Vorhersage von Zeitreihen durch Nutzung von Machine Learning-Ansätzen am Beispiel einer Niederschlagsabflussmodellierung

Vorgelegt von Kadir Polat und Hamza Salhi

# *Abstract*

In diesem Paper wird die Nutzung von Machine Learning Ansätzen als Alternative zu deterministischen Modellen zur Vorhersagetreffung von Zeitreihen am Beispiel einer Niederschlagsabflussmodellierung vorgestellt.

Bei diesen Ansätzen ist im Gegensatz zu deterministischen Modellierungen kein detailliertes Verständnis der hydrologischen Prozesse und Abflussvorgänge notwendig, da diese auf Beobachtungsdaten basieren. Auf Grundlage der Beobachtungsdaten eines bestimmten Zeitraums ist die Vorhersage der Abflüsse möglich, wodurch ein Zusammenhang zwischen Niederschlag und Abfluss aus Zeitreihen ableitbar wird.

Zur Modellierung werden *Supervised Learning* Verfahren genutzt. Genauer werden Standard *Long Short-Term Memory* (LSTM) Netze mit einem *Convolutional Neuronal Network* kombiniertem LSTM Netz (CNN-LSTM) verglichen. Zur Realisierung werden das Standard LSTM Netz als Datenquelle des 1-Dimensionalen *Camels* US-Datensatzes und das kombinierte CNN-LSTM Netz des 2-Dimensionalen *Daymet* us-amerikanischen Datensatzes genutzt.

Die Ergebnisse der Modellierung zeigen eindeutig die höhere Akkuratheit des kombinierten CNN-LSTM Netzes gegenüber dem Standard LSTM Netz auf. In zwei von drei der für die Modellierung herangezogenen Einzugsgebiete konnten eindeutig erfolgreichere Resultate erzielt werden. Beide Netzwerkarchitekturen erzielten für die jeweiligen Einzugsgebiete der Niederschlagabflussmodellierung in hohem Maße zutreffende und repräsentative Ergebnisse.

1. *Einführung*

Die künstliche Intelligenz stellt eine neue Möglichkeit zur Erstellung von Niederschlagabflussmodellierungen zu den bisher herangezogenen deterministischen Ansätzen dar. Für letztere sind exzellente Kenntnisse bezüglich des Einzugsgebiets und der mathematischen Zusammenhänge von Nöten. Durch das Vorhandensein von langzeitig erhobenen Daten können hingegen neue, leichter zugängliche Modellierungsansätze mittels *Deep Learning* Verfahren genutzt werden. Tiefgreifendes Wissen und Verständnis über das Einsatzgebiet sowie die dazugehörigen mathematischen Modellierungen sind hierbei nicht notwendig. Lediglich ein grundlegendes Verständnis über die hydrologischen Ereignisse sind für die Arbeit ausreichend. Gegeben der großen Anzahl von möglichen *Deep Learning* Netzarchitekturen gestaltet sich die passende Erstellung einer geeigneten Netzstruktur schwierig. In dieser Arbeit wird daher konkret die Architektur des LSTM Netzes mit der des kombinierten CNN-LSTM Netzes verglichen, um eine Aussage über ihre Genauigkeit und Eignung zur Niederschlagabflussmodellierung zu treffen.

Der erste Teil widmet sich der Erläuterung der hier verwendeten Methoden. Hierbei wird auf die eingesetzten Netze und Datensätze eingegangen. Darauf folgt eine genaue Vorstellung der Aufbereitung der Daten, welche sich aus Extrahierung der Merkmale und Anpassung der Form (engl. shape) zusammensetzt. Anschließend wird auf den spezifischen, in dieser Arbeit eingesetzten Netzaufbau eingegangen. Abschließend folgt die Auswertung der einzelnen Modelle und deren Vergleich betreffend aller Einzugsgebiete mittels der Verlustfunktion und der Ermittlung des *Nash Sutcliffe model Efficiency coefficient*.

Diese Vorgehensweise soll schließlich die Empfehlung hinsichtlich der in Zukunft präferiert verwendeten Netzarchitektur zur Vorhersage der Niederschlagsabflussmodellierung ermöglichen.

1. *Methoden*
   1. *Netzarchitekturen*
      1. *Long Short-Term Memory (LSTM)*

*Long Short-Term Memory* Netzwerke, kurz LSTM, eignen sich hervorragend für die Analyse sequenzieller Zeitreihen. Sie sind Teil der sogenannten *rekurrenten neuronalen Netze* (RNN). Grundlegend besteht der Vorteil der LSTM in deren Zusammensetzung aus einem *Forget Gate*, einem *Input Gate* und einem *Output Gate*, wie Abbildung 1 darstellt. Dabei entscheidet ersteres über das Vergessen oder Behalten von Informationen der inneren Zelle. Hierzu wird der verborgene Zustand zum Zeitpunkt t-1 und der Zustand zum Zeitpunkt t mit seinen spezifischen Gewichten multipliziert und anschließend addiert. Die darauf angewandte *Sigmoide Aktivierungsfunktion* führt zu einem Wert zwischen 0 und 1, welcher mit dem vorherigen Zellzustand zum Zeitpunkt t-1 multipliziert wird. Es gilt für das *Forget Gate*:

Formel 1: Forget Gate [[1]](#footnote-2)

Grundsätzlich führt ein Wert nahe 0 zum Vergessen und nahe 1 zum Behalten der Informationen. Das *Input Gate* wiederum bestimmt die Informationen, welche neu in die Zelle einfließen sollen. Hierbei handelt es sich um die gleiche, oben abgebildete Formel, jedoch hat dieses Gate seine eigenen spezifischen Gewichtungen, sodass gilt:

Formel 2: Input Gate[[2]](#footnote-3)

Der Vektor nimmt einen Wert zwischen 0 und 1 ein, wobei Werte nahe 1 den Eingang neuer Informationen und nahe 0 deren nicht stattfindende Übernahme in die Zelle darstellen. Der neue Zellenwert zum Zeitpunkt t wird separat errechnet, dabei gilt:

Formel 3: Zellzustand [[3]](#footnote-4)

Hierdurch kann der neue Zellenzustand, welcher sich aus dem vorherigen und dem neuen Zustand ergibt, wie folgt errechnet werden:

Formel 4: neuer Zellzustand [[4]](#footnote-5)

Darüber hinaus bestimmt das *Output Gate* den Informationsfluss aus der Zelle heraus.[[5]](#footnote-6), [[6]](#footnote-7) Es ergibt sich der neue Eingabewert für die nächsthöhere Zelle und der verborgene Zustand für den nächsten Zeitschritt:

Formel 5: Output Gate [[7]](#footnote-8)

Formel 6: Hidden State [[8]](#footnote-9)

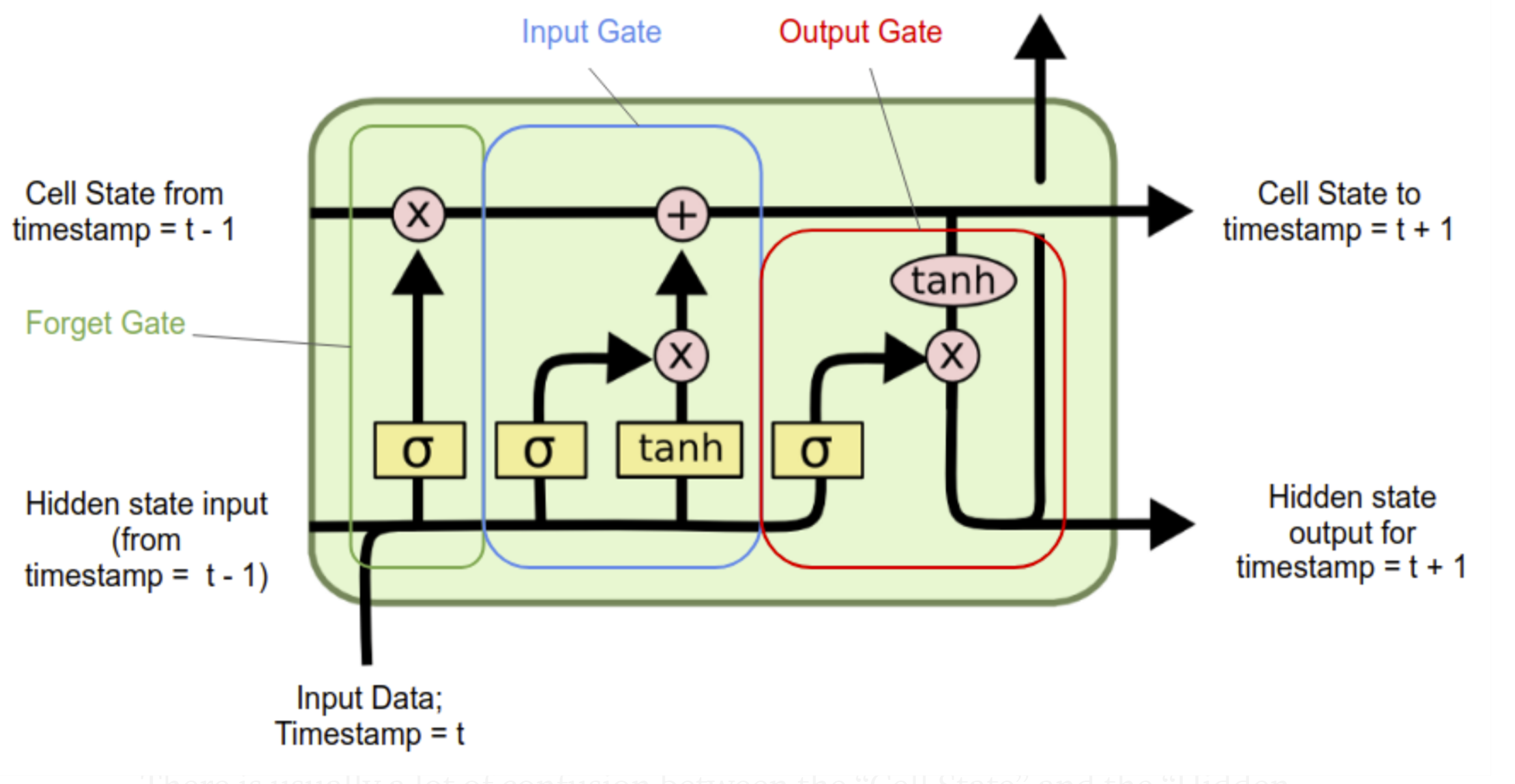


Abbildung 1: Aufbau einer LSTM Zelle [[9]](#footnote-10)

* + 1. *Convolutional Neuronal Network* (CNN)

Ein *Convolutional Neuronal Network*, kurz CNN, besteht hauptsächlich aus den drei Komponenten *Convolutional Layer* (CL), *Pooling-Layer* (PL) und *Fully-Connected-Layer* (FCL). Ersterer extrahiert bedeutsame Merkmale aus dem zu beobachtenden Bereich. Hierbei wird über die Bildmatrix eine sogenannte Faltungsmatrix, (*Filter Kernel*) gelegt, wodurch aus den überlappenden Matrixwerten das Produkt gebildet wird, siehe Abbildung 2. Anschließend werden die Produkte der überlappenden Matrixwerte summiert. Hierdurch wird die rechnerische Ermittlung der Aktivität eines Neurons möglich.

Ziel eines *Pooling-Layer* ist die Beschränkung auf die wichtigsten Informationen. In diesem Anwendungsfall werden spezielle Max-PL verwendet. Aus jedem betrachteten Matrix-Ausschnitt wird der größte Wert entnommen und alle anderen Werte verworfen.[[10]](#footnote-11) [[11]](#footnote-12) Die erhebliche Reduktion der Anzahl an Verbindungen zwischen den *Convolutional Layer* ermöglicht eine schnellere Berechnung unter geringerem Aufwand.

Der Vorgang der Kombination von *Convolutional Layer* und *Pooling-Layer* kann mehrfach wiederholt werden. Schließlich erfolgt die Implementierung eines *Fully-Connected-Layer* durch die Übergabe des Resultats der vorangegangen Schichten als Eingabe an ebendieses. Die abschließende Schlussfolgerung sowie die tatsächliche Ermittlung des Ergebnisses erfolgen an dieser Stelle.[[12]](#footnote-13) [[13]](#footnote-14)

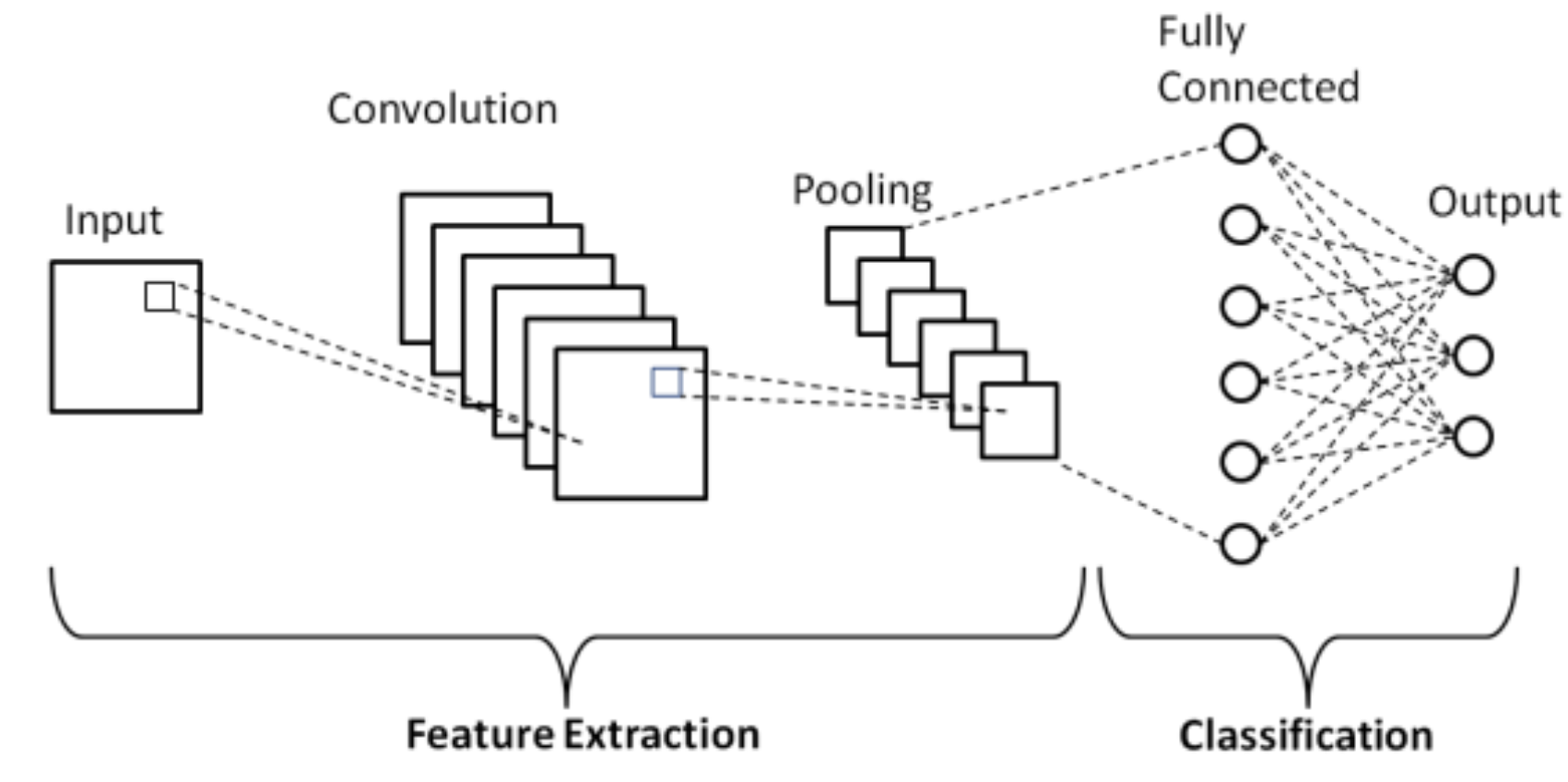


Abbildung 2: Aufbau eines CNN [[14]](#footnote-15)

* + 1. *CNN-LSTM*

Die Kombination eines *Convolutional Neuronal Network* mit einem *Long Short-Term Memory* Netz (CNN-LSTM) ermöglicht die Einbettung der Merkmalsextraktion in einen zeitlichen Kontext.[[15]](#footnote-16) Mittels dieser Methode werden die Vorteile der beiden Netzarchitekturen zusammengeführt. Hierfür werden zunächst CL und PL verwendet, auf welche LSTM Layer folgen. In der letzten Schicht wird ein *Fully-Connected-Layer* zur Ermittlung des Resultats verwendet.

* 1. *Kompilierung*

Als Verlustfunktion (engl. *Loss Function*) wird der *Mean Squared Error* (kurz MSE, deut. mittlere quadratische Abweichung) verwendet, welcher insbesondere für Regressionsanalysen herangezogen wird. Hierbei wird die Ableitung des Vorhersagefehlers durch die Quadrierung der Differenz zwischen tatsächlichem und vorhergesagtem Wert möglich.[[16]](#footnote-17)

Formel 7: Mean Squared Error [[17]](#footnote-18)

Zur Reduzierung des Verlusts wird die *Adaptive Moment Estimation* (kurz Adam) genutzt. Diese Optimierungsfunktion stellt ein effizientes, für seine geringe Speicherbelastung bekanntes Gradientenverfahren dar.[[18]](#footnote-19) Es wurden für alle Versuche die bereits in *Keras* vorimplementierten Standardeinstellungen übernommen.[[19]](#footnote-20) Die Standardeinstellungen haben sich in verschiedenen Problemansätzen bewährt, können aber unter Umständen angepasst werden. Hierfür wird auf die angeführte Lektüre verwiesen.[[20]](#footnote-21)

* 1. *Datensätze*

Das Training der Modelle erfolgt mithilfe unterschiedlicher Datensätze. Hierbei kommen der *Camels* US-Datensatz und der *Daymet* US-Datensatz zum Einsatz. Zur Gewährleistung eines erfolgreichen Vergleichs der beiden Netzarchitekturen werden gleiche Einzugsgebiete unterschiedlicher Klimazonen aus dem nordamerikanischen Kontinent berücksichtigt. Für ein feuchtes Klima wird Eureka in Kalifornien, als gemäßigtes Gebiet Seattle in Washington und als Beispiel eines ariden Klimas Mesa in Arizona berücksichtigt. Im Folgenden werden die Datensätze erläutert.

* + 1. *Daymet US-Datensatz*

Der Datensatz wird von meteorologischen Beobachtungen aus Nordamerika, Hawaii und Puerto Rico abgeleitet, wobei meteorologische Kenndaten auf ein 1x1 km Gitter projiziert werden. Jedes dieser Gitter enthält einen Kennwert zu Niederschlag, Maximal- und Minimaltemperatur, Dampfdruck, Kurzwellenstrahlung, Schneewasseräquivalenz, sowie Sonnenstunden pro Tag. Der Zeitraum der Datenerfassung bezieht sich auf die Jahre 1980 bis 2014. Jedes Jahr umfasst stets 365 Tage, wobei für Schaltjahre der 29. Februar, anstelle des 31.Dezembers berücksichtigt wird.[[21]](#footnote-22)

* + 1. *Camels US-Datensatz*

Der *Camels* Datensatz umfasst 671 Einzugsgebiete der USA und wird aus den Datensätzen von *Daymet*, *NLDAS* und *Maurer* abgeleitet.[[22]](#footnote-23) Dabei werden die sechs Hauptklassen Topologie, Klima, Abflussstrom, Geologie, Landbedeckung und Boden unterschieden. Zur Abflussmodellierung stehen die gleichen Kennwerte, wie für den bereits erwähnten *Daymet* US-Datensatz zur Verfügung. Jedoch werden die Kennwerte eines gesamten Einzugsgebiets gemittelt und nicht in 1x1 km Gitter gegliedert, sodass für jeden Kennwert nur ein einziger Wert des gesamten Einzugsgebietes zur Verfügung steht.[[23]](#footnote-24) Im Detail kann somit nicht nachvollzogen werden, an welcher Koordinate welcher Wert tatsächlich aufgetreten ist.[[24]](#footnote-25) Die genannten Eingangsdaten beschreiben in diesem Beispiel die Ausgangsdaten, sprich die Abflussströme. Der *Camels* Datensatz bezieht sich ebenfalls auf den Zeitraum der Jahre 1980 bis 2014. Hierin sind Schaltjahre einbezogen, sodass der Datensatz eines Jahres entweder aus 365 oder 366 Tagen besteht. Dies ist für die Aufbereitung und Vergleichbarkeit der Daten von essenzieller Bedeutung.

* + 1. *Aufbereitung der Datensätze*

Die Datensätze werden unterschiedlichen Prozessen unterzogen, um ihre Vergleichbarkeit sicherzustellen. Für beide Datensätze gilt, dass die Kennwerte der vergangenen 50 Tage den Abflussstrom, somit am Tag 50+1, modellieren sollen. Als Features werden hierbei Niederschlag, Maximal- und Minimaltemperaturen, Kurzwellenstrahlung und Dampfdruck verwendet. Zudem ist insbesondere für Schaltjahre zu beachten, dass der 31. Dezember der betrachteten Jahre im Daymet Datensatz nicht miteinbezogen wird. Dementsprechend muss der Abflussstrom an diesem Datum unberücksichtigt bleiben. Betrachtet wird der Zeitraum der Jahre 1980 bis einschließlich 1995. Anschließend wird der gesamte Datensatz, also die Eingangs- (Features) und die Ausgabewerte (Abflussstrom), normiert. Hierdurch befinden sich alle Werte im Skalenbereich 0 bis 1. Die sequenziell eingelesenen Daten werden so zugeschnitten, dass diese entsprechend in ein CNN bzw. LSTM-Layer eingeführt werden können. Am Ende besitzen die Daten für das CNN die Form (Stichproben, Zeitschritte, Länge, Breite, Eigenschaften). Der LSTM-Layer wird durch die Form (Stichproben, Zeitschritte, Merkmale) beschrieben. Nach Erstellung der richtigen Form erfolgt die Mischung der Daten, sodass die sequenzielle Reihenfolge der Daten, entsprechend von 1980 bis 1995 unterbunden wird und nun dem Zufall entspricht. Der geographische Standort der hier betrachteten Städte wird in Abbildung 3 erkennbar.

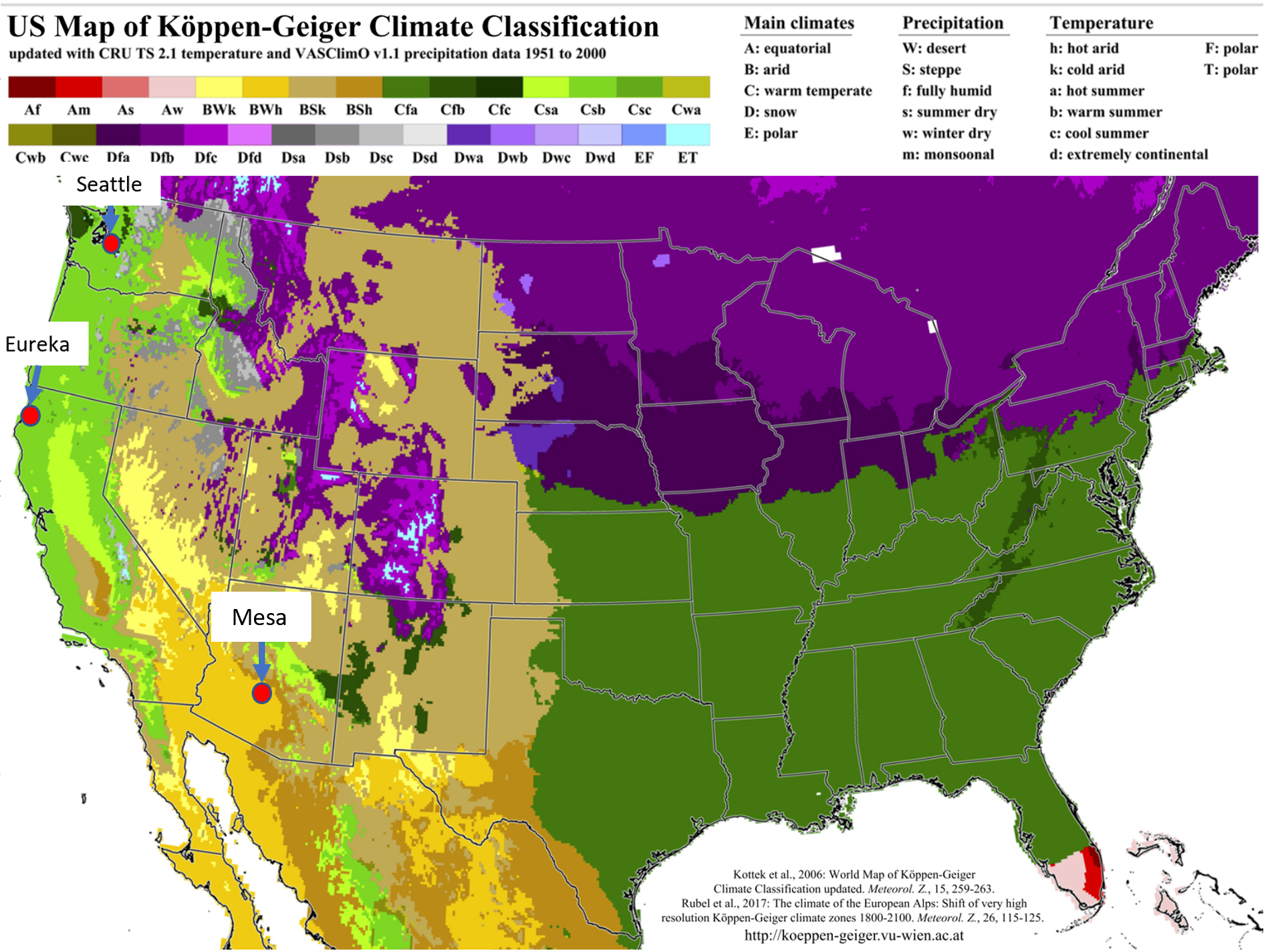


Abbildung 3: Klimakarte der USA mit den Standorten der hier verwendeten Einzugsgebiete [[25]](#footnote-26)

* 1. *Netzarchitektur*

Im Folgenden wird in Kürze auf die konkrete hier angewandte Netzstruktur eingegangen. Zur Gewährleistung der Vergleichbarkeit wurde für die beiden verwendeten Ansätze die gleiche Netzstruktur auf die jeweiligen Einzugsgebiete angewandt.

* + 1. *Architektur des Long Short-Term Memory Netzes*

Für das LSTM Netzwerk wurden drei aufeinander folgende LSTM Layer verwendet. Hierbei werden für die erste 50, für die zweite 30 und für die dritte Schicht 10 Hidden States verwendet. In der letzten Schicht wird ein sogenanntes *Dense Layer* eingesetzt. Trainiert werden alle CNN-LSTM Modelle mit 200 Epochen und einer Batchsize von 64.

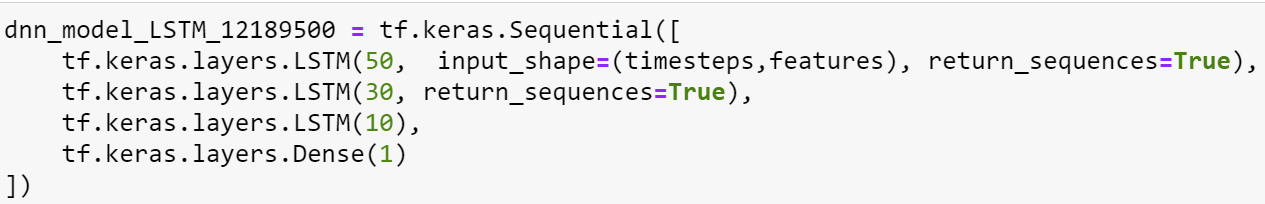


Abbildung 4: Netzarchitektur der LSTM Modelle

* + 1. *Architektur des CNN-LSTM Netzes*

Für das CNN-LSTM Netzwerk wird in der ersten Schicht ein zeitlich verteilter CL genutzt auf den ein PL folgt. Dies wiederholt sich ein weiteres Mal. Darauf folgt ein *Flatten Layer*, um die Eingabe in ein LSTM Layer zu ermöglichen. Dabei umschließt der erste LSTM Layer 30 und der darauffolgende 10 Hidden States. Abgeschlossen wird das Netzwerk von einem *Dense* *Layer*. Alle CNN-LSTM Modelle werden mit 100 Epochen und einer Batchsize von 64 trainiert.

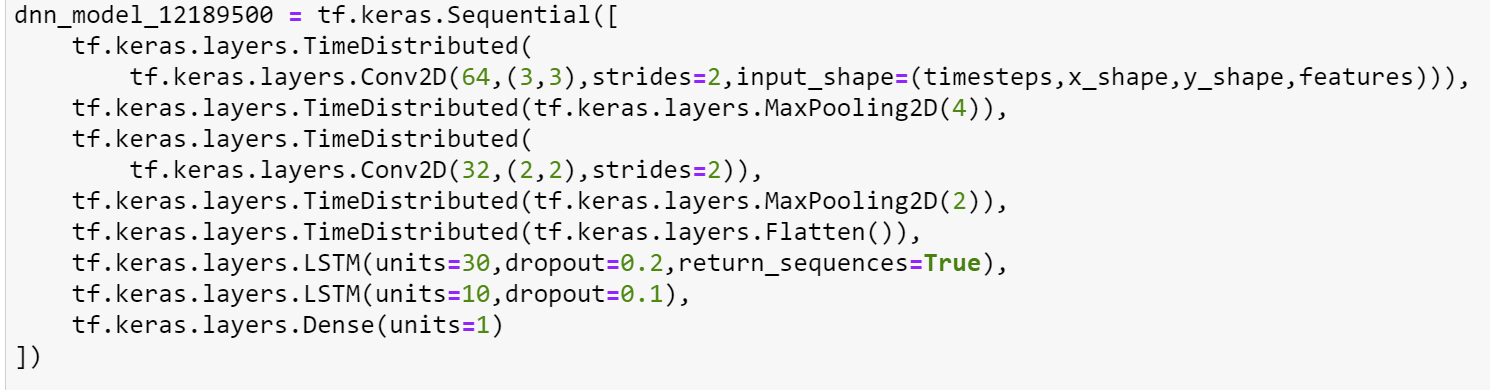


Abbildung 5: Netzarchitektur der CNN-LSTM Modelle

* 1. *Nash Sutcliffe model Efficiency coefficient* *(NSE)*

Der NSE bezeichnet ein Kriterium zur Bewertung und zum Vergleich hydrologischer Modelle. Mittels diesem werden reale mit simulierten Werten hinsichtlich ihrer Übereinstimmung überprüft. Der Skalenbereich liegt zwischen 1 und minus unendlich. Dabei zeigt der Wert 1 eine perfekte Übereinstimmung zwischen realen und simulierten Werten an. Ein Wert gleich 0 bedeutet, dass das Modell die gleiche Darstellungskraft, wie der Schnitt aller Beobachtungen vorweist. Werte kleiner 0 zeigen die ungenügende Aussagekraft der Vorhersage an. Die Formel zur Berechnung des NSE lautet:

Formel 8: Nash Sutcliffe model Efficiency coefficient

Hierbei wird der simulierte Abfluss mit dem tatsächlichen Wert zum Zeitpunkt t subtrahiert und quadriert. Anschließend folgt die Division aus dem Quadrat von der Subtraktion zwischen und , wobei den Mittelwert aller Abflussströme darstellt. Die aus allen Werten gebildete Gesamtsumme wird anschließend von 1 subtrahiert.[[26]](#footnote-27)

1. *Resultate*

Im Folgenden werden die erzielten Ergebnisse dargestellt und erläutert. Zur Gewährleistung der Vergleichbarkeit wurde auf eine individuelle Optimierung für die einzelnen Einzugsgebiete verzichtet. Dadurch ergibt sich ein unausgeschöpftes individuelles Potential für die einzelnen Einzugsgebiete. In allen Abbildungen ist die Epoche auf der X-Achse und der MSE auf der Y-Achse dargestellt.

Zu Beginn werden die Verlustkurven beginnend bei den CNN-LSTM Modellen erläutert. In Abbildung 6 ist die Kurve für das Einzugsgebiet Mesa zu erkennen. Dabei sind der Trainings- und Validierungsverlust nahezu überlappend. Weder ein Over- noch ein Underfit sind hierbei ersichtlich.

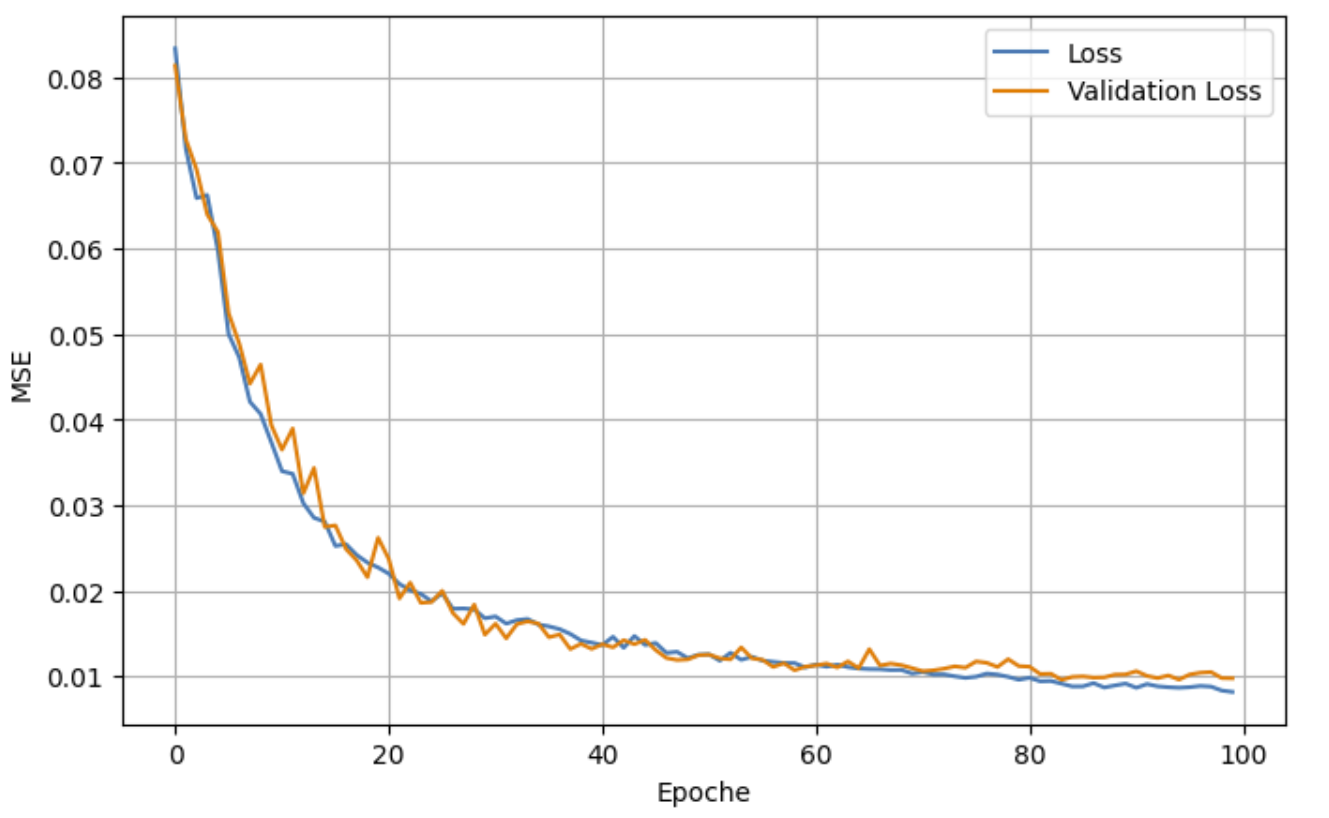


Abbildung 6: Verlustfunktion des Einzugsgebiets Mesa für das CNN-LSTM Modell

Die nachfolgende Abbildung zeigt die Verlustfunktion des Einzugsgebietes Eureka. Auch hier ist eine erfolgreiche Annäherung der Verlustfunktionen zu sehen. Lediglich am Ende trennen sich die Verlustfunktionen, jedoch ist ein klares Auseinandergehen dieser nicht erkennbar, weshalb insgesamt eine deutliche Annäherung vorhanden ist.

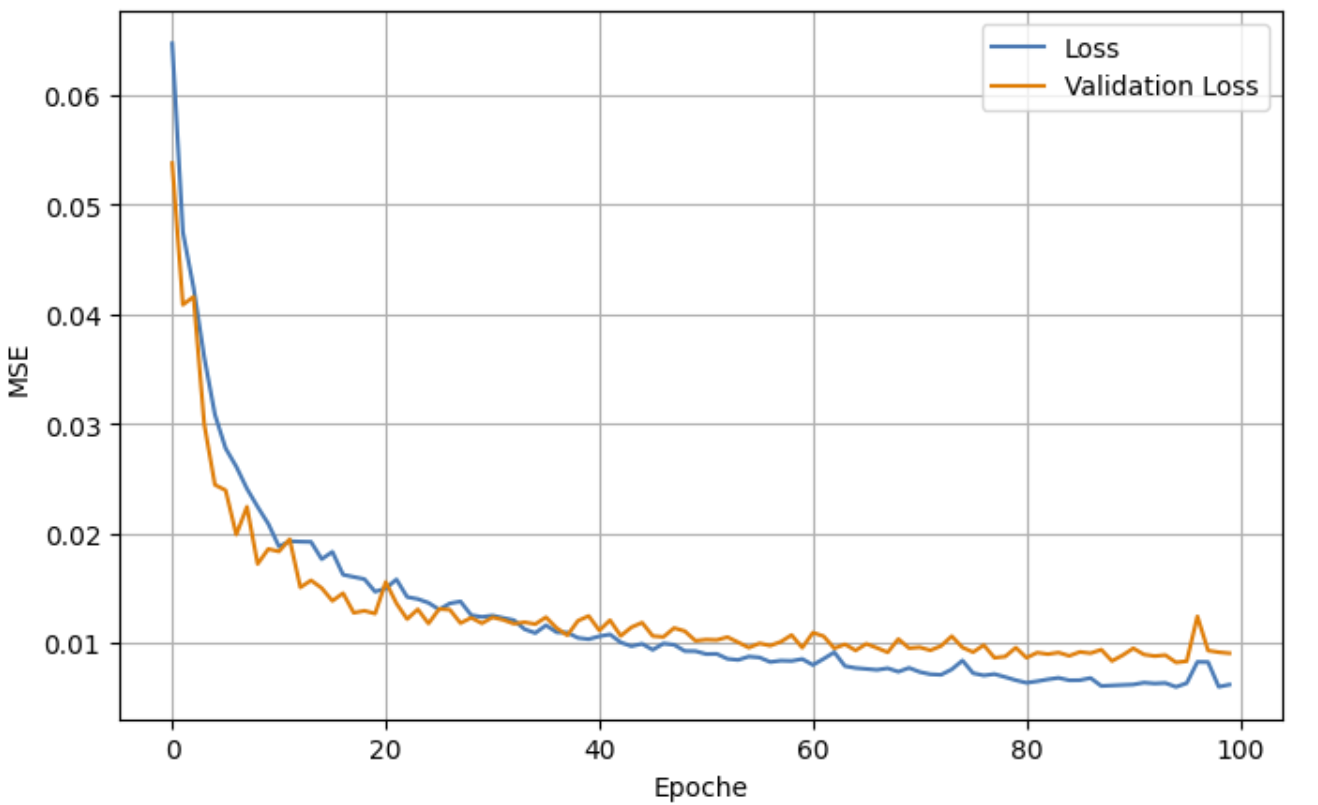


Abbildung 7: Verlustfunktion des Einzugsgebiets Eureka für das CNN-LSTM Modell

Für das letzte CNN-LSTM Modell wird das Einzugsgebiet Seattle betrachtet. Insgesamt scheinen sich beide Verlustfunktionen in einem konstanten nicht mehr verändernden Bereich zu entwickeln.

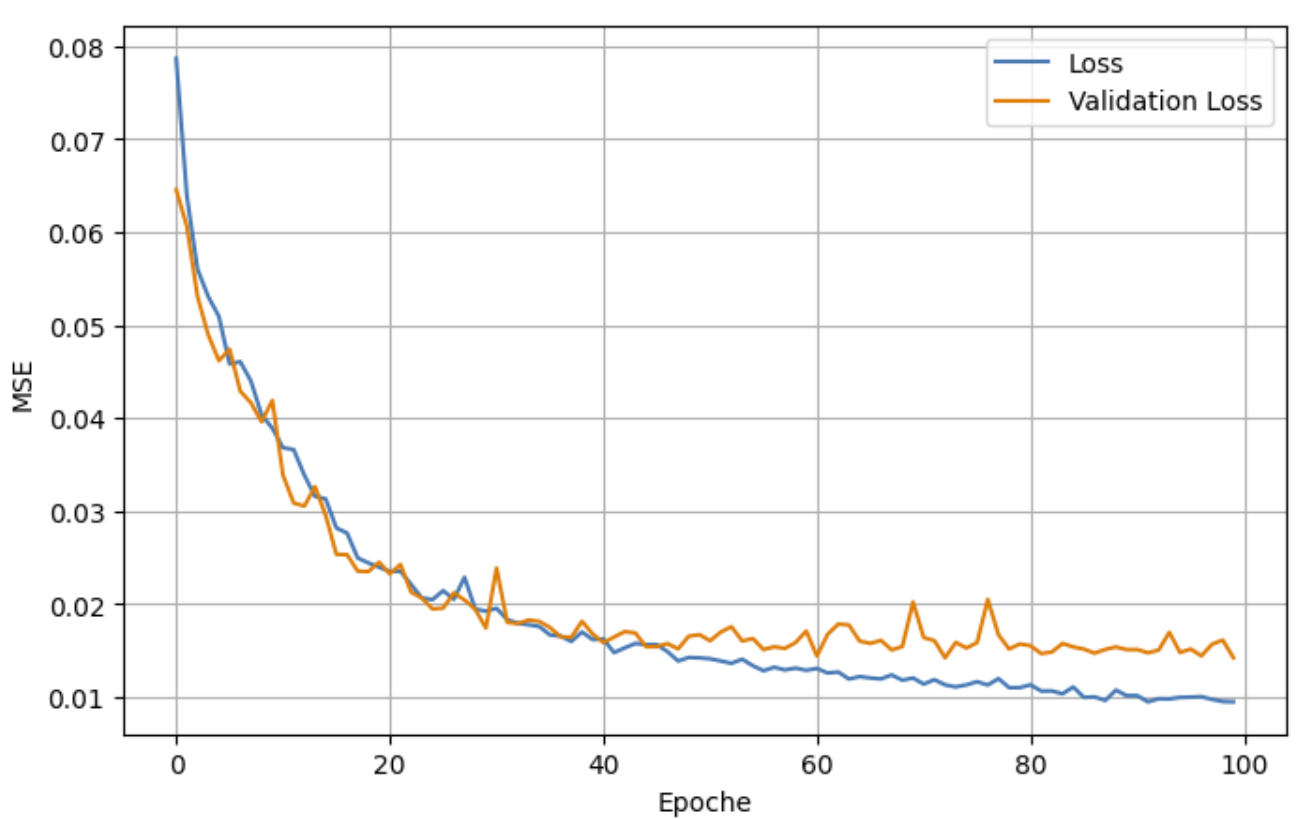


Abbildung 8: Verlustfunktion des Einzugsgebiets Seattle für das CNN-LSTM Modell

In den kommenden Abbildungen wird auf die Verlustfunktionen der reinen LSTM Modelle eingegangen. Hierzu wird erneut bei Mesa begonnen.

Abbildung 9 lässt eine hohe Anpassung der Verlustfunktionen und einen nahezu konstanten Abstand beider Funktionen zum Ende des Trainings hin erkennen. Es sollte beachtet werden, dass die Skala den Abstand im Bereich 10-4 abbildet, wodurch der Abstand graphisch größer wirkt, als dieser tatsächlich ist. Sichtbar wird ein Unterschied der Verlustfunktionen der CNN-LSTM Modelle im Bereich 10-3.

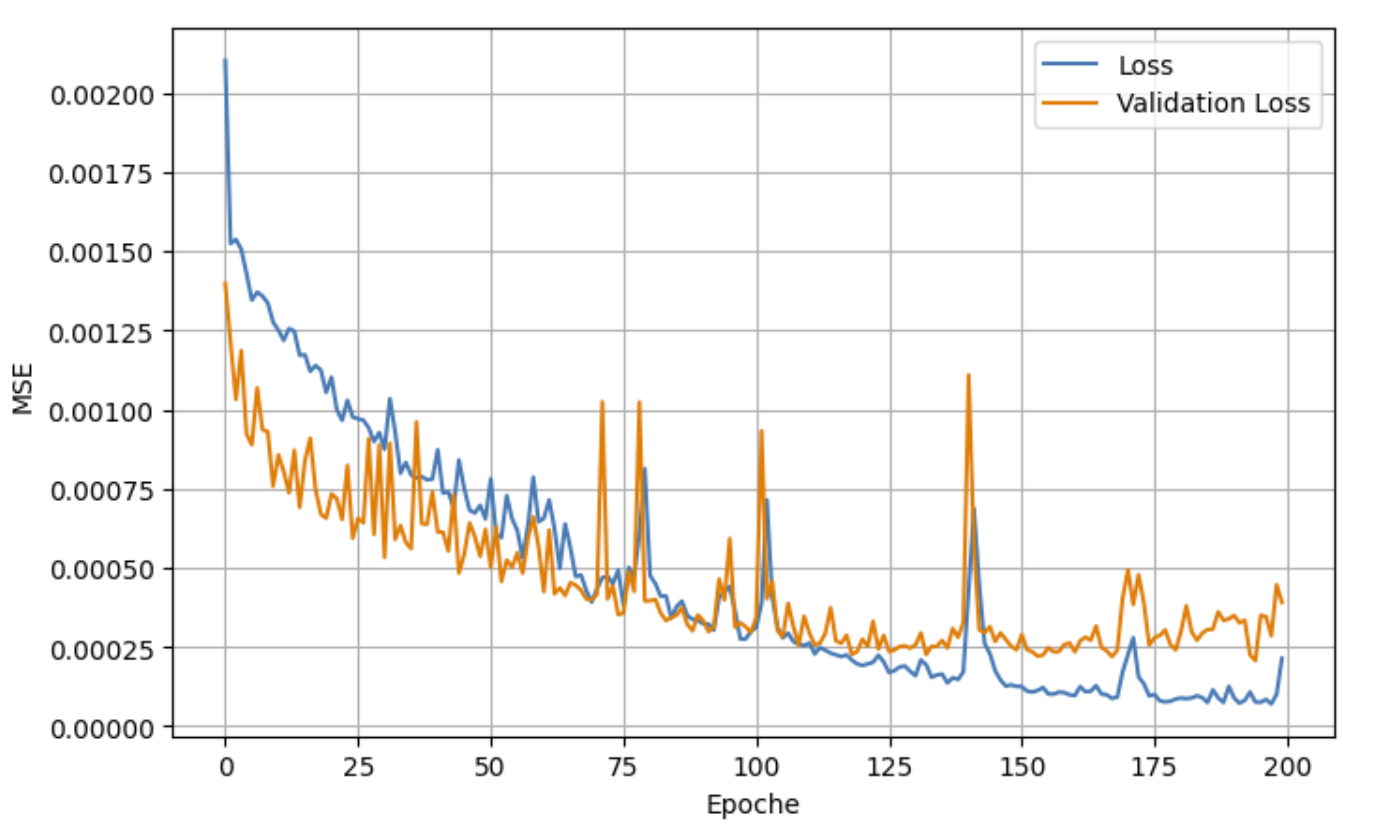


Abbildung 9: Verlustfunktion des Einzugsgebiets Mesa für das LSTM Modell

Die Verlustfunktion des Einzugsgebietes Eureka zeigt eine ähnliche Entwicklung, wie die des vorherigen Gebietes. Ebenfalls ist eine grundsätzliche erfolgreiche Annäherung erkennbar.

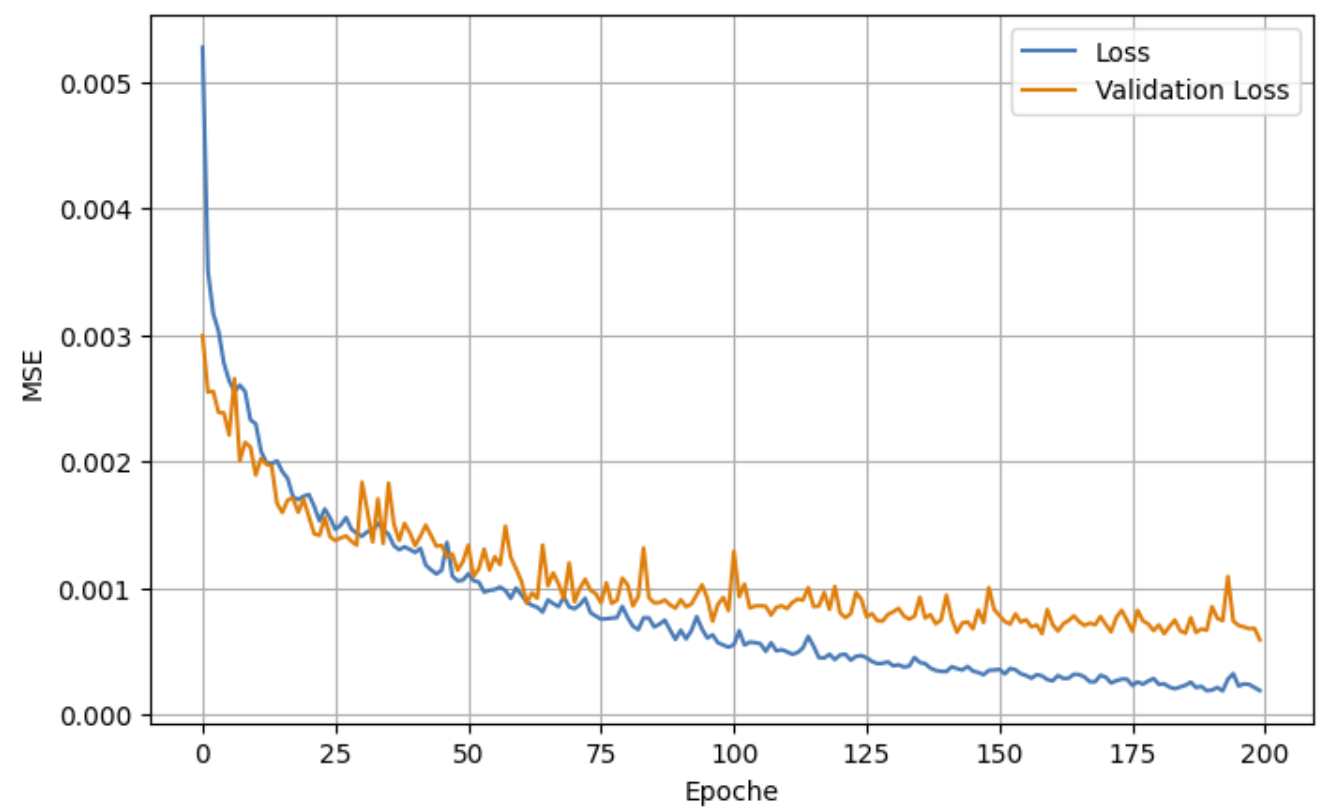


Abbildung 10: Verlustfunktion des Einzugsgebiets Eureka für das LSTM Modell

Zuletzt wird in Abbildung 11 die Verlustfunktion des Gebietes Seattle veranschaulicht. Zu erkennen ist, dass sich die Verlustfunktionen bereits bei Epoche 75 leicht spreizen. Anschließend verbessert sich die Verlustfunktion der Validierungsdaten nicht und stagniert, während die Verlustfunktion der Trainingsdaten weiterhin sinkt. Basierend hierauf wären vermutlich weitere Epochen zur Gewährleistung einer stabilen Verlustfunktion der Trainingsdaten notwendig.

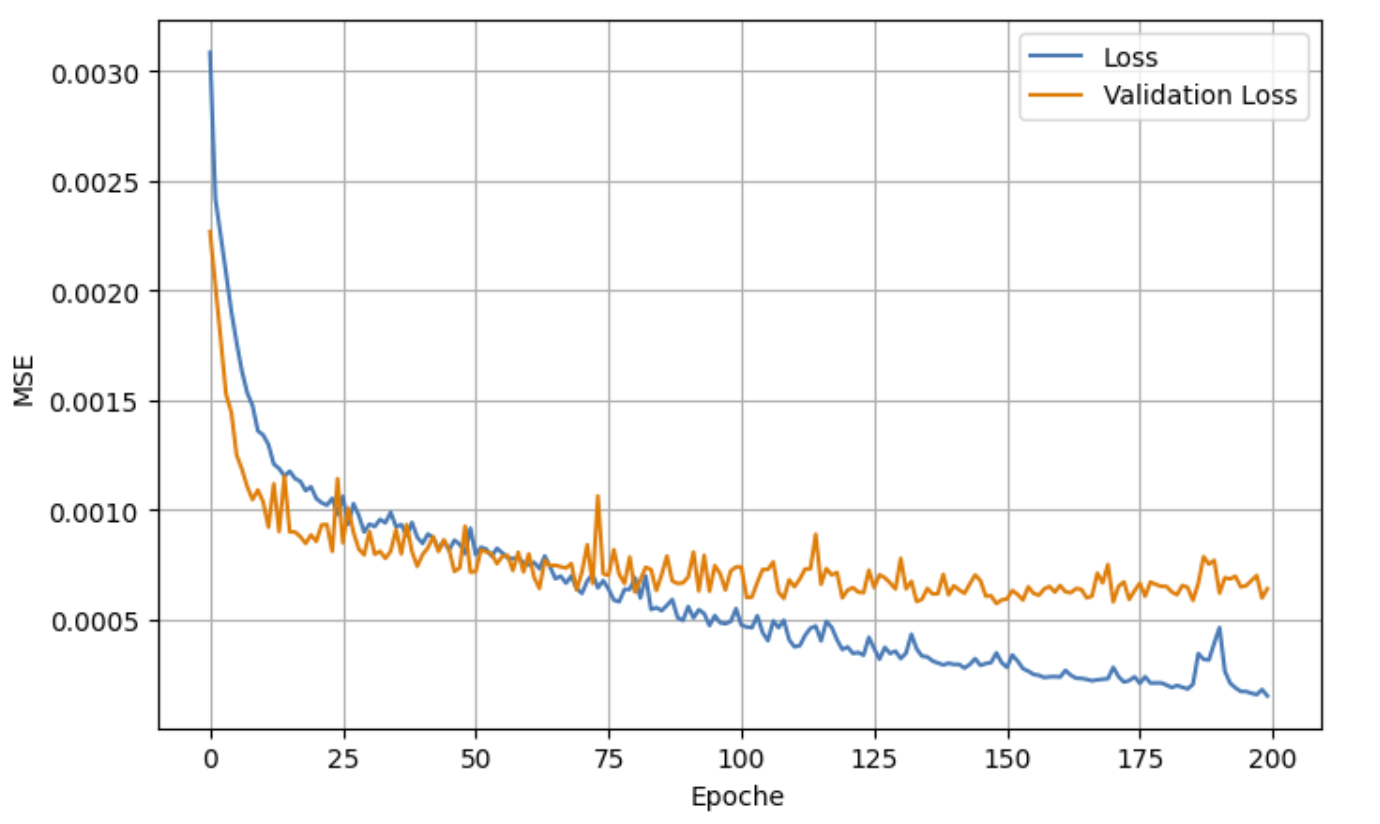


Abbildung 11: Verlustfunktion des Einzugsgebiets Seattle für das LSTM Modell

Insgesamt befinden sich alle Verlustfunktionen in einem überzeugenden Bereich und weisen eine adäquate Abbildung der Validierungsdaten durch das Modell auf.

Zuletzt wird die Bewertung der Modelle mittels Ermittlung des NSE Wertes für alle Einzugsgebiete vorgenommen. Abbildung 12 lässt ein deutlich zutreffenderes Ergebnis der CNN-LSTM Modelle im Vergleich zu den LSTM Modellen erkennen. Lediglich für das Einzugsgebiet Eureka weisen die Modelle einen kaum nennenswerten Unterschied auf. Das CNN-LSTM Modell liegt mit einem Wert von 0.8866 nur um 0.0023 Punkte treffender, als das LSTM Modell, welches einen Wert von 0.8843 erreicht. Für das Einzugsgebiet Mesa ergibt sich ein um etwa 0.07 Punkte präziseres Ergebnis für das CNN-LSTM Modell mit einem Wert von 0.8882 im Vergleich zum LSTM Modell, welches 0.8185 erreicht. Noch deutlicher ist das Ergebnis für das Einzugsgebiet Seattle. Hier weist das CNN-LSTM Modell ein um knapp 0.091 erfolgreicheres Resultat mit einem Ergebnis von 0.8601 gegenüber dem LSTM Modell mit einem Wert von 0.7688 auf. Insgesamt schneidet das Einzugsgebiet Seattle im direkten Vergleich zu den anderen Einzugsgebieten in beiden Modellen jeweils schwächer ab. Die Ergebnisse zeichnen sich unter Berücksichtigung eines perfektes NSE Wertes von 1 durch eine hohe Aussagekraft und Repräsentativität aus.

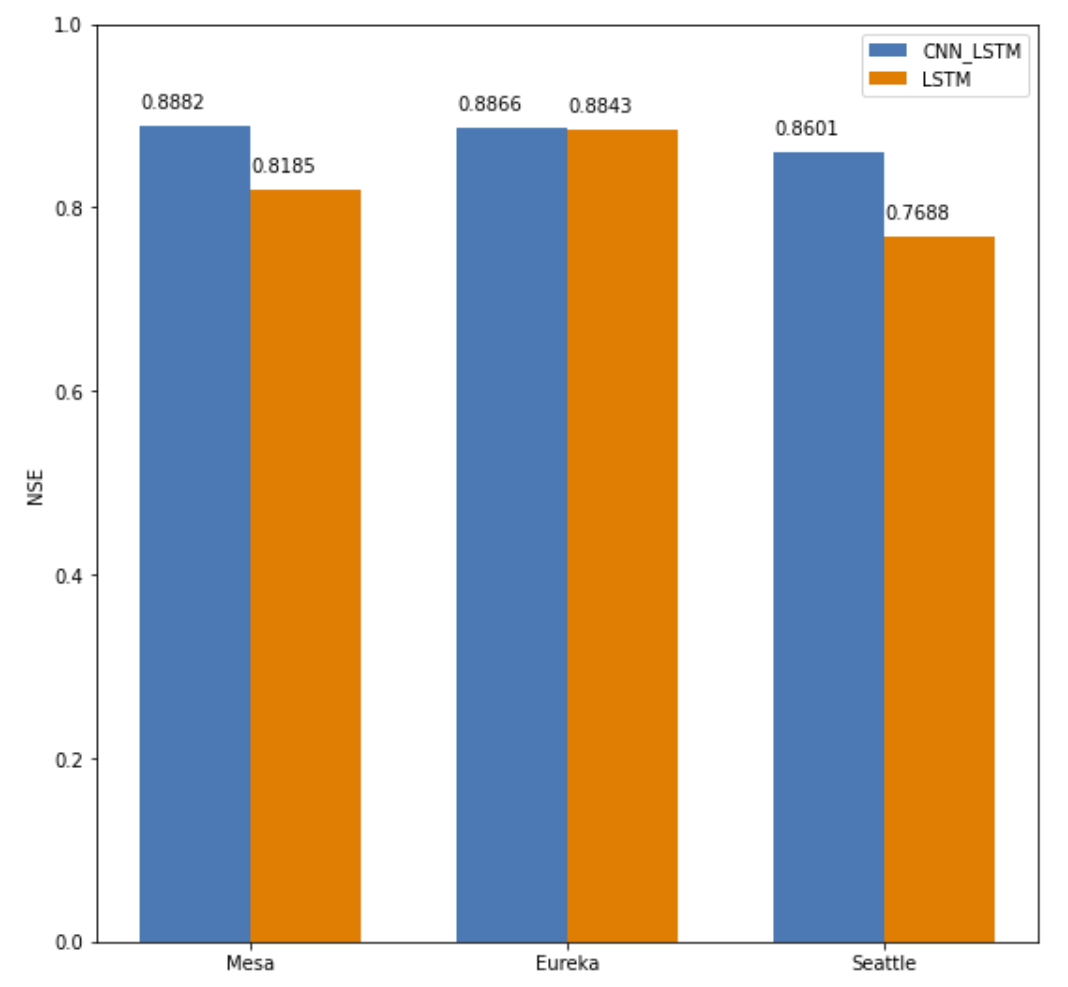


Abbildung 12: NSE Werte für alle Modelle

In Abbildung 13 und 14 werden die simulierten Ergebnissen mit den tatsächlich beobachteten Werten graphisch für jedes Einzugsgebiet dargestellt. Zwecks Überprüfbarkeit der Ergebnisqualität werden 100 zufällige Werte gegenübergestellt. In der ersten Abbildung sind die CNN-LSTM Einzugsgebiete zu sehen. Für den Großteil der Fälle ist eine Überlappung beider Graphen ersichtlich. Dies deckt sich mit den erfolgreichen Resultaten der NSE Werte, welche eine erfolgreiche Modellierung des Abflusses bestätigen. In Abbildung 14 werden die Ergebnisse für die LSTM Modelle dargestellt. Bezüglich der jeweiligen Einzugsgebiete sind auch hier größtenteils Überlappungen zu erkennen, was die positiven erreichten NSE Werte untermauert.

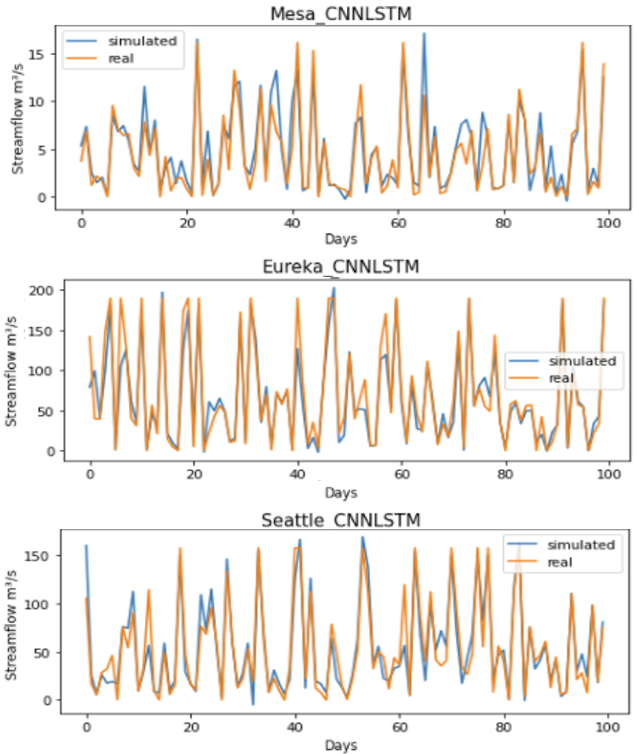


Abbildung 13: Graphik zum Vergleich der simulierten und tatsächlich beobachteten Werte für die CNN-LSTM Modelle

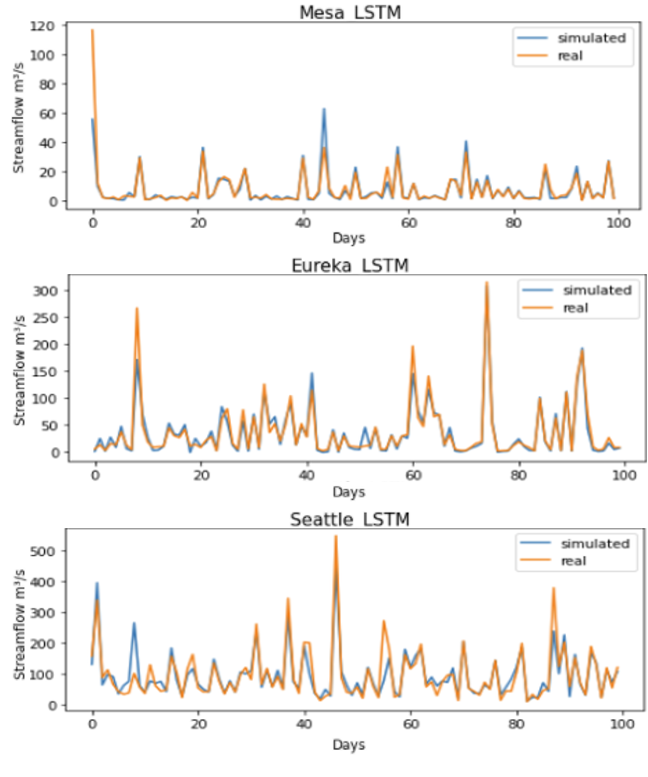


Abbildung 14: Graphik zum Vergleich der simulierten und tatsächlich beobachteten Werte für die LSTM Modelle

1. *Zusammenfassung*

Das übergeordnete Ziel dieser Arbeit bestand im Vergleich der hier eingesetzten Netzarchitekturen, um wesentliche Aussagen über ihre Qualität zu treffen. Spezifisch wurden Standard *Long Short-Term Memory* (LSTM) Netze mit *Convolutional Neuronal Network* kombinierten LSTM Netzen (CNN-LSTM) verglichen. Für das Fallbeispiel der Vorhersagetreffung von Zeitreihen einer Niederschlagabflussmodellierung wurden Daten der Camels und Daymet Datensätze zu drei Einzugsgebieten unterschiedlichen Klimas herangezogen. Hierbei waren in zwei von drei Einsatzgebieten deutliche Unterschiede, basierend auf dem NSE Kriterium, festzustellen. Lediglich im Gebiet Eureka konnte kein nennenswerter Unterschied der Ergebnisse der beiden Netzarchitekturen festgestellt werden, allerdings auch keine Überlegenheit des einfachen LSTM Modells. Grundsätzlich eignen sich beide Modelle zur erfolgreichen Abflussmodellierung. Basierend auf den hier gewählten Auswertungsmethoden zeichnet sich jedoch eine höhere Genauigkeit der CNN-LSTM Modelle ab. Diese erfordern zwar einen höheren Zeit- und Rechenaufwand, werden jedoch aufgrund ihrer höheren Modellqualität und Präzision empfohlen. Darüber hinaus bietet sich für jedes Einsatzgebiet ein individuelles Potential der Verbesserung des Ergebnisses. Dieses Potential ergibt sich durch die jeweilige Optimierung der Netzarchitektur, welche hier für alle Gebiete einheitlich ausgewählt wurde.

# *Referenzen*

Arvix: Gu, Jiuxiang [u.a.]: Recent Advances in Convolutional Neural. Networkshttps://arxiv.org/pdf/ 1512.07108.pdf%C3%A3%E2%82%AC%E2%80%9A (Abruf: 25.08.2021).

Arvix: Kingma, Diederik P. und Lei Ba, Jimmy: Adam. A Method for stochastic optimization. https://arxiv.org/pdf/1412.6980.pdf (Abruf: 25.08.2021).

Daymet: Daymet Science and Introduction. https://daymet.ornl.gov/overview (Abruf: 25.08.2021).

Htwsaar: Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning. https://www.htwsaar.de/htw/wiwi/fakultaet/personen/profile/selle-stefan/Selle2018e\_ Kuenstliche\_Neuronale\_Netzwerke.pdf (Abruf: 24.08.2021).

Hydrology and Earth System Sciences: Addor, Nans, Newman, Andrew J., Mizukami Naoki und Clark, Martyn P.: The CAMELS data set. Catchment attributes and meteorology for large-sample studies. https://hess.copernicus.org/articles/21/5293/2017/hess-21-5293- 2017.pdf (Abruf: 25.08.2021).

Hydrology and Earth System Sciences: Knoben, Wouter J., Freer ,Jim E. und Woods, Ross A.: Technical note: Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores. https://hess.copernicus.org/articles/23/4323/2019/ (Abruf: 25.08.2021).

Hydrology and Earth System Sciences: Newman, A.J. [u.a]: Development of a large-sample watershed- scale hydrometeorological data set for the contiguous USA: data set characteristics and assessment of regional variability in hydrologic model performance. https://hess.copernicus.org/articles/19/209/2015/ (Abruf: 25.08.2021).

Keras: Adam. https://keras.io/api/optimizers/adam/ (Abruf: 25.08.2021).

Koeppen-Geiger: US Map of Köppen-Geiger Climate Classification. http://koeppen-geiger.vu wien.ac.at/pics/KG\_USA\_5min.pdf (Abruf: 25.08.2021).

MDPI: Song, Wie, Gao, Ghao, Zhao, Yue und Zhao, Yandong: A Time Series Data Filling Method Based

on LSTM. Taking the Stem Moisture as an Example. https://doi.org/10.3390/s20185045. (Abruf: 24.08.2021).

Medium: T.J.J., Ryan: LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras. https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate- conceptual-guide-with-keras-2a650327e8f2 (Abruf: 25.08.2021).

NCAR: CAMELS: Large-Sample Hydrometeorological Dataset. https://ncar.github.io/hydrology/ datasets/CAMELS\_timeseries (Abruf: 25.08.2021).

ResearchGate: Phung, Van Hiep und Rhee, Eun Joo: A High‐Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural- network-CNN-architecture-26\_fig1\_336805909 (Abruf: 25.08.2021).

ResearchGate: Sepp Hochreiter, Jürgen Schmidhuber: Long short-term memory In: Neural Computation (journal), vol. 9, issue 8, S. 1735–1780, 1997. https://www.researchgate.net/publication/13853244\_Long\_Short-term\_Memory (Abruf: 25.08.2021).

ScienceDirect: Indolia, Sakshi, Goswarmi, Anil Kumar, Mishra, S.P., Asopa, Pooja: Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. https://www.sciencedirect.com/science/article/pii/S1877050918308019 (Abruf: 25.08.2021).

SpringerLink: Barzegar, Rahim, Aalami, Mohammed Taghi und Adamowski, Jan: Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. https://doi.org/10.1007/s00477-020-01776-2 (Abruf: 25.08.2021).

SpringerLink: Kratzert, Frederik, Gauch, Martin, Nearing, Grey, Hochreiter, Sepp und Klotz, Daniel: Niederschlags-Abfluss-Modellierung mit Long Short-Term Memory (LSTM). https://link.springer.com/content/pdf/10.1007/s00506-021-00767-z.pdf (Abruf: 25.08.2021).

SpringerLink: Sammut C. und Webb G.I.: Encyclopedia of Machine Learning. Mean Squared Error. https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8\_528 (Abruf: 25.08.2021).

1. Vgl. MDPI: Song, Wie, Gao, Ghao, Zhao, Yue und Zhao, Yandong: A Time Series Data Filling Method Based on LSTM. Taking the Stem Moisture as an Example. https://doi.org/10.3390/s20185045. S. 3 (Abruf: 24.08.2021). [↑](#footnote-ref-2)
2. Vgl. Htwsaar: Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning. https://www.htwsaar.de/htw/wiwi/fakultaet/personen/profile/selle-stefan/Selle2018e\_Kuenstliche\_

   Neuronale\_Netzwerke.pdf. S. 62 (Abruf: 25.08.2021). [↑](#footnote-ref-3)
3. Vgl. Ebd. S. 62. [↑](#footnote-ref-4)
4. Vgl. Ebd. S. 62. [↑](#footnote-ref-5)
5. Vgl. SpringerLink: Kratzert, Frederik, Gauch, Martin, Nearing, Grey, Hochreiter, Sepp und Klotz, Daniel: Niederschlags-Abfluss-Modellierung mit Long Short-Term Memory (LSTM). https://link.springer.com/content/pdf/10.1007/s00506-021-00767-z.pdf (Abruf: 25.08.2021). [↑](#footnote-ref-6)
6. Vgl. ResearchGate: Sepp Hochreiter, Jürgen Schmidhuber: Long short-term memory In: Neural Computation (journal), vol. 9, issue 8, S. 1735–1780, 1997. https://www.researchgate.net/publication/13853244\_Long\_Short-term\_Memory (Abruf: 25.08.2021). [↑](#footnote-ref-7)
7. Vgl. Selle: Künstliche Neuronale Netzwerke und Deep Learning. S. 62. [↑](#footnote-ref-8)
8. Vgl. Ebd. S. 62. [↑](#footnote-ref-9)
9. Medium: T.J.J., Ryan: LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras. https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2 (Abruf: 25.08.2021). [↑](#footnote-ref-10)
10. Vgl. ScienceDirect: Indolia, Sakshi, Goswarmi, Anil Kumar, Mishra, S.P. und Asopa, Pooja: Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. https://www.sciencedirect.com/science/article/pii/S1877050918308019 (Abruf: 25.08.2021). [↑](#footnote-ref-11)
11. Vgl. Arvix: Gu, Jiuxiang [u.a.]: Recent Advances in Convolutional Neural. Networkshttps://arxiv.org/pdf/ 1512.07108.pdf%C3%A3%E2%82%AC%E2%80%9A (Abruf: 25.08.2021). [↑](#footnote-ref-12)
12. Vgl. Arvix: Gu [u.a.]: Recent Advances in Convolutional Neural. Networkshttps://arxiv.org/pdf/ 1512.07108.pdf%C3%A3%E2%82%AC%E2%80%9A (Abruf: 25.08.2021). [↑](#footnote-ref-13)
13. Vgl. ScienceDirect: Indolia, Goswarmi, Mishra und Asopa: Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. https://www.sciencedirect.com/science/article/pii/S1877050918308019 (Abruf: 25.08.2021). [↑](#footnote-ref-14)
14. ResearchGate: Phung, Van Hiep und Rhee, Eun Joo: A High‐Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets.

    https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26\_fig1\_336805909 (Abruf: 25.08.2021). [↑](#footnote-ref-15)
15. Vgl. SpringerLink: Barzegar, Rahim, Aalami, Mohammed Taghi und Adamowski, Jan: Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. https://doi.org/10.1007/s00477-020-01776-2 (Abruf: 25.08.2021). [↑](#footnote-ref-16)
16. Vgl. SpringerLink: Sammut C. und Webb G.I.: Encyclopedia of Machine Learning. Mean Squared Error. https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8\_528 (Abruf: 25.08.2021). [↑](#footnote-ref-17)
17. Vgl. Ebd. [↑](#footnote-ref-18)
18. Vgl. Arvix: Kingma, Diederik P. und Lei Ba, Jimmy: Adam. A Method for stochastic optimization. https://arxiv.org/pdf/1412.6980.pdf (Abruf: 25.08.2021). [↑](#footnote-ref-19)
19. Vgl. Keras: Adam. https://keras.io/api/optimizers/adam/ (Abruf: 25.08.2021). [↑](#footnote-ref-20)
20. Vgl. Arvix: Kingma, Diederik P. und Lei Ba, Jimmy: Adam. A Method for stochastic optimization. https://arxiv.org/pdf/1412.6980.pdf (Abruf: 25.08.2021). [↑](#footnote-ref-21)
21. Vgl. Daymet: Daymet Science and Introduction. https://daymet.ornl.gov/overview (Abruf: 25.08.2021). [↑](#footnote-ref-22)
22. Vgl. NCAR: CAMELS: Large-Sample Hydrometeorological Dataset. https://ncar.github.io/hydrology/ datasets/CAMELS\_timeseries (Abruf: 25.08.2021). [↑](#footnote-ref-23)
23. Vgl. Hydrology and Earth System Sciences: Addor, Nans, Newman, Andrew J., Mizukami Naoki und Clark, Martyn P.: The CAMELS data set. Catchment attributes and meteorology for

    large-sample studies. https://hess.copernicus.org/articles/21/5293/2017/hess-21-5293-2017.pdf (Abruf: 25.08.2021). [↑](#footnote-ref-24)
24. Vgl. Hydrology and Earth System Sciences: Newman, A.J. [u.a]: Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: data set characteristics and assessment of regional variability in hydrologic model performance. <https://hess.copernicus.org/> articles/19/209/2015/ (Abruf: 25.08.2021). [↑](#footnote-ref-25)
25. Koeppen-Geiger: US Map of Köppen-Geiger Climate Classification. http://koeppen-geiger.vu-wien.ac.at/pics/KG\_USA\_5min.pdf (Abruf: 25.08.2021). [↑](#footnote-ref-26)
26. Vgl. Hydrology and Earth System Sciences: Knoben, Wouter J., Freer, Jim E. und Woods, Ross A.: Technical note: Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores. https://hess.copernicus.org/articles/23/4323/2019/ (Abruf: 25.08.2021). [↑](#footnote-ref-27)