

**Anwendung von  
künstlichen neuronalen  
Netzwerken auf die  
Berechnung von kritischen  
Parameterkurven**

**Erweiterte Machbarkeitsstudie  
und erste Anwendungen**

## **Anwendung von künstlichen neuronalen Netzwerken auf die Berechnung von kritischen Parameterkurve**

Erweiterte Machbarkeitsstudie  
und erste Anwendungen

Tatiana Reiche  
Robert Kilger  
Fabian Sommer

September 2025

### **Anmerkung:**

Das diesem Bericht zugrunde liegende Eigenforschungsvorhaben wurde mit Mitteln des Bundesministeriums für Umwelt, Klimaschutz, Naturschutz und nukleare Sicherheit (BMUKN) unter dem Förderkennzeichen 4722E03240 durchgeführt.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der GRS.

Der Bericht gibt die Auffassung und Meinung der GRS wieder und muss nicht mit der Meinung des BMUKN übereinstimmen.

**Deskriptoren**

KI, KNN, KNN-Modellierung, Kritikalitätsparameter, Künstliche Intelligenz, künstliche neuronale Netze, Vorhersage von Kritikalitätsparametern, Vorhersagemodelle

## Kurzfassung

Moderne Methoden des maschinellen Lernens, wie künstliche neuronale Netze, sind aktuell Gegenstand intensiver internationaler Forschung, da sie in der Lage sind, komplexe und nichtlineare Datenstrukturen zu erfassen und durch geeignete Modellarchitekturen und Trainingstechniken robuste Vorhersagen zu ermöglichen.

Im Themengebiet der nuklearen Sicherheitsforschung bezieht sich ein mögliches Anwendungsgebiet auf die Erstellung von kritischen Parameterkurven. Diese existieren meist für eine Vielzahl von Anreicherungen und mit einer umfangreichen Abdeckung des Konzentrationsbereichs von  $^{235}\text{U}$  über mehrere Größenordnungen hinweg, sodass ihre Erstellung ein zeit- und rechenintensives Unterfangen ist.

Der vorliegende Bericht dokumentiert die Untersuchungen der GRS zur möglichen Anwendung von künstlichen neuronalen Netzen zur Erstellung von kritischen Parameterkurven. Aufbauend auf einer Machbarkeitsstudie wurden dabei verschiedene Architekturen neuronaler Netze untersucht und deren Modellparameter und Trainingstechniken optimiert. Die Anwendung eines immer komplexeren Modells ist bei Anwendung eines einzigen Modells auf mehrere Ausgangsparameter, bzw. für die Anwendung auf mehrere chemischen Systeme notwendig.

In jeder Phase der Arbeiten wurde in einem weiten Konzentrationsbereich eine gute Reproduzierbarkeit der tatsächlichen Daten erreicht. In den extremen Randbereichen sehr niedriger und sehr hoher Spaltstoffkonzentration der Kurven, die von starken Nichtlinearitäten geprägt sind, verringert sich die Vorhersagekraft. Der Vergleich mit linearer Interpolation zeigte durchweg eine bessere Reproduzierbarkeit durch die neuronalen Netze, vor allem zu diesen Randbereichen hin.



## Abstract

Modern machine learning methods, such as artificial neural networks, are currently the subject of intensive international research. They are capable of capturing complex and nonlinear data structures and, through suitable model architectures and training techniques, enabling robust predictions.

In the field of nuclear safety research, one potential area of application lies in the generation of critical parameter curves. These typically exist for a wide range of enrichments and cover the concentration range of  $^{235}\text{U}$  over several orders of magnitude. Therefore, their generation is a time- and computation-intensive endeavor.

This report documents the investigations conducted by GRS into the potential application of artificial neural networks for generating critical parameter curves. Building on a feasibility study, various neural network architectures were examined, and their model parameters and training techniques optimized. The use of increasingly complex models is necessary when applying a single model to multiple input parameters or across various chemical systems.

At each stage of the work, good reproducibility of the actual data was achieved over a wide range of concentrations. In regions of very low or very high fissile material concentrations, at the extreme edge of the parameter curves, which are characterized by strong nonlinearities, the predictive power decreases. In comparison, linear interpolation consistently showed inferior reproducibility, especially in these edge regions, where the neural networks performed significantly better.



# Inhaltsverzeichnis

	<b>Kurzfassung.....</b>	<b>I</b>
	<b>Abstract.....</b>	<b>III</b>
<b>1</b>	<b>Einleitung .....</b>	<b>1</b>
1.1	Motivation .....	1
1.2	Machbarkeitsstudie.....	1
1.3	Ziel dieser Arbeit.....	2
1.4	Aufbau des Berichts.....	3
<b>2</b>	<b>Hintergrund und Methoden .....</b>	<b>5</b>
2.1	Einführung in neuronale Netze.....	5
2.2	Allgemeine Aspekte der verwendeten neuronalen Netzwerk-Typen .....	9
2.3	Verwendete neuronale Netzwerks-Typen .....	11
2.3.1	SNN-LR-Modell (Stacked Neural Networks with Linear Regressor) .....	11
2.3.2	MLP-Modell (Multi-Layer Perceptron) .....	12
2.3.3	MTL-Modell (Multi-Task Learning Model).....	13
2.3.4	MTL-SNN-LR-Modell (Multi-Task Learning Stacked Neural Networks with Linear Regressor).....	15
2.3.5	TMLP-Modell (Tri-Input MLP).....	16
2.4	Datenaufbereitung .....	17
2.4.1	Struktur der Daten .....	18
2.4.2	Zielvariablen und Gruppierung.....	19
2.4.3	Datenaufbereitungsschritte .....	22
2.5	Leistungsmetriken und Evaluierungsmethoden.....	24
2.5.1	Metriken zur Modellbewertung .....	24
2.5.2	Visualisierungen zur Modellevaluierung.....	28
<b>3</b>	<b>Ergebnisse und Analyse .....</b>	<b>31</b>
3.1	Phase 1: Entwicklung und Analyse von Basismodellen .....	31

3.1.1	Modellierungsansatz.....	32
3.1.2	Datengrundlage .....	33
3.1.3	Ergebnisse.....	34
3.1.4	Fazit.....	42
3.2	Phase 2: Integration eines neuen Kritikalitätsparameters.....	42
3.2.1	Modellierungsansatz.....	43
3.2.2	Datengrundlage .....	43
3.2.3	Ergebnisse.....	44
3.2.4	Fazit.....	56
3.3	Phase 3: Anwendung auf weitere chemische Systeme.....	56
3.3.1	Modellierungsansatz.....	56
3.3.2	Datengrundlage .....	57
3.3.3	Ergebnisse.....	57
3.3.4	Fazit.....	61
<b>4</b>	<b>KNN-Modellentwicklung: Erkenntnisse und Empfehlungen.....</b>	<b>63</b>
<b>5</b>	<b>Fazit und Ausblick.....</b>	<b>67</b>
	<b>Literaturverzeichnis.....</b>	<b>71</b>
	<b>Abbildungsverzeichnis.....</b>	<b>73</b>
	<b>Tabellenverzeichnis.....</b>	<b>75</b>
	<b>Abkürzungen.....</b>	<b>77</b>

# 1 Einleitung

## 1.1 Motivation

Die Modellierung kritischer Parameterkurven spielt eine zentrale Rolle in zahlreichen wissenschaftlichen und technischen Bereichen der Kritikalitätssicherheit. Klassische Ansätze, wie die lineare Interpolation, sind zwar einfach anzuwenden, stoßen jedoch schnell an ihre Grenzen, insbesondere wenn die zugrunde liegenden Zusammenhänge komplex und nichtlinear sind. Diese Einschränkungen führen oft zu ungenauen Vorhersagen und mangelnder Flexibilität.

Moderne Methoden des maschinellen Lernens, wie künstliche neuronale Netze (KNN), bieten vielversprechende Alternativen. Sie sind in der Lage, komplexe und nichtlineare Datenstrukturen besser zu erfassen und durch geeignete Modellarchitekturen und Trainingstechniken robuste Vorhersagen zu ermöglichen.

Die Anwendung solcher maschinellen Lernmethoden auf kritische Kurven ist ein neues Forschungsgebiet. Es stehen derzeit keine etablierten Standards oder Regeln zur Verfügung, was eine fundierte Bewertung dieser Ansätze erschwert. Daher ist eine umfassende empirische Untersuchung erforderlich, um die Stärken und Schwächen der Methoden besser zu verstehen und fundierte Aussagen über ihre Leistungsfähigkeit im Umfeld der Kritikalitätssicherheit, speziell bei der Anwendung auf kritische Parameterkurven, treffen zu können. Diese Arbeit beschränkt sich auf kritische Parameterkurven von Ammoniumdiuranat(ADU)- und Ammoniumuranylcarbonat(AUC)-Wasser- Systemen.

## 1.2 Machbarkeitsstudie

In einer vorangegangenen Machbarkeitsstudie /EIS 22/ wurde ein SNN-LR-Modell (Stacked Neural Networks with Linear Regressor, siehe Kapitel 2.3.1) entwickelt und analysiert. Erste Tests zeigten, dass dieses Modell bei der Vorhersage kritischer Parameter präziser war als die lineare Interpolation, die bisher als Referenzmethode galt. Trotz dieser Erfolge traten auch einige Herausforderungen zutage, insbesondere in Bezug auf die hohe Komplexität der Modellarchitektur und den erheblichen Aufwand bei der Optimierung der Hyperparameter. Hyperparameter bezeichnen die KNN-

Einstellungen, die vor dem Training eines Modells festgelegt werden und das Lernverhalten maßgeblich beeinflussen.

### **1.3 Ziel dieser Arbeit**

Das Ziel dieser Arbeit war die systematische Entwicklung und Analyse von Modellierungsansätzen zur Berechnung von kritischen Parameterkurven mithilfe künstlicher neuronaler Netze (KNN). Diese erstmalige Anwendung in der Kritikalitätssicherheit bietet eine potenziell effiziente Alternative zu herkömmlichen Interpolationsmethoden, wie der linearen Interpolation, und steht im Fokus dieser Arbeit. Da für diesen Anwendungsbereich bisher keine etablierten Standards oder Richtlinien existieren, wurde die Methodik schrittweise entwickelt und untersucht, um ein fundiertes Verständnis zu gewinnen und eine solide Grundlage für zukünftige Anwendungen zu schaffen.

Im Fokus stehen dabei die folgenden Fragestellungen:

- Untersuchung und Vergleich verschiedener Netzwerkarchitekturen.
- Optimierung von Modellparametern und Trainingsverfahren.
- Gegenüberstellung der KNN-Methodik mit der klassischen linearen Interpolation.
- Erweiterung der KNN-Modelle um zusätzliche Eingangs- und Zielvariablen, um ihre Generalisierungsfähigkeit zu testen.
- Untersuchung der Anwendbarkeit auf chemisch ähnliche Systeme.

Ein wesentliches Ziel ist es, besser zu verstehen, welche Problemkomplexitäten durch neuronale Netze sinnvoll erfasst werden können und wann diese Modelle an ihre Grenzen stoßen. Dabei wurde auch untersucht, ob einfache und nachvollziehbare Netzwerkarchitekturen ausreichen, um komplexe Probleme effektiv zu lösen.

Durch diese Untersuchungen soll ein fundiertes Verständnis der Stärken und Schwächen der untersuchten Methoden entwickelt werden. Diese Untersuchungen leisten einen Beitrag zur Etablierung von Best Practices in diesem bisher in der GRS wenig erforschten Anwendungsfeld. Die Weiterentwicklung und Standardisierung von Methoden zur Vorhersage von Kritikalitätsparametern kann so nachhaltig unterstützt werden.

## 1.4 Aufbau des Berichts

Der Bericht ist wie folgt gegliedert:

- **Kapitel 1:** Einleitung - Motivation, Vorarbeiten und Ziele dieser Arbeit.
- **Kapitel 2:** Grundlagen neuronaler Netze, Beschreibung der Datenbasis und Datenaufbereitung, Erläuterung der angewandten Evaluierungsmethoden.
- **Kapitel 3:** Ergebnisse der durchgeführten Untersuchungen in drei Phasen, wobei jede Phase eine spezifische Erweiterung oder Anpassung der Modelle behandelt.
- **Kapitel 4:** Generelle Erkenntnisse aus der Modellentwicklung und deren Relevanz für zukünftige Arbeiten.
- **Kapitel 5:** Fazit und Ausblick auf potenzielle zukünftige Forschungsrichtungen.



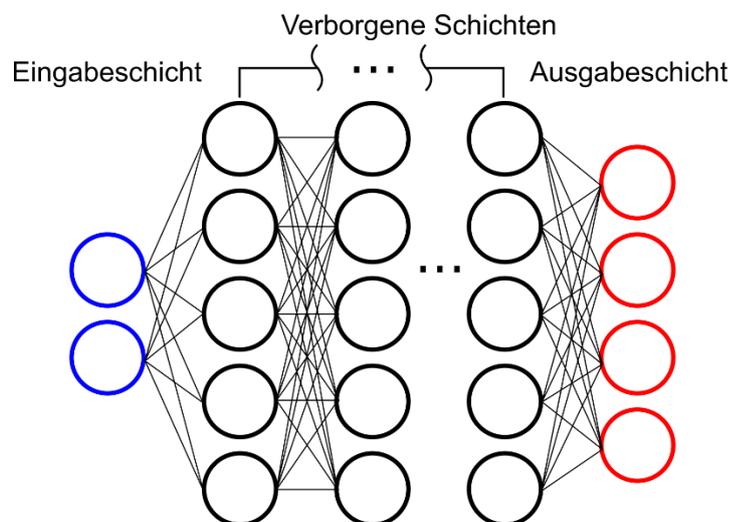
## 2 Hintergrund und Methoden

Dieses Kapitel beschreibt die verwendeten künstlichen neuronalen Netzwerke, die Datenaufbereitung, sowie die Methoden zur Evaluierung der Modellleistung.

Die gesamte Implementierung erfolgte in Python, einer Programmiersprache, die sich dank ihrer umfangreichen Bibliotheken und starken Unterstützung für maschinelles Lernen als Standardwerkzeug in diesem Bereich international etabliert hat /PYT 23/.

### 2.1 Einführung in neuronale Netze

*Künstliche neuronale Netze (KNN)* sind rechnergestützte Modelle, die von der Struktur und Funktionsweise biologischer Neuronen inspiriert sind. Indem KNN aus Trainingsdaten lernen, werden sie im Anschluss dazu verwendet, Muster in Daten zu erkennen und komplexe Aufgaben zu lösen. Sie bestehen aus miteinander verbundenen Knoten, auch *Neuronen* (engl.: *neurons*) genannt, die in der Regel in mehreren aufeinanderfolgenden *Schichten (layer)* organisiert sind. In der *Eingabeschicht (input layer)* werden die Eingabedaten in das Netzwerk eingespeist. Diese Daten durchlaufen eine oder mehrere der sogenannten *verborgenen Schichten (hidden layer)*, wobei jede Schicht zusätzliche Berechnungen durchführt. Die *Ausgabeschicht (output layer)* liefert die endgültige Vorhersage (*prediction*) des Modells. Eine optische Repräsentation eines typischen KNN ist in Abb. 2.1 dargestellt. Zur Vollständigkeit sei erwähnt, dass es zahlreiche Architekturen und Typen von KNN gibt, die von der hier dargestellten Illustration abweichen können.



**Abb. 2.1** Typische Repräsentation eines KNN

Die Verbindungen der Neuronen zwischen den einzelnen Schichten haben ein sogenanntes *Gewicht*, wobei dieses Gewicht die Stärke der Verbindung repräsentiert. Diese Gewichte werden während des Trainingsprozesses stetig angepasst, um die Vorhersagen des Netzes zu optimieren.

Trainiert werden KNN mit Trainingsdaten, wobei die Eingangsdaten durch das Netz propagiert werden und die Ausgabe des Netzes mit den jeweiligen Zieldaten abgeglichen wird. Der somit resultierende Fehler der Vorhersage wird mit einer sogenannten *Verlustfunktion (loss function)* quantitativ berechnet. Die Wahl der Verlustfunktion ist wichtig, da sie im allgemeinen problemspezifisch ist und den Trainingsprozess maßgeblich steuert. Der mit der Verlustfunktion bestimmte Fehler wird dazu verwendet, um die Vorhersagefähigkeit des Netzes zu verbessern. Dabei werden mithilfe des Fehlers die Gewichte der Verbindungen zwischen den Neuronen angepasst. Dieser Trainingsprozess basiert auf dem *Backpropagation*-Algorithmus [LI 12], wobei der Fehler rückwärts durch die Schichten propagiert wird, von der Ausgabeschicht zurück zur Eingabeschicht. Dabei werden die Gradienten der Verlustfunktion bezüglich der Gewichte berechnet und genutzt, um die Gewichte so anzupassen, dass der Fehler schrittweise minimiert wird. Da es sich dabei um einen iterativen Trainingsprozess handelt, sind in der Regel mehrere Trainingsläufe (*Epochen*, engl. *epochs*) notwendig, bis das KNN zufriedenstellende Vorhersagen trifft.

Die elementare Einheit eines KNN ist das Neuron mit der folgenden grundlegenden Funktionsweise: im Neuron werden alle dem Neuron zur Verfügung stehenden Eingangssignale  $x_i$  (eingehende Verbindungen zu Neuronen anderer Schichten) mit den jeweiligen trainierbaren Gewichten  $w_i$  multipliziert und die Produkte aufsummiert. Zusätzlich verfügt jedes Neuron über den sogenannten *Bias*  $b$  mit dem diese Summe verschoben werden kann. Da es sich beim Bias auch um einen trainierbaren Parameter handelt, unterscheiden wir im folgenden Text nicht zwischen den Gewichten und dem Bias. Um das Ausgangssignal  $y$  des Neurons zu erzeugen, wird auf diese Summe eine sogenannte *Aktivierungsfunktion*  $f$  angewandt. Somit ergibt sich für das Ausgangssignal eines Neurons:

$$y = f(\sum_i w_i x_i + b) \quad (2.1)$$

Die Wahl der Aktivierungsfunktion ist entscheidend, da sie die Modellierung nichtlinearer Zusammenhänge ermöglicht und somit die Lösung komplexer Probleme unterstützt.

Wesentliche Eigenschaften neuronaler Netze sind:

- **Lernfähigkeit:** KNN können Muster in Daten erkennen und nichtlineare Zusammenhänge modellieren.
- **Flexibilität:** Die KNN-Architektur lässt sich an die spezifischen Anforderungen eines Problems anpassen, z. B. durch die Wahl der Anzahl der Schichten, Neuronen pro Schicht oder Aktivierungsfunktionen.
- **Generalität:** KNN können unterschiedlichste Datentypen verarbeiten, einschließlich numerischer Daten, Text, Bilder oder Zeitreihen.

Es gibt keine festen Regeln für das optimale Netzdesign oder die idealen Hyperparameter (KNN-Konfiguration). Die Optimierung neuronaler Netze ist eine problemabhängige Herausforderung, die empirische Tests erfordert, und stellt somit für den Anwender eine zentrale Aufgabe im maschinellen Lernen dar.

Herausforderungen beim Einsatz von KNN:

- **Konvergenz:** Der Trainingsprozess muss sicherstellen, dass das Netz zu einem Zustand führt, in dem es sinnvolle Vorhersagen liefert.
- **Overfitting (Überanpassung):** Es muss verhindert werden, dass das Netz die Trainingsdaten "auswendig lernt", anstatt allgemeine Muster zu erkennen.
- **Lokale Minima:** Das Risiko, dass das Netz in suboptimalen Zuständen (lokalen Minima) "stecken bleibt", muss minimiert werden.
- **Effizienz:** Die Modellkomplexität und der Rechenaufwand müssen so gestaltet werden, dass der Ressourcenverbrauch und die Trainingszeit minimiert werden.

Wichtige Aspekte, die bei der Modellierung und Optimierung von KNN bestimmt werden müssen:

- **Architektur:**
  - Anzahl der Schichten (Eingabe-, Ausgabe- und verborgene Schichten).
  - Anzahl der Neuronen pro Schicht.
  - Wahl der Aktivierungsfunktionen (z. B. *exponential linear unit ELU*, *rectified linear unit ReLU*, *sigmoid*, *tanh*, ...).

- Wahl der Verlustfunktion (z. B. Mean Squared Error).
- **Hyperparameter** (KNN-Einstellungen, die vor dem Training eines Modells festgelegt werden und das Lernverhalten maßgeblich beeinflussen):
  - *Lernrate*: Bestimmt die Schrittgröße, mit der die Gewichtungen angepasst werden.
  - *Batch-Größe*: Anzahl der Datenpunkte, die pro Optimierungsschritt auf einmal verarbeitet werden.
  - *Epochen*: Anzahl der Durchläufe durch den gesamten Trainingsdatensatz.
  - *Regularisierung*: Methoden zur Vermeidung von *Overfitting* (z. B. *Dropout* oder *L2-Norm*).
- **Trainingstechniken**:
  - Verwendung von *Optimierungsalgorithmen* (z. B. *Adam*, *Stochastic Gradient Descent* oder *RMSprop*).
  - *Datenaugmentation*: Künstliches Erweitern der Trainingsdaten durch Transformationen (z. B. Einfügen von Rauschen).
  - *Cross-Validation*: Aufteilen der Daten in mehrere Teilmengen. Das Modell wird abwechselnd auf einigen dieser Teilmengen trainiert und auf den verbleibenden Daten validiert.
- **Evaluierung**:
 

Die Leistung des KNN wird in der Regel bereits während des Trainings mit sogenannten *Validierungsdaten* abgeschätzt, wobei diese Daten auch eingesetzt werden, um z. B. die Hyperparameter-Einstellungen des Modells zu optimieren und sind somit nicht unabhängig vom Trainingsprozess. Im Gegensatz dazu wird das finale, auf den Trainings- und Validierungsdaten optimierte Modell auf einem unabhängigen Datensatz, den sogenannten *Testdatensatz* evaluiert. Um die Modellleistung auf dem Validierungs- und/oder Testdatensatz beurteilen zu können, werden *Metriken* (siehe Kapitel 2.5.1) wie *Mean Squared Error (MSE)*, *Mean Absolute Error (MAE)* oder *Accuracy* verwendet.

KNN werden hauptsächlich für zwei Haupttypen von Problemen eingesetzt:

- *Regressionsprobleme*: Vorhersage kontinuierlicher Werte, wie bei der Modellierung kritischer Parameterkurven in dieser Arbeit.
- *Klassifizierungsprobleme*: Zuordnung von Eingaben zu diskreten Kategorien, z. B. bei der Objekterkennung in Bildern.

Die Art des Problems bestimmt maßgeblich die Wahl der Verlustfunktion und der Evaluierungsmetrik.

Die im Rahmen dieses Berichts verwendeten Netzarchitekturen sowie die relevanten Konfigurationen werden in den folgenden Kapiteln detailliert beschrieben.

## 2.2 Allgemeine Aspekte der verwendeten neuronalen Netzwerk-Typen

Für die Implementierung und das Training der neuronalen Netze wurde das Framework für künstliche Intelligenz (KI) *TensorFlow* verwendet [TEN 24]. Die Architektur der hier verwendeten KNN folgt dem Prinzip der *vollständigen Verbindung* (*fully connected architecture*), bei dem jedes Neuron einer Schicht mit allen Neuronen der nächsten Schicht verbunden ist, siehe Abb. 2.1.

Als Aktivierungsfunktion wird *ELU* (*Exponential Linear Unit*) eingesetzt. Sie minimiert den *Vanishing-Gradient-Effekt*, ein Problem, bei dem die Gradienten während der Backpropagation für tiefe Schichten verschwindend klein werden und infolgedessen das Training des KNN behindert werden kann. Gleichzeitig beschleunigt ELU die Konvergenz des Modells während des Trainings im Vergleich zu anderen Aktivierungsfunktionen wie der Tangens hyperbolicus (*tanh*)-Aktivierungsfunktion.

Als Optimierungsverfahren wird der *Adam-Optimierer* (*Adaptive Moment Estimation*) mit einer anfänglichen Lernrate von 0,1 eingesetzt. Der Adam-Optimierer ist ein leistungsfähiger Algorithmus zur Optimierung von maschinellen Lernmodellen, der sich als besonders effektiv bei großen Datensätzen und komplexen Modellen bewährt hat, da er adaptive Lernraten verwendet und stabil sowie rechenzeiteffizient ist.

Als *Verlustfunktion* wird der *mittlere quadratische Fehler* (*Mean Squared Error, MSE*) eingesetzt, der sich als gängige Wahl für Regressionsprobleme etabliert hat.

Um das Training weiter zu optimieren und *Overfitting* zu vermeiden, kommen verschiedene Techniken zum Einsatz:

- **Early Stopping:** Diese Technik beendet das Training automatisch, wenn die Leistung des Modells auf den Validierungsdaten über eine festgelegte Anzahl von Epochen hinweg keine Verbesserung mehr zeigt. Dadurch wird verhindert, dass das Modell zu lange trainiert wird und ein *Overfitting* auftritt.
- **ModelCheckpoint-Callback:** Ein *Callback* ist eine programmierte Methode, die während des Trainings Aktionen basierend auf bestimmten Ereignissen automatisch ausführt. Der *ModelCheckpoint-Callback* speichert das Modell mit dem niedrigsten *Validierungsverlust* – einem Maß für die Genauigkeit des Modells auf einem separaten Datensatz, der nicht zum Training verwendet wird –, um sicherzustellen, dass stets die beste Modellversion auf Basis der Validierungsdaten vorliegt.
- **ReduceLRonPlateau-Callback:** Dieser *Callback* passt die Lernrate dynamisch an, wenn die Verbesserung des *Validierungsverlusts* über mehrere Epochen hinweg stagniert. Dieser *Callback* wird nur bei einfacheren Architekturen, wie Multilayer Perceptrons (MLP) (siehe Kapitel 2.3.2) und Multi-Task-Learning (MTL) (siehe Kapitel 2.3.3) Modellen angewendet.
- **Dropout:** Die Regularisierung-Technik *Dropout* deaktiviert während des Trainings zufällig einen bestimmten Prozentsatz der Neuronen in den versteckten Schichten. Obwohl *Dropout* insbesondere bei komplexen Modellen (wie MTL und MTL-SNN-LR) effektiv sein sollte und ausdrücklich empfohlen wird, führte der Einsatz dieser Technik in den Voruntersuchungen, die in dieser Arbeit durchgeführt wurden, zu einer Verschlechterung der Modelleleistung. Aus diesem Grund wurde auf *Dropout* bei der Entwicklung der finalen KNN-Modelle verzichtet. Ein Grund für die schlechtere Leistung unter Einbeziehung von *Dropout* könnte der einfache funktionale Zusammenhang zwischen Eingangs- und Zielgrößen bei der hier betrachteten Problemstellung sein, wo eine gewisse Überanpassung des Modells unproblematisch bzw. sogar erwünscht sein könnte.

Die Wahl der *batch-Größe* und der *Epochenanzahl* erfolgt empirisch und wird durch systematische Tests optimiert, um die bestmögliche Leistung zu erreichen.

Durch diese Mechanismen und Techniken wird ein effizienter und flexibler Trainingsprozess gewährleistet, der den Anforderungen des vorliegenden Problems gerecht wird. Bei

den hier durchgeführten verschiedenen Studien, werden, sofern nicht explizit anders erwähnt, diese Einstellungen verwendet.

## 2.3 Verwendete neuronale Netzwerks-Typen

### 2.3.1 SNN-LR-Modell (Stacked Neural Networks with Linear Regressor)

Das *Stacked Neural Networks with Linear Regressor (SNN-LR) Modell* wurde im Rahmen der Machbarkeitsstudie (siehe /EIS 22/ und Kapitel 1.2) entwickelt und ansatzweise getestet, wobei die Daten für das Ammoniumdiuranat(ADU)-Wasser-System verwendet wurden. Ziel des Modells ist es, auf der Basis von den zwei Eingabeparametern Urankonzentration und  $^{235}\text{U}$ -Anreicherung Vorhersagen für die vier Hauptzielvariablen kritische Kugelmasse, Kugelvolumen, Zylinderdurchmesser und Schichtdicke zu treffen. Bei dem *Stacked Ensemble-Learning*-Ansatz, der hier implementiert wurde, handelt es sich um eine Technik des Ensemble Learnings, bei der mehrere Modelle kombiniert werden. Dabei wird durch die Anwendung dieses Ansatzes erwartet robustere Ergebnisse zu erzielen, als es mit einem einzelnen Modell möglich wäre.

#### Modellarchitektur und Training

Zunächst werden fünf unabhängige KNN auf einem Trainingsdatensatz trainiert. Jedes dieser Modelle, auch als *Basis-Modelle* bezeichnet, erhält die gleichen Eingabedaten. Jedoch ist für jedes dieser Netze der Lernprozess aufgrund der zufälligen Initialisierung der jeweiligen Gewichte unterschiedlich.

Nach zahlreichen empirischen Untersuchungen wurde für die Architektur und die Hyperparameter der Basis-Modelle folgende optimale Konfiguration ermittelt:

- **Eingabeschicht:** zwei Neuronen, entsprechend den Eingabeparametern.
- **Verborgene Schichten:** zehn vollständig verbundene Schichten mit jeweils fünf Neuronen.
- **Ausgabeschicht:** vier Neuronen, entsprechend den vier Hauptzielvariablen.
- **Batch-Größe:** 256
- **Trainingsdauer:** 10.000 Epochen

Um die Vorhersagen der fünf Basis-Modelle miteinander zu kombinieren, wurde ein *Meta-Modell* in Form einer *linearen Regression* implementiert. Nach dem Training der Basis-Modelle werden deren Vorhersagen auf einem separaten Validierungsdatensatz verwendet, um einen neuen Datensatz, den sogenannten *gestapelten Datensatz*, zu erstellen. Dieser Datensatz besteht aus den Vorhersagen der einzelnen KNN und dient als Eingabe für das Meta-Modell.

Das lineare Regressionsmodell lernt für jede Zielvariable spezifische Gewichtungen. Diese bestimmen den Beitrag der Vorhersagen jedes einzelnen neuronalen Netzes zur finalen Prognose. Für jede Zielvariable werden fünf Gewichtungen optimiert – eine für jedes der fünf KNN. Die endgültige Vorhersage ergibt sich als gewichtete lineare Kombination der Vorhersagen aller fünf KNN, basierend auf den erlernten Parameter.

### **Modellanwendung**

Beim Laden des trainierten Modells werden sowohl die Gewichte der Basis-Modelle als auch das trainierte Meta-Modell geladen. Die Vorhersage erfolgt in zwei Stufen:

1. Jedes der Basis-Modelle (KNN) liefern seine individuellen Vorhersagen.
2. Das Meta-Modell kombiniert anschließend diese Vorhersagen, um eine endgültige, optimierte Vorhersage zu berechnen.

### **2.3.2 MLP-Modell (Multi-Layer Perceptron)**

Als Alternative zum SNN-LR-Modell wurde ein weiteres KNN, das *Multi-Layer Perceptron (MLP)*, zur Vorhersage von Kritikalitätsparametern entwickelt. Im Gegensatz zum SNN-LR-Modell, das eine Kombination aus mehreren KNN darstellt, die in einer Ensemble-Methodik zusammenarbeiten, handelt es sich beim MLP-Modell um ein einzelnes KNN mit einer klar strukturierten, sequenziellen Architektur. Diese Struktur ermöglicht eine direkte und effiziente Verarbeitung der Eingabedaten, was einen interessanten Vergleich zum komplexeren SNN-LR-Ansatz bietet.

### **Modellarchitektur**

Das MLP-Modell verwendet eine schlichte, aber effiziente Struktur aus mehreren vollständig verbundenen Schichten. Nach umfangreichen empirischen Tests wurde die optimale Konfiguration der Architektur und der Hyperparameter wie folgt festgelegt:

- **Eingabeschicht:** zwei Neuronen, entsprechend den Eingabeparametern.
- **Verborgene Schichten:** vier vollständig verbundene Schichten mit je fünf Neuronen.
- **Ausgabeschicht:** vier Neuronen für vier Hauptzielvariablen.
- **Batch-Größe:** 128
- **Trainingsdauer:** 200 Epochen

Diese Architektur ermöglichte eine ausgewogene Balance zwischen Effizienz und Genauigkeit und erwies sich als effektiv für die Aufgabenstellung der ersten Phase dieser Arbeit.

### 2.3.3 MTL-Modell (Multi-Task Learning Model)

Im Rahmen der zweiten Phase dieser Arbeit (siehe Kapitel 3.2) wurde das bestehende MLP-Modell um eine weitere Zielvariable – den infiniten Multiplikationsfaktor  $k_{inf}$  – ergänzt und erweitert. Diese Anpassung stellte eine besondere Herausforderung dar, da die vier Hauptzielvariablen und die neue Zielvariable auf unterschiedlichen Datenbasen beruhen.

Die beiden Eingabeparameter, Urankonzentration und  $^{235}\text{U}$ -Anreicherung, sind für alle fünf Zielvariablen identisch. Allerdings teilen die vier Hauptzielvariablen eine weitgehend gemeinsame Datenbasis hinsichtlich der Urankonzentrationswerte. Die Urankonzentrationswerte für die fünfte Zielvariable, den infiniten Multiplikationsfaktor, stimmen jedoch nicht mit denen der Hauptzielvariablen überein (siehe Kapitel 2.4.2). Dies führte dazu, dass zwei separate Datensätze innerhalb eines einzigen Modells verarbeitet werden müssen.

Eine simple Erweiterung der Ausgabeschicht der bestehenden Modelle um ein zusätzliches Neuron für die neue Zielvariable war aufgrund der unterschiedlichen Datenbasen nicht möglich. Die Option, zwei separate Modelle zu trainieren – eines für die Hauptzielvariablen und eines für die abweichende Zielvariable – wurde bewusst ausgeschlossen: Ziel dieser Arbeit war es, ein Modell zu entwickeln, das schrittweise komplexer wird und in der Lage ist, komplexe Aufgaben gleichzeitig zu verarbeiten, ohne diese auf mehrere spezialisierten Modelle aufzuteilen.

Die Lösung bestand in der Entwicklung eines *Multi-Task Learning (MTL)* Modells. Dieser Ansatz ermöglicht es, mehrere verwandte Aufgaben gleichzeitig zu lernen, indem Informationen aus unterschiedlichen Datensätzen genutzt werden. Das Modell wurde so gestaltet, dass es zwei Eingabeschichten und zwei Ausgabeschichten umfasst, um die unterschiedlichen Anforderungen der Zielvariablen zu integrieren.

### **Modellarchitektur**

Nach umfassenden Tests und Optimierungen wurde die Architektur des MTL-Modells wie folgt festgelegt:

- **Eingabeschichten:** zwei separate Eingabeschichten, jeweils bestehend aus zwei Neuronen, entsprechend der jeweiligen Anzahl an Eingabeparametern.
- **Spezifische verborgene Schichten:** drei Schichten mit jeweils 32 Neuronen, die die Ausgaben der Eingabeschichten verarbeiten und kombinieren.
- **Gemeinsame verborgene Schichten:** drei weitere Schichten mit ebenfalls 32 Neuronen zur Verarbeitung komplexer Muster.
- **Ausgabeschichten:** zwei separate Ausgabeschichten:
  - eine Ausgabeschicht mit vier Neuronen für die Hauptzielvariablen,
  - eine Ausgabeschicht mit einem Neuron für die abweichende Zielvariable.
- **Hyperparameter:**
  - Batch-Größe: 64
  - Trainingsdauer: 150 Epochen

Diese Architektur erlaubt es dem Modell, sowohl gemeinsame Merkmale zu lernen als auch spezialisierte Vorhersagen für die jeweiligen Zielvariablen zu treffen.

### **Trainingsprozess**

Das Modell wurde mit zwei separaten Trainingsdatensätzen trainiert, wobei beide Eingabeschichten parallel verwendet wurden, um die jeweiligen Zielvariablen vorherzusagen.

Der Lernfortschritt wurde anhand der mittleren aufsummierten quadratischen Verluste (MSE) der beiden Ausgabeschichten überwacht. Dabei wurden die MSE für jede Ausgabeschicht separat berechnet und deren Summe als Gesamtverlustfunktion genutzt. Diese Methode gewährleistet eine gleichmäßige Berücksichtigung beider Aufgaben und verhindert eine Benachteiligung einer der Ausgaben.

Zusätzlich wurde der Verlust für jede Ausgabeschicht einzeln überwacht, um die Performance für jede spezifische Aufgabe differenziert zu analysieren. Eine dynamische Anpassung der Hyperparameter während des Trainings sorgte für zusätzliche Optimierungen.

#### **2.3.4 MTL-SNN-LR-Modell (Multi-Task Learning Stacked Neural Networks with Linear Regressor)**

Das *Multi-Task Learning Stacked Neural Networks with Linear Regressor (MTL-SNN-LR)* Modell wurde als Alternative zum MTL-Modell entwickelt und basiert auf einem *Stacked Neural Network* mit linearem Regressor (SNN-LR). Ziel war es, die Vorteile des Stacking-Verfahrens mit den Fähigkeiten eines MTL-Modells zu kombinieren, um eine höhere Vorhersagegenauigkeit und Flexibilität bei der Integration mehrerer Zielvariablen zu erreichen. Der Modellansatz sollte mit dem MTL-Modell verglichen werden, um Leistungsunterschiede zu evaluieren.

#### **Modellarchitektur**

Die Architektur des MTL-SNN-LR-Modells kombiniert die Prinzipien des SNN-LR-Modells (siehe Kapitel 2.3.1) mit den Eigenschaften eines MTL-Modells (Kapitel 2.3.3):

- **Einzelne Basis-KNN-Modelle:**

Grundlegend folgt die Architektur der Basis-KNN der MTL-Architektur gemäß Kapitel 2.3.3 mit folgenden Vereinfachungen:

- **Spezifische verborgene Schichten:** drei Schichten mit jeweils fünf Neuronen verarbeiten und kombinieren die Ausgaben der Eingabeschichten.
- **Gemeinsame verborgene Schichten:** drei weitere Schichten mit ebenfalls jeweils 5 Neuronen erkennen komplexe Muster und Zusammenhänge.

- **Hyperparameter** wurden nach umfangreichen empirischen Tests wie folgt optimiert:
  - Batch-Größe: 64
  - Trainingsdauer: 10.000 Epochen
- **Meta-Modell:**
  - Ein linearer Regressor (siehe Kapitel 2.3.1) wird als Meta-Modell eingesetzt. Dieser kombiniert die Vorhersagen der Basis-Modelle und liefert die finalen Werte für alle Zielvariablen.

### 2.3.5 TMLP-Modell (Tri-Input MLP)

In der dritten Arbeitsphase (siehe Kapitel 3.3) wurde das *Tri-Input-Multi-Task Learning* (TMLP), eine Erweiterung des MLP-Modells, entwickelt. Als Weiterentwicklung zum MLP kann das TMLP Kritikalitäts-Vorhersagen für mehrere verschiedene chemische Systeme treffen. Dazu wurde das ursprüngliche MLP um einen zusätzlichen Eingabeparameter erweitert, der zwischen den chemischen Systemen unterscheidet. Das chemische System wird hierbei als kategoriale Variable modelliert und muss für die Verarbeitung im Modell numerisch kodiert werden. Dazu eignen sich zwei Ansätze:

- **Label-Encoding** (kategorialer Parameter mit Integer-Werten (0, 1, 2, ...)): Jeder Kategorie wird ein eindeutiger Integer-Wert zugewiesen. Vorteile sind Einfachheit und Effizienz; für die Implementierung des Parameters wird nur ein zusätzliches Neuron in der Eingabeschicht benötigt. Ein potenzieller Nachteil ist die mögliche Einführung einer unbeabsichtigten Ordnungsbeziehung zwischen den Kategorien, falls mehr als zwei Kategorien berücksichtigt werden. Dies kann in bestimmten Situationen problematisch sein.
- **One-Hot-Encoding (OHE):** Jede Kategorie erhält einen eigenen Eingangskanal, dessen Signal mit 1 oder 0 angibt, ob die jeweilige Kategorie zutrifft. Dadurch entsteht keine Ordnungsbeziehung zwischen den Kategorien. Auch können mehrere, gleichzeitig zutreffende Kategorien modelliert werden. Allerdings kann die Anzahl an Eingangskanälen und somit an Neuronen in der Eingabeschicht bei vielen Kategorien stark ansteigen und verbraucht somit mehr Speicherplatz.

In der vorliegenden Arbeit werden nur zwei chemische Systeme (ADU und AUC) und somit zwei Kategorien betrachtet. Daher wurde, um das KNN-Modell einfach zu gestalten, das Label-Encoding gewählt: 0 für ADU, 1 für AUC.

### **Modellarchitektur**

Das TMLP-Modell verwendet, ähnlich wie das MLP, mehrere vollständig verbundene Schichten. Nach umfangreichen Tests wurde folgende optimale Konfiguration der Architektur und der Hyperparameter festgelegt:

- **Eingabeschicht:** drei Neuronen, entsprechend den Eingabeparametern.
- **Verborgene Schichten:** vier vollständig verbundene Schichten mit jeweils fünf Neuronen.
- **Ausgabeschicht:** vier Neuronen für vier Hauptzielvariablen.
- **Batch-Größe:** 128
- **Trainingsdauer:** 200 Epochen

## **2.4 Datenaufbereitung**

Im Rahmen dieser Arbeit wird der Ansatz *Überwachtes Lernen (Supervised Learning)* zum Trainieren der KNN angewendet. Dafür sind Datenpaare erforderlich, die aus *Eingabeparametern (input parameters)* und den entsprechenden *Zielvariablen (labels oder outputs)* bestehen.

Als Eingabeparameter dienen die Urankonzentration und die <sup>235</sup>U-Anreicherung sowie – bei den komplexen Modellen der Phase 3 – das chemische System. Die zugrunde liegenden Daten stammen aus Berechnungen kritischer Parameterkurven für unterschiedliche Materialsysteme, die zuvor mit dem Code *SCALE (Standardized Computer Analyses for Licensing Evaluation)* /REA 17/ durchgeführt wurden. Diese Kurven sind im Handbuch zur Kritikalität der GRS dokumentiert /GRS 19/. Weitere Details zur Berechnung und Validierung der Kurven finden sich in /SOM 19/. Die berechneten Daten sind in Dateien in einem festgelegten Format gespeichert.

## 2.4.1 Struktur der Daten

Die Rohdaten, aus denen die Trainingsdaten extrahiert werden, sind wie folgt aufgebaut und werden anhand einer Beispieldatei in Abb. 2.2 dargestellt:

- **Header:** Enthält Kommentare und Metadaten. Unter anderem wird angegeben, um welches Materialsystem und welchen Kritikalitätsparameter es sich handelt sowie welche  $^{235}\text{U}$ -Anreicherungen betrachtet wurden.
- **Daten:**
  - Die eigentlichen Daten sind in mehreren *Serien* organisiert, wobei jede *Serie* einer im Header definierten  $^{235}\text{U}$ -Anreicherung entspricht. Die Reihenfolge der Serien folgt der im Header definierten Reihenfolge.
  - Jede Serie enthält eine Reihe von *Datenpunkten*. Jeder Datenpunkt besteht aus Paaren von Werten: Der erste Wert ist die Urankonzentration, der zweite Wert ist der zugehörige Kritikalitätsparameter (z. B. kritische Kugelmasse).

```
# comment
# Daten neu gerechnet 2018: *****
iLengthData: 90 95 94 93 89 97 92 94 95 97 95 98
sLegends: 3,5/_% 4/_% 4,5/_% 5/_% 5,5/_% 6/_% 7/_% 8/_% 10/_% 25/_% 50/_% 100/_%
sLegendsEng: 3,5/_% 4/_% 4,5/_% 5/_% 5,5/_% 6/_% 7/_% 8/_% 10/_% 25/_% 50/_% 100/_%
sColors: black red green blue cyan magenta yellow orange chartreuse brown navy olive
sLinestyles: - - - - - - - - - - -
fXrange: 0.01 1.5
fYrange: 0.5 5000
bLogX: True
bLogY: True
sTitle: U235-AUC-HOM
sDate: 23.01.2018
sDateEng: 01-23-2018
sTitleX: Urankonzentration/_$c_ \mathbf{U}$/_[g/_U/cm$^3$]
sTitleXEng: uranium/_concentration/_$c_ \mathbf{U}$/_[g/_U/cm$^3$]
sTitleY: kritische/_Kugelmasse/_$M_ \mathbf{k}$/_[kg/_U]
sTitleYEng: critical/_sphere/_mass/_$M_ \mathbf{k}$/_[kg/_U]
iPosLegend: 2
sComments: 30/_cm/_H$ 2$0/_Reflektor
sCommentsEng: 30/_cm/_H$ 2$0/_reflector
fCommentsPosRel: 0.05,0.02

data: 1. series
7.663900000000000157e-01 4.414208069937031716e+06
7.675199999999999800e-01 1.577298291512043914e+06
```

**Abb. 2.2** Ausschnitt einer Beispieldatei mit Rohdaten, die für das Training weiterverarbeitet werden

## 2.4.2 Zielvariablen und Gruppierung

In dieser Arbeit wurden fünf Kritikalitätsparameter als Zielvariablen untersucht, die in zwei Gruppen aufgeteilt wurden:

- **Hauptzielvariablen:**

- kritische Kugelmasse ( $M_k$ )
- kritisches Kugelvolumen ( $V_k$ )
- kritischer Zylinderdurchmesser ( $D_k$ )
- kritische Schichtdicke ( $S_k$ )

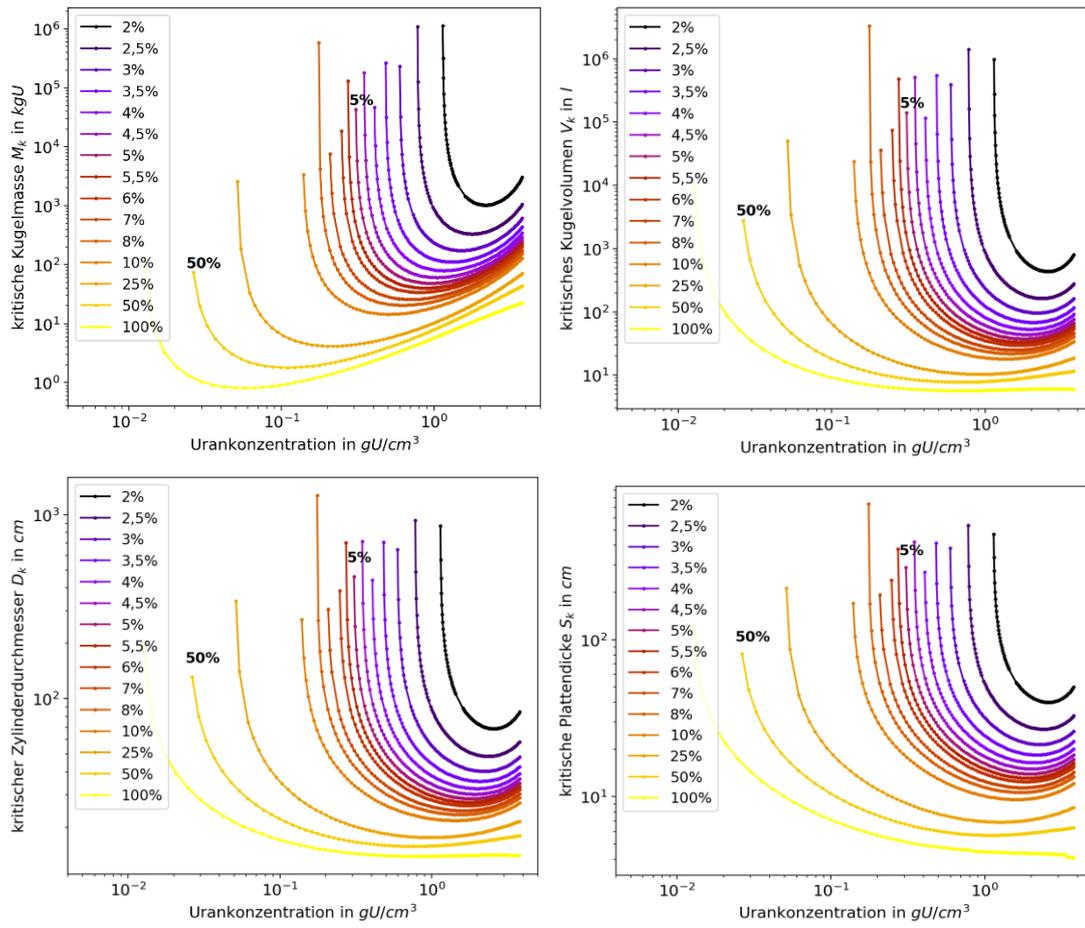
Diese Variablen weisen für beide in dieser Studie untersuchten chemischen Systeme, homogene Ammoniumdiuranat(ADU)- und Ammoniumuranylcarbonat(AUC)-Wasser-Systeme, ähnliche kritische Parameterkurven auf (siehe Abb. 2.3 und Abb. 2.4). Alle kritischen Werte der ADU- und AUC-Systeme wurden für vollständig wasserreflektierte Systeme bestimmt: 30 cm dicker Wasserreflektor, Dichte: 1 g/cm<sup>3</sup>.

Für jede der berechneten <sup>235</sup>U-Anreicherung besitzen beide chemischen Systeme Datenpunkte für alle Hauptzielvariablen bei denselben Urankonzentrationswerten, was die Datenverarbeitung und das Training der KNN deutlich vereinfachen.

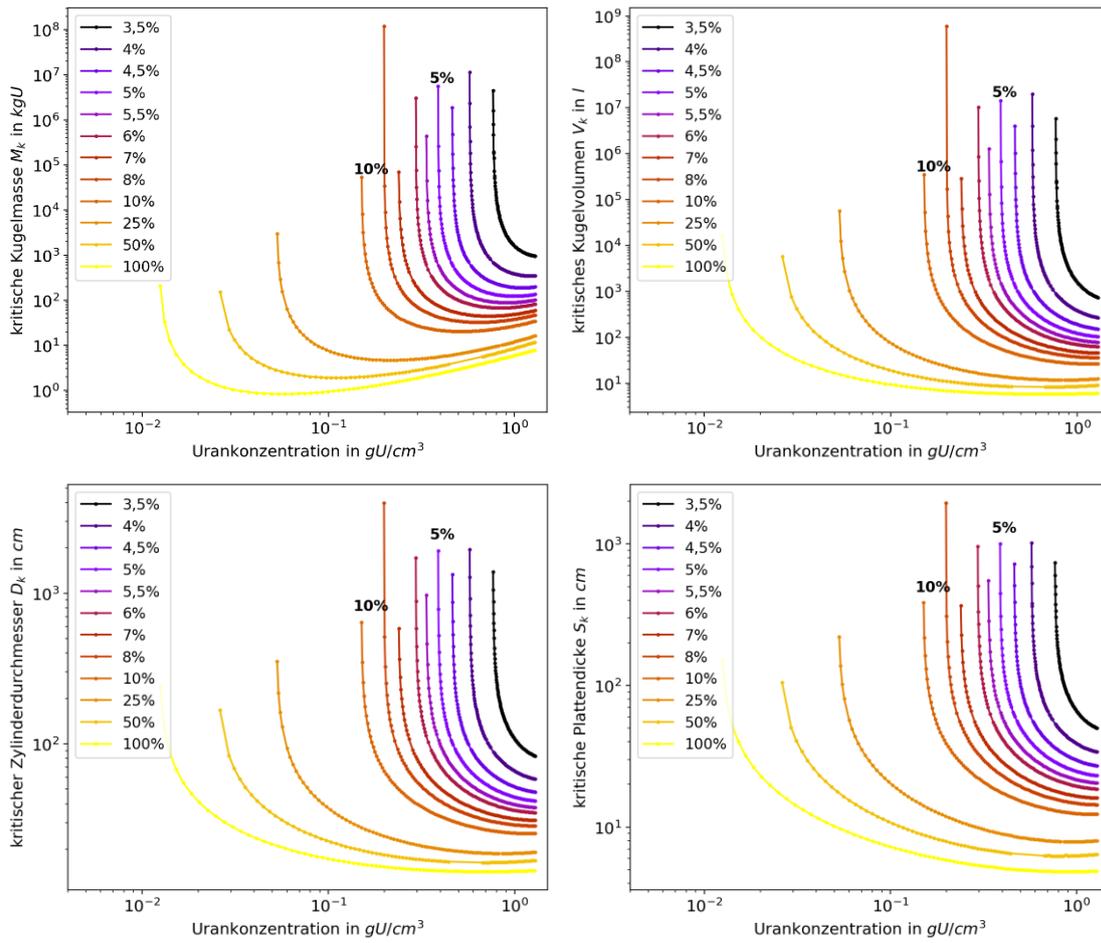
- **Abweichende Zielvariable:**

- unendlicher Multiplikationsfaktor ( $k_{inf}$ )

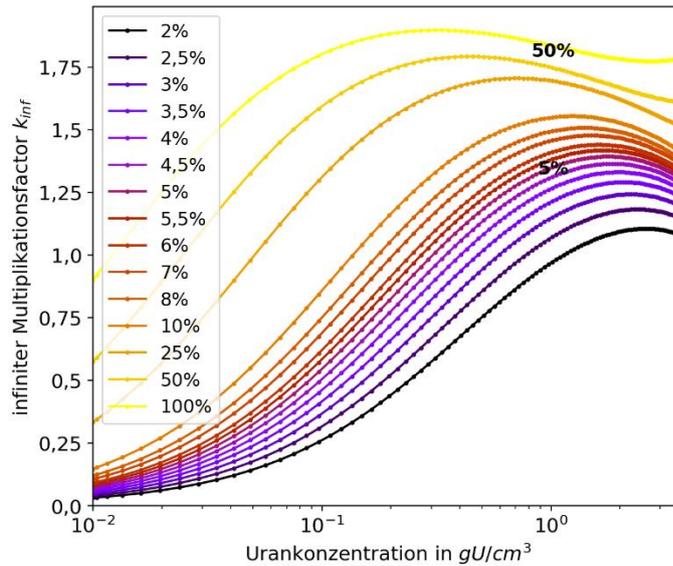
Die funktionale Abhängigkeit von  $k_{inf}$  von der Urankonzentration unterscheidet sich signifikant von entsprechenden Kurvenverläufen der Hauptzielvariablen (siehe Abb. 2.5). Zudem stimmen die Urankonzentrationswerte für jede der <sup>235</sup>U-Anreicherung nicht mit den Werten der Hauptzielvariablen überein. Aufgrund dieser Besonderheiten wird  $k_{inf}$  als abweichende Zielvariable bezeichnet. Die  $k_{inf}$ -Kurven haben für die beiden untersuchten chemischen Systeme einen sehr ähnlichen Kurvenverlauf.



**Abb. 2.3** Kritikalitätsparameter von homogenen ADU-Systemen für verschiedene  $^{235}\text{U}$ -Anreicherungen als Funktion der Urankonzentration



**Abb. 2.4** Kritikalitätsparameter von homogenen AUC-Systemen für verschiedene  $^{235}U$ -Anreicherungen als Funktion der Urankonzentration



**Abb. 2.5**  $k_{inf}$  als Funktion der Urankonzentration für verschiedene  $^{235}U$ -Anreicherungen von homogenen Ammoniumdiuranat(ADU)-Wasser-Systemen

### 2.4.3 Datenaufbereitungsschritte

Im Folgenden werden die verschiedenen Schritte zur Aufbereitung der Daten dargestellt.

#### 1. Auswahl geeigneter Datensätze für die Hauptzielvariablen:

Es wurden nur Datensätze verwendet, die für eine bestimmte  $^{235}U$ -Anreicherung und Urankonzentration berechnete Werte für alle Zielvariablen (alle vier kritischen Parameter) enthalten. Datensätze mit fehlenden Werten wurden entfernt. Dies macht etwa 10 % der Daten aus. Diese entfernten Datensätze stammen vor allem von den Rändern des jeweils betrachteten Konzentrationsbereichs, den Bereichen mit hohen Werten der Kritikalitätsparameter. Bei den entfernten Datensätzen wurde eines der Stoffsysteme oder einer der Hauptzielvariablen nicht als kritisch berechnet. Die Bereiche zeigen zudem starke Nichtlinearitäten auf und waren bereits zu Beginn im Datensatz unterrepräsentiert. Das Entfernen dieser Daten hat die ohnehin geringe Abdeckung dieser nichtlinearen Bereiche weiter reduziert, was potenziell negative Auswirkungen auf die Modellgüte haben könnte.

## 2. Speicherung der Daten:

Die verbleibenden Daten wurden in einem *pandas DataFrame* gespeichert, einer tabellenartigen Datenstruktur, die in Python häufig verwendet wird, um Daten in Zeilen und Spalten zu organisieren. Die Spalten umfassen:

- `['uranium concentration', 'critical spherical mass', 'critical spherical volume', 'critical cylindrical diameter', 'critical slab thickness', 'enrichment']`
- und `['k inf']`, wenn der unendliche Multiplikationsfaktor  $k_{inf}$  einbezogen wurde,
- sowie `[material_type]`, wenn chemische Systeme unterschieden werden müssen.

## 3. Logarithmieren:

Die Daten in allen Spalten wurden logarithmisch transformiert, um Extreme zu glätten, da die Werte über mehrere Größenordnungen variieren.

## 4. Skalierung:

Anschließend wurden die Daten skaliert, sodass sie Werte im Bereich  $[0, 1]$  liegen. Dies erleichtert das Training neuronaler Netze, da diese empfindlich auf unterschiedliche Wertebereiche der Eingabegrößen reagieren.

## 5. Entfernen von Ausreißern:

Dieser Schritt ist bei der Datenaufbereitung grundsätzlich wichtig. Für die in dieser Arbeit betrachteten Fälle war dies jedoch nicht erforderlich, da die Daten nicht experimentell ermittelt sind, sondern aus Simulationen stammen und keine Ausreißer enthalten.

## 6. Aufteilung der Daten:

Die Daten wurden in folgende Datensätze unterteilt:

- Trainingsdatensatz: für das Modelltraining.
- Validierungsdatensatz: zur Überprüfung der Modellgüte während des Trainings und Optimierung der Modellparameter (Hyperparameter, Early Stopping, Regularisierung).
- Testdatensatz: zur abschließenden, unabhängigen Bewertung des Modells.

## 7. Datenaugmentation:

Datenaugmentation (*data augmentation*) ist eine Technik, die verwendet wird, um die Menge und Vielfalt von Trainingsdaten zu erhöhen, indem bestehende Daten modifiziert oder neue Varianten erstellt werden. Es gibt verschiedene Ansätze dazu.

In dieser Arbeit wurde folgender Ansatz verfolgt: Um die Datenmenge und -vielfalt zu erhöhen, wurde Rauschen zu den bestehenden Trainingsdaten hinzugefügt. Die Originaldaten blieben dabei unverändert und wurden durch die hinzugefügten augmentierten Daten ergänzt. Dieser Ansatz eignet sich besonders bei kleinen Datensätzen und erhöht die Robustheit des Modells. Zusätzlich reduziert er das Risiko von *Überanpassung* (*Overfitting*), da das Modell auf variierte Eingaben trainiert wird.

## 2.5 Leistungsmetriken und Evaluierungsmethoden

Die Leistungsbeurteilung der Modelle erfolgt mit Hilfe quantitativer Metriken, qualitativer visueller Analysen und durch Vergleiche mit sogenannten *Baseline*-Modellen. Beim Letzteren handelt es sich um einfache Vorhersagemodelle um die Leistung der viel komplexeren KNN-Modelle einzuordnen. In dieser Studie dient als Baseline die lineare Interpolation, eine etablierte Methode zur Berechnung von Kritikalitätsparametern. Neben den numerischen Maßzahlen, die eine objektive Bewertung der Modellgüte ermöglichen, bieten Visualisierungen wertvolle Einblicke in die Verteilung und das Verhalten der Vorhersagen im Vergleich zu den tatsächlichen Werten.

Diese Ansätze ergänzen sich, um eine umfassende Beurteilung der Modellleistung sicherzustellen. Im Folgenden werden die verwendeten Metriken und Visualisierungsmethoden detailliert erläutert.

### 2.5.1 Metriken zur Modellbewertung

In diesem Abschnitt werden die wichtigsten Metriken zur Bewertung von Regressionsmodellen vorgestellt:

- MSE (*Mean Squared Error*, mittlerer quadratischer Fehler)
- MAE (*Mean Absolute Error*, mittlerer absoluter Fehler)
- RMSRE (*Root Mean Square Relative Error*, relativer MSE)

- $R^2$  (Bestimmtheitsmaß der (oft linearen) Regression)

Jede Metrik bietet spezifische Einblicke in die Modellleistung. Die Verwendung mehrerer Metriken ermöglicht eine umfassende Beurteilung, da verschiedene Aspekte der Modellleistung beleuchtet werden. Die Kombination der Metriken hilft, ein ausgewogenes und robustes Modell zu identifizieren.

In dieser Arbeit dient der MSE als Verlustfunktion im Trainingsprozess, während der MAE als ergänzende Metrik auf den Validierungsdaten eingesetzt wird.  $R^2$  und RMSRE werden ausschließlich in der Testphase zur Modellevaluierung eingesetzt.

### **MSE (Mean Squared Error)**

Der MSE misst die durchschnittliche quadratische Differenz zwischen den tatsächlichen und den vorhergesagten Werten. Er gewichtet größere Fehler stärker durch die Quadrierung, was ihn besonders nützlich macht, wenn größere Abweichungen eine höhere Relevanz haben. Der MSE wird wie folgt berechnet:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (2.2)$$

$n$  Anzahl der Datenpunkte

$\hat{y}_i$  vorhergesagter Wert für das  $i$ -te Datenpunkt

$y_i$  tatsächlicher Wert für das  $i$ -te Datenpunkt

Der MSE eignet sich für Situationen, in denen größere Fehler verstärkt berücksichtigt werden sollen, beispielsweise bei sicherheitskritischen Anwendungen oder bei Vorhersagen mit hohem Risikopotenzial. Die Quadrierung macht den MSE jedoch empfindlich gegenüber *Ausreißern* – Datenpunkten, die deutlich vom allgemeinen Muster abweichen. Dies kann die Modellbewertung verzerren und zu überproportionalem Einfluss dieser Werte führen. Im Allgemeinen ist ein möglichst kleiner MSE-Wert anzustreben, wobei 0 perfekte Vorhersagen bedeuten würde. Bei skalierten Daten gelten Werte unter 0,01 allgemein als gut und unter 0,001 als exzellent.

### MAE (Mean Absolute Error)

Der MAE misst die durchschnittliche absolute Differenz zwischen den tatsächlichen und den vorhergesagten Werten. Im Vergleich zum MSE bestraft der MAE größere Fehler geringer, da er den Fehler nur linear anstelle quadratisch berücksichtigt. Er wird wie folgt berechnet:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (2.3)$$

Der MAE ist besonders geeignet, wenn alle Abweichungen gleich gewichtet werden sollen und einzelne große Fehler (beispielweise Ausreißer) die Gesamtbewertung des Modells nicht überproportional beeinflussen sollen. Der MAE ist weniger empfindlich gegenüber Ausreißern als der MSE, was je nach Anwendungsfall vorteilhaft oder nachteilig sein kann. In Szenarien, in denen größere Fehler mehr Gewicht haben sollten, kann der MAE zu einer verzerrten Modellbewertung führen. Ein MAE von 0 bedeutet perfekte Vorhersagen. Bei skalierten Werten zwischen 0 und 1 gilt ein MAE-Wert unter 0,1 allgemein als gut, während Werte unter 0,005 als exzellent angesehen werden. Die Interpretation hängt jedoch stark vom spezifischen Anwendungsfall ab.

### RMSRE (Root Mean Square Relative Error)

RMSRE misst die durchschnittliche Größe des relativen Fehlers, wobei die Fehler relativ zu den tatsächlichen Werten berechnet werden:

$$RMSRE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{\hat{y}_i - y_i}{y_i + \varepsilon} \right)^2} \quad (2.4)$$

Dabei ist  $\varepsilon$  eine kleine Konstante, die hinzugefügt wird, um Divisionen durch Null zu vermeiden. RMSRE eignet sich besonders für Situationen, in denen die relative Genauigkeit wichtig ist. Die Metrik ist robust gegenüber unterschiedlichen Größenordnungen der Zielwerte. Der RMSRE kann jedoch bei sehr kleinen tatsächlichen Werten Probleme aufweisen. Selbst geringe absolute Abweichungen können bei kleinen Zielwerten zu großen relativen Fehlern führen, was den RMSRE überproportional erhöht. Dadurch besteht die Gefahr, dass gewöhnliches Datenrauschen fälschlicherweise als signifikanter Fehler interpretiert wird. Bei normierten Werten zwischen 0 und 1 ist dieses Problem zwar weniger ausgeprägt, bleibt aber für Werte nahe Null relevant. Daher ist bei der Interpretation des RMSRE, insbesondere in Datensätzen mit zahlreichen kleinen Zielwerten,

besondere Vorsicht geboten. Ein RMSRE-Wert von 0 bedeutet perfekte Vorhersagen. Werte unter 0,1 gelten als exzellent, während Werte über 0,5 auf Verbesserungspotenzial hindeuten. Der akzeptable RMSRE-Bereich variiert je nach Anwendungsgebiet. In physikalischen und technischen Anwendungen wird oft eine sehr hohe Genauigkeit ( $\text{RMSRE} < 0,1$ ) angestrebt, da selbst kleine Abweichungen signifikante praktische Auswirkungen haben können.

### **R<sup>2</sup> (Bestimmtheitsmaß)**

R<sup>2</sup> gibt an, wie gut das Modell die Varianz der tatsächlichen Werte erklärt. Ein hoher R<sup>2</sup>-Wert bedeutet, dass das Modell die Variabilität der Zielvariablen gut erfasst und signifikant besser abschneidet als ein einfaches Mittelwert-Modell. Das Bestimmtheitsmaß wird wie folgt berechnet:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (2.5)$$

wobei  $\bar{y}_i$  der Mittelwert der tatsächlichen Werte ist. R<sup>2</sup> wird verwendet, um zu bewerten, wie gut ein Modell die Zielvariablen erklärt, verglichen mit einem einfachen Mittelwert-Modell. Ein hoher R<sup>2</sup>-Wert zeigt, dass das Modell die zugrunde liegenden Beziehungen gut erkannt hat. In dieser Arbeit wird R<sup>2</sup> in der Testphase bei der Modellevaluierung berechnet. Das R<sup>2</sup>-Maß unterliegt jedoch einigen Einschränkungen. So kann der R<sup>2</sup>-Wert irreführend sein, wenn das Modell viele unwichtige Variablen – Variablen, die zwar existieren, aber nicht zur Erklärung beitragen – enthält, da der Wert mit jeder zusätzlichen Variable tendenziell steigt. Auch bedeutet ein hohes R<sup>2</sup> nicht automatisch ein gutes Modell, besonders wenn es durch Überanpassung zustande kommt, d. h., wenn das Modell die Trainingsdaten „auswendig gelernt hat“ und dadurch schlecht auf neue Daten generalisiert. Andererseits deutet ein niedriger R<sup>2</sup>-Wert möglicherweise darauf hin, dass wesentliche erklärende Variablen fehlen oder die zugrunde liegenden Beziehungen im Modell nicht angemessen erfasst werden. R<sup>2</sup> ist insbesondere für lineare Zusammenhänge zwischen den Variablen sinnvoll interpretierbar. Für nichtlineare Modelle können andere Metriken aussagekräftiger sein. Ein R<sup>2</sup>-Wert von 1 zeigt eine perfekte Anpassung an. Werte unter 0,5 weisen darauf hin, dass das Modell möglicherweise verbessert werden sollte. In der Praxis hängt die Interpretation stark vom spezifischen Anwendungsfall ab.

## 2.5.2 Visualisierungen zur Modellevaluierung

Die Evaluierung der Modelleistung erfolgt zusätzlich durch verschiedene grafische Darstellungen (Plots). Diese Visualisierungen, basierend auf ausgewählten Testdaten, dienen dazu, Ergebnisse anschaulich darzustellen und systematische Muster oder Schwächen der Modelle zu identifizieren. Sie bieten eine fundierte Grundlage für gezielte Modellverbesserungen. Jeder Plot analysiert dabei spezifische Aspekte der Modellperformance und trägt so zu einer umfassenden Evaluierung bei.

### Vergleichsplots der Kritikalitätsparameter

Vergleichsplots der Kritikalitätsparameter in Abhängigkeit von der Urankonzentration zeigen sowohl die tatsächlichen als auch die vorhergesagten Werte der Kritikalitätsparameter als Funktion der Urankonzentration. Diese Darstellung ist besonders hilfreich, um die Modelleistung visuell zu bewerten und mögliche konzentrationsabhängige Abweichungen zu identifizieren.

Bei den hier verwendeten Vergleichsplots ist auf der X-Achse die Urankonzentration und auf der Y-Achse der jeweilige Kritikalitätsparameter aufgetragen. Die doppelt logarithmische Skalierung der Achsen für die Hauptzielvariablen ermöglicht eine anschauliche Darstellung von Zusammenhängen über mehrere Größenordnungen hinweg und reduziert den Einfluss extremer Werte. Dadurch lassen sich Vorhersagefehler auch in Bereichen mit niedrigen oder hohen Konzentrationen besser analysieren. Für  $k_{inf}$  wurde eine einfach logarithmische Skalierung gewählt, da der Wertebereich eine deutlich geringere Spanne aufweist.

### Scatterplot: tatsächliche vs. vorhergesagte Werte

Streudiagramme (Scatterplots) vergleichen die tatsächlichen Werte der Kritikalitätsparameter (X-Achse) mit den vorhergesagten Werten (Y-Achse). Diese Darstellungen helfen, systematische Fehler, Ausreißer und die generelle Modellgüte zu bewerten. Ein zentrales Element des Scatterplots ist die diagonale *Referenzlinie* ( $y = x$ ), die eine perfekte Übereinstimmung zwischen Vorhersage und tatsächlichen Werten darstellt. Je näher die Datenpunkte an dieser Linie liegen, desto genauer sind die Modellvorhersagen. Größere Abweichungen deuten auf systematische Fehler oder eine begrenzte Modellgenauigkeit hin. Darüber hinaus erlaubt der Scatterplot eine differenzierte Analyse der Modellfehler. Er zeigt ob Modellungenauigkeiten, die mit integralen Leistungsmetriken (z. B. RMSRE)

gemessen wurden, auf wenige Punkte (wie z. B. Ausreißer) zurückzuführen sind oder ob größere systematische Probleme vorliegen.

### **Vergleichsplots mit Baseline-Modell: relative Abweichungen**

Vergleichsplots mit dem Baseline-Modell zeigen die relativen Abweichungen der Modellvorhersagen und der Baseline-Methode von den tatsächlichen Werten. Der relative Fehler, definiert als der Unterschied zwischen den vorhergesagten und den tatsächlichen, mit SCALE direkt berechneten, Werten normiert auf den tatsächlichen Wert, wird für beide Methoden dargestellt. Positive Werte zeigen eine Überschätzung, negative eine Unterschätzung an. Die Darstellung der relativen Fehler ermöglicht folgende Analysen:

- Einen direkten Vergleich der Modelleistung mit der Baseline-Methode.
- Eine Bewertung, in welchen Bereichen das Modell signifikante Verbesserungen bietet.
- Eine Identifikation von Schwachstellen der KNN-Modelle, in denen die Baseline-Methode bessere Ergebnisse liefert.

Dieser Vergleich ist entscheidend, um den Mehrwert des neuronalen Netzes gegenüber der linearen Interpolation zu quantifizieren und gezielte Optimierungen vorzunehmen.



### 3 Ergebnisse und Analyse

Die Arbeiten wurden in drei aufeinander aufbauende Phasen gegliedert, um die zunehmende Komplexität der Problematik schrittweise anzugehen:

- **Phase 1** (Kapitel 3.1):

Im ersten Schritt wurde das im Rahmen der Machbarkeitsstudie /EIS 22/ entwickelte Modell weiter untersucht und getestet. Ergänzend dazu wurde ein MLP-Modell als Alternativmodell mit einfacherer Architektur entwickelt, um die Vorhersagekraft und Praxistauglichkeit unterschiedlicher Ansätze systematisch zu vergleichen.

- **Phase 2** (Kapitel 3.2):

In der zweiten Phase wurde die Menge der Zielvariablen um den infiniten Multiplikationsfaktor  $k_{inf}$  erweitert. Diese Variable weist eine grundlegend andere Kurvenform in ihrer Abhängigkeit von der Urkonzentration auf als die bisher betrachteten Kritikalitätsparameter, was die Modellkomplexität erheblich erhöhte.

Eine besondere Herausforderung bestand darin, zwei separate Datensätze – einen für die bisherigen Kritikalitätsparameter und einen für  $k_{inf}$  – so zu integrieren, dass sie von einem gemeinsamen KNN verarbeitet werden konnten. Dafür wurde ein entsprechendes MTL- und *MTL-SNN-LR*-Modell entwickelt.

- **Phase 3** (Kapitel 3.3):

In der dritten Phase wurde die Methodik auf ein weiteres chemisches System übertragen. Dies führte zu einer zusätzlichen Komplexitätssteigerung durch die Einführung eines neuen Eingabeparameters und der Entwicklung eines TMLP-Modells.

Die Ergebnisse und Erkenntnisse aus diesen Phasen werden in den folgenden Kapiteln detailliert beschrieben und analysiert.

#### 3.1 Phase 1: Entwicklung und Analyse von Basismodellen

Die Zielsetzung der Machbarkeitsstudie /EIS 22/ bestand darin, die grundsätzliche Eignung von KNN-Modellen zur Vorhersage von Kritikalitätsparametern zu evaluieren. Dabei wurde ein erster Prototyp, das beschriebene Modell SNN-LR, entwickelt und initial getestet. Mit dem Ziel die bisherigen Analysen im Anschluss mit weiteren KNN-Modellen

zu vergleichen, wurde eine detaillierte Analyse des bestehenden SNN-LR-Modells mit den folgenden Zielen durchgeführt:

- Validierung der Ergebnisse der Machbarkeitsstudie, um die Belastbarkeit der gewonnenen Erkenntnisse sicherzustellen.
- Identifikation von Optimierungspotenzial des Modells hinsichtlich Effizienz und Genauigkeit.
- Entwicklung eines tiefgreifenden Verständnisses der bestehenden Modellarchitektur, insbesondere im Hinblick auf zukünftige Modellerweiterungen.

Da die SNN-LR-Architektur sehr komplex ist, wurde beschlossen, ein KNN-Modell mit einfacherer Struktur zu entwickeln und zu untersuchen. Ein einfacheres Modell bietet eine größere Flexibilität für Weiterentwicklungen und erleichtert die Anpassung an komplexere Problemstellungen im Kontext der Kritikalitätsparameter.

Für KNN-Modelle in diesem Anwendungsbereich existieren bisher keine etablierten Standards oder Richtlinien. Daher war es im Vorfeld nicht möglich, eine optimale Modellarchitektur und/oder geeignete Hyperparameter festzulegen. Die durchgeführten Analysen bilden somit die Grundlage für die zielgerichtete Weiterentwicklung der KNN-Modelle entsprechend der Arbeitsziele und gewährleisten eine optimale Abstimmung auf die spezifischen Anforderungen der Kritikalitätsparametervorhersage.

### **3.1.1 Modellierungsansatz**

Im Rahmen der ersten Phase wurde als einfachere Alternative zum komplexeren SNN-LR-Modell ein MLP-Modell entwickelt (siehe Kapitel 2.3.1 und 2.3.2). Während das SNN-LR-Modell ein Ensemble aus mehreren KNN repräsentiert, handelt es sich beim MLP-Modell um ein einzelnes, sequenziell aufgebautes neuronales Netz. Diese Architektur ermöglicht eine direkte und effiziente Verarbeitung der Eingabedaten. Nach umfassenden empirischen Untersuchungen wurden für beide Modelle optimale Architekturen und Hyperparameter ermittelt, die zu einer guten Modelleistung führten.

Die Modelle wurden mit jeweils zwei Eingabeparametern trainiert:

- Urankonzentration
- <sup>235</sup>U-Anreicherung

Die Zielvariablen des Modells umfassen vier Hauptkritikalitätsparameter (Hauptzielvariablen, siehe Kapitel 2.4.2):

- kritische Kugelmasse ( $M_k$ )
- kritisches Kugelvolumen ( $V_k$ )
- kritischer Zylinderdurchmesser ( $D_k$ )
- kritische Schichtdicke ( $S_k$ )

Weitere Details zu den Modellarchitekturen und -konfigurationen finden sich in den jeweiligen Kapiteln zum SNN-LR (Kapitel 2.3.1) und zum MLP (Kapitel 2.3.2).

### 3.1.2 Datengrundlage

Die Entwicklung und Evaluierung der Modelle basierte auf einem Datensatz für homogene Ammoniumdiurandat(ADU)-Wasser-Systeme, siehe Abb. 2.3 und Referenz /SOM 19/. Diese Systeme umfassen:

- $^{235}\text{U}$ -Anreicherungen im Bereich von 2 Gew% bis 100 Gew%
- Urankonzentrationen zwischen etwa  $10^{-2}$  gU/cm<sup>3</sup> und 4 gU/cm<sup>3</sup>

Die Datenpunkte der kritischen Parameterkurven mit einer  $^{235}\text{U}$ -Anreicherung von 5 % und 50 % wurde als unabhängiger Testdatensatz zurückgehalten, um die Vorhersagekraft des Modells zu testen. Die Auswahl dieser Testdaten wurde so getroffen, dass es einem möglichen Anwendungsfall entspricht bei dem z. B. die kritischen Parameterkurven für weitere Anreicherung ermittelt werden sollen. Zudem entsprechen die 5 %  $^{235}\text{U}$ -Anreicherung einer mittleren Anreicherung, während andererseits die 50 %  $^{235}\text{U}$ -Anreicherung einem Grenzfall zuzuordnen ist und somit ein Hätetest für das Modell darstellt. Die verbleibenden Daten wurden für das Training der Modelle verwendet. Während beim komplexeren SNN-LR-Modell eine Aufteilung diesen Datensatzes in Trainings- und Validierungsdatensatz erforderlich war, wurde das einfachere MLP-Modell nach der Optimierung der Hyperparameter abschließend auf dem gesamten Trainingsdatensatz trainiert.

### 3.1.3 Ergebnisse

Nach Abschluss des Trainings wurden beide Modelle mit den unabhängigen Testdaten (5 % und 50 %  $^{235}\text{U}$ -Anreicherung) analysiert und bewertet. Die dabei verwendeten Evaluierungsmethoden sind in Kapitel 2.5 ausführlich beschrieben.

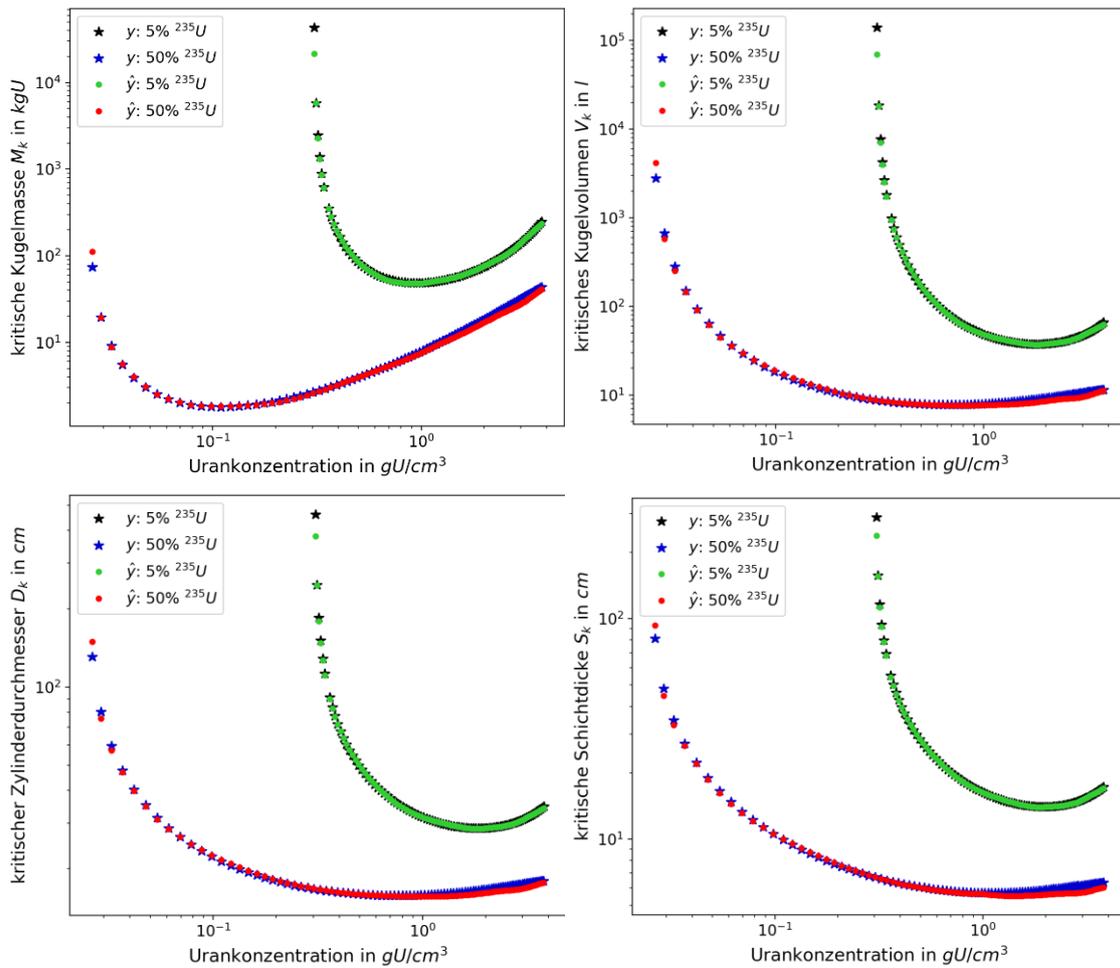
In Abb. 3.1 und Abb. 3.2 sind die Vorhersagen  $\hat{y}$  durch die SNN-LR- und MLP-Modelle für die kritischen Parameterkurven mit 5 % und 50 %  $^{235}\text{U}$ -Anreicherung sowie die tatsächlichen Werte der Parameter  $y$  in Abhängigkeit der Urankonzentration dargestellt.

Der visuelle Vergleich beider Modelle zeigt bei mittleren Urankonzentrationen eine gute Übereinstimmung der Vorhersagen mit den tatsächlichen Werten. Größere Abweichungen treten vor allem bei niedrigen und hohen Urankonzentrationen auf. Aus den Plots ist zu erkennen, dass die Abweichungen des MLP-Modells vor allem bei niedrigen Urankonzentrationen etwas geringer sind.

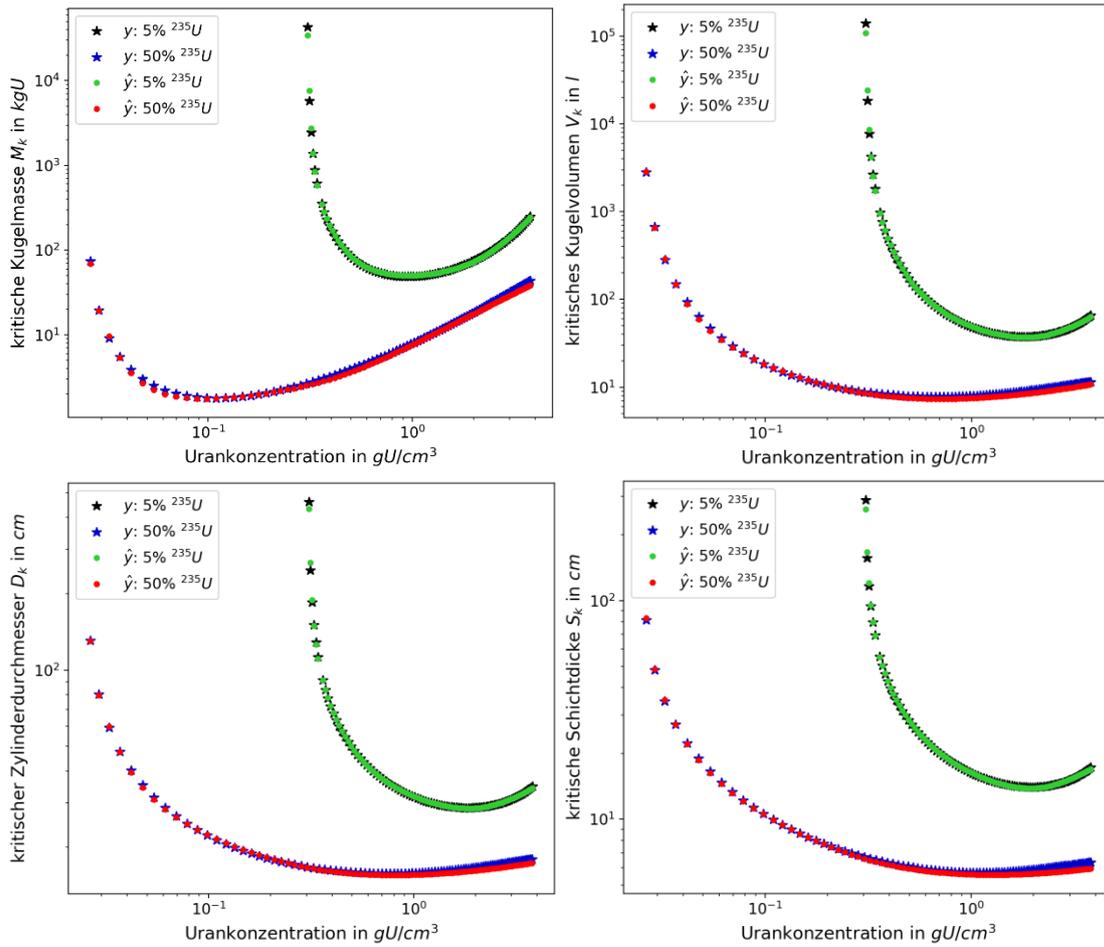
Die Abweichungen lassen sich vor allem auf zwei Faktoren zurückführen:

1. Ausgeprägte Nichtlinearitäten der kritischen Parameterkurven vor allem bei niedrigen Urankonzentrationen: In solch hochgradig nichtlinearen Bereichen ist die präzise Modellierung mathematischer Zusammenhänge besonders herausfordernd.
2. Eine geringere Repräsentation dieser Bereiche im Trainingsdatensatz, insbesondere bei kleinen Urankonzentrationen.

Die Abweichungen betreffen vor allem die kritische Kugelmasse und das kritische Kugelvolumen. Bei der optischen Bewertung der Plots muss der unterschiedliche Datenumfang der jeweiligen vertikalen Achse beachtet werden, der sich zwischen den kritischen Parametern um mehrere Größenordnungen unterscheidet.



**Abb. 3.1** Vorhersagen des SNN-LR-Modells für kritische Parameter im Vergleich zu tatsächlichen Werten



**Abb. 3.2** Vorhersagen des MLP-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten

Die Modellleistung wurde auch anhand der Metriken RMSRE und  $R^2$  bewertet. Diese sind in Tab. 3.1 zusammengefasst.

**Tab. 3.1** Phase 1: Vergleich der Metriken zwischen SNN-LR- und MLP-Modell für 5 % und 50 % Anreicherung

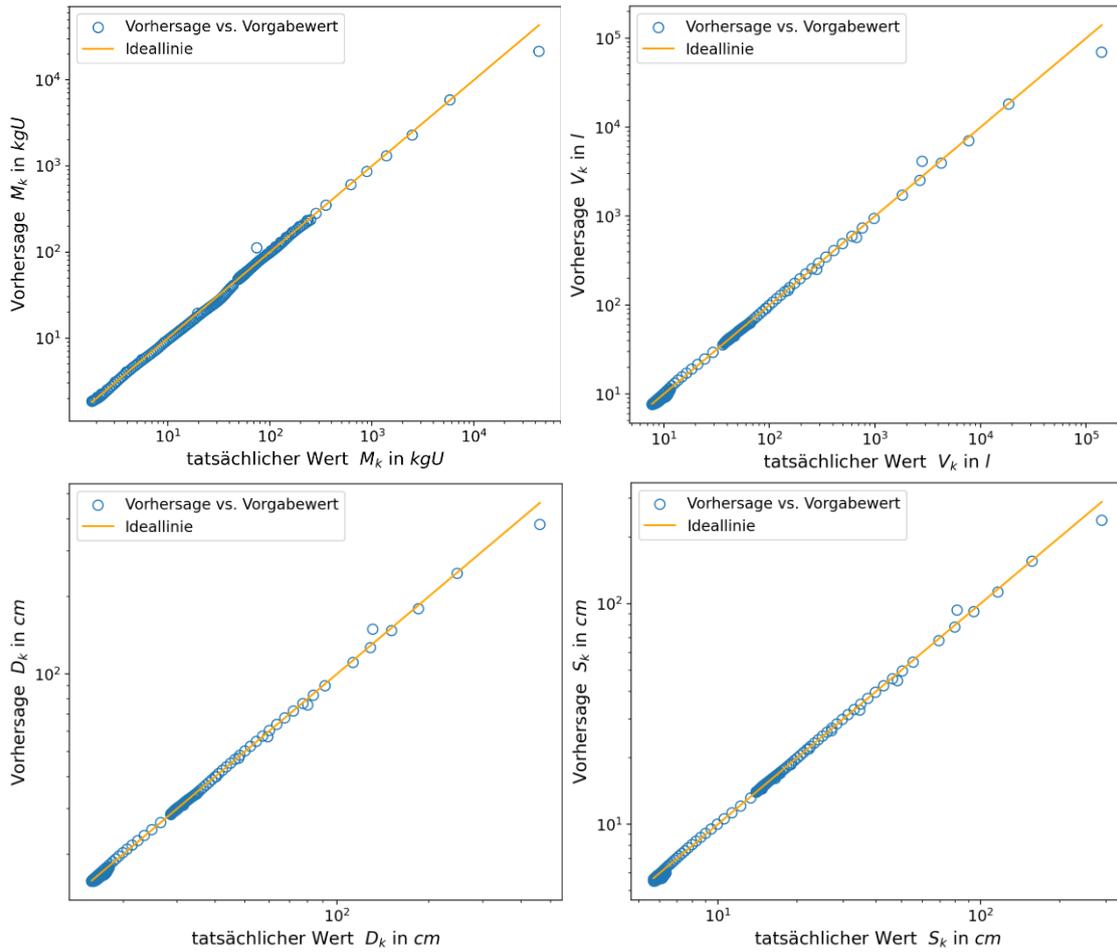
Anreicherung:	5 %				50 %			
Metrik:	RMSRE		$R^2$		RMSRE		$R^2$	
Modell:	SNN-LR	MLP	SNN-LR	MLP	SNN-LR	MLP	SNN-LR	MLP
$M_k$	5,6 %	4,6 %	0,746	0,957	8,4 %	6,5 %	0,903	0,984
$V_k$	5,7 %	4,3 %	0,747	0,947	7,7 %	5,8 %	0,781	1,0
$D_k$	2,0 %	1,1 %	0,976	0,996	2,7 %	2,3 %	0,982	0,999
$S_k$	2,0 %	1,3 %	0,977	0,993	4,2 %	3,5 %	0,981	0,999

Allgemein können folgende Erkenntnisse aus den Metriken gezogen werden:

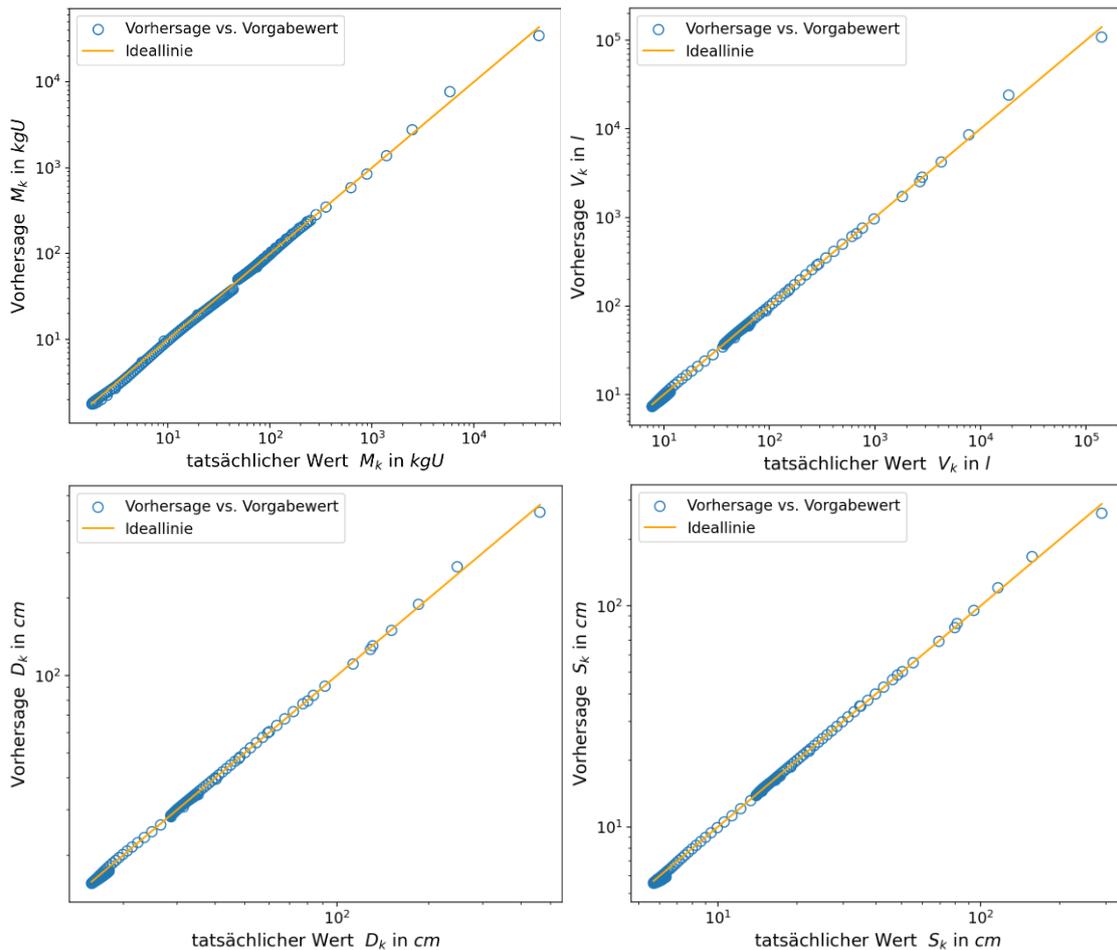
- **RMSRE:** Die Werte für Kugelmasse und Kugelvolumen sind etwas höher als für Zylinderdurchmesser und Schichtdicke, bleiben jedoch unter 10 % und sprechen somit für eine insgesamt gute Modellleistung.
- **$R^2$ :** Die Werte liegen durchgehend über 0,74, was einer guten Qualität der Modellvorhersagen entspricht.

Für alle vier Kritikalitätsparameter liefert das MLP-Modell sowohl für die Metrik RMSRE als auch für  $R^2$  bessere Werte als das SNN-LR-Modell. Dies bestätigt die Erkenntnis aus dem visuellen Vergleich, dass die Vorhersagekraft des MLP-Modells für diesen Anwendungsfall besser ist.

Abb. 3.3 und Abb. 3.4 zeigen den direkten Vergleich der vorhergesagten und tatsächlichen Werte (Erklärung der Plots, siehe Kapitel 2.5.2). Dabei zeigt sich wieder, dass bei hohen Parameterwerten größere Abweichungen von der Referenzlinie  $y = x$  auftreten. Diese hohen Werte korrespondieren mit niedrigen Urankonzentrationen und betreffen insbesondere Kugelmasse und Kugelvolumen. Dennoch sind Abweichungen auf wenige Datenpunkte beschränkt, was die robuste Leistung beider Modelle insbesondere bei mittleren Urankonzentrationen unterstreicht.



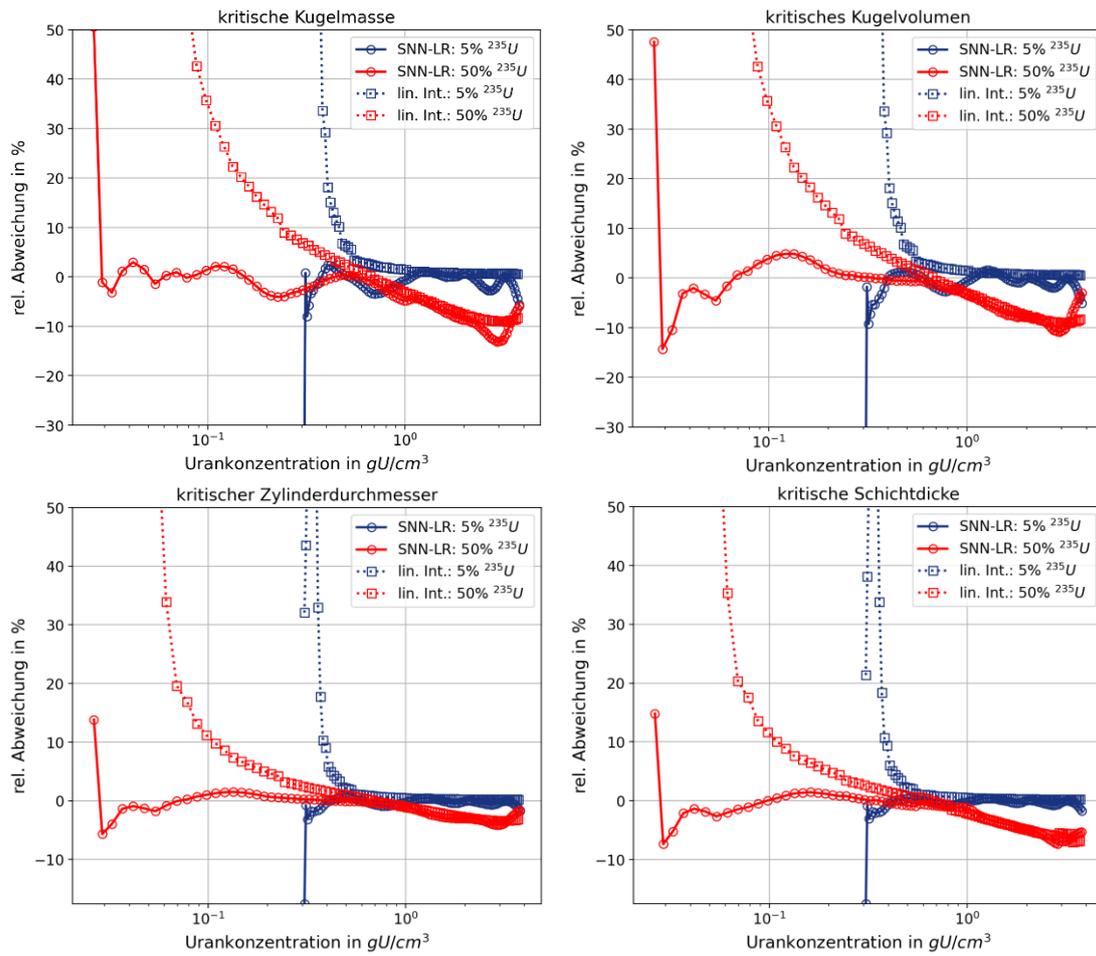
**Abb. 3.3** Vorhersagen des SNN-LR-Modells vs. tatsächlicher Werte mit Referenzlinie



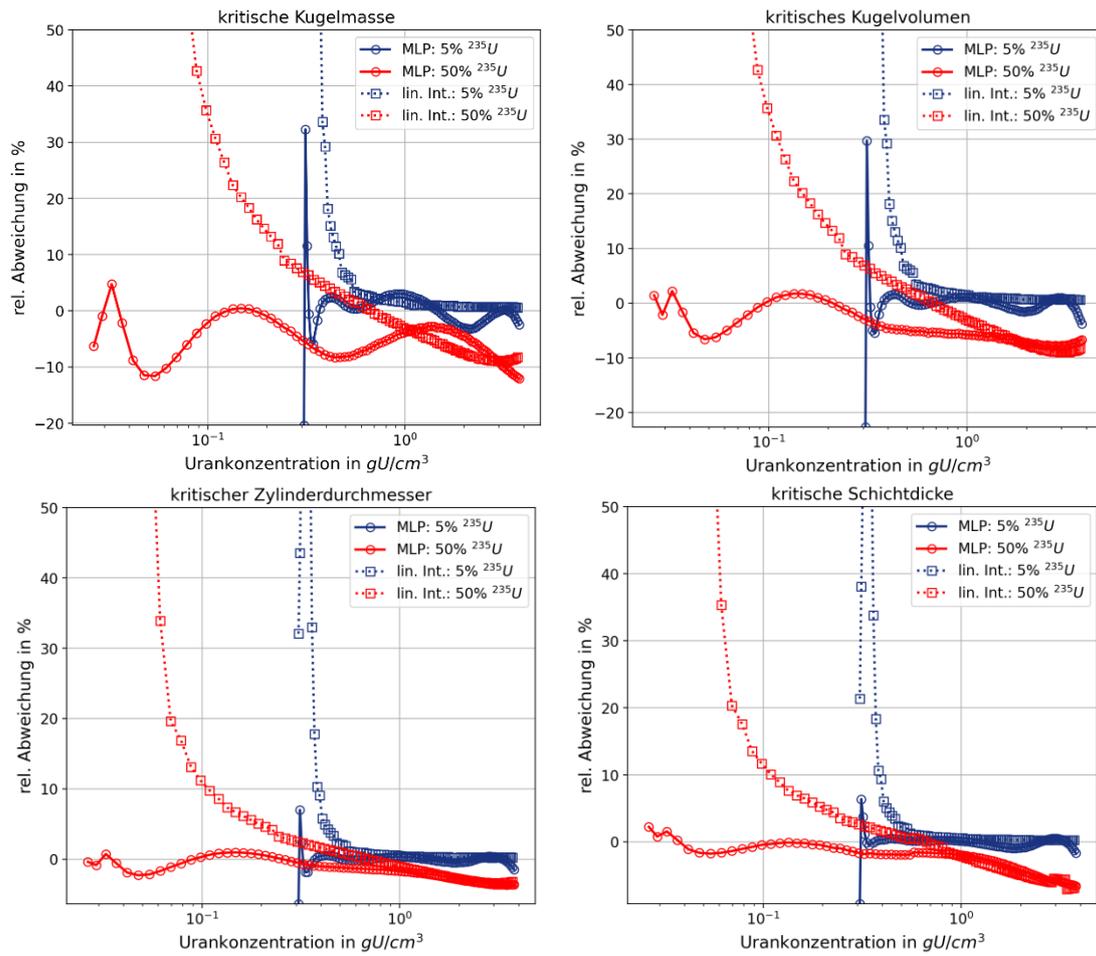
**Abb. 3.4** Vorhersagen des MLP-Modells vs. tatsächlicher Werte mit Referenzlinie

Um die Leistung der KNN-Modelle weiter zu evaluieren, wurde ein Vergleich mit einem gängigen Basismodell – der *linearen Interpolation* – durchgeführt. Die relativen Abweichungen zwischen den tatsächlichen und den vorhergesagten Werten sind für das SNN-LR-Modell in Abb. 3.5 und für das MLP-Modell in Abb. 3.6 dargestellt und werden dort jeweils mit den aufgrund der linearen Interpolation resultierenden relativen Abweichungen verglichen.

Dieser Vergleich zeigt, dass die KNN-Modelle insgesamt leistungsstärker als die lineare Interpolation sind. Obwohl die KNN-Modelle in stark nichtlinearen Bereichen weniger genaue Vorhersagen liefern als in linearen Bereichen, übertreffen sie dort die Vorhersagegenauigkeit der linearen Interpolation deutlich. In weniger komplexen, nahezu linearen Bereichen liefert die lineare Interpolation vergleichbare, bzw. teils bessere Ergebnisse.



**Abb. 3.5** Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: SNN-LR-Modell vs. lineare Interpolation



**Abb. 3.6** Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: MLP-Modell vs. lineare Interpolation

### 3.1.4 Fazit

Der Modellvergleich ergab insgesamt:

- Das **SNN-LR-Modell** zeigte sich robuster gegenüber Hyperparameter-Änderungen und bot eine stabilere Leistung, ist jedoch deutlich komplexer und erfordert längere Trainingszeiten.
- Das **MLP-Modell** zeichnet sich durch Einfachheit, bessere Verständlichkeit und eine kürzere Trainingszeit aus. Jedoch reagiert es empfindlich auf Änderungen der Hyperparameter, was den Optimierungsprozess (Hyperparameter-Tuning) verlängerte.

Beide Modelle eignen sich gut für die Aufgabenstellung und ergänzen sich durch ihre spezifischen Stärken, für die hier betrachteten Testfälle von 5 % und 50 % Anreicherung zeigte das MLP-Modell eine leicht bessere Leistung.

### 3.2 Phase 2: Integration eines neuen Kritikalitätsparameters

Nach der erfolgreichen Entwicklung von KNN-Modellen für vier Hauptkritikalitätsvariablen in Phase 1 wurde das Modell um den infiniten Multiplikationsfaktor  $k_{inf}$  als zusätzliche Zielvariable erweitert. Diese Erweiterung stellte aus zwei Gründen eine erhebliche Herausforderung dar:

1. **Abweichende Kurvenform:**  $k_{inf}$  zeigt im Vergleich zu den Hauptkritikalitätsvariablen eine grundlegend andere Abhängigkeit von der Urankonzentration (vgl. Abb. 2.3 mit Abb. 2.5). Dadurch wurde die Modellkomplexität deutlich erhöht, da verschiedene Muster und Abhängigkeiten gleichzeitig erfasst werden mussten.
2. **Unterschiedliche Datenbasis:** Die vorliegenden Urankonzentrationswerte für  $k_{inf}$  stimmen nicht mit denen der Hauptkritikalitätsvariablen überein. Dies erforderte die Entwicklung auf zwei separaten Datensätzen. Eine einfache Modellerweiterung durch das Hinzufügen eines zusätzlichen Neurons zur Ausgabeschicht ist daher ausgeschlossen. Stattdessen musste eine innovative Lösung entwickelt werden, um diese Heterogenität zu bewältigen.

Die Integration von  $k_{inf}$  führte zu einer signifikanten Erhöhung der Systemkomplexität.

### 3.2.1 Modellierungsansatz

Um die zusätzliche Zielvariable  $k_{inf}$  in einem KNN-Modell zu integrieren, wurde ein *Multi-Task Learning (MTL)* Modell entwickelt. Dieses Modell umfasst zwei Eingabe- und zwei Ausgabeschichten. Diese Architektur ermöglicht die Verarbeitung zweier verschiedener Datenquellen und die gleichzeitige Vorhersage aller fünf Zielvariablen. Details zur Modellarchitektur und -konfiguration sind in Kapitel 2.3.3 beschrieben.

Als Alternative zum MTL-Modell wurde anschließend ein *Multi-Task Learning Stacked Neural Networks with Linear Regressor (MTL-SNN-LR)* Modell entwickelt. Dieses Modell kombiniert die SNN-LR-Architektur mit dem MTL-Ansatz als Basiskomponente. Hierbei werden jedoch wesentlich einfachere Modellarchitekturen für die Basis-MTL-Modelle eingesetzt, als es beim reinen MTL-Modell der Fall ist. Ziel war es, die Stärken des Stacking-Verfahrens mit den Vorzügen des Multi-Task Learnings zu vereinen. Dadurch sollten sowohl die Vorhersagegenauigkeit als auch die Flexibilität bei der Integration verschiedener Zielvariablen optimiert werden. Eine umfassende Beschreibung des MTL-SNN-LR-Modells befindet sich in Kapitel 2.3.4.

Eingabeparameter für beide Modelle sind jeweils:

- Urankonzentration
- $^{235}\text{U}$ -Anreicherung

Die Zielvariablen der Modelle umfassen insgesamt fünf Kritikalitätsparameter:

- vier Hauptzielvariablen (Hauptkritikalitätsparameter)
- eine abweichende Zielvariable ( $k_{inf}$ )

### 3.2.2 Datengrundlage

Die für die Modellentwicklung verwendete Datengrundlage basiert auf den in Phase 1 beschriebenen Methoden. Es kamen dabei zwei separate Datensätze zum Einsatz:

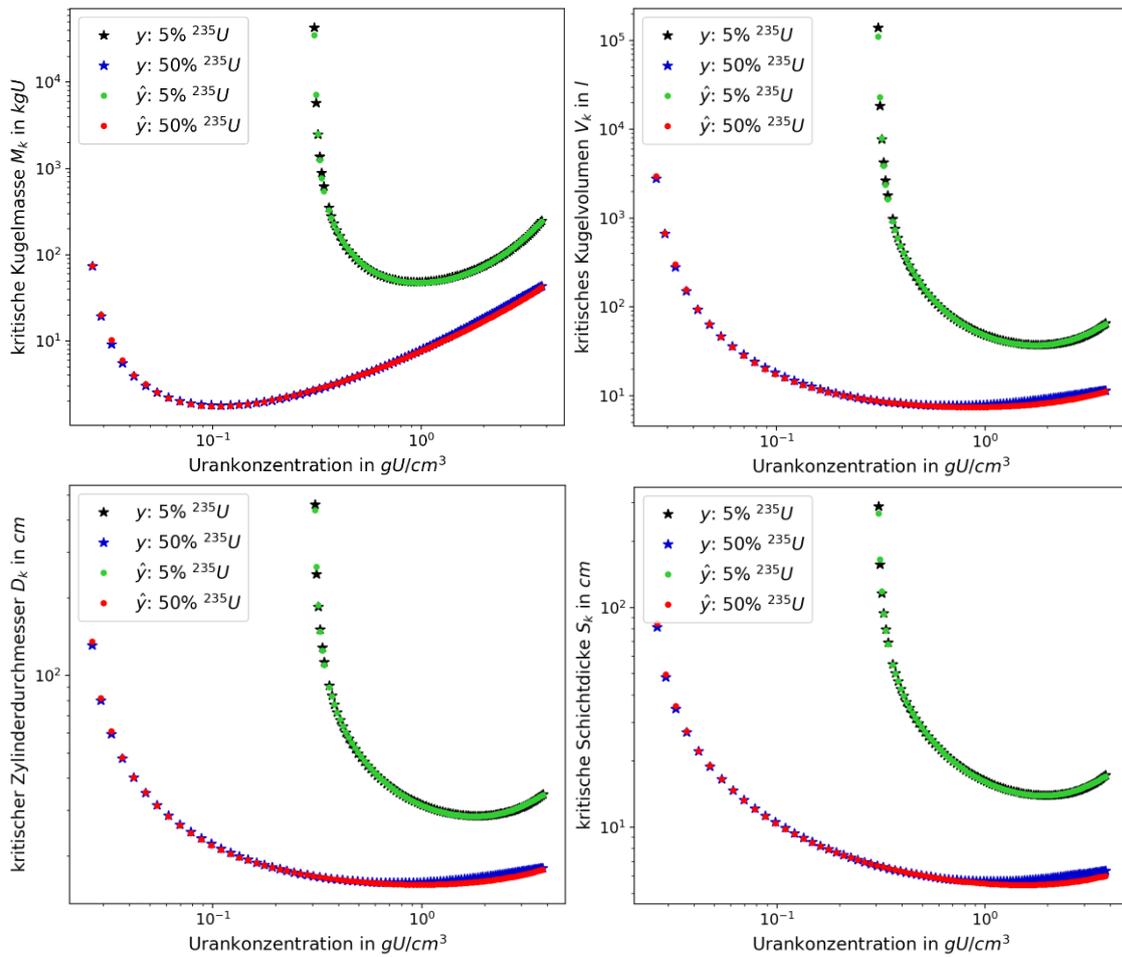
- Datensatz für die vier Hauptkritikalitätsvariablen
- Datensatz für  $k_{inf}$

Zum Testen wurden auch in Phase 2 die Datenpunkte der Kurven mit einer Anreicherung von 5 % und 50 %  $^{235}\text{U}$  verwendet.

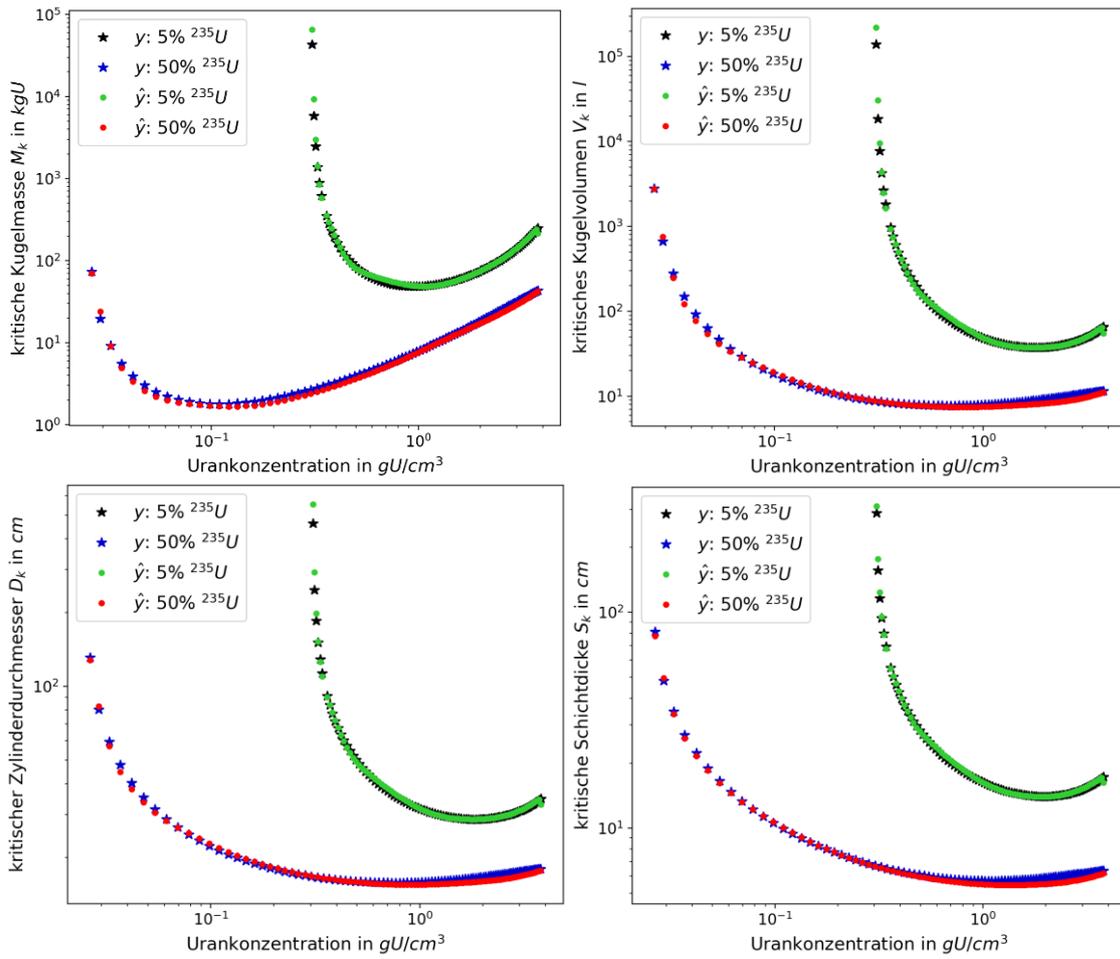
### 3.2.3 Ergebnisse

Die Ergebnisse der Modellevaluierung weisen klare Parallelen zu den in Phase 1 gewonnenen Erkenntnissen auf, zeigen jedoch spezifische Besonderheiten in Bezug auf die Zielvariable  $k_{inf}$ .

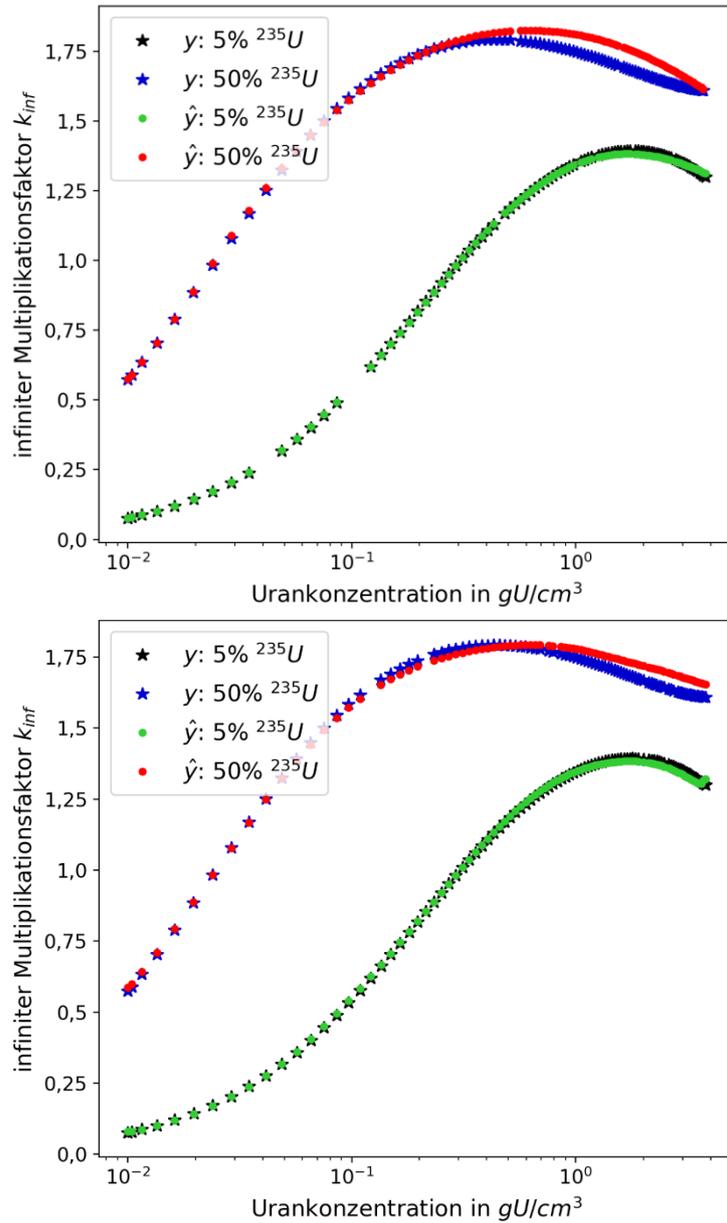
In Abb. 3.7 und Abb. 3.8 sind die Vorhersagen für die vier Haupt-Kritikalitätsparameter durch die MTL- bzw. MTL-SNN-LR-Modelle im Vergleich zu den tatsächlichen Werten in Abhängigkeit von der Urankonzentration dargestellt. Der visuelle Vergleich beider Modelle zeigt eine hohe Übereinstimmung bei mittleren Urankonzentrationen. Größere Abweichungen treten vor allem bei niedrigen und hohen Urankonzentrationen auf. Die für Phase 1 in Kapitel 3.1.3 erläuterten Gründe hierfür gelten auch für Phase 2.



**Abb. 3.7** Vorhersagen des MTL-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten



**Abb. 3.8** Vorhersagen des MTL-SNN-LR-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten



**Abb. 3.9** Vorhersagen des MLT und des MTL-SNN-LR-Modells für  $k_{inf}$  im Vergleich zu den tatsächlichen Werten

Beim Vergleich von  $k_{inf}$  ist der deutlich reduzierte Wertebereich und die lineare Skala der vertikalen Achse im Gegensatz zu den anderen Kritikalitätsparametern in Abb. 3.7 und Abb. 3.8 zu beachten.

Die Vergleiche zwischen den Ergebnissen der KNN-Modelle mit den tatsächlichen Werten von  $k_{inf}$  sind für beide Modelle in Abb. 3.9 gezeigt. Für  $k_{inf}$  ist zu beachten, dass der Wertebereich einen deutlich geringeren Umfang ausweist und Nichtlinearitäten deutlich geringer ausgeprägt sind. Daher ist in diesen Plots die vertikale Achse linear, nicht logarithmisch dargestellt. Dies ist auch bei der optischen Beurteilung der Vorhersage-

kraft zu berücksichtigen. Die beobachteten Abweichungen liegen zwar im niedrigen einstelligen Bereich, jedoch hat  $k_{inf}$  eine besonders sicherheitsspezifische Bedeutung, so dass bereits kleine Reaktivitätsunterschiede zwischen zwei Werten ( $\Delta\rho = 1/k_{inf}^1 - 1/k_{inf}^2$ ) in der Größenordnung von einigen hundert pcm als relevant gelten. Da beide Modelle für hohe Urankonzentrationen weitaus größere Abweichungen aufweisen, müssen diese Modelle in weiterführenden Arbeiten optimiert werden, um diesem Anspruch gerecht zu werden.

Die Leistungsbewertung erfolgte auch mittels der Metriken RMSRE und  $R^2$  und sind in Tab. 3.2 zusammengefasst.

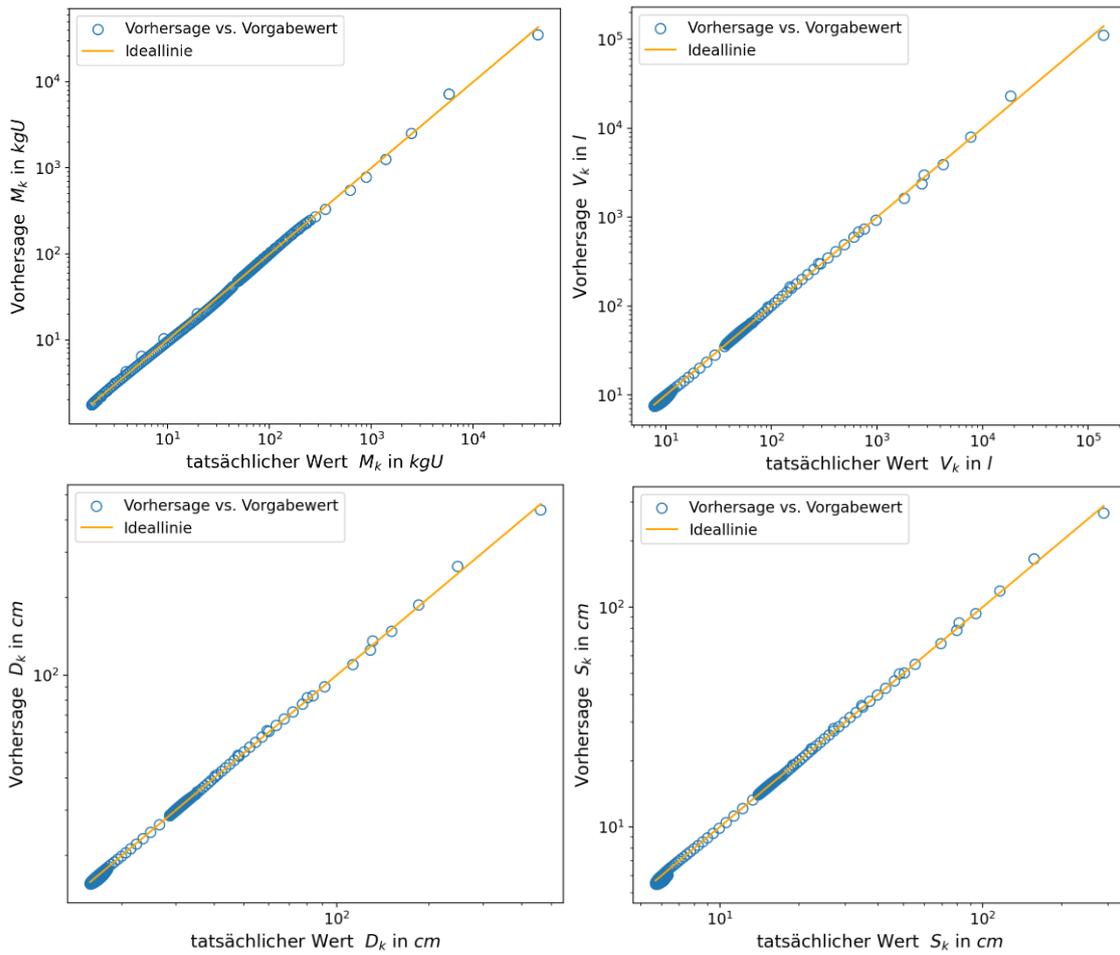
**Tab. 3.2** Phase 2: Vergleich der Metriken zwischen MTL- und MTL-SNN-LR-Modell für 5 % und 50 % Anreicherung

Anreicherung:	5 %				50 %			
Metrik:	RMSRE		$R^2$		RMSRE		$R^2$	
Modell:	MTL	MTL-SNN-LR	MTL	MTL-SNN-LR	MTL	MTL-SNN-LR	MTL	MTL-SNN-LR
$M_k$	4,2 %	9,1 %	0,964	0,728	5,4 %	8,6 %	0,991	0,987
$V_k$	4,1 %	10,0 %	0,955	0,667	6,6 %	9,3 %	0,996	0,97
$D_k$	1,1 %	3,0 %	0,997	0,962	2,6 %	3,2 %	0,998	0,994
$S_k$	1,2 %	2,0 %	0,995	0,992	4,2 %	4,3 %	0,998	0,997
$k_{inf}$	0,5 %	0,8 %	1,0	1,0	2,6 %	2,2 %	0,973	0,984

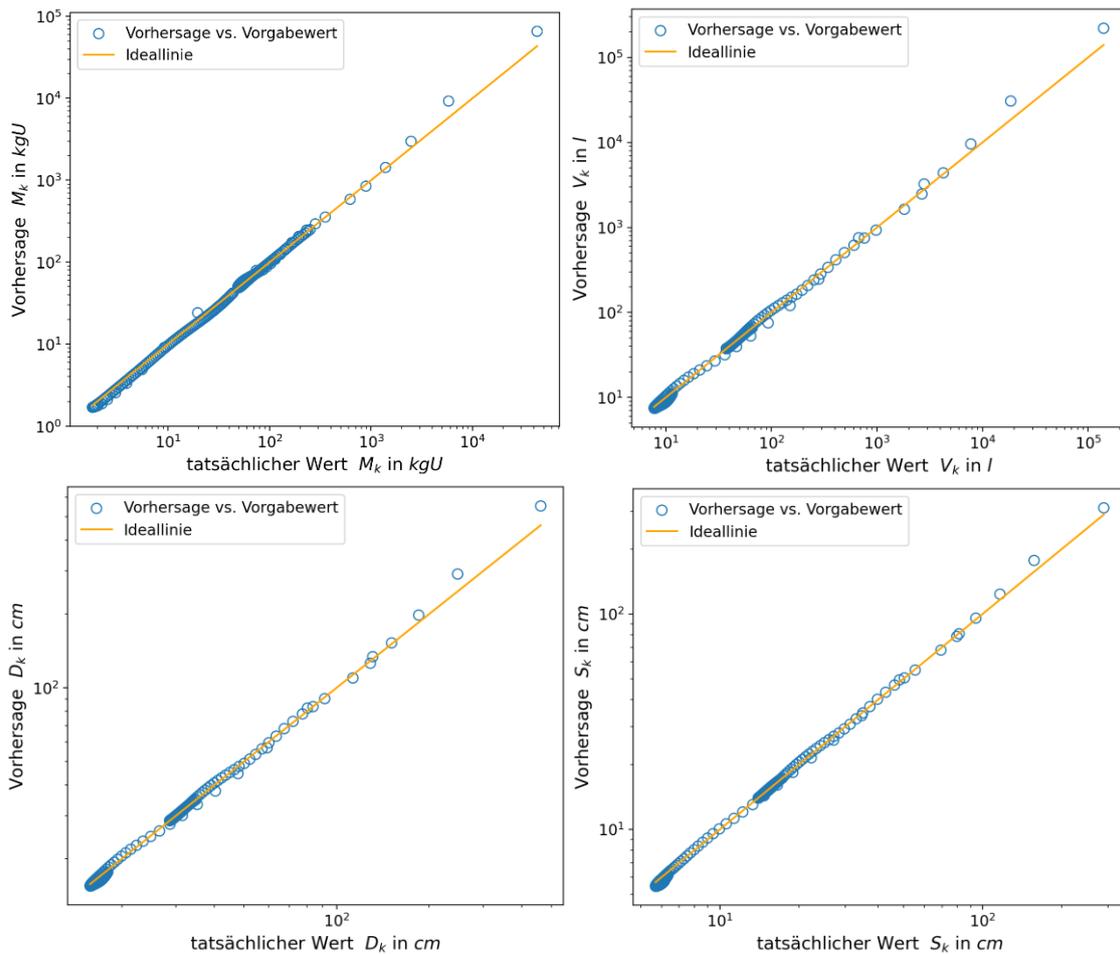
Allgemein können folgende Erkenntnisse aus den RMSRE- und  $R^2$ -Metriken gezogen werden:

- **RMSRE:** Die Werte für Kugelmasse und Kugelvolumen liegen teilweise deutlich über den anderen Kritikalitätsparameter, bleiben jedoch unter 10 %, was auf eine insgesamt gute Modellleistung hinweist.
- **$R^2$ :** Durchgängig hohe Werte über 0,67 bestätigen ebenfalls eine gute Modellleistung.

Für alle vier Haupt-Kritikalitätsparameter liefert das MTL-Modell sowohl für die Metrik RMSRE als auch für  $R^2$  bessere Werte als das MTL-SNN-LR-Modell. Es wird vermutet, dass durch die höhere individuelle Modellkomplexität des MTL bessere Vorhersagen erreicht werden als beim MTL-SNN-LR-Modell, wo mehrere weniger komplexe MTL zusammengefügt wurden. Für  $k_{inf}$  ist dieser Unterschied deutlich weniger ausgeprägt.

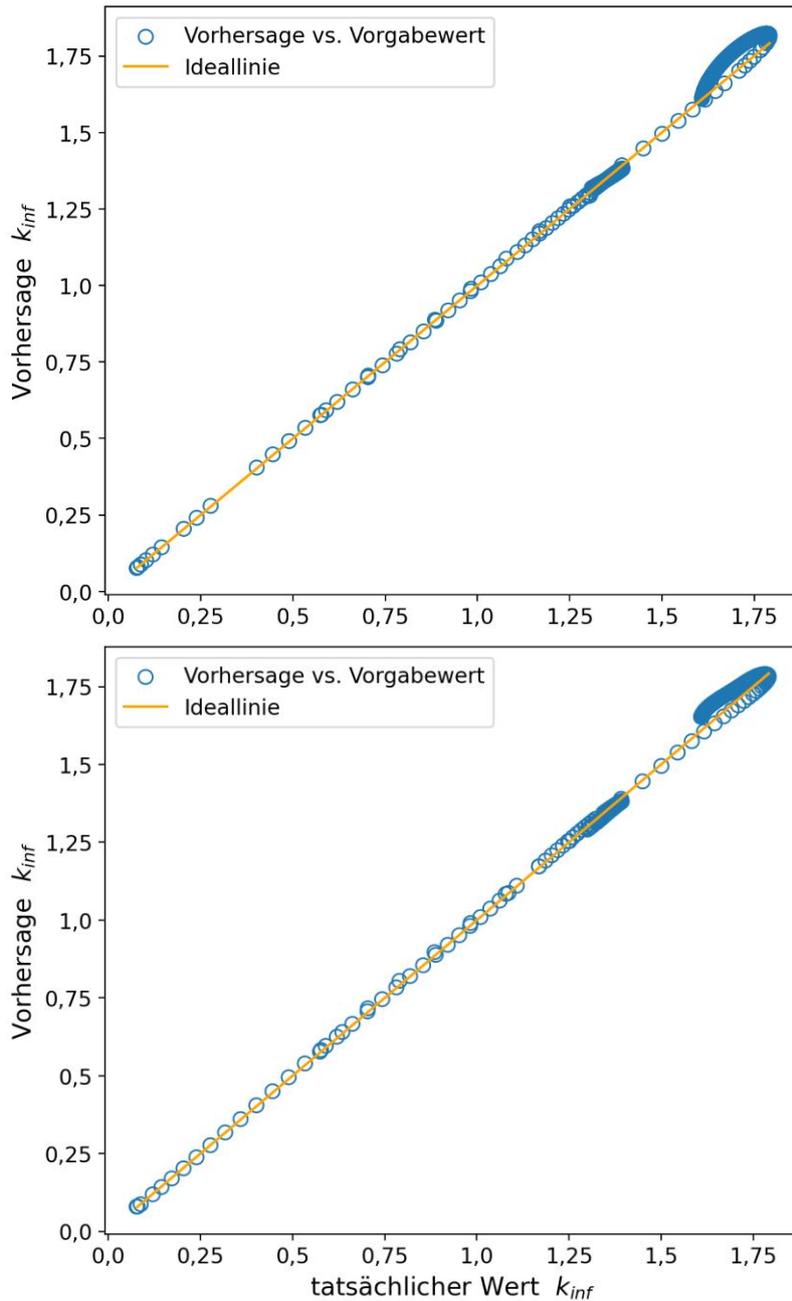


**Abb. 3.10** Vorhersagen des MTL-Modells vs. tatsächlicher Werte mit Referenzlinie



**Abb. 3.11** Vorhersagen des MTL-SNN-LR-Modells vs. tatsächlicher Werte mit Referenzlinie

Die Scatterplots Abb. 3.10 und Abb. 3.11 verdeutlichen, dass größere Abweichungen von der Referenzlinie  $y = x$  vor allem bei hohen Parameterwerten auftreten, die niedrigen Urankonzentrationen entsprechen. Diese Abweichungen betreffen jedoch nur wenige Datenpunkte und unterstreichen die robuste Leistung der Modelle, insbesondere im Bereich mittlerer Urankonzentrationen.

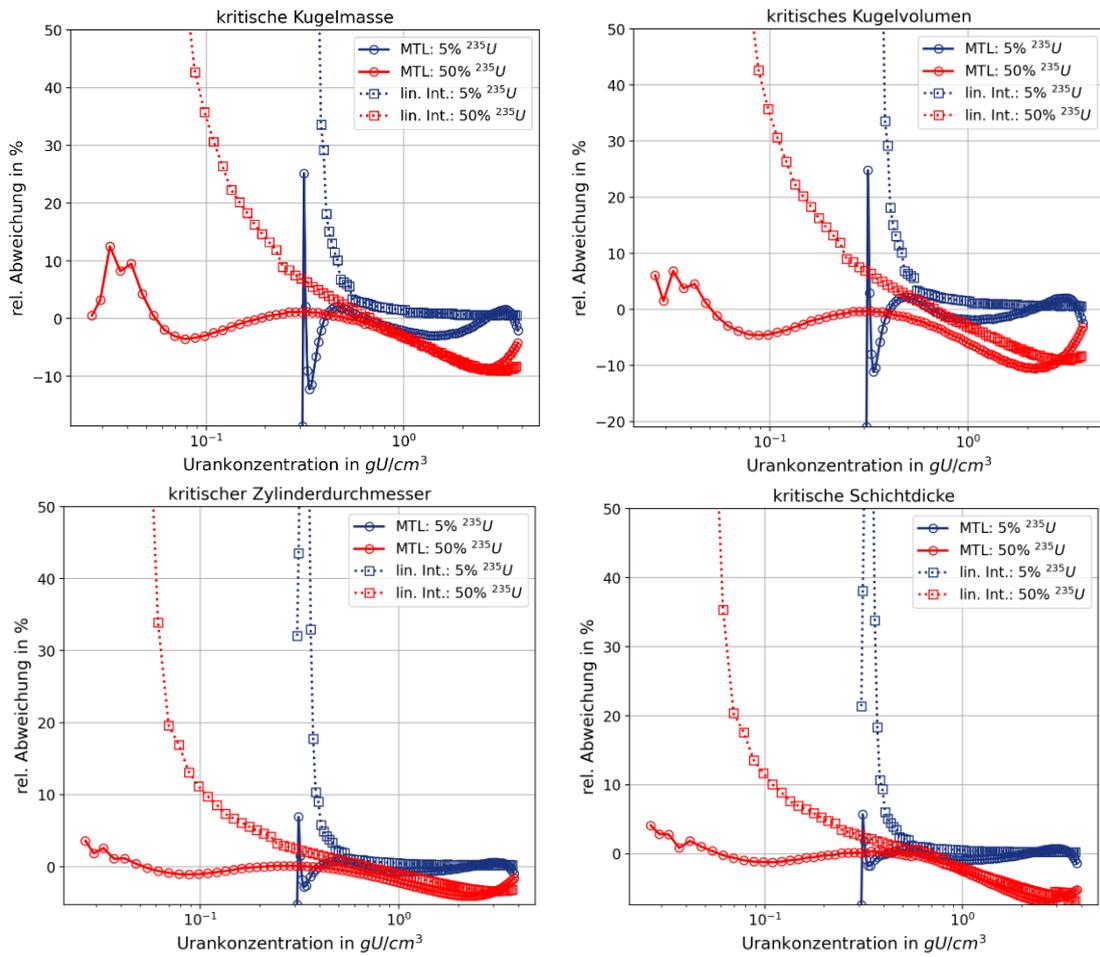


**Abb. 3.12** Vorhersagen des MTL (oben) und des MTL-SNN-LR-Modells (unten) für  $k_{inf}$  vs. tatsächliche Werte mit Referenzlinie

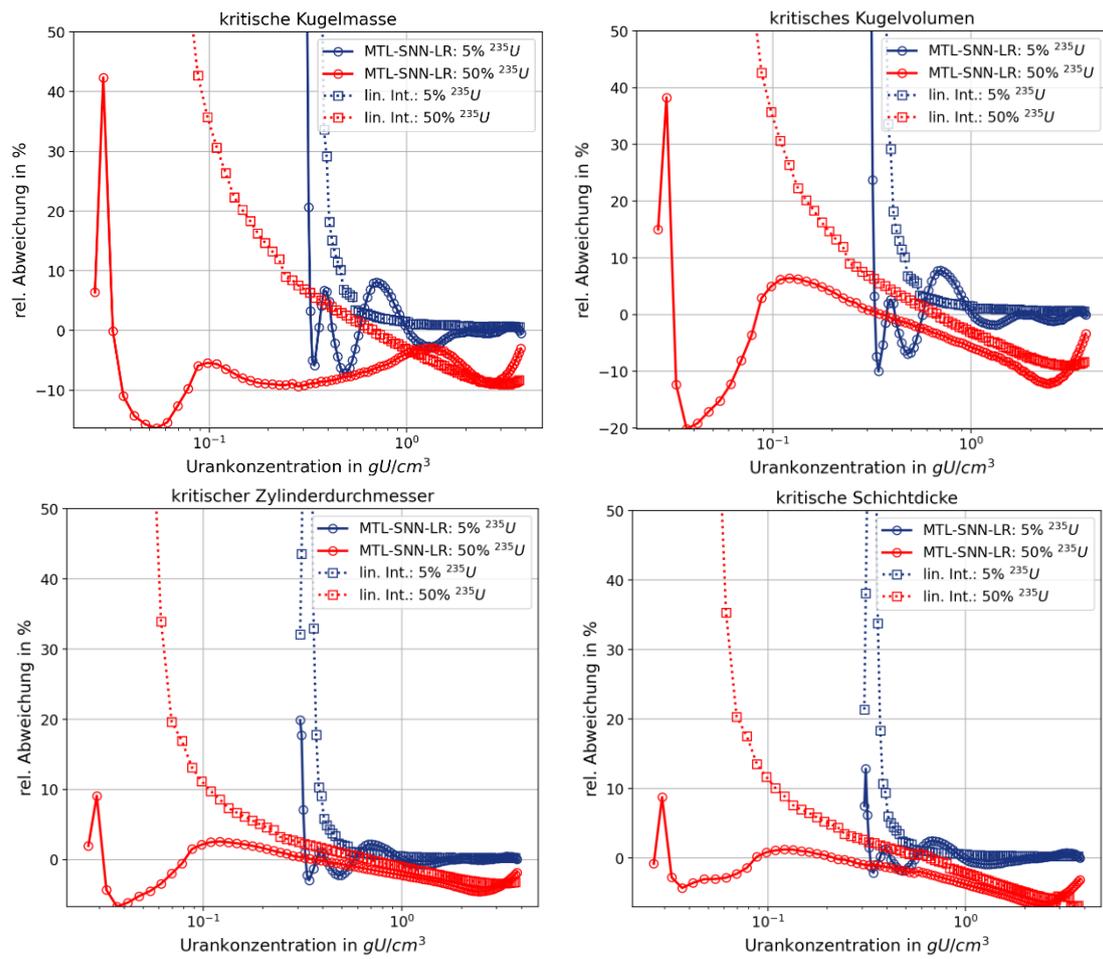
Die Vorhersagen für  $k_{inf}$  in Abb. 3.12 zeigen über einen weiten Wertebereich eine sehr gute Übereinstimmung mit den tatsächlichen Werten. Jedoch ergibt sich im oberen Wertebereich eine Überschätzung der tatsächlichen Werte. Bei dem qualitativen visuellen Vergleich ist erneut zu beachten, dass für  $k_{inf}$  eine lineare Darstellung gewählt wurde, im Gegensatz zu der logarithmischen Darstellung für die restlichen Kritikalitätsparameter. Es lässt sich keine Präferenz für eines der beiden Modelle feststellen.

Der Vergleich mit der linearen Interpolation in Abb. 3.13 bis Abb. 3.14 zeigt auch für Phase 2, dass die KNN-Modelle insgesamt leistungsstärker sind. Insbesondere in stark nichtlinearen Bereichen, in denen die KNN-Modelle zwar etwas schlechtere Leistung aufweisen als in linearen Bereichen, übertreffen sie die Vorhersagegenauigkeit der linearen Interpolation deutlich. In weniger komplexen, nahezu linearen Bereichen liefert die lineare Interpolation vergleichbare, bzw. teils etwas bessere Ergebnisse.

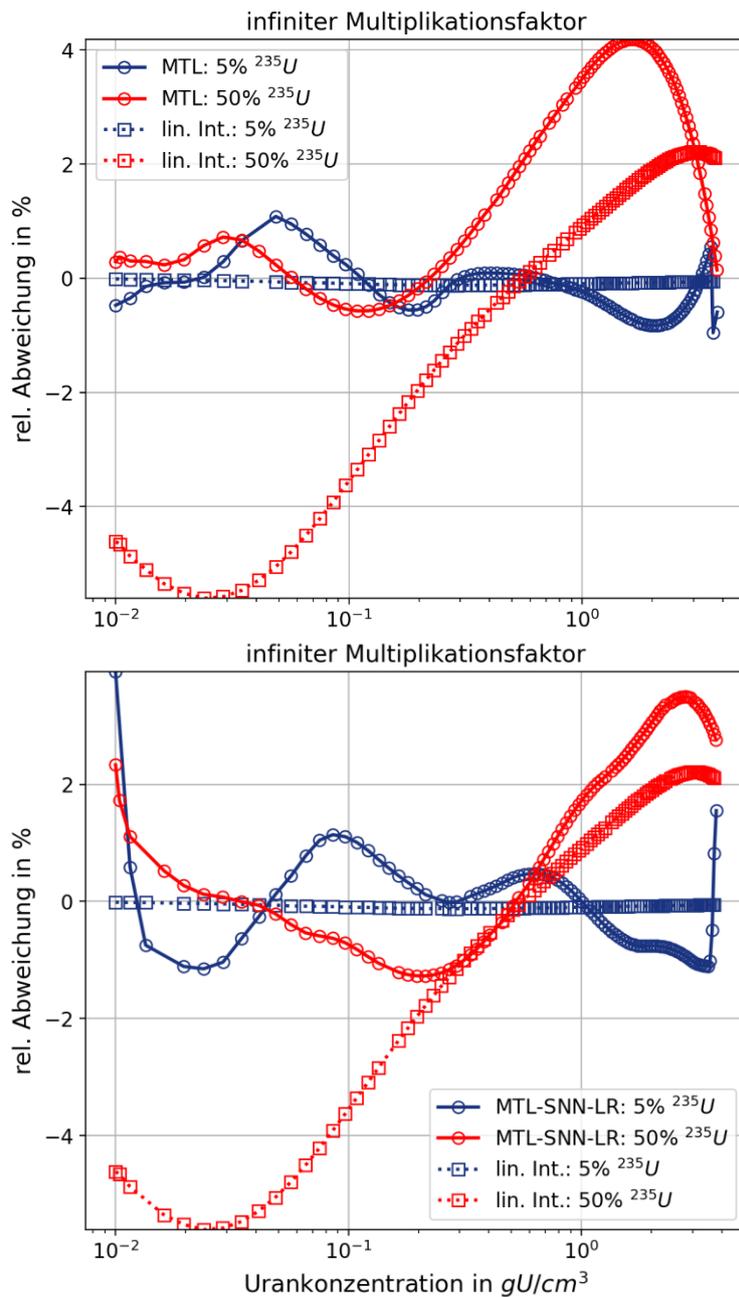
Der Vergleich von  $k_{inf}$  mit der linearen Interpolation in Abb. 3.15 zeigt ein differenziertes Bild. Für eine Anreicherung von 5 %  $^{235}\text{U}$  ergibt die lineare Interpolation eine exzellente Vorhersage mit sehr geringen Abweichungen von weniger als 0,12 %, während beide KNN-Modelle Abweichungen von bis zu 1 % zeigen. Diese exzellente Vorhersage der linearen Interpolation kann damit erklärt werden, dass der Abstand in den Daten zu den nächsten Anreicherungen (4,5 % und 5,5 %) sehr gering ist und die Daten selbst einen nahezu linearen Verlauf aufweisen. Für die Anreicherung von 50 %  $^{235}\text{U}$  zeigen die KNN-Modelle vor allem im unteren Bereich eine deutlich bessere Vorhersage der Werte, während sie im oberen Bereich von  $k_{inf}$  die Werte überschätzen. Über die gesamte Kurve betrachtet schneiden die KNN-Modelle etwas besser als die lineare Interpolation ab.



**Abb. 3.13** Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: MTL-Modell vs. lineare Interpolation



**Abb. 3.14** Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: MTL-SNN-LR-Modell vs. lineare Interpolation



**Abb. 3.15** Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen für  $k_{inf}$ : MTL (oben) und MTL-SNN-LR-Modell (unten) vs. lineare Interpolation

Für den Vergleich von  $k_{inf}$  ist zu beachten, dass der Wertebereich der vertikalen Achse deutlich reduziert, ist im Vergleich zu den anderen Kritikalitätsparametern in Abb. 3.13 und Abb. 3.14.

### 3.2.4 Fazit

Der Modellvergleich ergab für die betrachteten Systeme insgesamt:

- Das **MTL-Modell** ist einfacher zu verstehen und schneller zu trainieren, jedoch empfindlich gegenüber Änderungen der Hyperparameter, was den Optimierungsprozess verlängerte. Die Metriken des MTL-Modells (siehe Tab. 3.2) zeigen zudem sehr ähnlich Werte im Vergleich zu den Metriken des MLP-Modells aus Phase 1 (siehe Tab. 3.1).
- Das **MTL-SNN-LR-Modell** ist robuster gegenüber Hyperparameter-Tuning und zeigt eine stabilere Modellleistung. Es ist jedoch komplexer und mit längeren Trainingszeiten verbunden.

Durch die gezielte Erweiterung der Modellkomplexität und die Integration heterogener Datenquellen wurde die Grundlage für eine robustere und flexiblere Modellierungsstrategie geschaffen.

### 3.3 Phase 3: Anwendung auf weitere chemische Systeme

In der dritten Phase der Arbeiten wurde untersucht, ob ein einziges Modell auch für mehrere chemische Systeme eingesetzt werden kann, indem es systemübergreifende Muster erkennt und verallgemeinert. Daher wurde zusätzlich zum Ammoniumdiurandat -System (ADU) das Ammoniumuranylcarbonat-System (AUC) hinzugenommen.

Die Erweiterung des KNN-Modells um ein weiteres chemisches System erhöhte die Modellkomplexität, da ein zusätzlicher Eingabeparameter eingeführt wird. Eine weitere Herausforderung bei der Modellierung mehrerer chemischer Systeme ist der unterschiedliche Wertebereiche für Urankonzentrationen und  $^{235}\text{U}$ -Anreicherungen in den Trainingsdaten.

#### 3.3.1 Modellierungsansatz

Zur Vorhersage der Kritikalitätsparameter über mehrere chemischen Systeme hinweg wurde ein *Tri-Input-Multi-Task Learning (TMLP)* Modell konzipiert und implementiert. Dieses Modell erkennt systemübergreifende Muster und ermöglicht eine simultane Vorhersage für verschiedene chemische Systeme. Die detaillierte Architektur und Konfiguration des TMLP-Modells werden in Kapitel 2.3.5 beschrieben. Dabei handelt es sich im

Wesentlichen um eine Weiterentwicklung des MLP, um zusätzlich den kategorischen Parameter des chemischen Systems zu berücksichtigen. Das TMLP hat daher, anstelle der bisher zwei, nun drei Neuronen in der Eingabeschicht. Die zwei berücksichtigten chemischen Systeme werden binär kodiert dargestellt (0 für ADU, 1 für AUC). Die Zielvariablen des Modells umfassen weiterhin die vier Hauptkritikalitätsparameter: die kritische Kugelmasse  $M_k$ , das kritische Kugelvolumen  $V_k$ , der kritische Zylinderdurchmesser  $D_k$  und die kritische Schichtdicke  $S_k$ .

### 3.3.2 Datengrundlage

Die Entwicklung und Evaluierung des Modells basiert auf den Datensätzen der ADU- und AUC-Systeme. Ein Vergleich der beiden Datengrundlagen zeigt wichtige Unterschiede auf:

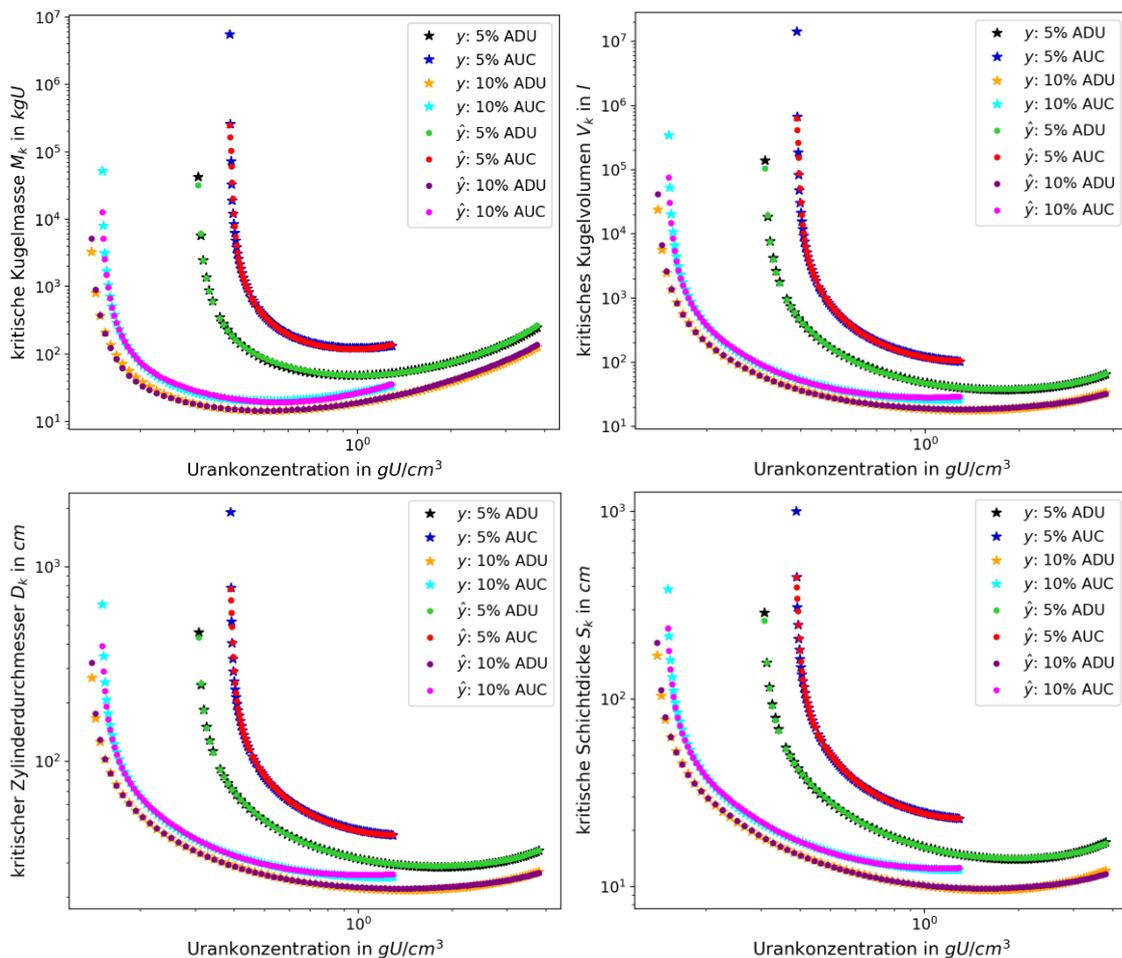
- Unterschiedliche Wertebereiche der Urankonzentration:
  - ADU: 0,01 bis 4 gU/cm<sup>3</sup>
  - AUC: 0,01 bis 1,5 gU/cm<sup>3</sup>
- Unterschiedliche Wertebereiche der Kritikalitätsparameter, hier am Beispiel der kritischen Kugelmasse:
  - ADU: 0,5 bis 10<sup>6</sup> kgU
  - AUC: 0,5 bis 10<sup>8</sup> kgU

Zum Testen wurden für beide Systeme die Daten zu <sup>235</sup>U-Anreicherungen von 5 und 10 Gew% verwendet. Die Vorgehensweise bei der Datenaufbereitung ist analog zu den bisher betrachteten Fällen und in Kapitel 2.4.3 ausführlich beschrieben.

### 3.3.3 Ergebnisse

In Abb. 3.16 werden die vom TMLP-Modell vorhergesagten Kritikalitätsparameter für beide chemischen Systeme und die tatsächlichen Werte in Abhängigkeit der Urankonzentration gegenübergestellt. Das Modell erreicht bei mittleren Urankonzentrationen eine hohe Übereinstimmung, während bei niedrigen Urankonzentrationen größere Abweichungen auftreten. Die in Kapitel 3.1.3 erläuterten Ursachen für diese Abweichungen gelten auch hier. Während die Größe der Abweichungen für das ADU-System

weitgehend den Werten aus dem MLP-Modell entsprechen (siehe Kapitel 3.1), sind die Abweichungen für das AUC-System stärker ausgeprägt. Am unteren Ende des Konzentrationsbereichs sind mehr Datenpunkte von einer starken Abweichung betroffen, als dies für ADU der Fall ist. Zudem weichen die Vorhersagen für die kritische Kugelmasse und das Kugelvolumen stärker von den tatsächlichen Werten ab als bei den anderen Kritikalitätsparametern. Hierbei muss allerdings wiederum der unterschiedliche Wertebereich der jeweiligen vertikalen Achse beachtet werden, der sich zwischen den kritischen Parametern um mehrere Größenordnungen unterscheidet.



**Abb. 3.16** Vorhersagen des TMLP-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten

Die Leistungsbewertung des TMLP-Modells erfolgte auch mittels der RMSRE und  $R^2$  Metriken, die in Tab. 3.3 zusammengefasst sind.

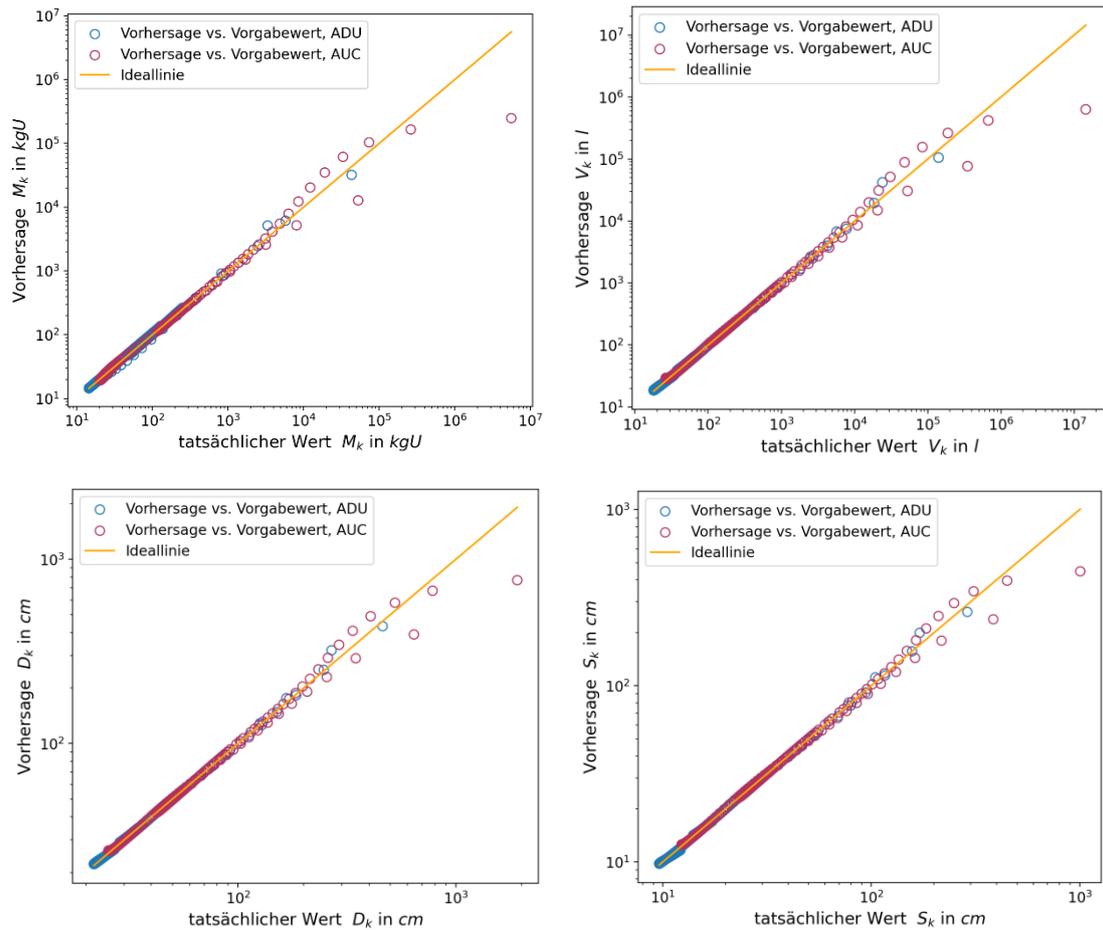
**Tab. 3.3** Phase 3: Vergleich der Metriken zwischen dem ADU- und AUC-Systeme für das TMLP-Modell für 5 % und 10 % Anreicherung

Anreicherung:	5 %				10 %			
Metrik:	RMSRE		R <sup>2</sup>		RMSRE		R <sup>2</sup>	
System:	ADU	AUC	ADU	AUC	ADU	AUC	ADU	AUC
$M_k$	3,9 %	19,3 %	0,934	0,077	7,8 %	9,8 %	0,713	0,428
$V_k$	3,5 %	19,4 %	0,938	0,076	8,3 %	11,3 %	0,468	0,391
$D_k$	1,1 %	7,7 %	0,997	0,691	2,4 %	4,9 %	0,972	0,884
$S_k$	1,4 %	7,0 %	0,994	0,747	2,4 %	4,9 %	0,979	0,896

Allgemein können folgende Erkenntnisse aus den Metriken gezogen werden:

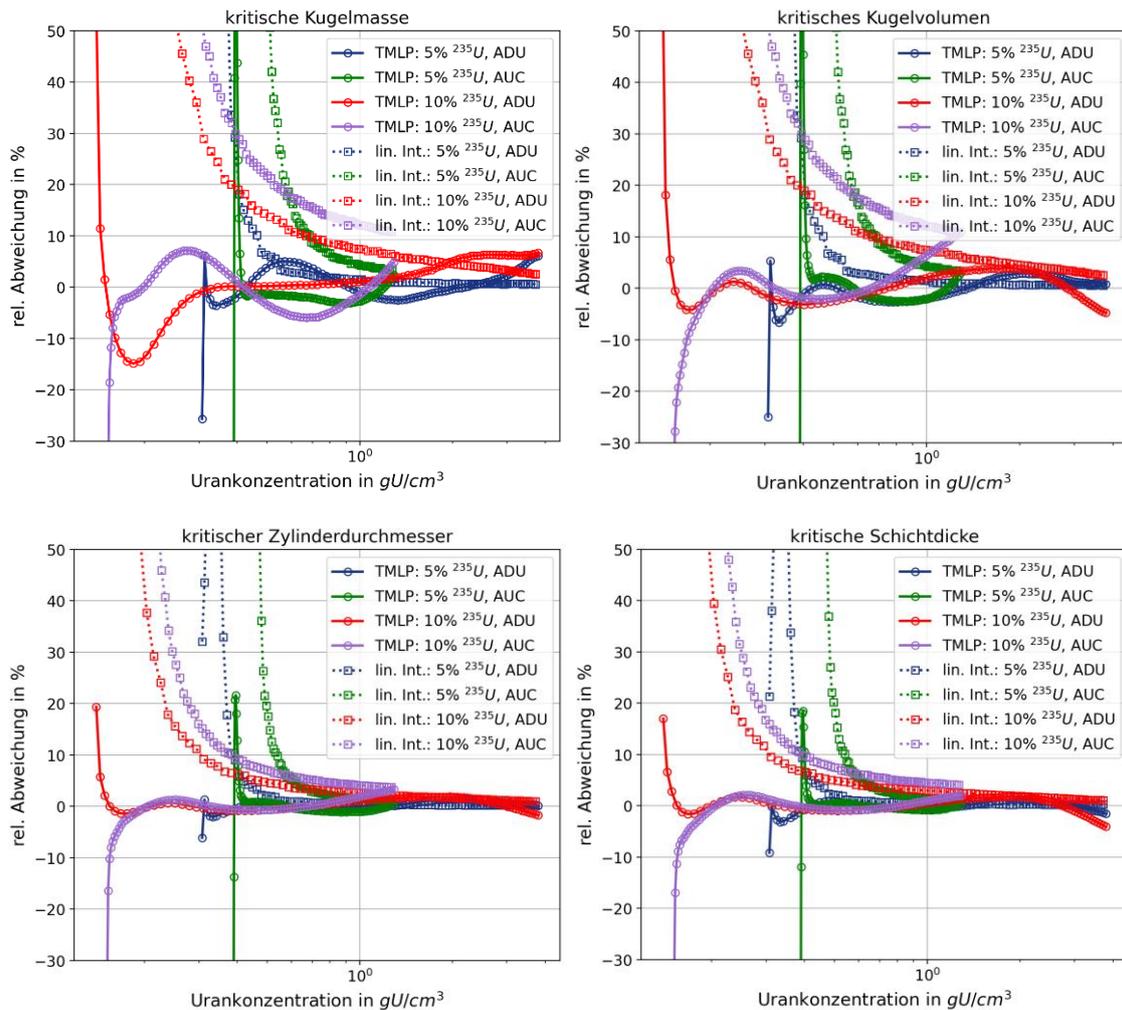
- **RMSRE:** Die Fehlerwerte für die kritische Kugelmasse und das Kugelvolumen sind höher als für den Zylinderdurchmesser und die Schichtdicke. Für die ADU-Systeme ergeben sich weitgehend dieselben Werte wie für das MLP-Modell, das nur auf ADU-Systeme trainiert wurde (siehe Kapitel 3.1 und Tab. 3.1). Für AUC werden große Fehler gemessen. Die visuelle Analyse zeigt, dass diese Abweichungen hauptsächlich durch wenige Datenpunkte bei niedrigen Urankonzentrationen im stark nicht-linearen Bereich verursacht werden.
- **R<sup>2</sup>:** Während sich für ADU wiederum sehr ähnliche Werte wie für das MLP-Modell zeigen, weist der Vergleich für AUC teilweise deutlich niedrige Werte auf, vor allem für die Kugelmasse und das Kugelvolumen. Dies korreliert mit den höheren RMSRE-Werten.

Die Scatterplots in Abb. 3.17 zeigen die Unterschiede zwischen den Vorhersagen und den tatsächlichen Werten. Die Daten für ADU sind in Blau angetragen, die Daten für AUC in Rot. Die Darstellung zeigt, dass signifikante Abweichungen von der Ideallinie  $y = x$  hauptsächlich bei hohen Parameterwerten auftreten, die niedrigen Urankonzentrationen entsprechen. Lässt man diese Abweichungen außen vor, die nur wenige Datenpunkte betreffen, demonstriert das Modell eine robuste Leistung, insbesondere im Bereich mittlerer Urankonzentrationen.



**Abb. 3.17** Vorhersagen des TMLP-Modells vs. tatsächlicher Werte mit Referenzlinie

Im Vergleich mit der linearen Interpolation in Abb. 3.18 zeigt das TMLP-Modell eine insgesamt stärkere Leistung. Besonders in stark nichtlinearen Bereichen erreicht es eine deutlich höhere Genauigkeit als die lineare Interpolation. In weniger komplexen, annähernd linearen Bereichen liefert die lineare Interpolation vergleichbare, bzw. teilweise etwas bessere Ergebnisse.



**Abb. 3.18** Vergleich der Vorhersagegenauigkeit anhand der relativen Abweichungen: TMLP-Modell vs. lineare Interpolation

### 3.3.4 Fazit

Das TMLP-Modell zeigt insgesamt eine gute Leistungsfähigkeit bei der Vorhersage der Hauptkritikalitätsparameter, insbesondere im mittleren Bereich der Urankonzentrationen. Die geringere Genauigkeit der Vorhersagen im AUC-System im Vergleich zum ADU-System, vor allem für die Kugelmasse und das Kugelvolumen, bleibt jedoch eine offene Fragestellung. Eine detaillierte Analyse dieser Unterschiede war im Rahmen dieser Arbeit nicht möglich und soll Gegenstand zukünftiger Forschungsarbeiten sein.

Mögliche Ursachen für diese Diskrepanz sind:

- **Datenqualität:** Eine möglicherweise schlechtere Datenbasis für AUC könnte die Generalisierungsfähigkeit des Modells einschränken.
- **Skalierung:** Die gemeinsame Skalierung beider Datensätze könnte problematisch sein, da sich die Wertebereiche der Kritikalitätsparameter für ADU und AUC um mehrere Größenordnungen unterscheiden. Die Verwendung getrennter Skalierungsverfahren könnte vorteilhaft sein.
- **Unterschiedliche Sensitivitäten:** Die Kritikalitätsparameter könnten bei AUC empfindlicher auf kleine Änderungen in den Eingabeparametern reagieren (z. B. bei Augmentation), was die Vorhersage erschweren würde.
- **Modellkomplexität:** Ein komplexeres Modell könnte für die Vorhersage in mehreren chemischen Systemen besser geeignet sein. Ein Meta-Modell, das einen gestapelten Datensatz verarbeitet, der von mehreren einfacheren Modellen erzeugt wurde (ähnlich wie SNN-LR und MTL-SNN-LR), könnte eine mögliche Lösung sein.

## 4 KNN-Modellentwicklung: Erkenntnisse und Empfehlungen

In dieser Arbeit wurden im Laufe der Entwicklung der KNN-Modelle zahlreiche Erkenntnisse gewonnen, von denen zukünftige Forschungsarbeiten profitieren können, um ähnliche Herausforderungen zu meistern. Diese Erkenntnisse und die daraus abgeleiteten Empfehlungen werden im Folgenden zusammengefasst.

- **Datensatzgröße und Augmentation**

Ob eine akzeptable Modellleistung erreicht werden kann, hängt maßgeblich von der Größe des Trainingsdatensatzes ab. Um die begrenzte Datenbasis zu erweitern, wurden zusätzliche Datenpunkte durch Augmentation generiert und dem Originaldatensatz hinzugefügt. Diese Technik ermöglicht nicht nur eine quantitative Vergrößerung des Datensatzes, sondern fördert auch die Generalisierbarkeit der Modelle. Durch die Einführung kontrollierter Variationen in den Trainingsdaten lernen die Modelle, robuster auf verschiedene Eingaben zu reagieren und können somit besser auf neue, ungesehene Daten generalisieren.

- **Auswahl des Testdatensatzes**

Die Bewertung der Modellleistung erfolgt anhand der Testdaten. Es wurde festgestellt, dass es aus mehreren Gründen wichtig ist, wie der Testdatensatz zusammengestellt wird.

Zwei Ansätze wurden erprobt, wobei die Ergebnisse der zufälligen Auswahl nicht im Bericht dargestellt sind:

1. **Spezifische Anreicherungen entfernen:** Datenpunkte zu bestimmten  $^{235}\text{U}$ -Anreicherungen wurden vollständig aus dem Originaldatensatz entfernt und ausschließlich für das Testen verwendet. In der vorliegenden Studie waren dies immer zwei Anreicherungen.
2. **Zufällige Auswahl:** 20 % der Datenpunkte wurden zufällig ausgewählt und aus dem Originaldatensatz entfernt. Der Testdatensatz enthält dann Datenpunkte mit allen  $^{235}\text{U}$ -Anreicherungen.

Es ist zu empfehlen, den Testdatensatz nach der ersten Methode zu bilden. Die Gründe sind folgende:

- Dieser Einsatz ermöglicht eine gezielte visuelle Überprüfung, ob und wie gut das Modell charakteristische Muster und funktionalen Abhängigkeiten (Kursverläufe) erkennt.
- Diese Methode erleichtert auch die Identifikation von Bereichen mit guter oder schlechter Modelleistung und hilft, mögliche Systematiken dabei zu erkennen.
- Dieses Vorgehen kommt einem realen Anwendungsfall nahe, bei dem z. B. die kritischen Parameterkurven für eine Anreicherung ermittelt werden soll, die nicht im Datensatz vorhanden ist.

- **Visuelle und numerische Evaluierung**

Die Verwendung numerischer Metriken wie RMSRE und  $R^2$  zur Leistungsbewertung sind wichtig, können jedoch Einschränkungen aufweisen. Numerische Werte können durch wenige Datenpunkte (z. B. Ausreißer) stark beeinflusst werden, was das Gesamtbild verzerrt. Die ergänzende, qualitative visuelle Bewertung ermöglicht es, Abweichungen im Kontext zu bewerten.

- **Unterschiede in der Vorhersagegenauigkeit**

Die Modellgenauigkeit variiert zwischen den Zielvariablen. Während die Vorhersagen für die kritische Kugelmasse und das kritische Kugelvolumen weniger genau sind, liefern der kritische Zylinderdurchmesser und die kritische Schichtdicke präzisere Ergebnisse. Eine mögliche Ursache dieser Diskrepanzen sind die weitaus größeren Wertebereiche, die die kubischen Größen, also Kugelmasse und das Kugelvolumen, im Vergleich zu den anderen Kritikalitätsparametern abdecken. Auf alle Kritikalitätsparameter werden zur Verwendung im KNN vorab Datentransformationen (Logarithmieren, Skalieren) angewandt. Zur Berechnung des Fehlers werden die KNN-Vorhersagen aber zurücktransformiert. Dadurch kann diese Fehlerverzerrung bei den verschiedenen Kritikalitätsparametern entstehen.

- **Einfluss von batch-Größe und Lernrate**

Die Wahl der batch-Größe beeinflusst sowohl die Modellgenauigkeit als auch die Stabilität des Trainingsprozesses:

- **Kleinere batch-Größen:** Der Wert der Verlustfunktion während des Trainings schwankt stärker, was zu ungleichmäßigen Lernkurven führen kann.

- **Größere batch-Größen:** Der Wert der Verlustfunktion während des Trainings verläuft stabiler, da die Gradientenberechnungen glatter sind. Allerdings besteht erhöhte Gefahr, dass das Modell in einem suboptimalen Minimum verharrt, was die Genauigkeit beeinträchtigt.

Es hat sich als vorteilhaft erwiesen, mit verschiedenen batch-Größen zu experimentieren, um eine optimale Modellleistung zu erreichen. Zudem sollte die Lernrate sorgfältig auf die batch-Größe abgestimmt werden, da beide Parameter stark miteinander korrelieren.



## 5 Fazit und Ausblick

Die Ergebnisse dieser Arbeit zeigen, dass KNN-Modelle eine vielversprechende Methode sind, um komplexe Zusammenhänge in Daten zu erfassen. Speziell bei der Berechnung von Kritikalitätsparametern stellen sie eine leistungsfähige Alternative zu klassischen Ansätzen wie der linearen Interpolation dar.

Im Rahmen dieser Arbeit wurden Modell-Prototypen entwickelt, die in der Praxis eingesetzt werden können. Diese Modelle dienen ebenfalls als Grundlage für die Identifikation weiterer Anwendungsfelder und liefern wertvolle Erkenntnisse für zukünftige Entwicklungen. Diese Arbeit leistet einen Beitrag zur Standardisierung und zur Entwicklung von Best Practices im in der GRS noch wenig erforschten Bereich der KNN-Modellierung, insbesondere im Kontext der Berechnung von Kritikalitätsparametern.

Für die zukünftige Arbeit bei der Anwendung von KNN für die Generierung kritischer Parameterkurven ergeben sich folgende wichtige Perspektiven:

- **Erhöhung der Problemkomplexität**

Ein zentraler Ansatzpunkt für zukünftige Arbeiten ist die Erhöhung der Problemkomplexität, um die Grenzen von KNN-Modellen besser zu verstehen. Praktisch relevante Erweiterungen können beispielsweise sein:

- Erweiterung der Eingabeparameter um den Spaltstoff: Neben Uran könnte auch Plutonium als Spaltstoff in das Modell aufgenommen werden.
- Erhöhung der Anzahl der Zielvariablen: Zusätzliche Zielvariablen würden umfassendere Vorhersagen ermöglichen. Dadurch würden noch komplexere Muster und Zusammenhänge zwischen den Eingabe- und Zielvariablen entstehen.

Mit zunehmender Modellkomplexität müssen unterschiedliche Muster und Zusammenhänge innerhalb der Daten erkannt und verstanden werden. Es ist wichtig zu untersuchen, ob KNN-Modelle in der Lage sind, diese erhöhte Komplexität effektiv zu erfassen, ohne dass die Modelleleistung darunter leidet.

- **Alternative Lösungsansätze für komplexe Probleme**

Die in dieser Arbeit entwickelte Architektur mit mehreren Ein- und Ausgabeschichten (MTL-Modell) hat sich als praktikabel erwiesen. Mit zunehmender Problemkomplexität könnte dieser Ansatz jedoch an seine Grenzen stoßen, sodass alternative

Modellansätze erforderlich werden. Dabei sollte untersucht werden, ob fortschrittlichere oder hybride Techniken besser geeignet sind, diese Herausforderungen zu bewältigen.

- **Unterschiede zwischen AUC und ADU**

Die geringere Genauigkeit der Vorhersagen im AUC-System im Vergleich zum ADU-System, vor allem für die Kugelmasse und das Kugelvolumen, sollen in zukünftigen Forschungsarbeiten weiter untersucht werden. Mögliche Ursachen könnten eine schlechtere Datenbasis, ungeeignete Skalierungsverfahren, modellbedingte Einschränkungen oder grundlegende Unterschiede in der physikalischen Systematik sein. Eine detaillierte Analyse könnte helfen, gezielte Verbesserungen an den Modellen oder den Datenvorbereitungsmethoden vorzunehmen.

- **Aufbereitung der Daten**

Bei der Aufbereitung der Daten stellten vor allem nichtlineare Bereiche an den Rändern der kritischen Parameterkurven eine Herausforderung dar, da diese schlechter im Datensatz repräsentiert sind. Außerdem sind auch hohe  $^{235}\text{U}$ -Anreicherungen im Datensatz unterrepräsentiert. Dies spiegelt sich in der Vorhersageleistung der KNN wider, die für diese Datenbereiche im Allgemeinen schlechter abschnitten. Eine mögliche Verbesserung könnte mit einer stärkeren Gewichtung dieser Datenbereiche im Trainingsdatensatz erzielt werden.

- **Erweiterung der Evaluierungsmethoden**

Um eine objektivere Bewertung der Modelleistung zu ermöglichen, sollten in zukünftigen Arbeiten fortschrittlichere Interpolationsverfahren in Betracht gezogen werden, insbesondere solche, die Nichtlinearitäten in den zugrunde liegenden Daten besser erfassen können, z. B. quadratische oder Spline-Interpolation. Dies würde genauere Aussagen über die Anwendbarkeit von KNN-Modellen ermöglichen.

- **Entwicklung einer benutzerfreundlichen Schnittstelle**

Ein weiterer wichtiger Schritt ist die Entwicklung einer intuitiven und benutzerfreundlichen Oberfläche für die KNN-Modelle. Eine solche Schnittstelle würde den praktischen Einsatz der Modelle erleichtern, insbesondere für Anwender ohne tiefgehende Kenntnisse in der Modellierung oder Implementierung. Dadurch könnte die Akzeptanz und Verbreitung der Modelle erheblich gesteigert werden.

Durch diese Maßnahmen kann das Potenzial von KNN-Modellen im Rahmen weiterer Studien untersucht werden, um genauere Vorhersagen in komplexen Szenarien zu ermöglichen und die Einsatzmöglichkeiten dieser Modelle auf verschiedene Bereiche zu erweitern.



## Literaturverzeichnis

- /EIS 22/ Eisenstecken, T., Sommer, F.: Application of Neural Networks on the Modelling of Critical Curves. In: Song, C.-H., Sun, X., Pointer, D., Lucas, D. (Hrsg.): Advances in Thermal Hydraulics (ATH 2022). Anaheim, CA, 12. - 16. Juni 2022, S. 68–75, ISBN 978-0-89448-781-1, DOI 10.13182/T126-37480, American Nuclear Society: Illinois, Juni 2022/Juni 2022.
- /GRS 19/ Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH (GRS): Handbuch zur Kritikalität, Band 2 - Zusammenstellung der Kritikalitätskenngrößen, Teil 1 - Daten für 235Uran, überarbeitete Auflage. GRS-380, ISBN 978-3-944161-61-7: Garching b. München, Juni 2019.
- /LI 12/ Li, J., Cheng, J., Shi, J., Huang, F.: Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement. In: Jin, D., Lin, S., Li, J., Cheng, J., Shi, J., Huang, F. (Hrsg.): Advances in Computer Science and Information Engineering, Proceeding of CSIE2012. CSIE, Zhengzhou, China, 19. - 20. Mai 2012, Advances in Intelligent and Soft Computing, 2 // 169, Bd. 169, S. 553–558, ISBN 978-3-642-30222-0, DOI 10.1007/978-3-642-30223-7\_87, Springer: Berlin, 2012.
- /PYT 23/ Python Software Foundation.: Python Language Reference, version 3.12. Erreichbar unter <https://docs.python.org/3/>, Stand von 2023.
- /REA 17/ Rearden, B. T., Jessee, M. A. (Hrsg.): SCALE Code System. Oak Ridge National Laboratory (ORNL), ORNL/TM-2005/39, Version 6.2.2, 2747 S.: Oak Ridge, Tennessee, Februar 2017.
- /SOM 19/ Sommer, F., Bartos, O., Behler, M., Hannstein, V., Kaufholz, P., Kilger, R., Krüger, J., Rowold, F.: Bereitstellung und Weiterentwicklung von Methoden, Daten und Rechenmethoden zu nuklearen Sicherheitsanalysen. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH (GRS), GRS-Bericht, GRS-542, ISBN 978-3-947685-27-1: Garching, 2019.
- /TEN 24/ Tensorflow: Tensorflow. Erreichbar unter <https://www.tensorflow.org/>, Stand von 2024.



## Abbildungsverzeichnis

Abb. 2.1	Typische Repräsentation eines KNN .....	5
Abb. 2.2	Ausschnitt einer Beispieldatei mit Rohdaten, die für das Training weiterverarbeitet werden.....	18
Abb. 2.3	Kritikalitätsparameter von homogenen ADU-Systemen für verschiedene <sup>235</sup> U-Anreicherungen als Funktion der Urankonzentration .....	20
Abb. 2.4	Kritikalitätsparameter von homogenen AUC-Systemen für verschiedene <sup>235</sup> U-Anreicherungen als Funktion der Urankonzentration .....	21
Abb. 2.5	<i>kinf</i> als Funktion der Urankonzentration für verschiedene <sup>235</sup> U-Anreicherungen von homogenen Ammoniumdiuranat(ADU)-Wasser-Systemen .....	22
Abb. 3.1	Vorhersagen des SNN-LR-Modells für kritische Parameter im Vergleich zu tatsächlichen Werten.....	35
Abb. 3.2	Vorhersagen des MLP-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten.....	36
Abb. 3.3	Vorhersagen des SNN-LR-Modells vs. tatsächlicher Werte mit Referenzlinie.....	38
Abb. 3.4	Vorhersagen des MLP-Modells vs. tatsächlicher Werte mit Referenzlinie.....	39
Abb. 3.5	Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: SNN-LR-Modell vs. lineare Interpolation .....	40
Abb. 3.6	Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: MLP-Modell vs. lineare Interpolation .....	41
Abb. 3.7	Vorhersagen des MTL-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten.....	45
Abb. 3.8	Vorhersagen des MTL-SNN-LR-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten.....	46
Abb. 3.9	Vorhersagen des MLT und des MTL-SNN-LR-Modells für <i>kinf</i> im Vergleich zu den tatsächlichen Werten .....	47
Abb. 3.10	Vorhersagen des MTL-Modells vs. tatsächlicher Werte mit Referenzlinie.....	49

Abb. 3.11	Vorhersagen des MTL-SNN-LR-Modells vs. tatsächlicher Werte mit Referenzlinie.....	50
Abb. 3.12	Vorhersagen des MTL (oben) und des MTL-SNN-LR-Modells (unten) für <i>k<sub>inf</sub></i> vs. tatsächliche Werte mit Referenzlinie .....	51
Abb. 3.13	Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: MTL-Modell vs. lineare Interpolation .....	53
Abb. 3.14	Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen: MTL-SNN-LR-Modell vs. lineare Interpolation.....	54
Abb. 3.15	Vergleich der Vorhersagegenauigkeit anhand relativer Abweichungen für <i>k<sub>inf</sub></i> : MTL (oben) und MTL-SNN-LR-Modell (unten) vs. lineare Interpolation.....	55
Abb. 3.16	Vorhersagen des TMLP-Modells für Kritikalitätsparameter im Vergleich zu tatsächlichen Werten.....	58
Abb. 3.17	Vorhersagen des TMLP-Modells vs. tatsächlicher Werte mit Referenzlinie.....	60
Abb. 3.18	Vergleich der Vorhersagegenauigkeit anhand der relativen Abweichungen: TMLP-Modell vs. lineare Interpolation.....	61

## **Tabellenverzeichnis**

Tab. 3.1	Phase 1: Vergleich der Metriken zwischen SNN-LR- und MLP-Modell für 5 % und 50 % Anreicherung .....	37
Tab. 3.2	Phase 2: Vergleich der Metriken zwischen MTL- und MTL-SNN-LR-Modell für 5 % und 50 % Anreicherung.....	48
Tab. 3.3	Phase 3: Vergleich der Metriken zwischen dem ADU- und AUC-Systeme für das TMLP-Modell für 5 % und 10 % Anreicherung.....	59

## **Anhang A**

Tab. A. 1	Im Dokument verwendete Abkürzungen .....	77
-----------	--	----



## Abkürzungen

Tab. A. 1 Im Dokument verwendete Abkürzungen

Abkürzung	Bedeutung
Adam	Adaptive Moment Estimation
ADU	Ammoniumdiurandat
AUC	Ammoniumuranylcarbonat
ELU	Exponential Linear Unit
GRS	Gesellschaft für Anlagen- und Reaktorsicherheit
KI	Künstliche Intelligenz
KNN	Künstliches Neuronales Netz
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MTL	Multi-Task Learning
MTL-SNN-LR	Multi-Task Learning Stacked Neural Networks with Linear Regressor
MSE	Mean Squared Error
RMSRE	Root Mean Square Relative Error
SCALE	Standardized Computer Analyses for Licensing Evaluation
SNN-LR	Stacked Neural Networks with Linear Regressor
TMLP	Tri-Input-Multi-Task Learning

**Gesellschaft für Anlagen-  
und Reaktorsicherheit  
(GRS) gGmbH**

Schwertnergasse 1  
**50667 Köln**

Telefon +49 221 2068-0

Telefax +49 221 2068-888

Boltzmannstraße 14

**85748 Garching b. München**

Telefon +49 89 32004-0

Telefax +49 89 32004-300

Kurfürstendamm 200

**10719 Berlin**

Telefon +49 30 88589-0

Telefax +49 30 88589-111

Theodor-Heuss-Straße 4

**38122 Braunschweig**

Telefon +49 531 8012-0

Telefax +49 531 8012-200

[www.grs.de](http://www.grs.de)