

HS'BI

Hochschule
Bielefeld
University of
Applied Sciences



iDaS

Institute for Data Science
Solutions of HSBI

Evaluierung der Leistungsfähigkeit von LSTM- Modellen für die Approximation physikalischer Systeme

Schriftenreihe des Institute for Data Science Solutions

Nr. 2/ 2024

DOI: <https://doi.org/10.60802/sidas.2024.2>
ISSN: 2943-3509

Dieses Dokument ist lizenziert gemäß CC BY <https://creativecommons.org/licenses/by/4.0/> Aus-
genommen von der Lizenz sind alle Wort-/Bildmarken und Logos



Die Autor*innen tragen die Verantwortung für die Einhaltung der urheberrechtlichen Bestimmungen.
Zum Zeitpunkt

der Drucklegung führten die Verweise auf Internetseiten zu den gewünschten Inhalten. Sollten zu einem späteren
Zeitpunkt die Internetseiten verändert worden sein, distanzieren sich die Autor*innen von den inhaltlichen
Aussagen der Internetseiten.

Evaluierung der Leistungsfähigkeit von LSTM-Modellen für die Approximation physikalischer Systeme

Vincent Katter^a, Bjarne Jaster^a und Marvin Schöne^a

^aHochschule Bielefeld, Center for Applied Data Science

Abstract. This study investigates the application of Long Short-Term Memory (LSTM) models to approximate complex physical processes. LSTM models, which are characterized by their ability to process large data sets and detect hidden patterns, are evaluated for their prediction capabilities. The research focuses on the advantages of LSTM models, such as accurate predictions, as well as challenges such as data dependency and the risk of overfitting. Experiments on a heating system show that LSTM models with optimized inputs and hyperparameters can be effectively used for predictive maintenance and improving operational efficiency in industrial applications. Bayesian optimization is used for hyperparameter optimization, which enables an improvement in model accuracy and a reduction in evaluation time. The results of this optimization and the knowledge gained from it offer valuable insights for the use of LSTM models in technical and industrial areas.

1 Einleitung

Diese Arbeit untersucht die Anwendung von Long Short-Term Memory (LSTM)-Modellen zur Approximation komplexer physikalischer Prozesse. Im folgenden Kapitel wird zunächst die Motivation für den Einsatz dieser Modelle beschrieben, gefolgt von der Zielsetzung dieser Arbeit. Dabei werden die Herausforderungen und Möglichkeiten bei der Implementierung von LSTM-Modellen dargestellt.

1.1 Motivation

Angesichts der zunehmenden Komplexität physikalischer Systeme stehen Wissenschaft und Industrie vor der Herausforderung, diese Systeme zu analysieren und präzise zu modellieren, um zuverlässige Prognosen zu erstellen [14]. Die genaue Modellierung physikalischer Prozesse ist entscheidend, da sie Produktionsabläufe optimieren, Risiken managen, die Qualität verbessern und die Nachhaltigkeit fördern kann [7] [6]. Physikalische Prozesse sind in der Regel kausal, das bedeutet, gegenwärtige Zustände und zukünftige Entwicklungen hängen von vergangenen Zuständen ab. Diese Prozesse werden häufig durch Zeitreihen beschrieben, die Trends, saisonale Muster und nichtlineare Beziehungen im zeitlichen Verlauf darstellen. Ein dynamisches System beschreibt solche Prozesse, indem es die zeitliche Entwicklung der Zustände des Systems darstellt. Aufgrund zahlreicher interagierender Komponenten und nichtlinearer Wechselwirkungen sind dynamische Systeme oft sehr komplex. Diese Systeme können durch mathematische Modelle abgebildet werden, um die zugrunde liegenden Prozesse besser zu verstehen.

Traditionell basiert die Modellierung solcher Systeme auf physikalischen Gesetzen und Formeln, um nichtlineare Zusammenhänge mathematisch zu beschreiben [15]. Dafür müsste das Systemverhalten vollständig bekannt sein, was in der Praxis oft nicht möglich ist. Daher beruhen Modelle häufig auf Annahmen und Vereinfachungen, um auch bei unvollständigem Wissen nützlich zu sein und physikalische Prozesse weitgehend zu verstehen und zu modellieren [2]. Unvollständiges Wissen kann jedoch zu unzuverlässigen Parametern, ungenauen Vorhersagen und geringerer Modellrobustheit führen, was die Validierung und den praktischen Nutzen erschwert. Mit zunehmender Komplexität steigt auch die Herausforderung, die gesamte Dynamik zu erfassen, und traditionelle Methoden erfordern umfangreiche Berechnungen, die für Echtzeitreaktionen zu langsam sind.

In diesem Zusammenhang haben sich datenbasierte Modelle, insbesondere rekurrente neuronale Netzwerke (RNN) wie LSTM Architekturen, als geeignet erwiesen. Diese Modelle sind in der Lage, umfangreiche und komplexe Datensätze zu verarbeiten, um daraus tiefgehende Erkenntnisse über nichtlineare Dynamiken zu gewinnen. Sie können verborgene Muster und Korrelationen in den Daten erkennen und diese zur Approximation des zugrundeliegenden Prozesses nutzen, ohne eine vollständige Kenntnis der genauen Zusammenhänge innerhalb eines Systems zu haben [1]. Darüber hinaus sind datenbasierte Modelle in der Lage, schnelle Inferenz und Vorhersagen durchzuführen, da sie auf der Grundlage bereits gelernter Muster operieren und keine aufwendigen Berechnungen für jeden neuen Datenpunkt benötigen. Im Gegensatz zu klassischen Modellen, die zeitaufwendige Berechnungen wie die Navier-Stokes-Gleichungen zur Modellierung von Flüssigkeitsbewegungen oder die Gesetze der Thermodynamik zur Beschreibung von Wärmeübertragungen erfordern, bieten datenbasierte Modelle eine effizientere Alternative. Dies macht sie zu einer praktischen Alternative oder Ergänzung zu traditionellen Methoden [18].

Trotz ihrer Fähigkeit, verborgene Muster und Korrelationen in den Daten zu erkennen und zu modellieren, stehen Anwender vor erheblichen Herausforderungen, die es zu lösen gilt, um die datenbasierten Modelle effektiv zu nutzen. Eine der größten Herausforderungen bei der Nutzung von datenbasierten Modellen ist ihre starke Abhängigkeit von der Qualität der zugrundeliegenden Daten [31]. Da datenbasierte Modelle keine physikalischen Formeln oder Gesetze berücksichtigen, können sie unrealistische Prognosen liefern, wenn bestimmte Verhaltensweisen in den Daten nicht vorhanden sind.

Zusätzlich erschwert die komplexe interne Struktur dieser Modelle das Verständnis ihrer Funktionsweise und der internen

Datenabbildung, was die Interpretation der Ergebnisse und das Vertrauen in die Vorhersagen beeinträchtigt [28]. Dies stellt insbesondere in sicherheitskritischen Anwendungen ein erhebliches Hindernis dar, da dort Nachvollziehbarkeit und Transparenz entscheidend sind.

Die Auslegung des Modells stellt ebenfalls eine bedeutende Herausforderung dar. Die Auswahl der Eingangsgrößen und Modellparameter beeinflusst die Leistung stark und erfordert sorgfältige Analysen. Während klare Richtlinien oder universelle Standards häufig fehlen, existieren branchenspezifische Anwendungsbeispiele und empirische Heuristiken, die Orientierung bieten können. Obwohl fortschrittliche Techniken wie automatische Hyperparameter-Optimierungstools hilfreich sein können, sind traditionell umfassende experimentelle Tests und Validierungen erforderlich.

1.2 Zielsetzung

Das übergeordnete Ziel dieser Forschungsarbeit ist es, die Fähigkeit von LSTM-Modellen zur Approximation physikalischer Prozesse, basierend auf sequenziellen Daten, zu untersuchen und valide Vorhersagen zu ermöglichen. Dies umfasst die Identifikation und optimale Aufbereitung relevanter Eingangsgrößen sowie die zugehörige Feinabstimmung der Modellparameter, um die Leistungsfähigkeit der LSTM-Modelle zu steigern.

Ein weiterer Schwerpunkt liegt auf der Interpretation und Validierung der Modelle. Der Einsatz diagnostischer Werkzeuge soll den Einfluss verschiedener Eingangsgrößen auf die Modellvorhersagen bestimmen und zusätzliche Erkenntnisse liefern.

Die Untersuchungen werden exemplarisch an einem realen physikalischen Prozess, der Heizstrecke, durchgeführt. Praktische Experimente sollen zeigen, wie effektiv LSTM-Modelle physikalische Daten approximieren können. Die gewonnenen Erkenntnisse werden sich hauptsächlich auf das untersuchte physikalische System beziehen, mit der Möglichkeit, Ansätze und Methoden für ähnliche Anwendungen zu evaluieren.

2 Stand der Technik und theoretische Grundlagen

Dieses Kapitel beschreibt den aktuellen Stand der Technik im Bereich der Anwendung von LSTM-Netzwerken und erläutert die theoretischen Grundlagen. Ziel ist es, eine solide Basis für die anschließende praktische Untersuchung des LSTM-Modells zur Approximation eines physikalischen Prozesses zu schaffen.

Die detaillierte Erklärung der LSTM-Architektur bildet die Basis, um die verschiedenen Mechanismen nachvollziehen zu können, die das Potenzial zur Leistungssteigerung der Modelle haben. Diese Mechanismen umfassen die Anpassung der Eingangsdaten und die Optimierung der Modellparameter. Zusätzlich wird ein modelldiagnostisches Tool vorgestellt, das eine zusätzliche Validierung und Interpretation der Modelle ermöglicht.

2.1 Approximation physikalischer Systeme durch LSTM

LSTM-Netzwerke sind darauf ausgelegt, neuronalen Netzen die Fähigkeit zu verleihen, sequentielle Informationen über längere Zeitintervalle hinweg zu speichern und zu verarbeiten. Diese Eigenschaft ist nützlich für Anwendungen, die auf der Analyse von

Zeitreihen bzw. sequentiellen Daten beruhen. LSTM-Netzwerke finden ein breites Spektrum von Anwendungen in unterschiedlichen Branchen. Im Finanzsektor werden sie beispielsweise zur Vorhersage von Aktienkursen eingesetzt [32]. Im Gesundheitswesen unterstützen sie die Früherkennung von Krankheiten durch die Erkennung von Mustern in medizinischen Daten [8]. In der Industrie tragen sie zur präventiven Wartung bei, indem sie Betriebsdaten analysieren [34], und in der Meteorologie helfen sie, historische Daten zu nutzen, um zukünftige klimatische Bedingungen vorherzusagen [21].

Aktuelle Forschungsarbeiten haben das Potenzial von LSTMs in einer Vielzahl industrieller Einsatzgebiete aufgezeigt. So wurden sie beispielsweise zur Kontrolle von Schadstoffwerten in Kläranlagen [23], zur Vorhersage der Produktqualität bei variierender Chargenlänge in der Fertigung [27] und zur Bestimmung des Sauerstoffgehalts in Müllverbrennungsanlagen [29] erfolgreich eingesetzt.

In der industriellen Produktion können die Daten der komplexen Prozesse häufig mehrdimensional und verrauscht sein, was die direkte Kontrolle der Parameter und die Messung der Produktqualität erschwert. In einer Forschungsarbeit wurde ein LSTM-Modell eingesetzt, um die Leistung und Energieeffizienz komplexer petrochemischer Systeme vorherzusagen. In diesem Beispiel wurde das Modell als Soft-Sensor verwendet [12].

Die Anwendung von Soft-Sensoren, auch als virtuelle Sensoren bekannt, bietet eine vielversprechende Perspektive. Soft-Sensoren nutzen vorhandene Messdaten und Prozessinformationen, um eine Zielgröße zu approximieren, die nicht direkt gemessen werden kann. Dies kann beispielsweise dazu dienen, einen Sensorausfall zu kompensieren oder schwer messbare Größen kontinuierlich zu überwachen [11]. Durch den Einsatz maschinellen Lernens zur Mustererkennung bieten Soft-Sensoren nicht nur eine kosteneffiziente und weniger aufwendige Alternative zu traditionellen Messmethoden, sondern sie ermöglichen auch tiefere Einblicke in industrielle Prozesse. [16]

Trotz der vielen Vorteile von LSTM-Modellen gibt es auch Einschränkungen bei deren Einsatz in Soft-Sensor-Anwendungen. Insbesondere haben LSTMs Schwierigkeiten, die Dynamik genau zu erfassen, wenn sich die Ausgangsgröße aufgrund einer Totzeit nicht unmittelbar nach einer Änderung der Eingangsparameter anpasst [22]. Diese Einschränkungen motivieren zu einer detaillierteren Untersuchung der Modellarchitektur und unterschiedlichen Methoden im Umgang mit sequenziellen Daten in LSTMs.

2.2 LSTM Netzwerke

Um die Alleinstellungsmerkmale von LSTM-Netzwerken vollständig zu verstehen, ist es wichtig, zunächst die Funktionsweise RNNs zu betrachten, da sie die Grundlage für die Approximation sequentieller Daten darstellen.

RNNs sind darauf ausgelegt, sequentielle Daten zu verarbeiten, indem sie Informationen aus vorherigen Zeitschritten in nachfolgende einfließen lassen. Diese Fähigkeit ist entscheidend für Anwendungen wie Spracherkennung und Textverarbeitung, bei denen zeitliche Abhängigkeiten wichtig sind [13].

Der grundlegende Aufbau eines RNN (siehe Abbildung 1) umfasst den Eingabevektor \mathbf{x}_t , den Zustandsvektor \mathbf{h}_t , das *Hidden-Layer* und den Ausgabevektor \mathbf{o}_t . Der Eingabevektor repräsentiert die Daten zum Zeitpunkt t , während der Zustandsvektor Informationen über den aktuellen Zustand speichert und durch den Eingabevektor und den vorherigen Zustandsvektor \mathbf{h}_{t-1} aktualisiert wird. Das Hidden-Layer besteht aus Neuronen, die durch Gewichte \mathbf{W} , \mathbf{W} und \mathbf{V} verbunden

sind. Diese Gewichte bestimmen die Verbindungen und beeinflussen die Informationsverarbeitung im Netzwerk. Der Ausgabevektor o_t ist das Resultat der aktuellen Verarbeitung. Die Gewichte werden während des Trainings optimiert, um die Netzwerkleistung zu verbessern. Optimierungsalgorithmen wie der Gradientenabstieg passen diese Gewichte an, um die Vorhersagegenauigkeit zu verbessern.

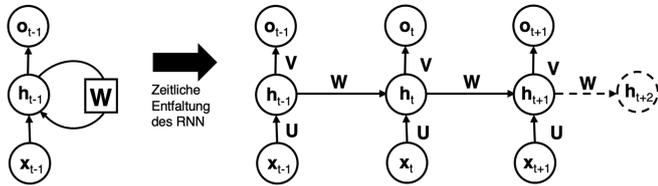


Abbildung 1: Zeitliche Entfaltung eines rekurrenten neuronalen Netzwerks

Das Training von RNNs und LSTMs erfolgt durch einen Prozess namens *Backpropagation Through Time* (BPTT). *Backpropagation* ist eine Methode zur Berechnung des Gradienten eines neuronalen Netzwerks, die darauf abzielt, die Fehlerfunktion zu minimieren. Diese Methode funktioniert, indem sie den Fehler von der Ausgangsschicht des Netzwerks zurück zur Eingangsschicht propagiert, und dabei die Gewichte in jeder Schicht so anpasst, dass der Fehler reduziert wird.

RNNs sind jedoch häufig nicht in der Lage, langfristige zeitliche Zusammenhänge zu modellieren. Diese Einschränkung ist auf das Phänomen des verschwindenden Gradienten zurückzuführen [19]. In einem RNN entsteht das Problem des verschwindenden Gradienten durch die Berechnung des Gradienten über das Produkt vieler partieller Ableitungen, basierend auf der Kettenregel. Da RNNs darauf ausgelegt sind, Informationen über viele Zeitschritte hinweg zu verarbeiten, bewirkt die Multiplikation der verketteten partiellen Ableitungen, dass der Gradient bei langen Sequenzen gegen Null konvergiert. Dies verhindert, dass die Gewichte in frühen Zeitschritten effektiv aktualisiert werden, wodurch das Netzwerk langfristige Muster und Abhängigkeiten nicht erkennen kann. Dieses Problem wird von LSTMs adressiert, die in der Lage sind, sowohl kurz- als auch langfristige Abhängigkeiten in den Daten zu modellieren [11]. LSTMs erweitern RNNs, indem sie nicht nur bestimmen, wie stark historische Werte in die aktuellen Vorhersagen einfließen sollen, sondern auch, ob Informationen langfristig oder kurzfristig bewahrt werden sollen.

2.2.1 Architektur

Der wesentliche Unterschied von LSTM-Netzwerken zu den einfachen RNNs liegt in der Erweiterung des Informationsflusses um die Langzeitgedächtnis-Komponente.

Die Komponente, die den RNNs das Erkennen von zeitlichen Zusammenhängen ermöglicht, ist der *Hidden State*, in dem Informationen aus vergangenen Zeitschritten gespeichert werden. Der *Hidden State* fungiert dabei als Kurzzeitgedächtnis. Der kontinuierliche Prozess der Aktualisierung und Übertragung des *Hidden State* erstreckt sich über das gesamte Netzwerk, wodurch ein stetiger Informationsfluss von einem Zeitschritt zum nächsten gewährleistet wird.

LSTM-Netzwerke erweitern diese Komponente um das Langzeitgedächtnis, den *Cell State*. Dadurch können sowohl kurz- als auch langfristige Informationen aus vergangenen

Zeitschritten verwendet werden. Der *Cell State* repräsentiert das Langzeitgedächtnis, während der *Hidden State* weiterhin als Kurzzeitgedächtnis dient. Der Aufbau der LSTM-Zelle unterscheidet sich grundlegend vom RNN, da die LSTM-Zelle zusätzliche Mechanismen, sogenannte *Gates*, enthält, die den Informationsfluss gezielt steuern und regulieren.

Diese Aufteilung in Lang- und Kurzzeitgedächtnis wird mit Hilfe von fünf sogenannten *Gates* umgesetzt (siehe Abbildung 2).

Insgesamt gibt es in einer LSTM-Zelle fünf sogenannte *Gates*, die unterschiedliche Aufgaben haben, um den Informationsfluss zu steuern, wie in Abbildung 2 dargestellt.

Das *Forget Gate* spielt eine zentrale Rolle bei der Aktualisierung des *Cell State* C . Es nutzt sowohl den vorherigen *Hidden State* H_{t-1} als auch die aktuelle Eingabe x_t , um zu entscheiden, welche Informationen beibehalten oder verworfen werden sollen. Die Sigmoid-Aktivierungsfunktion, die auf die Kombination aus dem vorherigen *Hidden State* H_{t-1} und dem aktuellen Eingabewert x_t angewendet wird, skaliert den Output auf das Intervall $[0, 1]$. Der resultierende Wert liegt zwischen 0 und 1 und bestimmt für jede Komponente des *Cell States*, wie stark die vorherige Information berücksichtigt werden soll. Ein Wert nahe 0 bedeutet, dass die Information vergessen, also nicht berücksichtigt wird, während ein Wert nahe 1 die Beibehaltung der Information signalisiert. Dadurch kann das *Forget Gate* präzise steuern, welche Informationen nicht mehr relevant sind und daher vergessen werden können.

Die zweite Modifikation der LSTM-Zelle wird durch das *Candidate Gate* und das *Input Gate* gesteuert. Das *Candidate Gate* berechnet neue Informationen, die für den *Cell State* relevant sein könnten. Dazu wird die *Tangens hyperbolicus* (*tanh*) Aktivierungsfunktion auf eine Kombination aus dem vorherigen *Hidden State* und dem aktuellen Eingabewert angewendet, um eine *Kandidateninformation* zu erzeugen. Die *tanh*-Funktion sorgt dafür, dass die Werte im *Candidate Gate* immer im Bereich von -1 bis 1 liegen. Diese Skalierung der Werte hilft, extrem große Werte zu vermeiden, was das Training des Netzwerks stabiler und effizienter macht, da es die Gradienten im Trainingsprozess reguliert und numerische Instabilitäten minimiert. Das *Input Gate* bestimmt, wie stark die potenziell neuen Informationen aus dem *Candidate Gate* in den *Cell State* einfließen sollen. Ähnlich wie das *Forget Gate* nutzt das *Input Gate* eine Sigmoid-Aktivierungsfunktion, die auf eine Kombination aus dem vorherigen *Hidden State* und dem aktuellen Eingabewert angewendet wird. Das Ergebnis bestimmt die Stärke der Berücksichtigung. Die Gewichtung der neuen Informationen durch den Output des *Input Gate* ermöglicht es dem LSTM, selektiv zu entscheiden, welche neuen Informationen in den *Cell State* einbezogen werden sollen, und trägt dazu bei, die Relevanz und Bedeutung der aktuellen Eingabe für die Aktualisierung des *Cell State* zu bestimmen. Das Produkt der Multiplikation beider *Gates* erzeugt einen Wert, der zum Langzeitgedächtnis (*Cell State*) hinzugefügt wird.

Im *Final Gate* wird der Wert des *Cell State* durch die Anwendung der *tanh*-Aktivierungsfunktion aktiviert, um eine Größe zu erzeugen, die das potenzielle Kurzzeitgedächtnis repräsentiert. Das Produkt der Ergebnisse aus dem *Output Gate* und dem *Final Gate* bildet den neuen *Hidden State* H_t für den nächsten Zeitpunkt t in der Sequenz.

Die Kombination dieser *Gates* ermöglicht es LSTM-Netzwerken, langfristige Abhängigkeiten in sequenziellen Daten zu modellieren, indem sie es ihnen erlauben, relevante Informationen zu behalten und unwichtige Informationen zu vergessen.

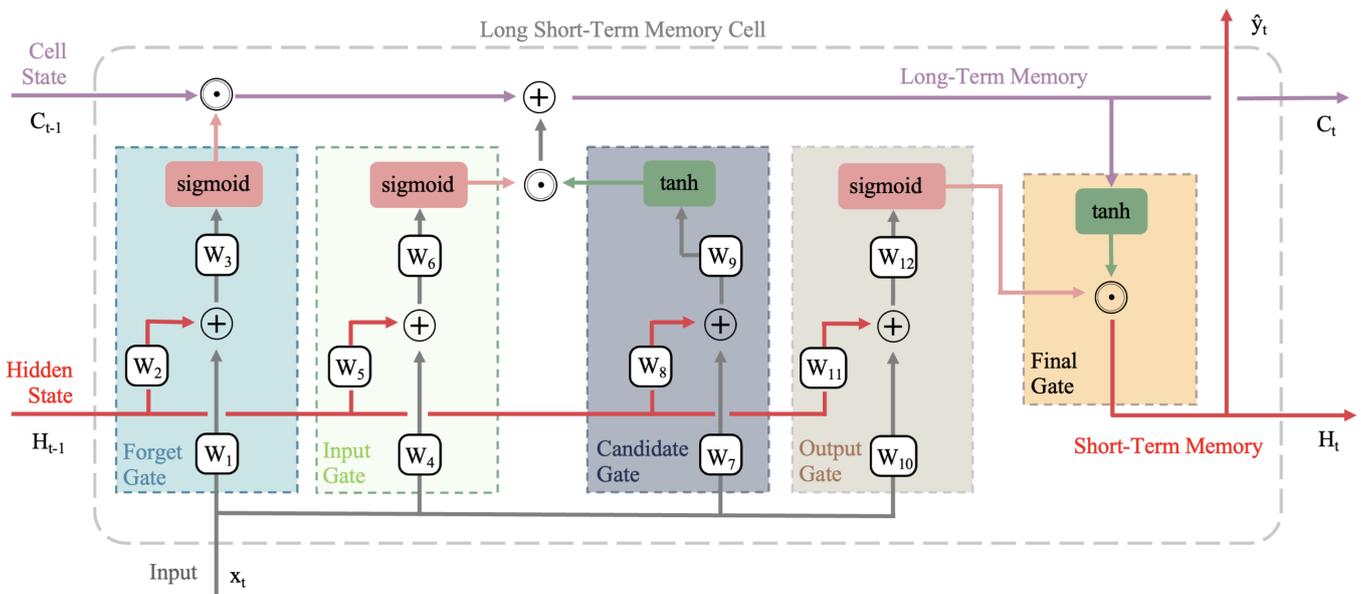


Abbildung 2: Aufbau einer Long Short-Term Memory Zelle

Beim Vergleich der Informationsflüsse in Abbildung 2 wird deutlich, dass im Unterschied zum *Hidden State* der *Cell State* keine direkt verbundenen Gewichte aufweist. Dadurch bleibt der *Cell State* unbeeinflusst von den direkten Anpassungen der Gewichte während des Trainings. Diese Trennung trägt dazu bei, Probleme mit verschwindenden Gradienten zu vermeiden und ermöglicht so die Aufrechterhaltung von Informationen über lange Sequenzen hinweg. Der *Hidden State* wiederum nutzt Gewichte, um Informationen effektiv zu modellieren und Kurzzeitabhängigkeiten in den Daten zu erfassen. Im Gegensatz zu herkömmlichen RNNs, bei denen die Backpropagation durch die gesamte Sequenz erfolgt, beschränkt sich die Backpropagation bei LSTMs nur auf einen Zeitschritt zurück. Dies wird erreicht, indem die Korrektur des Vorhersagefehlers direkt von der Veränderung des *Cell States* abhängt, was eine effektive Lernfähigkeit auch über lange Sequenzen hinweg gewährleistet. Diese Struktur erleichtert eine effiziente Rückpropagierung des Fehlers durch das Netzwerk und trägt somit dazu bei, dass das LSTM effektiv trainiert werden kann, ohne Probleme mit verschwindenden oder explodierenden Gradienten zu verursachen.

Die Fähigkeit von LSTMs, relevante Informationen über lange Zeitintervalle zu speichern und effizient zu verarbeiten, macht sie zu einem leistungsstarken Werkzeug für die Modellierung von Sequenzen, insbesondere wenn es um die Erfassung langfristiger Muster geht [33].

2.3 Datenstrukturierung für LSTM-Netze

Die Datenvorverarbeitung ist ein wesentlicher Schritt, um das Training von LSTM-Netzwerken zu beschleunigen und sicherzustellen, dass das Modell die sequentielle Abhängigkeit effektiv lernt. Ein wichtiger Ansatz in der Datenvorverarbeitung für LSTM-Modelle ist der *Lookback*-Ansatz.

Der *Lookback*-Ansatz wird eingesetzt, um sicherzustellen, dass das Modell relevante historische Datenpunkte berücksichtigt, was besonders für zeitabhängige Vorhersagen von großer Bedeutung ist. Durch diesen Ansatz kann das Modell besser die sequentiellen

Muster in den Daten erfassen und lernen.

Der *Lookback*-Ansatz strukturiert die Daten so um, dass für

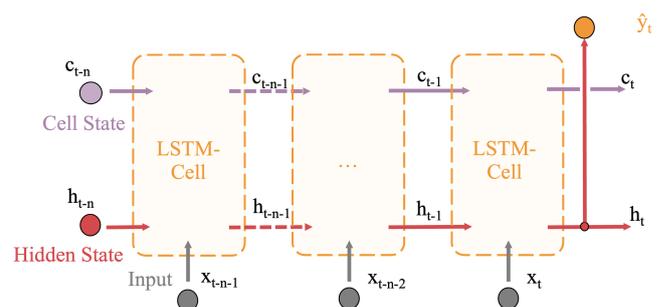


Abbildung 3: Funktionsweise des *Lookback*-Ansatzes

jeden Zeitpunkt t eine definierte Länge historischer Datenpunkte zugeordnet wird. Dies ermöglicht die gleichzeitige Verarbeitung mehrerer Sequenzen, insbesondere auf Hardware-Beschleunigern wie GPUs, ohne dass Kontextinformationen verloren gehen. Die Beibehaltung der sequentiellen Ordnung des Datensatzes durch den *Lookback* ermöglicht eine konsistente Verarbeitung relevanter Datenabschnitte.

Wie in Abbildung 3 dargestellt, sorgt der *Lookback*-Ansatz dafür, dass der Eingabevektor (x_t) zusammen mit den *Cell*- und *Hidden*-States durch die definierten historischen Werte angeglichen wird, bevor eine Vorhersage (\hat{y}_t) getroffen wird. Dies bedeutet, dass während des Trainings beim ersten historischen Datenpunkt (x_{t-n}) gestartet wird und die neuen Werte für den *Cell*- und *Hidden*-State berechnet werden. Dieser Prozess wird iterativ fortgeführt, bis der Zeitpunkt t erreicht ist und eine Vorhersage mit den angeglichenen *Hidden*- und *Cell*-States durchgeführt wird.

Durch diese Struktur kann das LSTM seine internen Zustände, basierend auf den letzten Zeitschritten, anpassen. Der *Lookback*-Ansatz hat somit keinen Einfluss auf die Größe des Eingaberaums, sondern definiert lediglich, wie viele historische Iterationen notwendig sind, bevor eine Vorhersage getroffen werden kann

Ein weiterer wichtiger Punkt ist, dass im Testbetrieb, wenn sequentiell neue Vorhersagen getroffen werden sollen, das LSTM nur den Eingabevektor ohne die historischen Werte berücksichtigt. Da die internen Zustände bereits während des Trainings angeglichen wurden, kann das Modell sofort mit neuen Eingabewerten arbeiten und Vorhersagen treffen. Dies gewährleistet eine kontinuierliche und konsistente Verarbeitung neuer Daten, ohne dass die gesamte Sequenz erneut durchlaufen werden muss.

Der *Lookback*-Ansatz bietet also nicht nur eine Möglichkeit zur Effizienzsteigerung und besseren Parallelisierung während des Trainings, sondern auch eine Methode, um sicherzustellen, dass das LSTM-Modell seine Aufmerksamkeit auf die relevantesten jüngsten Daten fokussiert. Dies führt zu präziseren und verlässlicheren Vorhersagen, da das Modell die entscheidenden zeitlichen Abhängigkeiten besser erlernen kann.

Ein Beispiel für die erfolgreiche Anwendung ist die Vorhersage von Aktienkursen an der nepalesischen Börse: Bei der angepassten *Lookback*-Zeiträume zu einer verbesserten Vorhersagegenauigkeit führte [25]. Zudem zeigte die Analyse von Tesla-Aktienkursen über 986 Tage, dass die Wahl eines optimalen *Lookback*-Zeitraums entscheidend ist, da zu kleine Fenster wichtige Signale übersehen und zu große Fenster die Informationen durch Rauschen verfälschen können [24].

2.3.1 Selbstregulierung durch Feedback-Schleifen

Der *Lookback*-Mechanismus ermöglicht es dem Modell, sich auf die letzten relevanten Zeitpunkte zu fokussieren und die sequentiellen Abhängigkeiten besser zu erfassen. Ein weiterer Ansatz zur Verbesserung der Vorhersagegenauigkeit in LSTM-Netzwerken ist die Einführung von Feedback-Schleifen. Dabei werden vorherige Modellausgaben als zusätzliche Eingaben für zukünftige Prognosen verwendet. Diese Rückführung der Modellausgabe beschreibt den Prozess, bei dem die Ausgabe des Modells für einen bestimmten Zeitschritt (t) als Teil der Eingabe für den nächsten Zeitschritt ($t + 1$) genutzt wird.

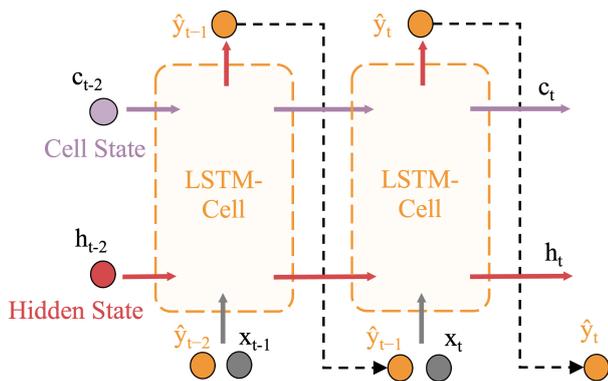


Abbildung 4: Schematische Darstellung der Rückführung der Modellausgangsgröße

Wie in Abbildung 4 dargestellt, zeigt die Feedback-Schleife, dass die vorherige Vorhersage \hat{y}_{t-1} als Teil der Eingabe für die nächste Vorhersage \hat{y}_t verwendet wird. Dies wird im Diagramm durch die schwarzen, gestrichelten Linien illustriert, die die Ausgabe eines Zeitschritts zur Eingabe des nächsten führen. Dadurch kann das LSTM-Modell kontinuierlich die internen Zustände (*Cell State* C_t und *Hidden State* H_t) basierend auf früheren Vorhersagen anpassen,

wie durch die lila und roten Pfeile angezeigt. Auf diese Weise kann das Modell aus seinen eigenen Vorhersagen lernen und diese Informationen in weiteren Schätzungen berücksichtigen.

Darüber hinaus besteht die Möglichkeit, den *Lookback*-Mechanismus mit der Selbstregulierung durch Feedback-Schleifen zu integrieren. Dabei werden nicht nur die historischen Eingangsgrößen, sondern auch die vorherigen Modellausgaben für zukünftige Schätzungen berücksichtigt. Diese Kombination ermöglicht es dem Modell, sowohl auf historische Daten als auch auf frühere Vorhersagen zurückzugreifen, um die Prognosegenauigkeit zu erhöhen.

Ein konkretes Beispiel für die Anwendung von LSTM-Modellen mit Selbstregulierung durch Feedback-Schleifen ist ihr Einsatz in der Industrie, wo die Integration dieser Methode zu einer signifikanten Steigerung der Vorhersagegenauigkeit geführt hat, indem die Modellausgaben erneut ins Modell eingespeist wurden, um die Informationsdichte zu erhöhen [10]. Ein weiteres Beispiel zeigt die Entwicklung eines LSTM-Modells, bei dem die Rückführung historischer Modellausgaben aus dem Trainingsdatensatz ebenfalls zur Steigerung der Genauigkeit beiträgt [30].

2.4 Modelldiagnostik mit TimeSHAP

LSTM-Modelle gelten häufig als Blackbox-Modelle, was bedeutet, dass es schwierig sein kann, den Einfluss einzelner Eingabeparameter und Zeitschritte auf die Vorhersageleistung nachzuvollziehen. In vielen Fällen ist es jedoch wichtig, diese Einflüsse zu untersuchen, um das Modell besser zu verstehen und zu optimieren. Zu diesem Zweck wird das Diagnostik-Tool *TimeSHAP* verwendet, welches entwickelt wurde, um den Beitrag jedes Eingangsparameters zur Vorhersage eines Modells zu quantifizieren und visuell darzustellen [5].

TimeSHAP bietet eine systematische Methode, welche die Auswirkungen der Eingangsvariablen auf das Ergebnis eines LSTM-Modells berechnet und visualisiert. Dadurch wird ersichtlich, welche Variablen den größten Einfluss auf die Vorhersagen des Modells haben und wie sich dieser Einfluss über die Zeit verändert. Diese Informationen können wertvoll sein, da sie das Verständnis der Modellfunktionen verbessern und dazu beitragen können, wichtige Entscheidungen über Modellanpassungen und die Auswahl von Merkmalen für weitere Analysen zu treffen.

Ein wesentlicher Aspekt von *TimeSHAP* ist die Berechnung von Shapley-Werten für jede Eingangsvariable, einem zentralen Maß aus der kooperativen Spieltheorie, das den Beitrag ϕ_i jeder Eingangsvariablen i zu einem kooperativen Spiel, hier dem Vorhersagemodell, quantifiziert. Diese werden wie folgt berechnet:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S))$$

Hierbei steht ϕ_i für den Shapley-Wert der Eingangsvariablen i , N für die Menge aller Eingangsvariablen, S für eine Teilmenge von N ohne i , n für die Gesamtzahl der Merkmale und $v(S)$ für den Vorhersagewert des Modells für die Eingabemenge S . Diese Formel berechnet den durchschnittlichen marginalen Beitrag von i zu den Vorhersagen des Modells, indem sie über alle möglichen Kombinationen von Eingangsvariablen summiert, die i nicht enthalten [26].

TimeSHAP führt diese Berechnungen durch gezielte Veränderungen der Eingangsvariablen durch und beobachtet die daraus resultierenden Auswirkungen auf die Modellausgabe.

Das *TimeSHAP*-Modul für die Modelldiagnostik bietet drei verschiedene Analyseebenen, um den Einfluss von Merkmalen und Zeitereignissen auf die Vorhersagen zeitabhängiger Modelle zu verstehen: die Ereignisebene, die Merkmalsebene und die Zellebene. Die Ereignisebene betrachtet den Einfluss einzelner Zeitereignisse auf die Modellvorhersage und hilft dabei, kritische Zeitpunkte oder Muster in der Zeitreihe zu identifizieren. Auf der Merkmalsebene wird untersucht, wie verschiedene Merkmale eines Eingabewertes x innerhalb eines bestimmten Zeitpunkts zur Modellvorhersage beitragen, wodurch die Bedeutung einzelner Variablen verdeutlicht wird. Schließlich analysiert die Zellebene den Einfluss einzelner Zeitschritte in einem LSTM-Netzwerk auf die Vorhersage, indem sie zeigt, wie gespeicherte Informationen aus der Vergangenheit die Modellentscheidungen beeinflussen. Diese drei Analyseebenen bieten umfassende Einblicke in die Entscheidungsprozesse von zeitabhängigen Modellen und verbessern deren Erklärbarkeit.

TimeSHAP gewährt Einblicke in die Entscheidungsprozesse von LSTM-Netzwerken, indem es die Beiträge einzelner Eingabefeatures über die Zeit hinweg bewertet. In einer Forschungsarbeit, die sich der Verbesserung von Methoden zur Vorhersagegenauigkeit bei Zeitreihendaten widmet, wird *TimeSHAP* eingesetzt, um die Effektivität neuronaler Netze zu analysieren. Diese Erkenntnisse sind nicht nur für die Überprüfung der Modelle relevant, sondern auch für ihre gezielte Optimierung [3]. Darüber hinaus trägt *TimeSHAP* zur Steigerung der Transparenz und Nachvollziehbarkeit von maschinellen Lernmodellen bei. Es unterstützt die Vertrauensbildung in diese Technologien und kann dazu beitragen, ihre Akzeptanz und Anwendung in verschiedenen Bereichen zu erweitern [17].

3 Experimentelle Umgebung

Der physikalische Prozess, an dem die Approximation mit Hilfe eines LSTMs durchgeführt werden soll, ist eine Heizstrecke. Dabei sollen verschiedene Parameter an dem LSTM verändert werden, um seine Leistungsfähigkeit zu validieren.

Die Grundlage für den physikalischen Prozess eine Heizstrecke der Firma *ELWE Technik* verwendet. Neben einem Lüfter und einer Heizung, die beide variabel leistungsregulierbar sind, sind auch Sensoren integriert, welche die Lufttemperatur und den Luftmassenstrom erfassen. Zusätzlich befindet sich eine manuell verstellbare Drosselklappe am Lufteinlass, die zur Erzeugung von Störungen dient. Eine Darstellung dieses Labormodells findet sich in Abbildung 5. Das System lässt sich somit als ein Mehrgrößensystem charakterisieren, das über zwei Eingangsgrößen – die Lüfterleistung P_L und die Heizleistung P_H –, zwei Ausgangsgrößen – die Temperatur T und den Luftmassenstrom \dot{m}_L – und eine Störgröße durch den Öffnungswinkel der Drosselklappe α repräsentiert. Dieser physikalische Prozess kann als thermodynamischer Prozess betrachtet werden, bei dem Luft durch eine Heizstrecke strömt und dabei ihre Temperatur und ihren Massenstrom ändert. Daher ist es eine Art von Wärmekraftprozess oder Wärmeübertragungsprozess, der in vielen technischen Anwendungen, wie z.B. in der Heizungs- und Klimatechnik, relevant ist.

In dieser Untersuchung werden nur die Eingangsgrößen Temperatur- und Lüfterleistung verwendet, während die Störgröße konstant gehalten wird. Die durchgeführten Experimente auf der Heizstrecke umfassten die kontinuierliche Erfassung von Daten bezüglich Temperatur, Heiz- und Lüfterleistung. Insbesondere werden die Heiz- und Lüfterleistung systematisch variiert, um verschiedene Betriebsbedingungen zu simulieren und ihre Auswirkungen auf

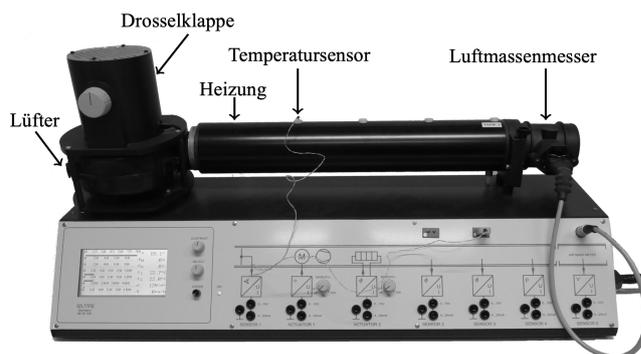


Abbildung 5: Labormodell der Heizstrecke

die Systemleistung zu analysieren. Als Anregungssignal wird ein amplitudenmoduliertes Pseudo-Rausch-Binär-Signal (APRBS) verwendet, das den Wertebereich $[0,3; 0,7]$ abdeckt.

Diese methodische Variation der Eingangsgrößen ermöglicht die Erstellung einer umfangreichen Zeitreihe, die eine grundlegende Basis für die Forschungsarbeit bildet. Mit über 200.000 Datenpunkten bietet diese Zeitreihe eine solide Grundlage für die Analyse der Systemreaktionen auf Veränderungen der Betriebsparameter. Ein Ausschnitt dieser Zeitreihe ist in Abbildung 6 zu sehen.

4 Modelluntersuchungen

Das nachfolgende Kapitel gliedert sich in drei Unterkapitel, die jeweils einen Einblick in die Approximation des physikalischen Prozesses bieten und die damit verbundenen Methoden validieren, um den produktiven Einsatz eines LSTM-Modells zur Approximation eines physikalischen Prozesses zu ermöglichen.

Im ersten Unterkapitel werden die verschiedenen Eingangsgrößen untersucht, einschließlich der Berücksichtigung der Selbstregulierung durch Feedback-Schleifen (SFL), die das Modell für präzise Vorhersagen benötigt. Das zweite Unterkapitel widmet sich der Länge des *Lookbacks*, um die sequentiellen Abhängigkeiten korrekt zu erfassen. Schließlich werden im dritten Unterkapitel weitere Hyperparameter des LSTM-Modells analysiert und optimiert, um die Leistungsfähigkeit des Modells zu maximieren und dessen Eignung zu validieren.

4.1 Betrachtung der Eingangsgrößen für LSTM-Modelle: Evaluierung der SFL-Methode

In diesem Abschnitt wird der Einfluss der Eingangsgrößen auf die Genauigkeit von LSTM-Modellen zur Approximation physikalischer Prozesse untersucht. Genauer gesagt soll überprüft werden, ob die Berücksichtigung der Ausgangsgröße im Eingaberaum zu einer verbesserten Vorhersagegenauigkeit führt.

Der Einfluss der Rückführung auf das LSTM-Modell wird mithilfe von *TimeSHAP* visualisiert. Diese Analyse soll Aufschluss über die Leistungsfähigkeit der entwickelten Vorhersagemodelle geben und die Bedeutung der gewählten Eingangsgrößen verdeutlichen.

4.1.1 Modellkonfiguration

Die Konfiguration des LSTM-Modells für diese Untersuchung umfasst die Verwendung eines LSTM-Netzwerks mit 200 Neuronen, das mit dem Adam-Optimizer trainiert wurde. Der Adam-Optimizer ist

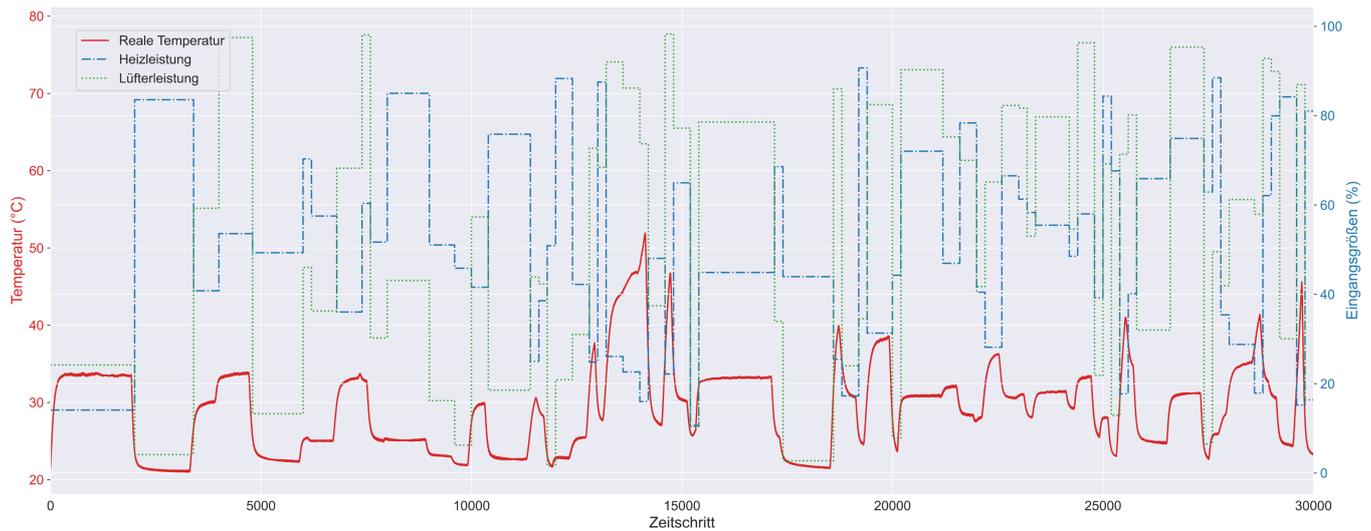


Abbildung 6: Darstellung der Eingangsgrößen und Ausgangsgröße der Heizstrecke

eine adaptive Lernmethode, welche die Lernrate für jedes Gewicht im Netzwerk individuell anpasst, basierend auf den vergangenen Gradienten [9]. Zusätzlich wurde ein *Lookback*-Wert von 100 festgelegt. Das Training erstreckt sich über 50 Epochen. Diese Parameter wurden in Anlehnung an eine andere Forschungsarbeit ausgewählt, welche denselben Datensatz untersucht [4]. Diese Konfiguration dient für diese Arbeit als Ausgangskonfiguration, um die Leistungsfähigkeit des Modells weiter zu optimieren.

Für das Training und die Evaluierung der Modelle werden die ersten 80 000 Datenpunkte des Datensatzes ausgewählt. Diese Auswahl basiert darauf, dass die Zeitreihe mit einem APRBS-Signal aufgenommen wurde. Dieses Signal erzeugt eine breite Varianz und bietet eine repräsentative Abdeckung der Systemdynamik. Aufgrund dieser Eigenschaften enthalten die ausgewählten Datenpunkte genügend Varianz, um eine verlässliche Beurteilung der Modelleleistung zu ermöglichen. Diese Vorgehensweise ermöglicht ein effizientes Training und spart gleichzeitig Ressourcen, ohne die Aussagekraft der Vorhersagen zu beeinträchtigen. Der Datensatz wird im Verhältnis 80 % zu 20 % aufgeteilt, um Trainings- und Testdaten zu generieren.

Zur Bewertung der Vorhersagegenauigkeit des Modells wird der Root Mean Square Error (RMSE) verwendet. Der RMSE misst die durchschnittliche Abweichung zwischen den vorhergesagten und den tatsächlichen Werten und ist eine gängige Metrik zum Vergleich verschiedener Modelle. In diesem Fall wird der RMSE auf dem Testdatensatz berechnet, um die Generalisierungsfähigkeit des Modells zu bewerten.

Um die Effektivität der SFL-Methode zu gewährleisten, wird ein weiteres LSTM mit den realen Eingangsgrößen (Lüfter- und Heizleistung) trainiert, um die Ausgangsgröße des physikalischen Modells (Temperatur) zu approximieren. Dieses Modell dient im Weiteren als Basis- oder Referenzmodell, um sämtliche Anpassungen zu validieren und zu überprüfen, ob diese zu einer Verbesserung der Vorhersagegenauigkeit führen. Der Verlauf der Vorhersage im Vergleich zum Testdatensatz sowie der RMSE ist in Abbildung 7 dargestellt.

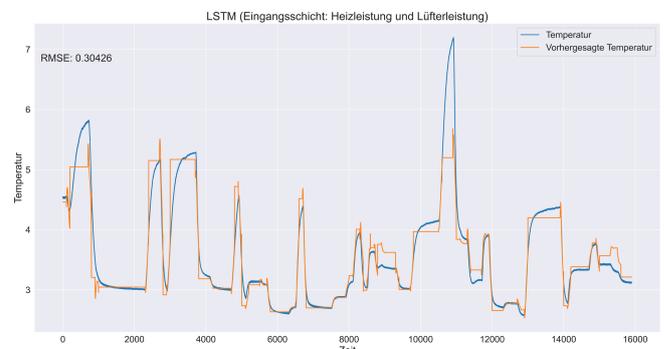


Abbildung 7: Genauigkeit der LSTM-Approximation bei realen Eingangsgrößen

4.1.2 Validierung der SFL

Wie im Kapitel 2.3.1 erläutert, zielt die SFL-Methode darauf ab, die Vorhersagegenauigkeit zu verbessern, indem die Ausgangsgröße, in den Eingaberaum integriert wird. Um diese Methode zu validieren, wird das Modell um ein zusätzliches Neuron in der Eingabeschicht erweitert und der Datensatz so angepasst, dass neben den beiden Eingangsgrößen (Lüfter- und Heizleistung) auch die Ausgangsgröße (Temperatur) berücksichtigt wird. Die restlichen Konfigurationen des Modells bleiben unverändert.

Die Genauigkeit der Vorhersage unter Berücksichtigung der Ausgangsgröße im Eingaberaum ist in Abbildung 8 dargestellt. Diese Abbildung zeigt die Modellvorhersage im Vergleich zu den realen Testdaten und dem berechneten RMSE.

Die nahezu überlappenden Kurven und der geringe RMSE-Wert (siehe Abbildung 8) zeigen, dass das LSTM mit Hilfe der gemessenen Temperatur des vorherigen Zeitschritts nahezu perfekt in der Lage ist, die Temperatur des nächsten Zeitschritts vorherzusagen. Geht man aber davon aus, dass zur Testzeit nicht die gemessene Temperatur zur Verfügung steht, wie es bei Eingangs erwähnten Soft-Sensoren (siehe Kapitel 2.1) der Fall ist, muss man die gemessene Temperatur durch die Vorhersage des Modells vom letzten Zeitschritt ersetzen. Zwar können historische Werte der Ausgangsgröße für das Training des Modells verwendet werden, jedoch stehen in einer produktiven Soft-Sensorik keine historischen Werte zur Verfügung. In der

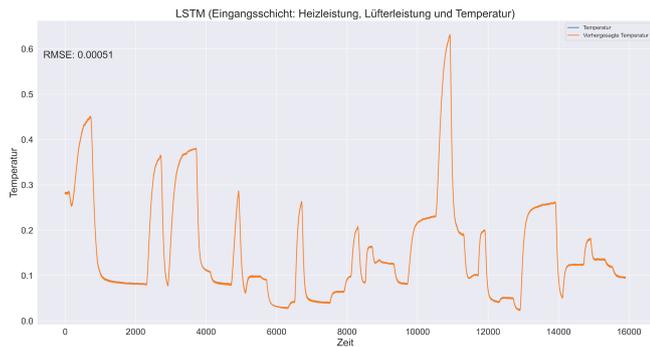


Abbildung 8: Genauigkeit der LSTM-Approximation bei der Berücksichtigung der Ausgangsgröße im Eingaberaum

Forschung gibt es Beispiele, in denen Modelle lediglich mit dem Testdatensatz validiert wurden, ohne die Modellausgangsgröße zurückzuführen [10] [30]. Dieser Ansatz kann zu einer künstlichen Erhöhung der Modellperformance führen und zeigt die Diskrepanz zwischen der Testumgebung und den realen Einsatzszenarien. Daher ist es notwendig zu validieren, ob eine Rückführung der Modellvorhersage zu ähnlichen Vorhersagen führen kann, um die tatsächliche Vorhersagegenauigkeit in Soft-Sensorik-Anwendungen zu bewerten. Um den Mechanismus des *Lookbacks* aufrechtzuerhalten, werden für die ersten Zeitschritte historische Werte verwendet. Schritt für Schritt werden diese durch die Modellvorhersagen ersetzt. Diese initialen historischen Werte sind notwendig, damit das LSTM-Modell die internen States der Zelle angleichen kann.

Die Rückführung der Ausgangsgröße des Modells wird nachfolgend

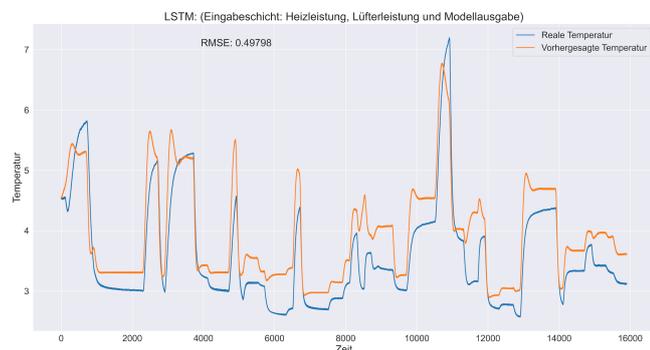


Abbildung 9: Vorhersagegenauigkeit bei der Rückführung der Ausgangsgröße

als Simulation bezeichnet, da sie die potenzielle Anwendung des Modells in einer produktiven Soft-Sensorik beschreibt. Die Simulation des Modells (siehe Abbildung 9) verdeutlicht, dass die Vorhersagegenauigkeit unter der Rückführung des Modellausgangs leidet. Die Ergebnisse sind deutlich schlechter als bei der Validierung mit dem historischen Testdatensatz (Abbildung 8), was darauf hindeutet, dass eine Validierung mit Testdaten in Soft-Sensorik-Anwendungen die Performance des Modells überschätzen kann. Dies bestätigt sich, wenn man den RMSE zum Referenzmodell in Abbildung 7 vergleicht.

Dies wirft die Frage auf, warum das Modell mit den historischen Daten nahezu perfekte Testergebnisse erzielt. Um diese Frage zu beantworten, wird im nächsten Abschnitt der Einfluss der Rückführung der Ausgangsgröße mit *TimeSHAP* analysiert.

4.1.3 Analyse der Modelleffizienz mit *TimeSHAP*

Zur Bewertung der Feature-Relevanz in einem RNN werden die vorgestellten LSTM-Modelle mit der *TimeSHAP*-Methode analysiert. Unter Verwendung einer Stichprobengröße von 5000 wurden die Shapley-Werte im Bereich von $[-0.05, 0.05]$ untersucht. Die Achsenbegrenzung ist speziell reduziert, um feine Ausprägungen in den zeitlichen Mustern der Shapley-Werte hervorzuheben. Der obere Plot von Abbildung 10a und 10b stellt den Einfluss der einzelnen Zeitschritte auf die Vorhersage dar, während der untere Plot den Einfluss der Eingangsgrößen, gemittelt über die Zeit, darstellt.

Als Ausgangspunkt wird das LSTM-Modell mit den beiden realen

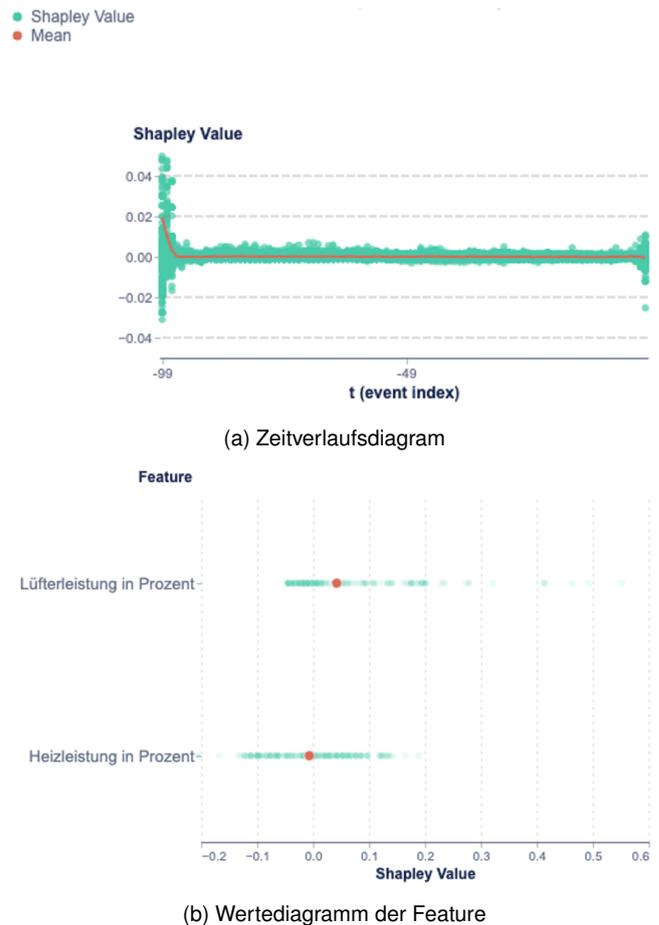
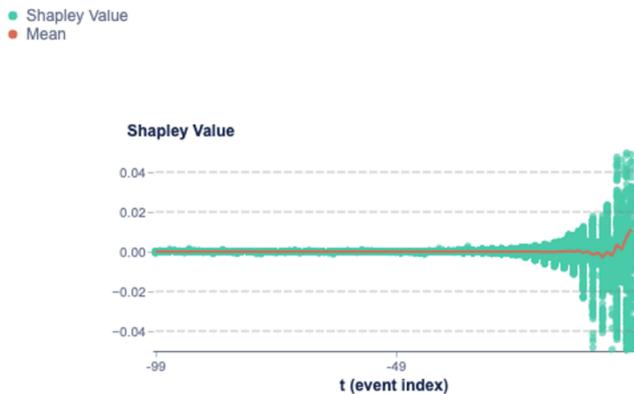


Abbildung 10: Modelldiagnostik des LSTM mit den realen physikalischen Eingangsgrößen

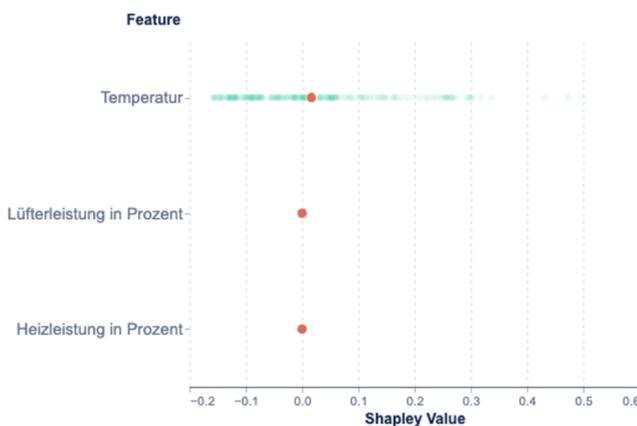
physikalischen Eingangsgrößen mit *TimeSHAP* analysiert, um eine Referenz zu haben. Das Ergebnis der Analyse in Abbildung 10 zeigt größtenteils eine gleichmäßige Verteilung der Shapley-Werte entlang der Sequenz, was auf eine konsistente Berücksichtigung der zeitlichen Daten durch das Modell hinweist. Auffallend sind die Ausschläge zu Beginn und besonders stark am Ende der Sequenz, die auf eine bevorzugte Gewichtung dieser Zeitpunkte hindeuten. Im summarischen Wertediagramm werden die Shapley-Werte für *Lüfterleistung in Prozent* und *Heizleistung in Prozent* überwiegend nahe Null dargestellt, was auf einen gleichmäßigen und ähnlichen Einfluss dieser Features hindeutet. Das summarische Wertediagramm zeigt eine hohe Varianz und reflektiert die differenzierte Gewichtung der Features in Abhängigkeit ihrer

zeitlichen Position. Die Überlappung der Streupunkte beider Features lässt auf keine signifikante Differenzierung ihrer Einflüsse schließen, wobei die Lüfterleistung einen leicht größeren Einfluss zeigt.

Im Gegensatz dazu stützt sich das LSTM mit den drei



(a) Zeitverlaufdiagramm



(b) Wertediagramm der Feature

Abbildung 11: Modelldiagnostik des LSTM mit der Rückführung der Ausgangsgröße

Eingangsgrößen auf das Attribut der Temperatur, welches zugleich die Ausgangsgröße darstellt. Im Zeitverlaufdiagramm 11a wird eine Variabilität der Shapley-Werte über den Zeitraum ersichtlich, wobei die Werte eine Präferenz des Modells für Datenpunkte der nahen Vergangenheit aufzeigen. Dieser Trend wird durch einen Anstieg der durchschnittlichen Shapley-Werte untermauert, welcher die zunehmende Bedeutung aktuellerer Zeitpunkte für die Vorhersagekraft des Modells belegt. Das summarische Wertediagramm stellt eine Dominanz der *Temperatur* als einflussreichstes Feature dar, während *Heizleistung in Prozent* und *Lüfterleistung in Prozent* nur geringe Einflussnahme aufzeigen. Diese Beobachtung unterstreicht eine ausgeprägte Konzentration des Modells auf Temperaturwerte und deutet auf eine mögliche Vernachlässigung anderer dynamischer Features im Datensatz hin. Die Gewichtung des Modells auf aktuelle Werte einer Größe lässt sich durch den Time-Lag-Effekt erklären. Dieser tritt bei univariaten und multivariaten LSTM-Modellen auf, wenn sie Eingangssequenzen mit einer zeitlichen Verzögerung reproduzieren

[32]. Dieses Problem wird durch eine übermäßige Konzentration der Modelle auf die Ausgangsgrößen verschärft, was tiefere kausale Zusammenhänge vernachlässigt. Ist der Vorhersagehorizont sehr gering, schwimmt die zeitliche Verzögerung und das Modell versucht lediglich, kurzfristig die Werte zu reproduzieren. Da das Modell eigenständig die Eingangsgrößen gewichtet, um die Ausgangsgröße zu berechnen, stützt es sich stark auf die Ausgangsgröße, die nahe an der tatsächlichen Vorhersage liegt. Dies führt zu sehr genauen Vorhersagen, wie in Abbildung 8 zu sehen, ohne dass ein tatsächlicher Zusammenhang in den Daten erlernt wurde. Wenn dieser Wert nicht mehr auf historischen Werten, sondern auf der Modellvorhersage basiert, lassen sich auch die schlechteren Vorhersagen in Abbildung 9 erklären. Das Modell driftet ab, da es sich nicht mehr an den realen Werten orientieren kann. Das Modell zeigt Schwachstellen bei der Berücksichtigung der realen physikalischen Eingangsgrößen, was seine Eignung für Anwendungen in der Soft-Sensorik infrage stellt.

Das LSTM mit den beiden realen physikalischen Eingangsgrößen wird darauf trainiert, die Zusammenhänge der Daten zu erlernen, um die Ausgangsgröße bestmöglich zu approximieren. Da es bessere Ergebnisse in der Simulation erzielt wird dieses LSTM für die weitere Anpassung der Parameter verwendet.

Die Ergebnisse betonen die Bedeutung einer angemessenen Berücksichtigung relevanter zeitlicher Features für die Verlässlichkeit von LSTM-basierten Vorhersagemodellen und unterstreichen die Bedeutung der *TimeSHAP*-Analyse für die Interpretierbarkeit von rekurrenten neuronalen Netzwerken und deren Entscheidungsmechanismen. Darüber hinaus suggerieren die Ergebnisse, dass das Modell mit ausschließlich realen physikalischen Eingangsgrößen eine adäquatere Abbildung des Ausgangswertes bietet als das Modell mit zurückgeführter Ausgangsgröße. Dies wird durch den geringeren RMSE-Wert im Vergleich zur Simulation des LSTM mit der zurückgeführten Ausgangsgröße untermauert.

4.2 Evaluierung des Einflusses verschiedener Lookback-Längen

Das Ziel dieser Untersuchungen ist es, die Performance des Modells weiter zu optimieren und zu überprüfen, ob durch die Modifikation der Daten eine bessere Approximation der realen Prozessbedingungen ermöglicht wird.

Die Analyse der Zeitverlaufdiagramme in Abbildung 10 deutet darauf hin, dass die Länge der Sequenz, der sogenannte *Lookback*-Parameter, einen wesentlichen Einfluss auf die Modelleistung haben könnte. Im Zeitverlaufdiagramm sind spezifische Ausschläge am Ende der Sequenz, also beim ältesten Wert, erkennbar, die für eine Verlängerung der Sequenz bzw. Vergrößerung des *Lookback*-Wertes sprechen könnten. Daher wird im nächsten Schritt die *Lookback*-Länge systematisch variiert, um die Auswirkungen auf die Leistungsfähigkeit des LSTM-Modells zu evaluieren.

Um den Einfluss der Länge des *Lookbacks* auf die Genauigkeit der Modellvorhersage zu untersuchen, bleibt die Modellkonfiguration gemäß Abschnitt 4.1.1 unverändert, während die *Lookback*-Länge variiert wird. Als Vergleichsmetrik dient der RMSE-Wert. Zusätzlich wird das Zeitverlaufdiagramm von *TimeSHAP* genutzt, um den zeitlichen Einfluss der Eingangsgrößen zu visualisieren. Die Ergebnisse dieser Analyse werden in Tabelle 1 dargestellt. Anhand der Tabelle ist ersichtlich, dass sich mit steigendem *Lookback* die Genauigkeit der Vorhersage verbessert.

Diese Erkenntnis wird durch die Zeitverlaufdiagramme von

Tabelle 1: Einfluss des der *Lookbacks* (LB) auf den RMSE

LB	5	10	50	250	500	1000	2000
RMSE	0.778	0.682	0.473	0.153	0.104	0.085	0.042

TimeSHAP in Abbildung 12 bestätigt. In den Zeitverlaufdiagrammen wird das LSTM-Modell mit einem *Lookback* aus der Menge {5, 50, 250, 1000} dargestellt. Aus den Plots geht hervor, dass Modelle mit geringem *Lookback* nicht genügend Informationen für präzise Vorhersagen haben. Dies bedeutet, dass weder die internen Zustände der LSTM-Zelle noch die *Lookback*-Länge ausreichen, um die schnell wechselnden Eingabewerte angemessen zu erfassen.

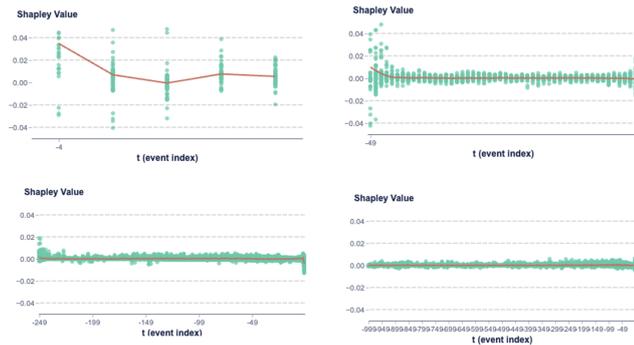


Abbildung 12: Übersicht der verschiedenen *Lookback* Werten im Zeitverlaufdiagramm von *TimeSHAP*

Der *Lookback* sollte so groß wie nötig dimensioniert werden. Wie in Kapitel 2.3 beschrieben, dient der *Lookback* dazu, die internen Zustände der Zelle an die Daten anzupassen, um für den Zeitschritt t eine möglichst präzise Approximation zu ermöglichen. Die Länge des *Lookbacks* bestimmt, wie viele Datenpunkte das Netzwerk benötigt, um die erste Vorhersage treffen zu können. Nach der ersten Modellvorhersage werden zu jedem Zeitpunkt der aktuelle Datenpunkt und die internen Zustände aus den vorherigen Zeitpunkten betrachtet, um den neuen internen Zustand bzw. den Modellausgang zu berechnen.

Obwohl das Ergebnis bei einem *Lookback* von 1000 am besten

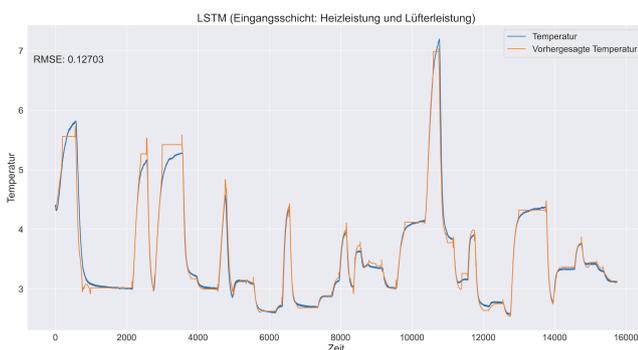


Abbildung 13: Vorhersagegenauigkeit eines LSTM mit einem *Lookback* von 250

ist, wird für die folgenden Untersuchungen ein *Lookback* von 250 gewählt, da die Rechenzeiten sonst zu lang wären. Bei einem *Lookback*-Wert von 250 sind die Shapley-Werte über die gesamte Sequenz gleichmäßig verteilt. Dies ist der erste Modellansatz,

der sowohl die ältesten als auch die aktuellsten Datenpunkte im Zeitverlaufdiagramm in gleichem Maße berücksichtigt. Der Rest der Sequenz wird dabei relativ ausgeglichen genutzt.

Die Genauigkeit der Approximation eines LSTM mit einem *Lookback* Wert von 250 ist in Abbildung 14 zu sehen. Diese Konfiguration des LSTM bildet eine gute Grundlage, um den Einfluss weiterer Konfigurationen zu untersuchen.

4.3 Evaluierung weiterer Modellkonfigurationen

Maschinelles Lernen, insbesondere die Arbeit mit tiefen neuronalen Netzwerken, ist ein empirisches Feld. Es gibt keine allgemeingültigen Regeln, die für alle Datensätze und Anwendungen gleichermaßen gelten. Eine der größten Herausforderungen besteht darin, die Hyperparameter so zu wählen, dass das Modell weder zu komplex noch zu einfach ist.

Das händische Durchprobieren aller möglichen Konfigurationen ist jedoch zeitaufwendig und ineffizient. Daher können Optimierungsalgorithmen benutzt werden, die diesen Prozess beschleunigen und verfeinern. Eine besonders effektive Methode zur Optimierung von Hyperparametern ist die *Bayes'sche Optimierung*.

4.3.1 Bayes'sche Optimierung

"Die *Bayes'sche Optimierung* ist eine leistungsstarke Methode zur Hyperparameteroptimierung, die auf einem Wahrscheinlichkeitsmodell basiert" [20]. Diese Methode nutzt ein sogenanntes *Surrogat-Modell*, das eine vereinfachte Annäherung an die tatsächliche Zielfunktion darstellt. Häufig wird hierfür ein Gaußscher Prozess verwendet. Das *Surrogat-Modell* hilft dabei, die Kosten und Komplexität der direkten Evaluierung der Zielfunktion zu reduzieren.

Eine zentrale Rolle spielt die *Akquisitionsfunktion*, die bestimmt, welcher Punkt im Hyperparameterraum als Nächstes untersucht werden soll. Diese Funktion balanciert zwischen der Erforschung neuer Bereiche (*Exploration*) und der Ausnutzung bereits gefundener guter Bereiche (*Exploitation*). Zu den gängigen *Akquisitionsfunktionen* gehören *Expected Improvement*, *Upper Confidence Bound* und *Probability of Improvement*.

Der Optimierungsprozess beginnt mit der Evaluierung einiger zufällig gewählter Punkte, um initiale Daten für das *Surrogat-Modell* zu sammeln. Anschließend wird das *Surrogat-Modell* iterativ aktualisiert und die *Akquisitionsfunktion* maximiert, um neue Punkte zur Evaluierung auszuwählen. Diese Schritte werden wiederholt, bis ein vordefiniertes Kriterium, wie die Anzahl der Iterationen oder eine zufriedenstellende Modelleleistung, erreicht ist.

In dieser Arbeit wird die *Bayes'sche Optimierung* zur Konfiguration eines LSTM-Modells eingesetzt. Optimiert werden Hyperparameter wie die Anzahl der LSTM-Einheiten, die Anzahl der dichten Einheiten, die *Dropout-Rate* und der Optimierertyp (Adam oder RMSprop). Die *Dropout-Rate* ist ein Regularisierungsparameter, der dazu verwendet wird, Overfitting zu verhindern, indem zufällig ausgewählte Neuronen während des Trainings deaktiviert werden. Ein Optimierertyp bestimmt, wie das Modell seine Parameter anpasst, um die Fehlerfunktion zu minimieren. Adam und RMSprop sind gängige Optimierungsalgorithmen, die für ihre Effizienz und gute Leistung bei tiefen neuronalen Netzwerken bekannt sind.

4.3.2 Durchführung und Ergebnisse

Die *Bayes'sche Optimierung* beginnt mit der Definition der Black-Box-Funktion, welche das LSTM-Modell mit verschiedenen Hyperparameterkombinationen trainiert und evaluiert. Der Optimierungsalgorithmus maximiert die Akquisitionsfunktion, um die vielversprechendsten Hyperparameterkombinationen zu identifizieren und zu testen. Nach einer anfänglichen Phase zufälliger Auswahl folgt eine zielgerichtete Exploration des Hyperparameterraums, basierend auf den Wahrscheinlichkeitsmodellen.

Die Ergebnisse der Optimierung zeigen, dass die *Bayes'sche*

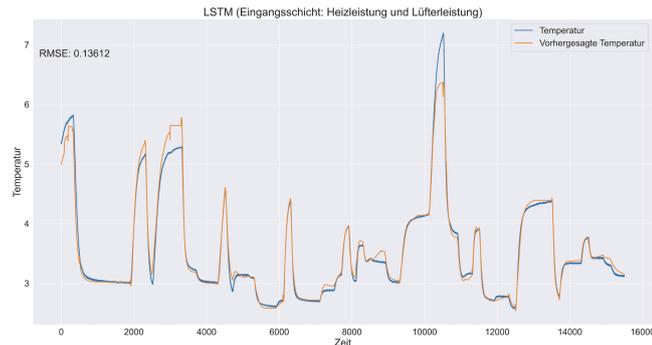


Abbildung 14: Vorhersagegenauigkeit des bayes'schen optimierten LSTM

Optimierung die Evaluierungszeit signifikant reduziert und die Modellgenauigkeit verbessert. Abbildung 14 illustriert die Vorhersagegenauigkeit des optimierten LSTM-Modells, das durch die *Bayes'sche Optimierung* erreicht wurde. Vergleicht man die Ergebnisse mit Abbildung 14, so zeigt sich eine leichte Verbesserung der Modellgenauigkeit.

Tabelle 2 zeigt die optimalen Hyperparameterkombinationen und die entsprechenden RMSE Werte. Die *Bayes'sche Optimierung* hat sich als hilfreich zur Bestimmung der optimalen Hyperparameter eines LSTM-Modells erwiesen. Durch den Einsatz eines Wahrscheinlichkeitsmodells und einer *Akquisitionsfunktion* konnte der Parameterraum effizient durchsucht werden. Diese Methode reduziert die Anzahl der notwendigen Modelltrainings und liefert dennoch eine hohe Genauigkeit bei der Bestimmung der optimalen Hyperparameter. Die Ergebnisse der Optimierung zeigen, dass die *Bayes'sche Optimierung* eine wertvolle Technik zur Verbesserung der Modellleistung in empirischen Studien darstellt.

5 Zusammenfassung und Diskussion

Die vorliegende Arbeit untersucht die Leistungsfähigkeit von LSTM-Modellen zur Approximation physikalischer Prozesse. Im Fokus stehen dabei die Optimierung der Eingangsgrößen und Modellkonfiguration, um die Vorhersagegenauigkeit zu verbessern. Ein wesentliches Ziel dieser Untersuchung ist die Validierung des SFL, also die Berücksichtigung der Ausgangsgröße in der Eingabeschicht. Es zeigt sich, dass die Validierung auf gemessenen Daten die Leistung überschätzt und die Ergebnisse in einer simulierten Umgebung schlechter ausfallen, als wenn das LSTM-Modell mit den realen physikalischen Eingangsgrößen ohne SFL trainiert wird. Dieses Erkenntnis wird durch *TimeSHAP* unterstützt, indem der Einfluss der Eingangsgrößen auf die

Modellvorhersage visualisiert wird. Weitere Optimierungen der Modellkonfiguration wurden ausschließlich an dem LSTM-Netzwerk mit den realen physikalischen Eingangsgrößen durchgeführt, da dieses Netzwerk die Eingangsgrößen gleichmäßig berücksichtigt und den physikalischen Prozess genauer approximieren kann.

Nachfolgend wurde der Parameter des *Lookbacks* untersucht, indem mit *TimeSHAP* und dem RMSE die Genauigkeit der Approximation evaluiert und der Parameter variiert wurde. Im Zeitverlaufsdiagramm von *TimeSHAP* ist ersichtlich, welche Teile der Sequenz das Modell für die Vorhersage am stärksten berücksichtigt. Ein größerer *Lookback*-Wert erzielt bessere Ergebnisse, weil die Änderungen der Eingangsgrößen oft mehrere hundert bis tausend Zeitschritte auseinanderliegen. Dadurch kann das Modell längerfristige Abhängigkeiten besser erfassen und somit präzisere Vorhersagen treffen.

Zuletzt wurde der Einfluss der Modellkonfiguration auf die Genauigkeit der Approximation mit dem Bayes'schen Optimierer untersucht. Diese Methode ermöglicht eine effiziente und präzise Anpassung der Hyperparameter, wie der Anzahl der LSTM-Einheiten, der Dense-Einheiten, der Dropout-Rate und des Optimierertyps. Die Ergebnisse zeigen, dass durch die optimierten Parameter die Modellleistung verbessert wird. Insbesondere wird eine deutliche Reduzierung des RMSE erreicht, was die Effizienz der LSTM-Modelle zur Approximation physikalischer Systeme bestätigt.

Dennoch treten auch Herausforderungen und Einschränkungen auf. Die Länge des *Lookbacks* und sämtliche Modellkonfigurationen müssen für jedes physikalische System individuell angepasst werden und können nicht direkt übernommen werden. Ein weiteres Problem liegt in der Interpretierbarkeit der LSTM-Modelle, da deren interne Verarbeitungsmechanismen schwer zu durchschauen sind. Dies erfordert eine umfassende Datenvorverarbeitung und die sorgfältige Auswahl geeigneter Eingangsgrößen. *TimeSHAP* liefert wertvolle Erkenntnisse über die Eingangsparameter und Sequenzlängen, die das Modell für Vorhersagen am stärksten berücksichtigt. Diese Informationen sind wichtig für die Interpretierbarkeit des Modells. Jedoch bleibt die Wahl der besten Modellkonfiguration, insbesondere die Optimierung der Hyperparameter, eine schwierige und intransparente Aufgabe, die separat behandelt werden muss.

Zukünftige Forschungsarbeit sollte die Untersuchungen auf andere physikalische Systeme und größere Datensätze ausweiten, insbesondere solche mit einer größeren Anzahl an Eingangsgrößen, um die Komplexität des Problems zu erhöhen. Außerdem ist die Erforschung von Transformer-Modellen mit Self-Attention-Mechanismen wichtig, da diese eine Weiterentwicklung der LSTM-Modelle darstellen und die Leistungsfähigkeit neuronaler Netzwerke in industriellen und physikalischen Prozessen weiter verbessern können.

Abschließend lässt sich festhalten, dass LSTM-Modelle leistungsfähige Werkzeuge zur Approximation physikalischer Prozesse sind. Die gewonnenen Erkenntnisse liefern wertvolle Einsichten für den Einsatz von LSTM-Modellen in technischen und industriellen Anwendungen und könnten auch auf andere physikalische Prozesse angewendet werden.

Literatur

- [1] Dynamics evolution prediction from time series data with recurrent neural networks in a complex system, 2022.
- [2] Abd-Elsalam R. Abd-Elhay, Wael A. Murtada, and Mohamed I. Yosof, 'A high accuracy modeling scheme for dynamic systems: spacecraft

Tabelle 2: Ergebnisse der Bayes'schen Optimierung

Iteration	Einheiten der Dense Layer	Dropout Rate	LSTM Einheiten	Optimiererotyp	RMSE
1	85	0.39	2	adam	0.566308
2	31	0.14	39	adam	0.175736
3	30	0.18	38	rmsprop	0.499627
4	35	0.10	41	adam	0.975061
5	30	0.41	41	adam	0.502247
6	33	0.37	36	rmsprop	0.484141
7	31	0.12	33	adam	0.385750
8	35	0.32	31	rmsprop	0.413647
9	33	0.36	29	adam	0.368404
10	35	0.26	27	adam	0.541604
11	29	0.47	30	adam	0.528415
12	31	0.17	26	rmsprop	0.495777
13	38	0.25	33	rmsprop	0.669865
14	30	0.47	35	adam	0.546976
15	30	0.38	22	adam	0.617218
16	33	0.10	33	adam	0.435547
17	14	0.15	161	adam	0.174881
18	16	0.38	160	rmsprop	0.136877
19	14	0.38	158	adam	0.610747
20	16	0.22	163	rmsprop	0.464485
21	18	0.15	159	adam	0.345269
22	12	0.39	163	adam	0.254011

- reaction wheel model', *Journal of Engineering and Applied Science*, **69**(1), (January 2022).
- [3] Ebrahim Balouji, Jonas Sjöblom, Nikolce Murgovski, and Morteza Haghir Chehreghani. Prediction of time and distance of trips using explainable attention-based lstms, 2023.
- [4] Jessica Bartsch, 'Identifikation nichtlinearer dynamischer systeme mit hilfe von long short-term memory-netzen'. Studienarbeit, 1. Prüfer: Prof. Dr.-Ing. Martin Kohlhasse, Abgabedatum: 14. Januar 2021, 2021.
- [5] Joao Bento, André Cruz, and Pedro Saleiro, 'Timeshap: Explaining recurrent models through sequence perturbations', *Feedzai Techblog*, (Jan 2023). Published on Medium.
- [6] Iñigo Bonilla-Campos, Nerea Nieto, Luis del Portillo-Valdes, Bakartxo Egilegor, Jaio Manzanedo, and Haizea Gaztañaga, 'Energy efficiency assessment: Process modelling and waste heat recovery analysis', *Energy Conversion and Management*, **196**, 1180–1192, (September 2019).
- [7] I.T. Cameron and G.D. Ingram, 'A survey of industrial process modelling across the product and process lifecycle', *Computers Chemical Engineering*, **32**(3), 420–438, (2008).
- [8] Huu-Anh-Duc Cap, Trong-Hop Do, Demeke Shumeye Lakew, and Sungrae Cho, 'Building a time-series forecast model with automated machine learning for heart rate forecasting problem', in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, (October 2022).
- [9] Zihan Chang, Yang Zhang, and Wenbo Chen, 'Effective adam-optimized lstm neural network for electricity price forecasting', in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 245–248, (2018).
- [10] Francesco Curreri, Luca Patanè, and Maria Gabriella Xibilia, 'Rnn- and lstm-based soft sensors transferability for an industrial process', *Sensors*, **21**(3), 823, (January 2021).
- [11] Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo, and Maria Gabriella Xibilia, *Soft sensors for monitoring and control of industrial processes*, Advances in Industrial Control, Springer, London, England, 2007 edn., December 2006.
- [12] Yongming Han, Chenyu Fan, Meng Xu, Zhiqiang Geng, and Yanhua Zhong, 'Production capacity analysis and energy saving of complex chemical processes using lstm based on attention mechanism', *Applied Thermal Engineering*, **160**, 114072, (2019).
- [13] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014.
- [14] Harold M Hastings, Jörn Davidsen, and Henry Leung, 'Challenges in the analysis of complex systems: introduction and overview', *Eur. Phys. J. Spec. Top.*, **226**(15), 3185–3197, (December 2017).
- [15] Victor Manuel Hernández-Guzmán and Ramón Silva-Ortigoza, *Physical System Modeling*, 25–85, Springer International Publishing, September 2018.
- [16] Wensi Ke, Dexian Huang, Fan Yang, and Yongheng Jiang, 'Soft sensor development and applications based on lstm in deep neural networks', in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, (November 2017).
- [17] Christophe Krech, *Erklärbarkeit maschineller Lernverfahren: Feature Engineering mit Black-Box-Modellen*, Master of science (m.sc.), Fachbereiche Mathematik und Naturwissenschaften & Informatik, Studiengang Data Science, Germany, 9 2019. Referent: Prof. Dr. Markus Döhring, Korreferent: Prof. Dr. Horst Zisgen.
- [18] Erwin Jose Lopez Pulgarin, Tugrul Irmak, Joel Variath Paul, Arisara Meekul, Guido Herrmann, and Ute Leonards, *Comparing Model-Based and Data-Driven Controllers for an Autonomous Vehicle Task*, 170–182, Springer International Publishing, 2018.
- [19] N. Manaswi, 'Rnn and lstm', 115–126, (2018).
- [20] Mohammad Masum, Hossain Shahriar, Maria Valero, Md Abdullah Khan, Md Jobair Hossain Faruk, Maria Valero, Md Abdullah Khan, Mohammad A. Rahman, Muhaiminul I. Adnan, and Alfredo Cuzzocrea, 'Bayesian hyperparameter optimization for deep neural network-based network intrusion detection', (2022).
- [21] Koduru Nitesh, Yanamala Abhiram, Rayapudi Krishna Teja, and S. Kavitha, 'Weather prediction using long short term memory (lstm) model', in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, (January 2023).
- [22] Yasith S. Perera, D.A.A.C. Ratnaweera, Chamila H. Dasanayaka, and Chamil Abeykoon, 'The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review', *Engineering Applications of Artificial Intelligence*, **121**, 105988, (May 2023).
- [23] Ivan Pisa, Ignacio Santín, Antoni Morell, Jose Lopez Vicario, and Ramon Vilanova, 'Lstm-based wastewater treatment plants operation strategies for effluent quality improvement', *IEEE Access*, **7**, 159773–159786, (2019).
- [24] Ehsan Rokhsatyazdi, Shahryar Rahnamayan, Hossein Amirinia, and Sakib Ahmed, 'Optimizing lstm based network for forecasting stock market', in *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–7, (2020).
- [25] Arjun Singh Saud and Subarna Shakya, 'Analysis of look back period for stock price prediction with rnn variants: A case study on banking sector of nepse', *Procedia Computer Science*, **167**, 788–798, (2020).
- [26] L. S. Shapley, *17. A Value for n-Person Games*, 307–318, Princeton University Press, December 1953.
- [27] Feifan Shen, Jiaqi Zheng, Lingjian Ye, and Xiushui Ma, 'Lstm soft sensor development of batch processes with multivariate trajectory-based ensemble just-in-time learning', *IEEE Access*, **8**, 73855–73864, (2020).
- [28] Hendrik Strobelt, Sebastian Gehrmann, H. Pfister, and Alexander M. Rush, 'Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks', *IEEE Transactions on Visualization and*

- Computer Graphics*, **24**, 667–676, (2016).
- [29] Jian Sun, Xi Meng, and Junfei Qiao, 'Prediction of oxygen content using weighted pca and improved lstm network in mswi process', *IEEE Transactions on Instrumentation and Measurement*, **70**, 1–12, (2021).
- [30] Bo Wang, Yongxin Nie, Ligang Zhang, Yongxian Song, and Qiwei Zhu, 'An soft-sensor method for the biochemical reaction process based on lstm and transfer learning', *Alexandria Engineering Journal*, **81**, 170–177, (October 2023).
- [31] Philip B. Weerakody, K. K. F. Wong, Guanjin Wang, and W. Ela, 'A review of irregular time series data handling with gated recurrent neural networks', *Neurocomputing*, **441**, 161–178, (2021).
- [32] Dou Wei, 'Prediction of stock price based on lstm neural network', in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. IEEE, (October 2019).
- [33] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang, 'A review of recurrent neural networks: Lstm cells and network architectures', *Neural Computation*, **31**(7), 1235–1270, (2019).
- [34] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li, 'Predictive maintenance in the industry 4.0: A systematic literature review', *Computers amp; Industrial Engineering*, **150**, 106889, (December 2020).