant Earthquakes*. <https://earthquake.usgs.gov/earthquakes/browse/significant.php?year=1900>. Zugriff: 31. März 2024.
- [63] United States Geological Survey. *United States Geological Survey*. 2023.
- [64] Qian Wang, Yu Guo, Lei Yu und Lei Pan. "Earthquake prediction based on Spatio-Temporal Data Mining: An LSTM Network approach". In: *IEEE Transactions on Emerging Topics in Computing* 8.1 (2020), S. 148–158. DOI: [10.1109/tetc.2017.2699169](https://doi.org/10.1109/tetc.2017.2699169). URL: <https://doi.org/10.1109/tetc.2017.2699169>.
- [65] Y. Wang, Z. Wang, Z. Cao und J. Lan. "Deep learning for magnitude prediction in earthquake early warning". In: *Gondwana Research* 123 (2023), S. 164–173. DOI: [10.1016/j.gr.2022.06.009](https://doi.org/10.1016/j.gr.2022.06.009).
- [66] P. Xiong, L. Tong, K. Zhang, X. Shen, R. Battiston, D. Ouzounov, R. Iuppa, D. Crookes, C. Long und H. Zhou. "Towards advancing the earthquake forecasting by machine learning of satellite data". In: *Science of the Total Environment* 771 (2021), S. 145256. DOI: [10.1016/j.scitotenv.2021.145256](https://doi.org/10.1016/j.scitotenv.2021.145256). URL: <https://doi.org/10.1016/j.scitotenv.2021.145256>.
- [67] F.-J. Yang. "An Implementation of Naive Bayes Classifier". In: *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA, 2018, S. 301–306. DOI: [10.1109/CSCI46756.2018.00065](https://doi.org/10.1109/CSCI46756.2018.00065).
- [68] M. Yousefzadeh, S. A. Hosseini und M. Farnaghi. "Spatiotemporally explicit earthquake prediction using deep neural network". In: *Soil Dynamics and Earthquake Engineering* 144 (2021), S. 106663. DOI: [10.1016/j.soildyn.2021.106663](https://doi.org/10.1016/j.soildyn.2021.106663). URL: <https://doi.org/10.1016/j.soildyn.2021.106663>.
- [69] G. Zhang, B. E. Patuwo und M. Y. Hu. "Forecasting with artificial neural networks". In: *International Journal Of Forecasting* 14.1 (1998), S. 35–62. DOI: [10.1016/s0169-2070\(97\)00044-7](https://doi.org/10.1016/s0169-2070(97)00044-7). URL: [https://doi.org/10.1016/s0169-2070\(97\)00044-7](https://doi.org/10.1016/s0169-2070(97)00044-7).