



Wie KI bei der Lebensmittelqualität unterstützen kann

Die Herstellung von Kochschinken bringt einige Herausforderungen mit sich. Für die Produktion werden Fleischteile mithilfe von Salzlake in einem mehrstufigen Pökel- und Garprozess verarbeitet. Dabei kann es zu Qualitätsschwankungen kommen, die auf Strukturfehler im Fleisch zurückzuführen sind. Das Resultat: Die Salzlake wird nicht optimal aufgenommen. Ein auf historischen Daten trainiertes KI-Modell soll das Problem lösen.

Schlüsselwörter

Computational Intelligence, Künstliche Neuronale Netze, Parameteroptimierung, nachhaltige Lebensmittelproduktion, Kochschinken, KI-Prototyp

Tim Zeiser, M. Sc. ist wissenschaftlicher Mitarbeiter an der Hochschule Offenburg, Mitglied in der Forschungsgruppe Digital Supply Chain sowie beim Institute for Machine Learning and Analytics.

Prof. Dr.-Ing. Theo Lutz ist Professor für Wirtschaftsinformatik, insbesondere betriebliche Informationssysteme an der Hochschule Offenburg, Leiter der Forschungsgruppe Digital Supply Chain und stellv. Leiter des Institute for Machine Learning and Analytics.

Corinna Köters, M. Sc. ist wissenschaftliche Mitarbeiterin an der Hochschule Niederrhein, Fachbereich Oecotrophologie, sowie Doktorandin an der Universität Witten/Herdecke.

Prof. Dr.-Ing. Maik Schürmeyer ist Professor für Lebensmittellogistik an der Hochschule Niederrhein, Fachbereich Oecotrophologie.

Prof. Dr. Dr. Dr. habil. Alexander Prange ist Rektor und Professor für Lebensmittel-Mikrobiologie und Futtermittelsicherheit an der Fachhochschule Südwestfalen, Fachbereich Agrarwirtschaft und Apl. Professor für Mikrobiologie am Lehrstuhl für Virologie und Mikrobiologie an der Universität Witten/Herdecke.

Kontakt

tim.zeiser@hs-offenburg.de

www.hs-offenburg.de/dsc

DOI: 10.30844/I4SD.24.6.40

Parameteroptimierung für einen Lakeinjektor

Entwicklung einer KI-Pipeline an einem Beispiel aus der Fleischindustrie

Tim Zeiser und Theo Lutz, Hochschule Offenburg, Corinna Köters, Hochschule Niederrhein und Universität Witten/Herdecke, Maik Schürmeyer, Hochschule Niederrhein, Alexander Prange, Fachhochschule Südwestfalen und Universität Witten/Herdecke

Der Beitrag behandelt die Fragestellung, wie mittels Machine Learning die Einstellungen der Parameter eines Lakeinjektors in der Kochschinkenproduktion optimiert werden können, um prozessbedingte Schwankungen im Kochschinken zu reduzieren. Als zentraler Parameter wird das Lakegewicht identifiziert. Bei dieser Zielgröße handelt es sich um die Menge an Lake, die das Fleisch aufnimmt. Sie wird pro Produktionseinheit optimiert. Dazu wird mit einem kooperierenden Hersteller von Kochschinken ein Datensatz aus historischen Produktionsdaten vorverarbeitet. Anschließend werden verschiedene KI-Modelle trainiert, wobei sich ein XGBoost-Modell als beste Lösung herausgestellt hat. Dieses ermöglicht auf den Testdaten einen Root Mean Squared Error (RMSE) von 4,5 kg, was einer Verbesserung von 200 % im Vergleich zum aktuellen Ansatz der Parameterbestimmung entspricht. Nach Entwicklung der KI wird ein Prototyp in der Kochschinkenproduktion implementiert und evaluiert.

Gewebe auflockert. Hierdurch wird Muskeleiweiß freigesetzt, sodass ein späterer Zusammenhalt der Scheiben des Kochschinkens erzielt wird. Daraufhin werden die Fleischteilstücke in Formen gelegt oder in Därme gefüllt und unter Temperatureinwirkung gegart und schließlich einem Kühlprozess unterzogen, um somit eine mikrobiologische Sicherheit zu erlangen. Je nach End-

produkt kann zusätzlich ein optionales Räuchern des Schinkens erfolgen [2], [3].

Produktionsprozess Kochschinken

Nach den Leitsätzen für Fleisch und Fleischerzeugnisse wird die Bezeichnung Schinken, auch in Wortverbindungen, nur für Erzeugnisse von mindestens gehobener Qualität verwendet. Zur Herstellung von Schinken werden nur unzerkleinerte Fleischteilstücke wie Ober- und Unterschale, Nuss und/oder Hüfte eingesetzt [1].

Zur Produktion von Kochschinken werden die genannten Fleischteilstücke während der Nasspökung durch Muskelspritzverfahren bearbeitet. Dabei wird unter Einsatz von Injektoren mittels Druck Lake in das Muskelfleisch injiziert. Hierdurch findet eine gleichmäßige Diffusion der Lake statt. Das Pökeln hat unterschiedliche Wirkungen auf das Lebensmittel. Es hemmt den mikrobiologischen Verderb und hat eine farb- sowie aromabildende Wirkung auf das Produkt. Anschließend wird das mechanische Verfahren „Tumbeln“ durchgeführt, welches das

In diesem Prozess können Schwankungen der Produktqualität beispielsweise durch destrukтуриerte Stellen im Fleisch vorliegen, welche sich qualitätsmindernd auf die Fleischbeschaffenheit auswirken. Hierbei handelt es sich um Strukturfehler mit unterschiedlichen Ursachen. Im Kochschinken lassen sich zwei Typen feststellen: Die trockene, strohige sowie die nasse, zerfallende Destrukturierung. Laut [4] entstehen Destrukturierungen durch Veränderungen im Rohmaterial, wie z. B. durch eine eingeschränkte Funktionalität der Proteine. Dadurch ist das Wasserbindungsvermögen verringert und es kann nicht die optimale Menge an Lake aufgenommen werden. Ein Einreißen der Scheiben des Kochschinkens beim Schneiden kann eine Auswirkung destrukтуриert Stellen sein [4]. Basierend auf diesem Hintergrund ist festzustellen, dass eine gänzlich ideale Dosierung von Lake aufgrund natürlicher Schwankungen der Fleischteilstücke nicht erreicht werden kann.

Verfügbare Daten für das KI-Training

Die Herangehensweise für das Bereinigen der Daten sowie dem Training der KI-Modelle orientiert sich an CRISP-DM. Somit bildet es einen kontinuierlichen Kreislauf aus Business Understanding, Datenverständnis, Datenvorverarbeitung, Modellierung, Evaluation und Deployment [5]. Der vorliegende Datensatz wurde zuerst aus den Produktionssystemen des kooperierenden



Die ORCID-Identifizierungsnummern der Autoren dieses Beitrags sind einsehbar unter <https://doi.org/10.30844/I4SD.24.6.40>

Dies ist ein Open-Access-Artikel unter den Bedingungen der Creative Commons Attribution License, die die Nutzung, Verbreitung und Reproduktion in jedem Medium erlaubt, sofern das Originalwerk ordnungsgemäß zitiert wird.

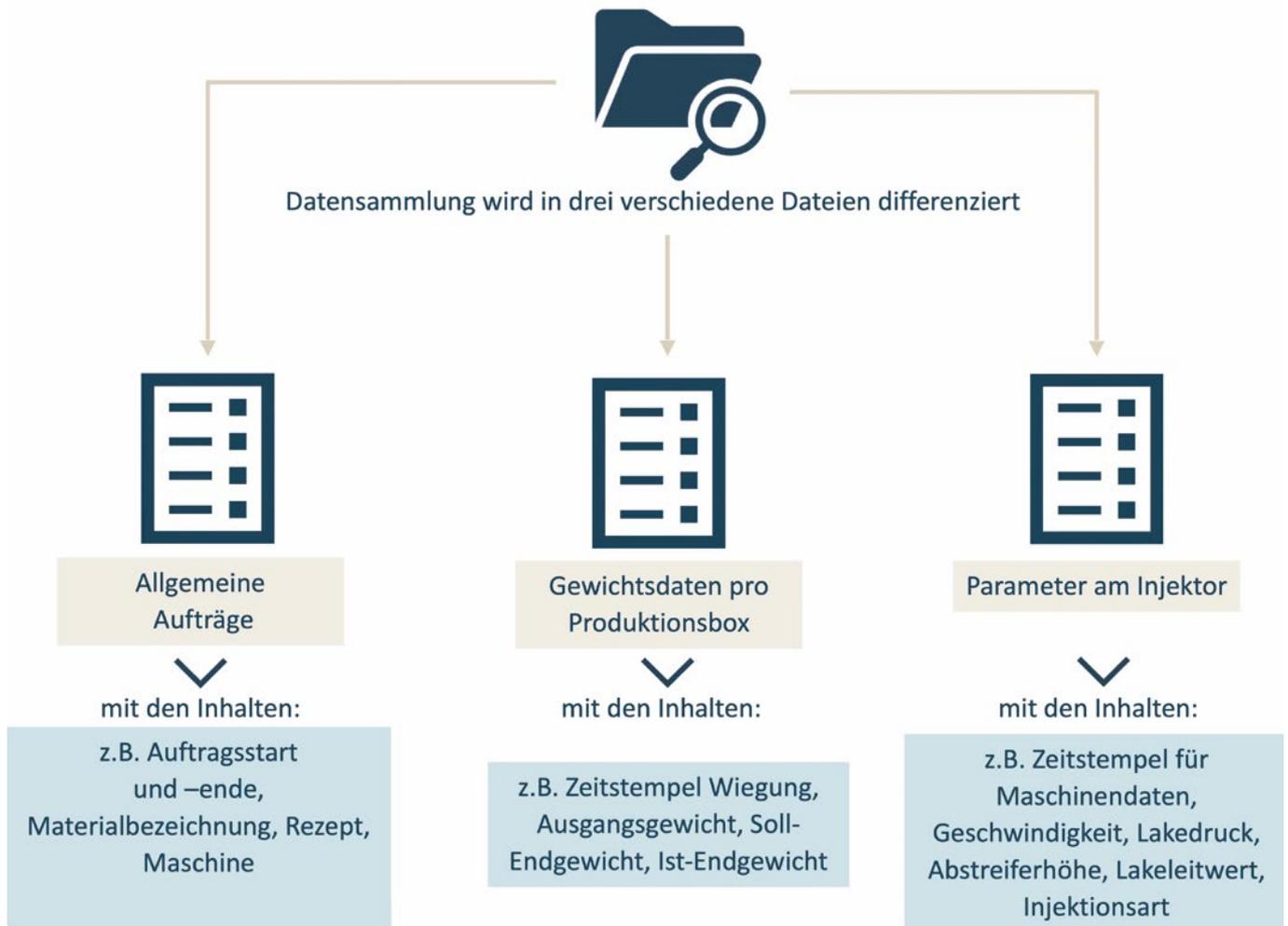


Bild 1: Dateien aus der Kochschinkenproduktion.

den Unternehmens im CSV-Format exportiert. Später wurden die Daten automatisiert über eine selbst entwickelte REST-Schnittstelle direkt in MinIO S3-Buckets gespeichert, um von dort aus die entsprechenden Verarbeitungsschritte für die Daten und das Training des KI-Modells anzustoßen. Der Datensatz enthält die Daten von Anfang des Jahres 2014 bis Ende des Jahres 2023 mit insgesamt ca. 225.000 Aufträgen, verteilt auf mehrere Maschinen.

Der Datensatz teilt sich in drei Dateien auf. Erstens für die Aufträge, zweitens für die Gewichtsdaten bezogen auf eine Produktionsbox sowie eine Datei für die eingestellten Parameter des Lakeinjektors. Die Dateien enthalten folgende Daten: Auftragsstart und -ende, Materialbezeichnung, Rezept, Ausgangsgewicht (in kg), Endgewicht (in kg), Zeitstempel für die Wiegung und für die Maschinendaten, Geschwindigkeit (in T/min), Lakedruck (in Bar; Ist und Soll), Abstreiferhöhe (in cm; Ist und Soll), Lakeleitwert (in mS/cm; Ist und Soll) und Injektionsart.

Datenvorverarbeitung

Damit die Daten für die KI-Modelle verwendet werden können, müssen diese zuerst vereint werden. Dies gilt zum einen für die Injektordaten, welche pro Tag und Maschine in jeweils einer Datei gespeichert werden. Nach dem Vereinen der Dateien werden etwaige Duplikate entfernt. Die Verbindung zwischen Auftrags- und Gewichtsdaten findet über die Auftragsnummer statt. Die Verbindung zu den Parametern des Injektors wird anschließend über den Zeitstempel der Parametereinstellung und Auftragsstart und -ende durchgeführt.

Die anschließende Bereinigung des Datensatzes wurde mittels Domänenwissen aus der Lebensmittelverarbeitung durchgeführt, um hier vor allem die Injektorparameter und die Gewichtsangaben in die entsprechenden, für den Produzenten üblichen Wertebereiche zu bereinigen. Zusätzlich wurden nicht relevante Einträge aus den Daten entfernt, welche durch produktionsbedingte Pausen wie Maschinenwartungen entstehen. Ausreißer werden anhand

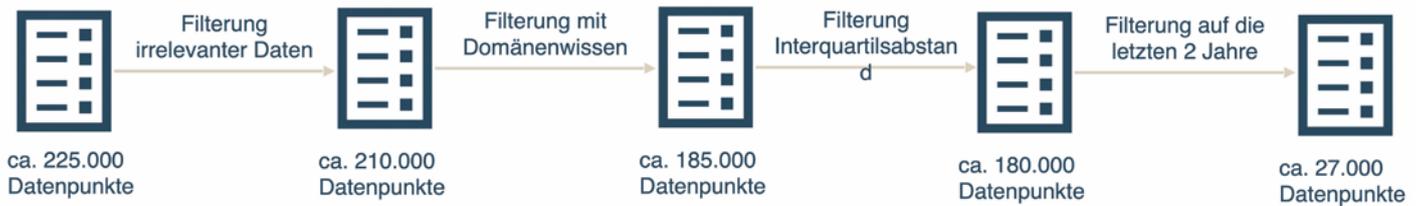


Bild 2: Durchführung der Datenbereinigung.

des Interquartilsabstands mit einem Threshold von 1,5 bereinigt [6]. Der Datensatz enthält nach Bereinigung ca. 180 000 Einträge. Anschließend wurde eine Filterung auf die letzten zwei Jahre durchgeführt, da sich ab diesem Zeitraum die Rezepte aufgrund von anderen Anforderungen der Kunden und die Vorgaben des Gesetzgebers geändert haben. Nach den Filterungen bleiben noch ca. 27 000 Datenpunkte übrig, die für Training und Testing der KI-Modelle verwendet werden können. **Bild 2** zeigt hierbei die angewendeten Schritte. Zusätzlich wurden noch weitere Features hinzugefügt, wie das Gewicht der Lake und die Sättigungsrate des Fleisches innerhalb einer Box.

Damit das KI-Modell anschließend mit den kategorischen Werten wie der Materialbezeichnung, dem Rezept oder der Injektionsart arbeiten kann, werden diese kodiert und in numerische Werte mittels Pandas umgewandelt. Da für die verschiedenen Materialien unterschiedliche Schreibweisen existieren, wurde ein Mapping erstellt, um diese zu vereinheitlichen. Der finale Datensatz, welcher auch für das Training der KI-Modelle verwendet wird, enthält folgende Features: Materialbezeichnung, Rezept, Ausgangsgewicht (in kg), Geschwindigkeit (in T/min), Ist-Lakedruck (in bar), Ist-Abstreiferhöhe (in cm) und Ist-Lakeleitwert (in mS/cm).

Ansatz für die Parameteroptimierung

Die Vorgehensweise steht somit im Gegensatz zu den in der Literatur verwendeten Ansätzen, dort werden in der Regel Klassifikationen verwendet [6]. Diese Herangehensweise ist in diesem Fall nicht zielführend, da das Lakegewicht durch kontinuierliche Werte gegeben ist und somit nicht klassifiziert werden kann. Ebenfalls verwendet werden könnte ein Reinforcement-Learning, wie es in [7] zum Einsatz kommt. Die Autoren haben sich aber aufgrund der geringen Datenlage dagegen entschieden.

Der gewählte Ansatz sieht vor, eine Regression anzuwenden, um das Lakegewicht pro Produktionseinheit vorherzusagen. Dies wird über ein KI-Modell realisiert, welches mit der Materialbezeichnung, dem Rezept, dem Ausgangsgewicht, der Abstreiferhöhe, dem Lakeleitwert, der Injektionsart und der Box-Nummer trainiert wird, um zu lernen,

welche Parameter zu welchem Lakegewicht führen. Dieses Modell wird dann genutzt, um die Parameteroptimierung durchzuführen. Zu optimierende Parameter sind – produktionsbedingt – die Geschwindigkeit und der Lakedruck. Aus diesen werden alle möglichen Kombinationen gebildet, welche durch Domänenwissen aus der Produktion eingegrenzt werden. Mittels Brute Force werden diese Kombinationen zusammen mit den restlichen Werten, welche aus dem jeweiligen Auftrag kommen, in das KI-Modell gegeben. Hierbei wird evaluiert, welche Kombination die geringste Differenz zum zu erreichenden Lakegewicht hat. Dies wird den Mitarbeitenden in der Kochschinkenproduktion als Ergebnis ausgegeben. Eine alternative Möglichkeit wäre es, hier die Parametersuche durch eine Bayes'sche Optimierung zu ersetzen, um somit einen Vorteil in der Zeit und der Performance zu erhalten.

Auswahl der Evaluationsmetrik

Damit das KI-Modell die entsprechenden Zusammenhänge lernen kann, werden Aufträge, welche eine höhere Abweichung zum Zielgewicht besitzen und damit die Produktqualität verändern, herausgefiltert. Um den richtigen Grenzwert zu finden, werden verschiedene Filterungen angewendet. Damit die einzelnen Modelle evaluiert werden können, wird aus den vorhandenen Daten ebenfalls die Soll-/Ist-Abweichung des Lakegewichts berechnet, die mit e bezeichnet wird. Als Kennzahl wird der RMSE verwendet, der sich wie folgt berechnet:

$$\sqrt{\text{mean}(e_i^2)}$$

Damit ist sichergestellt, dass die Ergebnisse des KI-Modells innerhalb der Produktion direkt interpretiert und verglichen werden können. Ebenfalls eignet sich der RMSE für den Vergleich von verschiedenen Ansätzen [7].

Auswahl von KI-Ansätzen

Für die Benchmark der Leistungsfähigkeit der KI-Modelle wird in die Auswahl ebenfalls ein einfaches lineares Regressionsmodell aufgenommen. Die anderen ausgewählten Modelle sind ein Random Forest, eine Support Vector Regression, alle aus der Scikit-learn Bibliothek [8], zwei

Modell/Ist-Abweichung	20 kg	15 kg	10 kg	5 kg
Lineare Regression	7,3	7,3	7,4	7,7
Support Vector Regression	7,5	7,4	7,6	7,9
Random Forest	6,0	5,6	5,9	6,6
XGBoost	6,2	4,5	5,3	6,5
LightGBM	6,1	5,0	5,6	6,7
Künstliches neuronales Netz	7,6	7,3	8,4	7,2

Bild 3: Ausgewählte KI-Modelle mit korrespondierenden RMSE.

Gradient Boosting Tree Modelle (XGBoost [9] und LightGBM [10]) sowie ein künstliches neuronales Netz, welches mit Keras [11] und TensorFlow [12] erstellt wurde. Vor dem Training der einzelnen Modelle wurden die Daten in Trainings- und Testdaten im Verhältnis 9:1 geteilt. Somit wurden für das Training ca. 22 000 Datenpunkte verwendet.

Nach dem Training wurde eine Hyperparameteroptimierung der Modelle durchgeführt. In **Bild 3** angegeben sind jeweils die Fehlerraten (RMSE) für die Testdaten, ohne Filterung auf eine Soll-/ Ist-Abweichung mit den optimierten Modellen, damit eine Vergleichbarkeit der Modelle und verschiedenen Filterungen gewährleistet wird. Auf den Rohdaten beträgt dieser 15 kg.

Die Ergebnisse der einzelnen KI-Modelle auf den Testdaten zeigen auf, dass bei einer zu starken Filterung auf eine geringe Abweichung zwischen Soll- und Ist-Lakegewicht die Abstraktionsfähigkeit der einzelnen Modelle auf neue, noch nicht gesehene Daten deutlich sinkt. Das Modell von XGBoost zeigt hierbei die beste Performance bei einer maximalen Soll-/ Ist-Abweichung von 15 kg. Die Fehlerrate liegt hierbei im Durchschnitt bei 4,5 kg und beträgt somit weniger als ein Drittel des mittleren Fehlers, welcher sich in den Trainingsdaten befindet.

Vom KI-Modell in die Produktion

Nachdem die Auswahl für das KI-Modell getroffen wurde, wird dieses nun in die Kochschinkenproduktion implementiert. Damit die Mitarbeitenden das KI-Modell nutzen können, erfolgt die Umsetzung über eine Web-GUI. Dieses wurde mit dem Framework Streamlit umgesetzt, das sich gut für die schnelle Umsetzung von Prototypen eignet. Über die GUI können die einzelnen Werte des Auftrages über Dropdown-Felder oder Schieberegler durch Mitar-

beitende der Produktion eingegeben werden. Diese werden anschließend über eine REST-Schnittstelle an einen Server übergeben, welcher die Berechnung der idealen Parameter übernimmt. Hierüber kann diese Funktionalität auch in andere Programme übernommen und integriert werden. Um parallel zur Nutzung eine Evaluation durchführen zu können, wurde dies direkt in das Web-GUI integriert. Die so gesammelten Daten werden über die identische REST-Schnittstelle in MinIO Buckets gespeichert und können anschließend ausgewertet werden. Beide

Bild 4: Web-GUI für die Kochschinkenproduktion.

The screenshot shows a web interface for injector parameter optimization. It includes the following elements:

- Title:** Injektor Parameteroptimierung
- Material:** A dropdown menu currently showing 'Unterschale'.
- Rezept:** A dropdown menu currently showing '1'.
- Ausgangsgewicht in kg:** A slider ranging from 550 to 700, with a red dot at 630.
- Sättigungsrate %:** A slider ranging from 5.00 to 20.00, with a red dot at 10.50.
- Abstreiferhöhe in mm:** A slider ranging from 0 to 30, with a red dot at 20.
- Soll Lakeleitwert in mS/cm:** A slider ranging from 130 to 250, with a red dot at 150.
- Action:** A button labeled 'Ideale Parameter berechnen'.

Teile der Lösung werden mittels Docker [13] als Container bereitgestellt und bieten somit eine gute Flexibilität hinsichtlich der Laufzeitumgebung und der Integration in die Produktion. Zum Zeitpunkt der Veröffentlichung ist die Evaluation noch nicht abgeschlossen.

Zukünftige Entwicklungen

Die entwickelten KI-Modelle tragen nach Implementierung in der Kochschinkenproduktion dazu bei, die Maschinenparameter am Lakeinjektor so einzustellen, dass eine zielgerichtete Dosierung von Lake in dem entsprechenden Produktionsprozess durchgeführt werden kann. Die Ergebnisse der einzelnen KI-Modelle zeigen, dass die Genauigkeit im Vergleich zum aktuellen Vorgehen in der Kochschinkenproduktion (deutlich) erhöht werden kann. Trotz der geringen Datenmenge von ca. 22 000 Datenpunkten konnte ein zielgerichtetes Modell geschaffen werden.

Mit der derzeit laufenden Datensammlung wird das KI-Modell weiterhin mit neuen Datensätzen trainiert und kann somit immer bessere Ergebnisse erzielen. Die aktuell laufende Evaluation soll zum einen die Funktionsfähigkeit und Nutzerfreundlichkeit des Ansatzes aufzeigen. Außerdem wird evaluiert, inwieweit sich nach Einsatz der KI-Modelle der Anteil von prozessbedingten Destrukturierungen im Kochschinken reduziert. Ebenso ist es möglich, die Optimierung dahingehend auszuweiten, dass auch weitere Parameter, wie etwa der Lakeleitwert, integriert werden. Hierdurch könnte auch die Dosierung selbst optimiert und eine ressourcenschonende Produktionsweise angestrebt werden.

Zukünftig ist außerdem denkbar, die hier beschriebenen und andere KI-Ansätze in weitere Prozessschritte der Kochschinkenproduktion zu übertragen. Als Beispiel dafür kann die KI-gestützte Sichtprüfung der angelieferten Fleischteilstücke und das Erkennen von qualitätsmindernden Eigenschaften wie destrukturierter Stellen in Fleischteilstücken genannt werden.

Dieser Beitrag entstand im Rahmen des Projekts „Künstliche Intelligenz für nachhaltige Lebensmittelqualität in Lieferketten (KINLI)“ das von dem Bundesministerium für Ernährung und Landwirtschaft (BMEL) unter dem Kennzeichen 28DK124D20 gefördert wird.

Literatur

- [1] BMEL: Leitsätze für Fleisch und Fleischerzeugnisse. URL: https://www.bmel.de/SharedDocs/Downloads/DE/_Ernaehrung/Lebensmittel-Kennzeichnung/LeitsaetzeFleisch.pdf?__blob=publicationFile&v=16, Abrufdatum 06.08.2024.
- [2] Brombach C. u. a.: Fleischerei heute. 6. Auflage. Hamburg 2023.
- [3] Krämer, J.; Prange, A.: Lebensmittel-Mikrobiologie: 48 Tabellen, 8. Auflage. In: UTB Lebensmittel- und Ernährungswissenschaften, Biologie, Nr. 1421. Stuttgart 2023.
- [4] Müller Richli, M.; Bee, G.; Stoffers, H.; Scheeder, M.: Strukturfehlern in Schweineschinken auf der Spur. URL: https://www.agroscope.admin.ch/agroscope/de/home/themen/wirtschaft-technik/betriebswirtschaft/publikationen/_jcr_content/par/externalcontent.bitexternalcontent.exturl.html, , Abrufdatum 06.08.2024.aHR0cHM6Ly9pcmEuYWdyb3Njb3BILmNoL2VuLVVTL1BhZ2UvUH/VibGlrYXRpb24vSW5kZXgvMjk5OTg=.html
- [5] Wirth, R.; Hipp, J.: CRISP-DM: Towards a standard process model for data mining. In: Proc. 4th Int. Conf. Pract. Appl. Knowl. Discov. Data Min (Januar 2000).
- [6] Ross, S. M.: Introductory statistics, 3. Auflage. Burlington, MA 2010.
- [7] Hyndman, R. J.; Koehler, A. B.: Another look at measures of forecast accuracy. In: Int. J. Forecast. 22 (2006) Nr. 4, S. 679–688. DOI: 10.1016/j.ijforecast.2006.03.001.
- [8] Pedregosa, F. u. a.: Scikit-learn: Machine learning in Python. In: J. Mach. Learn. Res. 12 (2011), S. 2825–2830.
- [9] Chen, T.; Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in KDD '16, S. 785–794. New York, NY 2016. DOI: 10.1145/2939672.2939785.
- [10] Ke, G. u. a.: Lightgbm: A highly efficient gradient boosting decision tree. In: Adv. Neural Inf. Process. Syst. 30 (2017), S. 3146–3154.
- [11] Chollet, F. u. a.: Keras. URL: <https://github.com/fchollet/keras>, Abrufdatum 06.08.2024.
- [12] Abadi, M. u. a.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. URL: <https://www.tensorflow.org/>, Abrufdatum 06.08.2024.
- [13] Merkel, D.: Docker: lightweight linux containers for consistent development and deployment. In: Linux J. (2014) 239, S. 2.