

Abschlussbericht des Projekts der Deutschen Bundesstiftung Umwelt

Prädiktive Foulingdetektion in der Lebensmittelproduktion mit Modellen basierend auf realen Daten – FideLio

AZ 37305/01-36

Projektzeitraum 01.02.2022 – 31.07.2024

Berichtszeitraum 01.02.2022 – 31.07.2024

Antragssteller

Technische Universität Braunschweig
Institut für Chemische und Thermische
Verfahrenstechnik (ICTV)
Langer Kamp 7, 38106 Braunschweig

Dr.-Ing. Wolfgang Augustin
Akademischer Direktor und Leiter der
AG Fouling & Reinigung
Telefon: +49(0)531 391 2789
E-Mail: w.augustin@tu-braunschweig.de
Web: <http://www.ictv.tu-braunschweig.de>

Fraunhofer-Institut für Verfahrenstechnik und
Verpackung (IVV), Institutsteil Verarbeitungstechnik
Heidelberger Str. 20, 01189 Dresden

Dipl.-Ing. Jialiang Yin
Team Data Science, Digitalisierung und Assistenzsysteme
Telefon: +49(0)351 43614 33
E-Mail: jialiang.yin@ivv-dd.fraunhofer.de
Web: <https://www.ivv.fraunhofer.de>

30. Oktober 2024

Inhaltsverzeichnis

1.	Zusammenfassung.....	3
2.	Einführung und Motivation	3
3.	Methodik, Vorgehensweise und Projektablauf	4
4.	Projektergebnisse	5
4.1.	AP1: Sammeln und Aufbereiten von historischen Daten.....	5
4.2.	AP2: Untersuchung der Eignung von externen Clamp-On Messsystemen	16
4.3.	AP3: Aufbau eines dezentralen Messsystems zur Datenerfassung und Foulingdetektion.....	21
4.4.	AP4: Schaffung eines Gesamtsensorkonzeptes für neue Anlagen	24
4.5.	AP5: Aufbau von Modellen und Methoden zur Datenaufbereitung	25
5.	Öffentlichkeitsarbeit, Veröffentlichungen und Vorträge	50
6.	Fazit und Ausblick.....	51
7.	Literaturangaben.....	53

1. Zusammenfassung

Der Klimawandel und die damit verbundenen Gegenmaßnahmen stellen alle Industrien in Deutschland vor große Herausforderungen. Einer der wichtigsten Einflussfaktoren ist das Treibhausgas Kohlenstoffdioxid (CO₂), welches aufgrund dieser Rolle zu einer wichtigen Komponente für Einsparmaßnahmen geworden ist. Gerade im Bereich Energieversorgung (Heizprozesse, Pumpleistung), Frisch- und Abwasser sowie Chemikalienbedarf gibt es gerade im Bereich der Lebensmittelindustrie große Einsparpotentiale. Wie so gut wie jeder andere Bereich der chemischen Industrie hat auch die Lebensmittelverarbeitung mit ungewünschten Ablagerungen (Fouling) zu kämpfen, welches umfassende Reinigungsschritte notwendig macht. Hier werden aufgrund der hohen Anforderungen an die Produktsicherheit große Sicherheiten in Bezug auf einzusetzende Energie, Spülwasser- und Reinigungsmittelmenge aufgeschlagen. Eine sensorbasierte Erfassung des Verschmutzungszustands mit einer anschließenden Vorhersage auf Basis von Modellen birgt erhebliches Einsparpotential im Bezug auf die oben genannten Ressourcen und verkörpert aufgrund des hohen Marktanteils dieses Industriezweigs einen wichtigen Teilschritt zur Erreichung klimapolitischer Ziele.

Das erste Kernelement dieses Forschungsvorhabens ist die Vorbereitung der Weiterverarbeitung von historischen Produktionsdaten für das Modelltraining. Hier wurden alle notwendigen Informationen eingeholt und Unterlagen für die Weitergabe an die Industriepartner erstellt. Weiterhin wurde aus einer umfassende Marktübersicht ein Clamp-On-Messsystem ausgewählt und in ein im Projekt entwickeltes Internet of Things (IoT)-System integriert, getestet und charakterisiert. Dies wurde durch ein erweiterbares dezentrales System zur Erfassung und Abspeicherung der Daten und Übertragung der Nachrichten umgesetzt. Weiterhin wurden bereits erste Schritte für die Generierung eines Gesamtsensorkonzeptes gemacht, indem in einer umfassenden Versuchsreihe systematisch das Foulingverhalten von Anlagenkomponenten verglichen wurde, um potentielle Messorte für die Sensorplatzierung zu charakterisieren. Für das erste Antrainieren der Vorhersagemodelle wurde ein umfassendes Screening an einer Plattenwärmeübertrager-Foulinganlage im Technikumsmaßstab durchgeführt. Die hier generierten Daten dienen als Vorbereitung für das Training mit historischen Produktionsdaten bzw. den durch den Einsatz der Clamp-On-Sensoren zusammen mit dem IoT-System generierten Daten. Bei den Vorhersagemodellen handelt es sich um eine Zeitreihenvorhersage. Zur Umsetzung der Zeitreihenvorhersage wurde das Sequenz-zu-Sequenz-Konzept verwendet. Mithilfe dieses Konzepts wurden drei Deep-Learning-Modelle entwickelt und miteinander verglichen. Anschließend wurde ein Demonstrator zur Veranschaulichung der Vorhersagen erstellt. Auf nationalen und internationalen Konferenzen sowie in verschiedenen Veröffentlichungen konnten bereits große Teile der Projektergebnisse einem breiten Fachpublikum präsentiert werden.

2. Einführung und Motivation

Um die von der Bundesregierung angestrebten Ziele im Bezug auf Klimaschutz und Resourceneinsparung zu erreichen, sind weitgreifende Transformationen, insbesondere in

der fertigenden Industrie, notwendig. Dabei weist die Lebensmittelindustrie aufgrund der hohen hygienischen Anforderungen sowie einer großen Produktvielfalt ein signifikantes Einsparpotential auf. Dies gilt insbesondere für Prozesse, in welchen Lebensmittel thermisch behandelt werden und hierbei insbesondere in Bezug auf ungewollte Ablagerungen und deren anschließende Reinigung. Um diese optimieren zu können, müssen große Mengen an anfallenden Prozessdaten gesammelt, aufbereitet und analysiert werden. Darauf aufbauend können Modelle auf Basis von Künstlicher Intelligenz (KI) helfen, genauere Vorhersagen über den Verschmutzungszustand und die gezielte notwendige Reinigung zu machen. Für das Trainieren von Modellen des Maschinellen Lernens (ML) ist eine große Datenbasis essentiell, welche das jeweilige Problem im Idealfall möglichst ganzheitlich abbilden können soll.

Damit neben den historischen Prozessdaten ebenfalls neu generierte Daten erhalten werden können, müssen zusätzliche Sensoren eingesetzt werden. Da die üblicherweise in einer Produktionsanlage platzierten Sensoren nicht vor dem Hintergrund verbaut wurden, das Foulingverhalten eines Apparates besser vorherzusagen, sondern um wichtige Größen für die Prozesssteuerung zu erfassen, ist Wissen über den Prozess und das Foulingverhalten des vorliegenden Stoffsystems erforderlich. Dies ist essentiell, um die optimale Auswahl und Platzierung dieser Sensoren zu erreichen. Diese können dann an weiteren, für das Modelltraining vielversprechenden, Messstellen in einer Produktionsanlage angebracht werden. Da Produktionsanlagen im Lebensmittelbereich jedoch einer sehr hohen Anforderung bezüglich Hygiene unterliegen und in der Regel als Komplettanlagen zugekauft werden, gestaltet sich der nachträgliche, invasive Einbau von Sensoren als schwierig. Hier bietet sich die Verwendung von Clamp-On-Sensoren an, welche als Anlegemessgerät von außen an ein Rohr der Produktionsanlage angeklemmt werden können.

Um die Daten der Clamp-On-Sensoren flexibel, mobil und möglichst effizient einzusetzen zu können, werden sie in ein im Projektrahmen konzeptioniertes IoT-System eingebunden. Eine Kombination aus physikalischen Zusammenhängen und ML wird zur Erstellung von Vorhersagemodellen eingesetzt, wobei historische Daten aus industriellen Prozessen und IoT-Geräten mit erweiterbarer Sensorik als Datenquelle dienen. Die IoT-Geräte ermöglichen es, Daten in Echtzeit an die Modelle zu übermitteln, um bei kritischen Werten eine Meldung vor Ort zu generieren. Das vorgestellte Konzept kann sowohl für bestehende als auch für neue Anlagen angewendet werden und ist auf andere Branchen wie z.B. die Chemie- und Pharmaindustrie übertragbar.

3. Methodik, Vorgehensweise und Projektablauf

Um das übergeordnete Ziel eines autarken Gesamtsystems zur Prädiktion des Foulingverhaltens zu erreichen, müssen diverse Zwischenziele erreicht werden. So vielfältig wie die angestrebten Ziele sind dabei auch die notwendigen Schritte und Methoden. Für die erforderliche breite Datenbasis sind Prozessdaten aus Industrieanwendungen unerlässlich. Hierfür wurden in enger Zusammenarbeit mit Industriepartnern Anforderungs- und Fragenkataloge entwickelt, welche die Vorbereitung und Bereitstellung von historischen

Industriedaten ermöglichen. Diese werden dann für die Verwendung als Trainingsdaten vorbereitet und die entsprechenden Modelle trainiert.

Ein weiterer Schritt, welcher im Projektablauf parallel erfolgte, ist die Generierung von Trainingsdaten mittels eines Modellstoffsystems an einer Anlage im Technikumsmaßstab unter Laborbedingungen. Hier konnte ein typischer Prozess unter definierten Bedingungen mit einer Vielzahl an Sensoren untersucht werden, um eine hohe Anzahl an breit gestreuten Trainingsdaten zu generieren. Hier konnte ebenfalls das im Projektrahmen entwickelte IoT-System für die berührungslose Erfassung von Temperaturdaten erprobt werden und bereits wichtige Informationen liefern.

Durch den Einsatz von nicht-invasiven Sensoren kann das System an verschiedenen Orten nach Bedarf installiert werden. Zunächst ist die Entwicklung eines industrietauglichen Messsystems erforderlich, um Daten zu erfassen, die über Netzwerk übertragen werden können. In diesem Zusammenhang bietet sich der Einsatz eines Raspberry Pi an, der über eine Netzwerkschnittstelle verfügt und mit einem Betriebssystem ausgestattet ist, womit mehrere Hilfsprogramme ausgeführt werden können. Zur Realisierung eines dezentralen Systems soll ein microservice-orientiertes Software-Framework eingesetzt werden.

4. Projektergebnisse

4.1. AP1: Sammeln und Aufbereiten von historischen Daten

Für die verlässliche Vorhersage von Ablagerungen in einer Produktionsanlage sind Daten, die beim langfristigen Betrieb einer solchen Anlage generiert wurden, unabdingbar. Diese sollten im Projektrahmen von den Mitgliedern des projektbegleitenden Beirats zur Verfügung gestellt werden. Damit diese Zusammenarbeit möglichst effizient gestaltet werden kann, wurden im Vorfeld der Anfrage von den Forschungspartnern Anforderungs- und Fragenkataloge erstellt, mit welchen die Industriepartner prüfen konnten, ob geeignete Datensätze verfügbar sind. Tabelle 1 fasst die Sammlung relevanter Prozessinformationen für die im Projektrahmen besonders betrachteten Wärmeübertrager zusammen; als Beispiel für Apparate, welche mit der höchsten Relevanz für die Modellgenerierung und das Training bewertet wurden.

Tabelle 1: Sammlung relevanter Prozessinformationen für die Abfrage historischer Produktionsdaten bei Industriepartnern.

Parameter	Details
Temperatur	Wärmeübertrager (Produkt Ein- & Auslass, Heizmedium Ein- & Auslass)
Druck	Wärmeübertrager (Produkt Ein- & Auslass, Heizmedium Ein- & Auslass)
Volumenstrom/ Strömungsgeschwindigkeit	Produktstrom, Heizmedium
Wärmeübertragergröße und -geometrie sowie Abmessungen	Rohr- oder Kanalgröße, -anzahl und allgemeine Abmessungen
Produkt	Dichte, Konzentration, Viskosität, Trübung, ...
Pumpenparameter	Drehzahl, Drehmoment, Frequenz, Fehlermeldungen, ...

Zusatzinformationen	Fehlermeldungen (Foulingereignisse betreffend), notwendige Abweichungen von üblichen Reinigungsprotokollen (Musste ggf. länger gereinigt werden als sonst, um einen Apparat sauber zu bekommen bzw. die Ausgangsparameter wieder zu erreichen?)
---------------------	---

Neben den prozessrelevanten Informationen, welche die Datensätze enthalten, sind eine Reihe von weiteren Aspekten wichtig, um diese Daten zielgerichtet verarbeiten zu können. Tabelle 2 stellt den dazu erstellten Fragenkatalog dar, welcher bei der Einordnung der Datensätze im Bezug auf z. B. Datenherkunft, -format und -typ wichtig für die weitere Verarbeitung ist. Die abgefragten Informationen sind nur beispielhaft und mussten für eine erfolgreiche Verarbeitung nicht vollständig angegeben werden. Die Algorithmen des ML sind gerade auch deshalb so leistungsfähig, weil sie aus dem Datensatz selbst bereits eine große Menge an Zusammenhängen und Informationen ableiten können. Jedoch sind Zusatzinformationen in fast jedem Fall immer hilfreich und können im besten Fall eine Vorhersage noch genauer machen bzw. ggf. fehlende oder lückenhafte Daten kompensieren.

Tabelle 2: Fragenkatalog für die Sammlung relevanter Hintergrundinformationen über die Herkunft der Prozessdaten für die Datenanalyse.

Kategorie	Frage
Datenherkunft	<ul style="list-style-type: none"> - Wurde der Datensatz von einem physischen Anlagensor sensor generiert? - Falls ja, von welchem (Position und Messgröße)?
Datenformat	<ul style="list-style-type: none"> - In welchem Datenformat liegt der Datensatz vor (ganzzahlig, boolesch, real, komplex)?
Datentyp	<ul style="list-style-type: none"> - Liegen die Daten als Datei (.txt, .csv, .xlsx, ...) oder in einer Datenbank (relational, dokumentenorientiert, ...) vor?
Labelling	<ul style="list-style-type: none"> - Wie sind die Datensätze gelabelt? - Gibt es eine hinter liegende Struktur, die offengelegt werden kann?
Zusatzinformationen	<ul style="list-style-type: none"> - Sind Zusatzinformationen verfügbar? - Können Beobachtungen von z. B. Anlagenpersonal mit Messdaten synchronisiert werden?
Organisatorisches	<ul style="list-style-type: none"> - Wie können die Daten übertragen werden (Datenträger, Cloud-Service, ...)? - Müssen Geheimhaltungen beachten werden?

Die erstellten Anforderungs- bzw. Fragenkataloge wurden an Industriepartner übergeben.

4.1.1. Analyse und Aufbereitung der Daten aus Molkereiindustrie

Im Projektzeitraum wurden zahlreiche Unternehmen angefragt; letztendlich haben wir Zusagen von zwei Firmen erhalten. *Firma 1* hat die Produktionsdaten von zehn Tagen zusammengefasst und an uns weitergeleitet, während *Firma 2* uns die Berechtigung erteilt hat, das Messsystem an einem Wärmeübertrager zu installieren und damit die Produktionsdaten zu erfassen.

4.1.1.1. Datenanalyse – *Firma 1*

Firma 1 hat die Daten direkt aus der Steuerung abgerufen und in einer Excel-Tabelle gespeichert. Die von *Firma 1* übermittelten Daten enthalten sowohl Produktions- als auch Reinigungsdaten. Da das Ziel des Projekts die Vorhersage der Foulingmasse ist, wurde bei der Auswertung der Fokus auf die Produktionsdaten gelegt. Diese Daten umfassen umfangreiche Sensordaten, wie Motorleistung der Förderpumpe, Motorleistung der Boosterpumpe, Ein- und Ausgangstemperaturen des Produkts und des Spülwassers, Temperatur- und Druck des Erhitzers sowie der Kühlung und des CIP (Clean-in-Place).

Des Weiteren wurde eine Anleitung zur Erläuterung der Daten bereitgestellt, die auch eine schematische Darstellung des Gesamtsystems und die Bedeutung der einzelnen Prozessschritte enthielt. In Abbildung 1 ist die schematische Darstellung des Wärmeübertragers präsentiert. Der Rohstoff wird während der Produktion in mehreren Schritten verarbeitet:

1. Der Rohstoff wird über die Förderpumpe zum Regenerator weitergeleitet, in dem der Rohstoff durch das erwärmte Produkt vorgewärmt wird.
2. Danach leitet die Boosterpumpe den Rohstoff in den Erhitzer weiter, wo er durch den Heißwasserstrom vollständig erhitzt wird. Während der Produktion wird die Ausgangstemperatur des Erhitzers kontrolliert und überwacht. Aus lebensmittelsicherheitsrechtlichen Gründen muss der Temperaturwert mindestens einen festgelegten Minimalwert erreichen. Die Regelgröße ist die Produkttemperatur am Ausgang. Sollte die Ausgangstemperatur des Produkts den Minimalwert nicht erreichen, muss die Temperatur des Heißwasserstroms erhöht werden. Wenn Fouling auftritt, steigt diese Temperatur weiter an.
3. Nach der Erwärmung im Erhitzer wird der Rohstoff als Hitzemedium in den Regenerator eingespeist, wodurch das Produkt gekühlt wird. Nach dem Regenerator wird das Produkt weiter gekühlt und anschließend in einem Tank gelagert.

Bei den oben genannten Prozessschritten kann jeweils Fouling auftreten, jedoch sollte beim Kühlen nach dem Regenerator kein weiteres Fouling mehr auftreten.

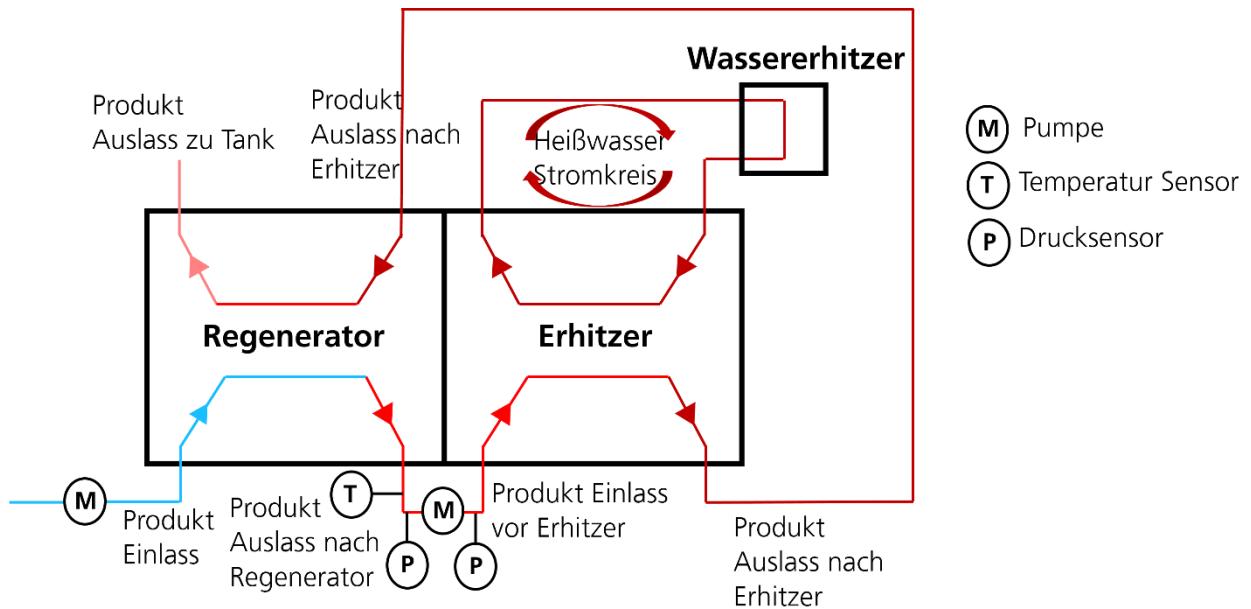


Abbildung 1 Schematische Darstellung von Wärmeübertrager mit Regenerator aus *Firma 1*

Die Produktionsdaten wurden basierend auf den Prozessschritten extrahiert und nach Datum sortiert. Insgesamt wurden innerhalb von 10 Tagen 16 Produktionen durchgeführt. Die Produktionsdaten werden einer eindeutigen Produktions-ID zugewiesen, sodass eine Reihe von Daten mit Produktions-IDs von 'production_0' bis 'production_15' verfügbar waren. Diese Daten sind für das Training, die Validierung und das Testen des Modells vorgesehen. Eine Aufbereitung der Daten war jedoch zunächst erforderlich.

Das Hauptziel des Projekts ist die Vorhersage der Foulingmasse, welche im echten Betrieb schwer zu bestimmen ist, da die Produktion nicht gestört werden kann. Deshalb wurden foulingrelevante Parameter untersucht. Sobald sich während der Produktion Fouling im Wärmeübertrager bildet, wird die Effizienz des Wärmeübergangs und hydrodynamische Leistung beeinträchtigt. Die Temperatur und der Druck sind also entscheidende Indikatoren, da die Foulingschicht, die sich auf dem inneren Rohr oder der Platte des Wärmeübertragers ablagert, den freien Strömungsquerschnitt verringert und eine Temperatur- und Druckveränderung verursacht. Aus diesen Gründen ergeben sich zwei Annahmen:

1. Die Temperatur am Produktausgang nach dem Erhitzer bleibt konstant. Daher sollte auch die Temperatur nach dem Regenerator, d.h. vor dem Tank, ohne das Vorhandensein von Fouling konstant bleiben. Sollte diese Temperatur jedoch im Laufe der Zeit vom konstanten Wert nach oben abweichen, lässt dies auf Fouling schließen.
2. Der Volumenstrom soll gemäß der Betriebsanleitung konstant sein. Bei Ablagerung von Foulingmasse muss die Leistung der Boosterpumpe erhöht werden, was zu einem Anstieg des Drucks am Produktausgang nach dem Regenerator führt.

Abbildung 2 zeigt normalisierte Werte der Produktausgangstemperatur nach dem Regenerator. Es ist zu beobachten, dass die Temperatur während der Produktion kontinuier-

lich sinkt, obwohl das Heizmedium nach dem Erhitzer eine konstante Temperatur aufweist. Dies deutet darauf hin, dass die Wärmeübertragungseffizienz im Regenerator durch Fouling beeinträchtigt wird.

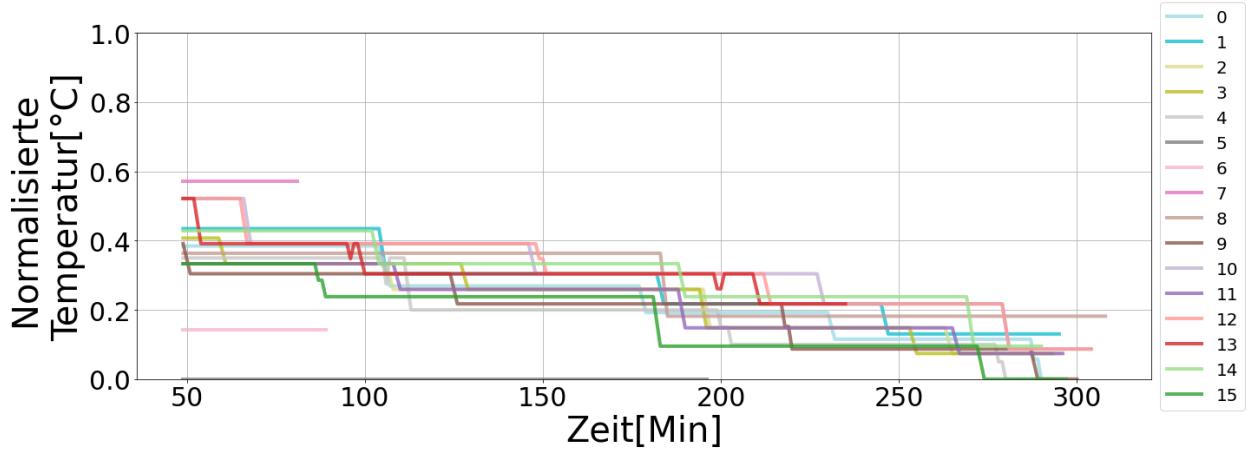


Abbildung 2: Normalisierte Temperaturwerte am Produktausgang nach dem Regenerator für ProduktionsIDs von 0 bis 15 während der Produktion

In den folgenden Abbildungen werden die Motorleistung der Boosterpumpe und der Druck am Produkteingang vor dem Erhitzer dargestellt. Die Steuerung des Heizprozesses erfolgt über die Produkttemperatur, die nach dem Erhitzer auftritt, wobei diese Temperatur einen minimalen Wert erreichen muss. Wie bereits erwähnt, muss im Falle von Fouling die Temperatur des Heizmediums erhöht werden, um einen möglichen Rückgang des Wärmeübergangs zu kompensieren. Außerdem ist es erforderlich, dass das Produkt einen konstanten Volumenstrom in der Anlage durchläuft. Das Produkt wird kontinuierlich mit einer gleichbleibenden Fördermenge durch die Booster-Pumpe gepumpt. Beim Auftreten von Foulingschichten muss die Leistung der Booster-Pumpe zur Kompensation des reduzierten Volumenstrom erhöht werden, wodurch der Druck auf der Ausgangsseite der Booster-Pumpe steigt.

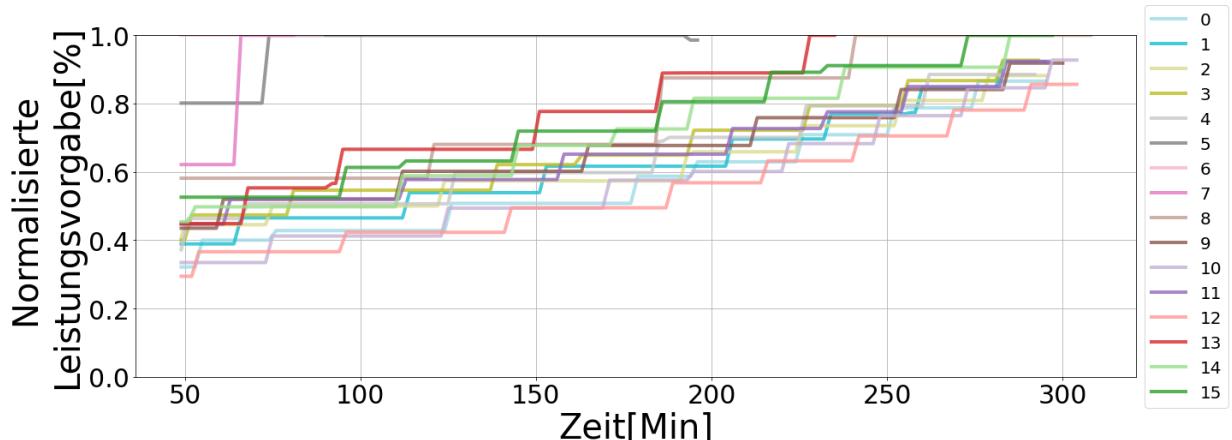


Abbildung 3: Normalisierte Booster Pumpe Leistung für ProduktionsIDs von 0 bis 15 während der Produktion

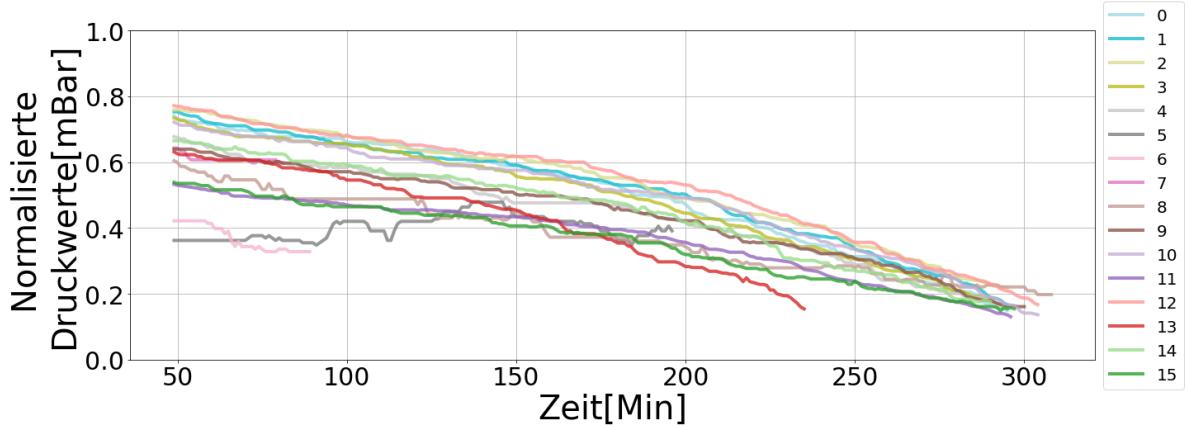


Abbildung 4: Normalisierte Druckwerte am Produktausgang nach dem Regenerator vor dem Erhitzer

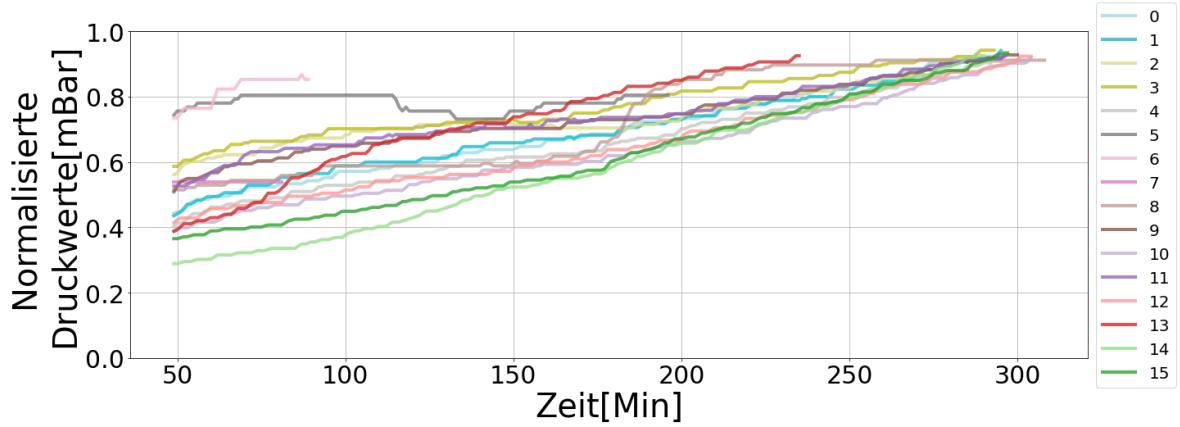


Abbildung 5: Normalisierte Druckwerte am Produkteingang nach dem Regenerator vor dem Erhitzer

Neben der Darstellung der normalisierten Daten wird auch das Integral der Signalautokorrelationsfunktion berechnet. Zuerst wird über die Formel:

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}}$$

die Korrelation zwischen zwei Signalen bestimmt, wobei C_{ij} die Kovarianz und C_{ii} die Varianz des Signals darstellt. Da hier die Autokorrelationsfunktion bestimmt wird, muss zuerst ein Verzögerungsintervall definiert werden. Die maximale Länge des Intervalls beträgt die Hälfte der gesamten Signallänge. Mit dem Intervall lässt sich eine Autokorrelationsfunktionskurve berechnen, die eine Funktion der Verzögerung ist. Um die Autokorrelationsfunktion besser zu quantifizieren, kann das Integral der Funktion berechnet werden. Aus dem Integral lässt sich deutlich erkennen, wie stark das Signal mit sich selbst korreliert. Je größer das Integral ist, desto stärker ist die Abhängigkeit des Signals von sich selbst. Aus der folgenden Tabelle 3 lässt sich auch erkennen, dass die ausgewählten Sensoren eine relativ große Autokorrelation aufweisen. Deshalb werden die Sensoren als Feature für Modelltraining verwendet.

Tabelle 3: Integral der Autokorrelationsfunktion der ausgewählten Sensoren

Sensor	Integral der Autokorrelationsfunktion
Reg_Produkt_Aus_T	0,5594
Booster_Pumpe_Leistung	0,8231
Reg_Produkt_Aus_P	0,6272
Erhitzer_Produkt_In_P	0,7059

4.1.1.2. Datenanalyse – *Firma 2*

Firma 2 bietet die Möglichkeit, die Temperaturwerte vor Ort zu messen. Dabei sind jedoch einige Anforderungen zu beachten:

1. Keine Eingriffe in die Hardware: Die Produktionsanlage darf durch unseren Systemaufbau weder geschnitten noch gebohrt werden.
2. Kein WLAN: Es steht kein WLAN zur Verfügung.
3. Kein Eingriff in das Steuerungssystem: Das Steuerungssystem der Anlage darf nicht beeinflusst werden.
4. Keine Produktionsunterbrechung: Die Produktion darf nicht gestört werden.

Zur Umsetzung dieser Anforderungen wurde eine Messbox entwickelt, deren genauer Aufbau im Abschnitt 4.3.1.2 beschrieben wird. Um Eingriffe in die Anlage zu vermeiden, verwenden wir ausschließlich Clamp-On-Temperatursensoren für die Vor-Ort-Messungen, die vier Sensorwerte zur Verfügung stellen:

- Clamp-On Produkteingangstemperatur ($C_T_Produkt_Ein$),
- Clamp-On Produktausgangstemperatur ($C_T_Produkt_Aus$),
- Clamp-On Heißwassereingangstemperatur ($C_T_Heißwasser_Ein$) und
- Clamp-On Heißwasserausgangstemperatur ($C_T_Heißwasser_Aus$).

In der Molkereiindustrie ist geregelt, dass die Produktausgangstemperatur einen minimalen Wert erreichen muss. Wie bereits beschrieben, gilt die Annahme, dass bei einer Foulingbildung die Heißwasserausgangstemperatur ansteigen wird. Ziel ist es, durch die Modellbildung den steigenden Trend der Heißwasserausgangstemperatur vorherzusagen.

Das System misst kontinuierlich die vier Temperaturwerte und speichert die erfassten Daten in einer Datenbank. Daher werden die Zeitpunkte, zu denen die Produktion gestartet und gestoppt wird, ebenfalls erfasst. Um diese Information zu verarbeiten, stellt *Firma 2* einen Produktionsplan im .txt-Format zur Verfügung. Aus dieser Datei werden die relevanten Produktionsstart- und -stopzeitstempel extrahiert und als Query in der Datenbank verwendet, um die Produktionsdaten zu analysieren.

Der Produktionsplan umfasst ca. 40 Tage und dabei wurden die Sensorwerte alle 30 Sekunden abgefragt. Im Vergleich zum Datensatz von *Firma 1* stehen uns somit umfangreichere Daten für die Modellbildung zur Verfügung. Aufgrund der begrenzten Auswahl

an Sensoren haben wir zusätzlich zwei Softsensoren basierend auf den vorhandenen Daten entwickelt:

- Der Differenzwert zwischen der Heißwassereingangs- und der Produktausgangstemperatur ($C_T_{Diff_HW_Ein_Prod_Aus}$).
- Der Differenzwert zwischen der Produktausgangs- und der Heißwasserausgangstemperatur ($C_T_{Diff_HW_Aus_Prod_Aus}$).

Somit stehen uns neben den vier Hardware-Sensoren nun auch zwei zusätzliche Softsensoren zur Verfügung.

Filterung der Daten zur Rauschentfernung

Da die Daten stark verrauscht sind, müssen Filter eingesetzt werden, um das Rauschen zu beseitigen. Aufgrund eigener Erfahrungen wurde ein Savitzky-Golay-Filter sowie die Methode des gleitenden Mittelwerts angewendet. Der Savitzky-Golay-Filter ist ein Verfahren zur Glättung von Daten, bei dem kleine Gruppen benachbarter Punkte (Fenster) mit einem Polynom niedrigen Grades modelliert werden. Als Fittingskriterium wird dabei die Methode der linearen kleinsten Quadrate verwendet. Ein entscheidender Vorteil dieses Filters besteht darin, dass er den Charakter des Signals, insbesondere wichtige Merkmale wie Spitzen und Vertiefungen, beibehält – ein Aspekt, der bei anderen Glättungsmethoden oft verloren geht. Der zweite verwendete Filter ist der gleitende Mittelwert. Diese Methode berechnet den Mittelwert innerhalb eines definierten Bereichs benachbarter Punkte. Während der Savitzky-Golay-Filter das Rauschen effizient reduziert und die Signalmerkmale beibehält, tendiert die gleitende Mittelwert-Methode dazu, auch Merkmale in benachbarten Punktbereichen zu glätten. Da eine Produktionslaufzeit etwa 8 bis 9 Stunden beträgt, fällt eine große Menge an Daten an. Die direkte Anwendung der Methode des gleitenden Mittelwerts ist daher weniger effizient als der Einsatz des Savitzky-Golay-Filters. Wie in Abbildung 6 zu sehen ist, wird bei einer Fenstergröße von 15 immer noch viel Rauschen beibehalten. Abbildung 7 zeigt, dass das Signal bei einer Fenstergröße von 100 zu stark vom ursprünglichen Signal abweicht.

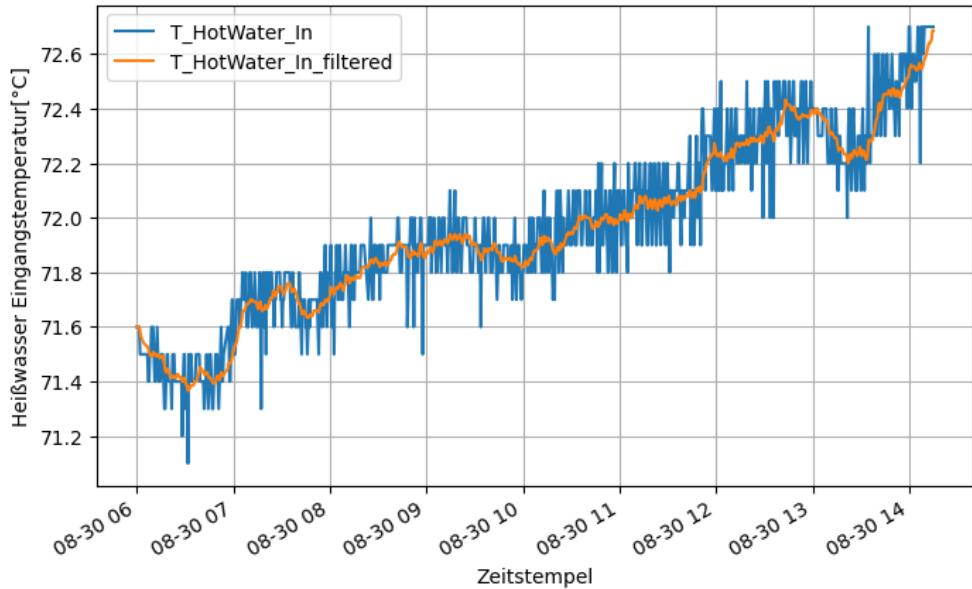


Abbildung 6: Vergleich des Heißwasser Eingangstemperaturverlaufs, Rohdaten vs. gefilterte Daten mit gleitendem Mittelwert (Fenstergröße 15)

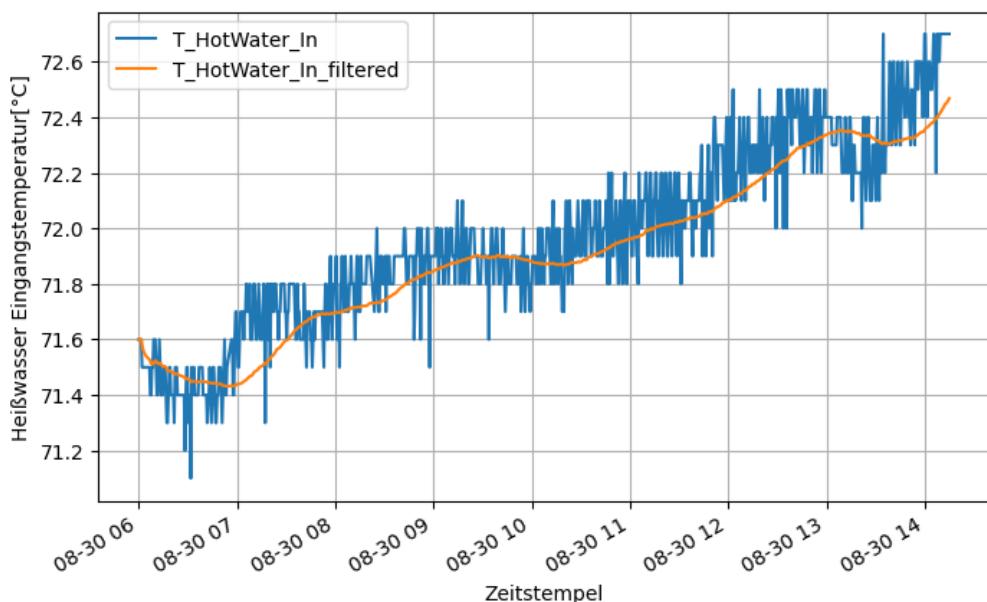


Abbildung 7: Vergleich des Heißwasser Eingangstemperaturverlaufs, Rohdaten vs. gefilterte Daten mit gleitendem Mittelwert (Fenstergröße 100)

Die alleinige Anwendung des Savitzky-Golay-Filters reicht jedoch nicht aus. In Abbildung 8 ist zu sehen, dass das gefilterte Signal zwar die wesentlichen Merkmale beibehält und das Rauschen bereits stark reduziert ist. Bei genauerer Betrachtung enthält das gefilterte Signal jedoch noch Restrauschen mit kleiner Amplitude. Deshalb kann ergänzend die Methode des gleitenden Mittelwerts mit einer kleinen Fenstergröße eingesetzt werden. In Abbildung 9 ist das vollständig gefilterte Signal dargestellt.

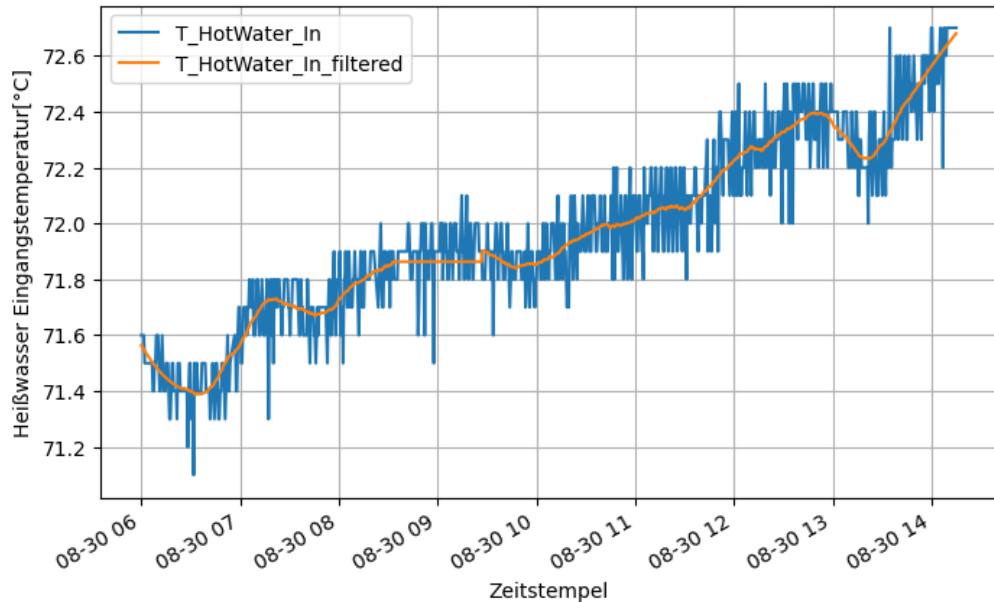


Abbildung 8: Vergleich des Heißwasser Eingangstemperaturverlaufs, Rohdaten vs. gefilterte Daten mit der Savitzky-Golay-Methode (Fenstergröße 100, Grad 2)

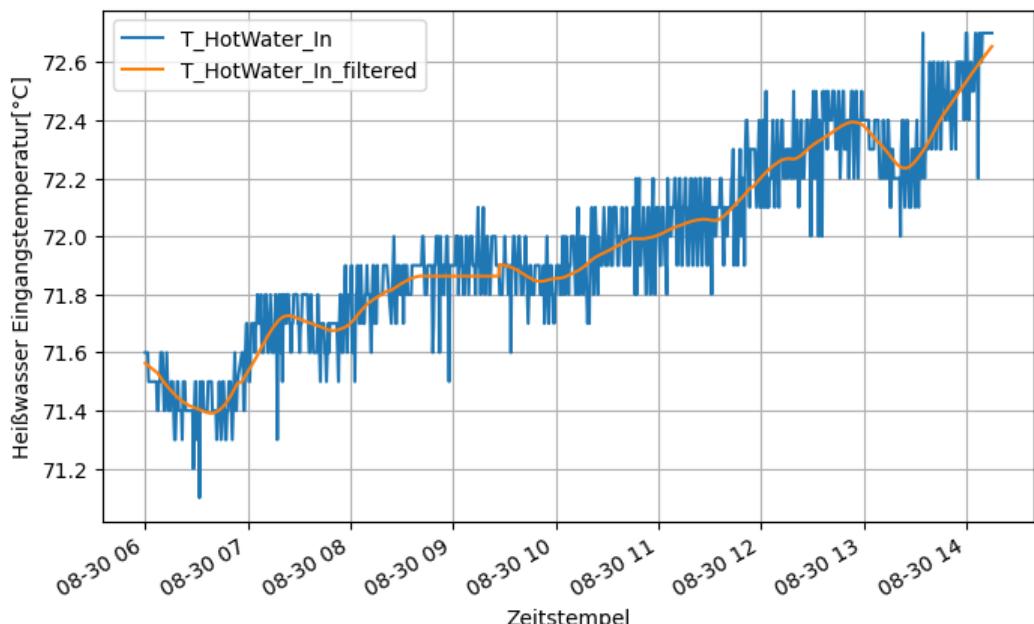


Abbildung 9: Vergleich des Heißwasser Eingangstemperaturverlaufs, Rohdaten vs. gefilterte Daten mit der Savitzky-Golay-Methode (Fenstergröße 100, Grad 2) und dem gleitendem Mittelwert (Fenstergröße 15)

4.1.2. Datenaufbereitung

In diesem Projekt ist ein Hauptziel, das foulingrelevante Parameterprofil basierend auf Echtzeit-Sequenzmessungen der Parameter vorherzusagen. Die Zeitreihenwerte werden von mehreren Sensoren aufgezeichnet. Die Messwerte werden mit den entsprechenden Zeitstempeln organisiert, wodurch das Problem in eine Zeitreihenvorhersageaufgabe umgewandelt wird. Somit wird der folgende Ansatz verwendet. Die Modelle können so trainiert werden, um die Zeitreihe von Sequenz zu Sequenz zu rekonstruieren, wobei die vorhergesagten Sequenzwerte als Eingabe für die nächste Sequenz verwendet werden.

Dieser Ansatz führt zu einem gewissen akkumulierten Fehler, der jedoch geringer bleibt als bei der Einzel-Schritt-Rekursiven-Methode. Der gewählte Lösungsansatz ist in Abbildung 10 dargestellt und die Ausgabe-Sequenz kann entsprechend formuliert werden.

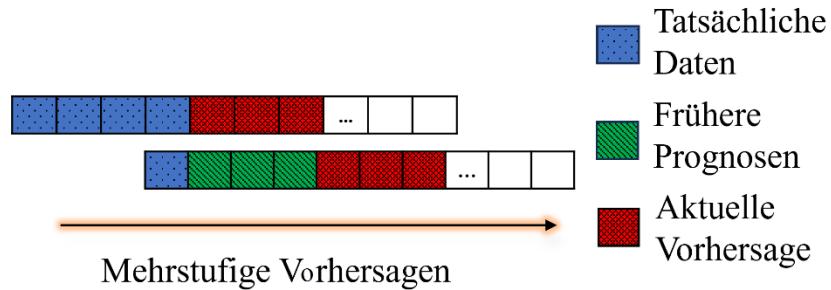


Abbildung 10: Darstellung der mehrstufigen Vorhersagen

$$\{y_t\}_{t=M}^N = F(\{x_t\}_t^M)$$

Die Zeitreihenvorhersage ist eine Form der Multi-Input- und Multi-Output-Regression, bei der Merkmale durch eine Funktion F auf Zielwerte abgebildet werden, wie in der voranstehenden Gleichung dargestellt. x_t stellt die Eingabe des Modells dar, d.h. eine Reihe von Werten, die zu Zeitstempeln von t bis M abgetastet wurden. Dasselbe gilt für y_t , die Ausgabe des Modells, aber von M bis N, also die nachfolgende Serie nach x_t . In diesem Kontext bestehen die Merkmale aus Werten der letzten Zeitstempel, und die Zielwerte sind die Werte der nahen Zukunft. Die gesammelten Daten müssen in drei Datensätze aufgeteilt werden: für das Training, die Validierung und das Testen des Modells.

Um dies zu erreichen, wurde ein Sliding-Window-Algorithmus verwendet. Abbildung 11 veranschaulicht das Konzept dieser Methode, bei der die blauen und grünen Rechtecke mit abgerundeten Ecken ein Fenster fester Größe darstellen. Die Serien im blauen Rechteck repräsentieren Eingabesequenzen, während die im grünen Rechteck für die Ausgabesequenzen stehen. Das Fenster bewegt sich schrittweise von Anfang bis Ende der Zeitreihe. So werden Eingaben und Ausgaben extrahiert. Zur Generalisierung des Modelltrainings und der Validierung wird ein Zufallszahl von 0 bis 2 variiert, um zufällig Trainings-, Validierungs- und Testdatensätze auszuwählen.

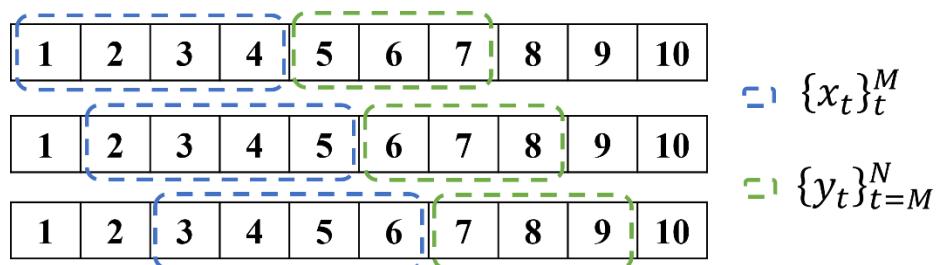


Abbildung 11: Sliding-Window Methode

4.2. AP2: Untersuchung der Eignung von externen Clamp-On Messsystemen

Um einen Überblick zu erhalten, welche Clamp-On-Sensoren kommerziell erhältlich und für welche Prozesse sie geeignet sind, wurde eine umfassende Marktstudie durchgeführt und die Ergebnisse in einer fortlaufend aktualisierten Datenbank gesammelt (siehe Tabelle 4). Das Angebot und die Spezifikationen der Sensoren sind sehr vielseitig und unterscheiden sich teils stark. Generell wurden drei Messgrößen identifiziert, die gut und sinnvoll für die hier thematisierten Produktionsanlagen einsetzbar sind: Volumenstrom, Temperatur und Dichte. Weiterhin ist zu bemerken, dass mit der Komplexität der Messung auch der Preis stark schwankt. Da die Prozessgröße Temperatur für das Foulingverhalten thermisch sensibler Stoffe, wie das hier untersuchte Molkeprotein, sehr entscheidend ist, wurde sich mit Berücksichtigung der Kosten und die durch die Baugröße ergebende Flexibilität bei der Verwendung der Messgeräte für die ersten Messreihen für einen Clamp-On-Temperatursensor (EGT311F031, Sauter-Cumulus GmbH, Niederlande) entschieden.

Die zentrale Frage bei der Verwendung einer solchen Messtechnik ist die Abweichung des Messsignals von Inline-Sensoren, welche direkt mit dem Medium in Kontakt stehen und so eine solche Einschätzung zulassen. Es ist ersichtlich, dass die Platzierung des Clamp-On-Sensors nur zu einer indirekten Erfassung der Mediumstemperatur führt. Um zu erfassen, ob die Auflösung dieser indirekten Erfassung sensitiv genug für einen Foulingversuch ist, wurde, wie in Abbildung 12 dargestellt, ein beispielhafter Fouling- und Reinigungsversuch mit den Clamp-On-Sensoren durchgeführt. Es zeigt sich, dass die Sensoren einen physikalisch sinnvollen und mit den Inline-Sensoren qualitativ vergleichbaren Verlauf wiedergeben. Es kann die graduelle Abnahme des Temperatursignals des Produktausgangs beobachtet werden, welches auf eine zunehmende Verschmutzung des Wärmeübertragers und damit auf eine Verringerung der Wärmeübertragerleistung zurückgeführt werden kann. Ferner ist zu sehen, dass auch bei den schnellen und wiederholten Temperaturwechseln, die während der Reinigung und dem dabei erfolgten Mediumswechsel erfolgen, die Sensoren schnell dem Temperaturwechsel folgen können.

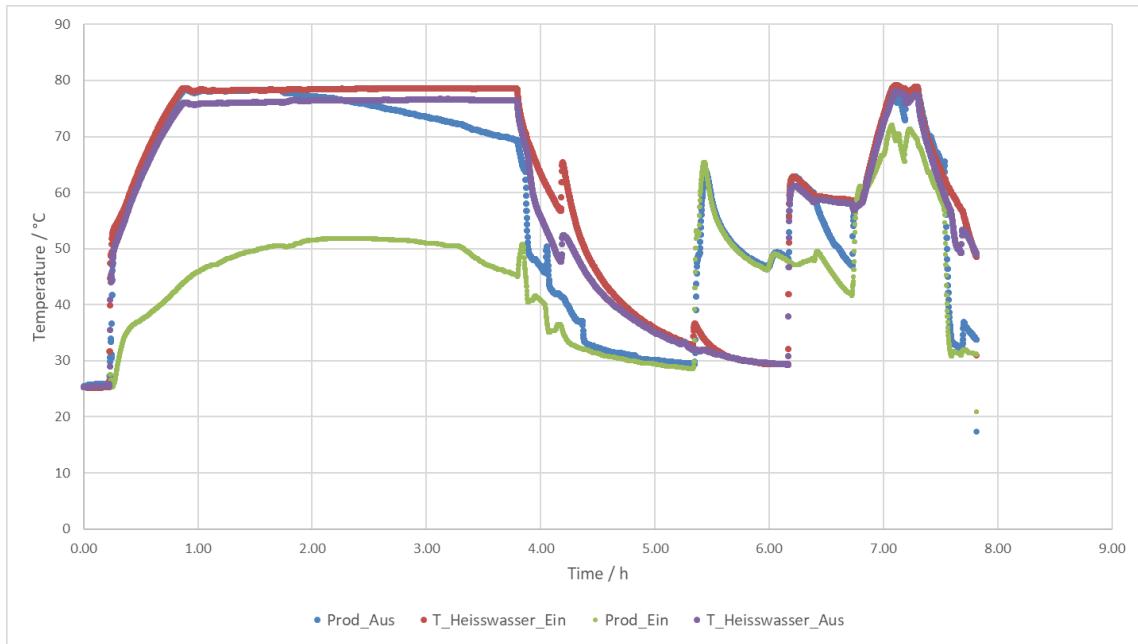


Abbildung 12: Beispielhafter Temperaturverlauf von vier Clamp-On-Sensoren an den vier Ein- und Ausgängen eines Plattenwärmeübertragers während eines Fouling- und eines Reinigungsschritts mit Molkeprotein-Konzentrat.

Selbst wenn, wie oben beschrieben, qualitativ gezeigt werden konnte, dass sowohl die Inline- als auch die Clamp-On-Sensoren den gleichen Signalverlauf anzeigen, ist es dennoch wichtig, die genaue Abweichung zu quantifizieren. Hierzu wurden die Temperaturdifferenzen der beiden Sensortypen an derselben Stelle miteinander verglichen und aufgetragen (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.a**). Es ist ersichtlich, dass der Clamp-on-Sensor die Temperatur des Mediums unterschätzt (negativ abweicht). Dies ist nachvollziehbar, da ein Anlegetemperaturfühler die Rohraußentemperatur misst, welche durch die kühlere Umgebungsluft geringer ist, als die Mediumskerntemperatur.

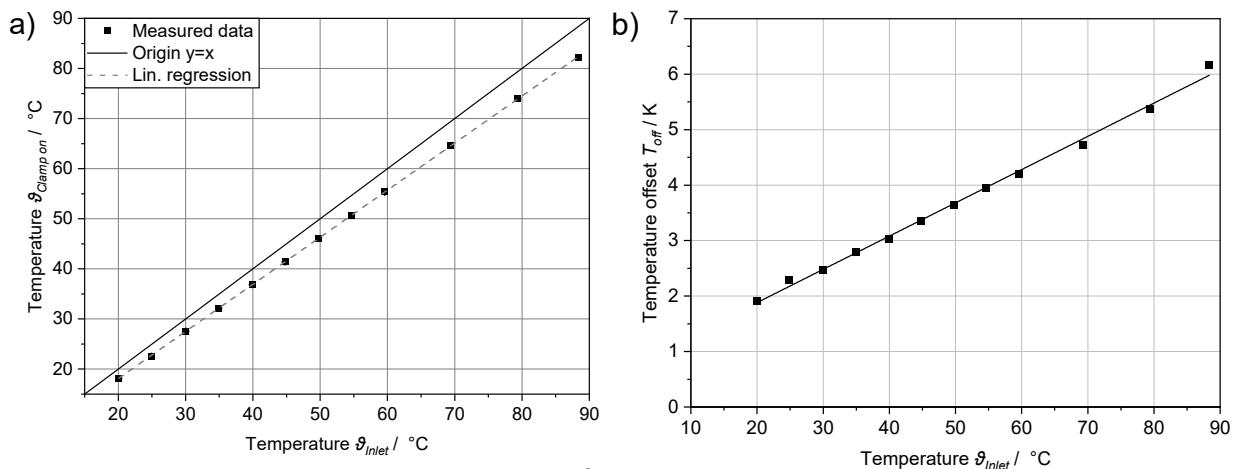


Abbildung 13: Temperaturdifferenz zwischen Inline- und Clamp-On-Sensoren. Die Temperaturdifferenz wurde aus den jeweiligen Temperaturen eines Clamp-on-Sensors und des korrespondierenden Inline-Temperatursensors in unmittelbarer Nähe berechnet (Jarmatz et al., 2024).

Strömungsgeschwindigkeit														
Herstel-ler	Gerät	Messprinzip	Messbereich		Rohrdurchmesser		Prozesstempe-ratur		Umgebungstempe-ratur		Messgenauig-keit	Ein-gänge	Ausgänge	Preis
			min	max	min	max	min	max	min	max				
Krohne	Optisonic 6300 P	Ultraschall	0.5 m/s	20 m/s	DN15	DN400	-25 °C	+120 °C	-40 °C	+60 °C	± 1 % MW	Binär	Strom (Hart), Puls, Status, Modbus	6,576.75 €
Endress+ Hauser	Proline Prosonic Flow 93T	Ultraschall	0 m/s	15 m/s	DN15	>DN200	-40 °C	+80 °C	-40 °C	+60 °C	± 0,5 % mit Werks-Messprotokoll		0/4 ... 20 mA (Hart), Impuls, Profibus DP/PA, Foundation Fieldbus	4,279.70 €
Nivus	NivuFlow	Ultraschall	-10 m/s	+10 m/s	DN50	DN6000	0 °C	+80°C	-30 °C	+80 °C	±0.1 % vom Messwert	4 ... 20 mA	4 ... 20 mA, Modbus TCP, Modbus RTU, Ethernet TCP/IP	5,050.70 €
Siemens	Sitrans FS290	Ultraschall	-12 m/s	+12 m/s	DN20	DN1200	+40°C	+230°C			1% Präzision		4-20 mA, mit Impuls-ausgang, mit Digital-ausgang, Modbus	
Montar	Greyline PDFM 5.1	Ultraschall	0,5 m/s	+12 m/s			-40 °C	+150 °C			±2%		4-20 mA	4,968.00 €
Wess Measure-ment	FLO100	Ultraschall	0	30 m/s			-10°C	70°C			± 1 % MW		4-20mA, or relay, digi-tal, or pulse	
Temperatur														
Labom	GA2610	PT100	-40 °C	+150 °C	4 mm	300 mm	-40 °C	+150 °C	-20 °C	-80 °C	Typisch 0.1 °C Reproduzier-barkeit		4 ... 20 mA (Messum-former PA2430), IO-Link (PA2530)	351.96 €
Sauter	EGT 311/ EGT 411	TP1000/Ni200/Ni1000	-35 °C	+90 °C	10 mm	50 mm	-35 °C	+90 °C	-35 °C	+70 °C	Je nach Typ ±0.3 / 0.4 K		4 ... 20 mA	86.25 €
GMH Mess-technik GmbH	GTL737	PT100	-20 °C	+150 °C	20 mm	89 mm	-20 °C	+160 °C	-20 °C	+60 °C	Unter 2.5 % je nach Kalibrie-rung		4 ... 20 mA	293.20 €
Krohne	OPTI-TEMP TRA-G10	PT100	-50 °C	+200 °C			-50 °C	+200 °C						
Thermal Detection	HSP	PT100/Type T Ther-moelement	-50 °C	+150 °C			-50 °C	+150 °C						
Dichte														
Wess Measure-ment	ENV200C	Ultraschall	1 g/cm³	1,5 g/cm³			-20 °C	+70 °C			± 1 % MW	4 ... 20 mA	4-20 mA, 3 SPDT, RS232C Digital	
EIP En-viro Con-trols	ENV200	Ultraschall					-40 °C	+185 °C					4 – 20 mA/HART/RS-485/Modbus	

Tabelle 4: Ergebnisse der Marktstudie und Auszug aus der Datenbank über kommerziell verfügbare Clamp-On-Sensoren. Preise Stand Juni 2022

Die Darstellung in Fehler! Verweisquelle konnte nicht gefunden werden.a zeigt jedoch auch, dass diese Abweichung über den relevanten Temperaturbereich linear ansteigt und damit durch eine einfache Geradengleichung korrigierbar ist. Fehler! Verweisquelle konnte nicht gefunden werden.b stellt ferner die absolute Abweichung zwischen den beiden Sensoren über die Temperatur des Inline-Sensors dar. Diese Abweichung ist bei der späteren Verwendung dieser Sensoren zu berücksichtigen. Um in einem nächsten Schritt quantifizieren zu können, wie der Temperaturabfall von Foulingschichten beeinflusst wird, wurde eine weitere Versuchsreihe durchgeführt. Ein Rohr wurde so präpariert, dass ein Hohlrohr eingeschoben und ein Clamp-on-Sensor außen an das Rohr angebracht werden kann (siehe Abbildung 14). Ferner befinden sich im Ein- und Auslauf Inline-Temperatursensoren, welche den direkten Vergleich zwischen den Sensoren ermöglichen. Diese Kunststoffrohreinsätze simulieren ungewollte Ablagerungen (Fouling) in diesem Rohrstück, welche direkt über die Clamp-on-Messung erfasst werden.

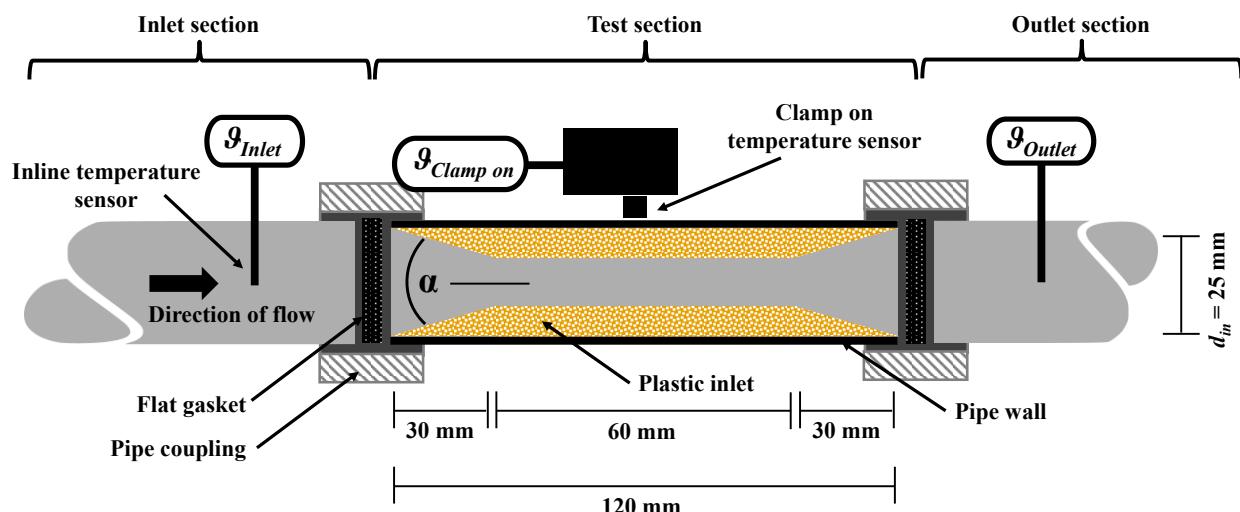


Abbildung 14: Schematische Darstellung des Aufbaus des Messrohres für die Untersuchung von künstlichen Foulingschichten mittels Clamp-on-Sensor in einem Messrohr (Jarmatz et al., 2024).

Abbildung 15 stellt die Ergebnisse der umfangreichen Screening-Versuche mit vier unterschiedlichen Kunststoffen (PVC, PA-6, PE-HD und Macor®) dar. Bei PVC, PA-6 und PE-HD handelt es sich um viel verwendete Massenprodukte, welch eine Wärmeleitfähigkeit im unteren und mittleren Bereich von typischen realen Foulingschichten aufweisen. Macor® ist ein Hochleistungskunststoff, welcher eine vergleichsweise hohe Wärmeleitfähigkeit aufweist und dadurch den oberen Bereich von typischen Foulingschichten abdeckt. Durch die zeitgleiche Variation der Schichtdicke ($\delta = 2 \text{ mm}, 4 \text{ mm}, 6 \text{ mm}$) konnte ein breiter Bereich bezüglich des resultierenden Foulingwiderstands abgebildet werden. Abbildung 15a zeigt die resultierenden Temperaturdifferenz in Abhängigkeit der Schichtdicke und der verschiedenen Kunststoffe. Es ist ersichtlich, dass die Kunststoffe entsprechend ihrer Wärmeleitfähigkeit angeordnet werden können und eine Erhöhung der Schichtdicke zu einer linearen Vergrößerung des Temperaturabfalls führt. Wird der Temperaturabfall über die berechneten Foulingwiderstände für alle Kunststoffe und Schichtdicken aufgetragen, ergibt sich in Übereinstimmung mit der Literatur ebenfalls ein linearer Zusammenhang, welches ein hohes nutzbares Potential für die untersuchten Clamp-on-Sensoren für die direkte Foulingerfassung durch Temperaturmessungen aufzeigt.

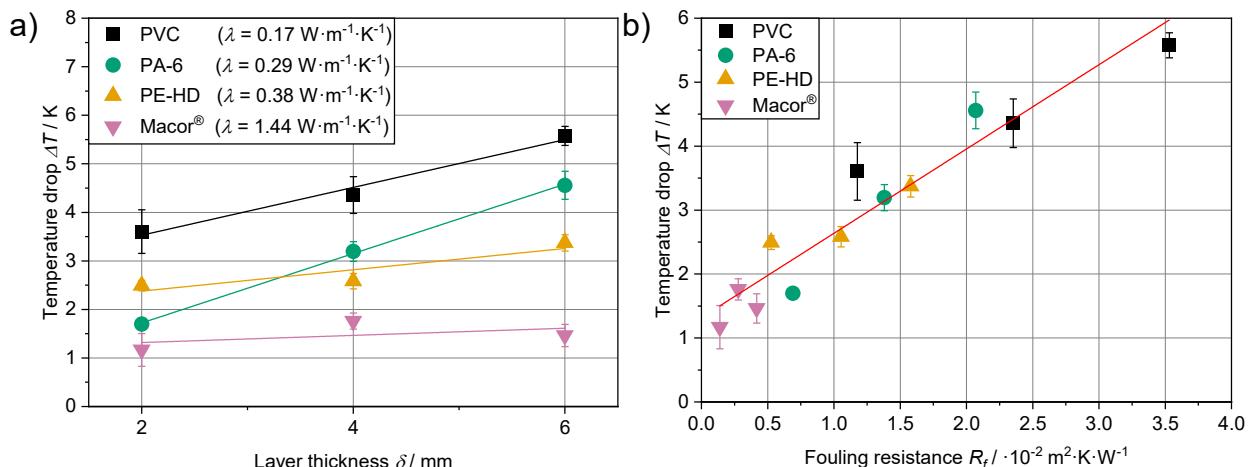


Abbildung 15: Erfasste Temperaturdifferenz zwischen dem Clamp-on- und Inline-Sensor aufgetragen über a) der Schichtdicke und b) dem berechneten Foulingwiderstand für vier ausgewählte Kunststoffe (Jarmatz et al., 2024).

Um nun den Einsatz der Clamp-On-Sensoren in einer industriellen Produktionsanlage gewährleisten zu können, wurde auch hier ein Fragenkatalog (siehe Tabelle 5) für die Abfrage relevanter Informationen erstellt. Dies beinhaltet vor allem Aspekte der Netzwerk- und Stromversorgung sowie der Sicherheit vor Ort.

Tabelle 5: Fragenkatalog für die Vorbereitung des Einsatzes von externen Clamp-On-Sensoren in Produktionsanlagen.

Kategorie	Frage
Netzwerk	<ul style="list-style-type: none"> - Kann eine Datenübertragung online stattfinden (Ethernet, WiFi) oder muss ein lokaler Speicher genutzt werden?
Stromversorgung	<ul style="list-style-type: none"> - Kann für den Einplatinencomputer ein PoE-Adapter oder 230 V Netzanschluss zur Verfügung gestellt werden? - Kann für das Netzteil der Clamp-On-Sensoren ein 230 V Netzanschluss zur Verfügung gestellt werden? - Wie weit ist die Stromversorgung entfernt (Kabellänge)?
Sicherheit	<ul style="list-style-type: none"> - Welchen Schutz (IP) müssen die Geräte bezüglich Spritzwasserschutz aufweisen? - Besteht Ex-Schutz? - Gibt es sonstige Vorgaben?
Anlagendaten	<ul style="list-style-type: none"> - Welcher Rohrdurchmesser liegt an der Messstelle vor? - Liegen Inline-Sensoren für den gleichen Prozessparameter in der Nähe vor, um das Signal vergleichen zu können?
Zusatzinformationen	<ul style="list-style-type: none"> - Darf Wärmeleitpaste zwischen Sensorkopf und Rohr aufgebracht werden? - Welche Personen sind direkte Ansprechpartner an der Produktionsanlage?

Mit Abschluss dieses Arbeitspaketes ist eine Marktübersicht erstellt, stellvertretend ein geeigneter Sensor ausgewählt und in Laborversuchen überprüft sowie in Hinblick auf seine Eignung getestet worden. Mit der erfolgten Übergabe des Fragenkatalogs an die Industriepartner können zeitnah die Details für einen Einsatz der Sensoren im industriellen Umfeld geklärt werden.

4.3. AP3: Aufbau eines dezentralen Messsystems zur Datenerfassung und Foulingdetektion

4.3.1. Hardwareentwicklung

4.3.1.1. Hardware für Laborversuche

Im Rahmen des Projekts ist geplant, mehrere Raspberry Pi Einplatinenrechner als Messsystem zu verwenden. Durch seine General Purpose Input/Output-Ports (GPIOs) ist es möglich mit anderen Peripheriegeräten wie Sensoren zu kommunizieren. Das integrierte WLAN-Modul und die LAN-Schnittstelle ermöglichen den Datentransfer über das Internet, sodass sich ein Internet of Things (IoT) System entwickeln lässt.

Der Clamp-On Sensor EGT311F03, der in AP2 ausgewählt wurde, liefert ausschließlich ein analoges Ausgangssignal. Da der Raspberry Pi 4B nur über digitale Ein- und Ausgänge verfügt, wurde ein Erweiterungsboard für die Messung entworfen. Zentrales Element ist dabei ein AD-Wandler (ADC), der die Signale über I²C an den Raspberry Pi liefert. Um eine Industrietauglichkeit und Robustheit sicherzustellen, wurden passende Schutzschaltungen, Isolatoren und Spannungswandler von den industrietypischen 0..10 V auf 0..1 V für den ADC entworfen. Das Schema ist in Abbildung 16 dargestellt. Das ganze Modul wird als PiO-Modul bezeichnet und in

Abbildung 17 dargestellt.

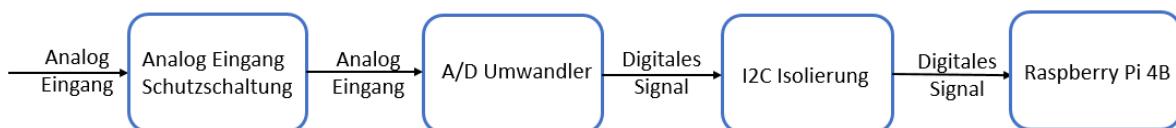


Abbildung 16: Analog-Digital-Umwandler

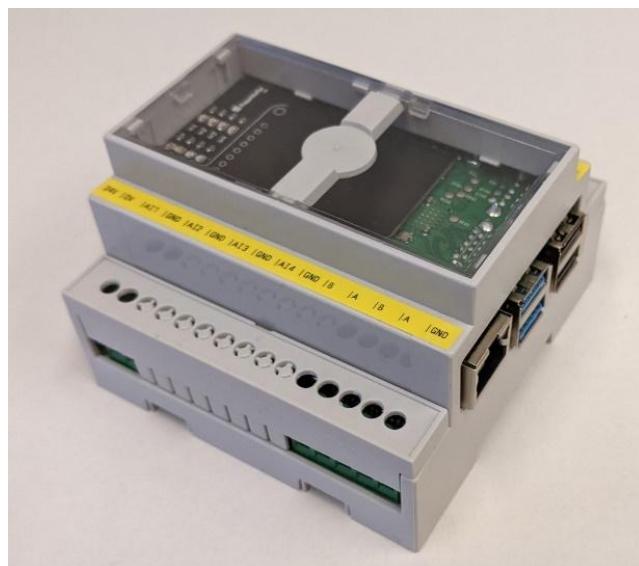


Abbildung 17: Raspberry Pi 4 mit PiO-Modul

4.3.1.2. Hardware für echte Produktion in Molkereiindustrie

Für den Einsatz in der Molkerei wurde eine Hardwarelösung entwickelt, die speziell an die besonderen Umgebungsbedingungen angepasst ist. In der Produktionshalle herrschen hohe Luftfeuchtigkeit und eine erhöhte Temperatur, weshalb es erforderlich ist, dass die Hardware zuverlässig gegen Wasser geschützt ist. Zu diesem Zweck wird die gesamte Elektronik in einer wasserdichten Kiste untergebracht, die zusätzlich eine Toolbox zur Ablage von Kabeln und Werkzeug enthält. Diese Einheit wird auf einem leicht transportierbaren Sackkarren montiert, sodass das System leicht transportierbar ist. Zum Signalempfang ist auf der oberen Seite der Kiste eine Antenne installiert, die in Kombination mit einer SIM-Karte den Zugang zum 4G LTE-Netz ermöglicht. Diese Lösung wurde gewählt, da in vielen Produktionshallen kein zuverlässiger WLAN-Empfang verfügbar ist und die Verlegung eines LAN-Kabels aufwendig und kostenintensiv für die Firmen wäre. Ein LED auf der Oberseite zeigt den aktuellen Systemstatus an.

Im Schaltschrank sind folgende Komponenten verbaut:

- **Industrie-PC (IPC)**: Dieser fungiert als zentrale Recheneinheit des Systems, auf dem alle Softwarekomponenten laufen.
- **LTE-Router**: Ermöglicht dem Industrie-PC (IPC) den Internetzugang über das Mobilfunknetz.
- **IO-Link Master**: Der IO-Link Master (Murrelektronik GmbH, Deutschland) dient als Schnittstelle für die angeschlossenen Sensoren.

An den IO-Link Master ist ein IO-Link Hub angeschlossen, der die Verbindung zu den vier Clamp-On-Sensoren ermöglicht. Diese Sensoren messen die Temperatur in der Produktionshalle und liefern wichtige Prozessdaten für die Überwachung und Steuerung.

In der Abbildung 18 wird das System visuell dargestellt und zeigt die wesentlichen Komponenten sowie die Transportierbarkeit der gesamten Einheit.



Abbildung 18: Darstellung des Messsystems für die Produktion in Molkereiindustrie

4.3.2. Konfiguration des Systems

Zur automatisierten Konfiguration der Systeme wird Ansible eingesetzt. In Ansible sind dabei verschiedene Konfigurationen („Roles“), bspw. für die Container-Umgebung Docker oder das VPN zur Fernwartung hinterlegt und wurden entsprechend für die Raspberry Pis und das aktuelle Projekt angepasst. Das Ansible Tool dient ebenfalls für die IPC in der Messbox.

4.3.3. Softwareentwicklung

Die lokale Software zur Datenerfassung und Archivierung basiert auf dem FastIoT-Framework (Klaeger and Merker, 2022), welches am Fraunhofer IVV entwickelt wurde. Das Framework ermöglicht die Entwicklung von Microservice-Anwendungen, bei denen jeder Service nur eine bestimmte Aufgabe übernimmt. Die einzelnen Services liegen jeweils in einem Docker-Container und kommunizieren über einen Message Broker miteinander.

Auf dieser Basis können die Aufgaben in diesem Projekt voneinander getrennt werden, sodass die Software aus verschiedenen Services besteht:

- **analog_sensor_reader** erfasst die Daten des AD-Wandlers mithilfe einer Python-Bibliothek. Über eine Konfigurationsdatei können die Abtastfrequenz und auszusende Kanäle definiert werden. Die erfassten Daten werden durch ein Tiefpassfilter gefiltert, um Rauschen zu reduzieren und als Objekt vom Typ „Thing“ auf den Message Broker gesendet.
- **object_storage** abonniert Nachrichten vom Typ "Thing" und speichert sie in einer MongoDB.
- **dash** ist eine WebUI zur Anzeige von historischen und Live-Daten, erneut vom Typ „Thing“
- **object_storage_backup** stellt die Verbindung zu einer weiteren Datenbank auf einem extra eingerichteten Backup-Server her und kopiert dort in regelmäßigen Abständen die erfassten Daten um einen Verlust, bspw. bei Defekt der SD-Karte im Raspberry Pi, zu vermeiden.
- **MongoDB** ist eine weit verbreitete dokumentenbasierte NoSQL-Datenbank und wird ebenfalls als Container auf dem Raspberry Pi sowie einem Backup-System gestartet.
- **io_link_reader**: Zur Datenerfassung aus dem IO-Link-Master wird ein spezieller Service entwickelt. Dieser Service ruft die Daten über die vom IO-Link bereitgestellte REST-Schnittstelle ab und stellt sie über den NATS-Broker zur Verfügung.
- **live_displayer** empfängt die Daten, die vom Datenerfassung Service gesendet werden. Auf Grundlage der empfangenen Daten wird eine Echtzeitprognose durchgeführt, die auf dem präzisesten trainierten Modell basiert. Die Prognose wird auf einem Web-UI dargestellt. Die empfangenen und vorhergesagten Daten werden durch eine vertikale rote Linie getrennt, welche den aktuellen Zeitstempel darstellt.

Abbildung 19 zeigt beispielhaft, wie der Service **dash** Echtzeitdaten anzeigen kann.

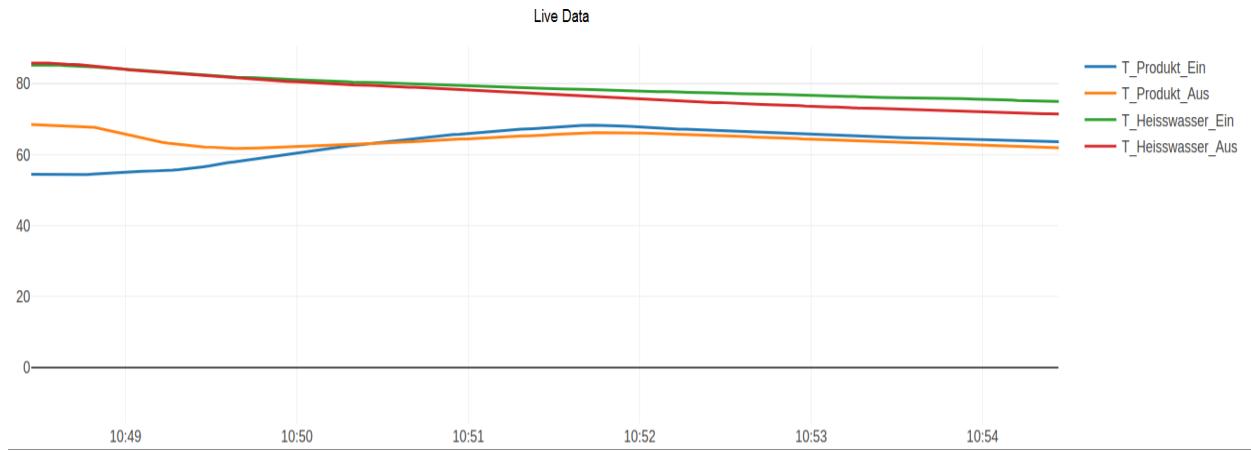


Abbildung 19: Ausschnitt aus dem Dashboard des Services "Dash".

An der Versuchsanlage des ICTV ist solch ein Raspberry Pi als Messgerät mit vier Clamp-On-Temperatursensoren installiert. Der Service `analog_sensor_reader` übermittelt über den Messagebroker Messdaten an `object_storage`, woraufhin diese in die lokale MongoDB gespeichert werden. Alle 30 min wird über den Service `object_storage_backup` eine Sicherungskopie auf den IVV FideLio-Server gesendet. Außerdem ist ein Cronjob-Service für das Datenbackup aktiviert, damit die Daten jede Nacht in der MongoDB gesichert und auf dem IVV FideLio-Server abgelegt werden.

4.4. AP4: Schaffung eines Gesamtsensorkonzeptes für neue Anlagen

Um nicht nur mittels der an einer bestimmten Produktionsanlage generierten Trainingsdaten eine besonders genaue Vorhersage für ebendiese Anlage oder einen besonders für Fouling anfälligen Apparat machen zu können, sondern mit geringem Aufwand auch Empfehlungen für die Neukonstruktion einer vergleichbaren Anlage geben zu können, soll ein Gesamtsensorkonzept entwickelt werden. Die Empfehlungen des generierten Konzeptes sollen dann in dieser neuen Anlage die Foulingvorhersage noch weiter verbessern, da für die Vorhersage relevante Sensoren direkt beim Bau in optimaler Anzahl und an der dafür passenden Stelle montiert werden können.

Damit das Gesamtsensorkonzept in einem ersten Schritt mit Informationen bezüglich der Messposition gefüttert werden kann, wurden umfangreiche systematische Untersuchungen über das Foulingverhalten von kritischen Anlagenbauteilen durchgeführt. Bei solchen Bauteilen handelt es sich z. B. um Messstutzen, Bögen, Ventile, Diffusoren oder Düsen. Diese sind jedoch im Gegensatz zu Wärmeübertragern wenig anfällig für thermisch induziertes Fouling, sondern aufgrund des Einflusses auf das Strömungsverhaltens des Fluids eher von Sedimentationsfouling betroffen. Daher wurde für die Untersuchung ein Modellstoffsystem aus Quarzglaspartikeln genutzt, mit welchen eine sehr reproduzierbare und vergleichende Untersuchung der Bauteile möglich ist. Abbildung 20 zeigt exemplarisch die Ergebnisse der Variation der Position im Raum für die Bauteile a) Messstutzen und b) Bogen und die Auswirkung auf die abgelagerte Partikelmasse pro Fläche. Es ist ersichtlich, dass die Orientierung im Raum für ein Bauteil mit Totzonen (a) eine erhebliche Rolle spielt, während dies für ein Bauteil ohne Totzonen (b) weniger ausschlaggebend ist.

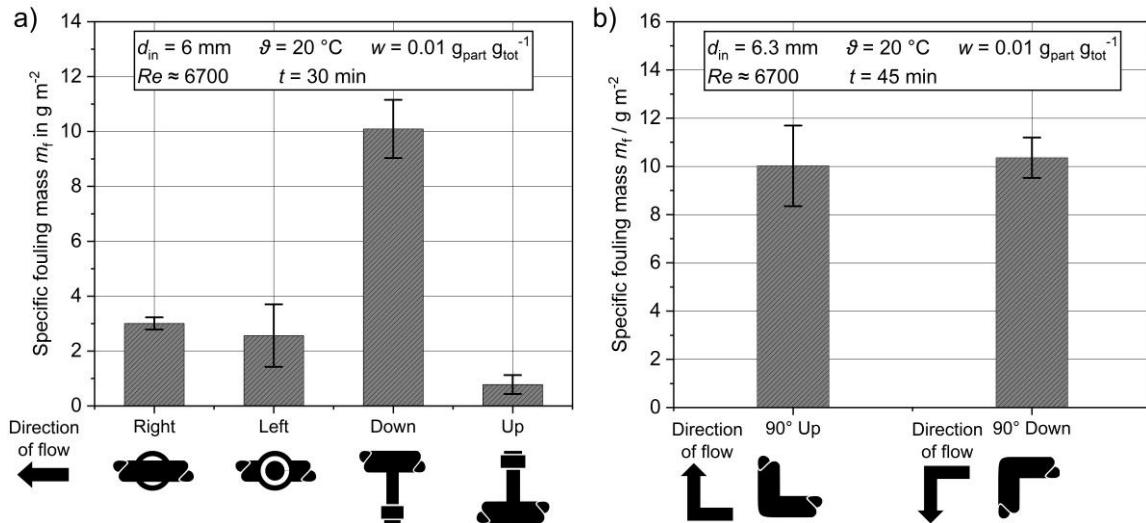


Abbildung 20: Systematische Variation der Orientierung a) eines Messstutzens und b) eines Bogens für die Bestimmung der spezifischen Foulingmasse (Jarmatz et al., 2023).

Werden bei gleichen Parametern die verschiedenen Bauteile untereinander verglichen, ergibt sich auch hier ein sehr heterogenes Bild (siehe Abbildung 21). Hier sticht vor allem wie erwartet das Ventil heraus, welches mit viel Totraum bei gleichzeitig kleinerer innerer Oberfläche eine hohe flächenspezifische Ablagerung an Partikeln aufweist.

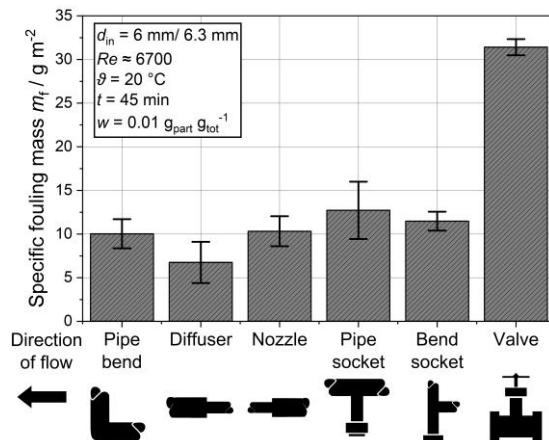


Abbildung 21: Direkter Vergleich diverser Rohrfittings in Bezug auf die abgelagerte Foulingmasse (Jarmatz et al., 2023).

Für eine spätere konkrete Anwendung muss untersucht werden, welche Prozessparameter für eine Foulingvorhersage auf Basis von historischen Daten besonders relevant sind („Was muss gemessen werden?“). Zusätzlich ist das Augenmerk auf den zeitlichen Verlauf der Messungen bei den Foulingvorgängen zu legen. Hiervon kann dann die notwendige Messauflösung des Sensors und der Umfang des nötigen Trainingsdatensatzes abgeleitet werden („Wann und wie oft muss gemessen werden?“).

4.5. AP5: Aufbau von Modellen und Methoden zur Datenaufbereitung

Um für das Anlernen des Modells geeignete Trainingsdaten zur Verfügung stellen zu können, wurden im Rahmen dieses APs umfangreiche Screening-Versuche an einer Plattenwärmeübertrager-Foulinganlage im Technikumsmaßstab durchgeführt. Um die

Modelle möglichst effizient und zielführend anzutrainieren, wurde eine breite Parametervariation für Molkeprotein-Konzentrat als Stoffsystem durchgeführt. Dazu wurde der thermische Foulingwiderstand als Parameter zur Charakterisierung des Wärmeübergangs, und der Druckverlust für den Einfluss des Foulings auf die Fluidodynamik berechnet. Abbildung 22 zeigt exemplarisch an zwei Beispielversuchen die zeitlichen Verläufe der beiden berechneten Größen. Dies zeigt, dass die Verläufe sehr unterschiedlich sein können und damit eine gute Grundlage für das Modelltraining bieten. Die Durchführung der Screening-Foulingversuche sind damit abgeschlossen und in einem nächsten Schritt wurden die generierten Trainingsdaten aufbereitet, um sie für das Modelltraining zu nutzen.

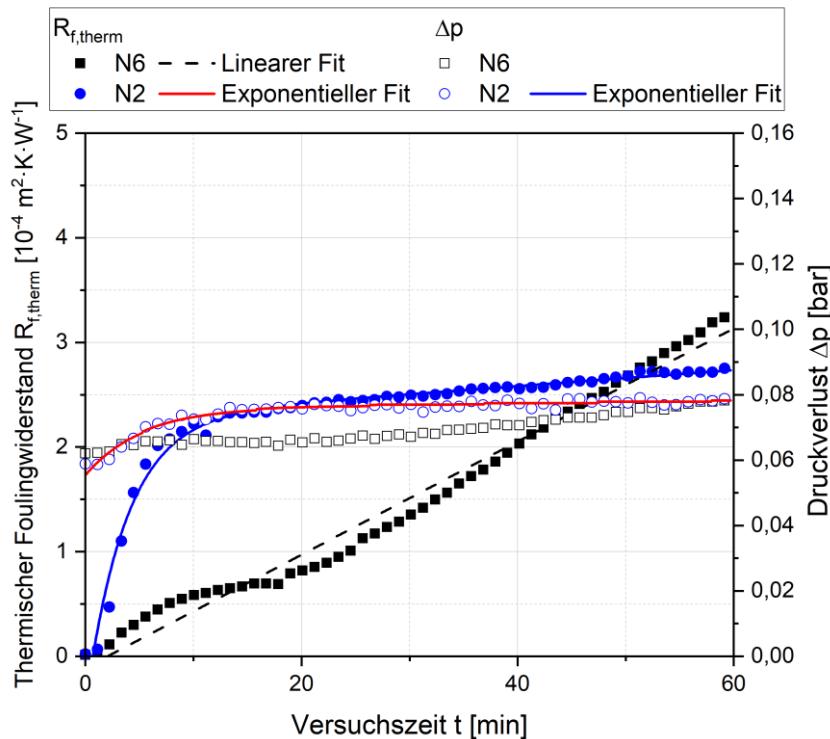


Abbildung 22: Exemplarische Darstellung des berechneten thermischen Foulingwiderstands und des resultierenden Druckverlusts über die Versuchszeit für zwei Versuchsdurchläufe (Sidorow, 2023).

Relevant für den Aufbau von Machine Learning Modellen sind die sog. Features. Aus dem Prozess lassen sich diese als Kennwerte extrahieren, wobei in Anlehnung an dimensionslose Kenngrößen eine möglichst große Unabhängigkeit von bspw. der Bauteilgeometrie angestrebt wird. Mit diesen Merkmalen können dann die Modelle aufgebaut werden, um anschließend eine Zeitreihenvorhersage durchzuführen. Hierzu wurden bereits erste Recherchen und Vergleiche durchgeführt. Vielversprechend für das aktuelle Vorhaben erschienen dabei v. a. Vector Autoregression (Akkaya, 2021), Bayesian Structural Time Series (Qiu et al., 2018) oder Long Short-Term Memory Netzwerke (LSTM) (Hochreiter and Schmidhuber, 1997) zu sein.

4.5.1. Evaluationsmetrics

Bevor die Modelle zur Foulingvorhersage aufgebaut werden, müssen zuerst die Evaluationsmetrics definiert werden. Da die Zeitreihenvorhersage mit der Regression übereinstimmt, wird die konventionelle Metrik, der Mean Squared Error (MSE), während des Trainings und der Validierung verwendet:

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

Während des Modelltests wird der Mean Absolute Percentage Error (MAPE) eingesetzt:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%$$

In beiden Gleichungen steht y_i für die tatsächlichen Werte aus dem Validierungsdatensatz, und \hat{y}_i entspricht den vorhergesagten Werten. n bezeichnet die Gesamtanzahl der Zeitreihen. In dieser Studie wird die Metrik 1 - MAPE als Leistungsindikator angewendet, um die Modellleistung besser darzustellen. Je größer der resultierende Wert, desto zuverlässiger wird die Modellleistung eingeschätzt. In diesem Bericht bezeichnen wir diesen 1 – MAPE-Verlust als resultierenden MAPE.

4.5.2. Skalierungsmethode

Da die Daten aus drei verschiedenen Quellen stammen und die Sensorwerte unterschiedliche Verteilungen aufweisen, müssen sie vor dem Modelltraining skaliert werden. Hierfür wird die MinMax-Skalierungsmethode verwendet. Durch diese Skalierung werden die Daten normalisiert und für das Modelltraining weiterverwendet.

$$X_{scale} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

4.5.3. Neuronale Netzwerkarchitektur

In diesem Projekt konzentrieren wir uns auf rekurrente neuronale Netze (RNN). Im Gegensatz zu herkömmlichen Feed-Forward-Netzen (Sharma et al., 2013), bei denen der Aktivierungsfluss nur in eine Richtung verläuft, ermöglichen RNNs, dass der Output des letzten Schritts als Input für den nächsten Schritt verwendet wird und somit die folgenden Schritte beeinflusst (Aurelien, 2018). In diesem Projekt wird das Long Short-Term Memory (LSTM), eine moderne RNN-Variante, die sich bei der Verarbeitung von sequenziellen Daten als effektiv erwiesen hat, eingesetzt. Die Einführung von LSTM hat die Herausforderungen im Zusammenhang mit dem Gradientenverlust und -explosion (Hochreiter and Schmidhuber, 1997) gelöst. Dadurch wird es dem Modell ermöglicht, auch längere Kontexte zu verarbeiten, was eine Einschränkung herkömmlicher Modelle adressiert.

4.5.3.1. Long Short-Term Memory (LSTM)

Wie in Abbildung 23 gezeigt, besteht die Long Short-Term Memory (LSTM)-Architektur aus den folgenden grundlegenden Komponenten: der Forget Gate F_t , der Input Gate I_t , dem Input Node C_t , und der Output Gate O_t . Ihre Hauptfunktion besteht darin, vorangegangene Daten kurzzeitig im 'Kurzzeitgedächtnis' zu speichern. Zunächst werden der anfängliche verdeckte Zustand H_{t-1} und der Input x_t konkateniert und durch die Forget Gate F_t verarbeitet, die eine Sigmoidfunktion gemäß unten stehender Gleichung verwendet, deren Werte im Bereich $[0, 1]$ liegen. Der Ausgang der Schleuse wird dann mit dem vorherigen internen Zustand \tilde{C}_{t-1} , multipliziert, um zu bestimmen, ob der interne Zustand berücksichtigt werden soll.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

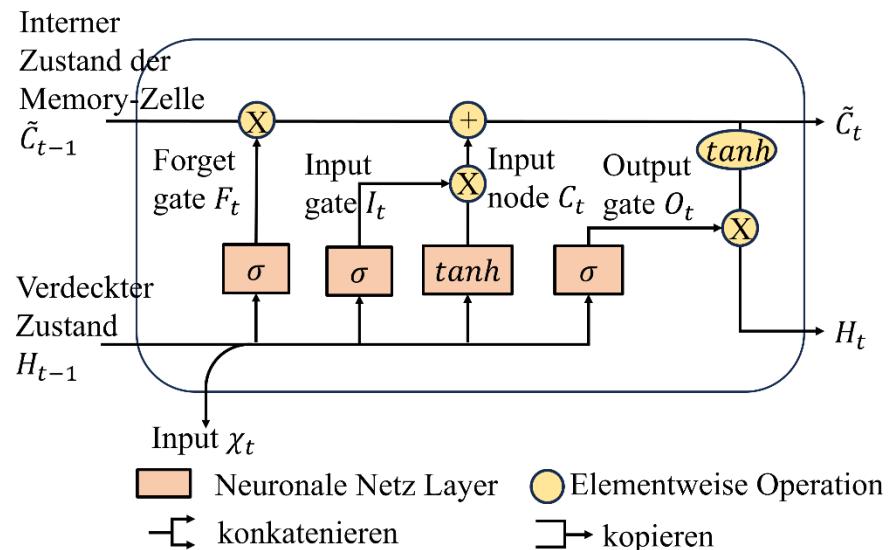


Abbildung 23: Long Short-Term Memory Zelle Architektur ähnlich (Zhang et al., 2023)

Die folgenden Komponenten sind die Input Gate I_t und der Input Node C_t . Die tanh -Funktion erzeugt einen neuen Kandidaten für den internen Zustand \tilde{C}_t , wobei I_t entscheidet, ob dieser neue Zustand beibehalten wird. Nach der Erstellung wird er mit dem vorherigen internen Zustand \tilde{C}_{t-1} kombiniert. Der endgültige interne Zustand \tilde{C}_t ist auf den Bereich $[-1, 1]$ beschränkt und wird mit den Werten der Output Gate O_t multipliziert. Bemerkenswert ist, dass H_t nicht nur als Zustand zwischen den Schritten dient, sondern auch als Ausgang dieser Schicht fungiert. Diese LSTM-Zelle wird im Pytorch-Framework (Paszke et al., 2019) implementiert, das in diesem Projekt hauptsächlich verwendet wurde.

4.5.3.2. Encoder-Decoder Long Short-Term Memory (Encoder-Decoder LSTM)

Die in Abbildung 24 dargestellte Encoder-Decoder LSTM Architektur stellt eine erweiterte Variante des traditionellen LSTM dar. Das Encoder-Decoder-Modell ist ein umfassendes

Framework für die Sequenz-zu-Sequenz (seq2seq) -Vorhersage, das in (Sutskever et al., 2014) eingeführt wurde. Die Komponenten, die zum Aufbau des Encoders und Decoders verwendet werden, können verschiedene Typen von rekurrenten neuronalen Netzen (RNNs) umfassen.

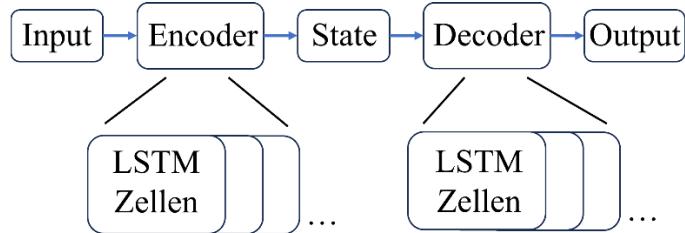


Abbildung 24: Encoder-Decoder LSTM Architektur

Die Einschränkung besteht darin, dass LSTM nur den letzten versteckten Zustand H_t an die nächste Schicht übergibt. Im Gegensatz dazu erfasst der Encoder, eine Folge von LSTM-Zellen, die gesamte Sequenz von Informationen über seine Schichten hinweg und ermöglicht die umfassende Übertragung von Wissen an den Decoder, der ebenfalls aus einer Reihe von LSTM-Zellen besteht. Diese intrinsische Eigenschaft verbessert theoretisch die Modellleistung, da sie eine effektivere Darstellung sequenzieller Daten ermöglicht und die gesamte Vorhersagefähigkeit der Encoder-Decoder LSTM Architektur verbessert.

4.5.3.3. Convolutional Neuronales Netzwerk mit Long Short-Term Memory (CNN-LSTM)

Convolutional Neural Networks (CNNs) stellen eine weit verbreitete Technologie des tiefen Lernens dar (Zhao et al., 2019), (Xiong et al., 2021). Eine typische CNN-Architektur besteht aus konvolutionalen Schichten, Pooling-Schichten und vollständig verbundenen Schichten (O’Shea and Nash, 2015). Die konvolutionale Schicht verwendet konvolutionale Operationen, die spezifische Masken (Bose and Meyer, 2003) einsetzen, um Merkmale aus den Eingabedaten zu extrahieren, was Aufgaben wie die Kantenerkennung in Signalen ermöglicht. Pooling-Schichten spielen eine entscheidende Rolle bei der Reduzierung der räumlichen Größe des Outputs der vorhergehenden Schicht, wodurch die Berechnungen und benötigten Gewichte verringert werden. In dieser Arbeit wird die Max-Pooling-Schicht ausgewählt, um den maximalen Wert innerhalb eines Kerns zu berechnen. Darauf folgt die vollständig verbundene Schicht, die die extrahierten Merkmale abbildet, um das endgültige Ergebnis zu erzeugen. Obwohl CNNs traditionell mit der Bildverarbeitung assoziiert werden, haben mehrere Forscher ihre Anpassungsfähigkeit an eindimensionale Zeitreihendaten für Vorhersageaufgaben untersucht (Wibawa et al., 2022), (Yang et al., 2017). Allerdings sind CNNs nicht normalerweise so angepasst, dass sie komplexe und lange Abhängigkeiten korrekt verwalten (Bengio et al., 2013).

Angesichts der intrinsischen Eigenschaft von Long Short-Term Memory (LSTM)-Netzwerken, Zeitreihendaten zu verarbeiten, stellt eine Kombination aus CNNs und LSTMs, wie in Abbildung 25 gezeigt, einen vielversprechenden Ansatz zur Leistungssteigerung

dar (Livieris et al., 2020), (Lu et al., 2020). Mithilfe des CNN-Netzwerks werden die Zeitreihendaten durch eine Maske gefiltert, bevor sie an das LSTM übergeben werden. Diese Kombination nutzt die Stärken beider Architekturen und verbessert die Fähigkeit, lokale Muster in Zeitreihendaten zu erkennen, und steigert folglich die Vorhersageergebnisse.

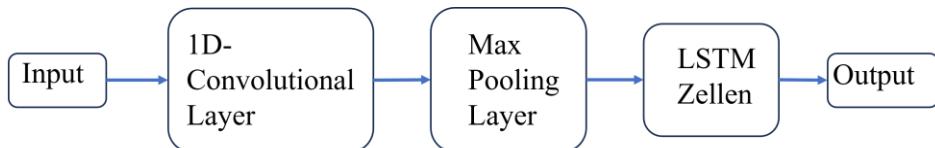


Abbildung 25: CNN-LSTM Architektur

4.5.4. Wichtige Einflussfaktoren bei der Modellbildung

4.5.4.1. Kombination von Eingangsparametern

Da die Datensätze meistens von verschiedenen Sensoren entstammen, muss ein Auswahlprozess der Eingangsparameter für das Modelltraining durchgeführt werden. Deswegen werden in folgenden Abschnitten die Vorhersagegenauigkeiten bei unterschiedlichen Parameterkombinationen miteinander verglichen.

4.5.4.2. Hyperparameter Tuning

Das Tuning der Hyperparameter spielt eine entscheidende Rolle beim Modelltraining und trägt zu einer verbesserten Leistung bei. Während die Eingabe- und Ausgabelängen flexibel sind, darf die Summe die minimale Zeitdauer von Produktion/Experiment nicht überschreiten. In diesem Projekt wurden drei Modelle zur Zeitreihenvorhersage entwickelt, die alle auf dem LSTM-Modell basieren. Daher teilen sie gemeinsame Hyperparameter, die für LSTM relevant sind. Die Anzahl der versteckten Schichten und die Anzahl der Neuronen in jeder Schicht wurden für die drei Modelle variiert. CNN hat außerdem seine eigenen Hyperparameter, nämlich die Größe der Maske. Für die Sequence to Sequence-Architektur wird eine spezifische Trainingsstrategie namens Teacher-Forcing angewendet. Teacher-Forcing zielt darauf ab, die Trainingseffizienz und Stabilität zu verbessern, indem dem Decoder während des Trainings mit einer bestimmten Wahrscheinlichkeit der korrekte oder tatsächliche Output des vorherigen Zeitschritts bereitgestellt wird. Dieser Ansatz verhindert, dass das Modell während des Trainings Fehler anhäuft, was zu einer schnelleren Konvergenz und mehr Stabilität führt (Williams and Zipser, 1989). Um Überanpassung während des Modelltrainings zu verringern, wurde auch eine Dropout-Schicht in jedes Modell integriert.

4.5.5. Klassische Regressionsmodelle

In diesem Projekt wird das foulingbezogene Parameterprofil mithilfe eines Seq2Seq-Modells vorhergesagt. Dieses Profil wird anschließend verwendet, um die Foulingmasse vorherzusagen. Mit nur wenig verfügbaren Proben für das Modelltraining und die -testung

bei der Schätzung der Foulingmasse wird es als angemessen erachtet, kleinere Modelle mit einer begrenzten Anzahl von Parametern auf diesen kleinen Datensätzen zu verwenden (Ahmad et al., 2018) (Mahamat et al., 2024). Um dieses Problem zu überwinden und die Generalisierung und Zuverlässigkeit zu verbessern, werden Algorithmen wie Support Vector Regression (SVR), lineare Regression (LR) und ein Ensemble-Modell, der Random Forest (RF) (Tibshirani et al., n.d.) betrachtet.

SVR ist eine weit verbreitete statistische Lernmethode, die dazu dient, Beziehungen zwischen Eingangs- und Ausgangsvariablen herzustellen. Ihr Ziel ist es, Vorhersagefehler zu minimieren und gleichzeitig die Komplexität der Regressionsfunktion zu steuern (Tibshirani et al., n.d.).

LR, ein weiteres Werkzeug zur Etablierung von Beziehungen zwischen Eingangs- und Ausgangsvariablen, zielt darauf ab, Koeffizienten für eine lineare Gleichung zu bestimmen, während die Summe der quadrierten Unterschiede zwischen vorhergesagten und tatsächlichen Werten minimiert wird, wodurch eine unkomplizierte Methode für Vorhersagen bereitgestellt wird (Tibshirani et al., n.d.).

RF konstruiert ein robustes prädiktives Modell, indem während des Trainings mehrere Entscheidungsbäume erzeugt und die durchschnittliche Vorhersage der einzelnen Bäume aggregiert wird. Basierend auf Entscheidungsbäumen, die Daten rekursiv in Untergruppen gemäß den bedeutendsten Attributen an jedem Knoten aufteilen, wird eine baumartige Struktur gebildet, bei der die Blätter die vorhergesagten Ergebnisse anzeigen. Ein Random Forest, der aus zahlreichen Entscheidungsbäumen besteht, dient dazu, Überanpassung zu mildern und die Generalisierung zu verbessern (Tibshirani et al., n.d.).

4.5.6. Beschreibung Modelltraining, -validierung und -testprozess

Der Datensatz wird nach der Methode, die im Abschnitt 4.1.2 beschrieben wurde, für das Modelltraining vorbereitet. Die Modelle LSTM, Encoder-Decoder LSTM und CNN-LSTM, die in oberen Abschnitten vorgestellten Architekturen, wurden an die Aufgabenstellung der Zeitreihenvorhersage angepasst und hinsichtlich ihrer Eignung zur Foulingmasseerkennung verglichen. Die Ein- und Ausgangsparameter für das Modelltraining werden im jeweiligen Abschnitt beschrieben werden. Der Implementierungscode wurde in Python 3.10 und PyTorch (Paszke et al., 2019) auf einem Rechner mit Nvidia H100 GPU geschrieben, was den Modelltrainingsprozess im Vergleich zur Ausführung auf der CPU erheblich beschleunigen kann. Der Optimierungsprozess verwendete den ADAM-Optimizer (Kingma and Ba, 2014) zur Berechnung der Backpropagation. Während der Trainings- und Validierungsphasen des Modells wurde der MSE-Verlust aufgezeichnet und gespeichert, während in der Testphase der MAPE-Verlust gespeichert wurde.

4.5.7. Modellbildung mit Laborversuchsdaten

4.5.7.1. Datenerfassung aus Versuchsanlage

Im Laborversuch wurde ein Plattenwärmeübertrager (PHE) (VT04 CD-16, GEA Ecoflex GmbH, Deutschland) eingesetzt, um einen typischen Foulingprozess in der Molkereiindustrie zu simulieren, bei dem Molkenproteininkonzentrat (WPC) als klassisches Stoffsysteem verwendet wurde. Die schematische Darstellung dieser Anlage, in der das Fouling klassisch auftritt, ist in Abbildung 26 gezeigt. Insgesamt wurden 19 Experimente durchgeführt. Der für dieses Projekt ausgewählte Clamp-On-Sensortyp (EGT311F031, Sauter-Cumulus GmbH, Deutschland) weist einen Messbereich von -10 °C bis 120 °C auf, passend zum Parameterbereich des Prozesses. Die Sensoren wurden an der Rohrleitungs-oberfläche des Heißwassers sowie am Produkt-Ein- und Ausgang angebracht. Um den durch das Setup induzierten Offset zu bestimmen, wurden Inline-Temperatursensoren (PT1000, Klasse AA, Therma Thermofühler GmbH, Deutschland) an denselben Positionen integriert, um die Temperatur der Flüssigkeit im Hauptstrom zu messen. Außerdem wurden die Drücke an den Ein- und Ausgängen mithilfe eines Drucktransmitters (PA-33X, KELLER AG, Winterthur, Schweiz) aufgezeichnet.

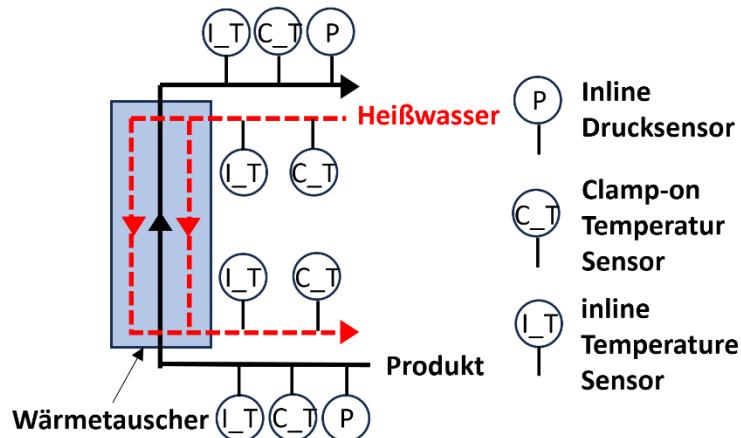


Abbildung 26: Schematische Darstellung der Versuchsanlage mit Clamp-On und Inline-Sensoren

Um die von den Sensoren generierten Daten zu erfassen, wurden das in Abschnitt 4.3.1.1 genannte PiLO-Modul und die entsprechenden Softwarekomponenten verwendet. Dieses Gerät wurde aufgrund seines leichten Designs und seiner Eignung für die Integration in die Anlage ausgewählt. Zusätzlich wurden die Inline-Temperaturdaten mit einem handelsüblichen Datenlogger (Agilent 34970A, Keysight Technologies, Inc., USA) aufgezeichnet.

Zur Datenerfassung wurde eine Reihe von Experimenten durchgeführt, die unter verschiedenen Bedingungen ausgeführt wurden: WPC-Konzentration $C_{WPC} = 70, 75, 80 \text{ g}\cdot\text{L}^{-1}$, Versuchsdauer $t = 60, 120, 180 \text{ Minuten}$, und Eingangsprodukt-Temperatur $T_{Bulk} = 50, 55, 60 \text{ }^{\circ}\text{C}$. Zur Vorbereitung jedes Experiments wurde das WPC-Pulver in Wasser auf einer Heizplatte gelöst, während der Prüfstand vorgeheizt wurde. Um die Experimente zu starten, wurde der Volumenstrom auf der Produktseite auf $0.08 \text{ m}^3\cdot\text{h}^{-1}$ und der

Heizwasserkreislauf auf $1.508 \text{ m}^3 \cdot \text{h}^{-1}$ eingestellt. Die Testplatten des Plattenwärmeübertragers (PHE) wurden vor jedem Experiment extern gereinigt und die trockene sowie nasse Foulingmasse wurde nach jedem Experiment mit einer Laborwaage (Signum 3, Sartorius, Deutschland) bestimmt. Insgesamt wurden 8115 Datenpunkte der Temperaturprofile und 19 der Foulingmasse erfasst. Der Datensatz wurde anschließend in Trainings-, Validierungs- und Testdatensätze aufgeteilt, mit 12 Produktionen für das Training, 4 für die Validierung und 3 für den Test. Um die Ergebnisse zu verallgemeinern, wurden Kreuzvalidierungsmethoden verwendet, indem eine Reihe von Zufallszahlen definiert wurde.

4.5.7.2. Auswahl der Eingangsparameter

Da der Datensatz aus verschiedenen Sensoren und experimentellen Bedingungen besteht, mussten Eingangsparameter für das Modelltraining ausgewählt werden. Fünf Kombinationen von Sensorwerten wurden als Eingaben für das Modell ausgewählt. Tabelle 6 fasst die relevanten Eingabeparameter und die entsprechenden Kombinationen für das Modell zusammen. Vergleichsweise enthalten andere Parameter nicht genügend Informationen. Als Eingabeparameter wurden gezielt gewählte Erfahrungswerte ausgewählt, was zur Leistung des Modells beitragen kann.

Tabelle 6: Eingangsparameter für das Modelltraining

Relevante Eingangsparametern	Kombinationen				
	C1	C2	C3	C4	C5
C_T_Prod_Aus					
I_T_Prod_Aus					
C_T_Heißwasser_Prod_Aus_Diff					
P_Prod_Ein_Aus_Diff					
WPC					
I_T_bulk					

4.5.7.3. Hyperparameter zum Modellaufbau

Die Hyperparameter zum Modelltraining für Labordaten werden im Folgenden aufgelistet.

1. Länge der Eingangssequenz: {50, 70, 100}
2. Länge der Ausgangssequenz: {50, 70, 100}
3. Teacher-Forcing Ratio: {0,2, 0,4}
4. Anzahl der versteckten Schichten: {2}
5. Anzahl des Neurons in einzelne versteckte Schichten: {32, 64}
6. Anzahl der Dropout Ratio: {0,2, 0,4}
7. Größe der Maske für CNN: {3}

4.5.7.4. Ergebnis von Zeitreihenprognose

Abbildung 27 fasst die durchschnittlichen Trainings- und Validierungsverluste während des Trainings für drei Modelle zusammen unter Berücksichtigung verschiedener Zufallszustände bei der zufälligen Auswahl von Trainings- und Testdatensätzen für die Kreuzvalidierung. Die Trainings- und Validierungskurve für das LSTM-Modell ändert sich während der Trainingsphase kaum, wie in Abbildung 27a zu sehen ist. Der MSE-Verlust für das Training liegt bei etwa 0,05 und für die Validierung bei etwa 0,02, was deutlich größer als erwartet ist und danach relativ konstant bleibt. Idealerweise sollte sich der MSE-Verlust nach jedem Backpropagationsschritt in gewissem Maße ändern und letztendlich zur Konvergenz des Modells führen. Im Fall des LSTM-Modells scheinen die Parameter jedoch während des Trainingsprozesses unverändert zu bleiben. Dieser Mangel an signifikanten Veränderungen deutet darauf hin, dass das LSTM-Modell nicht effektiv aus dem Trainingsdatensatz gelernt hat.

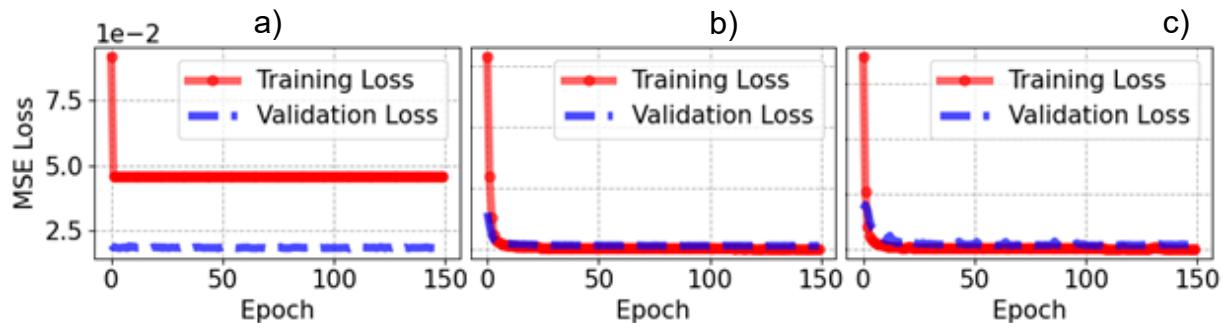


Abbildung 27: Training-Validierung MSE-Verlustprofil für die untersuchten Modelle basierend auf Labordaten: a) Ergebnis von LSTM, b) Ergebnis von CNN-LSTM, c) Ergebnis von Encoder-Decoder LSTM

Vergleichsweise zeigen die CNN-LSTM und Encoder-Decoder LSTM ein unterschiedliches Verhalten in ihren Trainings- und Validierungsphasen, wie in den Abbildung 27a) und Abbildung 27b) dargestellt. In beiden Modellen beginnt der Validierungsverlust bei 0,015 bzw. 0,02 und konvergiert nach etwa 70 Epochen, ähnlich wie der Trainingsverlust, der bei etwa 0,075 begann und nahezu auf 0 konvergierte. Das CNN-LSTM erreicht nach etwa 50 Epochen eine Stabilisierung innerhalb eines akzeptablen Toleranzbereichs, während das Encoder-Decoder LSTM fast 80 Epochen benötigt. Trotz dieser unterschiedlichen Konvergenzzeiten weisen beide Modelle einen Validierungsverlust von etwa 0,0014 auf. Es ist bemerkenswert, dass das CNN-LSTM Modell zu Beginn des Trainings einen niedrigeren initialen Verlust aufweist als das Encoder-Decoder LSTM. Außerdem erscheint die Validierungskurve des CNN-LSTM Modells glatter als die des Encoder-Decoder LSTM. Dieses Verhalten kann auf die Eigenschaften des CNN zurückgeführt werden, bei denen Zeitreihenmerkmale mithilfe einer Maske extrahiert werden, was eine frühere Konvergenz im Vergleich zum Encoder-Decoder LSTM ermöglicht. Zusätzlich ist die Variation des Trainingsverlusts im CNN-LSTM geringer als im Encoder-Decoder LSTM.

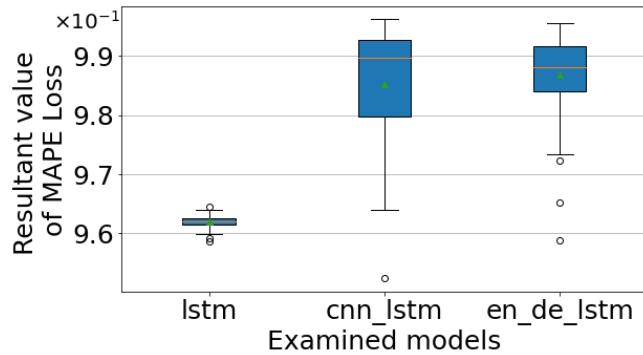


Abbildung 28: Ergebnis des MAPE-Verlusts durch untersuchte Modelle basierend auf Testdatensatz aus Labordaten

Abbildung 28 zeigt die Testverluste, dargestellt in einem Boxplot der MAPE für drei Modelle. Das LSTM Modell weist den kleinsten Interquartilsabstand auf, was auf ein konsistentes Verhalten durch unveränderte Parameter während des Trainings hinweist. Die CNN-LSTM und Encoder-Decoder LSTM erreichen durchschnittliche Genauigkeiten von 98,53 % bzw. 98,68 %. Der Interquartilsbereich für CNN-LSTM erstreckt sich von 97,97 % bis 99,27 % und ist breiter als der des Encoder-Decoder LSTM, der von 98,41 % bis 99,17 % reicht. Bemerkenswert ist, dass der engere Bereich des Encoder-Decoder LSTM und die höhere durchschnittliche Genauigkeit auf eine größere Stabilität und Präzision der Vorhersagen in den Testergebnissen im Vergleich zum CNN-LSTM hindeuten.

Wie bereits erwähnt, wurde auch der Einfluss der Eingabewahl auf die Modellleistung untersucht, um zu prüfen, ob zusätzliche experimentelle Parameter die Wirksamkeit erhöhen. Abbildung 29 veranschaulicht die Analyse von fünf Datenkombinationen und hebt die suboptimale Leistung in c4 und c5 hervor, die auf die konstante Natur von WPC und die Eingangsprodukttemperatur zurückzuführen ist. Diese Parameter, die den Anfangsbedingungen ähneln, tragen nur minimal bei und beeinträchtigen die Modellleistung sogar leicht. Im Gegensatz dazu haben die Modelle mit c1, c2 und c3 als Eingabe eine bessere Leistung erzielt. c1, das sowohl Temperatur als auch Druck umfasst, trug aufgrund der begrenzten Variabilität des Druckunterschieds im Vergleich zur Temperatur nur minimal zur Modellschulung bei. Im Gegensatz dazu erwies sich c2 als bester Performer mit einer mittleren Genauigkeit von 98,14 % und überlegener Stabilität, die auf die Einbeziehung von C_T_product_out, I_T_product_out und C_T_hw_p_out_diff zurückzuführen ist. Die c3-Kombination, die nur C_T_product_out und I_T_product_out umfasst, zeigte eine weniger günstige Leistung und es fehlte an Präzision in der linearen Beziehung zwischen Clamp-On- und Inline-Sensoren. Die optimale Leistung, die von c2 erreicht wurde, kann der Ähnlichkeit der Eingangsparameter zugeschrieben werden, die es dem Modell ermöglicht, effektiver aus den drei Zeitreihendatensätzen zu lernen. Die Daten zeigen, dass die Variablen C_T_product_out und I_T_product_out den größten Einfluss auf die Modellleistung haben. Es wäre auch wertvoll, die I_T_product_out-Sensordaten aus dem Trainingsprozess des Modells auszuschließen, um zu bewerten, wie das trainierte Modell ohne diese Eingabe abschneidet. Im Gegensatz dazu fehlt es c3 an der Einbeziehung der Differenzwerte zwischen der Heißwasser- und der Produktausgangstemperatur, was dazu führt, dass das Modell wichtige Informationen verpasst. Darüber hinaus weist der Eingangsparameter Druck, der in der c1-Kombination verwendet wird, eine signifikant

andere Datenverteilung im Vergleich zur Temperatur auf, was zu einem Rückgang der Leistung führt.

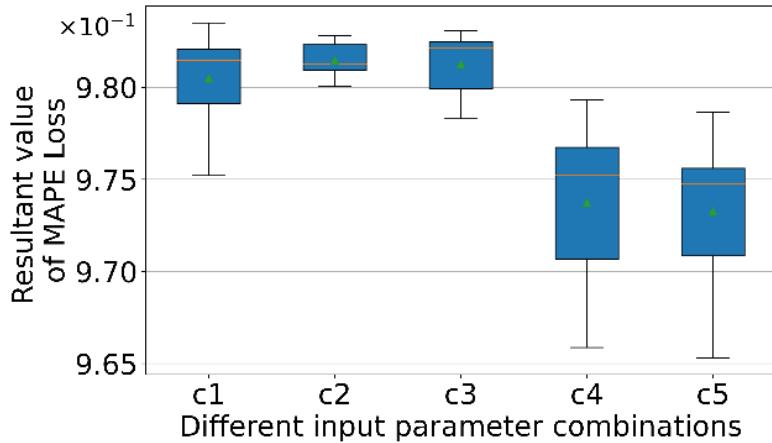


Abbildung 29: Vergleich des MAPE-Verlusts bei verschiedenen Kombinationen von Eingangsparametern basierend auf Labordaten

Um einen klareren Überblick über die Leistung der verschiedenen Modelle zu geben, wurde ein Testdatensatz ausgewählt und auf die trainierten Modelle angewendet, um die Produktausgangstemperatur basierend auf diesem Datensatz vorherzusagen. Die Ergebnisse sind in Abbildung 30 dargestellt. Wie zu beobachten ist, bleiben die Ergebnisse des reinen LSTM Modells konstant, ähnlich wie in Abbildung 27, wo der MSE-Verlust nicht konvergiert. Dies führt zu einer schlechten Leistung bei der Vorhersage von Sequenzdaten. Im Gegensatz dazu schneiden die Encoder-Decoder LSTM und CNN LSTM recht gut ab. Beide Modelle können den abnehmenden Trend der Produktausgangstemperatur vorhersagen, wobei der maximale Vorhersagefehler unter 1 °C bleibt.

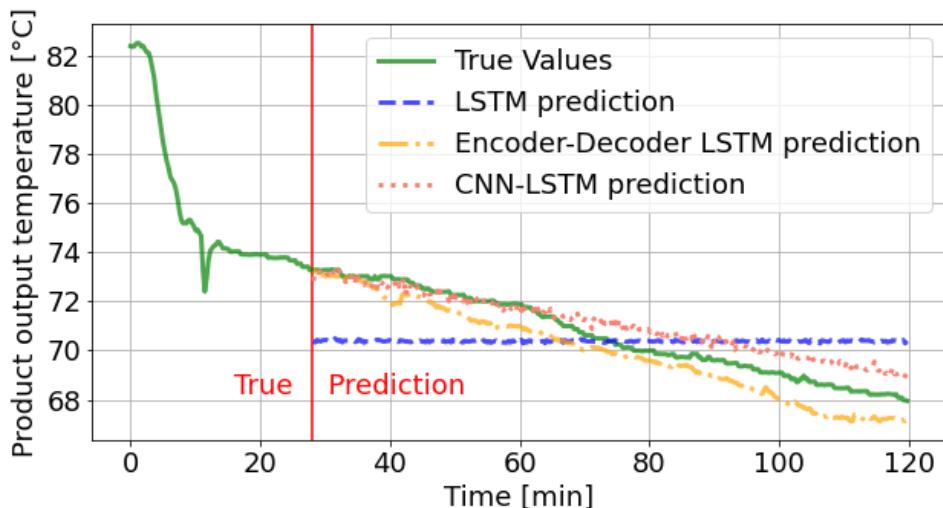


Abbildung 30: Vergleich der vorhergesagten Werte verschiedener Modelle basierend auf Labordaten

4.5.7.5. Ergebnis von Vorhersage der Foulingmasse

Da die Dauer der Foulingversuche in der Laborumgebung variabel ist, um als Trainingsdaten für die Regressionsmodelle genutzt werden zu können, dient hier die Produktausgangstemperatur als Indikator zur Bestimmung der Foulingmasse. Daher bestehen die folgenden Sensorwerte aus dem Merkmalsvektor für die Regression: WPC, Zeitdauer, T_Bulk und die Clamp-On Produktausgangstemperatur. Der Zielvektor ist die Foulingmasse.

Zur Schätzung der Foulingmasse wurden klassische Regressionsmodelle verwendet. Variabilität wurde während des Trainings durch die Einführung eines zufälligen Seeds für eine verbesserte Generalisierbarkeit der Ergebnisse eingeführt. Die in Abbildung 31 gezeigten Ergebnisse stellen dar, dass SVR den niedrigsten MAPE von 80,78 % erreicht hat, während LR und RF vergleichbare Genauigkeiten von 83,06 % bzw. 83,48 % erzielten. SVR hat Schwierigkeiten, da die Effizienz seiner Hyperebene bei der Trennung hochdimensionaler Datenpunkte eingeschränkt ist. RF übertrifft LR um 0,42 %, indem es eine Gruppe von Schätzern nutzt, um die Eingangs-Ausgangs-Beziehung besser darzustellen. Trotz der längeren Trainingszeit von RF (50-mal länger) wird die Vorhersagedauer weniger wichtig, sobald das Modell erstellt ist. LR weist eine durchschnittliche Vorhersagezeit von 0,3 ms auf, während RF 9 ms benötigt, also 30-mal länger, jedoch immer noch innerhalb eines akzeptablen Zeitrahmens. Nach der Modellauswahl umfassen die optimalen Hyperparameter eine maximale Baumtiefe von 3 und 50 Schätzer, was zu einer Vorhersagegenauigkeit von 85,12 % führt.

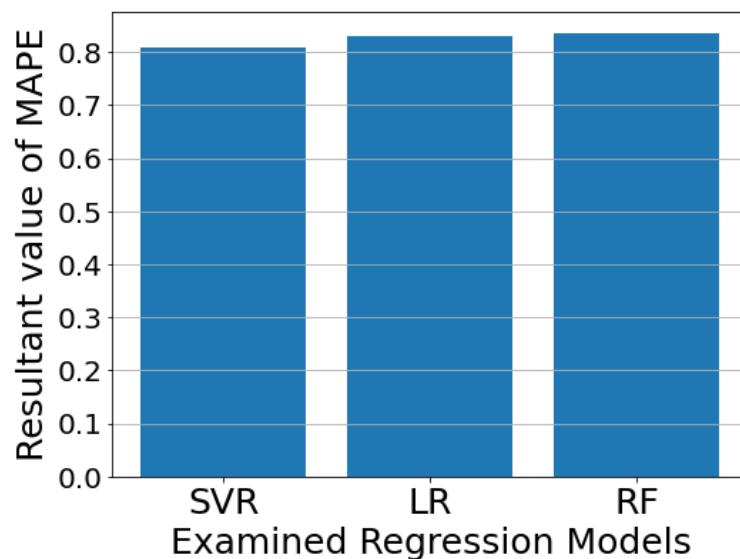


Abbildung 31: Resultierender MAPE Verlust von verschiedenen Regressionsmodellen basierend auf Testdatensatz aus Labordaten

4.5.8. Modellbildung mit Daten aus Firma 1

Der Datensatz aus Firma 1 umfasst 16 Produktionen über 10 Tage. Einige der Produktionen wurden jedoch nicht vollständig abgeschlossen, weshalb 4 Produktionsdaten ignoriert wurden. Die verbleibenden 12 Produktionen werden in Trainings-, Validierungs- und Testdatensätze aufgeteilt, mit 6 Produktionen für das Training, 3 für die Validierung und

3 für den Test. Um die Generalisierbarkeit der Ergebnisse zu verbessern, wird zudem eine Kreuzvalidierung durchgeführt. Zuerst wird eine Reihe zufälliger Zahlen definiert, mit deren Hilfe die Datensätze variiert werden können, wodurch die Ergebnisse allgemeingültiger werden. Aus Abschnitt 4.5.7 lässt sich feststellen, dass das reine LSTM-Modell bei der Sequenz-zu-Sequenz-Vorhersage nicht gut funktioniert. Daher wird das reine LSTM-Modell für weitere Fälle nicht in Betracht gezogen.

4.5.8.1. Auswahl der Eingangsparameter

Aus der Datenanalyse von AP1 wurde herausgefunden, dass die folgenden Sensoren für das Modelltraining verwendet werden sollten, da in den Sensordaten deutliche Tendenzen erkennbar sind:

- Produkt Eingangsdrucksensor nach dem Regenerator vor dem Erhitzer (Erhitzer_Produkt_in_P)
- Leistungssensor von Booster Pumpe (Booster_Pumpe_Leistung)
- Produkt Ausgangsdrucksensor nach dem Regenerator vor Erhitzer (Reg_Produkt_Aus_P)
- Produkt Ausgangstemperatursensor nach dem Regenerator vor Erhitzer (Reg_Produkt_Aus_T)
- Temperatursensor von Reinigungsmedium bei Rückgabe von CIP (Rückgabe_CIP_T)

Außerdem zeigen die Verläufe dieser vier Sensoren ähnliche Trends, sodass auch die Eingangsparameter für das Modelltraining variiert werden konnten. Tabelle 6 zeigt, wie die Sensoren kombiniert werden.

Tabelle 6: Kombination von relevanten Parametern als Input für das Modelltraining

Eingangsparameter	Kombinationen		
	C1	C2	C3
Erhitzer_Produkt_in_P			
Booster_Pumpe_Leistung			
Reg_Produkt_Aus_P			
Reg_Produkt_Aus_T			
Rückgabe_CIP_T			

Hinzu kommt die Rücklauftemperatur des CIP (Rückgabe_CIP_T). Laut dem Ansprechpartner der Firma spielt dieser Parameter in der Produktion keine wesentliche Rolle. Allerdings zeigen die Verläufe ebenfalls eine steigende Tendenz. Der durchschnittliche Wert des Integrals der Autokorrelationsfunktion beträgt während der Produktion 0,9839.

Aus diesem Grund wird der Sensor als optionaler Parameter für das Modelltraining betrachtet. Es sollte überprüft werden, ob dadurch die Genauigkeit der Modellvorhersage verbessert werden kann. Der Heißwasser-Eingangstemperatursensor (Erhitzer_Heißwasser_in_T) wird nicht berücksichtigt. In Abbildung 32 ist der Temperaturverlauf dargestellt. Es lässt sich erkennen, dass nicht jeder Temperaturverlauf in allen Produktionen einen signifikanten Anstieg zeigt. Zudem beträgt das durchschnittliche Integral der Autokorrelationsfunktion 0,233, was im Vergleich zu den anderen ausgewählten Sensoren kleiner ist.

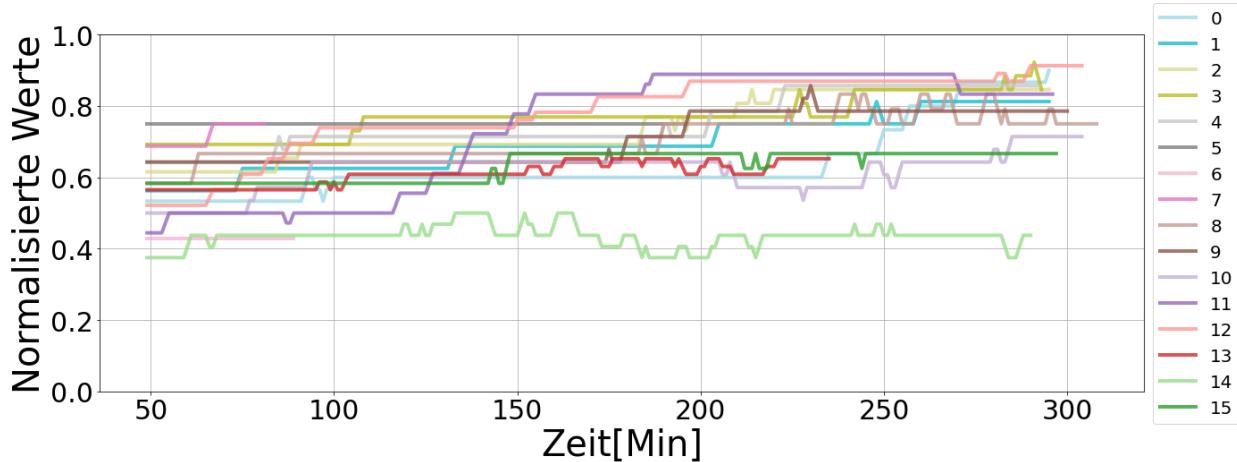


Abbildung 32: Normalisierte Temperaturwerte am Heißwassereingang vor dem Erhitzer für einzelne ProduktionsID während der Produktion

Deshalb wurden drei Kombinationen ausgewählt, die als Eingangsparameter für das Modell dienen.

4.5.8.2. Hyperparameter zum Modellaufbau

Die Hyperparameter zum Modelltraining für Firma 1 werden im Folgenden aufgelistet.

1. Länge der Eingangssequenz: {50, 70, 100}
2. Länge der Ausgangssequenz: {50, 70, 100}
3. Teacher-Forcing Ratio: {0,2, 0,4}
4. Anzahl der versteckten Schichten: {2}
5. Anzahl des Neurons in einzelne versteckte Schichten: {32, 64}
6. Anzahl der dropout Ratio: {0,2, 0,4}
7. Größe der Maske für CNN: {3}

4.5.8.3. Ergebnis der Zeitreihenprognose

Vergleichsweise werden die Training-Validierungskurve CNN-LSTM und Encoder-Decoder LSTM in Abbildung 33 gezeigt. Beim CNN-LSTM beginnt der Validierungsverlust bei 0,147, und beim Encoder-Decoder bei 0,137, und beide Modelle konvergieren nach etwa 50 Epochen, ähnlich wie der Trainingsverlust, der bei 0,26 beginnt und nach 50 Epochen

gegen 0 konvergiert. Das Encoder-Decoder LSTM stabilisiert sich innerhalb eines akzeptablen Toleranzbereichs nach ungefähr 30 Epochen, während das CNN-LSTM fast 50 Epochen benötigt.

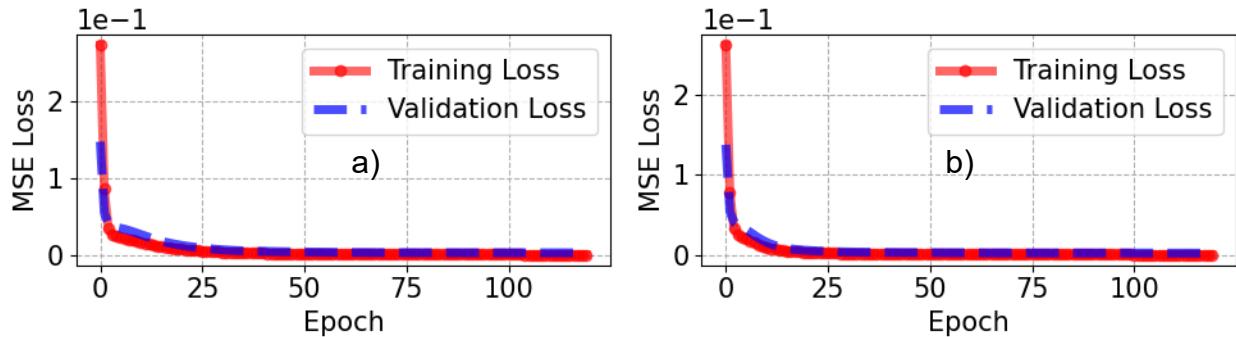


Abbildung 33: Training-Validierung MSE-Verlustprofil für die untersuchten Modelle basierend auf Firma 1 Daten: a) Ergebnis von CNN-LSTM, b) Ergebnis von Encoder-Decoder LSTM

Die Abbildung 34 zeigt die Vorhersagegenauigkeit der Modelle auf dem Testdatensatz. CNN-LSTM und Encoder-Decoder LSTM erzielten durchschnittliche Genauigkeiten von 99,49 % bzw. 99,46 %. Daraus ergibt sich, dass CNN-LSTM im Durchschnitt eine höhere Genauigkeit aufweist als Encoder-Decoder LSTM. Zudem ist die Interquartilsspanne von CNN-LSTM breiter als die des Encoder-Decoder LSTM, was darauf hinweist, dass die Vorhersagestabilität von CNN-LSTM besser ist. Der Grund dafür könnte sein, dass die Rohdaten bei Encoder-Decoder LSTM ohne Filter als Trainingsdaten verwendet wurden. Das Rauschen könnte die Vorhersagegenauigkeit des Encoder-Decoder LSTM beeinflusst haben. Im Gegensatz dazu enthält CNN-LSTM eine eingebaute konvolutionale Schicht, die als gleitender Durchschnittsfilter wirkt. Dadurch wird das Rauschen in den Rohdaten teilweise gefiltert, bevor die Daten in das Modell eingespeist werden.

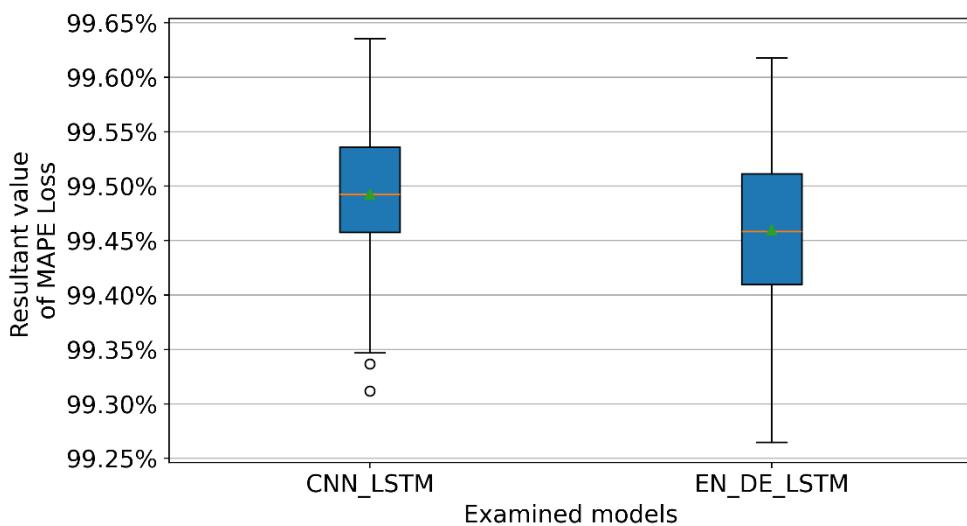


Abbildung 34: Ergebnis des MAPE-Verlusts durch untersuchte Modelle basierend auf Testdatensatz aus Firma 1 Daten

Abbildung 35 zeigt die Modellvorhersagegenauigkeit auf dem Testdatensatz bei unterschiedlichen Kombinationen von Eingangsparametern. Aus dieser Abbildung ist zu erkennen, dass die Kombination c1 im Durchschnitt die größte Genauigkeit aufweist. Allerdings weist die c1-Kombination im Vergleich zu anderen Kombinationen auch mehr Ausreißer auf. Das könnte bedeuten, dass c1 zwar eine gute Vorhersagegenauigkeit hat, in einigen Fällen jedoch nicht stabil ist. Die Ausreißer können in weiteren Trainingsprozessen durch Anpassung der Hyperparameter beseitigt werden. Da durch Kreuzvalidierung und Hyperparameter-Tuning eine große Anzahl an Testergebnissen generiert wurde, etwa 2880 Datenpunkte, spielen die etwa sieben Ausreißer bei der Auswertung eine vernachlässigbare Rolle. Wir können davon ausgehen, dass die Vorhersagegenauigkeit und -stabilität von c1 besser sind als die der anderen Kombinationen.

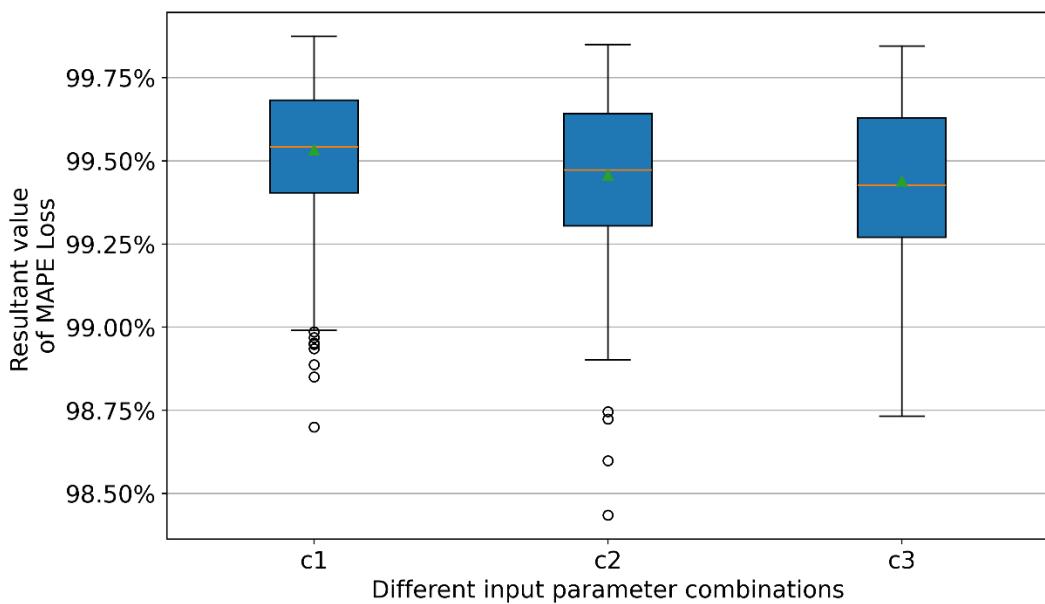


Abbildung 35: Vergleich des MAPE-Verlusts bei verschiedenen Kombinationen von Eingangsparametern basierend auf Firma 1 Daten

Gleich wie bei den Labordaten, wurde ein Testdatensatz ausgewählt und auf die trainierten Modelle angewendet. Die Ergebnisse sind in Abbildung 36 dargestellt. Das Encoder-Decoder und CNN-LSTM Modell schneiden relativ gut ab. Beide Modelle können den abnehmenden Trend des Produktausgangsdrucks vorhersagen, wobei sich der CNN-LSTM wenig besser als Encoder-Decoder LSTM verhält. Der maximale Vorhersagefehler bleibt unter 35 mbar, die vorhergesagten Werte nähern sich den echten Daten an. Die Werte des Encoder-Decoder-Modells weisen vergleichsweise eine geringere Vorhersagegenauigkeit auf. Der maximale Fehler beträgt ungefähr 50 mbar und es zeigt eine wesentlich größere Abweichung.

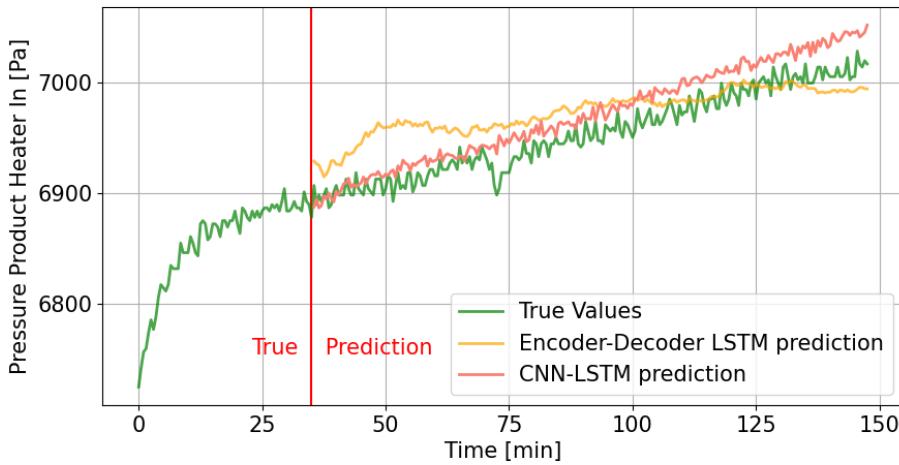


Abbildung 36: Vergleich der vorhergesagten Werte verschiedener Modelle

4.5.8.4. Vergleich der Ergebnisse mit den auf Labordaten trainierten Modellen

Die Ergebnisse sind im Vergleich zu denen, die mit Labordaten trainiert wurden, nicht so gut, da die Datenmenge eingeschränkt ist. Wir haben von Firma 1 16 Produktionsdatensätze erhalten, von denen 4 Produktionen nicht vollständig durchgelaufen sind. Diese wurden nicht zum Modellaufbau verwendet. Für das Training, die Validierung und den Test wurden die Datensätze in 6, 3 und 3 aufgeteilt. Die gesamte nutzbare Stichprobe beträgt 3508 Datenpunkte, nachdem die unvollständigen Daten entfernt wurden. Durch die Aufteilung standen nur knapp 2000 Datenpunkte für den Modellaufbau zur Verfügung. Daher ist auch die Modellvorhersagegenauigkeit eingeschränkt.

Wenn wir die Testergebnisse betrachten, sind sie im Durchschnitt größer als die aus den Labordaten. Das bedeutet jedoch nicht, dass die Modellvorhersagegenauigkeit auf den Daten von Firma 1 besser ist als auf den Labordaten. In den Labordaten sieht man, dass die Produktausgangstemperatur am Produktionsende um etwa 10 Grad von ungefähr 80 Grad sinkt, während der Eingangsdruck des Erhitzers nur eine geringe Differenz aufweist, etwa 200 mBar von rund 7.600 mBar. Betrachtet man das Verhältnis zwischen beiden Datensätzen, kann man sehen, dass das Verhältnis bei den Labordaten etwa 0,125 der gesamten Datenreichweite beträgt, während die Differenz zwischen Produktionsbeginn und -ende bei den Daten von Firma 1 nur um 0,025 variiert. Das bedeutet, dass die Modelle bei den Daten von Firma 1 einen geringeren Änderungstrend modellieren müssen. Folglich sind die Testergebnisse auf den Daten von Firma 1 zwar größer als bei den Labordaten, das bedeutet jedoch nicht zwangsläufig, dass die Vorhersagegenauigkeit besser ist. Um die Modellvorhersagegenauigkeit zu steigern, sehen wir zwei Möglichkeiten. Erstens muss die Datenmenge erhöht werden, damit die Modelle mehr Informationen aus den Daten extrahieren können. Zweitens sollte ein Intervall für die betrachteten Daten gewählt werden, das durch den minimalen und maximalen Wert während der Produktion definiert ist, denn dadurch wird der Nenner bei der Berechnung der MAPE-Metrik deutlich geringer als bei den Originalwerten.

4.5.8.5. Analyse der Reinigungsdaten zur Vorhersage der Foulingmasse

Im Gegensatz zu Labordaten ist es in der Industrie oft sehr schwierig, Daten zu erhalten, die direkt die Foulingmasse zeigen. In der Industrie sollte die Produktion möglichst nicht gestört oder gestoppt werden. Daher ist es nicht möglich, die Anlage zu beliebigen Zeitpunkten zu demontieren, um die Daten von Foulingmasse zu gewinnen. Aus diesem Grund kann nur der Reinigungsprozess untersucht werden, um herauszufinden, welche Informationen aus den Daten gewonnen werden können. Ein Beispiel wäre, dass nach einer Produktion mehr Fouling entstanden ist als bei anderen, was dazu führen sollte, dass bei der Reinigung die Temperatur des Reinigungsmediums erhöht wird.

In Abbildung 37 wird ein Sequenzdiagramm von Produktion und Reinigung dargestellt. Daraus ist zu erkennen, dass die Produktion und Reinigung normalerweise nacheinander durchgeführt werden. In einigen Fällen werden zwei Produktionen zur Herstellung der Produkte 6 und 7 durchgeführt, gefolgt von einer Reinigung. Nach der Produktions-ID 12 zur Herstellung des Produktes 1 werden vier Reinigungen durchgeführt. Nach unserer Annahme sollte die Temperatur bei den Reinigungen 10, 11, 12 und 13 stets niedriger sein, da das Fouling durch die letzten Reinigungen entfernt wird.

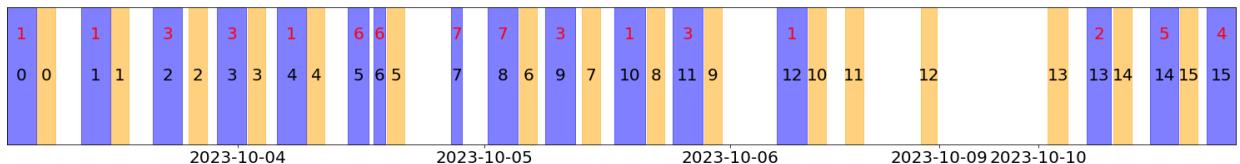


Abbildung 37: Sequenzdiagramm, stellt die Produktions- und Reinigungsphasen über die Zeit dar. Die lila Blöcke repräsentieren die Produktionsphasen, während die senfgelben Blöcke die Reinigungsphasen darstellen. Die schwarzen Zahlen sind die entsprechende Produktion.

Die Reinigung 12 war nicht vollständig durchgelaufen bzw. war fehlerhaft und wird für weitere Analysen ignoriert. In den Abbildung 38, Abbildung 39 und Abbildung 40 ist zu sehen, dass die Temperatur beim Produkt des Erhitzers von Reinigung 10 bis 13 stetig abnimmt. Das Gleiche gilt auch für die Temperatur und den Druck beim Produktausgang des Regenerators. Dadurch kann festgestellt werden, dass die Dicke der Fouling-Schicht durch die effektive Reinigung immer dünner wird. Aus den Daten können wir lediglich folgern, dass die Fouling-Schicht durch die CIP-Prozesse entfernt wird. Allerdings kann die genaue Dicke der Fouling-Schicht nicht abgeleitet werden.

Die Temperatur und der Druck können noch nicht als Maß für die "Reinigungsschwierigkeit" verwendet werden. Bei den CIP-IDs 0, 1, 8 und 10 sind zudem Unterschiede in den Druckwerten beim Produktausgang des Regenerators zu erkennen. Theoretisch sollten diese Werte fast identisch sein, da das gleiche Produkt über die gleiche Zeitdauer hergestellt wurde. Zudem ist die Datenmenge zu klein, und da jede Produktion die gleiche Zeitdauer hat, ist es nicht möglich, die "Reinigungsschwierigkeit" zu einem beliebigen Zeitpunkt während der Produktion abzuschätzen.

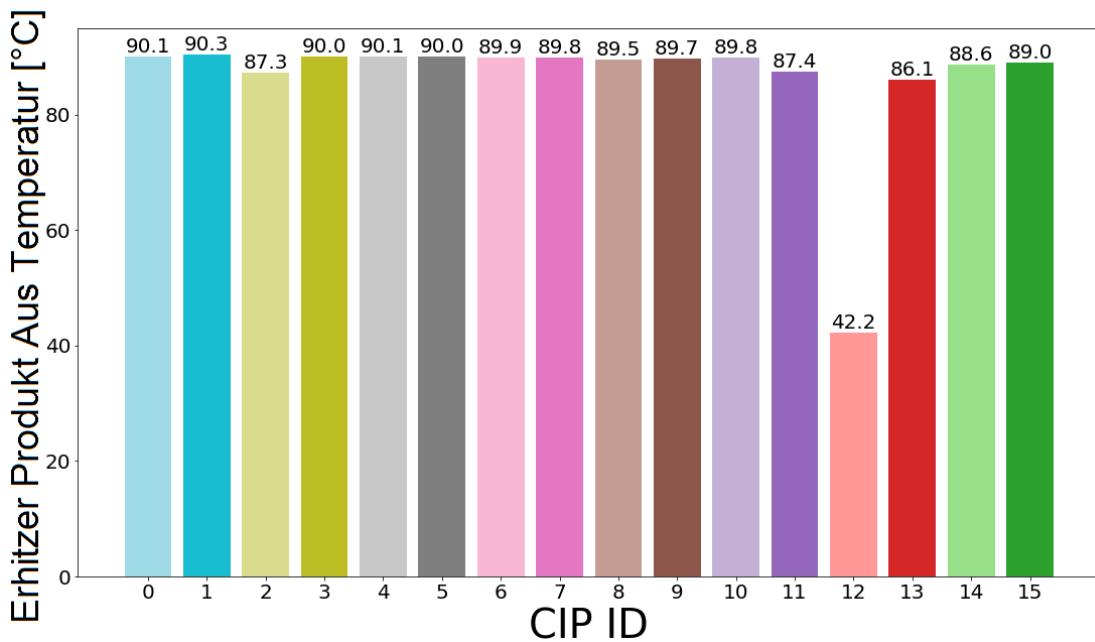


Abbildung 38: Durchschnittliche Temperaturwerte beim Produktausgang von Erhitzer

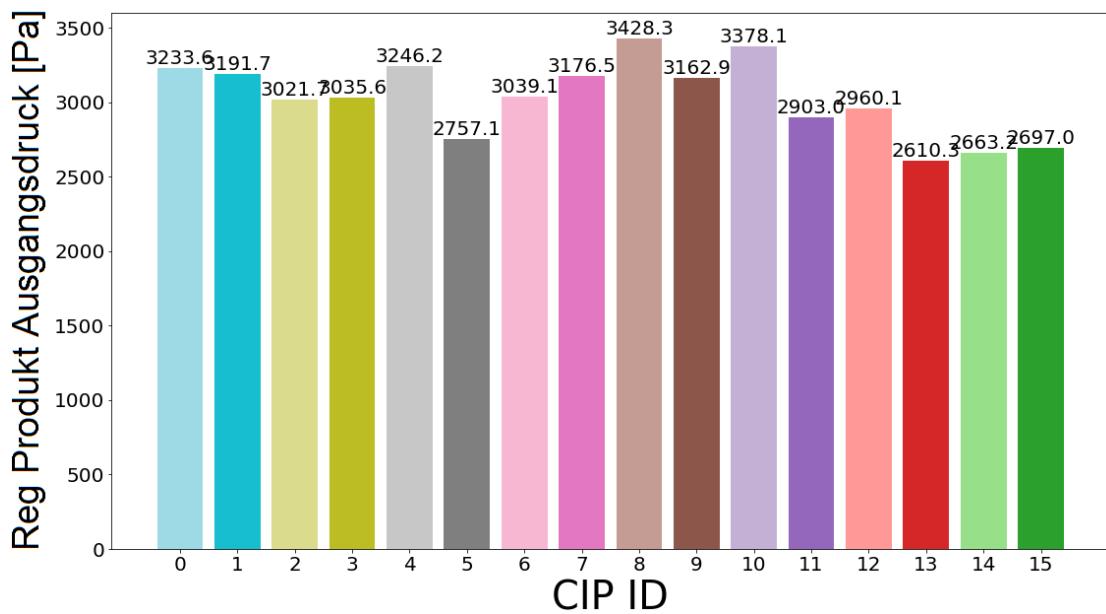


Abbildung 39: Durchschnittliche Druckwerte beim Produktausgang von Regenerator

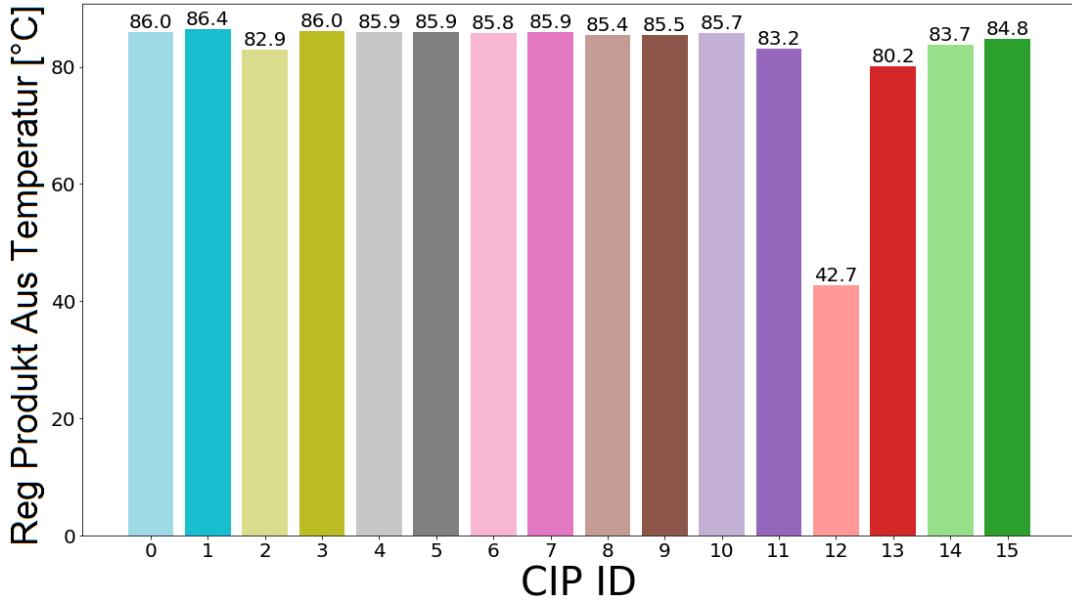


Abbildung 40: Durchschnittliche Temperaturwerte beim Produktausgang von Regenerator

4.5.9. Modellbildung mit Daten aus Firma 2

4.5.9.1. Auswahl der Eingangsparameter

Im Gegensatz zu den Datensätzen aus dem Labor und Firma 1, bei denen viele Sensoren zur Auswahl stehen, verfügt der Datensatz von Firma 2 nur über eine begrenzte Anzahl an Sensoren. Aus diesem Grund wurden 2 zusätzliche Softsensoren definiert. Tabelle 7 zeigt, wie die Sensoren kombiniert und als Eingabewerte in die Modelle eingespeist werden.

Tabelle 7: Kombination von relevanten Parametern als Input für das Modelltraining

Eingangsparameter	Kombinationen			
	C1	C2	C3	C4
C_T_Heißwasser_Ein				
C_T_Heißwasser_Aus				
C_T_Diff_HW_Ein_Prod_Aus				
C_T_Diff_HW_Aus_Prod_Aus				

4.5.9.2. Hyperparameter zum Modellaufbau

Die Hyperparameter zum Modelltraining für Firma 2 werden im Folgenden aufgelistet.

1. Länge der Eingangssequenz: {120, 180}

2. Länge der Ausgangssequenz: {60, 120}
3. Teacher-Forcing Ratio: {0,2, 0,4}
4. Anzahl der versteckten Schichten: {2}
5. Anzahl des Neurons in einzelne versteckte Schichten: {32, 64}
6. Anzahl der dropout Ratio: {0,2, 0,4}
7. Größe der Maske für CNN: {5}

Da die Produktionszeit bei Firma 2 wesentlich länger ist, ist die Länge der Eingangs- und Ausgangssequenzen größer als die der Daten aus dem Labor und von Firma 1. Deshalb wurde die Größe der Maske für das CNN entsprechend auf 5 erhöht.

4.5.9.3. Ergebnis von Zeitreihenprognose

Anhand der Abbildung 41 ist erkennbar, dass keines der beiden Modelle nahe an 0 konvergiert. Nach etwa 50 Epochen steigt die Validierungskurve des CNN-LSTMs an. Im Gegensatz dazu bleibt die Validierungskurve des Encoder-Decoder LSTM fast durchgehend in einem konstanten Abstand zur Trainingsverlustkurve. Dies deutet darauf hin, dass das CNN-LSTM nach etwa 50 Epochen beginnt zu overfitten, d.h. es fängt an, sich an den Trainingsdatensatz zu "erinnern". Im Vergleich dazu weist das Encoder-Decoder LSTM eine bessere Generalisierungsfähigkeit auf.

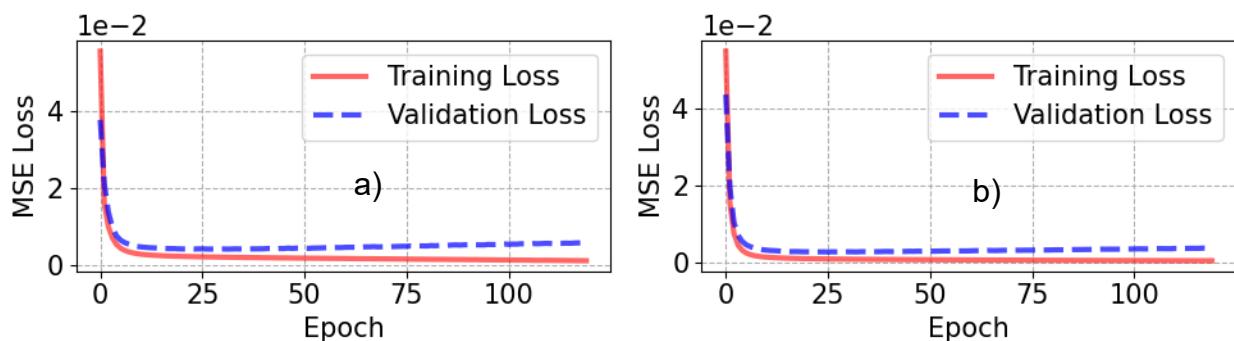


Abbildung 41: Training-Validierung MSE-Verlustprofil für die untersuchten Modelle aus Firma 2
Daten: a) Ergebnis von CNN-LSTM, b) Ergebnis von Encoder-Decoder LSTM

Wie bei den anderen beiden Datensätzen werden auch hier die Boxplots für den resultierenden MAPE-Verlust auf dem Testdatensatz beider Modelle in Abbildung 42 dargestellt. Daraus ist ersichtlich, dass die Testergebnisse beider Modelle eine nahezu gleiche durchschnittliche Vorhersagegenauigkeit aufweisen, wobei das CNN-LSTM um 0,05 % besser abschneidet als das Encoder-Decoder LSTM. Allerdings hat das CNN-LSTM ein breiteres Interquartilsintervall, was auf eine größere Variabilität in den Vorhersagen im Vergleich zum Encoder-Decoder LSTM hinweist. Das Encoder-Decoder LSTM zeigt eine relativ schmalere Verteilung, was auf eine konsistenter Leistung auf dem Testdatensatz hindeutet. Betrachtet man dazu Abbildung 41, sieht man, dass das CNN-LSTM nach etwa 80 Epochen nicht weiter konvergiert, was zu einer größeren Variabilität auf dem Testdatensatz führt.

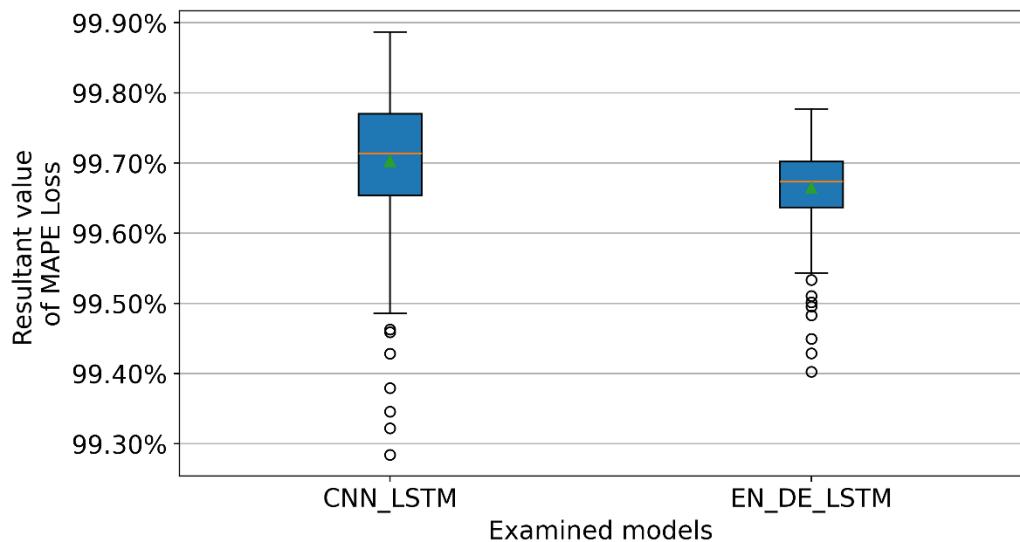


Abbildung 42: Ergebnis des MAPE-Verlusts durch untersuchte Modelle basierend auf Testdatensatz aus Firma 2 Daten

In Abbildung 43 wird ein Beispiel für die Vorhersagen verschiedener Modelle vergleichsweise dargestellt. Aus dieser Abbildung lässt sich erkennen, dass die Vorhersagegenauigkeit beider Modelle nicht besonders gut ist. Das Encoder-Decoder-LSTM kann das Signal noch bis zu 1,5 Stunden nach dem aktuellen Zeitpunkt nachbilden, danach beginnt das ursprüngliche Signal stark zu schwanken. Danach zeigt das Encoder-Decoder-LSTM keine gute Leistung auf dem Testdatensatz. Betrachtet man das CNN-LSTM, zeigt sich, dass es innerhalb von 1,5 Stunden eine ähnliche Leistung wie das Encoder-Decoder-LSTM erbringt. Nach diesen 1,5 Stunden steigt die Kurve jedoch an und fällt anschließend wieder ab.

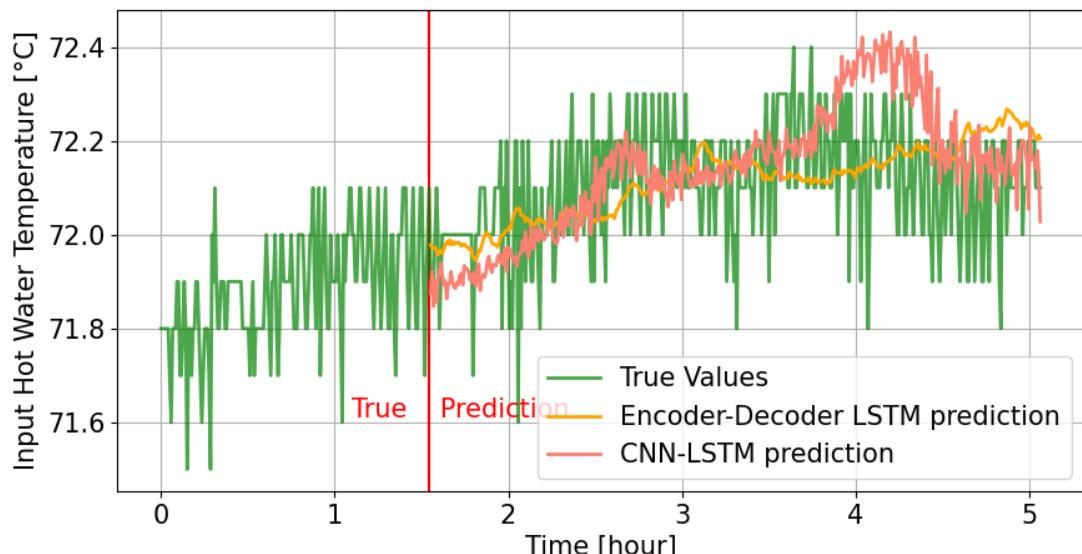


Abbildung 43: Vergleich der vorhergesagten Werte verschiedener Modelle

Die Modelle zeigen aus den Daten von Firma 2 nicht die gleiche Leistung wie aus den anderen beiden Datensätzen. Die trainierten Modelle können den Signalverlauf nicht ausreichend gut abbilden. Wenn man sich die Modellvorhersagen auf dem Testdatensatz ansieht, ist festzustellen, dass einige Modelle den steigenden Trend der Heißwassereingangstemperatur nicht korrekt erfassen. Dies kann aus den folgenden Gründen resultieren:

1. Unklarer Temperaturverlauf: Wie in Abbildung 44 zu sehen ist, weist der Temperaturverlauf keinen klaren Trend auf. Das führt möglicherweise dazu, dass bei der Datenaufbereitung für gleiche Eingabesequenzen unterschiedliche Ausgabesequenzen erstellt wurden. Zum Beispiel könnte eine Eingabesequenz [70, 71, 72, 73, 74, 75, 76, 77, 78, 79] einen steigenden Trend zeigen, während die zugehörige Ausgabesequenz ganz unterschiedliche Werte haben könnte, z.B. [80, 81, 82, 83, 84, 85, 86, 87, 88, 89], [80, 79, 78, 77, 76, 75, 74, 73, 72, 71] oder [71, 75, 73, 76, 79, 82, 78, 79, 80, 70]. Beim Modelltraining ist es möglich, dass solche inkonsistenten Eingabe- und Ausgabesequenzen verwendet werden. Da dieser Datensatz keine wiederholbaren Muster enthält, die die Modelle nachbilden könnten, ist es schwierig, einen passenden Parametersatz zu finden, um eine sinnvolle Ausgabesequenz zu erzeugen. Im gesamten Datensatz beträgt der Anteil solcher inkonsistenten Daten über 60%.

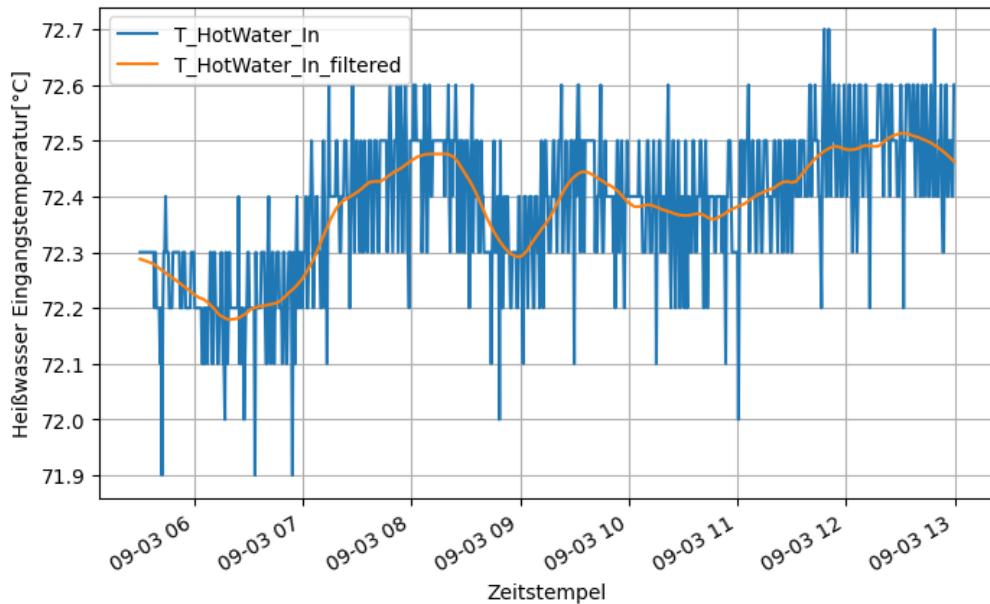


Abbildung 44: Beispieldaten einer Produktion von Firma 2: Der Temperaturverlauf zeigt während der Produktion keinen eindeutigen Trend. Die blaue Linie stellt die verrauschten Originaldaten dar, während die orange Linie die gefilterten Daten zeigt.

2. Ungeeignete Datenvorbereitungsmethode (Sliding-Window-Methode): Die Sliding-Window-Methode verwendet ein Fenster, das schrittweise von vorne nach hinten verschoben wird, um Ein- und Ausgabesequenzen zu erstellen. Der Datensatz wird auf diese Weise automatisch, ohne menschliche Kontrolle, vorbereitet. Daher wurden alle möglichen Ein- und Ausgabesequenzen in das Modelltraining eingespeist, was möglicherweise zu einer fehlerhaften Datenrepräsentation führte.

4.5.9.4. Vergleich der Ergebnisse mit den Daten aus Labor und Firma 1

Wenn man die Ergebnisse der Firma-2 Daten mit denen der Labor- und Firma-1 Daten vergleicht, sieht man, dass das Encoder-Decoder LSTM im Vergleich zum CNN-LSTM beim Modelltraining näher an 0 konvergiert. Dies deutet darauf hin, dass es beim Encoder-Decoder-LSTM schwieriger ist, ein Overfitting-Problem zu verursachen. Daher verfügt das Encoder-Decoder-LSTM über eine bessere Generalisierungsfähigkeit.

Bei den Labordaten haben das Encoder-Decoder-LSTM und das CNN-LSTM eine fast identische Leistung, da die Labordaten kaum Rauschen enthalten und einen signifikanten abfallenden Trend aufweisen. Das bedeutet, dass beide Modelle bei der Nachbildung solcher Signale eine ähnliche Leistung erbringen. Zudem läuft das Training des Encoder-Decoder-LSTM 1,5-mal schneller als das des CNN-LSTM.

Bei den Firma-1 Daten verhält sich das CNN-LSTM generell besser als das Encoder-Decoder LSTM, da das Signal Rauschen enthält. Dieses Rauschen kann das Encoder-Decoder LSTM verwirren und somit die Vorhersagegenauigkeit verringern. Daraus lässt sich schließen, dass das CNN-LSTM bei der Nachbildung verrauchter Signale eine bessere Leistung zeigt.

Bei den Firma-2 Daten schneiden beide Modelle nicht besonders gut ab. Im Machine Learning sind die Daten von entscheidender Bedeutung. Wenn die Datenqualität nicht gut ist, wird selbst das beste Modell scheitern, sinnvolle Ergebnisse zu liefern. Im vorherigen Abschnitt wurde bereits besprochen, dass die Firma-2-Daten inkonsistent sind. Das lässt sich mit einer maschinellen Übersetzung vergleichen: In einem Trainingsdatensatz lautet ein Beispiel für die Ein- und Ausgabe „Hello“ → „Hallo“, in einem anderen Beispiel lautet die Ausgabe jedoch „Danke“ für dieselbe Eingabe „Hello“. Ein solch inkonsistenter Datensatz führt zwangsläufig dazu, dass das Übersetzungsmodell fehlschlägt.

4.5.9.5. Aufbau des Demonstrators zur Veranschaulichung der Zeitreihenvorhersage

Wie in Abschnitt 4.3.3 erwähnt, dient der live_displayer Dienst als Graphical User Interface (GUI) Dienst, d.h. die gesammelten Daten und die daraus generierte Vorhersage werden auf der GUI dargestellt. Abbildung 45 zeigt einen Screenshot des laufenden Dienstes. Die grüne vertikale Linie stellt den aktuellen Zeitpunkt dar und trennt die aufgezeichneten Ist-Daten von den vorhergesagten Daten. Die blaue Linie stellt die erfasssten Sensorwerte dar. Die gestrichelte blaue Linie stellt die vorhergesagten Werte ab dem aktuellen Zeitpunkt dar. Die Anzeige der Vorhersage kann mit einer relativ geringen Verzögerung von durchschnittlich 2 ms erfolgen. Da die Größe des gelernten Modells relativ klein ist (460kb), wird auch nicht viel Arbeitsspeicher benötigt. Dadurch kann das Modell auf einem Edge Device laufen, z.B. Raspberry Pi.

Sensor Profile: C_T_Prod_Aus[°C] and forecasted Profile

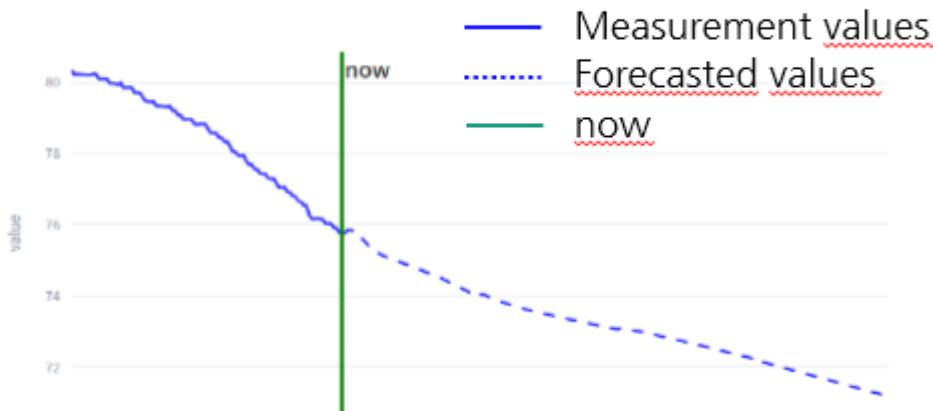


Abbildung 45: Darstellung der erfassten Daten und der daraus generierten Zeitreihenvorhersage. Auf der x-Achse ist die Zeit und auf der y-Achse die Temperatur aufgetragen.

5. Öffentlichkeitsarbeit, Veröffentlichungen und Vorträge

Bereits im frühen Verlauf des Projektes konnten das Kernkonzept sowie Versuchsdaten der im AP4 generierten Ergebnisse an verschiedenen Stellen veröffentlicht werden. Die frühe Vertiefung dieses Arbeitspakets bot sich an, da die hier durchgeführten Versuche sehr umfangreich sowie für die weitere Verwendung der im Projektrahmen entwickelten Konzepte elementar sind. Tabelle 8 zeigt die Übersicht der veröffentlichten Arbeiten.

Tabelle 8: Übersicht der bereits im Projektrahmen erfolgten Veröffentlichungen.

Typ	Details
Vorträge	Jarmatz, N., Augustin, W., Scholl, S., Optimization of fouling prediction by the targeted generation of experimental training data, ProcessNet Wärme- und Stoffübertragung 2022, Würzburg, Germany, 18.-20.07.2022 Jarmatz, N., Augustin, W., Scholl, S., Generation of experimental data for model training to optimize fouling prediction, Heat Exchanger Fouling and Cleaning 2022, Wagrain, Austria, 05.-10.06.2022 Jarmatz, N., Augustin, W., Yin, J., Mauermann, M. Projektkurzvorstellung FideLio, VDMA Technische Sitzung Nahrungsmittelmaschinen, Online-Veranstaltung, 18.10.2023 Yin, J., Jarmatz, N., Mauermann, M., Augustin, W., Scholl, S. Development of a new retrofittable concept for fouling detection using Machine Learning based on real data from dairy heat exchangers, Heat Exchanger Fouling and Cleaning 2024, Lisabon, Portugal, 21 – 25.04.2024

	<p>Augustin, W.: On the importance of monitoring fouling and cleaning in food production and the necessity of a holistic view, 13. MIV-Ideenbörse Forschung, Fulda, Germany, 13.-14.11.2024</p> <p>van Asselt, A.J., Schuten, H.J., Huellemeier, H.A., Augustin, W.: Is Cleaning In Place ready for the future? - a critical assessment, Fouling and Cleaning in Food Processing Conference FCFP2025, Dresden, Germany, 25.-27.03.2025</p>
Poster	<p>Jarmatz, N., Paul, L., Deseke, O., Augustin, W., Scholl, S., Jahrestreffen der DECHEMA-Fachgruppen Computational Fluid Dynamics and Heat and Mass Transfer, Sensor based detection of particulate deposition by temperature measurements, Frankfurt, Germany, 06.-08.03.2023</p>
Zeitschriftenbeiträge	<p>Jarmatz N., Augustin W., Scholl S. (2023) Comprehensive Parameter Screening for the Investigation of Particulate Fouling in Pipe Fittings. Chemie-Ingenieur-Technik 95/5 S. 708-716: https://doi.org/10.1002/cite.202200208</p> <p>Jarmatz, N., Augustin, W., Scholl, S., 2023. Generation of Experiential Data for Model Training to Optimize Fouling Prediction. Heat Mass Transf. https://doi.org/10.1007/s00231-023-03393-5</p> <p>Jarmatz, N., Augustin, W., Scholl, S., 2024. Characterization of artificial fouling layers by static and dynamic temperature measurements with clamp on sensors. Heat Mass Transf. Submitted.</p> <p>Yin, J., Jarmatz, N., Mauermann, M., Augustin, W., & Scholl, S. Predictive fouling detection in food production using machine learning models based on real data. Proceedings of the International Conference on Heat Exchanger Fouling and Cleaning - 2024 (Peer-reviewed), April 21–26, 2024, Lisbon, Portugal. Editors: H.U. Zettler and E.M. Ishiyama. ISBN: 978-0-9984188-3-4. Published online at www.heatexchanger-fouling.com, pp. 73–83. Accepted for Journal of Heat Transfer.</p>

6. Fazit und Ausblick

Bereits zu einem sehr frühen Zeitpunkt der Projektbearbeitung konnten alle Arbeitspakete begonnen bzw. einige bereits beendet werden. So wurden als Vorbereitung für das Anlernen der Modelle Anforderungs- bzw. Fragenkataloge erstellt und an die Industriepartner übergeben. Nach Absprache und eingehender Analyse der vorhandenen Datenlage wurden durch Firma 1 relevante Prozessdaten zur Verfügung gestellt. Nach sorgfältiger Auswertung der Prozessdaten von Firma 1 wurden die für das Fouling relevanten Sensoren ausgewählt und die Messwerte für das Modelltraining verwendet.

Eine umfassende Marktstudie für die Bewertung der Verfügbarkeit von kommerziellen Clamp-On-Sensoren wurde durchgeführt, welche zur Auswahl eines Clamp-On-Temperatursensors mit optimalen Eigenschaften für das Projekt führte. Die Sensoren konnten zusammen mit dem im Projektrahmen entwickelten IoT-System in Betrieb genommen

und an der Technikumsanlage im Labor des ICTV getestet werden. Das Gesamtsystem wurde charakterisiert und für den Einsatz an realen Produktionsanlagen vorbereitet. Sobald alle Details mit den entsprechenden Industriepartnern geklärt wurden, konnten die Clamp-On-Messsysteme im industriellen Umfeld eingesetzt werden. Die Ergebnisse, die im Rahmen des Projektes erarbeitet wurden, beleuchten ein breites Potential für die Anwendung solcher Clamp-on-Sensoren für diverse Betreiber und Problemstellungen. Wichtig ist, dass die Anlagenvoraussetzungen (Vorhandensein von Rohren, Temperaturbereich, evtl. notwendiger Explosionsschutz etc.) gegeben sind, um diese Sensoren verwenden zu können. Ferner sollte eine Kalibrierung im direkten Vergleich mit einem Inline-Sensor erfolgen, um die systemspezifische Abweichung quantifizieren und berücksichtigen zu können.

Die entwickelte Hardware wurde so als Demonstrator konzipiert, dass sie unter realen Produktionsbedingungen in Molkereien einfach eingesetzt werden kann. Es wurde eine mobile Messerfassung realisiert, die ohne bestehende Infrastruktur auskommt. Für das System wurde die Schutzklasse IP67 für das System angestrebt und eine einfache Konnektivität zu den ausgewählten Clamp-On Sensoren integriert. Die für das IoT entwickelten Software-Module ermöglichen es, sinnvolle Sensorwerte auszulesen und die Daten in einer Datenbank abzuspeichern sowie auf einen zentralen Server zu transferieren. Im nächsten Schritt sollte eine Methode zur Generierung dimensionsloser Kennzahlen als Service (Feature-Extractor) umgesetzt werden, die dann an den Predictor-Service weitergeleitet werden. Dieser Service ist für die Vorhersage des Foulingzustands zuständig und es wurde ein nach der Modelluntersuchung ausgewähltes Modell integriert. Zusätzlich ist eine Anpassung der Benutzerschnittstelle (GUI) erforderlich, die dem Bediener den aktuellen Foulingzustand anzeigt. Alle Softwarekomponenten und die Systemkonfiguration können mit Hilfe des Ansible-Playbooks schnell bereitgestellt werden.

Auch in Bezug auf das zu generierende Sensorkonzept wurden Fortschritte in Form von vergleichenden Untersuchungen an kritischen Anlagenbauteilen in Bezug auf Fouling durchgeführt. Im Rahmen der Untersuchungen wurden bekannte und demnach häufig als kritisch oder problematisch bewertete Anlagenkomponenten (Stutzen, Bogen, Düse, Diffusor, Ventil) genauer untersucht. Dadurch konnte gezeigt werden, wie sich die Foulingneigung der Komponenten im direkten Vergleich für das betrachtete Stoffsystem unterscheidet. Um in einem weiteren Schritt zu klären, welche Prozessparameter in welchem zeitlichen Intervall gemessen werden müssen, wurden zunächst die zeitlichen Verläufe des Ablagerungsverhaltens in realen Produktionsprozessen genauer untersucht.

Mit dem Abschluss der Sammlung eines Screening-Datensatzes an der Plattenwärmübertrager-Technikumsanlage konnten diese Daten für die weitere Verarbeitung vorbereitet werden, um sie schlussendlich für das Modelltraining zu verwenden. Aus den drei Datenquellen, Labor, Firma 1 und 2, wurden drei Deep-Learning-Modelle entwickelt und validiert. Mit Ausnahme des reinen LSTM-Modells zeigen die beiden anderen Modelle eine gute Vorhersagefähigkeit auf Basis der Datensätze. Erwartungsgemäß verhalten sich die Modelle basierend auf den Labordaten deutlich besser als auf den beiden anderen industriellen Datensätzen. Sie prognostizieren die Temperaturverläufe mit ausreichender Genauigkeit. Dies deutet darauf hin, dass die Modelle in der Lage sind, Daten

mit signifikanten Trends und geringem Rauschen nachzubilden. Bei Daten mit inkonsistenten Verläufen zeigen jedoch alle Modelle Schwächen. Folglich ist die Datenqualität eine wesentliche Voraussetzung für eine gute Modellleistung. Um eine höhere Vorhersagegenauigkeit zu erzielen, müssten mehr Daten gesammelt und die Modellarchitektur robuster gestaltet werden. Dazu wäre auch die Integration von aus der Dimensionsanalyse gewonnenen Kennzahlen hilfreich.

Grundsätzlich lässt sich sagen, dass jetzt ein apparatives und methodisches Konzept zur Verfügung steht, das die Foulingvorhersage ermöglicht. Die Anpassung auf die spezifische Anwendung ist naturgemäß erforderlich und hier im Rahmen der Beschreibung des Modelltrainings dargestellt.

7. Literaturangaben

- Ahmad, M.W., Reynolds, J., Rezgui, Y., 2018. Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of Cleaner Production* 203, 810–821.
<https://doi.org/10.1016/j.jclepro.2018.08.207>
- Aurelien, G., 2018. Neural Networks and Deep Learning. O'Reilly Media, Inc., Beijing, China.
- Bengio, Y., Courville, A., Vincent, P., 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1798–1828.
<https://doi.org/10.1109/TPAMI.2013.50>
- Bose, T., Meyer, F., 2003. Digital Signal and Image Processing | Guide books. John Wiley & Sons, Inc., USA.
- Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation* 9, 1735–1780.
<https://doi.org/10.1162/neco.1997.9.8.1735>
- Jarmatz, N., Augustin, W., Scholl, S., 2024. Characterization of artificial fouling layers by static and dynamic temperature measurements with clamp on sensors. *Heat and Mass Transfer* Submitted.
- Jarmatz, N., Augustin, W., Scholl, S., 2023. Comprehensive Parameter Screening for the Investigation of Particulate Fouling in Pipe Fittings. *Chemie-Ingenieur-Technik* 95, 708–716.
<https://doi.org/10.1002/cite.202200208>
- Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.
- Livieris, I.E., Pintelas, E., Pintelas, P., 2020. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput & Applic* 32, 17351–17360.
<https://doi.org/10.1007/s00521-020-04867-x>
- Lu, W., Li, J., Li, Y., Sun, A., Wang, J., 2020. A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity* 2020, 1–10.
<https://doi.org/10.1155/2020/6622927>
- Mahamat, A.A., Boukar, M.M., Leklou, N., Celino, A., Obianyo, I.I., Bih, N.L., Stanislas, T.T., Savastanos, H., 2024. Decision Tree Regression vs. Gradient Boosting Regressor Models for the Prediction of Hygroscopic Properties of Borassus Fruit Fiber. *Applied Sciences* 14, 7540.
<https://doi.org/10.3390/app14177540>
- O'Shea, K., Nash, R., 2015. An Introduction to Convolutional Neural Networks.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., pp. 8024–8035.
- Sharma, P., Malik, N., Akhtar, N., 2013. FEEDFORWARD NEURAL NETWORK: A Review. *International Journal of Advanced Research in Engineering and Applied Sciences (IJAREAS)* 2, 25–34.
- Sidorow, O., 2023. Experimentelle Generierung von Trainingsdaten für die Modellierung des Foulings in einem Plattenwärmeübertrager. ICTV, TU Braunschweig.
- Tibshirani, R., Friedman, J.H., Hastie, T., n.d. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Wibawa, A.P., Utama, A.B.P., Elmunsyah, H., Pujianto, U., Dwiyanto, F.A., Hernandez, L., 2022. Time-series analysis with smoothed Convolutional Neural Network. *J Big Data* 9, 44.
<https://doi.org/10.1186/s40537-022-00599-y>
- Williams, R.J., Zipser, D., 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* 1, 270–280.
<https://doi.org/10.1162/neco.1989.1.2.270>
- Xiong, Y., Liu, H., Gupta, S., Akin, B., Bender, G., Wang, Y., Kindermans, P.-J., Tan, M., Singh, V., Chen, B., 2021. MobileDets: Searching for Object Detection Architectures for Mobile Accelerators. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
<https://doi.org/10.1109/CVPR46437.2021.00382>
- Yang, B., Gong, Z.-J., Yang, W., 2017. Stock market index prediction using deep neural network ensemble, in: 2017 36th Chinese Control Conference (CCC). Presented at the 2017 36th Chinese Control

- Conference (CCC), IEEE, Dalian, China, pp. 3882–3887.
<https://doi.org/10.23919/ChiCC.2017.8027964>
- Zhang, A., Lipton, Z.C., Li, M., Smola, A.J., 2023. Dive into Deep Learning. Cambridge University Press.
- Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X., 2019. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems* 30, 3212–3232.
<https://doi.org/10.1109/TNNLS.2018.2876865>