DOKUMENTATION

02/2024

Abschlussbericht

Faktendaten für das Informationssystem ChemInfo

Entwicklung von Algorithmen zur quantitativen und qualitativen Aufbereitung von verbalisierten Faktendaten für das Informationssystem ChemInfo

von:
Verena Heike, Norman Eckert, Alexander Erbse, Tobias Turke, Matthias Merkle
SoftwareOne Deutschland GmbH, Leipzig

Michael Weinert, Jürgen Hürttlen
Fraunhofer Institut für Chemische Technologie, Pfinztal

Herausgeber:
Umweltbundesamt

Für Mensch & Umwelt

Umwelt Bundesamt

DOKUMENTATION 02/2024

AA-Forschungsplan des Auswärtigen Amtes

Forschungskennzahl 3720 65 489 0
FB001338

Abschlussbericht

Faktendaten für das Informationssystem ChemInfo

Entwicklung von Algorithmen zur quantitativen und qualitativen Aufbereitung von verbalisierten Faktendaten für das Informationssystem ChemInfo

von

Verena Heike, Norman Eckert, Alexander Erbse, Tobias Turke, Matthias Merkle
SoftwareOne Deutschland GmbH, Leipzig

Michael Weinert, Jürgen Hürttlen
Fraunhofer Institut für Chemische Technologie, Pfinztal

Im Auftrag des Umweltbundesamtes

Impressum

Herausgeber
Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

Durchführung der Studie:
SoftwareOne Deutschland GmbH
Blochstraße 1
04329 Leipzig

Fraunhofer Institut für Chemische Technologie
Jospeh-von-Fraunhofer-Straße 7
76327 Pfinztal

Abschlussdatum:
September 2023

Redaktion:
Fachgebiet IV 2.1 Informationssysteme Chemikaliensicherheit
Dr. Marco Büchler

Publikationen als pdf:
http://www.umweltbundesamt.de/publikationen

ISSN 2199-6571

Dessau-Roßlau, Mai 2024

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen*Autoren.

Kurzbeschreibung: Faktendaten für das Informationssystem ChemInfo

Die Gefahrstoffschnellauskunft (GSA), ein Teildatenbestand des ChemInfo-Systems, soll im Rahmen eines Projekts mit Hilfe von künstlicher Intelligenz verbessert werden. Ziel ist es, den Grad der Befüllung und den Informationsgehalt zu steigern und vereinfachte Gefahren- und Maßnahmentexte zu erarbeiten. Die Aufgaben umfassen die Analyse der vorhandenen Daten, die Evaluation und Umsetzung von Deep Learning-Modellen und die Implementierung von Algorithmen zur Verbalisierung der stoffbezogenen Faktendaten.

Zur Ergänzung von Stoffeigenschaften, welche im ChemInfo Datenbestand nicht vorliegen, wurde im Rahmen der Recherche nach einer Machine-Learning Lösung das Chemprop Modell identifiziert.

Es handelt sich hierbei um ein vortrainiertes Modell, welches mittels Transfer-Learning zum Zweck der Vorhersage von Stoffmerkmalen im ChemInfo Datensatz angepasst werden kann. Im Projekt wurden die technische Implementierung von Modelltraining, Evaluation sowie Ansätze zur Optimierung durchgeführt und in diesem Bericht beschrieben.

Um eine weitreichendere Anwendung des Chemprop-Modells auf die ChemInfo Stoffe und Merkmale zu ermöglichen, bedarf es einiger essenzieller Anpassungen am Datenbestand.

Wie in Kapitel 3.3 aufgezeigt, ist es nötig, folgende Maßnahmen durchzuführen:

- Clustern bzw. Zusammenführen von Werteräumen, welche durch unterschiedliche Verwendung von Beschreibungen entstanden sind
- Standardisierung von Nominalen, Ordinalen und Binären Merkmalen ausbauen, insofern möglich
- Bereinigung von Doppelaussagen, widersprüchlichen Beschreibungen, Falschwerten
- Erhöhung von Datenmengen (Validierungspartitionen): starke Unterrepräsentation von bestimmten Merkmalen vermeiden

Innerhalb des Projektes wurden mögliche Methoden aufgezeigt, welche die genannten Vorbedingungen verbessern bzw. zu einem konsistenteren Informationsgehalt beitragen.

Speziell durch die Informationsextraktion aus Freitextfeldern mit dem Fokus auf inhaltliche & strukturelle Analyse lassen sich einige Potentiale erkennen. Hierbei sind sowohl implizite Beschreibungen für andere Merkmalsfelder zu erkennen sowie die Verbesserung von Datenkonsolidierung.

Genauere Informationen dazu sind beispielhaft in Kapitel 4.1 dargestellt.

Die Erstellung der verbalisierten Faktendaten auf Basis der im ChemInfo-System vorherrschenden Merkmale ist durch die Anwendung des in Kapitel 5 dargelegten Regelwerks zu erreichen.

Grundlegend können Informationen aus Merkmalsfeldern zur Generierung von Einsatzhinweisen verwendet werden. Dabei gibt es Informationen, wie z. B. chemisch-physikalische Kenngrößen, welche direkt übernommen werden können und wiederum Andere welche durch Anwendung von Regeln in Einsatzhinweise überführt werden können. Außerdem ist es teilweise möglich fehlende Informationen aus anderen Merkmalen abzuleiten.

Für die Anwendung mit Hinblick auf eine umfangreiche Abdeckung der auf den Datenblättern dargestellten Informationen bedarf es auch in diesem Zusammenhang entsprechender Anpassungen in den genannten Bereichen der Datenstandardisierung, Erhöhung der

Datenmenge und Bereinigung. Des Weiteren sollte in Betracht gezogen werden, dass Änderungen an Merkmalsfeldern eine kontinuierliche Anpassung der implementierten Regeln zur Datenblatterstellung erfordern.

Abstract: Fact Data for the ChemInfo Information System

The Hazardous Substance Quick Information (GSA), a subset of the ChemInfo system, is intended to be improved using artificial intelligence as part of a project. The goal is to increase the level of data filling and information content and to develop simplified hazard and action texts. Tasks include the analysis of existing data, the evaluation and implementation of deep learning models, and the implementation of algorithms for verbalizing substance-related factual data.

To supplement substance properties that are not available in the ChemInfo dataset, the Chemprop model was identified during the search for a machine learning solution.

This is a pre-trained model that can be adapted by means of transfer learning for the purpose of predicting substance characteristics in the ChemInfo data set. Technical implementation of model training, evaluation, and approaches to optimization were performed in the project and described in this report.

To enable a more extensive application of the Chemprop model to ChemInfo substances and features, some essential adaptations to the dataset are required.

As pointed out in section 3.3, it is necessary to perform the following:

- Clustering or merging of value spaces, which have been created by different use of descriptions.
- Expand standardization of nominals, ordinals, and binary features, insofar as it is possible
- Cleanup of duplicate statements, contradictory descriptions, false values
- Increase of data volumes (validation partitions): avoid strong underrepresentation of certain features.

Within the project, possible methods were pointed out that improve the aforementioned preconditions or contribute to a more consistent information content.

Especially by extracting information from free text fields with a focus on content & structural analysis, some potentials can be identified. Implicit descriptions for other characteristic fields can be recognized as well as the improvement of data consolidation.

More detailed information on this is presented as an example in chapter 4.1.

The creation of the verbalized fact data on the basis of the characteristics prevailing in the ChemInfo system is to be achieved by the application of the set of rules presented in chapter 5.

Basically, information from feature fields can be used to generate deployment hints. There are pieces of information, such as chemical-physical characteristics, that can be directly adopted, while others can be transformed into deployment hints through the application of rules. Additionally, it is partially possible to deduce missing information from other features..

For the application with regard to a comprehensive coverage of the information presented on the data sheets, corresponding adjustments in the mentioned areas of data standardization, increase of data volume and cleansing are also required in this context. Furthermore, it should be taken into consideration that changes to characteristic fields require continuous adaptation of the implemented rules for data sheet creation.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	Erläuterung
bspw.	beispielsweise
bzw.	beziehungsweise
CAS-Nummer	Chemical Abstract Service Registry Number
etc.	et cetera
ETL	Extract Transform Load
f.	folgende
FDCI	Faktendaten ChemInfo
FDCI-UBA	Projektkürzel
Fraunhofer ICT	Institut für Chemische Technologie
ggf.	Gegebenenfalls
i. d. R.	in der Regel
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
MIT	Massachusetts Institute of Technology
o. ä.	oder ähnliches
PID	Photoionisationsdetektor
REG	Regular Expression
s.	siehe
s. u.	siehe unten
SMARTS	SMILES Arbitrary Target Specification
SMILES	Simplified Molecular Input Line Entry Specification
SQL	Structured Query Language
SWO	SoftwareOne
UBA	Umwetlbundesamt
z. B.	zum Beispiel

Zusammenfassung

1 Vorbemerkungen

1.1 Zielgruppe

Dieses Dokument ist für die Mitglieder des Projekts "Faktendaten für das Informationssystem ChemInfo" erstellt und dokumentiert die wissenschaftlichen Ansätze, Arbeiten und Ergebnisse des Forschungsprojekts.

1.2 Einsatzbereich

Dieses Dokument ist ausschließlich für den internen Gebrauch im Projekt FDCI-UBA und mögliche Folgeprojekte mit dem Umweltbundesamt bestimmt. Jegliche Vervielfältigung, Speicherung, Umformatierung, Übertragung oder Weitergabe in elektronischer oder physischer Form bedarf der vorherigen Genehmigung der SWO.

1.3 Vertraulichkeit

Das vorliegende Dokument unterliegt den Bestimmungen zur Behandlung von schutzbedürftigen Dokumenten. Personen, die nicht zu den Zugangsberechtigten gehören, dürfen keine Informationen über die Existenz und den Inhalt dieses Dokuments erhalten.

1.4 Verbindlichkeit

Die im Dokument genannten Richtlinien zur Projektarbeit sind für alle Projektmitglieder verpflichtend und ihre Einhaltung wird von der Projektleitung kontrolliert.

2 Problemstellung und Ziel

Die Gefahrstoffschnellauskunft (GSA), ein Teildatenbestand des ChemInfo-Systems, soll im Rahmen eines Projekts mit Hilfe von künstlicher Intelligenz verbessert werden. Ziel ist es, den Grad der Befüllung und den Informationsgehalt zu steigern und vereinfachte Gefahren- und Maßnahmentexte zu erarbeiten. Die Aufgaben umfassen die Analyse der vorhandenen Daten, die Evaluation und Umsetzung von Deep Learning-Modellen und die Implementierung von Algorithmen zur Verbalisierung der stoffbezogenen Faktendaten. Der Bericht deckt den Projektzeitraum vom 02.05.2022 bis 29.05.2023 ab.

3 Datenanalyse und Datenvorverarbeitung

3.1 Zielstellung

Das Ziel der Datenvorverarbeitung ist es, die Datenstruktur des ChemInfo Datenbestands zu verstehen und in eine vereinfachte tabellarische Form zu überführen. Dies ermöglicht die Untersuchung von Zusammenhängen, dem Befüllungsgrad oder der Werteverteilung. Auf dieser Basis können anschließend Methoden zur Ergänzung fehlender Informationen und zur Generierung von Datenblättern entwickelt und implementiert werden.

3.2 Datenvorverarbeitung

3.2.1 Aufbau ETL Prozess

Eine ETL-Pipeline wurde in der Azure Cloud implementiert, um die hierarchische Datenstruktur des JSON-Exports in eine tabellarische Form zu überführen, die besser in einer SQL-Datenbank verarbeitet werden kann. Die Transformationsschritte wurden mit Python, PySpark und SQL in Azure Databricks durchgeführt und durch Azure Data Factory Pipelines automatisiert. Die Pipeline umfasst Prozesse wie Data Flattening, Additional Substance Names, Feature Collection, Data Pipeline, Data Mining, Data Analysis und die Verarbeitung von Strukturdaten. Nach der Aufbereitung der Daten durch ETL-Prozesse werden diese analytisch verwendet.

3.2.2 Kategorisierung von Merkmalsfeldern nach Relevanz

Die Merkmale zur Stoffbeschreibung wurden hinsichtlich ihrer Relevanz für die Datenblattanzeige und zur Ableitung von Eigenschaften bewertet, um den Datenumfang zu reduzieren. Merkmale mit geringer oder keiner Bedeutung für die Anwendungsfälle des Umweltbundesamtes wurden als nicht relevant eingestuft. Unklare Merkmale wurden gesondert gekennzeichnet. Es wurde zwischen ableitbaren und nicht ableitbaren Merkmalen unterschieden. Im Laufe des Projekts wurde festgestellt, dass eine Unterteilung in wahrscheinlich relevante und nicht relevante Daten ausreichend ist. Nur Daten Merkmalsdaten, welche als relevant kategorisiert gelten, wurden im weiteren Prozess verwendet.

3.2.3 Verwendung von Strukturdaten (MOL-Dateien) zur Erzeugung von SMILES Codes und Zuordnung funktioneller Gruppen

Die Untersuchung der Verfügbarkeit von Strukturinformationen ergab, dass eine MOL-Datei für etwa 37.000 von insgesamt 49.357 Einzelinhaltsstoffen vorliegt. Mit Hilfe der MOL-Dateien und der Python Bibliothek RDKit wurden SMILES Codes erzeugt, um Strukturinformationen für Machine-Learning Modelle zu nutzen. Durch die Analyse dieser Codes konnten Moleküleigenschaften identifiziert und den untersuchten Stoffen zugeordnet werden. Im weiteren Projektverlauf wurden nur SMILES Codes und Informationen über das Vorkommen funktioneller Gruppen benötigt. Mit Hilfe von SMARTS Codes konnten funktionelle Gruppen in SMILES identifiziert werden. Nach der Identifikation dieser SMARTS in den verfügbaren SMILES wurden die ermittelten funktionellen Gruppen auf Korrektheit geprüft. Einige Probleme wurden identifiziert und entsprechende Maßnahmen durchgeführt.

3.2.3.1 Ermittlung fehlender Strukturinformationen (SMILES) mittels Web-Scraping

Um die Verfügbarkeit von MOL-Dateien zu erhöhen, wurde geprüft, ob zusätzliche Strukturdaten (SMILES Codes) durch Nutzung externer Quellen angereichert werden können. Nach Berücksichtigung lizenzrechtlicher Aspekte wurde eine zweite Quelle verwendet, um zusätzliche SMILES Codes in die Azure SQL Datenbank zu importieren. Durch die Verwendung der externen Quellen konnten ca. 5.700 zusätzliche SMILES Codes befüllt werden. Diese wurden jedoch nur in der Analyse zur Auswahl sinnvoller Steuerungsparameter verwendet und nicht zum Training der finalen Chemprop-Modelle oder zum Auffüllen von ChemInfo Feldern.

3.3 Analyse des Datenbestands

Die Analyse des Datenbestands im ChemInfo-System zeigt eine fragmentarische Befüllung der Merkmale und eine geringe Standardisierung der Eingabewerte. Nur 1,45% der relevanten Merkmale sind ausreichend befüllt und nur 1,09% erfüllen die Bedingungen des Leistungsangebots von maximal 25% Lücken. Die Analyse der Werteräume zeigt außerdem eine Vermischung von Informationen, Doppelaussagen, inkonsistente Umsetzung messbarer Eigenschaften und Mehrdeutigkeit in den Merkmalen. Die meisten Merkmale erlauben hauptsächlich Freitextfelder zur Beschreibung.

3.4 Zusammenfassung der Analyse des Datenbestands

Die Voranalyse der Daten im ChemInfo-System offenbart Probleme wie lückenhafte Merkmale und fehlende Standardisierung der Eingabewerte. Als Lösung wurden alternative Ansätze entwickelt, die sich auf vorhandene Strukturinformationen (MOL-Dateien) stützen. Auf dieser Basis soll eine Evaluierung eines Deep Learning-Modells unter Verwendung von SMILES Codes erfolgen. Rückblickend umfassen die wichtigsten Schritte dieses Abschnittes den Aufbau einer ETL-Pipeline zur Datenaufbereitung, die Analyse der Strukturinformationen und die Analyse der Datenverfügbarkeit und Werteverteilung. Diese Schritte bilden die Grundlage für Methoden zur Verbesserung des Befüllungsgrades.

4 Methoden zur Verbesserung des Befüllungsgrads

4.1 Informationsextraktion aus Freitextfeldern mittels Textmining Methoden

Im Rahmen einer Potenzialanalyse wurden Textmining-Methoden zur Informationsextraktion aus Freitextfeldern untersucht, um den Befüllungsgrad von Stoffmerkmalen zu erhöhen. ChemInfo-Felder mit Freitextcharakter und umfangreichem Inhalt wurden ausgewählt und auf relevante Informationen für andere Merkmale hin untersucht. Prototypische Methoden zur Extraktion relevanter Inhalte wurden implementiert und versucht, diese den existierenden Werten in inhaltlich äquivalenten Zielfeldern zuzuordnen.

4.1.1 Auswahl der Freitextfelder zur Informationsextraktion

Die Auswahl der Freitextfelder zur Informationsextraktion im ChemInfo-System basierte auf einer Analyse der Befüllung und Werteverteilung der Felder. Nach Anwendung verschiedener Filter verblieben etwa 350 Felder, aus denen eine Stichprobe von 27 Feldern gezogen wurde. Nach einer manuellen Prüfung der Feldinhalte wurden acht Felder als vielversprechend für die Anwendung von Textmining-Methoden ausgewählt. Diese Felder wurden im weiteren Projektverlauf nach Rücksprache mit dem UBA auf vier Felder reduziert, für die eine vollständige Umsetzung der Informationsextraktion mittels Textmining durchgeführt wurde.

4.1.2 Umsetzung der Informationsextraktion

Die Informationsextraktion wurde in folgenden Schritten umgesetzt, welche für die vier ausgewählten Merkmalsfelder identisch angewendet wurden.

4.1.2.1 Allgemeine Textbereinigung

Die Qualität der Textinformationen wird zu Beginn geprüft, um Probleme wie Groß- & Kleinschreibung, Rechtschreibung, Wortabkürzungen, unterschiedliche Sprachen und fehlerhafte Symboliken zu identifizieren. Diese Probleme werden dann durch verschiedene Transformations- und Bereinigungsmethoden gelöst, einschließlich Texttransformation in Kleinschreibung, Inklusion von Fehlwerten, Auflösung von Sonderzeichen und Mapping von Abkürzungen in Vollschreibweisen.

4.1.2.2 Strukturelle und inhaltliche Analysen

Im Rahmen der Textanalyse wurden verschiedene Symbolzählungen durchgeführt, um Textstrukturmerkmale zu identifizieren. Zudem wurden Ngrams und Wortzählungen erstellt. Besondere Beachtung fanden Marker für Negationen und Synonyme. Diese Analyse liefert einen Überblick über den Informationsgehalt des Textes und ermöglicht die Identifizierung von Such- und Gruppierungselementen.

4.1.2.3 Inhaltsextraktion

In der Phase der Inhaltsextraktion werden die Ergebnisse der vorherigen Analyseschritte zur Ermittlung möglicher Stoffeigenschaften genutzt. Für jedes gefundene Such- und Gruppierungselement wird eine Regular Expression (RegEx) erstellt und deren Resultate überprüft. Bei zu großer Streuung oder zu wenigen Informationen wird alternativ eine Wort-, Phrasen- oder Volltextextraktion angewendet. Auffällige Sonderinformationen werden in einer zugehörigen Liste erfasst und entweder direkt auf der Liste oder auf einer RegEx basierend auf Listenelementen gesucht.

4.1.2.4 Transformation in Datenbank speicherbare Informationen

In Fällen, in denen die Ergebnismenge größer als 1 ist, ist eine Transformation der Daten für die Speicherung in einer SQL-Datenbank erforderlich. Alle Ergebnismengen werden als Text oder Listen in Form eines konkatenierten Text bereitgestellt.

4.1.2.5 Naive Kontrolle des Feldinhalt-Bearbeitungs-Zustands

Um einen Überblick über den Bearbeitungsgrad eines Feldinhaltes zu erhalten, wird eine Kopie jedes Feldes erstellt und die extrahierten Elemente schrittweise entfernt. Dieser Ansatz dient als Maß für den Bearbeitungsgrad, erhebt jedoch keinen Anspruch auf Vollständigkeit. Dies wird jedoch nicht beim Feld FREIEMP.FREIEMP angewendet, aufgrund der Komplexität des Merkmalsfeldes.

4.1.3 Zusammenfassung der Ergebnisse

Für vier spezifische Felder wurden Textmining-Methoden zur Informationsextraktion implementiert und die extrahierten Informationen in der Azure SQL Datenbank gespeichert. Diese Informationen können als neue Felder oder zur Befüllung von bestehenden Feldern verwendet werden. Die Informationen umfassen Stoffeigenschaften, Identmerkmale, Anweisungen und Gefahreninformationen sowie Metainformationen. Die Häufigkeit des Auftretens dieser Informationen in den Feldern wurde aufgelistet. Es wurde festgestellt, dass die Häufigkeit nicht mit der Anzahl der Stoffe gleichzusetzen ist, sondern die Häufigkeit des Auftretens der jeweiligen Information im entsprechenden Feld darstellt. Die extrahierten Informationen und deren Ausprägungen können bestehenden ChemInfo-Feldern zur weiteren Befüllung zugeordnet werden.

4.2 Chemprop Deep Learning-Modell zur Vorhersage von Stoffeigenschaften

Im nächsten Schritt wird untersucht, ob Machein-Learning (Deep Learning) Ansätze auf Basis der in der ChemInfo-Datenbank vorhandenen Strukturinformationen (MOL-Dateien) möglich sind.

4.2.1 Modellansatz und Informationsquellen

Die Zielsetzung des Projekts war es, eine Deep-Learning-basierte Methode zur Verbesserung der Befüllung von ChemInfo-Datenbankfeldern zu finden. Hierfür wurde das am MIT entwickelte Modell Chemprop identifiziert, welches auf Basis von SMILES Codes zur Vorhersage von Molekül- oder Stoffeigenschaften verwendet werden kann. Auf diese Weise können prädiktive Modelle mit guter Qualität, selbst für kleinere, domänenspezifische Datensätze trainiert werden. Zudem existiert eine frei verfügbare Python-Bibliothek zur Anwendung des Chemprop-Modells, was den Entwicklungsaufwand reduziert. Weiterhin können selbst bereitgestellte Informationen als Features in das Transfer Learning des Chemprop-Modells einbezogen werden, um die Prognosegüte zu steigern.

4.2.2 Technische Umsetzung

Im Folgenden wird die technische Implementierung des Chemprop-Modells zur Vorhersage unbekannter Werte in ChemInfo-Merkmalsfeldern beschrieben.

4.2.2.1 Jupyter Notebook zum Modelltraining, Modellevaluation und Ergebnisspeicherung

Das Jupyter Notebook "Chemprop.ipynb" ist das Kernelement für das Modelltraining, die Modellevaluation und die Erzeugung von Vorhersagen unbekannter Werte. Es ermöglicht die direkte Ausgabe von Ergebnissen als Tabellen oder Visualisierungen und die Kombination von statischen Inhalten und Ergebnissen. Bei jeder Ausführung des Notebooks wird ein Chemprop-Modell trainiert und evaluiert, HTML-Berichte werden erstellt und gespeichert, und

Vorhersagen für Substanzen ohne bekannten Wert im Zielmerkmalsfeld werden erzeugt und in der Azure SQL Datenbank gespeichert. Für eine schnellere Durchführung der Machine-Learning-Prozesse können Grafikbeschleuniger verwendet werden. Einige Python Funktionen wurden in ein separates Python Skript ausgelagert, um den Code im Notebook übersichtlicher zu gestalten.

4.2.2.2 Python Skripte und Excel Datei zur Spezifikation von Steuerungsparametern zur Durchführung multipler Trainingsläufe in einem Satz

Die Ausführung des Jupyter Notebooks zur Modellgenerierung kann manuell, sequenziell auf einem lokalen Rechner oder parallelisiert in Microsoft Azure erfolgen. Bei sequenzieller und parallelisierter Ausführung wird ein Python Skript verwendet, das Steuerungsparameter aus einer Excel-Datei importiert und für jedes Parameter Set das Jupyter Notebook ausführt. Die Bedeutung und Funktion der Steuerungsparameter werden in den HTML-Berichten erläutert. Die automatisierte Ausführung in Microsoft Azure basiert auf einer Azure Databricks Python Umgebung.

4.2.3 Erläuterungen zur fachlichen Umsetzung der Modellerstellung

In folgenden Abschnitt werden die wesentlichen Aspekte und die Begründung für die gewählten Vorgehensweisen im Rahmen der fachlichen Umsetzung zur Erstellung von Chemprop-Modellen erläutert.

4.2.3.1 Informationen, welche in den HTML-Berichten enthalten sind

Die folgenden Informationen dienen als Zusatzinformationen zu den Beschreibungen in den HTML-Berichten der einzelnen Modelle. In den Berichten sind bereits Beschreibungen zur Bedeutung der Steuerungsparameter, den Ergebnistabellen und Visualisierungen sowie zu grundlegenden fachlichen Informationen enthalten.

4.2.3.2 Informationen zum Abruf der Modelldaten aus SQL-Datenbank

Die Daten für die Entwicklung der Chemprop-Modelle liegen in einer Tabelle der Azure SQLDatenbank und werden in den Python Prozess importiert. Die Daten enthalten SMILES Codes, Ausprägungen des Zielmerkmals und funktionelle Gruppen als zusätzliche Inputvariablen. Der importierte Datensatz wird in Trainings-, Validierungs- und Testpartitionen unterteilt. Die Trainingspartition wird zum Training der Feed-Forward-Layer des vortrainierten Chemprop-Modells verwendet. Die Validierungspartition wird zur Ermittlung der optimalen Anzahl von Trainingsepochen und zur Validierung bzw. Berechnung der Zielmetrik nach jeder Iteration im Hyperparameter Tuning verwendet. Die Testpartition wird zur Modellevaluation und Beurteilung der Prognosegüte sowie zur Durchführung weiterführender Analysen verwendet. Im entsprechenden Abschnitt des Berichts werden darüber hinaus vertiefende Details zu den Modelldaten und den Datenpartitionen dargestellt.

4.2.3.3 Umsetzung der Modellvalidierung

Zur Modellvalidierung wurde der klassische Ansatz der Kreuzvalidierung aufgrund von Problemen wie geringen Mengen von Beobachtungen und technischer Komplexität nicht angewendet. Stattdessen wurde ein „Double-Holdout"-Ansatz gewählt, bei dem neben der Trainingspartition zwei weitere Partitionen (Validierungs- und Testpartition) verwendet wurden. Um eine umfassende Validierung zu gewährleisten, wurden mehrere Trainingsläufe für ein Modell durchgeführt, wobei die Aufteilung der Daten in Partitionen so erfolgte, dass einzelne Beobachtungen auch in mehreren Modellierungsläufen in Validierungs- oder Testpartition vorkommen konnten. Dies ermöglichte eine umfassende Analyse der Ergebnisse.

4.2.4 Auswahl der modellierten ChemInfo Merkmale (Zielvariablen)

Die Auswahl der Merkmalsfelder zur Befüllung mittels Chemprop basierte auf mehreren Kriterien: hinreichende Befüllung, Wertespektrum, Kausalität und Art des Merkmalsfeldes und Modelltyp. Es ist zu betonen, dass insbesondere eine ausreichende Anzahl von Beobachtungen und eine standardisierte Werteverteilung erforderlich sind. Bei der Auswahl der zu modellierenden Merkmalsfeldern ist ebenfalls der anzuwendende Modelltyp zu berücksichtigen, da jeder Modelltyp eine individuelle Umsetzung der Prozesse erfordert. Aufgrund der begrenzten Entwicklungszeit wurde eine Abwägung hinsichtlich der Aufwand-Nutzen-Relation vorgenommen. In Abbildung 11 werden Informationen zu verschiedenen Feld- und den entsprechenden Modelltypen übersichtlich dargestellt.

4.2.4.1 Kategoriale Merkmalsfelder (Multi-Class Klassifikation) – implementiert

Für die Multi-Class Klassifikation wurden verschiedene kategoriale Merkmalsfelder implementiert, darunter Gesundheitsgefahr, Reaktionsgefahr, Wassergefährdungsklasse, Aggregatzustand, besondere Anweisungen, Feuerlöschmittel, Brandgefahr und Körperschutz-Stoffbehandlung. Die Gründe für die Auswahl dieser Felder sind die hohe Anzahl potenzieller Felder mit ausreichender Befüllung, die Möglichkeit der direkten Verwendung der Modelloutputs für Prognosen, das Fehlen spezieller fachlicher Problemstellungen und ein günstiges Aufwand-Nutzen-Verhältnis.

4.2.4.2 Kategoriale Felder (Multi-Label) – nicht implementiert

Die kategorialen Felder für den Modelltyp Multi-Label wurden geprüft, aber aufgrund verschiedener Problemstellungen nicht implementiert. Die Probleme umfassen unter anderem ein ungeeignetes Wertespektrum und die geringe Befüllung entsprechender Merkmalsfelder. Der Aufwand zur Implementierung von Multi-Label Modellen ist außerdem höher als bei anderen Modelltypen, da jede Ausprägung als einzelne Zielvariable behandelt werden muss. Nur die Merkmalsfelder der GHS-Symbole erfüllten ansatzweise die Auswahlkriterien für eine Modellierung.

4.2.4.3 Numerische Felder (Regression) – nicht implementiert

Die numerischen Felder für den Modelltyp Regression wurden geprüft, aber nicht implementiert. Die Felder umfassen Merkmale wie z. B. Dichte, Flammpunkt, Siedetemperatur, Wasserlöslichkeit und Dampfdruck. Die Gründe für die Nichtimplementierung sind die Unsicherheit über die Genauigkeit der Prognosen, der Bedarf einer statistischen Methodik zur Überprüfung der zu erwartenden Prognosefehler und das ungünstige Aufwand-Nutzen-Verhältnis im Vergleich zur Implementierung der Multi-Class Klassifikationsmodelle.

4.2.5 Analyse und Optimierung von Steuerungsparametern

4.2.5.1 Zielstellung

Nach Abschluss der technischen Umsetzung der Modellentwicklung für bestimmte ChemInfo Felder wurde eine Analyse zur Prüfung der Robustheit der Modellperformance gegenüber verschiedenen Train-Vali-Test Splits sowie zur Ermittlung der mittleren Modellgenauigkeit für verschiedene Sätze von Steuerungsparametern durchgeführt. Die Zielstellung hierin bestand in der Optimierung der Einstellungen für das Training der finalen Modelle.

4.2.5.2 Vorgehensweise

Die Modellvalidierung wurde umgesetzt, indem mehrere Sätze von Steuerungsparametern definiert und für jedes implementierte ChemInfo Feld und jedes Parameter Set 30 Modelle trainiert und evaluiert wurden. Insgesamt wurden somit 1440 Modelle trainiert. Um unterschiedliche Beobachtungen zu gewährleisten, wurde pro Satz von Steuerungsparametern

und Merkmalsfeld eine Stichprobe von 30 Beobachtungen (Modellen) generiert, um eine Anwendung statistischer Tests zu ermöglichen. Außerdem lagen damit die Laufzeiten zum Training der Modelle noch in einem angemessenen Bereich. Die Evaluationsergebnisse wurden anschließend in Python analysiert.

4.2.5.3 Sätze von Steuerungsparametern (Parameter Sets)

Im Rahmen der technischen Implementierung wurden verschiedene Sätze von Steuerungsparametern getestet, um Auswirkungen auf die Modellperformance zu prüfen. Ein Baseline-Parameterset wurde als Ausgangsbasis definiert und weitere Parametersets wurden erstellt, in denen zusätzliche Einstellungen aktiviert, deaktiviert oder variiert wurden. Insgesamt wurden sechs verschiedene Parametersets getestet, darunter solche, die zusätzliche Inputvariablen verwenden, die Größe der Validierungspartition variieren, Modellensembles verwenden, SMILES Codes ausschließen, die durch Webscraping ermittelt wurden, und Hyperparameteroptimierung aktivieren.

4.2.5.4 Auswertung der Ergebnisse

Nach der Durchführung der Trainingsläufe wurden verschiedene Auswertungen in Form von Abbildungen und Tabellen erstellt. Diese werden in den folgenden Abschnitten detailliert beschrieben und anschließend Abschnitts zusammengefasst.

4.2.5.5 Beschreibung von Abbildung 12: Verteilung der Modellgenauigkeit für verschiedene Sätze von Steuerungsparametern

Die Abbildung 12 zeigt die Verteilung der Modellgenauigkeit für verschiedene Sätze von Steuerungsparametern. Jede Box repräsentiert 30 Trainingsläufe und die Genauigkeit der Modelle, basierend auf den Modellprognosen im Vergleich zu den tatsächlichen Werten der Zielvariable.

4.2.5.6 Beschreibung von Tabelle 9: Ergebnisse Signifikanztest auf Unterschiede der Modellgenauigkeiten (Accuracy) für verschiedene Sätze von Steuerungsparametern (Parameter Sets)

In Tabelle 9 werden die Ergebnisse eines Signifikanztests präsentiert, der durchgeführt wurde, um festzustellen, ob sich die Median Accuracy der zusätzlichen Parameter Sets signifikant von der des "baseline" Parameter Sets unterscheidet. Der verwendete Test ist der Mann-Witney-U-Test, ein nicht-parametrischer Test, der bei kleinen Stichproben und bevorzugt wird, welche nicht zwingend normalverteilt sein müssen. Die Tabelle zeigt für jedes implementierte ChemInfo Feld und Parameter Set die Median Accuracy und die Ergebnisse des Tests. Ein p-Wert < 0,05 deutet auf einen signifikanten Unterschied hin.

4.2.5.7 Beschreibung der Abbildungen zur Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Abbildung 13 bis Abbildung 20)

Die Abbildungen 13 bis 20 zeigen die Verteilung der Modellgenauigkeit (Precision) für verschiedene Ausprägungen der Zielvariable für verschiedene Sätze von Steuerungsparametern. Jede Abbildung repräsentiert ein spezifisches ChemInfo Feld (Zielvariable) und zeigt die Verteilung der Precision für jede der möglichen Ausprägungen. Die Ausprägungen sind als Indexwerte nummeriert und die Median Anzahl der abgegebenen Prognosen ist für jede Ausprägung vermerkt.

4.2.5.8 Beschreibung der Abbildungen zur Verteilung der optimalen Anzahl von Trainingsepochen pro Zielvariable und von Steuerungsparametern (Abbildung 21 und Abbildung 22)

Die Abbildungen 21 und 22 zeigen Histogramme zur Verteilung der optimalen Anzahl von Trainingsepochen pro Zielvariable und Satz von Steuerungsparametern. Die Anzahl der

Trainingsepochen wurde auf 15 begrenzt, um die Trainingslaufzeit zu reduzieren. Um zu überprüfen, ob 15 Epochen ausreichend sind, wurden Histogramme erstellt, die die Anzahl der Epochen zeigen, die benötigt wurden, um den geringstmöglichen Validierungs-Loss zu erzielen. Für ein Parameter Set wurde die Anzahl der Epochen auf 25 erhöht, um zu prüfen, ob sich der Validierungs-Loss nicht unerwartet durch weitere Trainingsepochen über 15 hinaus weiter reduzieren lässt.

4.2.5.9 Zusammenfassung der Ergebnisse und Schlussfolgerungen

Die Analyse der Modellgenauigkeiten zeigt deutliche Unterschiede zwischen den einzelnen ChemInfo Feldern (Zielvariablen), die auf verschiedene Faktoren zurückzuführen sein können. Die zusätzliche Verwendung von SMILES Codes aus externen Quellen hat keinen systematisch positiven Effekt auf die Ergebnisse. Die Optimierung der Hyperparameter führt in den meisten Fällen zu einer Verschlechterung der Ergebnisse. Das Parameter Set „fg50val5ens5noscs“ liefert die besten Ergebnisse hinsichtlich der Genauigkeit und Robustheit und wird daher für das Training der finalen Modelle verwendet. Die spezifizierte Anzahl von 15 Trainingsepochen ist ausreichend.

4.2.6 Erstellung finaler Modelle und Verwendung zur Lückenbefüllung

Für die Befüllung der implementierten ChemInfo Felder wurden finale Modelle mit den ausgewählten Steuerungsparametern trainiert. Die Auswahl der finalen Modelle erfolgte nicht auf Basis der Stichprobenanalyse, um Overfitting zu vermeiden. Stattdessen wurden die Modelle mit zufälligen Seed Values trainiert und überprüft, ob die Anzahl der Trainingsepochen ausreichend war. Die Evaluationsberichte der finalen Modelle wurden dem Zwischenbericht beigefügt.

4.2.7 Offene Schritte zur Verwendung der Chemprop Vorhersagen in der Datenblatterstellung

Um fehlende Stoffinformationen in ChemInfo durch Chemprop Prognosen zu ergänzen, wurde ein Prozess ausgearbeitet, der eine Vorabfilterung der Prognosen, die Speicherung der Prognosen in der Azure SQL Datenbank, die Unterscheidung von Prognosen und Originaldaten aus ChemInfo und die Kennzeichnung von Datenblattinformationen, die auf Chemprop Prognosen basieren, beinhaltet. Die tatsächliche technische Implementierung des angedachten Prozesses wurde jedoch im Projektzeitraum nicht durchgeführt.

4.3 Regelwerk zur Generierung von DB-Einträgen

Zusätzlich zu den zuvor genannten Methoden zur Steigerung des Befüllungsgrad von ChemInfo Merkmalsfeldern können regelbasiert neue Sachverhalte erzeugt werden. Basierend auf den Entscheidungsbäumen der Vorstudie des Fraunhofer ICT können Regeln erstellt werden, um fehlende Werte in Merkmalsfeldern für bestimmte Stoffe zu ergänzen, wie z. B. die Bestimmung der Brennbarkeit eines Stoffes. Es ermöglicht auch die Ableitung von Gefahrenkategorien aus anderen Merkmalsfeldern, wenn bestimmte Bedingungen erfüllt sind. Bei widersprüchlichen Ergebnissen wird das Ergebnis mit der größeren Gefahr ausgegeben.

5 Regelwerk zur Verbalisierung der Faktendaten

5.1 Entwurf der Datenblätter

Ein Datenblattentwurf wurde erstellt, um Einsatzkräften an Einsatzstellen schnelle und übersichtliche Informationen zu den beteiligten Gefahrstoffen zu geben. Der Entwurf, der in Zusammenarbeit mit der ICT-Werkfeuerwehr erstellt wurde, richtet sich hauptsächlich an die Einsatzleitung der Feuerwehr und gliedert sich in vier Bereiche: Stammdaten, Charakterisierung/Nachweis, Eigenschaften und empfohlene Maßnahmen. Die ausgewählten

Symbole sollen allgemein verständlich sein und kritische Expositionswege verdeutlichen. Der Datenblattentwurf kann bei Bedarf für weitere Benutzergruppen angepasst werden.

5.2 Fachlicher Hintergrund zum Regelwerk

Das Regelwerk für die Anzeige von Informationen auf dem Faktendatenblatt basiert auf den Ergebnissen der Vorstudie. Es beinhaltet Regeln für die Extraktion und Generierung von Informationen aus Datenbankfeldern. Die Regeln sind in Gruppen unterteilt und beinhalten Merkmalsfelder, Merkmalsausprägungen und Hierarchien. Es gibt vier Arten von Aggregationsmethoden, die zur Kombination von Informationen aus mehreren Regeln verwendet werden, welche in diesem Abschnitt des Berichts detailliert beschrieben werden. Die Regeln sind in der Datei "Regelwerk.xlsx" aufgeführt und werden in Tabelle 10 dargestellt. Einige Beispiele für die Informationen, welche durch Anwendung der Regeln aufbereitet werden, sind Wasserlöslichkeit, Bildung gefährlicher Reaktionsprodukte bei Hitze oder NFPA-Gefahrendiamant-Gesundheitsgefahr.

5.2.1 Beispielhafte Erläuterung anhand Wasserlöslichkeit (ID 238)

Die Wasserlöslichkeit ist eine Regel, die auf der im Merkmalsfeld WL.WL angegebenen Wasserlöslichkeit basiert und verbale Aussagen aus Tabelle 11 wiedergibt. Die Tabelle zeigt die Einstufung der Wasserlöslichkeit nach dem Europäischen Arzneibuch, mit Bezeichnungen und Konzentrationsgrenzen von "sehr leicht löslich" (>1000 g·l−1 H2O) bis "praktisch unlöslich" (< 0,1 g·l−1 H2O).

5.2.2 Beispielhafte Erläuterung anhand der Bildung gefährlicher Reaktionsprodukte bei Hitze

Die Informationen zur Bildung gefährlicher Stoffe bei Erhitzen sind hauptsächlich in den Merkmalsfeldern GFRXREA.GFRXREA und KONBRT.KONBRT gespeichert. Eine Textanalyse wurde durchgeführt, um relevante Informationen zu extrahieren. GFRXREA.GFRXREA hat einen homogeneren Datenbestand und die Suche nach bestimmten Textbausteinen reicht aus. Die Textanalyse für KONBRT.KONBRT ist komplexer und erfordert die Bildung einer Untermenge mit bestimmten Textbausteinen und die weitere Filterung auf das Vorliegen bestimmter Begriffe. Es gibt einige Ausnahmen, die keiner konsistenten Datenstruktur entsprechen und daher nicht berücksichtigt werden. Es wird empfohlen, diese Merkmalsfelder zu überprüfen und die Syntax anzupassen.

5.2.3 NFPA-Gefahrendiamant – Gesundheitsgefahr

Die Aggregationsregel 1031 für den NFPA-Gefahrendiamant Gesundheitsgefahr aggregiert die Ergebnisse der Regeln 121, 123, 124, 126, 127 und 129. Bei der Aggregation erhält Regel 121, die auf das Merkmal NFPA.GF zugreift, die höchste Priorität. Wenn das Merkmal NFPA.GF leer ist, wird die größte Gefahrenkategorie aus den übrigen Regeln ermittelt. Diese Regeln berücksichtigen verschiedene Gesundheitsgefahren, wie z.B. Vorliegen bestimmter Gefahrenhinweise (H-Sätze) oder spezifische Eigenschaften des Stoffes wie "tiefkalt" oder "Flüssiggas".

5.3 Technische Umsetzung

Die technische Umsetzung des Regelwerks basiert auf Daten aus der SQL-Datenbank "UBA_Export_Stoffe" und einer Kopie der Datei "Regelwerks.xls". Nach Anwendung des Regelwerks werden die Ergebnisse in die SQL-Tabelle "HazardRulesResults" geschrieben. Die Implementierung der Regeln erfolgt in einem Python Databricks Notebook und wird auf Databricks Clustern ausgeführt. Jede Regel ist in Form einer eigenen Python Funktion abgebildet, in der die Logik in Python Code und SQL umgesetzt wird. Die Ergebnistabelle enthält

die Stoffe, für die die Regellogik angewendet werden konnte. Einige Regeln verwenden die Ergebnisse der "HazardRulesResults" Tabelle als zusätzliche Eingabe.

5.4 Status der technischen Umsetzung des Regelwerks

Im Projekt konnte das Regelwerk aufgrund seines großen fachlichen Umfangs nicht vollständig technisch umgesetzt werden. Der Status der technischen Umsetzung wird im Regelwerk.xls Dokument angezeigt und umfasst drei mögliche Einträge: „Implementiert“, „Implementierter Code muss angepasst werden“ und „nicht implementiert - nur fachliche Konzeption“. Zudem wurden die Spalten Ebene 1 bis 4 und die Spalte „Aggregationsmethode“ nicht technisch umgesetzt.

Summary

1 Preliminaries

1.1 Target Audience

This document is intended for members of the "Factual Data for the ChemInfo Information System" project and documents the scientific approaches, work, and results of the research project.

1.2 Scope

This document is exclusively for internal use within the FDCI-UBA project and possible follow-up projects with the Federal Environment Agency. Any reproduction, storage, reformatting, transmission, or distribution in electronic or physical form requires prior approval from SWO.

1.3 Confidentiality

This document is subject to the provisions for handling sensitive documents. Individuals who are not authorized to access it are not allowed to receive information about the existence and content of this document.

1.4 Obligations

The guidelines for project work mentioned in the document are mandatory for all project members, and compliance is monitored by the project management.

2 Problem Statement and Objective

The Hazardous Substance Quick Information (GSA), a subset of the ChemInfo system, is intended to be improved using artificial intelligence as part of a project. The goal is to increase the level of data filling and information content and to develop simplified hazard and action texts. Tasks include the analysis of existing data, the evaluation and implementation of deep learning models, and the implementation of algorithms for verbalizing substance-related factual data. The report covers the project period from May 2, 2022, to May 29, 2023.

3 Data Analysis and Data Preprocessing

3.1 Objectives

The aim of data preprocessing is to understand the data structure of the ChemInfo data repository and convert it into a simplified tabular form. This allows for the examination of relationships, data filling levels, or value distributions. Based on this, methods for filling missing information and generating data sheets can be developed and implemented.

3.2 Data Preprocessing

3.2.1 ETL Process Structure

An ETL pipeline was implemented in the Azure Cloud to transform the hierarchical data structure of the JSON export into a tabular form that can be better processed in an SQL database. The transformation steps were carried out using Python, PySpark, and SQL in Azure Databricks and automated through Azure Data Factory pipelines. The pipeline includes processes such as data flattening, additional substance names, feature collection, data pipeline, data mining, data analysis, and the processing of structural data. After data preparation through ETL processes, the data is used for analysis.

3.2.2 Categorization of Feature Fields by Relevance

The features for substance description were evaluated for their relevance to data sheet display and property derivation in order to reduce the data scope. Features with little or no relevance for the environmental agency's use cases were deemed non-relevant. Unclear features were marked separately. A distinction was made between derivable and non-derivable features. During the project, it was found that a division into likely relevant and non-relevant data was sufficient. Only feature data categorized as relevant were used in the subsequent process.

3.2.3 Use of Structural Data (MOL Files) for Generating SMILES Codes and Assigning Functional Groups

The examination of the availability of structural information revealed that there is an MOL file for approximately 37,000 out of a total of 49,357 individual ingredients. SMILES codes were generated using the MOL files and the Python library RDKit to use structural information for machine learning models. Through the analysis of these codes, molecular properties were identified and assigned to the substances under investigation. In the further course of the project, only SMILES codes and information about the presence of functional groups were needed. Functional groups in SMILES were identified using SMARTS codes. After identifying these SMARTS in the available SMILES, the determined functional groups were checked for correctness, and some issues were identified and addressed.

3.2.3.1 Determination of Missing Structural Information (SMILES) through Web Scraping

To increase the availability of MOL files, it was checked whether additional structural data (SMILES codes) could be enriched by using external sources. After considering licensing aspects, a second source was used to import additional SMILES codes into the Azure SQL database. By using external sources, approximately 5,700 additional SMILES codes were filled. However, these were only used in the analysis to select meaningful control parameters and were not used for training the final Chemprop models or filling ChemInfo fields.

3.3 Data Analysis

The analysis of the data in the ChemInfo system shows a fragmented filling of features and a low standardization of input values. Only 1.45% of the relevant features are sufficiently filled, and only 1.09% meet the conditions of the performance offer of a maximum of 25% gaps. The analysis of value ranges also shows a mixing of information, duplicate statements, inconsistent implementation of measurable properties, and ambiguity in the features. Most features mainly allow free-text fields for description.

3.4 Summary of Data Analysis

The preliminary data analysis in the ChemInfo system revealed issues such as incomplete features and a lack of standardization in input values. As a solution, alternative approaches were developed, relying on existing structural information (MOL files). Based on this, an evaluation of a deep learning model using SMILES codes is to be conducted. Looking back, the key steps in this section include building an ETL pipeline for data preparation, analyzing structural information, and examining data availability and value distribution. These steps form the basis for methods to improve data completeness.

4 Methods for Improving Data Completeness

4.1 Information Extraction from Free-Text Fields Using Text Mining Methods

In a feasibility analysis, text mining methods were explored to extract information from free-text fields in order to increase the completeness of substance characteristics. ChemInfo fields with free-text character and extensive content were selected and examined for relevant information for other features. Prototype methods for extracting relevant content were implemented, attempting to assign them to existing values in equivalent target fields.

4.1.1 Selection of Free-Text Fields for Information Extraction

The selection of free-text fields for information extraction in the ChemInfo system was based on an analysis of field completeness and value distribution. After applying various filters, approximately 350 fields remained, from which a sample of 27 fields was drawn. After a manual review of field contents, eight fields were selected as promising for the application of text mining methods. These fields were later reduced to four fields in the project's further course, for which complete implementation of information extraction via text mining was carried out.

4.1.2 Implementation of Information Extraction

Information extraction was implemented in the following steps, which were applied identically for the four selected feature fields:

4.1.2.1 General Text Cleaning

The quality of text information is checked at the beginning to identify issues such as capitalization, spelling, word abbreviations, different languages, and incorrect symbols. These issues are then resolved through various transformation and cleaning methods, including text transformation to lowercase, inclusion of missing values, resolution of special characters, and mapping of abbreviations to full spellings.

4.1.2.2 Structural and Content Analysis

Various symbol counts were performed as part of text analysis to identify text structure features. N-grams and word counts were also created. Special attention was given to markers for negations and synonyms. This analysis provides an overview of the information content of the text and allows the identification of search and grouping elements.

4.1.2.3 Content Extraction

In the content extraction phase, the results of the previous analysis steps are used to identify possible substance properties. For each found search and grouping element, a regular expression (RegEx) is created, and its results are checked. In case of significant variation or insufficient information, word, phrase, or full-text extraction is applied as an alternative. Noticeable special information is recorded in an associated list and is either directly included in the list or searched based on list elements using a RegEx.

4.1.2.4 Transformation into Database-Storable Information

In cases where the result set is greater than 1, data transformation for storage in an SQL database is required. All result sets are provided as text or lists in the form of concatenated text.

4.1.2.5 Naive Control of Field Content Editing Status

To get an overview of the editing status of a field content, a copy of each field is created, and the extracted elements are gradually removed. This approach serves as a measure of the editing status but does not claim to be exhaustive. However, this is not applied to the field FREIEMP.FREIEMP due to the complexity of the feature field.

4.1.3 Summary of Results

Text mining methods for information extraction were implemented for four specific fields, and the extracted information was stored in the Azure SQL database. This information can be used as new fields or to populate existing fields. The information includes substance properties, identification features, instructions and hazard information, as well as meta-information. The frequency of occurrence of this information in the fields was listed. It was found that the frequency is not equivalent to the number of substances but represents the frequency of occurrence of the respective information in the corresponding field.

4.2 Chemprop Deep Learning Model for Predicting Substance Properties

The next step is to investigate whether machine learning (deep learning) approaches based on the structural information available in the ChemInfo database (MOL files) are possible.

4.2.1 Model Approach and Information Sources

The project's goal was to find a deep learning-based method to improve the population of ChemInfo database fields. For this purpose, the Chemprop model developed at MIT was identified, which can be used to predict molecule or substance properties based on SMILES codes. This allows for predictive models with good quality, even for smaller, domain-specific datasets. Furthermore, there is a freely available Python library for applying the Chemprop model, reducing development effort. Additionally, self-provided information can be included as features in the Chemprop model's transfer learning to enhance predictive performance.

4.2.2 Technical Implementation

The following describes the technical implementation of the Chemprop model for predicting unknown values in ChemInfo feature fields.

4.2.2.1 Jupyter Notebook for Model Training, Model Evaluation, and Result Storage

The Jupyter Notebook "Chemprop.ipynb" is the core component for model training, model evaluation, and generating predictions for unknown values. It allows for direct output of results as tables or visualizations and combines static content with results. Each execution of the notebook trains and evaluates a Chemprop model, generates HTML reports, and stores predictions for substances with unknown values in the target feature field in the Azure SQL database. To expedite machine learning processes, graphics accelerators can be utilized. Some Python functions have been moved to a separate Python script to make the code in the notebook more organized.

4.2.2.2 Python Scripts and Excel File for Specifying Control Parameters for Running Multiple Training Runs in a Batch

The execution of the Jupyter Notebook for model generation can be done manually, sequentially on a local machine, or in parallel on Microsoft Azure. In the case of sequential and parallel execution, a Python script is used to import control parameters from an Excel file and execute the Jupyter Notebook for each parameter set. The meaning and function of the control parameters are explained in the HTML reports. Automated execution in Microsoft Azure is based on an Azure Databricks Python environment.

4.2.3 Explanation of the Technical Implementation of Model Creation

In the following sections, the essential aspects and rationale for the chosen approaches in the technical implementation of creating Chemprop models are explained.

4.2.3.1 Information Included in the HTML Reports

The following information serves as additional information to the descriptions in the HTML reports of individual models. The reports already contain descriptions of the significance of the control parameters, result tables and visualizations, as well as fundamental technical information.

4.2.3.2 Information on Retrieving Model Data from SQL Database

The data for developing the Chemprop models are stored in a table in the Azure SQL database and are imported into the Python process. The data include SMILES codes, target feature values, and functional groups as additional input variables. The imported dataset is divided into training, validation, and test partitions. The training partition is used to train the feed-forward layer of the pre-trained Chemprop model. The validation partition is used to determine the optimal number of training epochs and for validation or calculation of the target metric after each iteration in hyperparameter tuning. The test partition is used for model evaluation, assessing prediction quality, and conducting further analyses. In the corresponding section of the report, in-depth details about the model data and data partitions are presented.

4.2.3.3 Implementation of Model Validation

For model validation, the classical approach of cross-validation was not applied due to issues such as low observation quantities and technical complexity. Instead, a "Double-Holdout" approach was chosen, in which, in addition to the training partition, two more partitions (validation and test) were used. To ensure comprehensive validation, multiple training runs for a model were conducted, and data splitting into partitions was done in such a way that individual observations could appear in validation or test partitions in multiple modeling runs. This allowed for a comprehensive analysis of the results.

4.2.4 Selection of Modeled ChemInfo Features (Target Variables)

The selection of feature fields for populating using Chemprop was based on several criteria: sufficient population, value range, causality, and type of feature field and model type. It should be emphasized that, especially, a sufficient number of observations and a standardized value distribution are required. When selecting the feature fields to be modeled, the applicable model type must also be considered, as each model type requires an individual implementation of processes. Due to limited development time, a cost-benefit analysis was carried out. Figure 11 provides information on different fields and their corresponding model types in a clear manner.

4.2.4.1 Categorical Feature Fields (Multi-Class Classification) – Implemented

For multi-class classification, various categorical feature fields were implemented, including health hazard, reactivity hazard, water hazard class, physical state, special instructions, fire extinguishing agents, fire hazard, and body protection substance treatment. The reasons for selecting these fields include a high number of potential fields with sufficient population, the possibility of using model outputs directly for predictions, the absence of specific technical issues, and a favorable cost-benefit ratio.

4.2.4.2 Categorical Fields (Multi-Label) – Not Implemented

Categorical fields for the multi-label model type were considered but not implemented due to various issues. These issues include an unsuitable value range and low population of corresponding feature fields. Implementing multi-label models also requires more effort compared to other model types, as each category needs to be treated as an individual target variable. Only the feature fields of GHS symbols partially met the criteria for modeling.

4.2.4.3 Numeric Fields (Regression) – Not Implemented

Numeric fields for the regression model type were considered but not implemented. These fields include features such as density, flash point, boiling point, water solubility, and vapor pressure. The reasons for not implementing them include uncertainty about the accuracy of predictions, the need for a statistical methodology to validate expected prediction errors, and an unfavorable cost-benefit ratio compared to implementing multi-class classification models.

4.2.5 Analysis and Optimization of Control Parameters

4.2.5.1 Objective

After completing the technical implementation of model development for specific ChemInfo fields, an analysis was conducted to test the robustness of model performance across different training-validation-test splits and to determine the average model accuracy for different sets of control parameters. The objective was to optimize the settings for training the final models.

4.2.5.2 Approach

Model validation was implemented by defining multiple sets of control parameters, and for each implemented ChemInfo field and parameter set, 30 models were trained and evaluated. In total, 1440 models were trained. To ensure different observations, a sample of 30 observations (models) was generated per set of control parameters and feature field to enable the application of statistical tests. Additionally, this kept the runtime for model training within a reasonable range. The evaluation results were subsequently analyzed in Python.

4.2.5.3 Sets of Control Parameters (Parameter Sets)

During the technical implementation, various sets of control parameters were tested to assess their impact on model performance. A baseline parameter set was defined as a starting point, and additional parameter sets were created in which additional settings were activated, deactivated, or varied. In total, six different parameter sets were tested, including those using additional input variables, varying the size of the validation partition, using model ensembles, excluding SMILES codes obtained through web scraping, and enabling hyperparameter optimization.

4.2.5.4 Evaluation of Results

After conducting the training runs, various analyses were conducted in the form of figures and tables. These are detailed in the following sections and then summarized.

4.2.5.5 Description of Figure 12: Distribution of Model Accuracy for Different Sets of Control Parameters

Figure 12 depicts the distribution of model accuracy for different sets of control parameters. Each box represents 30 training runs and the accuracy of the models, based on model predictions compared to the actual values of the target variable.

4.2.5.6 Description of Table 9: Results of Significance Test on Differences in Model Accuracy (Accuracy) for Different Sets of Control Parameters (Parameter Sets)

Table 9 presents the results of a significance test conducted to determine whether the median accuracy of the additional parameter sets significantly differs from that of the baseline parameter set. The test used is the Mann-Witney U-test, a non-parametric test preferred for small samples that do not necessarily need to be normally distributed. The table shows the median accuracy and test results for each implemented ChemInfo field and parameter set. A p-value < 0.05 indicates a significant difference.

4.2.5.7 Description of Figures on the Distribution of Precision for Each Target Variable and Sets of Control Parameters (Figure 13 to Figure 20)

Figures 13 to 20 illustrate the distribution of model accuracy (precision) for different values of the target variable for various sets of control parameters. Each figure represents a specific ChemInfo field (target variable) and shows the distribution of precision for each of the possible values. The values are numbered as index values, and the median number of predictions made for each value is noted.

4.2.5.8 Description of Figures on the Distribution of the Optimal Number of Training Epochs per Target Variable and Sets of Control Parameters (Figure 21 and Figure 22)

Figures 21 and 22 display histograms showing the distribution of the optimal number of training epochs per target variable and set of control parameters. The number of training epochs was limited to 15 to reduce training runtime. To check if 15 epochs were sufficient, histograms were created to show the number of epochs required to achieve the lowest possible validation loss. For one parameter set, the number of epochs was increased to 25 to assess whether further training epochs beyond 15 would unexpectedly reduce the validation loss.

4.2.5.9 Summary of Results and Conclusions

The analysis of model accuracies shows clear differences between individual ChemInfo fields (target variables), which can be attributed to various factors. The additional use of SMILES codes from external sources does not have a systematically positive effect on the results. Optimization of hyperparameters leads to a deterioration in results in most cases. The parameter set "fg50val5ens5noscs" provides the best results in terms of accuracy and robustness and is therefore used for training the final models. The specified number of 15 training epochs is sufficient.

4.2.6 Creation of Final Models and Use for Data Sheet Population

For populating the implemented ChemInfo fields, final models with selected control parameters were trained. The selection of final models was not based on the sample analysis to avoid overfitting. Instead, the models were trained with random seed values, and it was checked whether the number of training epochs was sufficient. The evaluation reports of the final models were included in the interim report.

4.2.7 Next Steps for Using Chemprop Predictions in Data Sheet Generation

To supplement missing substance information in ChemInfo with Chemprop predictions, a process was developed that includes preliminary filtering of predictions, storing predictions in the Azure SQL database, distinguishing between predictions and original ChemInfo data, and labeling data sheet information based on Chemprop predictions. However, the actual technical implementation of this process was not carried out during the project period.

4.3 Rule Set for Generating Database Entries

In addition to the methods mentioned earlier for increasing the population of ChemInfo feature fields, rule-based approaches can be used to generate new facts. Based on the decision trees from the preliminary study conducted by Fraunhofer ICT, rules can be created to supplement missing values in feature fields for specific substances, such as determining the flammability of a substance. It also allows for the derivation of hazard categories from other feature fields when certain conditions are met. In case of conflicting results, the result with the higher hazard is output.

5 Framework for Verbalization of Fact Data

5.1 Design of Data Sheets

A data sheet design has been created to provide rapid and clear information to emergency responders at incident sites regarding the hazardous substances involved. The design, created in collaboration with the ICT company fire brigade, is primarily intended for use by the fire department's incident command and is divided into four sections: Master Data, Characterization/Evidence, Properties, and Recommended Actions. The selected symbols are intended to be easily understood and clarify critical exposure pathways. The data sheet design can be customized for additional user groups as needed.

5.2 Technical Background for the Framework

The framework for displaying information on the fact sheet is based on the results of the preliminary study. It includes rules for extracting and generating information from database fields. The rules are divided into groups and include feature fields, feature values, and hierarchies. There are four types of aggregation methods used to combine information from multiple rules, which are detailed in this section of the report. The rules are listed in the "Rulebook.xlsx" file and are presented in Table 10. Some examples of the information that is prepared by applying the rules include water solubility, formation of hazardous reaction products when heated, and NFPA hazard diamond-health hazard.

5.2.1 Exemplary Explanation Using Water Solubility (ID 238)

The water solubility is a rule based on the water solubility specified in the WL.WL feature field and reflects verbal statements from Table 11. The table shows the classification of water solubility according to the European Pharmacopoeia, with designations and concentration limits ranging from "very soluble" (>1000 g·l−1 H2O) to "practically insoluble" (< 0.1 g·l−1 H2O).

5.2.2 Exemplary Explanation Using the Formation of Hazardous Reaction Products When Heated

Information on the formation of hazardous substances when heated is primarily stored in the feature fields GFRXREA.GFRXREA and KONBRT.KONBRT. A text analysis was conducted to extract relevant information. GFRXREA.GFRXREA has a more homogeneous dataset, and searching for specific text segments is sufficient. Text analysis for KONBRT.KONBRT is more complex and requires the formation of a subset with specific text segments and further filtering for the presence of specific terms. There are some exceptions that do not conform to a consistent data structure and are therefore not considered. It is recommended to review these feature fields and adjust the syntax.

5.2.3 NFPA Hazard Diamond - Health Hazard

Aggregation rule 1031 for the NFPA hazard diamond-health hazard aggregates the results of rules 121, 123, 124, 126, 127, and 129. During aggregation, rule 121, which accesses the NFPA.GF feature, takes the highest priority. If the NFPA.GF feature is empty, the highest hazard category is determined from the remaining rules. These rules consider various health hazards, such as the presence of specific hazard statements (H-phrases) or specific properties of the substance, such as "cryogenic" or "liquefied gas."

5.3 Technical Implementation

The technical implementation of the framework is based on data from the SQL database "UBA_Export_Stoffe" and a copy of the "Rulebook.xls" file. After applying the framework, the results are written to the SQL table "HazardRulesResults." The implementation of the rules is done in a Python Databricks notebook and is executed on Databricks clusters. Each rule is represented as its Python function, where the logic is implemented in Python code and SQL. The

result table contains the substances for which the rule logic could be applied. Some rules use the results of the "HazardRulesResults" table as additional input.

5.4 Status of the Technical Implementation of the Framework

In the project, due to its extensive technical scope, the framework could not be fully technically implemented. The status of the technical implementation is displayed in the "Rulebook.xls" document and includes three possible entries: "Implemented," "Implemented code needs adjustment," and "not implemented - only conceptual." Additionally, columns Level 1 to 4 and the "Aggregation Method" column were not technically implemented.

1 Vorbemerkungen

1.1 Zielgruppe

Dieses Dokument wurde für die Mitglieder des Projektes „Faktendaten für das Informationssystem ChemInfo“ (FDCI–UBA) erstellt. Es dokumentiert die im Forschungsprojekt verfolgten wissenschaftlichen Ansätze, die durchgeführten Arbeiten und deren Ergebnisse.

1.2 Einsatzbereich

Dieses Dokument bzw. Auszüge daraus sind ausschließlich für den internen Gebrauch beim Projekt FDCI-UBA und eventuelle Folgeprojekte mit dem Umweltbundesamt bestimmt. Eine darüberhinausgehende Vervielfältigung, Speicherung, Umformatierung, Übertragung und/oder Weitergabe bzw. Verteilung in elektronischer und/oder physikalischer Form, auch von Auszügen, bedarf der vorherigen Genehmigung der SWO.

1.3 Vertraulichkeit

Für das vorliegende Dokument gelten die Bestimmungen zur Behandlung von schutzbedürftigen Dokumenten.

Über die Existenz und die Inhalte dieses Dokuments ist gegenüber Personen, die nicht zu den Zugangsberechtigten der Projektdokumentation gehören, Stillschweigen zu bewahren.

1.4 Verbindlichkeit

Die in diesem Dokument aufgeführten Richtlinien zur Projektarbeit sind für alle Projektmitglieder verbindlich, ihre Einhaltung wird durch die Projektleitung überwacht.

2 Problemstellung und Ziel

Die Gefahrstoffschnellauskunft, kurz GSA, ist Teildatenbestand des ChemInfo-Systems und stellt Angaben und Informationen für verschiedenste Zielgruppen bspw. aus den Bereichen der Brand- und Explosionsgefahr, chemische Reaktionen und Umweltgefahren bereit. Da es sich bei dieser Stoffdatenbank um ein umfangreiches und komplexes System handelt und nicht alle Stoffeigenschaften vollständig befüllt bzw. beschrieben sind, soll im Rahmen dieses Projekts analysiert werden wie mit Hilfe von künstlicher Intelligenz der Grad der Befüllung und somit der Informationsgehalt gesteigert werden kann. Des Weiteren sind vereinfachte Gefahren- und Maßnahmentexte zu erarbeiten und mit Hilfe von merkmalsbezogen Algorithmen weiteren Stoffen systematisch und weitestgehend automatisch zuzuordnen. Die konkretisierten Aufgabenpakete beinhalten die Analyse der im ChemInfo-System vorhandenen Daten, die Evaluation und Umsetzung möglicher Deep Learning-Modelle, sowie die Erarbeitung und Implementierung der benötigten Algorithmen zur Verbalisierung der stoffbezogenen Faktendaten.

Der Zeitraum dieses Berichts umfasst die Erkenntnisse und Ergebnisse aus dem Projektzeitraum vom 02.05.2022 bis einschließlich 29.05.2023. Dieser beinhaltet die Punkte der fachlichen sowie strukturellen Datenanalyse, die Evaluierung und Umsetzung von KI-Modellen sowie die Implementierung eines entsprechenden Regelwerks zur Generierung der Verbalisierten Faktendaten auf Basis der Merkmalsfelder aus dem ChemInfo-System.

3 Datenanalyse und Datenvorverarbeitung

3.1 Zielstellung

Ziel der Datenvorverarbeitung ist die Datenstruktur des ChemInfo Datenbestands zu verstehen und die hierarchische Struktur des JSON-Exports in eine vereinfachte tabellarische Form zu überführen.

Auf dieser Grundlage können die Daten in Bezug auf Zusammenhänge zwischen Informationen, dem Grad der Befüllung sowie dem Spektrum und der Verteilung von Werteräumen untersucht werden. Basierend auf der Datenvorverarbeitung und -analyse können anschließend Methoden zum Auffüllen fehlender Informationen sowie zur Generierung von Datenblättern entwickelt und implementiert werden.

3.2 Datenvorverarbeitung

3.2.1 Aufbau ETL Prozess

Um die Datenvoranalyse und anschließend weitere Schritte zu ermöglichen, wurde eine ETL-Pipeline in der Azure Cloud implementiert, welche Vorverarbeitungsschritte abbildet, um die hierarchische Datenstruktur des JSON-Exports in eine tabellarische zu überführen. Die daraus resultierende Form kann anschließend besser in einer entsprechenden SQL-Datenbank übertragen und verarbeitet werden. Die entsprechenden Transformationsschritte (Data Wrangling) wurden unter Verwendung von Python, PySpark und SQL in Azure Databricks durchgeführt. Die Reihenfolge und Ausführung der einzelnen Teilprozesse bzw. Databricks Notebooks sind durch Azure Data Factory Pipelines definiert und automatisiert.

Der Aufbau der Pipeline kann in Abbildung 1 nachvollzogen werden. In der Darstellung repräsentieren die großen mintgrünen Boxen Azure Data Factory Pipelines. Die kleineren eisgrünen Boxen stellen Azure Databricks Notebooks (Python Prozesse) dar, welche innerhalb der jeweiligen Pipeline ausgeführt werden. Die hellblauen Boxen zeigen Dateien, welche Inputs oder Outputs der Python Prozesse sind.

Zur Datenspeicherung werden ein Azure Storage Account mit einem Azure Data Lake Storage sowie eine Azure SQL-Datenbank verwendet. Es ist zu beachten, dass die Databricks Entwicklungsumgebung und auch die zur Automatisierung verwendete Data Factory auf die genannten Speichereinheiten zugreift, um die gewünschten Datenverarbeitungsprozesse durchzuführen.

Die wesentlichen Python Prozesse innerhalb der Pipelines sind folgende:

- **Data Flattening:**
  Erzeugung einer flachen (tabellarisch statt hierarchisch) Datenstruktur, welche analytisch besser prozessierbar ist.
- **Additional Substance Names:**
  Hier werden die verschiedenen Schreibweisen der Namen relevanter Stoffe gesammelt und in der Datenbank abgelegt.
- **Feature Collection:**
  Hier werden die relevanten Merkmale aus der Stoffmodelleditor-Struktur ermittelt und diesen die, durch die Chemiker*innen vom Fraunhofer Institut definierten, Relevanzen zugeordnet. Informationen bezgl. der Relevanzen sind in Abschnitt 3.2.2 dargestellt.

- **Data Pipeline:** Hier werden die verflachten Daten (Stoffsachverhalte und -thesauri) weiterverarbeitet. Datensätze werden vereinheitlicht, Merkmale nach Relevanzen gefiltert und es erfolgt eine Trennung der Daten in Einzelinhaltsstoffe und Stoffklassen. Anschließend werden die prozessierten Daten in entsprechende Tabellen in der SQL-Datenbank geschrieben.

- **Data Mining:**
  Mittels Textmining werden, basierend auf den Ergebnisdaten des Teilprozess Data Pipeline, Informationen aus ausgewählten Freitextfeldern extrahiert, um damit nach Möglichkeit bestehende Merkmalsfelder aufzufüllen.

- **Data Analysis:**
  Die Resultate der vorangegangenen ETL-Prozesse werden in diesem Schritt aggregiert und es wird eine CSV-Datei generiert. Der Prozess wird je einmal für den Gesamtdatenbestand sowie den GSA-Datenbestand durchgeführt. Die Ergebnisdateien enthalten Zählungen von Stoffen, verfügbareren und unterschiedlichen Werten und einen Indikator (Gini-Koeffizient) zur Ungleichverteilung für jedes der prozessierten ChemInfo Felder. Diese Ergebnisse können später (manuell) in Microsoft Excel eingelesen und aufbereitetet werden, um einen Gesamtüberblick über den Datenbestand zu erhalten. Die Ergebnisse dieser Analyse werden in Abschnitt 4.2.5 vorgestellt.

- **Verarbeitung von Strukturdaten (CAS2SMILES, Collect Molecular Data):**
  Es gibt zwei wesentliche Teilprozesse in der Data Wrangling Automation Pipeline, welche in Verbindung mit Daten zur Molekülstruktur zu nennen sind.

  Zum einen wird versucht mittels dem CAS2SMILES Prozess zusätzliche SMILES Codes von Stoffen ohne verfügbare MOL-Datei per Web-Scraping anzureichern.

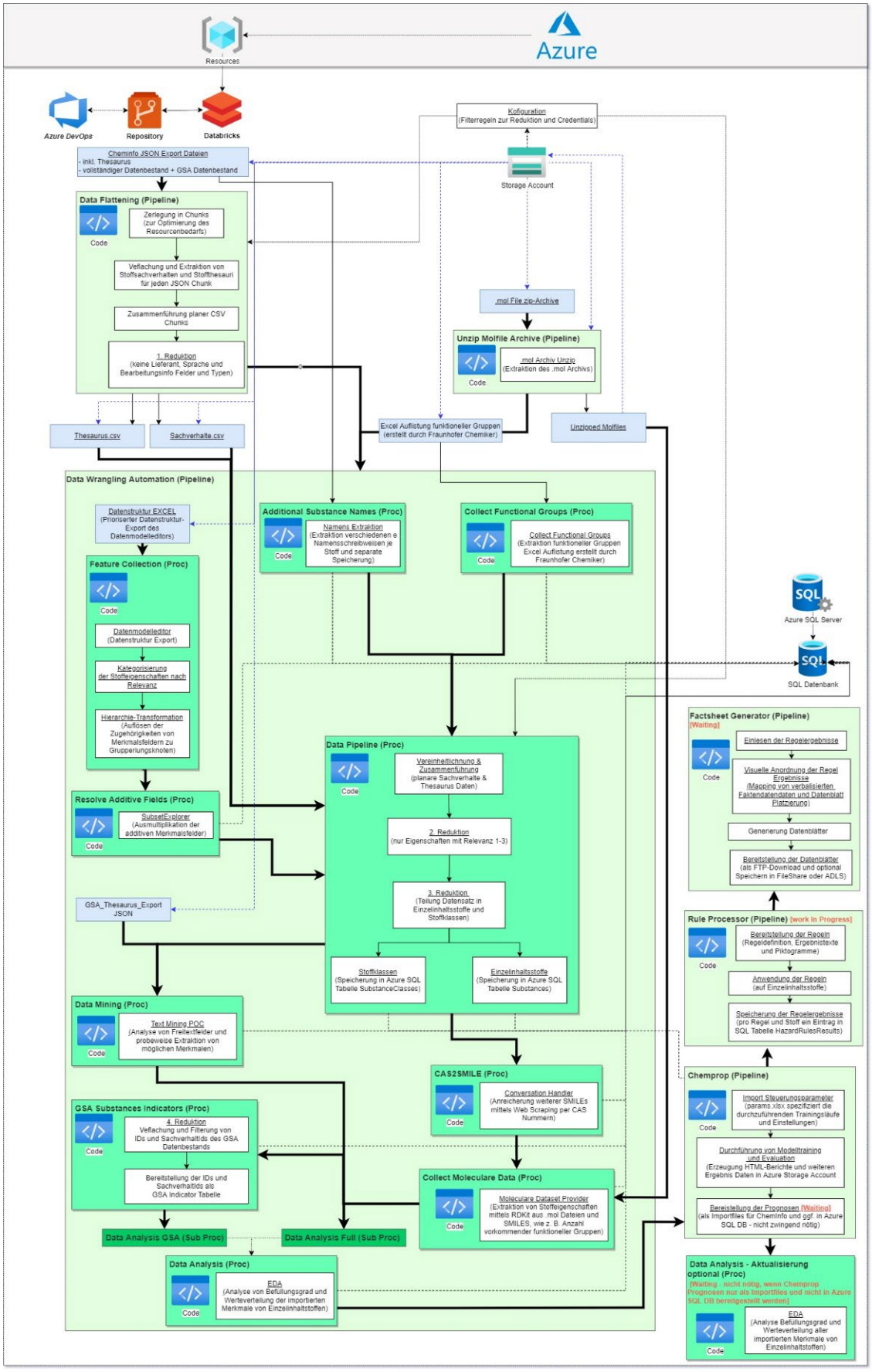
  Zum anderen werden anschließend im Teilprozess „Collect Molecular Data“ SMILES Codes aus den Moldateien ermittelt. Die so ermittelten SMILES Codes und die zuvor zusätzlich angereicherten SMILES Codes werden anschließend mittels Python Bibliothek RDKit weiterverarbeitet und es werden zusätzliche Strukturmerkmale, wie z. B. das Vorkommen von funktionellen Gruppen bei Stoffen, generiert. Diese Informationen (SMILES Codes und funktionelle Gruppen) werden anschließend als Variablen während des Chemprop-Modelltrainings verwendet.

- **Analytische Prozesse nach ETL-Pipeline (Chemprop, Rule Processor, Factsheet Generator):**
  Nach der Aufbereitung der Daten durch ETL-Prozesse, werden diese analytisch verwendet.

  Die einzelnen Prozesse werden in separaten Abschnitten dieses Berichts beleuchtet.

  - Chemprop: siehe Abschnitt 0
  - Rule Processor: siehe Abschnitt 5
  - Factsheet Generator: Dieser Teilprozess wird in Abstimmung mit dem Umweltbundesamt nicht umgesetzt.

Abbildung 1: Aufbau ETL-Pipeline



Quelle: eigene Darstellung, SoftwareOne.

3.2.2 Kategorisierung von Merkmalsfeldern nach Relevanz

Zur Reduktion des Datenumfangs wurden die Merkmale zur Stoffbeschreibung hinsichtlich ihrer Relevanz für die Datenblatt¬anzeige bzw. zur Ableitung von Eigenschaften für die Generierung konkreter Einsatzhinweise untersucht und anhand der Kriterien in Tabelle 1 bewertet. Informationen zu Stoffen, welche für die vom Umweltbundesamt geforderten Anwendungsfälle geringe bis gar keine Bedeutung besitzen, wurden als nicht relevant (Relevanz 4) eingestuft. Das betrifft beispielsweise Beziehungen von Einzelinhaltsstoffen und Merkmalsbeschreibungen hinsichtlich möglicher Lieferanten. Felder, deren grundsätzliche Bedeutung unklar waren, wurden mit Relevanz 5 versehen. Des Weiteren wurde eine Unterscheidung in aus anderen Merkmalsfeldern ableitbare (Relevanz 2 und 3) und nicht ableitbare Merkmale (Relevanz 1) unternommen. Während des Projektverlaufs stellte sich heraus, dass die Differenzierung von Relevanzen 1 bis 3 keinen Mehrwert bietet, sondern bereits eine Unterteilung in wahrscheinlich relevante (Relevanz 1-3) und nicht relevante Daten (Relevanz 4) genügt. Die Bedeutung der Felder mit Relevanz 5 wurden entweder geklärt oder stellten sich als nicht relevant heraus. Im weiteren Prozess wurde daher nur mit Daten mit den Relevanzen 1-3 gearbeitet.

Tabelle 1: Relevanzen zur Eingrenzung der Daten auf wesentliche Merkmalsfelder

Relevanz	Beschreibung
1	Relevante harte Fakten, physikalisch gemessene Daten
2	Relevante Daten, andere eindeutige Werte (z. B. toxisch, ätzend, H- und P-Sätze) Diese Daten können eventuell aus anderen Daten abgeleitet werden, z. B. das Attribut „ätzend“ aus dem PH-Wert.
3	Abgeleitete Daten, mit noch unklarer Bedeutung für das Projekt
4	Nicht relevante Daten
5	Bedeutung des Feldes unklar

Quelle: eigene Darstellung, SoftwareOne.

Eine genaue Auflistung der Merkmale und der entsprechenden Kategorisierung findet sich in dem mitgeltenden Dokument „Datenmodell Kategorisierung Nach Fraunhofer“.

3.2.3 Verwendung von Strukturdaten (MOL-Dateien) zur Erzeugung von SMILES Codes und Zuordnung funktioneller Gruppen

Die Untersuchung der Verfügbarkeit von Strukturinformationen zeigte, dass eine MOL-Datei bei ca. 37.000 von insgesamt 49.357 Einzelinhaltsstoffen vorliegt.

Um Strukturinformationen zur Erstellung von Machine-Learning Modellen zu verwenden, werden mit Hilfe der MOL-Dateien und Python Bibliothek RDKit[1] SMILES Codes erzeugt.

Durch Analyse des Aufbaus dieser standardisierten Zeichenketten können außerdem Moleküleigenschaften identifiziert werden, welche den zu untersuchenden Stoffen zugeordnet werden können. Beispiele hierfür sind z. B. Summenformel, Molmasse, Elementzähler, das Vorliegen funktioneller Gruppen und andere Standardformate zur Darstellung von Molekülstrukturen, wie SMILES oder InChi Codes. Neben den exemplarisch erwähnten Informationen könnte über bestimmte enthaltene funktionelle Gruppen (z. B. Peroxid- oder Nitro-Gruppen), auf gefährliche Eigenschaften wie Explosivität, Reaktivität usw. geschlossen

[1]Siehe auch: https://github.com/rdkit/rdkit

werden. Die Ergebnisse sind dem mitgeltenden Dokument „Informationen aus Strukturdaten“ zu entnehmen.

Auch wenn die Ableitung verschiedener solcher Informationen theoretisch möglich ist, wurden im weiteren Projektverlauf lediglich SMILES Codes und Informationen bzgl. des Vorkommens funktioneller Gruppen im Rahmen der Anwendung des Chemprop Machine Learning-Modells benötigt. Prinzipiell ist mittels funktioneller Gruppen auch eine Zuordnung zu Stoffgruppen z. B. Alkane, Aromaten, Carbonsäuren, Amine usw. möglich. Durch die Vielzahl organischer Stoffgruppen sowie der unterschiedlichen Schreibweise der Codes ist aber eine manuelle Zuordnung der relevanten Code-Elemente nicht möglich. Aus diesem Grund wurde nach einer bestehenden technischen Lösung zur Zuordnung von Stoffgruppen bzw. funktionellen Gruppen zu Stoffen mittels SMILES-Codes gesucht.

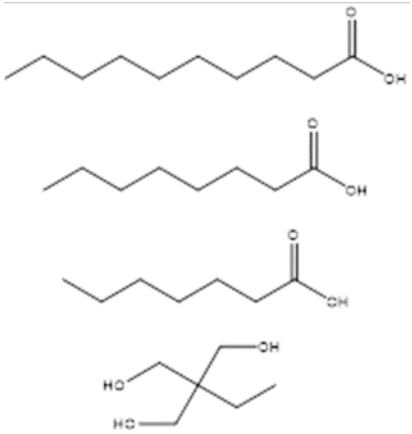
Die Recherche ergab, dass mit Hilfe von SMARTS Codes funktionelle Gruppen abgebildet werden, welche wiederum in SMILES identifiziert werden können. Dafür gibt es in RDKit bereitgestellte Funktionen, mit denen man in SMILES nach Substrukturen in Form von SMARTS suchen kann. Es kann so ermittelt werden, ob eine funktionelle Gruppe in einer Molekülstruktur vorliegt und ebenfalls wie oft. Dieser Prozess wurde im Rahmen der ETL-Pipeline automatisiert, für alle verfügbaren SMILES Codes durchgeführt und die Ergebnisse zur weiteren Nutzung (Chemprop-Modell) in die Azure SQL Datenbank übertragen.

Eine Liste der dafür erforderlichen SMARTS Codes wurde durch Recherche der Chemiker vom Fraunhofer Institut in einem Internet-Forum gefunden: Molecule Functionality node Request - Community Extensions / Indigo - KNIME Community Forum[2]. Die Liste liegt im XML-Format vor und wurde anschließend in ein tabellarisches Format überführt. Nach Durchführung des Prozesses der Identifikation dieser SMARTS in den verfügbaren SMILES wurden die ermittelten funktionellen Gruppen auf Korrektheit geprüft. Dafür wurde eine Stichprobe von 40 Stoffen zufällig ausgewählt und die Ergebnisse (identifizierte funktionelle Gruppen) wurden mit der Strukturformel und den SMILES Codes verglichen. Folgende Problematiken zeigten sich im Rahmen dieser Prüfung:

- Einige funktionelle Gruppen wurden nicht erkannt:
  Aliphatische Ringe, Heterocyclische Strukturen, Aromatische Strukturen mit Heteroatomen, Halogene, die nicht an Kohlenstoff gebunden sind, andere Heteroatome (N, P, O, etc.) die nicht an Kohlenstoff gebunden sind.
- Ionische oder salzartige Strukturen (z. B. GSBLRN 165662), Stereoisomerie, Anorganische Salze entweder als Metallorganik erkannt oder überhaupt nicht erkannt (z. B. GSBLRN 22927 SF6).
- Fehlerhafter SMILES-Code vorhanden:
  GSBL-Nr. 134274: Substanzen, die z. B. Mischungen aus verschiedenen Estern sind, werden aufgrund der SMILES-Schreibweise.
  [CCC(CO)(CO)CO.CCCCCCC(=O)O.CCCCCCCC(=O)O.CCCCCCCCCC(=O)O] als Mischungen aus den Edukten betrachtet (siehe Abbildung 2)

[2] Siehe auch: https://forum.knime.com/t/molecule-functionality-node-request/1230/3

Abbildung 2: Beispiel für fehlerhafte SMILES Darstellung bei Reaktionsprodukten



Quelle: eigene Darstellung, SoftwareOne.

Die SMILES repräsentieren statt dem Reaktionsprodukt die Edukte.

- Das kann zu falschen Vorhersagen der KI führen. Da die KI mit dem Reaktionsprodukt, dem Ester dieser Verbindungen arbeiten sollte.
- Deshalb wurden folgende Maßnahmen durchgeführt. Die SMARTS-Codes für aliphatische Ringe, heterocyclische und aromatische Strukturen, ionische/salzartige Strukturen, Gesamtzahl an Halogenatomen, andere Heteroatome und metallorganische Verbindungen wurden angepasst, wiederum mit Stichproben überprüft und validiert. Für konjugierte Doppelbindungen und alpha-beta-ungesättigte Carbonylverbindungen wurden neue SMARTS-Codes eingeführt. Diese Maßnahmen wurden händisch erbracht.
  Generelle Gruppierungen in „Organik“ oder „Anorganik“, die Angabe des Stickstoffgehalts oder die Existenz von Stereoisomeren/Chiralitätszentren können nicht über SMARTS-Codes erfolgen, sondern müssen z. B. aus der Summenformel abgeleitet werden (Organik = enthält „C“ und „H“; Anorganik = nicht Organik und nicht Metallorganik; Stickstoffgehalt(%) = (Anzahl N)*14*100 / Molmasse) oder können z. B. im Fall der Chiralitätszentren aus den MOL-Dateien direkt über das Chiralitätsflag in der Counts Line abgelesen werden.
- Eine umfangreiche Liste der funktionellen Gruppen findet sich im mitgeltenden Dokument „Informationen aus Strukturdaten“ mit entsprechenden Zuordnungen der funktionellen Gruppen zum jeweiligen SMARTS-Code.

3.2.3.1 Ermittlung fehlender Strukturinformationen (SMILES) mittels Web-Scraping

Die Untersuchung der Verfügbarkeit von Strukturinformationen zeigte, dass für insgesamt 49.357 Einzelinhaltsstoffen ca. 37.000 eine MOL-Dateien vorliegen. Um die Verfügbarkeit weiter zu erhöhen, wurde geprüft, ob zusätzliche Strukturdaten (SMILES Codes) durch Nutzung externer Quellen angereichert werden können. Im Folgenden sind die im Rahmen der Recherche gefundenen, potenziellen Quellen aufgeführt.

1. Chemical Identifier Resolver: CIR(Py)[3]
2. Common Chemistry CAS: SciFinder[4]
3. National Library of Medicine: PubChem (Py)[5]

[3] siehe: https://cactus.nci.nih.gov/chemical/structure, letzter Zugriff am 05.09.2023

[4] Siehe: https://commonchemistry.cas.org/detail?cas_rn=68309-98-8, letzter Zugriff am 05.09.2023

[5] Siehe: https://pubchem.ncbi.nlm.nih.gov/, letzter Zugriff am 05.09.2023

4. AMBINTER[6]
5. European Chemicals Agency[7]
6. National Library of Medicine: ChemIdPlus[8]

Bei der Bestimmung, welche der Quellen tatsächlich verwendet werden können, spielten hauptsächlich lizenzrechtliche Aspekte eine Rolle. Das UBA wurde in den Entscheidungsprozess involviert. In Abstimmung mit dem UBA wurde die zweite Quelle4 verwendet und als die entsprechenden SMILES Codes in die Azure SQL Datenbank importiert. Die Suche nach SMILES Codes auf den genannten Quellseiten erfolgt mittels CAS-Nummer, was bedeutet, dass für alle Stoffe ohne vorliegenden SMILES Code, bei denen jedoch eine CAS-Nummer vorliegt, eine Suche durchgeführt werden kann.

Es hat sich gezeigt, dass hier für einige Stoffe die SMILES ermittelt werden können. Dabei ist zu beachten, dass teilweise für dieselbe CAS-Nummer unterschiedliche Schreibweisen eines SMILES Codes zurückgegeben wurden. In Rücksprache mit dem Fraunhofer ICT konnte geklärt werden, dass es sich hierbei um die gleiche Stoffstruktur handelt. Der entsprechende SMILES Code wird von dem Punkt aus generiert, wo im Stoff die Atomzählung beginnt. Zur Absicherung, dass hier keine fehlerhaften Informationen in die Datenbasis eingebracht werden, wurde ein Kontrollmechanismus, welcher die kanonischen SMILES miteinander vergleicht, implementiert und als Qualitätskontrolle an die Ausgabedatenstruktur angehängt. Durch Verwendung der externen Quellen konnten ca. 5.700 zusätzliche SMILES Codes befüllt werden.

Abschließend ist anzumerken, dass SMILES Code in den analytischen Prozessen ausschließlich in der Chemprop-Modellierung verwendet wurden. Die zusätzlich mittels Webscraping ermittelten SMILES Codes fanden jedoch lediglich in der Erstellung der Analyse zur Auswahl sinnvollen Steuerungsparametern Anwendung. Da in den Resultaten der Untersuchung jedoch nicht erkennbar ist, dass die zusätzlichen Strukturinformationen zu merklichen Verbesserungen der Modellperformances führen, werden die SMILES Codes aus externen Quellen nicht zum Training der finalen Chemprop-Modelle bzw. zum Auffüllen von ChemInfo Feldern eingesetzt. Informationen zur genannten Analyse befinden sich in Abschnitt 4.2.5.

3.3 Analyse des Datenbestands

Um einen Überblick zur grundlegenden Verteilung von Inhalten zu erhalten, wurde der Grad der Befüllung von Merkmalsfeldern über alle Stoffe hinweg analysiert. Die Grundidee sieht es vor, Merkmale zu identifizieren, die entsprechend flächendeckend über nahezu alle Stoffe befüllt sind oder solche, die für fast keinen Stoff befüllt vorhanden sind, zu identifizieren. Das Ergebnis kann dann als Bewertungskriterium herangezogen werden, wie fragmentarisch die Merkmale in der ChemInfo-Datenbank befüllt sind und wie es im Hinblick auf einen KI-Ansatz mit der mengenmäßigen Verwertbarkeit aussieht. Neben der reinen Anzahl enthaltener Werte wurde auch die Anzahl von verschiedenen Ausprägungen zu den entsprechenden Merkmalen betrachtet. Des Weiteren wurde ein Maß (Gini-Koeffizient) zur Ungleichverteilung von Inhalten eingeführt. Zur Definition von verfügbaren Informationen (Datenlücke) wurden ebenfalls die Kriterien aus dem Kapitel 6.2.2.1.2 des Leistungsangebots zu Grunde gelegt.

- **Nicht-informative Werte:** standardisierte Platzhalter

[6] Siehe: https://www.ambinter.com/, letzter Zugriff am 05.09.2023

[7] Siehe: https://echa.europa.eu/advanced-search-for-chemicals?p_p_id=dissadvancedsearch_WAR_disssearchportlet&p_p_lifecycle=0&p_p_col_id=column-1&p_p_col_count=1, letzter Zugriff am 05.09.2023

[8] Siehe: https://chem.nlm.nih.gov/chemidplus/rn/3453-83-6, letzter Zugriff am 05.09.2023

- **Fehlwerte:** None/Null-Werte und Leertexte
- **Falschwerte:** Einträge die von Autoren oder Systemen falsch eingepflegt wurden (auch Formatierung)

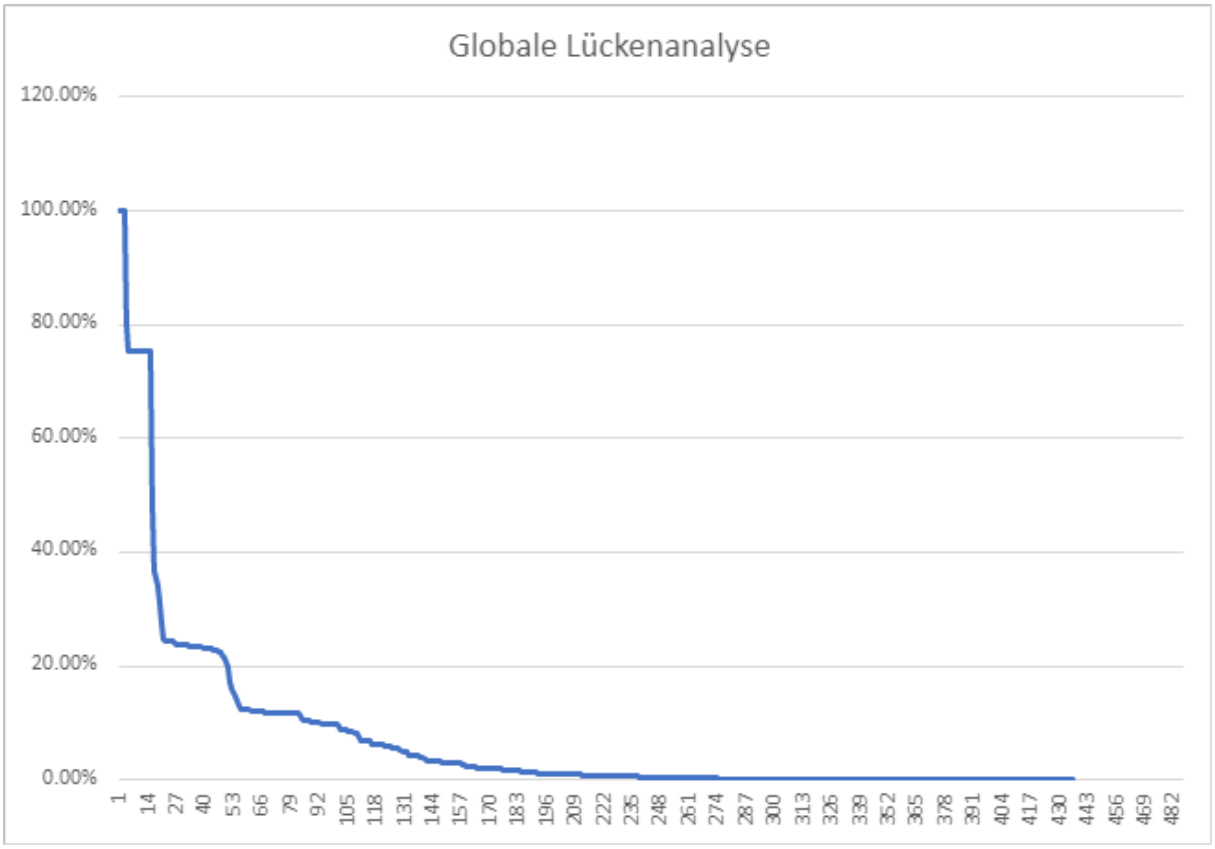
Außerdem wurden die Merkmale auf globaler Ebene, gemäß der im Leistungsangebot unter Kapitel 6.2.2.1 Darlegung der Erarbeitungsschritte beschriebenen Grundvoraussetzung untersucht. Dies umfasst folgende Punkte:

- Der Datensatz darf maximal 25 Prozent lückenhafte Daten aufweisen.
- Der Baseline-Datensatz muss eine nahezu gleichmäßige Verteilung über alle in Frage kommenden Stoffe aufweisen.

Das Ergebnis der Verfügbarkeitsanalyse von Inhalten ist im folgenden Diagramm (Abbildung 3) dargestellt. Es zeigt die Befüllung von Merkmalen im Verhältnis zu allen im ChemInfo-System vorhandenen Stoffen.

- **Y-Achse:** Prozentualer Anteil der Stoffe mit mindestens einem Wert
- **X-Achse:** Aufzählung der im ChemInfo-System vorhandenen Felder, absteigend nach Werten der Y-Achse

Abbildung 3: Verlauf Merkmalsverteilung



Quelle: eigene Darstellung, SoftwareOne.

Die Abbildung 3 zeigt deutlich einen starken Abfall der Merkmalsbefüllung über alle Stoffe hinweg. Zusätzlich zeigt sich, dass bei dem Großteil der Merkmale keine gleichmäßige Verteilung vorliegt, wie es im Leistungsangebot vorausgesetzt ist.

Im Folgenden zeigt Tabelle 2 einen Auszug aus dem mitgeltenden Dokument „Analyse_Merkmalsverteilung“.

- **Allgemeine Anzahl Werte:** Globale Anzahl der Merkmalswerte ohne Stoffbezug
- **Anzahl verschiedener Stoffe:** Anzahl der Einzelinhaltsstoffe, für die mind. ein Wert des Merkmals vorliegt
- **Anzahl verschiedener Merkmalswerte:** Werte des Merkmals, die mind. einem Einzelinhaltsstoff zugeordnet sind und einen unterschiedlichen Wert besitzen (Varianz des Werteraums)
- **Gini-Koeffizient:** Maß der ungleichen Verteilung (Häufigkeitsverteilung), Je näher der Wert bei 1 liegt, umso mehr Ungleichverteilung liegt für dieses Merkmal vor.

Tabelle 2: Beispiel Analyse Merkmalsverteilung

Anzahl Werte	Anzahl Stoffe	Merkmals-werte	Gini-koeffizient	Abgedeckte Stoffe in %	Kurz-beschreibung	Lang-bezeichnung
49357	49357	49357	0	100.00%	GSBL.GSBLRN	GSBL-RN
12754	11491	52	0.7976	23.28%	EHALLG.EHALLG	Allgemeine Maßnahmen
23032	11537	367	0.5557	23.37%	GFBR.GFBR	Brandgefahr

Quelle: eigene Darstellung, SoftwareOne.

- Betrachtet man alle 1375 relevanten Merkmale, so sind lediglich 20 von 1375, also ca. 1.45 Prozent der Merkmale, ausreichend befüllt. Diese Information kann dem mitgeltenden Dokument „Analyse Merkmalsverteilung“ entnommen werden. Aus dem niedrigen Grad der Befüllung ergibt sich eine starke Unterrepräsentierung über alle Merkmale hinweg.
- Setzt man die Untergrenze für den Grad der Befüllung auf nur 10 Prozent für die 49.357 benötigten Einträge, so verbleiben noch 126 von 1.375 möglichen Merkmalen. Dies entspricht einem prozentualen Wert von 9.16 Prozent.
- Betrachtet man den Datensatz stattdessen aus Sicht der Angabe gemäß des Leistungsangebots von 25 Prozent Lücken, was einem Grad der Befüllung von ca. 37.000 Einträgen pro Merkmal bei 49.357 Einzelinhaltsstoffen entspricht, dann zeigt sich, dass nur 15 aus 1.375 Merkmalen (1.09 Prozent) dieser Bedingungen gerecht werden. Dabei ist zu beachten, dass in den 15 Merkmalen auch die Pflichteingaben enthalten sind. Es ist zudem zu beachten, dass einige Stoffe für das gleiche Merkmal Doppelungen enthalten und dass für jeden Stoff mehrere Sachverhalte existieren können.

Neben der Analyse der Verfügbarkeit von Inhalten wurden die Daten ebenfalls auf vorhandene Werteräume untersucht. Diese Informationen können Aufschluss darüber geben, welche Felder ein annähernd standardisiertes Wertespektrum besitzen und welche Felder reine Freitexteingaben enthalten. Des Weiteren lassen sich numerische Felder, die kontinuierlich oder diskret befüllt sind, ermitteln. Folgende Ergebnisse sind aus dieser Betrachtung hervorgegangen:

► **Vermischung von Informationen in Merkmalen:** Merkmale enthalten große Werteräume im Hinblick auf Inhalte bzw. Informationen, zum Beispiel STBE.STBE in Abbildung 4 und ERSCHB.ERSCHB. in Abbildung 5.

Abbildung 4: Merkmal Stoffbeschreibung

Results Messages

	STBE.STBE
10	extrem dünnflüssig
11	Fasern
12	Feststoff
13	Feststoff in verschiedenen Formen
14	Feststoff oder Flüssigkeit
15	Feststoff oder Flüssigkeit; brennbar
16	Feststoff oder viskose Flüssigkeit
17	Feststoff, Lösung
18	Feststoff, Lösung; stabilisiert, explosibel
19	Feststoff, techn.: viskose Flüssigkeit
20	Feststoff, zersetzt sich schon unter Umgebungste...
21	Feststoff; brennbar
22	Feststoff; explosibel
23	Feststoff; polymerisiert
24	Feststoff; selbstentzündlich
25	Feststoff; stabilisiert, explosibel
26	Feuchte Kügelchen
27	Feuchter fester Stoff
28	Flocken
29	flücht. Flüssigkeit
30	flücht. Flüssigkeit o. Gas; brennbar, stabilisiert
31	flücht. Flüssigkeit o. Gas; leichtentz., stossempf.
32	flücht. Flüssigkeit; brennbar
33	flücht. Flüssigkeit; explosibel
34	flücht. Flüssigkeit; hochentzündlich
35	flücht. Flüssigkeit; leichtentz., licht-/luftempf.
36	flücht. Flüssigkeit; leichtentz.,stabilis. (92-84-2)
37	flücht. Flüssigkeit; leichtentzündl., polymerisiert
38	flücht. Flüssigkeit; leichtentzündlich
39	flücht. Flüssigkeit; selbstentzündlich
40	flücht. Flüssigkeit; stabilisiert (CAS 128-37-0)

Quelle: eigene Darstellung, SoftwareOne.

Abbildung 5: Merkmal Erscheinungsbild

Results | Messages

	ERSCHB.ERSCHB
16	Ätzende Flüssigkeit. Sichtbar als weisse Nebel.
17	Ätzende Flüssigkeit. Weiße ätzende Nebel bei Feu...
18	Ätzende gesundheitsschädliche brennbare Flüssigk...
19	Ätzende gesundheitsschädliche Flüssigkeit.
20	Ätzende giftige brennbare Flüssigkeit, Flammpunkt ...
21	Ätzende giftige brennbare Flüssigkeit, Flammpunkt ...
22	Ätzende giftige brennbare Flüssigkeit.
23	Ätzende giftige brennbare Flüssigkeit. Dämpfe leich...
24	Ätzende giftige Flüssigkeit.
25	Ätzende giftige reaktionfähige brennbare Flüssigkeit.
26	Ätzende giftige, schwer brennbare Flüssigkeit.
27	Ätzende giftige, sehr reaktionsfähige Flüssigkeit.
28	Ätzende hochgiftige brennbare Flüssigkeit.
29	Ätzende hochgiftige Flüssigkeit.
30	Ätzende wasseranziehende Flüssigkeit.
31	Ätzende, schwer brennbare Flüssigkeit, Flammpunk...
32	Ätzende, schwer brennbare Flüssigkeit. Dämpfe lei...
33	Ätzende, sehr reaktionsfähige Flüssigkeit.
34	Ätzender brandfördernder, sehr reaktionsfähiger Fe...
35	Ätzender brennbarer Feststoff, Flammpunkt >100°C.
36	Ätzender brennbarer Feststoff.
37	Ätzender brennbarer reaktionsfähiger Feststoff.
38	Ätzender Feststoff.
39	Ätzender Feststoff. Ab 180°C große Mengen ätzend...
40	Ätzender Feststoff. Ab 58°C ätzende giftige Flüssigk...
41	Ätzender Feststoff. An der Luft rauchende Substanz.
42	Ätzender Feststoff. Ätzende Staub/ Luftgemische.
43	Ätzender Feststoff / sehr reaktionsfähige Flüssigkeit.
44	Ätzender giftiger brennbarer Feststoff, ab 45°C Flüs...
45	Ätzender giftiger brennbarer Feststoff, Flammpunkt ...
46	Ätzender giftiger brennbarer Feststoff.

Quelle: eigene Darstellung, SoftwareOne.

- **Doppelaussagen:** Ein Problem stellen auch Doppelaussagen in einigen Merkmalen dar, da diese eine eindeutige Auswertung stark erschweren, zum Beispiel BBLOEMI (Löschmittel) in Abbildung 6. Hier sind die Vermischung von Informationen im Hinblick auf Positiv- und Negativaussagen bzw. widersprüchliche Inhalte in den Merkmalen zu erkennen. Im genannten Beispiel sind erlaubte und nicht erlaubte Löschmittel beschrieben.

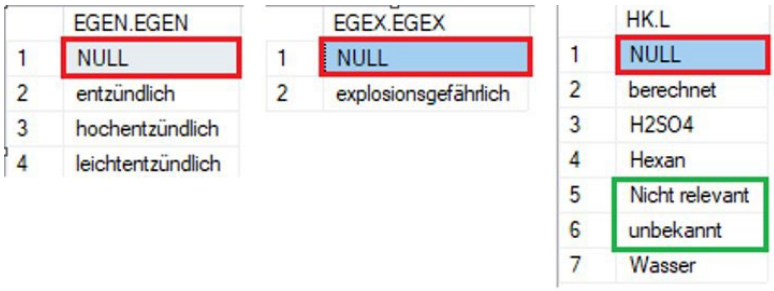
Abbildung 6: Merkmal Löschmittel

Quelle: eigene Darstellung, SoftwareOne.

- **Messbare Eigenschaften:** Werteräume sind hinsichtlich Entscheidungen, Rangfolgen und Unterteilung in relevante und nicht relevante Informationen inkonsistent umgesetzt bzw.

messbare Eigenschaften wie bspw. ordinale, nominale oder binäre Merkmale sind als solche nicht realisiert (siehe Abbildung 7).

Abbildung 7: Beispiele möglicher ordinale, nominale und binäre Merkmale



	EGEN.EGEN
1	NULL
2	entzündlich
3	hochentzündlich
4	leichtentzündlich

	EGEX.EGEX
1	NULL
2	explosionsgefährlich

	HK.L
1	NULL
2	berechnet
3	H2SO4
4	Hexan
5	Nicht relevant
6	unbekannt
7	Wasser

Quelle: eigene Darstellung, SoftwareOne.

- **Mehrdeutigkeit:** In einigen Merkmalen sind keine eineindeutigen bzw. objektiven Merkmalsbeschreibungen hinsichtlich einer fachlichen Auswertbarkeit gegeben. Diese Tatsache führt zu einem erhöhten Interpretationsspielraum aufgrund von subjektiven Beschreibungen, verdeutlicht durch (siehe Abbildung 8).

Abbildung 8: Merkmal Geruch (GER.GER)

Results | Messages

	GER.GER
1	NULL
2	"wie ""faules Heu"""
3	Alkoholischer Geruch
4	Alkoholischer Geruch; Angenehmer Geruch
5	Alkoholischer Geruch; Muffiger Geruch; Schwacher G...
6	Alkoholischer Geruch; Schwacher Geruch
7	Aminartiger Geruch
8	Aminartiger Geruch; Ammoniakgeruch
9	Aminartiger Geruch; Fast geruchlos
10	Aminartiger Geruch; Fischgeruch
11	Aminartiger Geruch; Schwacher Geruch
12	Ammoniakgeruch
13	Ammoniakgeruch; Fischgeruch
14	Ammoniakgeruch; Schwacher Geruch
15	Ananasgeruch
16	Angenehmer Geruch
17	Angenehmer Geruch; Acetongeruch
18	Angenehmer Geruch; Alkoholischer Geruch
19	Angenehmer Geruch; Alkoholischer Geruch; Süßlicher ...
20	Angenehmer Geruch; Aromatischer Geruch
21	Angenehmer Geruch; Aromatischer Geruch; Schwach...
22	Angenehmer Geruch; Blumenduft
23	Angenehmer Geruch; Blumenduft; Fruchtartiger Geruch
24	Angenehmer Geruch; Charakteristischer Geruch
25	Angenehmer Geruch; Etherischer Geruch
26	Angenehmer Geruch; Etherischer Geruch; Schwacher ...
27	Angenehmer Geruch; Fast geruchlos
28	Angenehmer Geruch; Fruchtartiger Geruch
29	Angenehmer Geruch; Fruchtartiger Geruch; Medizinale...
30	Angenehmer Geruch; Kampfergeruch; Pfefferminzähnli...
31	Angenehmer Geruch; Medizinaler Geruch

Quelle: eigene Darstellung, SoftwareOne.

Die Betrachtung der Werteräume des ChemInfo Datensatzes zeigt deutlich, dass die Standardisierung der Eingabewerte gering ausfällt. Was konkret bedeutet, dass das System hauptsächlich Freitextfelder zur Beschreibung von Merkmalen zulässt.

3.4 Zusammenfassung der Analyse des Datenbestands

Die Voranalyse der im ChemInfo-System vorhandenen Daten zur Konzeptionierung und Umsetzung eines für das ChemInfo-System zugeschnittenen Deep Learning-Modells legt einige Problematiken offen. Neben den stark lückenhaft befüllten Merkmalen zeigen sich zusätzliche Herausforderungen in der fehlenden Standardisierung der Eingabewerte. Aus diesem Grund wurden in Rücksprache mit dem Umweltbundesamt alternative Ansätze besprochen und verabschiedet. Im Wesentlichen orientiert sich das alternative Vorgehen an den im ChemInfo-System vorhandenen Strukturinformationen, welche in Form von MOL-Dateien vorliegen. Auf Basis dieser Informationen soll eine Evaluierung des präferierten Deep Learning-Modell-Ansatzes (KI) auf Basis von SMILES Codes durchgeführt werden. Informationen zum implementierten Modellansatz befinden sich in Abschnitt 0.

Die wesentlichen durchgeführten operativen Schritte sind im Folgenden aufgelistet:

- Überführung des Datenbestands in ein analytisch verwertbares Format durch Aufbau und Anwendung einer ETL-Pipeline.
- Analyse der im ChemInfo-System vorhandenen Strukturinformationen (MOL-Dateien) zur Verwertbarkeit im Zusammenhang mit KI-Ansätzen und Erzeugung von SMILES Codes und Nutzung von externen Quellen zur Anreicherung zusätzliche SMILES Codes.
- Analyse des Datenbestands hinsichtlich der Verfügbarkeit, der Werteverteilung und des Wertespektrums.

Die im oberen Kapitel beschriebenen Punkte dienen als Grundlage für die im folgenden aufgeführten Methoden zur Verbesserung des Befüllungsgrades.

4 Methoden zur Verbesserung des Befüllungsgrads

4.1 Informationsextraktion aus Freitextfeldern mittels Textmining Methoden

Zur Befüllung von Stoffmerkmalen können neben Machine-Learning-basierten Ansätzen bzw. Prognoseverfahren ebenfalls Methoden angewendet werden, um Informationen auf direktem Weg aus bestehenden Merkmalsdaten abzuleiten. Deshalb soll im Rahmen einer Potenzialanalyse geprüft werden, ob sich mit Textmining-Verfahren implizite Informationen aus Feldern extrahieren lassen, um diese anschließend in inhaltlich passende Zielfelder zur Steigerung des Befüllungsgrades zu übertragen.

Zur Evaluierung dieses Ansatzes wurden ChemInfo-Felder mit Freitextcharakter, also geringer Standardisierung des Wertespektrums, und umfangreichem bzw. gemischtem Inhalt ausgewählt und in Bezug auf geeignete Informationen für andere Merkmale untersucht. Es wurden prototypische Methoden zur Extraktion relevanter Inhalte implementiert und versucht, diese den existierenden Werten in inhaltlich äquivalenten Zielfeldern zuzuordnen.

4.1.1 Auswahl der Freitextfelder zur Informationsextraktion

Im Folgenden wird der Ablauf beschrieben, wie einige potenzielle ChemInfo Felder, zur Prüfung der Anwendung von Textmining Methoden zur Informationsextraktion, ausgewählt wurden.

Die Ausgangsbasis der Feldauswahl ist die Analyse der Befüllung und der Werteverteilung der einzelnen ChemInfo Felder, welche im Abschnitt 4.3 vorgestellt wurde. Im mitgeltenden Dokument „Analyse Merkmalsverteilung“ ist auf dem Arbeitsblatt „Summary“ die Liste aller, in die Azure SQL Datenbank überführten Felder inklusive verschiedener Metriken, wie z. B. die Anzahl von Substanzen (ChemInfo Ids) oder die Anzahl verschiedener Werte pro Feld, dargestellt. Diese Feldliste wurde durch Anwendung folgender Filter grob eingegrenzt:

- Textfeld (W_Typ = Text)
- Keine Subsetfelder bzw. additiven Felder (Untergruppe = leer)
- Feld ist für mehr als 10 Stoffe befüllt (count_nonnull_id > 10)
- Feld enthält mehrere verschiedene Werte (count_distinct > 1)

Nach dieser Eingrenzung verbleiben ca. 350 Felder, aus welchen Kandidaten zur Informationsextraktion ausgewählt werden können. Dazu wurde eine Stichprobe wie folgt aus dem Bestand der verbleibenden Felder gezogen. Die Felder wurden nach, den von den Chemiker*innen des Fraunhofer Instituts bestimmten, Relevanzen (siehe Abschnitt 3.2.2) sowie innerhalb der Relevanzen nach Anzahl befüllter Substanzen sortiert. Anschließend wurden aus jeder Relevanz die zehn Felder, welche keine Identifikationsmerkmale (z. B. Namen oder CASRN) darstellen, mit der größten Anzahl befüllter Stoffe ausgewählt. Da bereits nach dem ersten Filterprozess weniger als zehn Felder mit einer Relevanz von 1 übrig waren, ergibt sich eine Stichprobe von 27 Feldern aus diesen die tatsächlich zu verwendenden Felder ausgewählt werden.

Für die finale Auswahl der Felder, auf welche Textmining Methoden angewendet werden, wurde eine qualitative bzw. manuelle Prüfung der Feldinhalte in der Azure SQL Datenbank für jedes der 27 Kandidatenfelder durchgeführt. Im Wesentlichen wurde dabei überblicksmäßig das Wertespektrum der einzelnen Felder betrachtet und eine Einschätzung über die

Weiterverwendbarkeit der enthaltenen Informationen zur Extraktion und ggf. Befüllung anderer Felder abgegeben. Daraus resultierend erwiesen sich fünf der 27 vorausgewählten Kandidaten als vielversprechend:

- STBE.STBE
- VERWAS.VERWAS
- GFRXREA.GFRXREA
- GFBR.GFBR
- BBLOEMI.BBLOEMI

Aus der Datenvoranalyse (siehe Abschnitt 3.3) war uns ein weiteres Feld bekannt, welches ähnlich wie Feld STBE.STBE ebenfalls Informationen hinsichtlich stofflicher Eigenschaften enthält und somit zusätzlich zur Anwendung von Textmining Methoden bestimmt wurde., obwohl es in der ursprünglichen Stichprobe der Felder nach Filterung nicht enthalten war:

- ERSCHB.ERSCHB

Weiterhin wurden zwei zusätzliche Felder durch das UBA vorgeschlagen, sodass diese ebenfalls ergänzend in die Liste der Felder zur Informationsextraktion aufgenommen wurden:

- FREIEMP.FREIEMP
- FREIBIN.FREIBIN

unterhalb stellt zusammenfassend die Liste der 27 Kandidaten inklusive der nachträglich ergänzten Felder dar. Die acht ausgewählten Felder zur prototypischen Implementierung von Textmining-Methoden zur Informationsextraktion sind rot eingefärbt. Außerdem sind in der Tabelle die Anzahl der Stoffe mit mindestens einem vorliegenden Wert im jeweiligen Feld und die Anzahl verschiedener Werte bzw. Ausprägungen des Feldes dargestellt. Die Spalte Kommentar enthält Notizen zur Verwendbarkeit oder bzgl. Problemstellungen im Fall einer Verwendung des Feldes zur Informationsextraktion.

Tabelle 3: Kandidatenliste Informationsextraktion

Kurzbezeichnung	Anzahl befüllter Stoffe	Anzahl verschiedener Werte	Textmining anwendbar Einschätzung	Kommentar
LIFO.LIFO	39645	34130	Eventuell	Aufteilung organischer/ anorganischer Stoffe
FINF.SUFO	37478	20708	Eventuell	Aufteilung organischer/ anorganischer Stoffe
MAKDFG.DD	383	297	Ja	Extraktion Werte + Konvertierung in Zahlen, Skalen
LOES.L	1649	224	Ja	Schreibfehler und Verallgemeinerung von Einträgen
EG1005_09.SUM	182	82	Eventuell	Aufteilung organischer/ anorganischer Stoffe
EU517_14.CHF	55	54	Eventuell	Aufteilung organischer/ anorganischer Stoffe
REDOX.L	133	16	Eventuell	Sehr kleine Stoffaufzählung + fehlende Standardisierung
GFRXREA.GFRXREA	11566	3010	Ja	Extraktion Gefahren, Stoffeigenschaften + fehlende Standardisierung
GFBR.GFBR	11536	367	Ja	Extraktion Gefahren, Stoffeigenschaften + fehlende Standardisierung
FB.FB	11147	1511	Ja	Extraktion Stoffeigenschaften + fehlende Standardisierung
VERWAS.VERWAS	8437	337	Ja	Extraktion Gefahren, Stoffeigenschaften + fehlende Standardisierung
GFEXDIR.GFEXDIR	10573	64	Ja	Extraktion Gefahren, Stoffeigenschaften + fehlende Standardisierung
FINF.ELZR	37235	579	Nein	Nur Auflistung von einzelnen Elementen und deren Vorkommen/ Anzahl
STBE.STBE	15417	699	Ja	Extraktion Gefahren, Stoffeigenschaften
GFRXZER.GFRXZER	11504	131	Ja	Extraktion Gefahren, Stoffeigenschaften
GFEXIND.GFEXIND	11178	45	Ja	Extraktion Gefahren, Stoffeigenschaften

Kurzbezeichnung	Anzahl befüllter Stoffe	Anzahl verschiedener Werte	Textmining anwendbar Einschätzung	Kommentar
AZ.AZ	9734	3	Nein	Nur Konvertierung der Textdarstellung in Zahlendarstellung
BBLOEMI.BBLOEMI	13578	1000	Ja	Extraktion Gefahren, Stoffeigenschaften + fehlende Standardisierung
BBBRAND.BBBRAND	11879	756	Eventuell	Extraktion Gefahren
FREIEMP.FREIEMP	12005	5110	Eventuell	Auf Empfehlung von UBA - Extraktion Gefahren, Stoffeigenschaften
GGALL.GGALL	12274	1957	Ja	Extraktion Gefahren, Stoffeigenschaften
ENTEMP.ENTEMP	12112	470	Eventuell	Extraktion Gefahren
FREISCHUTZ.FREISCHUTZ	12000	792	Eventuell	Extraktion Gefahren, Stoffeigenschaften
EHHAUT.EHHAUT	11740	361	Eventuell	Extraktion Gefahren
BBSCHUTZ.BBSCHUTZ	11996	616	Eventuell	Extraktion Gefahren, Stoffeigenschaften
EHAUGE.EHAUGE	11753	88	Ja	Extraktion Gefahren
EHINH.EHINH	11740	48	Eventuell	Extraktion Gefahren + fehlende Standardisierung
ERSCHB.ERSCHB	750	194	Ja	Extraktion Stoffeigenschaften / Merkmal in STBE.STBE überführen
FREIBIN.FREIBIN	3007	75	Ja	Auf Empfehlung von UBA - Splitting von verketteten Werten in mehrere Einzelwerte pro Stoff möglich

Quelle: eigene Darstellung, SoftwareOne.

Im weiteren Projektverlauf wurde die Liste der zu verwendenden Felder später nach Rücksprache mit dem UBA im Jour Fix am 26.07.2022 weiter eingeschränkt, sodass eine vollständige Umsetzung der Informationsextraktion mittels Textmining für folgende Felder umgesetzt wurde:

- STBE.STBE
- ERSCHB.ERSCHB
- FREIEMP.FREIEMP
- FREIBIN.FREIBIN

Vollständige Umsetzung bedeutet in diesem Kontext, dass die Extraktion von Inhalten in Databricks Notebooks umgesetzt wurde und extrahierte Informationen zum Mapping auf bestehende Felder zur weiteren Befüllung an das UBA zur Beurteilung übergeben wurden (siehe mitgeltendes Dokument „Textmining_Ergebnismengen“). Außerdem wurde die Speicherung extrahierter Informationen in der Azure SQL Datenbank gespeichert.

4.1.2 Umsetzung der Informationsextraktion

Die operativen Schritte zur Informationsextraktion lassen sich im Wesentlichen durch die folgenden Punkte beschreiben und wurden für die vier ausgewählten Merkmalsfelder identisch umgesetzt.

4.1.2.1 Allgemeine Textbereinigung

Zu Beginn wird die Qualität der Textinformationen geprüft, um Probleme wie Groß- & Kleinschreibung, Rechtschreibung, Wortabkürzungen, Anwendung unterschiedlicher Sprachen und fehlerhafte Symboliken erkennen zu können. Die ermittelten Problemquellen, werden anschließend mittels implementierter Transformations- & Bereinigungsmethoden für den Nachfolgeprozess aufgelöst.

- Rechtschreibung: Wird im Fall des Auftretens als Error angezeigt, benötigt aber manuelle Sammlung als Eintrag in der Error-Prüfliste.
- Groß- & Kleinschreibung: Auflösung durch vollständige Texttransformation in Kleinschreibung.
- Unterschiedliche Sprachen: Direkte Inklusion der Fehlwerte in alle Extraktionsprozesse, da dieser Problemfall sehr selten vorkam.
- Sonderzeichen: Sonderzeichen werden in den einzelnen Prozessschritten aufgelöst und Überbleibsel am Ende, wenn kein Informationsverlust entsteht, entfernt.
- Abkürzungen: Um eine Satztrennung zu realisieren, wurde ein Mapping der Abkürzungen in die zugehörigen Vollschreibweisen vorgenommen.

4.1.2.2 Strukturelle und inhaltliche Analysen

Danach werden verschiedene Symbolzählungen bspw. von Sonder- oder Satzzeichen wie bspw. Punkte, Kommas, Semikolon, Klammern o. ä. durchgeführt, um Textstrukturmerkmale wie Label, Gruppierungselemente, Auflistungen, Sätze usw. zu identifizieren. Anschließend erfolgt die Erzeugung und Betrachtung von Ngrams der Länge drei (Trigrams) sowie einer Wortzählung.

Hier gilt es zwei wichtige Sonderfälle zu beachten:

1. Beachtung der Marker für Negationen, welche gegensätzliche Informationen anzeigen können.
2. Beachtung von Synonymen die ggf. für gleiche Aussagen stehen.
   Bspw. Pulver oder Kristalle, welche den Aggregatzustand „fest“ definieren.

Alle genannten Informationen geben einen Überblick über den Informationsgehalt des Merkmalsfeldes, sowie mögliche Such- und Gruppierungselemente (bspw. Wortstamm oder Verneinung).

4.1.2.3 Inhaltsextraktion

Die Ergebnisse aus den zuvor genannten Analyseschritten werden je nach Anwendbarkeit zur Ermittlung möglicher Stoffeigenschaften herangezogen. Dafür wird für jedes der gefundenen Such- & Gruppierungselemente eine Regular Expression (RegEx) konstruiert und dessen Resultate kontrolliert. Falls möglich, werden mehrere Elemente in einer RegEx zusammengefasst, wenn sie eine ähnliche Ergebnismenge abbilden. Sollten RegEx nicht direkt anwendbar sein, da bspw. eine zu große Streuung oder zu wenige Informationen für ein entsprechendes Element vorliegen, so wird alternativ eine Wort-, Phrasen- oder Volltextextraktion angewendet. Um die alternativen Extraktionsprozesse zu realisieren, werden auffällige Sonderinformationen in einer dem Such- & Gruppierungselement zugehörigen Liste zugeordnet. Anschließend wird entweder direkt auf der Liste oder auf einer aus expliziten Gruppen bestehenden RegEx (basierend auf Listenelementen) gesucht.

4.1.2.4 Transformation in Datenbank speicherbare Informationen

In einigen Fällen, wenn die Ergebnismenge > 1 ist, werden die Ergebnisse als Liste ausgegeben.

Daher ist eine Transformation der Daten zur Speicherung in einer SQL-Datenbank notwendig.

Der Einfachheit halber werden alle Ergebnismengen für die nachfolgende Evaluation unverändert als Text oder konkatenierter Text bereitgestellt. Für die Transformation von Listenelementen zu einem Text, wird daher das Trennsymbol „||“ als Separator angewendet.

4.1.2.5 Naive Kontrolle des Feldinhalt-Bearbeitungs-Zustands

Um einen vereinfachten Überblick über den Bearbeitungsgrad (Grad der Prozessierung) eines Feldinhaltes zu bekommen, wird für jedes Feld eine Kopie angelegt, aus der Schritt für Schritt die extrahierten Elemente entfernt werden. Hier kann es zu Überschneidungen kommen, welche durch die Anordnung der Extraktionsreihenfolge und eine Vollersetzungsregel naiv abgedeckt wird. Dieser naive Ansatz hat keinen Anspruch auf Vollständigkeit und dient nur als hinreichendes Maß für den Bearbeitungsgrad eines Feldinhaltes.

Hinweis: Wird nicht bei FREIEMP.FREIEMP angewendet, da hier nur eine teilweise inhaltliche Extraktion aufgrund der Merkmalsfeldkomplexität durchgeführt wird.

4.1.3 Zusammenfassung der Ergebnisse

Für die vier in Abschnitt 4.1.1 genannten Felder wurden in Azure Databricks Notebooks Textmining-Methoden zur Informationsextraktion implementiert. Auf Basis der Feldinhalte jedes Feldes wurden verschiedene Arten von Informationen identifiziert, welche extrahiert und in der Azure SQL Datenbank gespeichert wurden. Die extrahierten Informationen können entweder zusätzlich als neue Felder oder zur weiteren Befüllung von bestehenden Feldern verwendet werden.

Im Folgenden werden in Tabelle 4 alle gefundenen Informationen und die Häufigkeit ihres Auftretens in den untersuchten Feldern aufgelistet. Es gibt dabei verschiedene Informationen, wie zum Beispiel:

- Stoffeigenschaften (z. B. Aggregatzustand, Toxizität),
- Identmerkmale (z. B. CASRN),

- Anweisungen und Gefahreninformationen (z. B. zu verwendende Bindemittel, Gefahrenbereiche) und
- Metainformationen (z. B. Vorhandensein einer Negation im Text).

Außerdem ist zu berücksichtigen, dass die angegebenen Häufigkeiten nicht mit der Anzahl von Stoffen gleichzusetzen sind. Die Häufigkeit entspricht vielmehr der Häufigkeit des Auftretens der jeweiligen Information im entsprechenden Feld. So kann für einen Stoff bspw. im Feld STBE.STBE mehrere Informationen zur Brennbarkeit gefunden werden, wobei jede einzelne Information gezählt wurde.

Jede der aufgeführten Informationen kann mehrere Ausprägungen annehmen (z. B. Aggregatzustand: fest, flüssig, gasförmig). Eine detaillierte Darstellung der möglichen Ausprägungen auf Stoff- und Sachverhaltsebene befindet sich im mitgeltenden Dokument „Textmining Ergebnismengen“. Dieses Dokument wurde dem UBA bereits zur Prüfung vorgelegt und kann u. a. dazu verwendet werden, um die extrahierten Informationen und deren Ausprägungen bestehenden ChemInfo-Felder zur weiteren Befüllung zuzuordnen.

Es wurde mit dem Umweltbundesamt besprochen, dass damit die Arbeiten in puncto Informationsextraktion aus Freitextfeldern mittels Textmining abgeschlossen sind.

Tabelle 4: Häufigkeit des Vorkommens extrahierter Informationen mittels Textmining

Enthaltene Information	ERSCHB.ERSCHB	STBE.STBE	FREIBIN:FREIBIN	FREIEMP.FREIEMP
Bindemittel	0	0	5.455	18.464
Bindemittel Type	0	0	1.005	4.093
Zündquellen	0	0	0	1.209
Nachschlagewerke	0	0	0	4.566
Zu Vermeiden	0	0	0	17.532
Negation	0	0	0	13.932
Beispiele	0	0	0	2
Verhalten mit Wasser	0	0	0	10.521
Verstäuben	0	0	0	117
Freigewordene Stoffe	0	0	0	3.840
Gefahren	9	0	0	115
Gefahrenbereiche	0	0	0	4.636
Zusatzinfos	0	0	883	0
Nicht Bindemittel	0	0	1.146	0
Bindemittel Eigenschaften	0	0	1.583	0
Bindemittel Bedingte Gefahr	0	0	157	0

Enthaltene Information	ERSCHB.ERSCHB	STBE.STBE	FREIBIN:FREIBIN	FREIEMP.FREIEMP
Nicht Bindemittel-Bindemittel-Hinweise	0	0	735	0
Partikelgröße	0	77	0	0
Technische Info	0	20	0	0
Aggregatzustand	776	24.335	0	0
Flammpunkt	96	0	0	0
Explosiv	1	46	0	0
Brennbar	522	640	0	0
Brandfördernd	20	67	0	0
Toxisch	247	0	0	0
Korrosiv	229	0	0	0
Reizend	5	0	0	0
Reaktiv	51	7	0	0
Gesundheitsschädlich	0	0	0	0
Hygroskopisch	59	256	0	0
Zerfließlich	18	0	0	0
Verflüssigt	28	23	0	0
Flüchtig	2	126	0	0
Verdichtet	23	0	0	0
PH-Wert	2	0	0	0
Rauchend	2	87	0	0
Komprimiert	0	34	0	0
Hydrolysiert	0	31	0	0
Polymerisiert	0	14	0	0
Amorph	0	82	0	0
Bedingung	118	15	0	0
Sonderbedingung	2	0	0	0
UN-Nummer	1	0	0	0
Stickoxide	6	0	0	0
Stoff/Gemische	220	9.643	0	0
Zersetzung	0	13	0	0
Viskosität	0	278	0	0
Empfindlich	1	76	0	0

Enthaltene Information	ERSCHB.ERSCHB	STBE.STBE	FREIBIN:FREIBIN	FREIEMP.FREIEMP
Stabilisiert	0	28	0	0
CASRN	0	5	0	0
Alternative	0	253	0	0
Metalleigenschaft	0	136	0	0
Sublimiert	0	20	0	0
Produktinfo	0	26	0	0
Peroxidbildung	0	4	0	0
Radioaktiv	0	33	0	0
Beschreibung	38	239	0	0
Gesundheitsschädlich	64	0	0	0

Quelle: eigene Darstellung, SoftwareOne.

4.2 Chemprop Deep Learning-Modell zur Vorhersage von Stoffeigenschaften

Im Folgenden soll untersucht werden, ob es möglich ist, auf Basis der Strukturinformationen, welche in Form von MOL-Dateien in der ChemInfo-Datenbank vorliegen, mögliche Deep Learning-Ansätze zu verfolgen.

4.2.1 Modellansatz und Informationsquellen

Die Zielstellung bestand darin eine Deep-Learning basierte Vorgehensweisen zu finden, um die Befüllung von ChemInfo-Datenbankfeldern, welche zur Erstellung der Datenblätter verwendet werden, zu steigern. Im Rahmen der Recherche wurde ein Modell mit der Bezeichnung Chemprop ausfindig gemacht. Das Modell wurde am MIT entwickelt, auf einem Datensatz von Molekülstrukturen in Form von SMILES Codes vortrainiert und kann mittels Transfer Learning zur Vorhersage von Molekül- bzw. Stoffeigenschaften auf der Basis von Datensätzen aus dem ChemInfo-System angepasst bzw. weitertrainiert werden.

Der Vorteil des Transfer Learning-Ansatzes besteht darin, dass mit Re-Training auch für kleinere, domänenspezifische Datensätze prädiktive Modelle mit guter Qualität erzeugt, werden können, was mit selbständig neu entwickelten Modellen auf Basis eigener Datenbestände oft nicht oder nur mit mangelnder Prognosequalität zu realisieren ist.

Darüber hinaus gibt es bereits eine frei verfügbare Python Bibliothek zur Anwendung des Chemprop-Modells. Dies mindert den Entwicklungsaufwand, da einige Kernprozesse wie zum Beispiel das Modelltraining bereits in Form von Python Funktionen vorliegen. Außerdem bietet die Chemprop-Bibliothek einige weitere nützliche Optionen. Neben SMILES Codes, als Repräsentation von Molekülstrukturen, können zur Steigerung der Prognosegüte selbst bereitgestellte Informationen als Features (erklärende Variablen), wie bspw. funktionelle Gruppen in das Transfer Learning des Chemprop-Modells einbezogen werden.

Die wesentlichen Quellen, auf die sich die Recherche bezieht, sind im Folgenden ausgewiesen:

- Chemical Predictions with 3 lines of code | by Mathias Gruber | Towards Data Science[9]
- Analyzing Learned Molecular Representations for Property Prediction | Journal of Chemical Information and Modeling (acs.org)[10]
- GitHub - chemprop/chemprop: Message Passing Neural Networks for Molecule Property Prediction[11]
- Chemprop — chemprop 1.5.1 documentation[12]
- Chemprop++ - Google Präsentationen[13]
- A Deep Learning Approach to Antibiotic Discovery: Cell[14]

4.2.2 Technische Umsetzung

Im Folgenden wird beschrieben, wie die Verwendung des Chemprop-Modells zur Befüllung bzw. zur Vorhersage unbekannter Werte in ChemInfo-Merkmalsfeldern technisch umgesetzt wurde.

4.2.2.1 Jupyter Notebook zum Modelltraining, Modellevaluation und Ergebnisspeicherung

Das Kernelement zum Modelltraining, der Modellevaluation und der Erzeugung von Vorhersagen unbekannter Werte ist ein Jupyter Notebook (Chemprop.ipynb). Der Vorteil von Jupyter Notebooks gegenüber gewöhnlichen Python Skripten besteht darin, dass neben ausführbarem Python Code direkt Ergebnisse als Tabellen oder Visualisierungen im Notebook ausgegeben werden. Der Python Code wird dabei auf verschiedene Zellen aufgeteilt und falls eine ausgeführte Zelle ein Ergebnis liefert, wird dieses unterhalb der Zelle („inline“) angezeigt. Neben Zellen, welche Python Code enthalten, können außerdem Zellen mit statischen Inhalten (meist Text z. B. für Überschriften) im Markdown Format im Notebook verwendet werden. Die Kombination aus statischen Inhalten und Ergebnissen ist eine schlanke Möglichkeit Ergebnisberichte zu erzeugen. Es muss dafür kein separates Framework wie z. B. LaTeX verwendet werden. Des Weiteren können Jupyter Notebooks automatisiert in verschiedene Formate konvertiert werden. Hier wurde HTML gewählt, da es im Gegensatz zum PDF-Format keine zusätzlichen Installationen erfordert, was zu technischen Problemen bei Ausführung in der Cloud führen könnte. Außerdem lassen sich in HTML-Dokumenten auch große bzw. breite Tabellen ohne zusätzlichen Aufwand abbilden, was in PDF-Dokumenten auf Grund der Seitenumbrüche und der vorgegebenen Seitenbreite nicht der Fall ist. Es werden bei jeder Notebook-Ausführung zwei Versionen des HTML-Berichts – also jeweils mit und ohne Python Code – erzeugt.

[9] Gruber, Mathias (2021), Chemical Predictions with 3 lines of code [online], Chemical Predictions with 3 lines of code | by Mathias Gruber | Towards Data Science , letzter Abruf am 14.12.2022.

[10] Yang, Kevin et al. (2019), Analyzing Learned Molecular Representations for Proberty Prediction [online]

Analyzing Learned Molecular Representations for Property Prediction | Journal of Chemical Information and Modeling (acs.org) , letzter Abruf am 14.12.2022

[11] GitHub (2022), chemprop Molecular Proberty Prediction [online] https://github.com/chemprop/chemprop#molecular-property-prediction, letzter Abruf am 14.12.2022

[12] Swanson, Kyle et al. (2020), chemprop 1.5.1 documentation [online] https://chemprop.readthedocs.io/en/latest/index.html, letzter Abruf am 14.12.2022

[13] Swanson, Kyle et al. (ohne Datum), Chemprop++ [online], Chemprop++ - Google Präsentationen, letzter Abruf am 14.12.2022

[14] Stokes, Jonathan M. et al. (2020), A Deep Learning Approach to Antibiotic Discovery [online] A Deep Learning Approach to Antibiotic Discovery: Cell, , letzter Abruf am 14.12.2022

Während einer Ausführung bzw. in einem Durchlauf des Jupyter Notebooks werden folgende Vorgänge durchgeführt:

- Ein Chemprop-Modell wird gemäß der spezifizierten Steuerungsparameter trainiert und evaluiert.
- HTML-Berichte (mit und ohne Python Code) werden erzeugt und gemeinsam mit den zugrundeliegenden Python Ergebnisdaten und Modellen in einem Ausgabeordner gespeichert.
- Für Substanzen ohne bekannten Wert im jeweiligen Zielmerkmalsfeld wird das Modell angewendet, um Vorhersagen zu erzeugen und diese zusammen mit den berechneten Wahrscheinlichkeiten zur weiteren Verwendung in der Azure SQL Datenbank zu speichern.

Wie üblich müssen zur Ausführung des Notebooks die benötigten Bibliotheken in der verwendeten Python Umgebung installiert werden (sowie bei lokaler Verwendung als auch in der Cloud). Die Bibliotheken inkl. deren Versionen sind in einem requirements.txt im Git Repository hinterlegt. Für eine schnellere Durchführung der Machine-Learning-Prozesse (Training und Prediction), können Grafikbeschleuniger verwendet werden, falls verfügbar. Dafür muss Nvidea Cuda auf dem verwendeten Rechner installiert sein, um PyTorch zu befähigen Grafikprozessoren zu nutzen. PyTorch ist das zugrundeliegende Python Deep Learning-Framework, auf welchem das Chemprop-Modell basiert.

Außerdem wurden einige, teils mehrfach verwendete Python Funktionen in ein Python Skript (ChempropClasses.py) ausgelagert, um den Code im Notebook übersichtlicher zu halten.

4.2.2.2 Python Skripte und Excel Datei zur Spezifikation von Steuerungsparametern zur Durchführung multipler Trainingsläufe in einem Satz

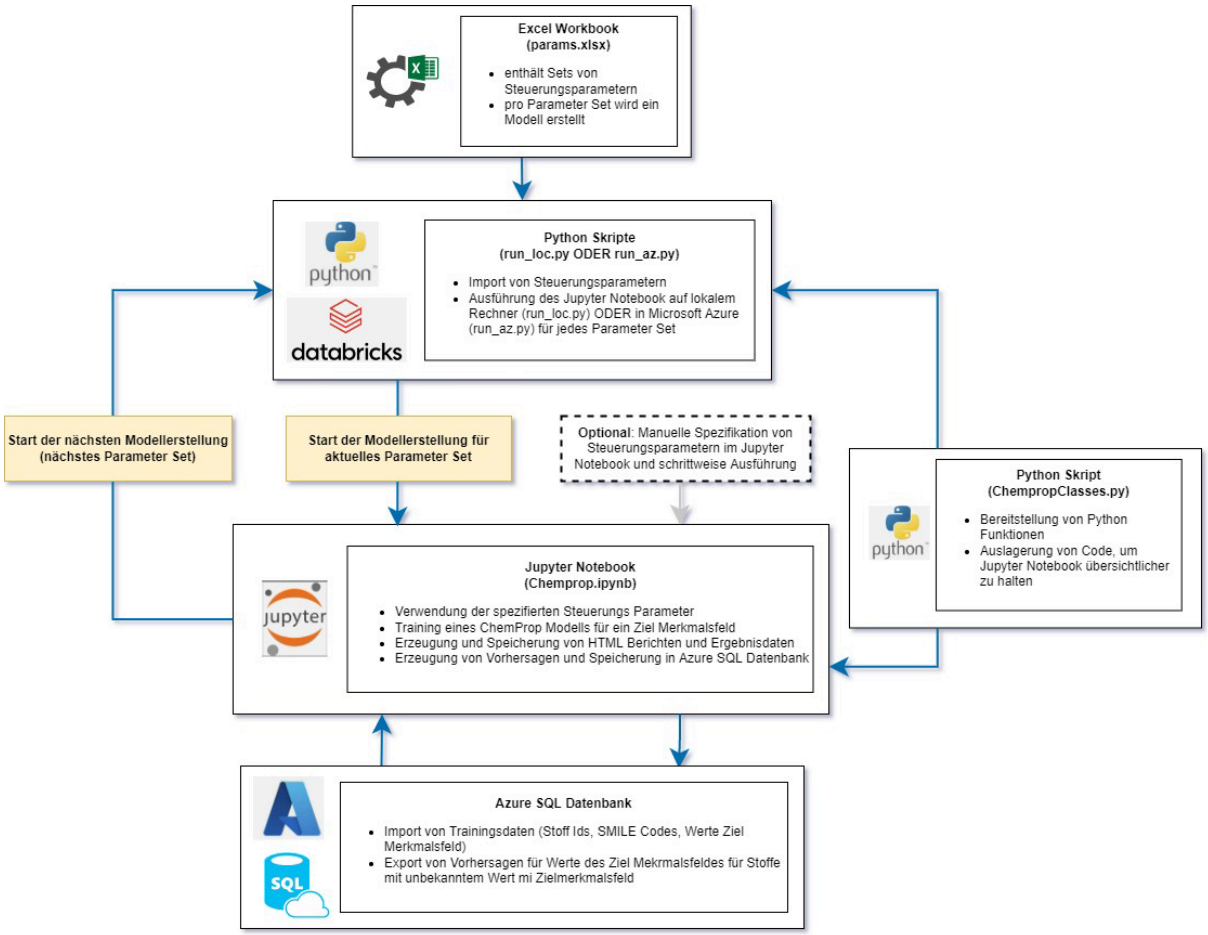
Die Ausführung des Jupyter Notebooks kann wie folgt stattfinden:

1. Durch manuelle Einzelausführung des Jupyter Notebook bzw. der Code Zellen im Notebook auf einem lokalen Rechner zur Code Entwicklung bzw. Weiterentwicklung oder zu Demonstrationszwecken.
2. Mittels Python Skript run_loc.py zur automatisierten Mehrfachausführung (sequenziell) des Jupyter Notebooks auf einem lokalen Rechner zur Generierung mehrerer Modelle in einem Satz.
3. Mittels Python Skript run_az.py zur automatisierte Mehrfachausführung (parallelisiert) des Jupyter Notebooks in Microsoft Azure (Cloud) zur Generierung einer Vielzahl von Modellen in möglichst kurzer Zeit.

Im zweiten und dritten Fall wird anstelle des Jupyter Notebooks ein Python Skript ausgeführt, welches die Sets von Steuerungsparameter (weitere Informationen folgen im nächsten Abschnitt 4.2.3) aus einem Microsoft Excel Workbook („params.xlsx“) importiert und anschließend für jedes vorgegebene Parameter Set das Jupyter Notebook ausführt bzw. ein Modell generiert. Die einzelnen Steuerungsparameter, deren Bedeutung und Funktion werden direkt in den HTML-Berichten erläutert. Für die aufrufenden Python Skripte gibt es je eine Version zur Ausführung auf einem lokalen Rechner oder in Microsoft Azure. Letztere Variante wurde, wie bereits die ETL-Prozesse, in der Azure Data Factory automatisiert und basiert auf einer Azure Databricks Python Umgebung.

Der technische Prozess der Modellerstellung lässt sich wie folgt visuell beschreiben (s. Abbildung 9):

Abbildung 9: Technische Umsetzung der Erstellung von Chemprop-Modellen



Quelle: eigene Darstellung, SoftwareOne.

4.2.3 Erläuterungen zur fachlichen Umsetzung der Modellerstellung

Im Folgenden sollen wesentliche Aspekte der fachlichen Umsetzung zur Erstellung von Chemprop-Modellen erläutert werden. Außerdem wird begründet, warum bestimmte Vorgehensweisen gewählt wurden.

4.2.3.1 Informationen, welche in den HTML-Berichten enthalten sind

Die folgenden Ausführungen sollen als Zusatzinformationen zu den Beschreibungen dienen, welche bereits in den HTML-Berichten der einzelnen Modelle (mitgeltendes Dokument Chemprop Modelle) enthalten sind. Für folgende Informationen wird daher auf die HTML-Berichte verwiesen:

- Bedeutung der Steuerungsparameter,
- Beschreibung von Ergebnistabellen und Visualisierungen und
- grundsätzlich fachliche Informationen, welche bereits erläutert wurden.

4.2.3.2 Informationen zum Abruf der Modelldaten aus SQL-Datenbank

Einzelinhaltsstoffe und deren Merkmale, liegen als Ergebnis der beschriebenen ETL-Prozesse in einer Tabelle („Substances“) der Azure SQL Datenbank vor. Die Daten bilden die Grundlage zur Entwicklung der Chemprop-Modelle. Für jedes Feld, welches mittels Chemprop aufgefüllt

werden soll, werden deshalb im ersten Schritt der Ausführung des Jupyter Notebooks die entsprechenden Daten aus der Datenbank in den Python Prozess importiert. Die Daten enthalten SMILES Codes, Ausprägungen des Zielmerkmals und vorkommende funktionelle Gruppen als zusätzliche Inputvariablen, wenn dies durch den entsprechenden Steuerungsparameter aktiviert ist. Theoretisch wäre es möglich, weitere Merkmalsfelder als Inputvariablen zu verwenden. Da jedoch der Befüllungsgrad der meisten Felder zu gering ist, aber für jeden Stoff ein Wert für jede Inputvariable vorliegen muss, würde dies zu Problemen führen. Würde man bspw. ein chemisch physikalisches Merkmalsfeld verwenden, müssten die fehlenden Werte entweder mit fiktiven bzw. falschen Werten (z. B. 0, Mittelwert, o.ä.) aufgefüllt werden oder die entsprechenden Stoffe sowohl vom Training als auch der Erzeugung von Prognosen ausgeschlossen werden. Beide Optionen sind im konkreten Anwendungsfall nicht zielführend, da das Problem fehlender Werte in den zusätzlichen Inputvariablen für einen großen Teil der Stoffe auftreten würde. Dies ist bei Verwendung der funktionellen Gruppen nicht der Fall, da diese aus den SMILES Codes abgeleitet werden, welche für den Großteil der Stoffe vorliegen.

Für alle Merkmale wurden für diesen Zweck SQL Statements erstellt, welche die Daten für Modelltraining und zur Evaluation vorbereiten und importieren. Folgende vorbereitenden Schritte werden in dem Zuge durchgeführt:

- Einschränkung der Daten auf Sachverhalte des GSA-Datenbestands zur Erreichung eindeutiger Zuordnung von Werten des jeweiligen Zielmerkmals zu Stoffen. Jeder Stoff darf nur einmal im Modellierungsdatensatz vorkommen und für jeden Stoff darf nur eine Merkmalsausprägung vorliegen, da das Modell anderenfalls auf widersprüchlichen oder mehrdeutigen Informationen trainiert werden würde. Eine Ausnahme gibt es für Merkmal AZ.AZ, welches nicht im GSA-Datenbestand enthalten ist. Dort wurde die Filterung auf GSA-Sacherhalte nicht durchgeführt, sondern lediglich Stoffe, bei denen mehrere Aggregatzustände vorliegen, ausgeschlossen.
- SMILES Codes werden als Inputvariable (Features) in den Modellen genutzt. Jeder SMILES Code sollte im Trainingsdatensatz nur einmal vorliegen, da dem Modell ansonsten widersprüchliche Informationen übergeben werden, wenn für den gleichen SMILES Code verschiedene Ausprägungen vorliegen. Im Rahmen der Implementierung ergab eine Analyse der SMILES Codes, ChemInfo-IDs und Registriernamen, dass es vorkommen kann, dass gleiche SMILES Codes auf chemische Substanzen mappen, die sich in ihren molekularen Eigenschaften teilweise deutlich unterscheiden. Beispiele hierfür sind Polymere (z. B. Polyethylenglykol), die sich in ihrer Kettenlänge unterscheiden oder anorganische Verbindungen, die sich im Hydratwassergehalt unterscheiden. Außerdem kann es vorkommen, dass bei Isomeren die Stereoinformationen nicht vorhanden sind. In diesen Fällen mappt ein SMILES Code auf verschiedene Eigenschaften eines Features (z. B. unterschiedliche Aggregatzustände für einen SMILES Code aufgrund unterschiedlicher Kettenlänge des Polymers). Eine Zählung ergab, dass es 740 SMILES Codes gibt, welche für mehrere ChemInfo IDs (Einzelinhaltsstoffe) vorliegen. In diesen Fällen wurde zufällig eine ChemInfo ID bzw. ein SMILES Code zur Verwendung im Trainingsdatensatz ausgewählt. Prinzipiell sollten diese Stoffe nur in sehr geringen Mengen im Datensatz vorkommen, da diese nur verwendet werden, wenn auch die jeweilige Zielvariable befüllt ist. Somit hat die zufällige Auswahl der Stoffe einen zu vernachlässigenden Einfluss auf die Modellergebnisse.
- Bei den Feldern HAZC.KB, HAZC.FZ, NFPA.BG, NFPA.GF, NFPA.RG repräsentieren die Ausprägungen Codes. Zur besseren Verständlichkeit der Inhalte der HTML-Berichte wurden den jeweiligen Codes die verbalisierten Bedeutungen angehängt. Die Bedeutungen haben die

mitwirkenden Chemiker*Innen vom Fraunhofer Institut bereitgestellt. Exemplarisch ist hier die Übersetzung für das Feld HAZC.KB dargestellt (s. Abbildung 10):

Abbildung 10: Beispiel zur Erweiterung der Ausprägungen der Zielvariable um verbalisierte Beschreibungen

```
'P': 'P - Violence - Full - Dilute',
'R':  'R -  - Full - Dilute',
'S':  'S - Violence - BA - Dilute',
'[S]':    '[S] -  - BA for fire only - Dilute',
'T':  'T -  - BA - Dilute',
'[T]':    '[T] -  - BA for fire only - Dilute',
'W':  'W - Violence - Full - Contain',
'X':  'X -  - Full - Contain',
'Y': 'Y - Violence - BA - Contain',
'[Y]': '[Y] -  - BA for fire only - Contain',
'Z': 'Z -  - BA - Contain',
'[Z]': '[Z] -  - BA for fire only - Contain'
```

Quelle: eigene Darstellung, SoftwareOne.

Informationen zu Trainings-, Test- und Validierungspartitionen

Für die Erstellung eines Modells (Training und Evaluation) wird ein Datensatz mit Inputvariablen (Features) auf Basis der Inhalte der Azure SQL Datenbank erzeugt. Weitere Informationen und ein Überblick über die Modelldaten sind in den HTML-Berichten

(mitgeltendes Dokument Chemprop Modelle) unter „Dataset Summary“ verfügbar.

Dieser Datensatz wird im weiteren Verlauf des Prozesses in Partitionen zerlegt. Die übliche Vorgehensweise der zufälligen Aufteilung sowie die Gruppierung von Minoritätsklassen als Vorbereitung sind ebenfalls in den HTML-Berichten beschrieben.

Die anteilige Verteilung der Modelldaten auf die einzelnen Partitionen kann Einfluss auf das trainierte Modell und damit Prognosequalität haben. Um dies untersuchen zu können wurden zur Kontrolle der Partitionsgrößen die Steuerungsparameter „val_dataset_size“ und „test_dataset_size“ implementiert. Die Parameter sind in der Tabelle der Steuerungsparameter in den HTML-Berichten beschrieben.

Die resultierenden Partitionen haben folgende Zwecke:

- **Trainingspartition (Train):**
  - Verwendung zum Training der Feed-Forward-Layer des vortrainierten Chemprop-Modells. Die Modellgewichte der einzelnen Knoten (Nodes) in jedem Layer werden über mehrere Epochen hinweg durch „feed forward“ und „backpropagation“ der Daten im Trainingsdatensatz durch das neuronale Netz bzw. Modell angepasst.
- **Validierungspartition (Val):**
  - Verwendung zur Ermittlung der optimalen Anzahl von Trainingsepochen. Jedes Mal, wenn ein Modell auf Basis der Trainingspartition trainiert wird (in jeder Iteration des Hyperparameter Tuning Prozesses oder beim Training des finalen Modells), wird nach jeder Epoche der entsprechende Loss (Abweichung zwischen Modelloutputs und Ausprägungen der Zielvariable) für die Validierungspartition berechnet. Nachdem das Modell für die spezifizierte Anzahl von Trainingsepochen (Steuerungsparameter) trainiert und validiert wurde, wird das Modell mit dem Status nach der Epoche mit dem geringsten Validierungs-Loss verwendet.

- Verwendung zur Validierung bzw. Berechnung der Zielmetrik nach jeder Iteration im Hyperparameter Tuning. Dies dient der Vermeidung einer übermäßigen Anpassung (Overfitting) der Hyperparameter an den Trainingsdatensatz und damit Verbesserung der Generalisierung des Modells. Dies geschieht nur, wenn Hyperparameter Tuning durch den entsprechenden Steuerungsparameter aktiviert ist. Weitere Informationen zum Hyperparameter Tuning, wie es in der Chemprop-Bibliothek implementiert ist, befinden sich in der Chemprop Dokumentation.

► **Testpartition (Test):**

- Verwendung zur Modellevaluation und Beurteilung der Prognosegüte. Die Testpartition repräsentiert „ungesehene" Daten, sprich Beobachtungen, welche nicht in Modelltraining oder Hyperparameter Tuning eingeflossen sind. Auf Basis der Modellprognosen für die Testpartition und den tatsächlichen Ausprägungen der Zielvariable der Testpartition werden in der Modellevaluation verschiedene Auswertungen durchgeführt, welche in den HTML-Berichten (mitgeltendes Dokument Chemprop Modelle) dargestellt sind.

- Verwendung zur Durchführung weiterführender Analysen. Auf Basis der Evaluationsergebnisse und Metriken, welche für die Testpartition ermittelt werden, können weiterführende Analysen durchgeführt werden. Zum Beispiel kann der Einfluss von Zufallseffekten auf die Prognosequalität quantifiziert werden. Weitere Erläuterungen hierzu befinden sich im Abschnitt 5.2.5 Analyse und Optimierung von Steuerungsparametern.

4.2.3.3 Umsetzung der Modellvalidierung

Im Folgenden Abschnitt werden die Umsetzung der Modellvalidierung und die Motivation des gewählten Ansatzes erläutert.

In der klassischen Kreuzvalidierung (k-Fold Cross-Validation) wird der Datensatz in mehrere (k = Anzahl der Teilmengen) Teilmengen („Folds") aufgeteilt. Jede Teilmenge repräsentiert eine Validierungspartition und die jeweils verbleibenden Teilmengen zusammen bilden die Trainingspartition. Für jede der k Validierungspartitionen wird anschließend ein Modell auf den verbleibenden Partitionen trainiert und auf Basis der jeweiligen Validierungspartition validiert. Schlussendlich werden die Validierungsergebnisse aggregiert (z. B. durch Berechnung des Mittelwerts der gewählten Metriken zur Validierung, wie bspw. Accuracy, ROC AUC o. ä.).

Dadurch, dass im Modellbildungsprozess, wie vorangegangen beschrieben, Optimierungen unter Einbezug der Validierungspartition stattfinden (Auswahl der Anzahl von Trainingsepochen und ggf. Hyperparameter Tuning), bedarf es für die Modellevaluation zusätzlich zur Validierungspartition einer Testpartition, welche im Training nicht berücksichtigt wurde, um das Modell auf „ungesehenen" Daten zu evaluieren.

Die Evaluationsergebnisse hängen davon ab, welche Datensätze (Beobachtungen) aus dem Gesamtdatensatz den einzelnen Partitionen (Training, Validierung, Test) zugeordnet werden und unterscheiden sich für verschiedene Aufteilungen des Gesamtdatensatzes (Train-Vali-Test Splits). Deshalb ist es sinnvoll mehrere Modelle mit unterschiedlichen Train-Vali-Test Splits zu trainieren und zu evaluieren, um Aufschluss über die Verteilung der Modellgenauigkeit (Streuung und Median) für verschiedene Splits zu geben (weitere Informationen siehe Abschnitt 5.2.5). Grundsätzlich soll genau dies durch die Anwendung der klassischen Kreuzvalidierung als Resampling Methode erreicht werden. Es liegen dabei jedoch folgende Problemstellungen vor, aufgrund derer eine Kreuzvalidierung im konkreten Anwendungsfall ungeeignet erscheint:

- Da bei der Kreuzvalidierung die Daten in disjunkte Teilmengen zerlegt werden (Zufallsziehung ohne zurücklegen) ergeben sich größerer Anzahl von Teilmengen („Folds“) Validierungspartitionen mit sehr geringen Mengen von Beobachtungen. Möchte man z. B. 30 Modelle auf 30 verschiedenen Validierungspartitionen evaluieren, würde der Anteil der Beobachtungen im Validierungsdatensatz 1/30, also ~3.33 Prozent betragen.
    - Beispiel: Bei einer 30-fold Kreuzvalidierung für das Merkmal NFPA.GF ergeben sich aufgrund der Menge von 3.583 verfügbarer Stoffe mit bekannter Zielvariable, 30 Validierungspartitionen mit nur ~119 Beobachtungen. Außerdem sind einige Ausprägungen des Merkmals stark unterrepräsentiert, was dazu führt, dass die Anzahl von Stoffen in der Validierungspartitionen mit diesen Ausprägungen nahe 0 betragen würde. Folglich wäre keine stichhaltige Validierung möglich.
- Hinzukommend wäre die nötige technische Umsetzung komplexer, um zu erreichen das für jede Validierungspartition eine vollständige Evaluation mit allen Auswertungen des HTML-Berichts durchgeführt wird.

Um diese Problemstellungen zu lösen, wurde der folgende Ansatz zur Modellvalidierung und Ergebnisanalyse gewählt:

Für die Erstellung eines Modells werden neben der Trainingspartition zwei weitere Partitionen (Validierungs- und Testpartition) verwendet, was auch als „Double-Holdout“ bezeichnet wird. Die Validierungspartition wird, wie zuvor bereits beschrieben, verwendet, um die optimale Anzahl von Trainingsepochen zu ermitteln und ggf. Hyperparameter Tuning durchzuführen. Die Testpartition wird verwendet, um das Modell auf ungenutzten Daten zu evaluieren.

Damit nicht nur anhand eines einzigen Modells bzw. einer einzigen Aufteilung der Daten in die verschiedenen Partitionen trainiert und evaluiert wird, wurde folgendermaßen vorgegangen:

Für ein Modell (ein Datensatz mit einer Zielvariable und einem Set von Steuerungsparametern), werden mehrere Läufe (Ausführungen des Chemprop.ipynb Notebook) durchgeführt. Dadurch werden ähnlich der Kreuzvalidierung mehrere Modelle auf dem gleichen Datensatz, aber mit unterschiedlichem Train-Vali-Test Split trainiert. Der entscheidende Vorteil besteht jedoch darin, dass bei der gewählten Vorgehensweise die Aufteilung der Daten in Partitionen so durchgeführt wird, das einzelne Beobachtungen auch in mehreren Modellierungsläufen in Validierungs- oder Testpartition vorkommen kann. Dies entspricht einer Zufallsziehung mit zurücklegen und ähnelt dem Prinzip des Bootstrapping. Dadurch wird erreicht, dass mit steigender Anzahl von Modellierungsläufen die Validierungs- und Testpartitionen nicht kleiner werden, wie es bei der Kreuzvalidierung der Fall ist. Außerdem werden alle Ergebnisdaten sowie die Evaluationsberichte für jeden der einzelnen Validierungsläufe generiert und gespeichert, sodass diese anschließend analysiert werden können. Die entsprechende Ergebnisauswertung für verschiedene Modelle und Steuerungsparameter folgt in Abschnitt 4.2.5.

4.2.4 Auswahl der modellierten ChemInfo Merkmale (Zielvariablen)

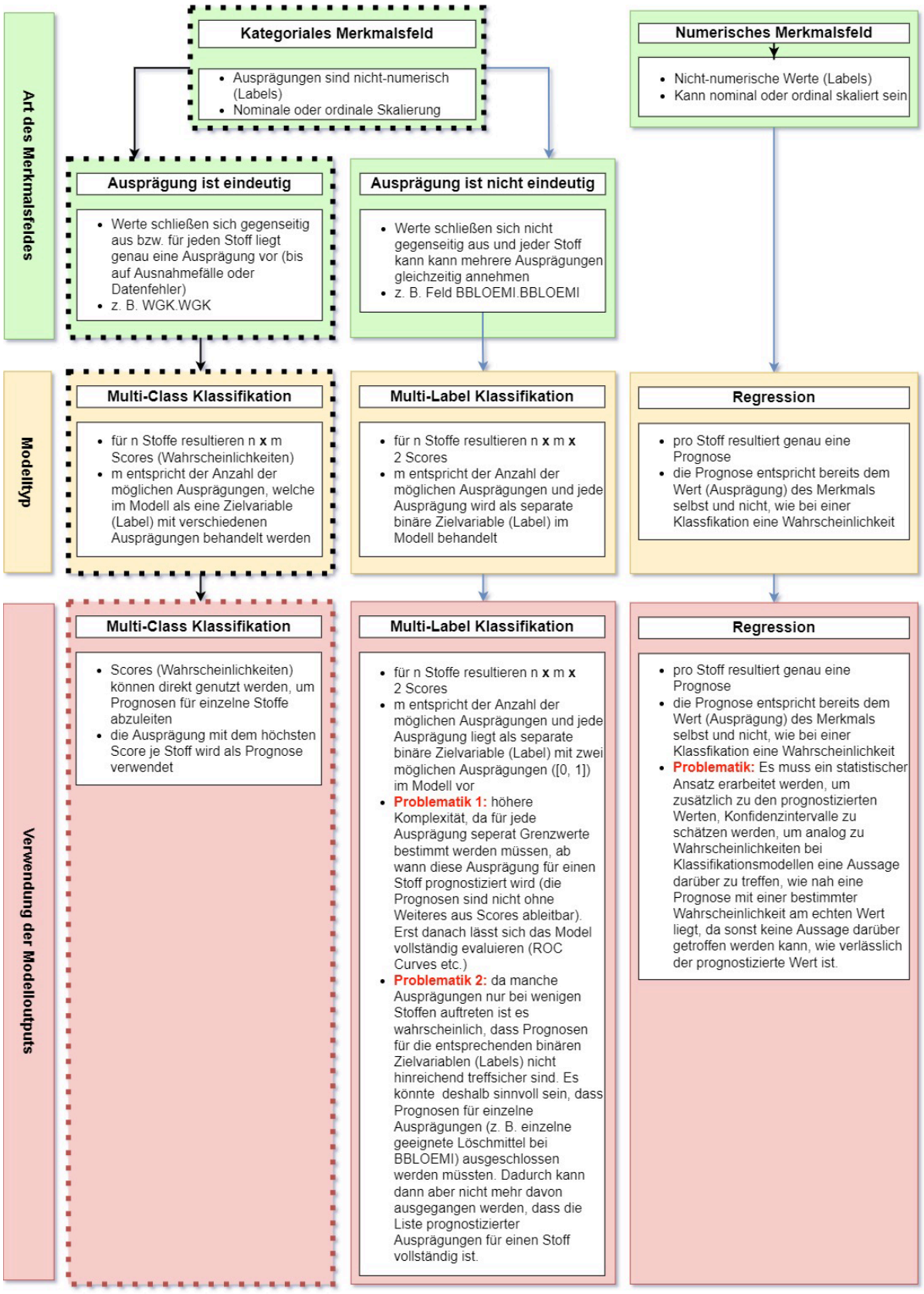
In der Analyse des Datenbestands (siehe Abschnitt 3.3) wurden bereits der Grad der Befüllung und das Wertespektrum der einzelnen ChemInfo Felder untersucht. Einige der grundlegenden Aussagen werden im Folgenden zur Verdeutlichung im Kontext Machine-Learning wiederholt.

Der Fokus, bei der Auswahl der Merkmalsfelder zur Befüllung mittels Chemprop lag auf den Feldern, welche auf den Datenblättern verwendet werden. Die Eignung dieser Felder zur Machine-Learning basierten Befüllung wurde nochmal genauer hinsichtlich folgender Aspekte untersucht:

- Hinreichende Befüllung:
  - Eine präzise Definition hinreichender Befüllung existiert nicht. Für den gewählten Transfer-Learning Ansatz werden zwar weniger Beobachtungen zum Re-Training eines Modells benötigt als beim initialen Training eines neuen Modells (kein Transfer-Learning), aber trotzdem sollte ein Mindestumfang des verwendeten Datensatzes vorliegen, um verwendbare Prognosen zu generieren. Ein Orientierungspunkt bei der Beurteilung, ob Felder hinreichend befüllt sind, konnte im Online Fachvortrag „Message-Passing Neutral Networks for Molecular Property Prediction using Chemprop“ (Minute 14:03) zur Anwendung des Chemprop-Modells gefunden werden. Auf Basis praktischer Erfahrungen wird erklärt, dass Trainingsdatensätze mit weniger als 1.000 Beobachtungen i. d. R. zu mangelhaften Ergebnissen führen. Außerdem wurde bei der Bestimmung der Anzahl von Stoffen mit Merkmalsdaten berücksichtigt, dass jeder Stoff nur einmal gezählt wird (z. B. durch Filterung auf Sachverhalte des GSA-Datenbestands) und ein SMILES Code vorliegt.
- Wertespektrum:
  - Die Anzahl vorkommender Ausprägungen sollte nicht zu groß sein. Je größer die Anzahl möglicher Werte, desto mehr Beobachtungen werden benötigt, sodass für jede mögliche Ausprägung genügend Beispiele vorliegen, welche in das Training des Modells einfließen.
  - Die Ausprägungen sollten möglichst nicht zu stark ungleich verteilt sein, da ansonsten wie bereits bei zu geringer Anzahl von Beobachtungen einige der Ausprägungen nicht oft genug auftreten. In der technischen Umsetzung wurde diese Problematik durch die optionale Gruppierung von Minoritätsklassen gelöst, jedoch wäre es günstiger, wenn dies nicht nötig ist. Denn auch wenn dadurch trotzdem ein Modell trainiert wird, können letztendlich für bestimmte Ausprägungen dadurch keine Prognosen abgegeben werden.
  - Das Wertespektrum sollte standardisiert sein. Eine mangelnde Standardisierung kann zum Beispiel dadurch verursacht sein, wenn es für semantisch identische Ausprägungen verschiedene Schreibweisen oder Formulierungen gibt. Das führt ebenfalls zu einer Anzahl verschiedener Ausprägungen, welche deutlich größer als nötig sind, was wiederum dazu führt, dass viele der Ausprägungen zu selten im Trainingsdatensatz vorkommen. Außerdem werden durch den Algorithmus im Modelltraining irrelevante (nicht kausale) Zusammenhänge verwendet, um inhaltlich gleiche Ausprägungen durch das Modell zu trennen.
- Kausalität:
  - Bei manchen Merkmalsfeldern stellt sich die Frage, ob die Hypothese aufgestellt werden kann, dass es eine Kausalität zwischen der Molekülstruktur und dem jeweiligen Merkmalsfeld gibt.
  - Zum Beispiel scheint es zweifelhaft, dass ein kausaler Zusammenhang zwischen Maßnahmen zur Brandbekämpfung (BBBRAND) oder z. B. dem zu verwendenden Messgerät bei Freisetzung (FREIMESS) unterstellt werden kann. In diesen Fällen bestände die Gefahr, dass durch ein Machine-Learning Modell Scheinzusammenhänge identifiziert werden.

- Art des Merkmalsfeldes und Modelltyp (s. Abbildung 11):
  - Jeder Modelltyp benötigt eine individuelle Umsetzung der Prozesse zur Erstellung des Modellierungsdatensatzes, Modelltraining, Modelevaluation und ggf. zugehörige Visualisierungen und Steuerungsparameter. Jeder Modelltyp erfordert also eine eigene Version des Jupyter Notebooks Chemprop.ipynb (siehe Abschnitt 4.2.2), als Kernstück des Modellierungsprozess.
  - Aufgrund der begrenzten Entwicklungszeit galt es abzuwägen, welche Gruppe von Merkmalsfeldern bzw. welcher Modelltyp in puncto Aufwand-Nutzenrelation priorisiert werden sollte.
  - Auf Basis der Betrachtung der Datenblatt relevanten Felder hinsichtlich der Kriterien: hinreichende Befüllung, Wertespektrum, Art des Merkmalsfeldes und dem zu verwendenden Modelltyp wurde entschieden welche Merkmalsfelder im Rahmen der bereits beschriebenen technischen Prozesse zur Befüllung (siehe Abschnitt 4.2.2) implementiert und evaluiert werden.

Abbildung 11: Art des Merkmalsfeldes und Modelltyp



Quelle: eigene Darstellung, SoftwareOne.

Im Folgenden werden zur Befüllung implementierte und nicht implementierte Felder aufgelistet und Begründungen zur Auswahl gegeben.

4.2.4.1 Kategoriale Merkmalsfelder (Multi-Class Klassifikation) – implementiert

Folgende Felder zur Anwendung des Modelltyps Multi-Class Klassifikation wurden implementiert:

Tabelle 5: Kategoriale Merkmalsfelder (Multi-Class Klassifikation) - implementiert

Merkmalsfeld Kurzbezeichnung	Merkmalsfeld Langbezeichnung	mitgeltendes Dokument
NFPA.GF	Gesundheitsgefahr Ziffer	NFPA.GF_model_evaluation.html
NFPA.RG	Reaktionsgefahr Ziffer	NFPA.RG_model_evaluation.html
WGK.WGK	Wassergefährdungsklasse	WGK.WGK_model_evaluation.html
AZ.AZ	Aggregatzustand	AZ.AZ_model_evaluation.html
NFPA.BA	Besondere Anweisungen	NFPA.BA_model_evaluation.html
HAZC.FZ	Feuerlöschmittel Ziffer	HAZC.FZ_model_evaluation.html
NFPA.BG	Brandgefahr Ziffer	NFPA.BG_model_evaluation.html
HAZC.KB	Körperschutz-Stoffbehandlung Buchstabe	HAZC.KB_model_evaluation.html

Quelle: eigene Darstellung, SoftwareOne.

Gründe diese Gruppe von Feldern zur Befüllung mittels Chemprop zu implementieren:

- Anzahl potenzieller Felder mit hinreichender Befüllung zum Modelltraining im Vergleich anderer Feldarten höchsten.
- Modelloutputs (Scores) können direkt zur Ableitung von Prognosen verwendet werden.
- Keine speziellere fachliche Problemstellung, wie bei Multi-Label- oder Regressionsmodellen. (siehe Abbildung 11)
- Aufwand / Nutzen Verhältnis am günstigsten.

4.2.4.2 Kategoriale Felder (Multi-Label) – nicht implementiert

Folgende Felder zur Anwendung des Modelltyps Multi-Label wurden geprüft, jedoch nicht zur Befüllung implementiert (siehe Tabelle 6).

Tabelle 6: Kategoriale Felder (Multi-Label) - nicht implementiert

Merkmalsfeld Kurzbezeichnung	Merkmalsfeld Langbezeichnung	Ausschlusskriterium
BBBRAND.BBBRAND	Einsatzhinweise bei Brand	Wertespektrum im aktuellen Zustand ungeeignet
BBLOEMA.BBLOEMA	Löschmaßnahmen	Wertespektrum im aktuellen Zustand ungeeignet
BBLOEMI.BBLOEMI	Löschmittel	Wertespektrum im aktuellen Zustand ungeeignet (Trennung in geeignete und ungeeignete Löschmittel nötig)
BBMESS.BBMESS	Messen||Nachweisen (Brand)	Wertespektrum im aktuellen Zustand ungeeignet / Kausalität fraglich
FREIMESS.FREIMESS	Messen||Nachweisen (Freisetzung)	Wertespektrum im aktuellen Zustand ungeeignet / Kausalität fraglich
FREISCHUTZ.FREISCHUTZ	Persönliche Schutzausrüstung	Wertespektrum im aktuellen Zustand ungeeignet / Kausalität fraglich
FREIEMP.FREIEMP	Freisetzung Empfehlung||Maßnahmen	Wertespektrum im aktuellen Zustand ungeeignet
FREIBIN.FREIBIN	Binde-u. Neutralisationsmittel	Geringe Befüllung / Wertespektrum im aktuellen Zustand ungeeignet
FREIEXP.FREIEXP	Explosionsschutz	Wertespektrum im aktuellen Zustand ungeeignet
GER.GER	Geruch	Wertespektrum im aktuellen Zustand ungeeignet
GER.BS	Geruchsbeschreibung	Geringe Befüllung / Wertespektrum im aktuellen Zustand ungeeignet
FREIWAS.FREIWAS	Verwendung von Wasser	Geringe Befüllung / Wertespektrum ungeeignet
FB.FB	Farbe	Wertespektrum im aktuellen Zustand ungeeignet
NMLUFT.NMLUFT	Prüfröhrchen	Geringe Befüllung / Wertespektrum im aktuellen Zustand ungeeignet / Kausalität fraglich

Merkmalsfeld Kurzbezeichnung	Merkmalsfeld Langbezeichnung	Ausschlusskriterium
VPMAT.VPMAT	Material	Geringer Befüllungsgrad / Wertespektrum im aktuellen Zustand ungeeignet (Trennung in geeignete und ungeeignete Materialien nötigt)
EG1272_08.KPIK & SEEG1272_08.KPIK	Kennzeichnung: Piktogramm	Bisher keine durch Merkmalsdaten bedingten Ausschlusskriterien festgestellt

Quelle: eigene Darstellung, SoftwareOne.

Gründe diese Gruppe von Feldern zur Befüllung mittels Chemprop nicht zu implementieren:

- Der Aufwand zur Implementierung von Multi-Label Modellen ist insgesamt höher als in den anderen beiden Fällen (Multi-Class und Regression), da jede Ausprägung als einzelne Zielvariable als Modelloutput vorliegt, für welche die jeweiligen Score-Grenzwerte individuell bestimmt werden müssen. Diese Vorgehensweise ist auch in der Abbildung zu den Arten von Merkmalsfeldern und Modelltypen (siehe Abbildung 11) beschrieben (siehe „Problematik 1“). Jede einzelne Ausprägung des Merkmalsfeldes, also Zielvariable des Multi-Label Modells, müsste anschließend evaluiert werden und die Ergebnisse der einzelnen Variablen sind dann zusammengefasst in Visualisierungen und Tabellen des HTML-Berichts darzustellen.
- Außer den Merkmalsfeldern der GHS-Symbole (EG1272_08.KPIK und SEEG1272_08.KPIK) gibt es keine für die Datenblätter relevanten Felder, welche in Ihrem gegenwärtigen Zustand die Auswahlkriterien, welche zu Beginn dieses Abschnitts beschrieben wurden und für eine Modellierung bzw. Machine-Learning basierte Befüllung essenziell sind. Insbesondere das Wertespektrum der jeweiligen Felder weist verschiedene Problematiken auf, welche vorangehend beschrieben wurden. Zusammenfassend kann man feststellen, dass ein oder mehrere, der folgenden Eigenschaften, bei den entsprechenden Feldern auftreten:
  - Zu viele verschiedene Ausprägungen,
  - Ausprägungen nicht standardisiert,
  - inhaltliche Überschneidungen,
  - Teils listenartige Inhalte (Beispiele: BBLOEMI, VPMAT),
  - zu geringe Befüllung und
  - mangelnde Kausalität (hypothetisch).

4.2.4.3 Numerische Felder (Regression) – nicht implementiert

Folgende Felder zur Anwendung des Modelltyps Regression wurden geprüft, jedoch nicht zur Befüllung implementiert (siehe Tabelle 7).

Tabelle 7: Numerische Felder (Regression) - nicht implementiert

Merkmalsfeld Kurzbezeichnung	Merkmalsfeld Langbezeichnung	Anzahl Stoffe (GSA / SMILES vorhanden) mit verfügbarem Wert
DI.DI_UWRT	Dichte	6378
FP.FP_UWRT	Flammpunkt	5703
SP.SP_UWRT	Siedetemperatur||Kondensationstemperatur	5621
SM.SM_UWRT	Schmelztemperatur||Gefriertemperatur	5503
WL.WL_UWRT	Wasserlöslichkeit || Sättigungskonzentration in Wasser (unterer Wert)	3651
DD.DD_UWRT	Dampfdruck	3403
EXUN.EXUN_UWRT	Untere Explosionsgrenze	1348
EXOB.EXOB_UWRT	Obere Explosionsgrenze	970
IONISE.IONISE_UWRT	Ionisierungspotential	963
ZUET.ZUET_UWRT	Zündtemperatur	785
GERS.GERS_UWRT	Geruchsschwelle	484
FWEINT.WRT_UWRT	Wert	42

Quelle: eigene Darstellung, SoftwareOne.

Gründe diese Gruppe von Feldern zur Befüllung mittels Chemprop nicht zu implementieren:

- Die Eignung der Modelloutputs bzw. Prognosen erscheint sowohl zur Verwendung im Rahmen der Datenblatterstellung als auch im ChemInfo System fragwürdig, da grundsätzlich keine Kenntnis darüber besteht, wie treffsicher die einzelnen Prognosen sind (siehe Abbildung 11). Dafür müsste erst eine statistische Methodik erarbeitet werden, wobei ex-ante nicht sicher ist zu welchem Grad dies gelingt bzw. anwendbar ist. Es schien daher sinnvoller die Umsetzung der Multi-Class Klassifikationsmodelle angesichts der begrenzt zur Verfügung stehenden Entwicklungszeit zu priorisieren.
- Aufwand / Nutzen Verhältnis erschien zu Beginn der Code Entwicklung etwas ungünstiger, als die Felder des Modelltyps Multi-Class Klassifikation zu implementieren, da weniger Felder mit hinreichender Befüllung vorlagen (siehe Abschnitt 4.2.3). Im weiteren Verlauf kamen dann weitere numerische Felder zur Verwendung auf den Datenblättern hinzu.

4.2.5 Analyse und Optimierung von Steuerungsparametern

4.2.5.1 Zielstellung

Nachdem die technische Umsetzung der Modellentwicklung für die in Abschnitt 5.2.4 beschriebenen ChemInfo Felder abgeschlossen war, wurde eine weiterführende Analyse durchgeführt, welche folgende Zielstellungen hat:

- Prüfung der Robustheit der Modellperformance gegenüber verschiedenen, zufälligen Train-Vali-Test Splits und Ermittlung der mittleren (erwartbaren) Modellgenauigkeit, in Form von Accuracy und Precision.
- Es soll untersucht werden, wie sich unterschiedliche Einstellungen von Steuerungsparametern (Parameter Sets) auf die Modellperformance auswirken, um zu bestimmen welche Einstellungen zum Training der finalen Modelle angewendet werden sollen.

4.2.5.2 Vorgehensweise

Zur Umsetzung der genannten Zielstellungen wurde die Methodik zur Modellvalidierung durchgeführt, welche bereits in Abschnitt 4.2.3 zur Umsetzung der Modellvalidierung beschrieben wurde. Konkret wurde wie folgt vorgegangen:

Es wurden mehrere Sätze von Steuerungsparametern (Parameter Sets) definiert, welche im nächsten Teilabschnitt näher erläutert werden. Ein Parameter Set kann als eine Methode zur Erstellung eines Modells verstanden werden. Für jedes der zur Modellierung implementierten ChemInfo Felder und jedes Parameter Set wurden 30 Modelle trainiert, evaluiert und die Evaluationsergebnisse gespeichert. Es wurden somit 1440 Modelle zur anschließenden Auswertung trainiert. (8 ChemInfo Felder * 6 Parameter Sets * 30 Modelle = 1440 Modelle). Damit pro Feld und Parameter Set je eine Stichprobe von 30 unterschiedlichen Beobachtungen vorliegt, wurden jeweils 30 Seed Values zufällig generiert. Seed Values sind die Startwerte von Zufallsgeneratoren, welche genutzt werden können, um Ergebnisse zu reproduzieren oder zu erzwingen, dass bestimmte Zufallsprozesse tatsächlich unterschiedlich verlaufen. Letzteres half im vorliegenden Anwendungsfall, um zu erreichen, dass für jeden der 30 Läufe die Daten zufällig in unterschiedliche Trainings-, Validierungs- und Testpartitionen aufgeteilt wurden.

Die Bestimmung des Stichprobenumfangs von 30 Beobachtungen folgte im Wesentlichen zwei Überlegungen. Zum einen sollten die Stichproben groß genug sein, dass die Anwendung statistischer Tests zur Erkennung signifikanter Unterschiede der mittleren Accuracy durchgeführt werden können und zum anderen galt es zu berücksichtigen, dass die Laufzeiten zum Training der Modelle in einem, aus wirtschaftlicher Sicht, realisierbaren Rahmen bleiben. Der gewählte Stichprobenumfang von 30 Beobachtungen (30 Modelle pro Zielvariable und Parameter Set) lässt hinreichend verlässliche Aussagen bei Anwendung nicht-parametrischer Testverfahren zur Identifikation von Unterschieden der mittleren Modellgenauigkeit zu. Außerdem ist der Rechenaufwand in puncto Laufzeit und Kosten für MS Azure Dienste mit dem Projektziel der Forschung nach Machine-Learning basierten Ansätzen zur Prognose von Stoffeigenschaften vereinbar. Der Rechenaufwand für einen einzelnen Trainingslauf hängt von diversen Faktoren ab, wie z. B. Zielvariable und dadurch bedingt Größe des Trainingsdatensatzes, Einzelmodell oder Ensemble, mit oder ohne Hyperparameteroptimierung, Hardwarekonfiguration der VM bzw. des lokalen Rechners, etc. Die Laufzeit zum Training der Modelle, unter Anwendung der final ausgewählten Steuerungsparameter, beträgt je nach Zielvariable ungefähr zwischen 20 und 50 Minuten (inkl. nötiger Datenladeprozesse und Transformationen).

Die Vorliegenden Evaluationsergebnisse für die jeweiligen Testpartitionen wurden anschließend mittels Python Code in einem separatem Jupyter Notebook analysiert und entsprechende Auswertungen werden im Folgenden in diesem Dokument erläutert.

4.2.5.3 Sätze von Steuerungsparametern (Parameter Sets)

Im Rahmen der technischen Implementierung wurden einige optionale Einstellungen bzw. Steuerungsparameter implementiert, um zu prüfen, inwieweit diese die Modellperformance beeinflussen. Da sich theoretisch eine sehr große Zahl möglicher Einstellungsvarianten ergibt und eine flächendeckende Optimierung hinsichtlich des Rechenaufwands im Verhältnis zur erwartbaren Modellverbesserung unverhältnismäßig wäre, wurde folgender Ansatz bei der Festlegung der zu untersuchenden Parameter Sets verfolgt. Es wurde ein Parameterset mit der Bezeichnung „baseline“ definiert, welches die Ausgangsbasis bzw. die Vergleichsgrundlage für weitere Parameter Sets bildet. Anschließend wurde schrittweise weitere Parameter Sets erstellt, in welchen zusätzliche Einstellungen aktiviert (z. B. „use_additional_features_fg“), deaktiviert (z. B. „use_scraped_smiles“) oder variiert (z. B. „val_dataset_size“) wurden. In Tabelle 6 sind alle getesteten Parameter Sets und deren Einstellungsvarianten dargestellt. Die Farben, der einzelnen Parameter Sets im Tabellenkopf werden zur Vergleichbarkeit ebenfalls in allen folgenden Abbildungen verwendet. Die hellgrün gefüllten Tabellenfelder markieren geänderte Einstellungen gegenüber dem Baseline Parameter Set.

Nachfolgend werden die individuellen Hintergrundinformationen zu den einzelnen Parametersets erörtert.

- Baseline:
  Vergleichsbasis mit Standardeinstellungen.

- fg50:
  Zusätzlich zum vorherigen Parameter Set werden die Top-50 meist vorkommenden funktionellen Gruppen als zusätzliche Inputvariablen (Features) verwendet. Die in den HTML-Berichten dargestellte Bezeichnung bzw. Nummerierung der funktionellen Gruppen (z. B. „X_FG.FG_214“) hat für die Ergebnisse von Modelltraining und Evaluation keine Relevanz, sondern stellt lediglich ein eindeutiges, zufällig vergebenes Synonym für die eigentliche Bezeichnung der funktionellen Gruppen dar, um die technische Verarbeitung und Darstellung der Daten zu vereinfachen. Die Zuordnung der Synonyme wird im Rahmen der Identifikation der funktionellen Gruppen in den SMILES Codes der Substanzen in den vorherigen ETL-Prozessen generiert und ist in einer Tabelle in der Azure SQL-Datenbank gespeichert. Die Daten der Zuordnungstabelle sowie alle anderen Inhalte des Datenbestands der Azure SQL-Datenbank werden als Teil der Projektergebnisse bereitgestellt.

- fg50val5:
  Zusätzlich zum vorherigen Parameter Set wird eine kleinere Validierungspartition (5 Prozent des Gesamtdatensatzes) verwendet, um dadurch einen größeren Trainingsdatensatz zu verwenden (statt 70 Prozent nun 80 Prozent des Gesamtdatensatzes).

- fg50val5ens5:
  Zusätzlich zum vorherigen Parameterset wird anstatt eines Einzelmodells, ein Modellensemble, bestehend aus fünf Einzelmodellen trainiert, welche sich auf Grund der unterschiedlichen Reihenfolge der Datensätze in der Trainingspartition unterscheiden.

- fg50val5ense5noscs:
  Zusätzlich zum vorherigen Parameterset werden in Modelltraining und -evaluation keine SMILES Codes verwendet, welche durch Webscraping ermittelt wurden, sondern lediglich

diejenigen, welche mittels RDKit aus den MOL-Dateien vom Umweltbundesamt abgeleitet wurden. Die Bitte um Prüfung inwiefern dies die Modelle beeinflusst, wurde vom Umweltbundesamt aufgeworfen. Eine verminderte Vorhersagequalität könnte ein Zeichen für die eingeschränkte Qualität der SMILES Codes der verwendeten externen Quelle sein.

- fg50val10ens5hpo3010ep25:
  Zusätzlich zum vorherigen Parameterset wurde die in Chemprop implementierte Hyperparameteroptimierung aktiviert, um Auswirkungen auf die Modellperformance zu prüfen. Weitere Informationen zum Hyperparameter Tuning sind in Abschnitt Verwendung zur Validierung bzw. Berechnung der Zielmetrik nach jeder Iteration im Hyperparameter Tuning beschrieben.

Tabelle 8: Überblick über untersuchte Sätze von Steuerungsparametern (Parameter Sets)

Steuerungsparameter	baseline	fg50	fg50val5	fg50val5ens5	fg50val5ens5 noscs	fg50val10ens 5hpo3010ep 25
use_scraped_smiles	1	1	1	1	0	0
use_rdkit_featurization	inaktiv	inaktiv	inaktiv	inaktiv	inaktiv	inaktiv
use_additional_features_fg	inaktiv	aktiv	aktiv	aktiv	aktiv	aktiv
fg_n_top_most_filled	0	50	50	50	50	50
fg_binarize	inaktiv	inaktiv	inaktiv	inaktiv	inaktiv	inaktiv
target_class_grouping	*individuell*	*individuell*	*individuell*	*individuell*	*individuell*	*individuell*
target_class_min_train_count	*individuell*	*individuell*	*individuell*	*individuell*	*individuell*	*individuell*
test_dataset_size	0,15	0,15	0,15	0,15	0,15	0,15
val_dataset_size	0,15	0,15	0,05	0,05	0,05	0,1
ensemble_size	1	1	1	5	5	5
run_hyperopt	inaktiv	inaktiv	inaktiv	inaktiv	inaktiv	aktiv
hyperopt_num_iters	0	0	0	0	0	30
hyperopt_startup_iters	0	0	0	0	0	10
batch_size	25	25	25	25	25	25
epochs	15	15	15	15	15	25
n_runs	30	30	30	30	30	30

Quelle: eigene Darstellung, SoftwareOne.

4.2.5.4 Auswertung der Ergebnisse

Nach der Durchführung der Trainingsläufe, wie in der Erläuterung der Vorgehensweise im Unterpunkt „Vorgehensweise“ zu Beginn dieses Abschnitts dargestellt, wurden verschiedene Auswertungen in Form von Abbildungen und Tabellen erstellt, welche im Folgenden detailliert beschrieben werden. Im Anschluss daran werden die Ergebnisse am Ende dieses Abschnitts zusammengefasst.

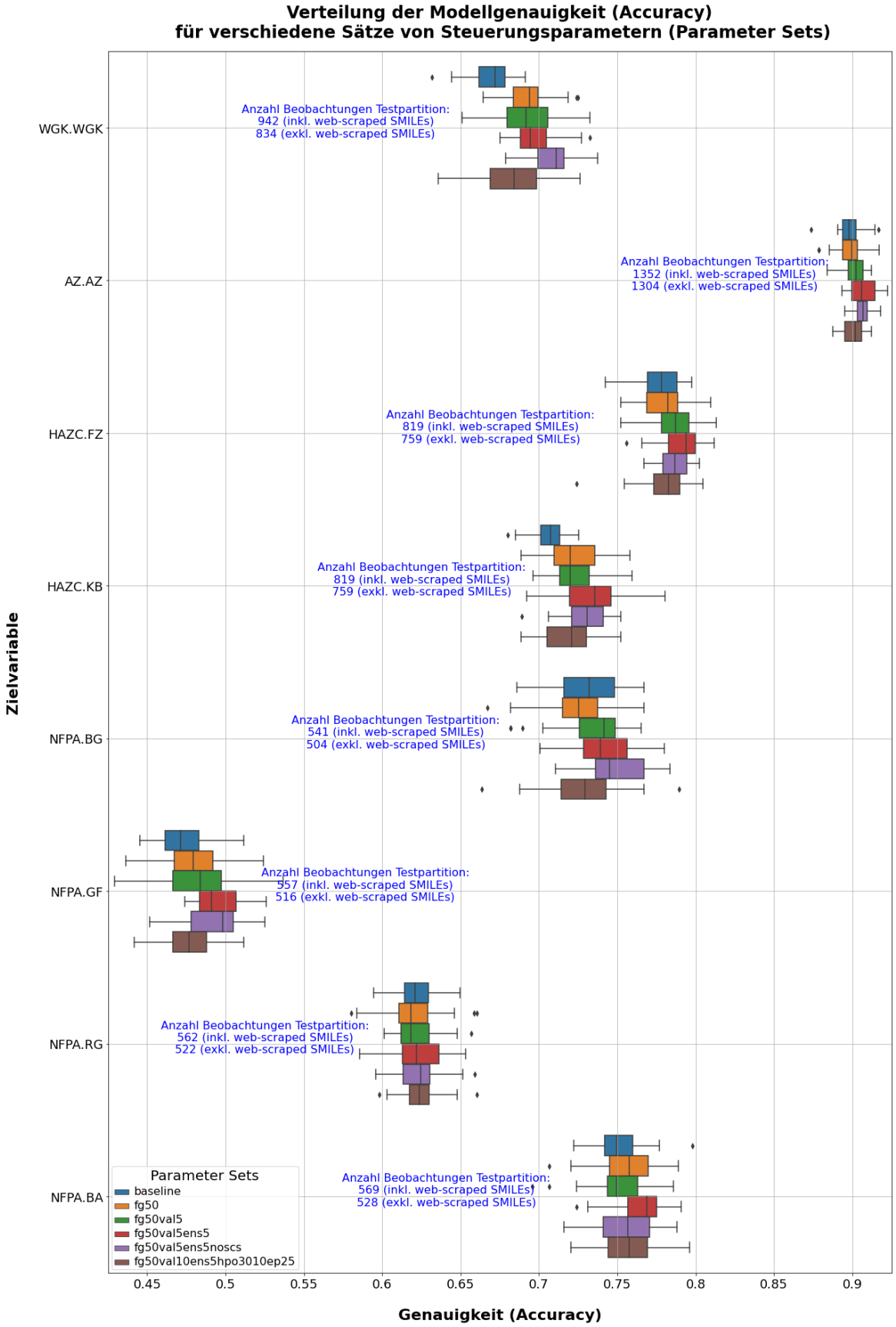
4.2.5.5 Beschreibung von Abbildung 12: Verteilung der Modellgenauigkeit für verschiedene Sätze von Steuerungsparametern

Als Gesamtüberblick über die Verteilung der Performance der Modelle pro Feld und Parameter Set dienen die Boxplots in Abbildung 12Abbildung 11. Jede Box repräsentiert 30 Trainingsläufe, also Beobachtungen der Accuracy der Modelle. Diese wurden jeweils auf Basis der Modellprognosen im Vergleich zu den tatsächlichen Werten der Zielvariable (jeweiliges Feld) für die Testpartition berechnet. Die Accuracy entspricht dem Anteil der korrekten Prognosen an allen Prognosen für die, in der Testpartition enthaltenen, Stoffe. Die blauen Annotationen zeigen informativ die Größe der Testpartition, für jede Zielvariable. Die Kurzbezeichnungen der Zielvariablen sind auf der Y-Achse angezeigt.

Die horizontale Ausdehnung der Box entspricht dem Interquartilsabstand bzw. dem Wertebereich der mittleren 50 Prozent der Beobachtungen. Die horizontale Ausdehnung der verschiedenen Whisker

(schwarze Linien, welche rechts und links an die Boxen angrenzen), erweitern den Wertebereich der Box in beide Richtungen bis maximal um den 1,5-fachen Interquartilsabstand. Beobachtungen, welche außerhalb des 1,5-fachen Interquartilsabstands liegen, werden als Ausreißer individuell als schwarze Symbole abgebildet. Zusammenfassend lässt sich sagen, dass zu bevorzugende Parameter Sets dadurch erkennbar sind, dass sie weiter rechts liegen und eine geringere horizontale Ausdehnung, sprich Streuung der Ergebnisse einzelner Trainingsläufe aufweisen.

Abbildung 12: Verteilung der Modellgenauigkeit für verschiedene Sätze von Steuerungsparametern



Quelle: eigene Darstellung, SoftwareOne.

4.2.5.6 Beschreibung von Tabelle 9: Ergebnisse Signifikanztest auf Unterschiede der Modellgenauigkeiten (Accuracy) für verschiedene Sätze von Steuerungsparametern (Parameter Sets)

In Tabelle 9 sind die Ergebnisse eines Signifikanztests dargestellt. Der Test wurde durchgeführt, um ergänzend zu den, in vorliegenden, visuellen Ergebnissen festzustellen, ob angenommen werden kann, dass sich für die einzelnen Zielvariablen die Median Accuracy der zusätzlichen Parameter Sets signifikant von der Median Accuracy des „baseline" Parameter Sets unterscheiden.

Das verwendete Testverfahren ist ein Mann-Witney-U-Test. Der Mann-Whitney-U-Test für unabhängige Stichproben testet, ob die zentralen Tendenzen zweier unabhängiger Stichproben verschieden sind. Der Mann-Whitney-U-Test wird verwendet, wenn die Voraussetzungen für einen t-Test für unabhängige Stichproben nicht erfüllt sind."[15] Weil die Stichproben, von 30 Beobachtungen eher klein sind und nicht bekannt ist, ob eine Normalverteilung der Accuracy Werte unterstellt werden kann, wurde ein nicht-parametrischer Test gegenüber einem parametrischen Test (z. B. t-Test für unabhängige Stichproben) bevorzugt. Die Boxplots deuten in den meisten Fällen bereits daraufhin, dass keine Symmetrische Verteilung vorliegt, was die Unterstellung einer Normalverteilung zusätzlich in Frage stellt. Für den Fall, dass in der Grundgesamtheit nun doch eine Normalverteilung zugrunde läge, wäre das jedoch auch kein fatales Problem, da nicht-parametrische Verfahren, insbesondere der gewählte Test kaum an Teststärke verlieren würde bzw. ähnlich oft zur richtigen Entscheidung führt wie das parametrische Pendant. Darüber sei bemerkt, dass ein Test für unabhängige Stichproben (auch ungepaart oder unverbunden) zu verwenden ist, da jedes der verschiedenen Modelle auf Basis anderer Trainings-, Validierungs- und Testpartitionen erstellt wurde bzw. die Accuracy Werte auf diesen basieren.

In Tabelle 9 sind für jedes implementierte ChemInfo Feld (Zielvariable) und Parameter Set (außer dem „baseline" Parameter Set) zeilenweise die Median Accuracy der jeweiligen Stichprobe (30 Modelle) und die Ergebnisse des Mann-Withney-U-Test (MWU-Test) dargestellt. Die Stichprobe für jedes Parameter Set, welche auf den Basiseinstellungen für das entsprechende Feld basiert, wird jeweils zum Vergleich verwendet. Die Differenzen der Median Accuracy Werte (zeilenweise) entsprechen somit den Abständen auf der X-Achse zwischen den Median Werten der Boxen aus Abbildung 12. Der Wert der Teststatistik des MWU-Tests ist lediglich informativ aufgeführt. Der p-Wert entspricht der Irrtumswahrscheinlichkeit bei Ablehnung Nullhypothese (H0), wobei diese besagt: Es besteht kein Unterschied zwischen den zentralen Tendenzen (Median) der beiden verglichenen Stichproben. Liegt also ein p-Wert < 0,05 vor, bedeutet dies, dass mit einer Wahrscheinlichkeit von 95 Prozent ein statistisch signifikanter Unterschied zwischen den Median Accuracy Werten der jeweiligen beiden Stichproben vorliegt. In diesem Fall ist die jeweilige Zelle in der Spalte „p-Wert" grün hervorgehoben und zusätzlich zur Verdeutlichung in der letzten Spalte (rechts) das Ergebnis der Prüfung, ob $p < 0.05$ ist, dargestellt. Je mehr grün markierte p-Werte für ein Parameter Set vorliegen, desto mehr Merkmale weisen eine signifikant höhere Median Accuracy im Vergleich zum jeweiligen Baseline Parameter Set auf.

[15] Universität Zürich (21.03.2022), Mann-Withney-U-Test, [online]
UZH - Methodenberatung - Mann-Whitney-U-Test, letzter Abruf am 07.12.2022

Tabelle 9: Ergebnisse Signifikanztest auf Unterschiede der Modellgenauigkeiten (Accuracy) für verschiedene Sätze von Steuerungsparametern (Parameter Sets)

Ziel-variable	Parameter Set	Median Accuracy Baseline Parameter Set	Median Accuracy Reference Parameter Set	Teststatistik (MWU-Test)	p-Wert	H0 ablehnen für p < 0,05
WGK.WGK	fg50	0,672	0,6937	95,5	0	Ja
AZ.AZ	fg50	0,8979	0,8994	407,5	0,2671	Nein
HAZC.FZ	fg50	0,7784	0,7821	405,5	0,2576	Nein
HAZC.KB	fg50	0,7076	0,7198	200,5	0,0001	Ja
NFPA.BG	fg50	0,732	0,7255	511,5	0,8205	Nein
NFPA.GF	fg50	0,4713	0,4794	339	0,051	Nein
NFPA.RG	fg50	0,621	0,6183	501,5	0,7794	Nein
NFPA.BA	fg50	0,7496	0,7575	369	0,1601	Nein
WGK.WGK	fg50val5	0,672	0,6916	134	0	Ja
AZ.AZ	fg50val5	0,8979	0,9024	312,5	0,0213	Ja
HAZC.FZ	fg50val5	0,7784	0,7869	291,5	0,0097	Ja
HAZC.KB	fg50val5	0,7076	0,7198	182	0	Ja
NFPA.BG	fg50val5	0,732	0,7412	391	0,1934	Nein
NFPA.GF	fg50val5	0,4713	0,4838	332	0,041	Ja
NFPA.RG	fg50val5	0,621	0,6183	476	0,6527	Nein
NFPA.BA	fg50val5	0,7496	0,7496	441,5	0,4529	Nein
WGK.WGK	fg50val5ens5	0,672	0,6943	33	0	Ja
AZ.AZ	fg50val5ens5	0,8979	0,9057	207	0,0002	Ja
HAZC.FZ	fg50val5ens5	0,7784	0,7937	222	0,0004	Ja
HAZC.KB	fg50val5ens5	0,7076	0,7357	123	0	Ja
NFPA.BG	fg50val5ens5	0,732	0,7394	348,5	0,0676	Nein
NFPA.GF	fg50val5ens5	0,4713	0,491	142,5	0	Ja
NFPA.RG	fg50val5ens5	0,621	0,6219	418,5	0,323	Nein
NFPA.BA	fg50val5ens5	0,7496	0,7689	238,5	0,0009	Ja
WGK.WGK	fg50val5ens5noscs	0,672	0,711	16	0	Ja
AZ.AZ	fg50val5ens5noscs	0,8979	0,9068	199	0,0001	Ja

Ziel-variable	Parameter Set	Median Accuracy Baseline Parameter Set	Median Accuracy Reference Parameter Set	Teststatistik (MWU-Test)	p-Wert	H0 ablehnen für p < 0,05
HAZC.FZ	fg50val5ens5noscs	0,7784	0,7866	276	0,0052	Ja
HAZC.KB	fg50val5ens5noscs	0,7076	0,7306	94	0	Ja
NFPA.BG	fg50val5ens5noscs	0,732	0,745	261	0,0027	Ja
NFPA.GF	fg50val5ens5noscs	0,4713	0,4981	199	0,0001	Ja
NFPA.RG	fg50val5ens5noscs	0,621	0,6245	414	0,2997	Nein
NFPA.BA	fg50val5ens5noscs	0,7496	0,7566	393	0,2017	Nein
WGK.WGK	fg50val10ens5hpo3010ep25	0,672	0,6842	250,5	0,0016	Ja
AZ.AZ	fg50val10ens5hpo3010ep25	0,8979	0,9016	365	0,1056	Nein
HAZC.FZ	fg50val10ens5hpo3010ep25	0,7784	0,7827	389,5	0,1874	Nein
HAZC.KB	fg50val10ens5hpo3010ep25	0,7076	0,721	227,5	0,0005	Ja
NFPA.BG	fg50val10ens5hpo3010ep25	0,732	0,7292	495,5	0,7519	Nein
NFPA.GF	fg50val10ens5hpo3010ep25	0,4713	0,4767	376,5	0,1399	Nein
NFPA.RG	fg50val10ens5hpo3010ep25	0,621	0,6237	397	0,2183	Nein
NFPA.BA	fg50val10ens5hpo3010ep25	0,7496	0,7575	362	0,0977	Nein

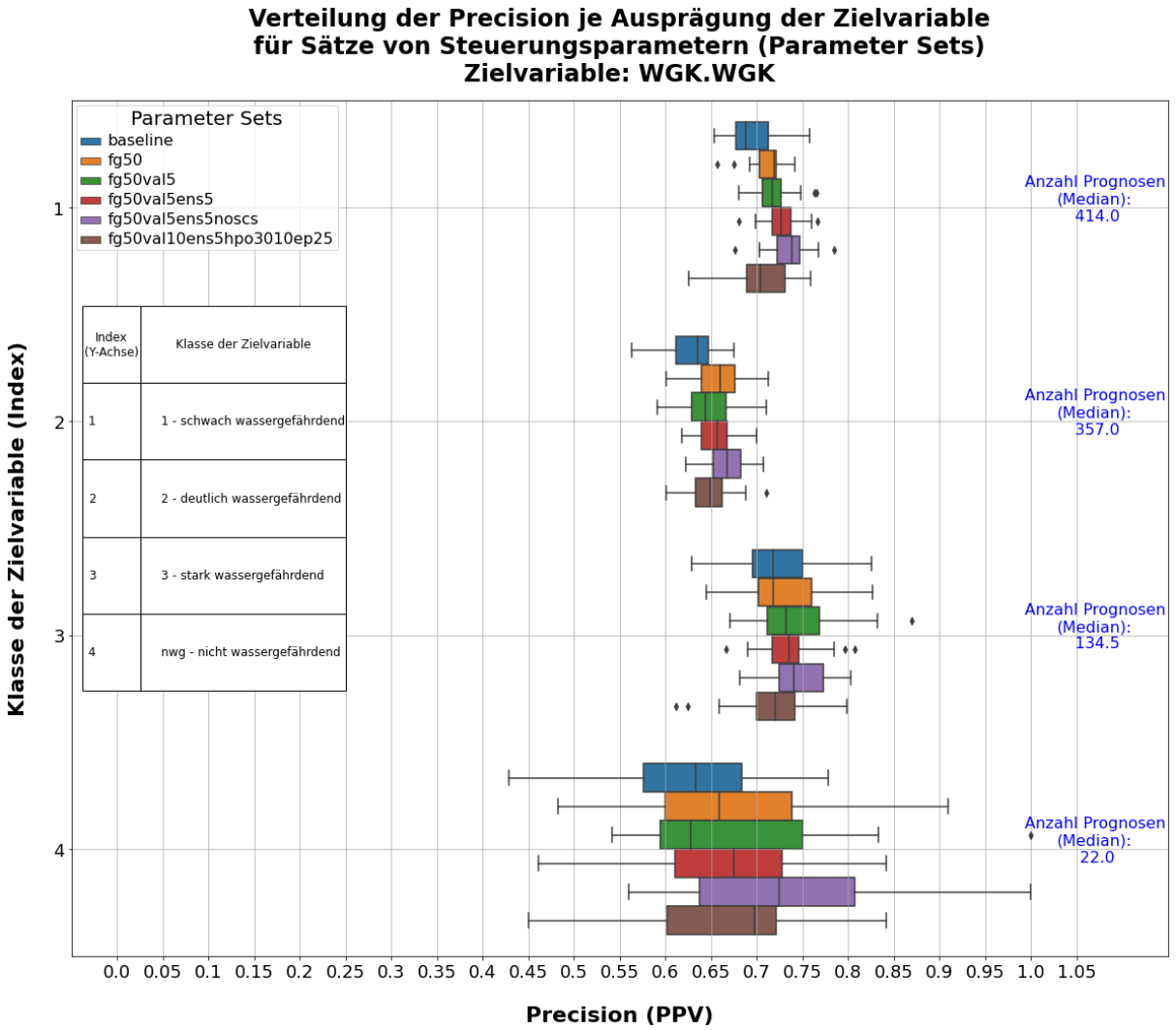
Quelle: eigene Darstellung, SoftwareOne.

4.2.5.7 Beschreibung der Abbildungen zur Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Abbildung 13 bis Abbildung 20)

Ergänzend zu Abbildung 12 brechen die folgenden Abbildungen (Abbildung 13 bis Abbildung 20) die Verteilung der Modellgenauigkeit auf Ausprägungsebene herunter. Dafür gibt es pro ChemInfo Feld (Zielvariable) eine Abbildung in welcher für jede mögliche Ausprägung (Y-Achse) die Verteilung der Precision für die verschiedenen Parameter Sets als Box dargestellt ist. Die einzelnen Ausprägungen wurden aus Platzgründen auf der Y-Achse als Indexwerte (nummeriert) angegeben. Welche Ausprägung die einzelnen Indexwert repräsentieren, ist in Form einer Tabelle mittleren linken Bereich der Plots dargestellt. Außerdem sind am rechten Rand auf Höhe der jeweiligen Ausprägung die Median Anzahl der abgegebenen Prognosen vermerkt. Hier wurde der Median verwendet, weil sich die Anzahl der Prognosen für die jeweilige Ausprägung innerhalb der 30 Modelle moderat unterscheidet. Die Reihenfolge der einzelnen Gruppen von Boxen (Ausprägungen) ergibt durch absteigende Sortierung nach dem der Anzahl von Prognosen (Median über einzelne Parameter Sets). Die Boxen werden von oben

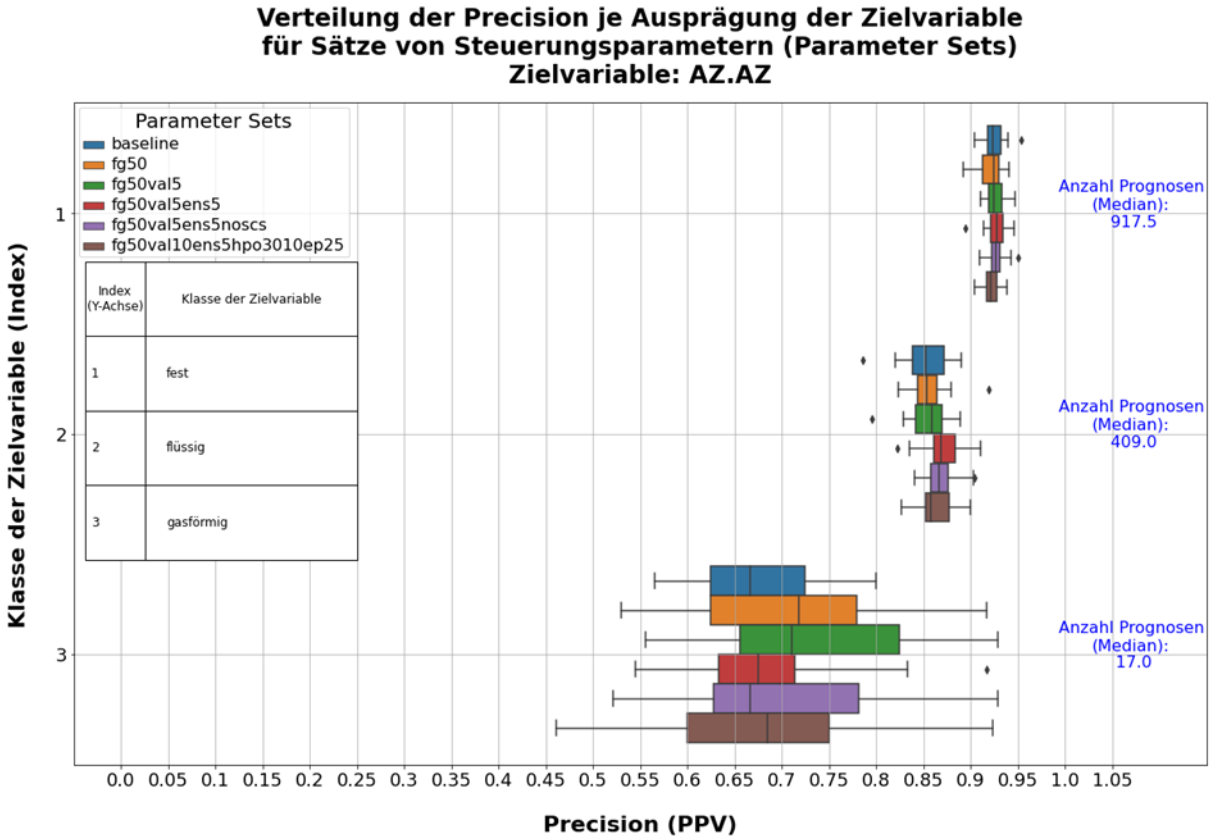
nach unten breiter, da die Precision bei Ausprägungen mit wenig Prognosen innerhalb der jeweils 30 Modelle stärker schwankt bzw. die Varianz höher ist.

Abbildung 13: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: WGK.WGK



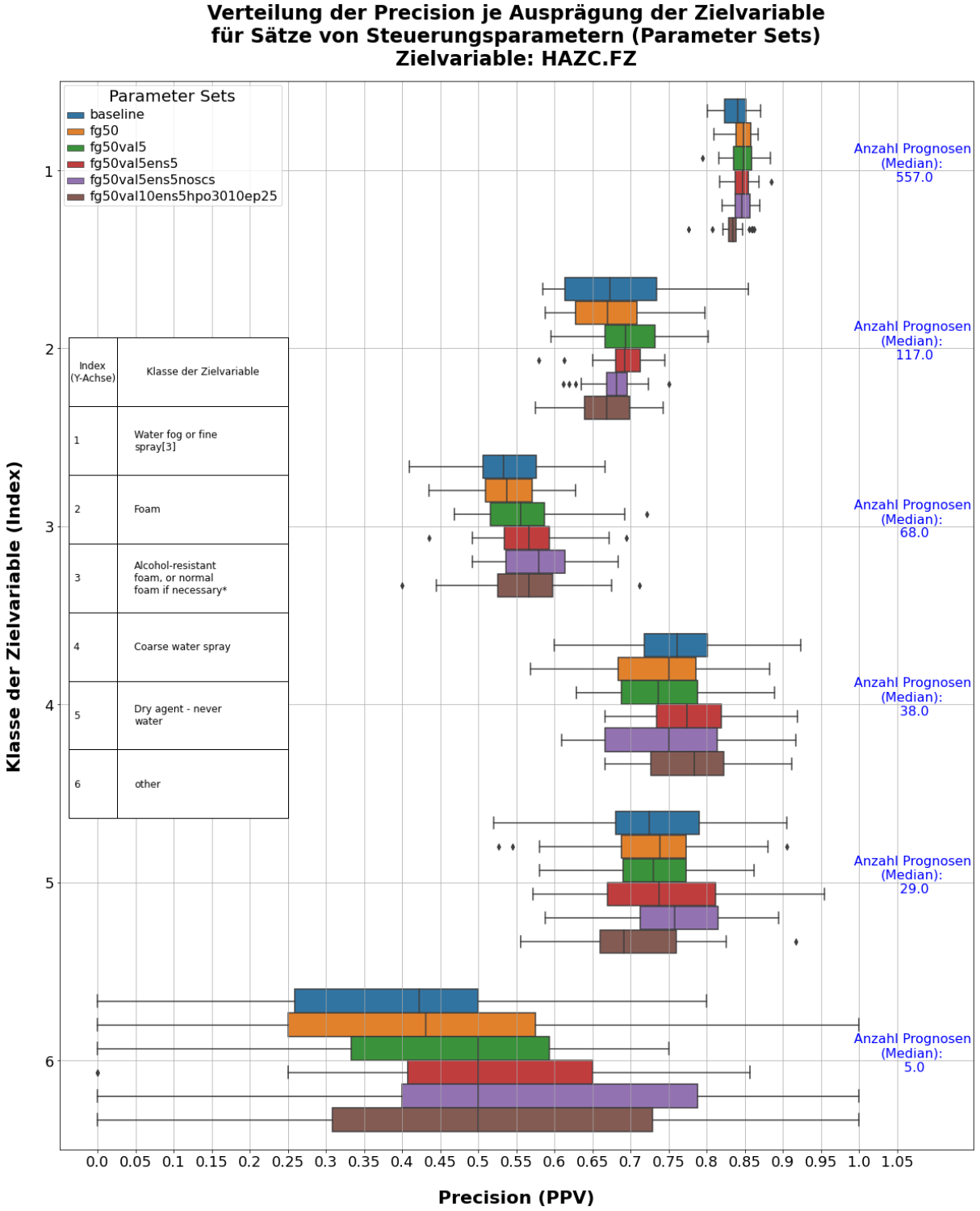
Quelle: eigene Darstellung, SoftwareOne.

Abbildung 14: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable AZ.AZ



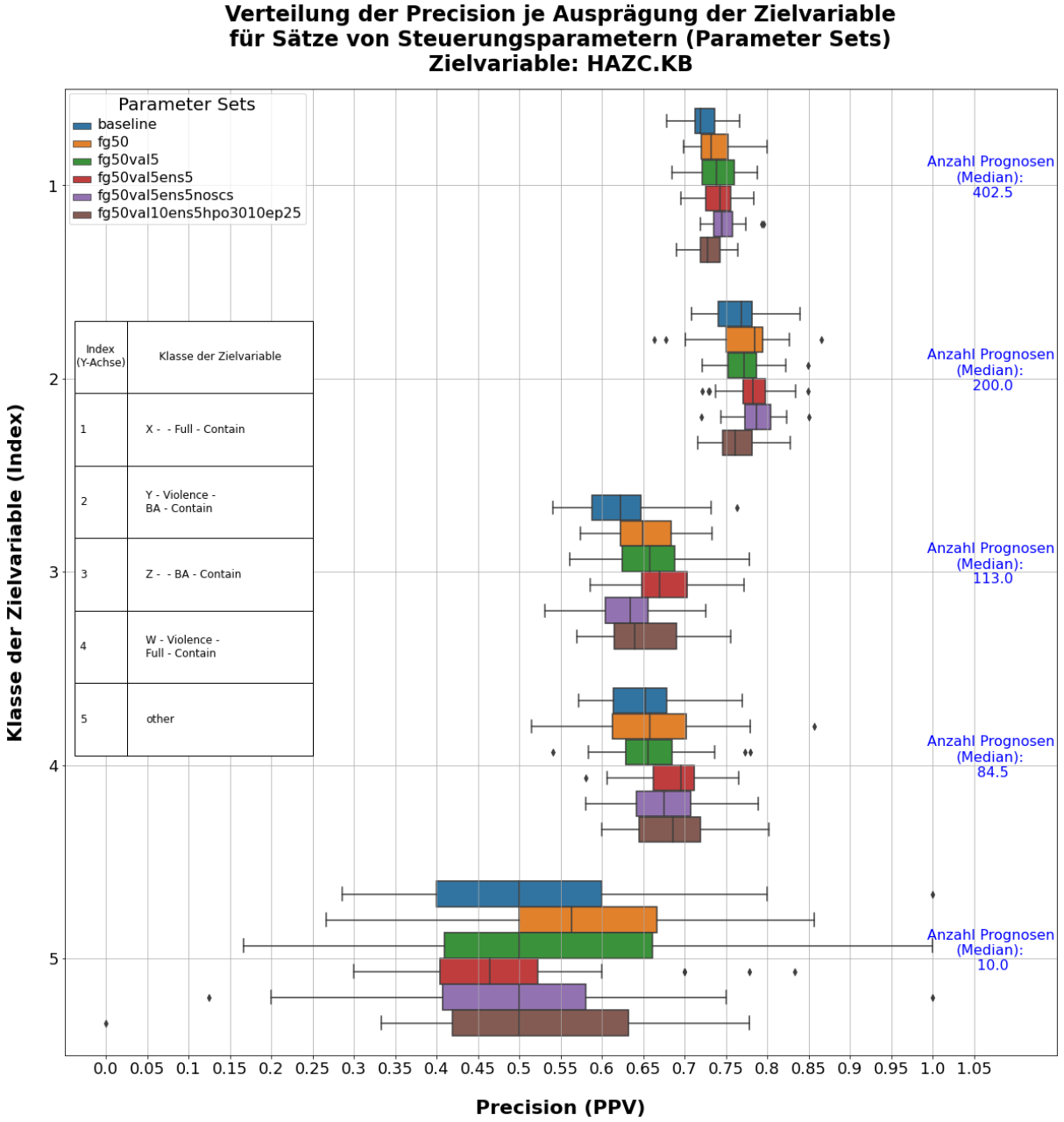
Quelle: eigene Darstellung, SoftwareOne.

Abbildung 15: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: HAZC.FZ



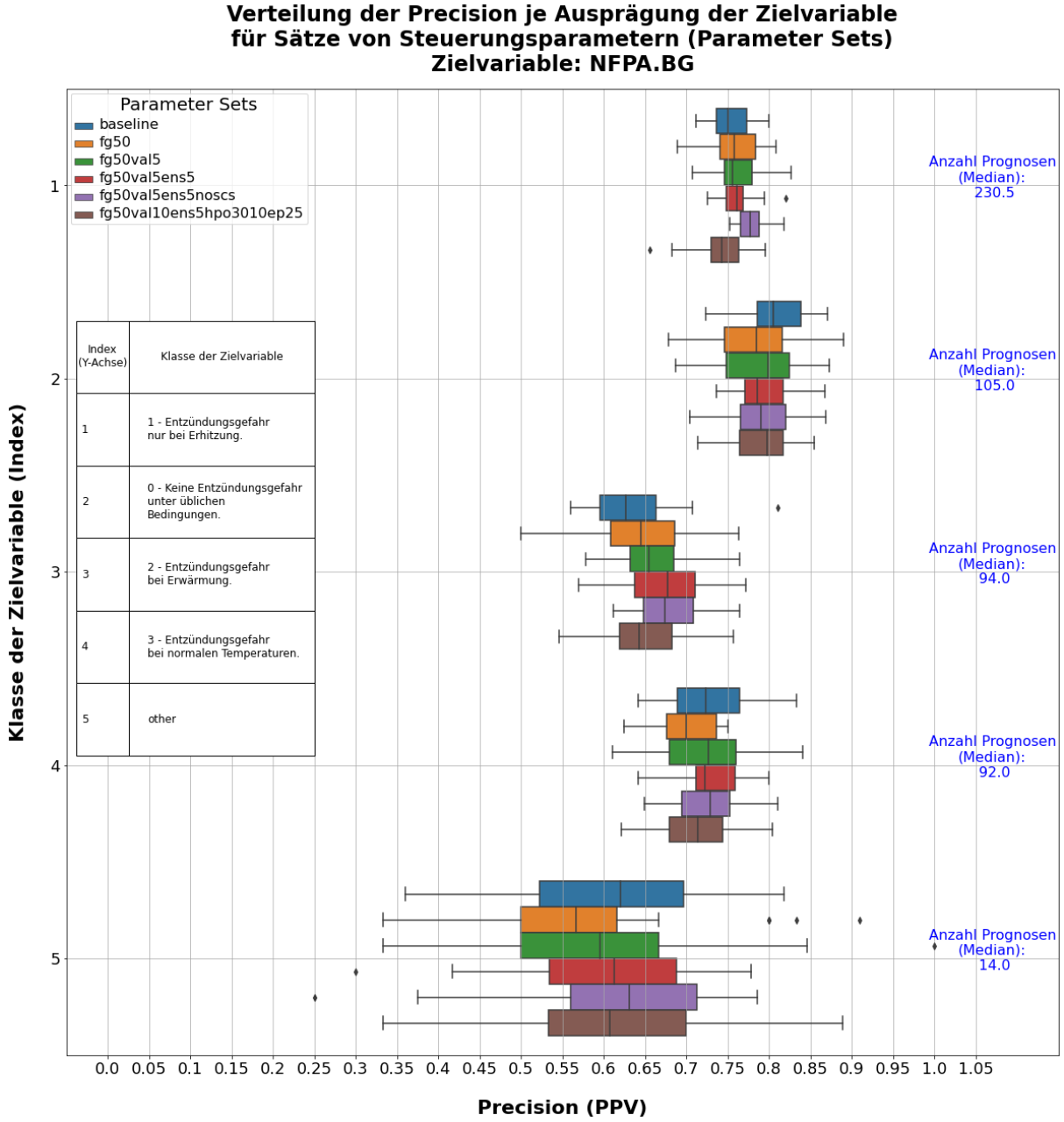
Quelle: eigene Darstellung, SoftwareOne.

Abbildung 16: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: HAZC.KB



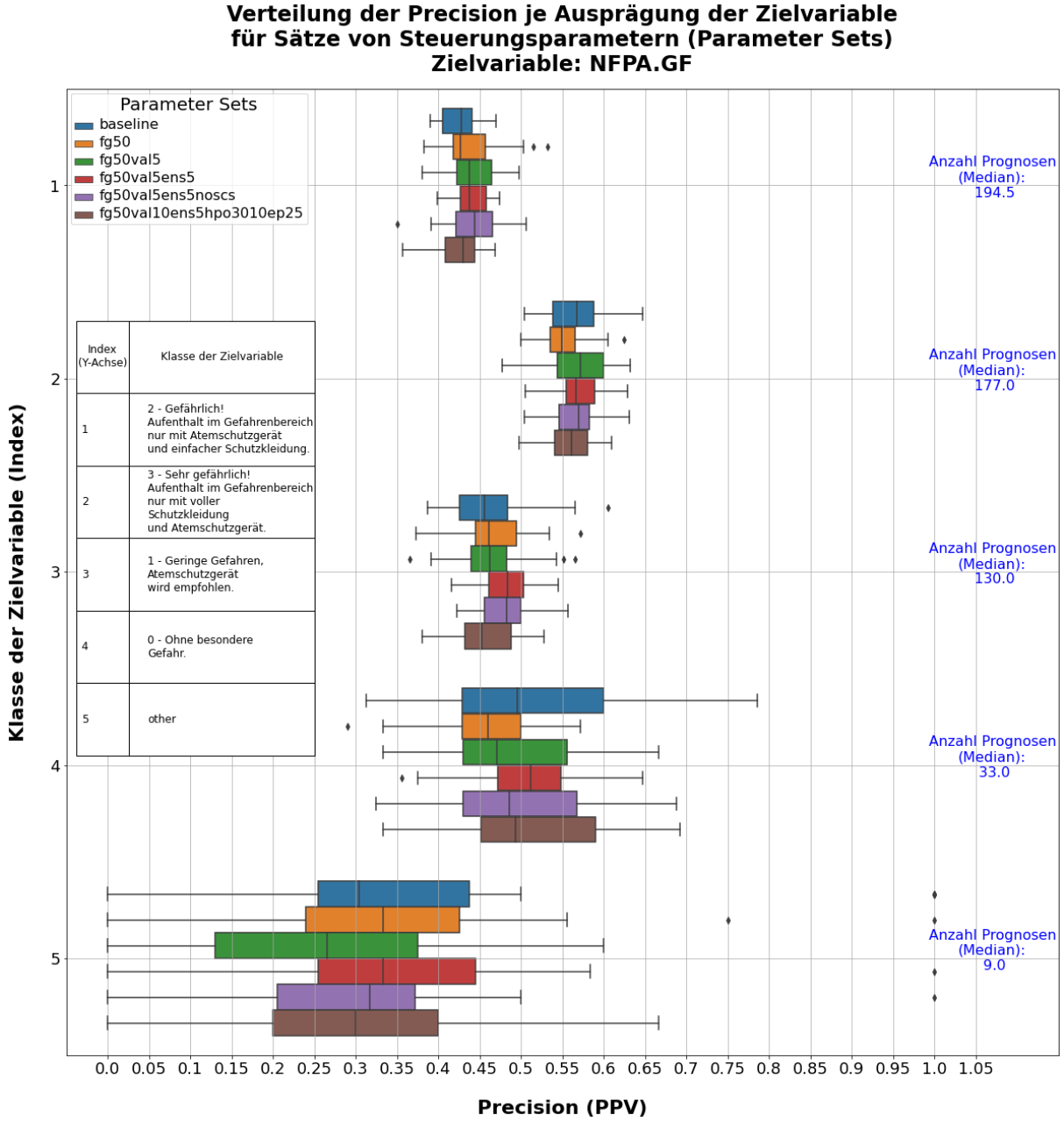
Quelle: eigene Darstellung, SoftwareOne.

Abbildung 17: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: NFPA.BG



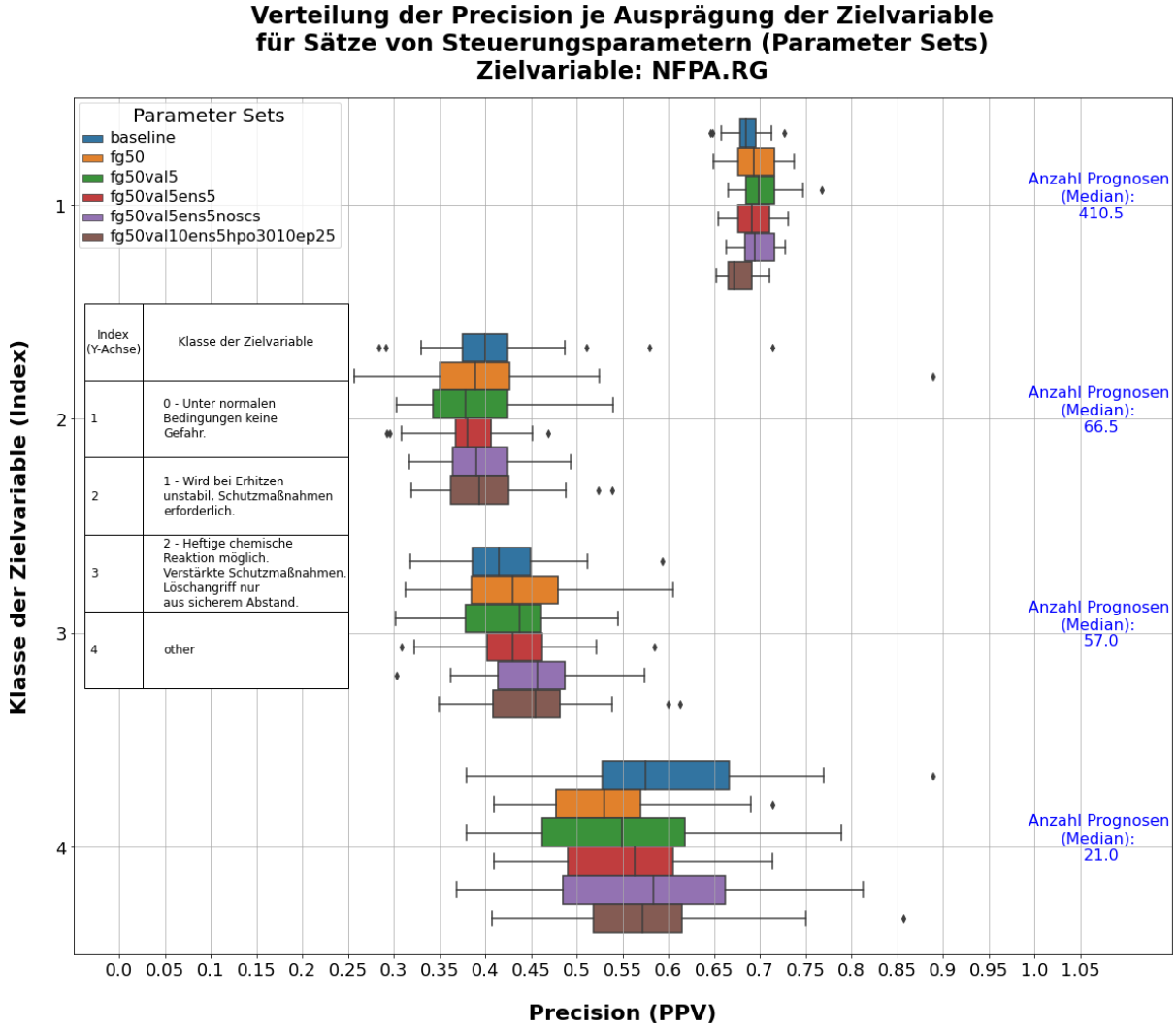
Quelle: eigene Darstellung, SoftwareOne.

Abbildung 18: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: NFPA.GF



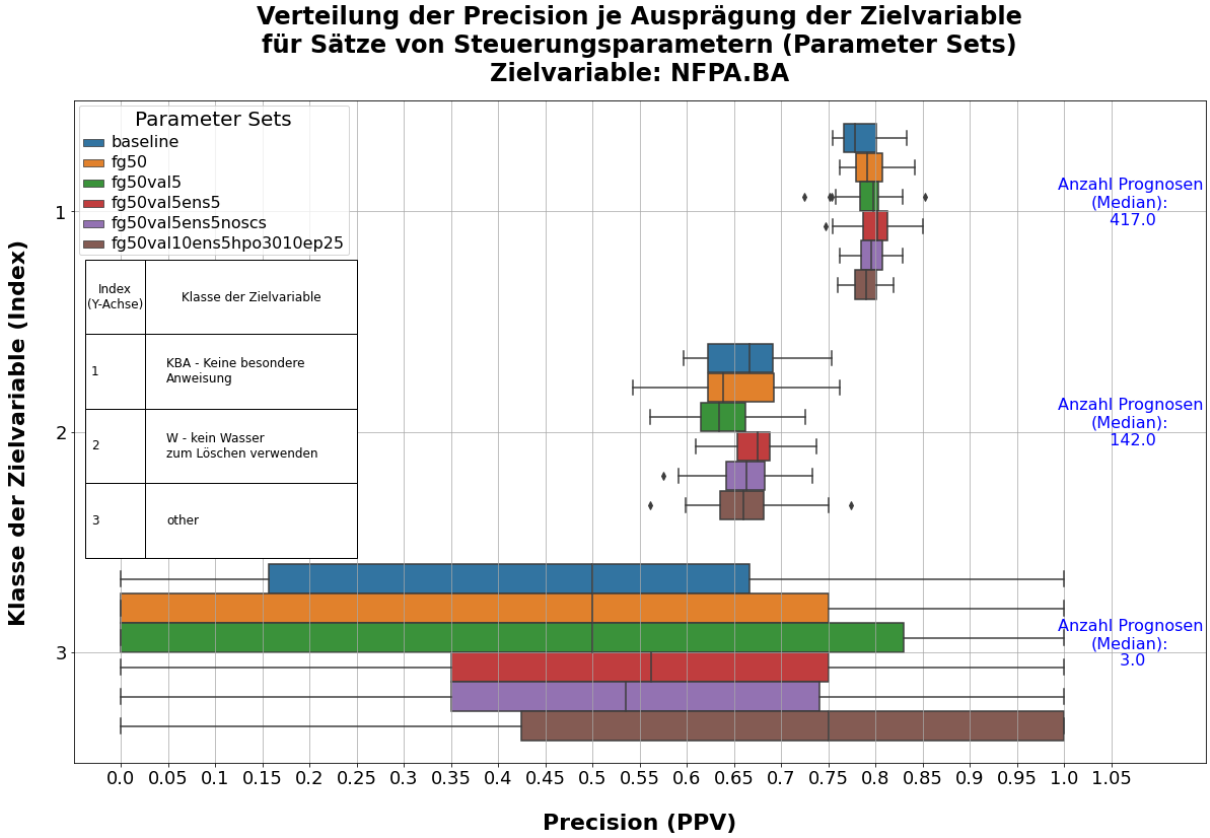
Quelle: eigene Darstellung, SoftwareOne.

Abbildung 19: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: NFPA.RG



Quelle: eigene Darstellung, SoftwareOne.

Abbildung 20: Verteilung der Precision je Ausprägung der Zielvariable für Sätze von Steuerungsparametern (Parameter Sets) - Zielvariable: NFPA.BA



Quelle: eigene Darstellung, SoftwareOne.

4.2.5.8 Beschreibung der Abbildungen zur Verteilung der optimalen Anzahl von Trainingsepochen pro Zielvariable und von Steuerungsparametern (Abbildung 21 und Abbildung 22)

Um die Laufzeit für das Training eines Modells auf die benötigte Anzahl von Epochen zu begrenzen, um das bestmögliche Ergebnis auf der Validierungspartition zu erzielen, sprich um nicht länger zu trainieren als nötig, wurde die Anzahl von Trainingsepochen mittels Steuerungsparameter „epochs“ auf 15 gesetzt. Weitere Informationen zur Bedeutung des Parameters befinden sich in den HTML-Berichten der finalen Modelle (mitgeltendes Dokument Chemprop Modelle). Würde die Anzahl der Epochen nicht explizit vorgeben, liegt der Standardwert, welcher in Chemprop implementiert ist, bei 30. Wenn also auch mit 15 Trainingsepochen, das gleiche Ergebnis erzielt werden kann, wie mit 30 Epochen, kann dadurch die Laufzeit halbiert werden.

Um zu überprüfen, ob 15 Trainingsepochen ausreichend sind, wurden in den folgenden beiden Abbildungen (Abbildung 21 und Abbildung 22) Histogramme erstellt. Die Histogramme zeigen die Verteilung der Anzahl von Trainingsepochen, welche nötig waren, um das geringstmöglichen Validierungs-Loss (Cross-Entropy-Loss) zu erzielen, sprich der Anzahl von Epochen, auf der das fertig trainierte Modell basiert. Diese Epochenanzahl wird in den Abbildungen als „Best Epoch“ bezeichnet und entspricht den Werten der X-Achse in den Histogrammen. Die Histogramme sind zeilenweise nach Parameter Set und spaltenweise nach ChemInfo Feld (Zielvariable) angeordnet. Die Darstellungen wurden aus Platzgründen auf zwei Abbildungen aufgeteilt.

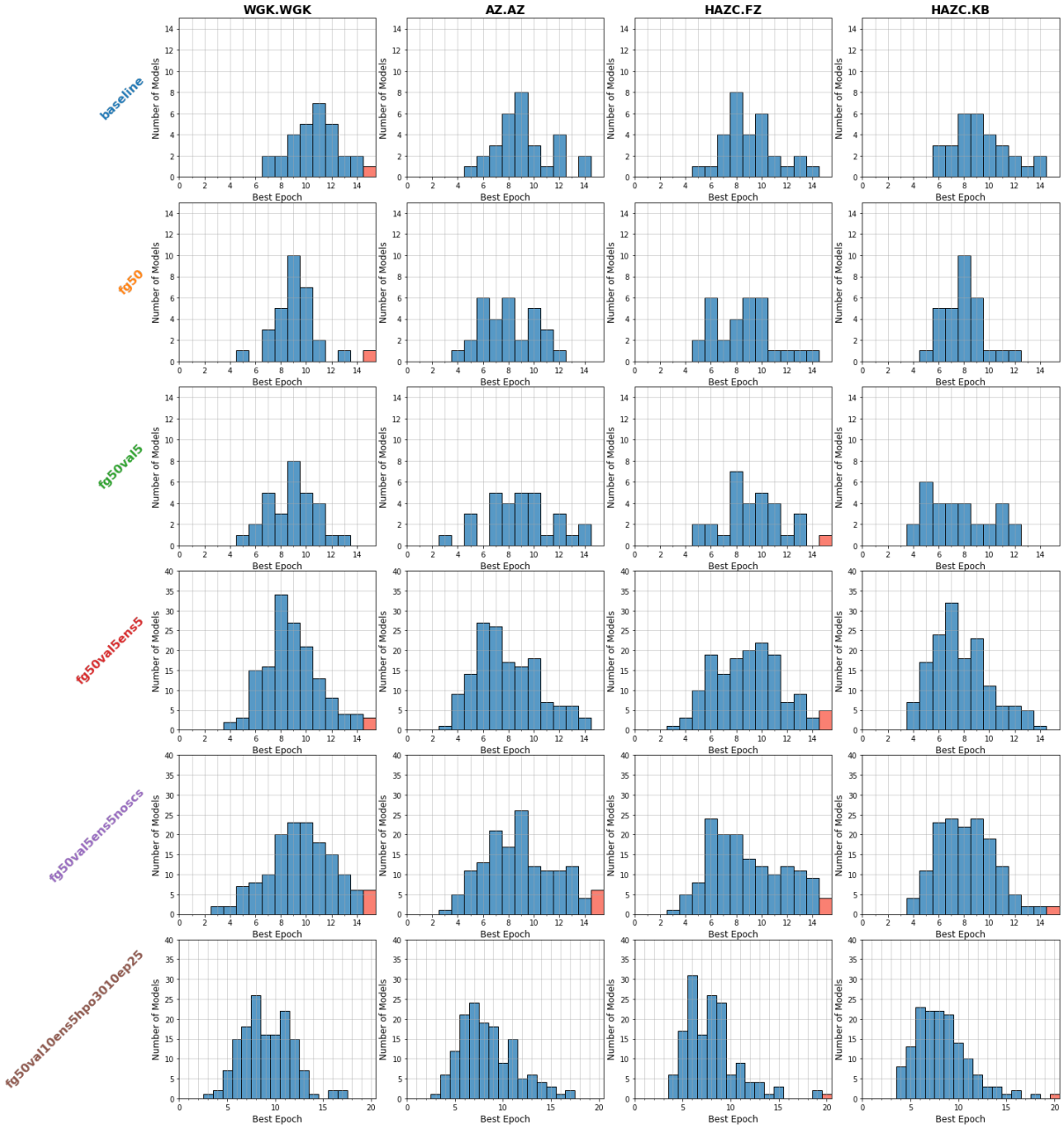
Für das Parameter Set in der untersten Zeile wurde die Anzahl der Epochen auf 25 gesetzt, um eine Gegenprobe zu den sonst durchgeführten 15 Trainingsepochen. Deshalb ist der Wertebereich auf der X–Achse in der letzten Zeile der Histogramme auf 25 Epochen erweitert gegenüber den anderen Zeilen (Parameter Sets). Die beiden folgenden Aspekte führten zu dieser Entscheidung:

- Bei Anwendung der Optimierung der Hyperparameter ist es möglich, dass sich, im Vergleich zur Anwendung der Standardwerte dieser Parameter, die Komplexität der zugrundeliegenden Modellarchitektur verändert. (z. B. zusätzliche Schichten oder Knoten im hinteren Teil des Netzwerks). Um auszuschließen, dass zur Erreichung einer vergleichbaren oder besseren Prognosequalität zusätzliche Trainingsepochen benötigt werden, wurde das Limit der Epochenanzahl erhöht.
- Zur Stützung der bis dato vorliegenden Erkenntnisse, welche sich aus den Ergebnissen der zuvor angewendeten Parameter Sets ergaben, erschien es sinnvoll in diesem letzten Parameter Set den Spielraum hinsichtlich der maximalen Anzahl möglicher Trainingsepochen zu erhöhen. Dadurch soll verifiziert werden, dass im weiteren Verlauf des Modelltrainings für mehr als 15 Trainingsepoche keine Steigerung der Prognosequalität auf der jeweils verwendeten Testpartition erzielt werden kann.

Für die Parametersets in den untersten drei Zeilen ist der Wertebereich der Y – Achse erweitert, da bei diesen Parameter Sets Ensemble Modelle, also 30 Modelle bestehend aus jeweils fünf Einzelmodellen, erstellt wurden und somit die Histogramme statt insgesamt 30 Werten für die beste Epoche, 150 Werte beinhalten.

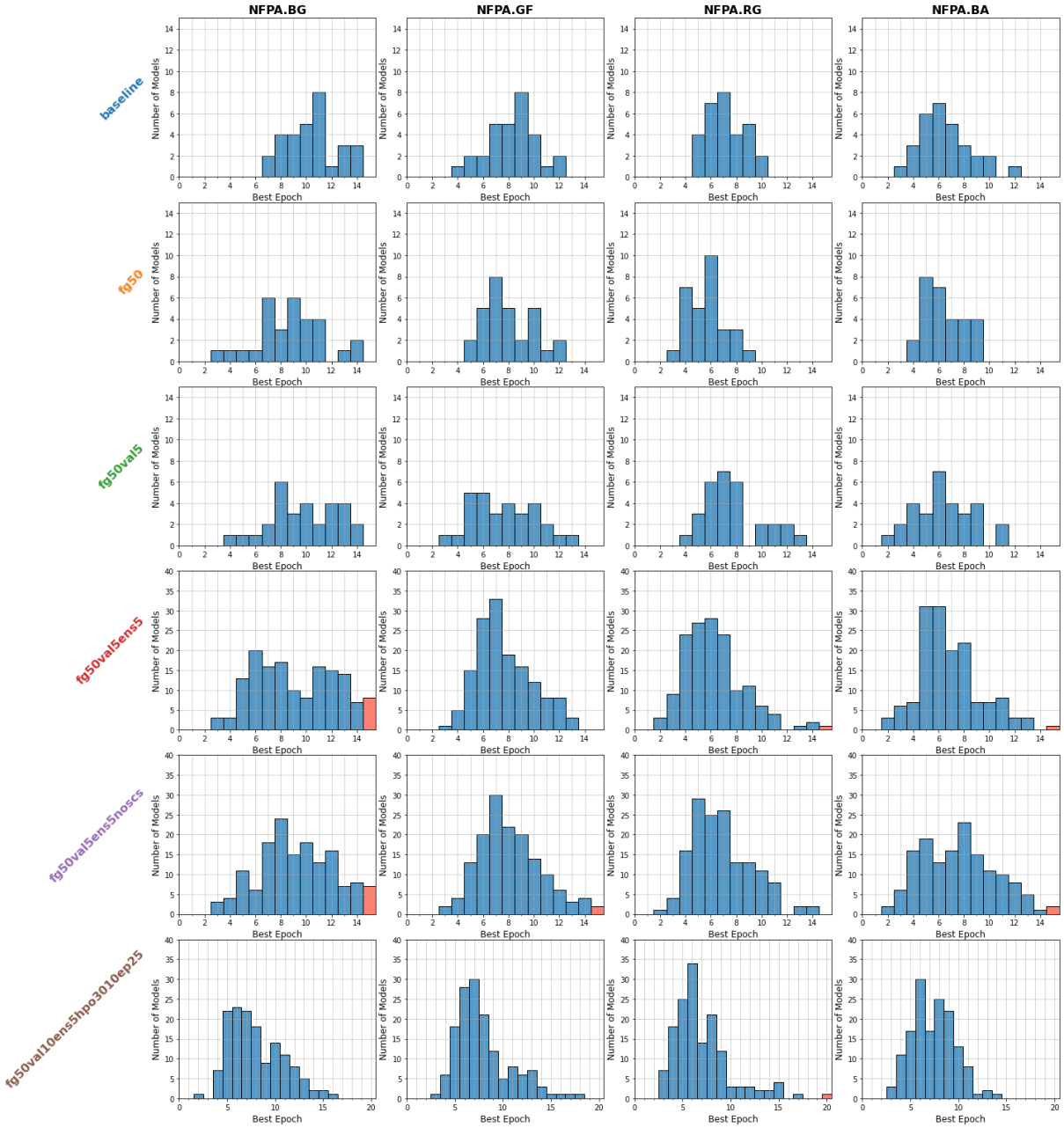
Die Fälle, wo die beste Trainingsepoche („Best Epoch“) der vorgegebenen Gesamtanzahl von Trainingspochen (Steuerungsparameter „epochs“) entspricht, sind farblich hervorgehoben. In diesen Fällen kann es entweder sein, dass tatsächlich die beste Trainingsepoche die letzte Trainingsepoche ist oder, dass es zusätzliche Trainingsepochen gebraucht hätte, um die beste Epoche zu erreichen.

Abbildung 21: Verteilung der optimalen Anzahl von Trainingsepochen pro Zielvariable und Satz von Steuerungsparametern (Parameter Set) - Teil 1



Quelle: eigene Darstellung, SoftwareOne.

Abbildung 22: Verteilung der optimalen Anzahl von Trainingsepochen pro Zielvariable und Satz von Steuerungsparametern (Parameter Set) - Teil 2



Quelle: eigene Darstellung, SoftwareOne.

4.2.5.9 Zusammenfassung der Ergebnisse und Schlussfolgerungen

Anhand der beschriebenen Auswertungen in Tabelle 9 sowie Abbildung 12 bis Abbildung 22 sollen im Folgenden Erkenntnisse zusammengefast werden und das für die finalen Modelle zu verwendende Parameter-Set bestimmt werden.

► Auswertung der Modellgenauigkeiten für die verschiedenen Felder und Parameter Sets

- Im absoluten Vergleich der Modelle der einzelnen ChemInfo Felder (Zielvariablen) zeigen sich deutliche Unterschiede. Dies kann diverse Gründe haben, wie beispielsweise der Umfang der Datenpartitionen, die Verteilung der Ausprägungen, Stärke des Zusammenhangs zwischen SMILES Codes (Strukturdaten) und Zielvariable, etc. Die

einzelnen Ursachen sind häufig in der Praxis nicht eindeutig identifizierbar oder quantifizierbar. Grundsätzlich kann allgemein gesagt werden, dass durch das Modell kein Erklärungsgehalt der Inputvariablen (SMILES Codes) gegenüber der Zielvariable identifiziert werden konnte. Für detaillierte Informationen zu den Modellen der einzelnen Felder sollten die HTML-Berichte (mitgeltendes Dokument Chemprop Modelle) herangezogen werden. Zusätzlich ist zu beachten, dass die Modellgenauigkeit (Accuracy) nicht die finale Genauigkeit im Produktiveinsatz darstellt, da diese anschließend noch mit der vereinbarten Methodik zur Anwendung von Precision Grenzen gesteuert bzw. erhöht werden kann.

- Der primäre Gegenstand dieser Analyse ist der relative Vergleich der Modellgenauigkeit für die verschiedenen Parameter Sets eines Feldes:
  - Außer bei Feld WGK.WGK scheint Parameter Set „fg50“ zur Verwendung funktioneller Gruppen als zusätzliche Features (Inputvariablen) keine signifikante Verbesserung der Accuracy zu bewirken.
  - Die zusätzliche Verringerung der Validierungspartition auf 5% der Gesamtdaten zugunsten des Umfangs der Trainingspartition (Parameter Set fg50val5) hat eine signifikante Verbesserung der Accuracy bei fünf von acht Feldern bewirkt.
  - Die Erzeugung von Modellensembles, bestehend auf fünf Einzelmodellen (Parameter Set „fg50val5ens5“) führt in allen Fällen, gemessen an den p-Werten im Vergleich zum vorherigen Parameter Set, zu einer leichten Verbesserung. Die Varianz der Accuracy hat sich jedoch entgegen der Hypothese nicht sichtbar verringert (Abbildung 12).
  - In Parameter Set „fg50val5ens5noscs“ wird auf die Verwendung der, zusätzlich durch Webscraping ermittelten, SMILES Codes verzichtet, um zu prüfen welchen Einfluss dies auf die Modellperformance hat. Diese Fragestellung wurde ursprünglich durch das Umweltbundesamt aufgeworfen. Tatsächlich zeigt sich, dass die zusätzlichen SMILES Codes keinen systematisch positiven Effekt auf die Resultate haben. Es ist eher so, dass ohne diese Daten häufig die Varianz (Streuung) der Accuracy verringert wird und teilweise die mittlere Accuracy steigt. Folgende Ursachen hierfür sind denkbar:
    - Es könnte sein, dass sich die zusätzlichen Stoffe in bestimmten Charakteristika systematisch von den anderen Stoffen der Modelldatensätze unterscheiden.
    - Die Qualität der, aus externen Quellen, ermittelten SMILES Codes ist mangelhaft.
    - Im ChemInfo Datenbestand fehlen vor allem bei Stoffen Strukturinformationen, bei denen eine eineindeutige Zuordnung einer Molekülstruktur schwer oder nicht möglich ist. Darum könnte es sein, dass die Strukturen aus anderen Datenbanken fehlerbehafteter sind.
  - Im letzten Parameter Set „fg50val10ens5hpo3010ep25“ wurde zusätzlich eine Optimierung der Hyperparameter durchgeführt. Dies führt bei sechs Zielvariablen zu einer sichtlichen Verschlechterung der Ergebnisse und bei zwei Modellen zu vergleichbaren Ergebnissen wie beim vorherigen Parameter Set. Dies könnte zum einen durch eine zu geringe Anzahl von Optimierungsiterationen (30) bedingt sein, wobei sich bereits bei dieser Einstellung die Trainingslaufzeit um Faktor 30 gegenüber dem Training ohne Optimierung erhöht und eine weitere signifikante

Erhöhung in puncto Rechenzeit unpraktikabel ist. Zum anderen besteht die Vermutung, dass die Optimierung der Hyperparameter entweder zu komplexeren Modellen und damit Overfitting oder zu stark vereinfachten Modellen und dadurch Underfitting im Vergleich zu den Standardeinstellungen der Hyper-Parameter führen könnte.

- Ergänzend zur Auswertung der Modellgenauigkeit wurde in Abbildung 13 bis Abbildung 20 die Precision für jede Ausprägung bzw. Klasse der Zielvariable in Form von Boxplots visualisiert. Die Darstellungen wurden zur Erhöhung der Transparenz und Informationstiefe in den Bericht aufgenommen, jedoch die Ergebnisse nicht im Detail beschrieben, da diese aus den Grafiken hervorgehen.

- Parameter Set „fg50val5ens5noscs“ wird zum Training der finalen Modelle verwendet. Die Begründung der Auswahl besteht darin, dass mit diesen sowie den Einstellungen für Parameter Set „fg50val5ens5“ die besten Ergebnisse hinsichtlich der Accuracy und Robustheit (Streuung der Accuracy) hervorgehend aus Abbildung 12 und Tabelle 9 vorliegen. Außerdem wird präferiert nicht auf die Verwendung zusätzlicher SMILES Codes aus externen Quellen angewiesen zu sein, welche per Webscraping ermittelt werden müssen, um lizenzrechtliche Fragestellungen und operative Risiken zu vermeiden. Dadurch könnte dieser Teil auch in der ETL-Pipeline ausgespart werden, um die Gesamtlaufzeit zu reduzieren.

► Verteilung der jeweils besten Trainingsepochen

- Bis auf wenige Ausnahmen sind 15 Trainingsepochen ausreichend, gemessen am Anteil der farblich hervorgehobenen Säulen in den Histogrammen (Abbildung 21 undAbbildung 22). Nach Training der finalen Modelle wurde geprüft, ob die benötigte Epochenanzahl nicht durch die Begrenzung der Trainingsepochen limitiert wurde. Eine dadurch bedingte Einschränkung der Modellperformance kann also ausgeschlossen werden.

► Zusammenfassung:

- Die Modellgenauigkeit ist differenziert zu betrachten, kann aber durch die Anwendung von Steuerungsparametern sowie anschließend der Beschränkung der Precision mittels Mindest-Scores verbessert werden.

- Der Einsatz zusätzlicher SMILES Codes, welche mittels Webscraping angereichert wurden, lohnt sich nicht, da die zusätzliche Abhängigkeit von externen Daten nicht durch systematische Modellverbesserungen kompensiert wird.

- Die Optimierung von Hyperparametern verursacht tendenziell eine Verringerung der Accuracy.

- Das Parameter Set „fg50val5ens5noscs“ wird zum Training der finalen Modelle verwendet.

- Die spezifizierte Anzahl von 15 Trainingsepochen ist ausreichend.

4.2.6 Erstellung finaler Modelle und Verwendung zur Lückenbefüllung

Zur Befüllung der implementierten ChemInfo Felder wurden final Modelle trainiert. Dabei wurden die Steuerungsparameter (fg50val5ens5noscs) angewendet, welche auf Basis der Analyseergebnisse in Abschnitt 4.2.5 ausgewählt wurden.

Die finalen Modelle wurde nicht aus den Stichproben zur Analyse ausgewählt (30 Modelle pro Parameter Set und Zielvariable), da vermieden werden sollte, dass eine willkürliche Auswahl anhand der Ergebnisse auf der Testpartition getroffen wird. Dies würde zu Overfitting bzw. einer Überanpassung der produktiv einzusetzenden Modelle an die Testpartition führen. Es kann nicht geschlussfolgert werden, dass das beste Modell gemessen an den Ergebnissen auf den Testdaten auch die höchste Genauigkeit auf den Daten bzw. Stoffen zur Befüllung aufweist. Die Stichproben dienten lediglich dem relativen Vergleich verschiedener Parameter Sets untereinander und nicht etwa dem Vergleich verschiedener Seed Values bzw. Trainings-, Validierungs-, Test-Datensplits.

Deshalb wurden die Modelle unter Verwendung zufälliger Seed Values trainiert. Anschließend wurde lediglich überprüft, ob die spezifizierte Anzahl von 15 Trainingsepochen ausreichend war bzw. dass die beste Epoche kleiner als 15 war, um zu gewährleisten, dass keine mangelnde Modellanpassung aufgrund zu weniger Trainingsepochen vorliegt.

Die HTML-Berichte zur Evaluation der finalen Modelle wurden dem Zwischenbericht als mitgeltendes Dokument beigefügt. Die Dateinamen der Dokumente sind in Tabelle 5 dargestellt.

4.2.7 Offene Schritte zur Verwendung der Chemprop Vorhersagen in der Datenblatterstellung

Fehlende Stoffinformationen in ChemInfo sollen durch Chemprop Prognosen ergänzt werden, um angezeigte Informationen auf den Datenblättern zu vervollständigen. Es wurde deshalb ein Prozess erarbeitet, um diese zusätzlichen Daten in die Ausführung der Datenblattregeln (s. Regelwerk zur Verbalisierung der Faktendaten) und die Datenblatterstellung zu integrieren. Die technische Implementierung wurde im Rahmen des Projekts nicht durchgeführt. Im Folgenden erläutern wir unseren Vorschlag zur Implementierung der Chemprop Vorhersagen in der Datenblatterstellung.

- Vorabfilterung der Chemprop Prognosen.
  - Im Jupyter Notebook zum Modelltraining (s. 5.2.2) muss ein Filter implementiert werden, sodass alle Prognosen mit einer Precision von weniger als 0.5 verworfen werden. Die Precision gibt an, wie hoch der Anteil der korrekten Prognosen an allen Prognosen für eine bestimmte Klasse ist. In der Modellevaluation wurden für verschiedene Mindest-Scores bzw. -Wahrscheinlichkeiten die resultierenden Precisions bestimmt. Filtert man nun die Prognosen, in dem man nur diejenigen verwendet, welche einen Score aufweisen, für den sich in der Modellevaluation eine Precision von mindestens 0.5 ergab, so beträgt die zu erwartende Quote korrekter Prognosen 50%. Die Precision Grenze ist grundsätzlich frei wählbar und sollte der individuellen Anwendung und Zielstellung gerecht werden. Die Intention einer weniger restriktiven Precision Grenze (0.5) besteht in der Eliminierung von den unsichersten Prognosen (geringste prognostizierte Wahrscheinlichkeiten bzw. Scores und damit höchste zu erwartende Fehlerquote), ohne jedoch zu viele Prognosen zu verwerfen. Eine Vertretung des Umweltbundesamtes hat diese Vorgehensweise suggeriert.
- Speicherung der Prognosen in der Azure SQL Datenbank.
  - Die, nach der Anwendung des Precision Filters verbleibenden, Prognosen müssen in der Azure SQL Datenbank gespeichert werden.
  - Für jeden Stoff wird dabei die Prognose mit der höchsten Wahrscheinlichkeit pro Zielvariable und Ausprägung in der Datenbank gespeichert.

- Durch Bereitstellung der Prognosen in der entsprechenden Datenbanktabelle („Substances“), in welcher die Daten aller Stoffmerkmale gespeichert sind, werden diese anschließend automatisch in der Regelausführung berücksichtigt.

► Unterscheidung von Prognosen und Originaldaten aus ChemInfo.

- Prognosen lassen sich sind in Datenbank anhand der fehlenden „SachverhaltId“ identifizieren und können so während der Regelausführung von ChemInfo Daten unterschieden werden.
- Das ist wichtig, da Prognosen nur zum Einsatz kommen, wenn die vorhergesagten Informationen nicht bereits aus anderen Feldern abgeleitet werden können, wie es beispielsweise in einigen Regeln zur Darstellung des Gefahrendiamanten vorkommt.

► Kennzeichnung von Datenblattinformationen, welche auf Chemprop Prognosen basieren.

- Bei der Ausführung der Regeln zur Vorbereitung der Informationen, welche auf den Datenblätter angezeigt werden, muss berücksichtigt werden, dass Regelergebnisse, welche auf Chemprop Prognosen, basieren, in der Datenbanktabelle der Regelergebnisse „HazardRulesResult“ gekennzeichnet werden müssen. Dies wird in Form einer zusätzlichen Spalte in der Ergebnistabelle realisiert.
- Diese Kennzeichnung kann anschließen im Rahmen der PDF-Erzeugung verwendet werden, um Machine Learning basierte Informationen visuell kenntlich zu machen.
- Hierzu wurden im Kick-Off Meeting am 06.02.2023 in Leipzig mit dem UBA mögliche Konventionen besprochen, wie modellbasierte Informationen auf dem Datenblatt gekennzeichnet werden können:
  - farbliche Hervorhebung (z. B. jeweiliges Segment im Gefahrdiamant ist gelb umrandet)
  - wenn die prognostizierte Ausprägung der zugrundeliegenden Zielvariable derjenigen mit dem geringsten Gefahrenpotenzial entspricht (z. B. NFPA.BA = 0), wird zusätzlich die geschätzte Wahrscheinlichkeit der Prognose angegeben (z. B. in der unteren Ecke des jeweiligen Segments im Gefahrendiamant)

4.3 Regelwerk zur Generierung von DB-Einträgen

Ohne die Verwendung von künstlicher Intelligenz lassen sich auf Basis der Entscheidungsbäume der Vorstudie[16] regelbasiert neue Sachverhalte erzeugen. So kann beispielsweise geprüft werden, ob Siedepunkt und Flammpunkt eines Stoffes in einem bestimmten Bereich liegen und dementsprechend die Brennbarkeit bestimmt werden. Ebenso betrifft es sämtliche Teile des NFPA-Gefahrendiamants. Hier können die Gefahrenkategorien bestenfalls direkt aus dem entsprechenden Merkmalsfeld (NFPA. *) entnommen und auf dem Faktendatenblatt angezeigt werden. Falls das Merkmalsfeld nicht befüllt ist, besteht die Möglichkeit aus anderen Merkmalsfeldern auf Basis der Entscheidungsbäume der Vorstudie regelbasiert die Gefahrenkategorien abzuleiten. Damit ein Stoff in eine Gefahrenkategorie eingestuft wird, muss die Merkmalsausprägung als Bedingung für eine Entscheidung erfüllt sein. Die Gefahrenkategorien werden einem Stoff aufgrund der Existenz entsprechender H-Sätze (z. B.

[16] Erarbeitung und Erfassung von Daten für gefährliche Stoffe zu Gefahren und Maßnahmen für den Datenbestand des gemeinsamen zentralen Stoffdatenpools des Bundes und der Länder (GSBL).

H220 - Extrem entzündbares Gas.) oder aufgrund textbasierter Regeln in Kombination mit physikalischen Messgrößen zugeordnet:

Einem Stoff wird die Gesundheitsgefahr Kat. 2 zugeordnet, wenn folgende Bedingungen erfüllt sind:

	(GGALL.GGALL enthält "tiefkalt" OR "Lokale Erfrierungen" OR „Flüssiggas")
OR	(STBE.STBE enthält "verflüssigtes Gas" OR "Flüssiggas"))
UND	(-55°C<=PC.SP.SP_LITUWRT<=-30°C)

Einem Stoff wird die Brandgefahr Kat. 4 zugeordnet, wenn folgende Bedingungen erfüllt sind:

(SEEG1272_08.KGH OR EG1272_08.KGH OR SEEG1272_08.EGH OR EG1272_08.EGH OR TRGS510.CLP OR SEEG1272_08.KZGH OR EG1272_08.KZGH)

enthält (H220 oder H222 oder H224 oder H230 oder H240 oder H260 oder H271 oder EUH018 oder EUH019)

Sollten Regeln zu unterschiedlichen Ergebnissen führen, wird das Ergebnis ausgegeben, das die größere Gefahr ergibt.

Da die so erhaltenen Ergebnisse nur in Anlehnung der rechtsverbindlichen Vorschrift (EG) Nr. 1272/2008 oder des Standards NFPA 704 erstellt wurden, muss zunächst geprüft werden, auf welchem Weg diese zurück in die Datenbank geschrieben werden sollen.

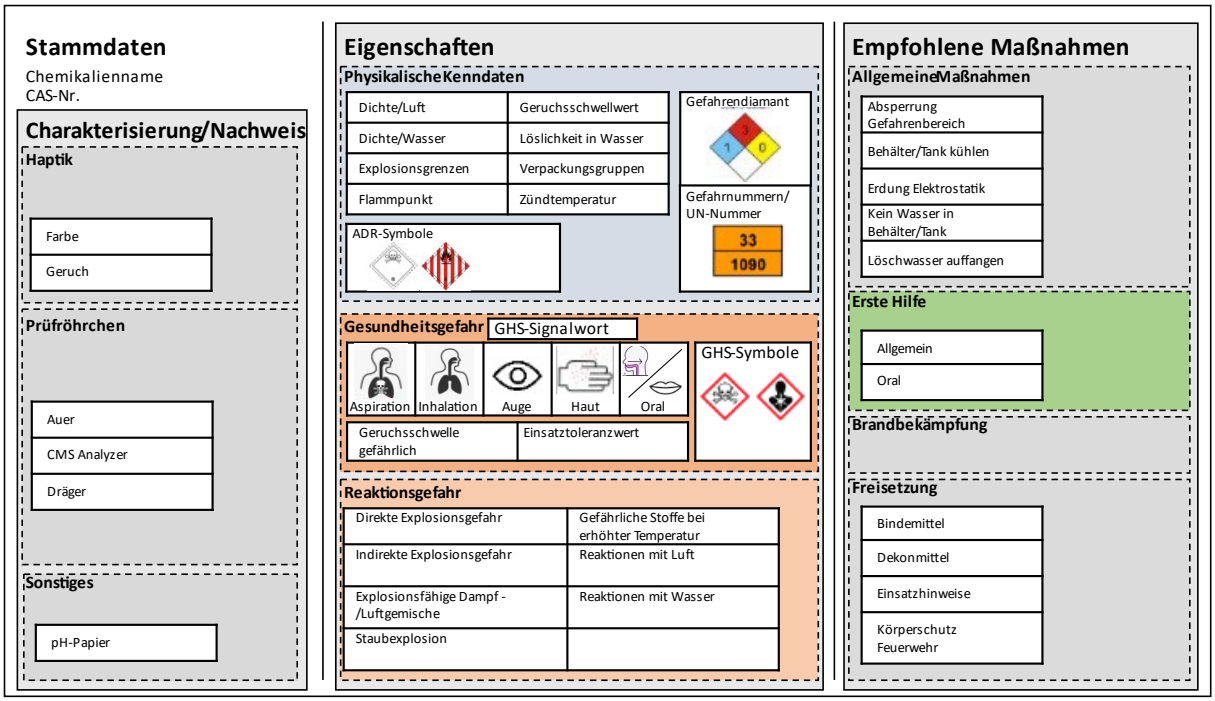
5 Regelwerk zur Verbalisierung der Faktendaten

5.1 Entwurf der Datenblätter

Um Einsatzkräften an Einsatzstellen schnelle und übersichtliche Informationen zu den beteiligten Gefahrstoffen zu geben, wurde ein Datenblattentwurf erstellt (Abbildung 23). Die Inhalte wurden in Abstimmung mit der ICT-Werkfeuerwehr festgelegt. Als Zielgruppe wurde vorrangig die Einsatzleitung der Feuerwehr identifiziert. Der Entwurf gliedert sich in vier Bereiche:

1. Stammdaten
2. Charakterisierung/Nachweis
   a. Haptik
   b. Prüfröhrchen
   c. Sonstiges
3. Eigenschaften
   a. Physikalische Kenndaten
   b. Gesundheitsgefahren
   c. Reaktionsgefahren
4. empfohlene Maßnahmen
   a. allgemeine Maßnahmen
   b. erste Hilfe
   c. Brandbekämpfung

Abbildung 23: Datenblattentwurf für Fachberatinnen und Fachberater der Feuerwehr



Quelle: eigene Darstellung, SoftwareOne.

Bei der Auswahl der anzuzeigenden Symbole wurde sich darauf verständigt, diese so allgemeinverständlich wie möglich zu gestalten. Dies ist vor allem bei den rechtsverbindlichen Symbolen wie GHS-, ADR-Symbolen, Gefahrentafel sowie Gefahrendiamant gegeben. Außerdem sollen bei den Gesundheitsgefahren die kritischen Expositionswege durch Anzeigen der betroffenen

Körperteile wie Lunge (jeweils tödliche Aspiration und Inhalation), Auge, Haut und Mund verdeutlicht werden. Weitere Symbole wurden zwar in Betracht gezogen, jedoch zugunsten unmissverständlicher Hinweistexte verworfen.

Aufgrund der umfangreichen Datengrundlage und dem modularen Regelaufbau können bei Bedarf Abwandlungen dieses Datenblattes abgestimmt auf weitere Benutzergruppen wie z. B. Fachberater Chemie, Rettungsdienst, Bevölkerung erstellt werden.

5.2 Fachlicher Hintergrund zum Regelwerk

Für die auf dem Faktendatenblatt anzuzeigenden Informationen ist nicht immer ein spezifisches Datenbankfeld vorhanden. So können die physikalischen Daten, wie z. B. Siede- oder Schmelzpunkt weitgehend direkt aus dem entsprechenden Datenbankfeld entnommen werden, während Einsatzhinweise, wie z. B. zur ersten Hilfe oder zur Auswahl geeigneter Löschmittel, zunächst regelbasiert extrahiert werden müssen. In anderen Fällen ist es notwendig, fehlende Informationen aus anderen existierenden Stoffeigenschaften regelbasiert zu generieren.

Die benötigten Regeln bauen auf den Ergebnissen der Vorstudie auf. Der vollständige Regelsatz ist in dem mitgeltenden Dokument „Regelwerk.xlsx“ enthalten.

Allgemeine Begriffserläuterungen zum Aufbau des Regelwerks:

- ID: eindeutige Regel ID, um eine Regel zu adressieren.
- Regelgruppe: Bezeichnet eine Untergruppe eines Anwendungsfalls (Use Case) z. B. „Gefahr durch erstickende Gase“.
- Merkmalsfelder: Dabei handelt es sich um die Datenbankfelder eines Stoffes, die zur Entscheidung herangezogen werden. Angegeben wird die Kurzbezeichnung aus der SQL-Tabelle "Substances".
- Merkmalsausprägung: Definition der Regellogik. Hierbei handelt es sich um das Kriterium (Regel), welches erfüllt sein muss, damit ein Text oder Piktogramm auf dem Datenblatt angezeigt wird.
- Hierarchie: Gibt wieder, ob eine Regel eigenständig ist (0) oder ob die Ausgabe vom Ergebnis weiterer Regeln abhängig ist (1 bzw. 2).
- Ausgabetext: Gibt an, welcher Text bei Regelerfüllung auf dem Datenblatt ausgegeben wird.
- Piktogramm/Tabelle: Gibt an, welches Piktogramm bei Regelerfüllung ausgegeben wird.
- Ebenen1-4: Gibt die jeweilige Gliederungsebene auf dem Faktendatenblatt wieder.
- Aggregationsmethode: Bedient sich eine Regel weiterer in der Priorität niedriger befindlicher Regeln, findet eine Regel-Aggregation statt. Die Aggregationsmethode ist hier angegeben (Erläuterung s. u.).

Im einfachsten Fall wird, wie oben beschrieben, direkt der Datenbankinhalt selbst oder ein entsprechendes Symbol wie z.B. das ADR-Symbol ausgegeben. Liegen mehrere Sachverhalte in einem Merkmal vor, müssen für die Erzeugung einer kompakten Ausgabe die Werte der Sachverhalte (W_Werte) zunächst aggregiert werden (**Werte-Aggregation**). In diesem Projekt wurden fünf verschiedene Typen für eine Werte-Aggregationen betrachtet, die im Folgenden zusammen mit den entsprechenden Regel-IDs aufgelistet sind:

1. zufällige Auswahl eines Werts (10 Regeln):
   21, 143, 238, 1006, 1007, 1008, 1015, 1020, 1021, 1024
2. kleinster oder größter Wert (11 Regeln):
   27, 56, 148, 149, 1002, 1003, 1004, 1010, 1011, 1018, 1012
3. Priorisierung der Ausgabe nach Ergebnis (1 Regel):
   1023
4. doppelte Werte streichen (Anwendung bei jeder Regel)
5. alle Regeln mit EG1272/SEEG1272: (37 Regeln):
   gedacht ist hier, dass zunächst auf EG1272_08.* zurückgegriffen wird und nur wenn dort keine Einträge vorhanden sind, sollen die Selbsteinstufungen in SEEG1272_08.* zu tragen kommen. Diese Regel wurde jedoch aufgrund des hohen Aufwandes nicht umgesetzt. Stattdessen wurden beide Felder gleichberechtigt betrachtet, was bei beidseitiger Befüllung möglicherweise zu einem höheren Gefährdungspotential führt.

Fließen mehrere Merkmale in eine Ausgabe ein, weil entsprechende Informationen entweder über mehrere Merkmale verteilt sind oder aus den Ergebnissen mehrerer bereits vorhandener Regeln generiert werden, muss eine **Regel-Aggregation** durchgeführt werden. Da diese Regeln die direkten Ausgaben für das Faktendatenblatt generieren, werden sie zuletzt ausgeführt (Hierarchie 2).

Die vier identifizierten Aggregationstypen werden nachfolgend aufgelistet und erklärt:

1. TYP1
   Eine Regel wird priorisiert. Erst wenn die Regel nicht greift, wird in der Ergebnismenge der Alternativregeln nachgesehen und der höchste Wert der Ergebnismenge ausgegeben. Dies wird zum Beispiel bei den Regeln zum NFPA-Gefahrendiamant angewendet. Für die Brandgefahr gibt Id26 den Originalwert aus dem Merkmal NFPA.BG der ChemInfo-Datenbank wieder. Ist das Merkmal nicht befüllt, wird auf die Ergebnisse der Regeln 22, 23, 24, 25, 27, 1001, 1002, 1003 und 1004 zurückgegriffen, die aus anderen Stoffeigenschaften Ziffern für die Brandgefahr generieren. Aus dieser Ergebnismenge wird durch die Regel-Aggregation der höchste Wert mit einem Sternchen markiert (z. B. „2*“) und ausgegeben. Die Markierung verdeutlicht, dass in diesem Fall die Brandgefahr mittels eines Algorithmus ermittelt wurde.
2. Typ2
   Die Resultate der zugrunde liegenden Regeln werden konkateniert wiedergegeben.
3. Typ3
   Wenn eine der zugrunde liegenden Regeln einen Wert zurückgibt („!=0“), wird die Regel ausgeführt. Dies ist zum Beispiel bei der Ermittlung einer potenziellen dermalen Expositionsgefahr (Id1036) der Fall. So wird die Anzeige des Piktogramms "GF_HAUT" empfohlen, wenn mindestens eine der Regeln Id90 (Sensibilisierung der Haut) oder Id222 (Dermale Sicherheitshinweise in EG1272_08.SIDER, die Haut betreffende H- und P-Sätze oder Expositionsweg „percutan“ o.ä. in SYMALLG.WEG) zutrifft.
4. Typ4
   Die zugrunde liegenden Regeln werden entsprechend einer Prioritätsreihenfolge ausgeführt, wobei nur das Ergebnis der ersten zutreffenden Regel ausgegeben wird. Beispielsweise wird vor der Ausgabe des Textes zur Selbstentzündungsgefahr an Luft (Id1042) das Ergebnis der Regeln 224 oder 225 ausgegeben, wobei die Id224 (Abfrage und Anzeige von H-Sätzen) Priorität vor der Id225 (Stichwortsuche in FREIEMP und GFRXREA, Ausgabe eines allgemeinen Textbausteins) hat. Id225 wird also nur ausgegeben, wenn Id224 kein Ergebnis geliefert hat.

Eine vollständige Auflistung aller auf dem Faktendatenblatt anzeigbarer Sachverhalte und die jeweils zugrundeliegenden Regeln sind in Tabelle 10 dargestellt. Die Aggregationsregeln sind fett markiert und sind von den hierarchisch niedrigeren Regeln gefolgt. Priorisierte Regeln sind unterstrichen.

Tabelle 10: Übersicht der auf dem Faktendatenblatt vorhandenen Felder mit jeweiliger Ausgaberegel.

Ebene1	Ebene2	Ebene3	Ebene4	Aggregationsregeln
Stammdaten	CAS-Nr.			1006
	Trivialname			1005
Charakterisierung/Nachweis	Haptik	Farbe		1008
		Geruch		1007
	Prüfröhrchen	AUER		**1047** (TYP2): 232, 233
		CMS Analyzer		229
		DRÄGER		230
	Sonstiges	pH-Papier		**1026** (TYP3): 21, 148 **1027** (TYP3): 143, 149
Eigenschaften	Gesundheitsgefahren	Einsatztoleranzwert		1024
		Expositionsweg	Inhalation_tödlich	154
			Inhalation	220
			Oral	**1038** (TYP4): 33>37>43
			Haut	**1036** (TYP3): 90 222
			Auge	**1037** (TYP3): 155, 223
		GHS-Signalwort		1023

Ebene1	Ebene2	Ebene3	Ebene4	Aggregationsregeln
		GHS-Symbole		1022
	Physikalische Kenndaten	ADR-Symbol		1017
		Dichte/Luft		1021
		Dichte/Wasser		1020
		Explosionsgrenzen		1009
		Flammpunkt		1010
		Gefahrendiamant	Brandgefahr	**1030** (TYP1): 22, 23, 24, 25, 26, 27, 1001, 1002, 1003, 1004
			Reaktionsgefahr	**1032** (Typ1)**:** 135, 136, 137, 138, 139, 224
			Besondere Anweisungen	**1033** (Typ2): 86, 132, 133
			Gesundheitsgefahren	**1031** (TYP1)**:** 121, 123, 124, 126, 127, 129
		Gefahrnummer		1016
		UN-Nummer		1015
		Geruchsschwelle gefährlich?		219
		Geruchsschwellwert		1012
		Löslichkeit in Wasser		238
		Verpackungsgruppe		1018

Ebene1	Ebene2	Ebene3	Ebene4	Aggregationsregeln
		Zündtemperatur		1011
	Reaktionsgefahren	direkte Explosionsgefahr		**1035** (TYP3): 77, 78
		Explosionsfähige Dampf/Luftgemische		**1041** (TYP3): 65, 68, 70, 71, 217
		GefährlicheStoffeBeiErhitzung		**1043** (TYP2): 218, 234
		indirekte Explosionsgefahr		81
		Reaktion_Luft		**1042** (TYP4): 224>225
		Reaktion_Wasser		**1050** (TYP2): 21, 64, 102, 103, 143, 148, 149
		Staubexplosion		**1044** (TYP4): 226>151>67
Maßnahmen	Allgemeine Maßnahmen	Absperrung/Gefahrenbereich		**1049** (TYP4): (227>228), 152
		Behälter/Tank kühlen		1029
		Erdung_Elektrostatik		**1048** (TYP3): 68, 70, 71, 77, 78, 81, 151
		kein Wasser in Behälter/Tank		**1046** (TYP3): 64, 102, 103, 104
		Löschwasser auffangen		**1045** (TYP3 + TYP2): 105, 119, 144
	Brandbekämpfung			**1034** (TYP2): 57, 111-118

Ebene1	Ebene2	Ebene3	Ebene4	Aggregationsregeln
	Erste Hilfe	Allgemein		**1051** (TYP4): 32>46>38
		Oral		**1038** (TYP4): 33, 37, 43
	Freisetzung	Bindemittel		1025
		Dekonmittel		1028
		Einsatzhinweise		**1039** (TYP2): 47-56
		Körperschutz Feuerwehr		**1040** (TYP2): 75, 150

Quelle: eigene Darstellung, SoftwareOne.

Im Folgenden werden die Regeln exemplarisch anhand der Beispiele Wasserlöslichkeit (Zuordnungsregel), Bildung gefährlicher Reaktionsprodukte bei Hitze (Texterkennung) und NFPA-Gefahrendiamant – Gesundheitsgefahr (Generierung neuer Inhalte) erläutert.

5.2.1 Beispielhafte Erläuterung anhand Wasserlöslichkeit (ID 238)

Bei der Wasserlöslichkeit handelt es sich um eine Regel, die abhängig von der im Merkmalsfeld WL.WL angegebenen Wasserlöslichkeit im Bereich von 15 °C bis 25 °C die verbalisierten Aussagen aus Tabelle 11 wiedergibt. Die Bezeichnungen und die Konzentrationsgrenzen orientieren sich am europäischen Arzneibuch[17].

Tabelle 11: Verbale Einstufung der Wasserlöslichkeit nach dem Europäischen Arzneibuch[18]

Bezeichnung	$g \cdot l^{-1}$ H_2O
sehr leicht löslich	>1000
leicht löslich	100 bis 1000
löslich	33 bis 100
wenig löslich	10 bis 33
schwer löslich	1 bis 10
sehr schwer löslich	0,1 bis 1
praktisch unlöslich	< 0,1

Quelle: Übertragung durch das Fraunhofer Institut für Chemische Technik aus dem Europäischen Arzneibuch, 8. Ausgabe, 2. Nachtrag, S. 5614 f.

5.2.2 Beispielhafte Erläuterung anhand der Bildung gefährlicher Reaktionsprodukte bei Hitze

Die Informationen zur Bildung gefährlicher Stoffe bei Erhitzen sind im Wesentlichen in den Merkmalsfeldern GFRXREA.GFRXREA und KONBRT.KONBRT gespeichert. Daher wurden die möglichen Ausprägungen beider Felder untersucht und Regeln erstellt, die eine Extraktion der relevanten Informationen erlauben.

Hierbei zeigte sich, dass GFRXREA.GFRXREA einen homogeneren Datenbestand hat. Es genügt die Suche nach den folgenden Textbausteinen und die anschließende Ausgabe des kompletten Inhalts von GFRXREA.GFRXREA:

- "Reaktivität mit Zersetzungsprodukten bei Feuer"
- "Reaktivität mit Zersetzungsprodukten bei Hitze"
- "Reaktivität bei Feuer:"
- "Reaktivität bei Hitze:"

Bei dem Merkmalsfeld KONBRT.KONBRT gestaltet sich die Textanalyse komplexer. Zunächst wird hier eine Untermenge mit folgenden Textbausteinen gebildet:

[17] Europäisches Arzneibuch, 8. Ausgabe, 2. Nachtrag, S. 5614 f. (1.4 Monographien).

[18] Europäisches Arzneibuch, 8. Ausgabe, 2. Nachtrag, S. 5614 f. (1.4 Monographien).

- "Bei Brand/Zersetzung"
- "Bei Erhitzung"
- "Bei starker Erhitzung"
- "Bei Brand/ therm.Zersetzung"
- "Bei therm.Zersetzung"
- "Beim Erwärmen"
- "Bei Brand"
- "An heißen Flächen"
- "Brandgase"

Anschließend wird weiter auf das Vorliegen einer der folgenden Begriffe/Begriffsfragmente gefiltert:

- ätzend
- giftig
- Bildung
- gesundheitsschäd
- narkotisch
- entwicklung
- entwickel
- reizung
- reizend
- Freisetzung
- Bildung
- Zersetzung
- schädigend
- gefährlich
- explosi
- entzünd

Fragmente wie z. B. „gesundheitsschäd“ bzw. „explosi“ schließen weitere Schreibweisen wie z. B. „gesundheitsschädlich“ oder „gesundheitsschädigend“ bzw. „explosiv“ oder „explosionsgefährlich“ mit ein. Sind die Kriterien beider Listen erfüllt, dann erfolgt die Ausgabe des Inhaltes von KONBRT.KONBRT.

Nach aktuellem Stand gibt es in beiden Merkmalsfeldern GFRXREA.GFRXREA und KONBRT.KONBRT einige wenige Ausnahmen, die keiner konsistenten Datenstruktur entsprechen. Diese Ausnahmen werden nicht berücksichtigt. Es wird daher empfohlen, diese Merkmalsfelder einem Review zu unterziehen und die Syntax entsprechend anzupassen.

Ein guter Ansatz wurde in den Merkmalsfeldern GFRXREA.B, GFRXREA.PRO und GFRXREA.S gefunden, denn dort werden die Informationen zu Reaktionsbedingungen (GFRXREA.B) und Reaktionsprodukten (GFRXREA.PRO) getrennt voneinander enthalten. In GFRXREA.S sind Informationen sowohl zu Reaktionspartnern als auch Reaktionsarten (z. B. Polymerisation, heftige Reaktion) oder beides (z. B. Heftige Reaktion mit Säuren) enthalten. Hier empfiehlt es sich für automatisierte Abfragen die Einheitlichkeit der Merkmalsausprägungen zu erhöhen.

5.2.3 NFPA-Gefahrendiamant – Gesundheitsgefahr

Die Aggregationsregel für den NFPA-Gefahrendiamant Gesundheitsgefahr ist die Regel mit der ID 1031. Sie überwacht die ihr zugrundeliegenden Regeln 121, 123, 124, 126, 127 und 129. Hierbei handelt es sich um eine TYP1-Aggregation. Da Regel 121 direkt auf das Merkmal NFPA.GF zugreift in dem die Gefahrenkategorie laut NFPA-Verordnung ermittelt wurde, erhält diese Regel bei der Aggregation die höchste Priorität. Erst wenn Das Merkmal NFPA.GF leer ist, wird aus den übrigen Regeln die Gefahrenkategorie ermittelt, die die größte Gefahr suggeriert.

- 121 - Merkmal NFPA.GF durchsuchen
  Ermittlung, ob das Merkmal NFPA.GF in der Substances-Tabelle gefüllt ist und ggf. den Wert zurückgeben.
- 123 - Gesundheitsgefahr „Kat. 2“
  Bei Vorliegen von H335, H319, H315 oder H317 wird die Gesundheitsgefahr Kat. 2 gesetzt.
- 124 – Gesundheitsgefahr „Kat. 2“
  Enthält das Merkmal “GGALL.GGALL” die Worte "tiefkalt", „lokale Erfrierungen" oder "Flüssiggas" oder das Merkmal STBE.STBE die Worte "verflüssigtes Gas" oder "Flüssiggas" liegt der Siedepunkt der Verbindung zischen -55°C und -30°C wird die Gesundheitsgefahr Kat. 2 gesetzt.
- 126 – Gesundheitsgefahr „Kat. 3“
  Bei Vorliegen von H314 oder H318 wird die Gesundheitsgefahr Kat. 3 gesetzt.
- 127 – Gesundheitsgefahr „Kat. 3“
  Enthält das Merkmal GGALL.GGALL die Worte "tiefkalt", "Lokale Erfrierungen" oder "Flüssiggas" oder das Merkmal STBE.STBE die Worte "verflüssigtes Gas" oder "Flüssiggas" und ist der Siedepunkt <= -55°C wird die Gesundheitsgefahr auf Kat. 3 gesetzt.
- 129 – Gesundheitsgefahr „Kat. 4“
  Bei Vorliegen von H300 oder H330 wird die Gesundheitsgefahr Kat. 4 gesetzt.

5.3 Technische Umsetzung

Als Datengrundlage für die Anwendung des Regelwerks dient die SQL-Datenbank „UBA_Export_Stoffe“ mit Daten aus der „Substances“ Tabelle. Als zweite Datenquelle wird eine Kopie der Datei „Regelwerk.xls“ aus dem Datalake (Storage Account) direkt verwendet, welche die Elemente aus den Spalten (Ausgabetext, Piktogramm/Tabelle, Ebene 1 bis 4) für die Ergebnismenge nach der Anwendung einer Regel, sowie die fachliche Logik für die

Implementierung der Regeln liefert In Tabelle 12 werden die Input Daten aufgelistet, welche im Abschnitt 5.2 bereits genauer beschrieben wurden.

Tabelle 12: Regelimplementierung - Input Daten

(Quelle) Tabelle	Spalte	Kommentar
(SQL DB) Substances	Id	Substance ID
(SQL DB) Substances	SachverhaltId	Sachverhalt ID
(SQL DB) Substances	Kurzbezeichnung	Kurzbezeichnung des Merkmals
(SQL DB) Substances	W_Wert	Wert
Regelwerk.xls	ID	Regel ID
Regelwerk.xls	Regelgruppe	Regel Name bzw. Anwendungsfall
Regelwerk.xls	Merkmalsfelder mit gleicher Ausprägung	**Logik** wird in Databricks (SQL, Python) umgesetzt
Regelwerk.xls	Merkmalausprägung	**Logik** wird in Databricks (SQL, Python) umgesetzt
Regelwerk.xls	Hierarchie	**Logik** wird in Databricks (SQL, Python) umgesetzt
Regelwerk.xls	Ausgabetext	**Ausgabewert** nach Anwendung der Regel
Regelwerk.xls	Piktogramm/Tabelle	**Ausgabewert** nach Anwendung der Regel

Quelle: eigene Darstellung, SoftwareOne.

Nach Anwendung des Regelwerks, werden die Ergebnisse in die SQL-Tabelle „HazardRulesResults“ geschrieben. In Tabelle 13 werden die Output Daten aufgelistet, welche unter anderem die Grundlage für die Erzeugung der Datenblätter ist.

Tabelle 13: Regelimplementierung - Output Daten

(Ziel) Tabelle	Spalte	Kommentar
(SQL DB) HazardRulesResults	Id	Entspricht der Substances ID
(SQL DB) HazardRulesResults	RuleId	Regel ID
(SQL DB) HazardRulesResults	SachverhaltId	konkatenierte Sachverhalte
(SQL DB) HazardRulesResults	Regelgruppe	Wert wird aus Regelwerk.xls übernommen
(SQL DB) HazardRulesResults	Ausgabetext	Wert wird aus Regelwerk.xls übernommen; Text enthält teilweise Platzhalter die in SQL/Python ausgewertet werden

(Ziel) Tabelle	Spalte	Kommentar
(SQL DB) HazardRulesResults	Piktogramm/Tabelle	Wert wird aus Regelwerk.xls übernommen

Quelle: eigene Darstellung, SoftwareOne.

Die Implementierung der Regeln erfolgt in einem Python Databricks Notebook und wird auf Databricks Clustern ausgeführt. Die fachliche Grundlage bzw. Regellogik liefert die Excel-Datei Regelwerk.xls. Im Notebook wird für jede Regel eine eigene Funktion erstellt, in welcher die Logik in Python und SQL umgesetzt wird. Dabei ist die Regel ID aus der der Regelwerk.xls Teil des Funktionsnamens, um eine eindeutige Zuordnung zu gewährleisten. Jede Funktion folgt einem ähnlichen Schema, welches am folgenden Beispiel in Abbildung 24 für Regel 32 exemplarisch erklärt wird:

Abbildung 24: Python Code Beispiel zu Regel 32

```python
def Regel32(excel_rules:pd.DataFrame,
            regel_id:int = 32,
            query:str = ("SELECT [Id] "
                         "       ,[Thesaurus] "
                         "       ,[SachverhaltId] "
                         "       ,[Kurzbezeichnung] "
                         "       ,[W_Typ] "
                         "       ,[W_Wert] "
                         " FROM [dbo].[Substances] "
                         " WHERE [Kurzbezeichnung] IN ('SEEG1272_08.KGH', "
                         "                             'EG1272_08.KGH', "
                         "                             'SEEG1272_08.EGH', "
                         "                             'EG1272_08.EGH') AND "
                         "       [W_Wert] LIKE '%H304%'")) -> None:

    rules = excel_rules[excel_rules['ID'] == regel_id].reset_index(drop = True)
    substance_content = HazardRulesSupport().get_query_result_pandas_df(query)

    substance_content['RuleId'] = rules['ID'].iat[0]
    substance_content['Kategorisierungsdimension'] = rules['Kategorisierungsdimension'].iat[0]
    substance_content['Ausgabetext'] = str(rules['Ausgabetext'].iat[0])
    substance_content['Piktogramm/Tabelle'] = str(rules['Piktogramm/Tabelle'].iat[0])
    substance_content['SachverhaltId'] = substance_content['SachverhaltId'].astype(str)
    substance_content_grouped = substance_content.groupby(['Id', 'RuleId', 'Kategorisierungsdimension',
                                                           'Ausgabetext' ,'Piktogramm/Tabelle'], as_index = False)
    substance_content = substance_content_grouped.agg({'SachverhaltId': lambda x: '||'.join(list(set(x)))})
    substance_content['SachverhaltId'] = substance_content['SachverhaltId'].where(~(substance_content['SachverhaltId']=='nan'), None)
    HazardRulesSupport().store_rule_result(substance_content)
```

Quelle: eigene Darstellung, SoftwareOne.

Die Abbildung 24 ist wie folgt zu lesen:

- Zeile 1 bis 14: hier wird der Name der Funktion „Regel32“ deren Eingangsparameter definiert:
  - 1. „excel_rules:pd.DataFrame“: ist ein Python Pandas Dataframe der die Tabelle aus dem Regelwerk.xls enthält.
  - 2. „regel_id:int“: wird direkt im Funktionskopf definiert und entspricht der Regel Id aus dem Funktionsnamen.
  - 3. „query:str“: wird direkt im Funktionskopf definiert und enthält einen String mit SQL Syntax (Zeile 3 bis 14). In diesem Fall enthält die SQL Abfrage die gesamte Logik der Regel 32 aus der Regelwerk.xls Datei.
- Zeile 16: der Dataframe wird auf die Regel ID = 32 gefiltert.

- Zeile 17: mit der Hilfsfunktion „get_query_results_pandas_df“ wird die definierte SQL Abfrage an die Datenbank abgeschickt. Das Ergebnis ist eine Tabelle in Form eines Python Pandas Dataframe „substance_content“ mit den Stoffen, für welche die Regellogik aus dem SQL Statement angewendet werden konnte.
- Zeile 19 bis 27: Die Datentypen der Tabelle werden zu Strings umgewandelt und die Werte für Sachverhalte konkateniert.
- Zeile 28: Die Transformierte „substance_content“ Tabelle wird in die SQL Datenbank geschrieben und den Einträgen in der Tabelle „HazardRulesResults“ hinzugefügt.

Ein exemplarischer Auszug aus der Ergebnistabelle wird in Abbildung 25 für die Substance Id = 11158 dargestellt.

Abbildung 25: Beispielauszug aus der Tabelle "HazardRulesResults"

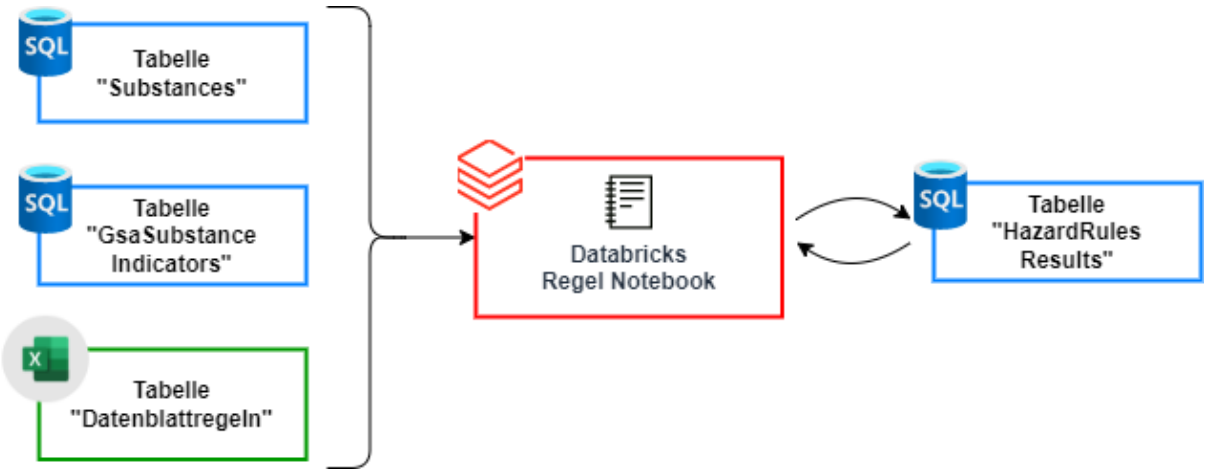
	Id	RuleId	SachverhaltId	Regelgruppe	Ausgabetext	Piktogramm/Tabelle
1	11158	27	5353573	Brandgefahr Kat. 3	False	<Gefahrendiamant mit BG = 3>
2	11158	39	3635295	EH_leere_Merkmale_EHAUGE	Sofort mit viel Wasser spülen; Bei andauernder Reiz...	False
3	11158	40	3531134	EH_leere_Merkmale_EHHAUT	Sofort mit viel Wasser spülen; Verwendung von Seif...	False
4	11158	42	3326240	EH_leere_Merkmale_EHINH	Opfer an die frische Luft bringen; Atemschwierigkeit...	False
5	11158	45	5071953	EH_leere_Merkmale_EHORAL_trinken1	Mund mit Wasser spülen; Bei Unwohlsein: Arzt/medi...	False
6	11158	48	5186640	Einsatzhinweis "entzündbar, flüssig"	Alle Zündquellen entfernen. Nur funkenfreies Werkz...	False
7	11158	53	2811486	Einsatzhinweis "flüssig, organisch"	Auslaufende Flüssigkeit auffangen/ eindämmen/ ab...	False
8	11158	55	1905272	Einsatzhinweis "generell"	In geeigneten Behälter der Rückgewinnung oder En...	False
9	11158	56	4558673	Einsatzhinweis "Tiefergelegene Bereiche meiden"	Tiefergelegene Bereiche meiden, wenn möglich abdi...	False
10	11158	57	4170679	Empfohlene Löschmittel	False	TRUE / TRUE
11	11158	75	5186640	Gasdichter Chemikalienvollschutzanzug	Gasdichten Chemikalienvollschutzanzug tragen	False
12	11158	78	3550492||3770569	Gefahr der direkten Explosion	False	True
13	11158	232	2892749	Prüfröhrchen_Auer_1	Überschrift: Auer||Kurzzeitröhrchen VC-1	False

Quelle: eigene Darstellung, SoftwareOne.

Der Prozess der Regelausführung ist in Abbildung 26 dargestellt. Die SQL-Tabellen sind in der Datenbank „UBA_Export_Stoffe“ angelegt und die „Regelwerk.xls“ Datei mit dem Tabellenblatt „Datenblattregeln“ ist im Datalake abgelegt und wird direkt vom Databricks Notebook ausgelesen.

Für Regeln der Hierarchiestufe 1 (siehe Tabellenspalte „Hierarchie“ in Datei Regelwerk.xls) werden die Ergebnisse der „HazardRulesResults“ Tabelle als zusätzliche Eingabe für Regelanwendung wiederverwendet. Aus diesem Grund werden im Notebook zuerst alle Regeln der Hierarchiestufe 0 ausgeführt. Die Tabelle „GsaSubstanceIndicators“ enthält die Spalten „Substance ID“ und „Sachverhalts ID“ aus dem GSA-Datenbestand und wird genutzt, um in einigen Regeln auf relevante Einträge zu filtern. Das bedeutet, dass in diesen Fällen, nur Sachverhalte betrachtet werden, die auch im GSA-Datenbestand vorliegen.

Abbildung 26: Übersicht Regelausführung



Quelle: eigene Darstellung, SoftwareOne.

5.4 Status der technischen Umsetzung des Regelwerks

In diesem Projekt konnte das Regelwerk wegen des großen fachlichen Umfangs nicht vollständig technisch implementiert werden. Im Regelwerk.xls Dokument wird der Status in der Spalte „Status technische Umsetzung“ kenntlich gemacht. Diese Spalte hat folgende 3 möglichen Einträge:

1. Implementiert: bedeutet die Regel wurde implementiert und das Ergebnis wurde fachlich überprüft
2. Implementierter Code muss angepasst werden: bedeutet das nach dem ersten Review der Regelergebnisse die fachliche Logik noch einmal angepasst wurde. Diese konnte jedoch nicht mehr technisch umgesetzt werden.
3. nicht implementiert - nur fachliche Konzeption: bedeutet, dass die Regel technisch nicht implementiert wurde. Das betrifft vor allem die Regeln zur Aggregation mit Hierarchie Ebene 2.

Weitere Punkte, die technisch nicht mehr umgesetzt wurden:

- Die Spalten Ebene 1, Ebene 2, Ebene 3 und Ebene 4 wurden nicht technisch umgesetzt
- Die Spalte „Aggregationsmethode“, welche die Werte-Aggregation beschreibt, wurde nicht technisch umgesetzt

6 Quellenverzeichnis

Monografien

Europäisches Arzneibuch, 8. Ausgabe, 2. Nachtrag, S. 5614 f. (1.4 Monographien).

Europäisches Arzneibuch, 8. Ausgabe, 2. Nachtrag, S. 5614 f. (1.4 Monographien).

Internetadressen

Umweltbundesamt (2023): Dokumentvorlagen. https://www.umweltbundesamt.de/dokumentvorlagen (30.08.2023)

GitHub, Inc. (2023): rdkit / rdkit. https://github.com/rdkit/rdkit (05.09.2023)

KNIME Community Forum (2011): Molecule Functionality node Request. Molecule Functionality node Request - Community Extensions - KNIME Community Forum (05.09.2023)

NCI/CADD Group (2023): Chemical Identifier Resolver. https://cactus.nci.nih.gov/chemical/structure (05.09.2023)

CAS Common Chemistry (2023): Cadmate(6-), [[[1,2-ethanediylbis[(nitrilo-κN)bis(methylene)]]tetrakis[phosphonato-κO]](8-)]-, potassium hydrogen (1:5:1), (OC-6-21)-. https://commonchemistry.cas.org/detail?cas_rn=68309-98-8 (05.09.2023)

National Library of Medicine (2023): PubChem. https://pubchem.ncbi.nlm.nih.gov/ (05.09.2023)

Ambinter c/o Greenpharma (2023): Startsite. https://www.ambinter.com/ (05.09.2023)

ECHA European Chemicals Agency (2023): Advanced Search for Chemicals. https://echa.europa.eu/advanced-search-for-chemicals?p_p_id=dissadvancedsearch_WAR_disssearchportlet&p_p_lifecycle=0&p_p_col_id=column-1&p_p_col_count=1 (05.09.2023)

Towards Data Science (Matthias Gruber) (2021): Chemical Predictions with 3 lines of code. https://towardsdatascience.com/chemical-predictions-with-3-lines-of-code-c4c6a4ce7378 (14.12.2022)

ACS Publications (Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay) (2019): Analyzing Learned Molecular Representations for Property Prediction. https://pubs.acs.org/doi/full/10.1021/acs.jcim.9b00237 (14.12.2022)

GitHub, Inc. (2023): chemprop / chemprop. https://github.com/chemprop/chemprop#molecular-property-prediction (14.12.2022)

Kyle Swanson, Kevin Yang, Wengong Jin, Lior Hirschfeld, Allison Tam (2020): chemprop. https://chemprop.readthedocs.io/en/latest/index.html (14.12.2022)

Kyle Swanson, Kevin Yang, Wengong Jin, Lior Hirschfeld, Tommi Jaakkola, Regina Barzilay (2020): Chemprop++. https://docs.google.com/presentation/d/14pbd9LTXzfPSJHyXYkfLxnK8Q80LhVnjImg8a3WqCRM/edit?pli=1#slide=id.p (14.12.2022)

CellPress (Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Tommi S. Jaakkola, Regina Barzilay, James J. Collins) (2020): A Deep Learning Approach to Antibiotic Discovery. https://www.cell.com/cell/fulltext/S0092-8674(20)30102-1 (14.12.2022)

Universität Zürich (2023): Mann-Whitney-U-Test. https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/unterschiede/zentral/mann.html (07.12.2022)

A Mitgeltende Unterlagen

Bei den nachfolgenden Anlagen handelt es sich um Dokumente und Ergebnisse, die dem Umweltbundesamt gesondert zur Verfügung gestellt wurde, da dies nicht an das Dokument anhängbar ist.

A.1 Analyse_Merkmalsverteilung

Es handelt sich hierbei um ein Ergebnis aus Arbeitspaket 1.

A.2 Chemprop_Modelle

Es handelt sich hierbei um ein Ergebnis aus Arbeitspaket 2

A.3 Datenmodell_Kategorisierung_Nach_Fraunhofer_v2

Es handelt sich hierbei um ein Ergebnis aus Arbeitspaket 1.

A.4 Informationen_aus_Strukturdaten

Es handelt sich hierbei um ein Ergebnis aus Arbeitspaket 1.

A.5 Regelwerk

Es handelt sich hierbei um das Ergebnis aus Arbeitspaket 3.

A.6 Textmining_Ergebnismengen

Es handelt sich hierbei um ein Ergebnis aus Arbeitspaket 2.