

Informationssammlung

Anwendung von KI-Modellen bei H₂-Leckagen

Ergebnisse einer Literaturrecherche

Autor: Dr. Werner Fröhlingsdorf | Stand: Oktober 2026

Inhaltsverzeichnis

1	Einführung.....	3
2	KI-Modelle bei H ₂ -Leckage – ein Überblick der Literaturquellen in Tabellenform.....	4
3	Literaturquelle 1: Machine learning regression CFD models for predicting hydrogen dispersion in a naturally ventilated area	6
4	Literaturquelle 2: Leak localization using distributed sensors and machine learning for hydrogen releases from a fuel cell vehicle in a parking garage.....	9
5	Literaturquelle 3: CFD Simulation and ANN Prediction of Hydrogen Leakage and Diffusion Behavior in a Hydrogen Refuelling Station	15
6	Literaturquelle 4: Data-augmented deep learning for hazard assessment of hydrogen accumulation in confined spaces: Multitask prediction and sensitivity analysis	19
7	Literaturquelle 5: Real-time hydrogen release and dispersion modelling of hydrogen refuelling station by using deep learning probability approach	23
8	Literaturquelle 6: Hybrid neural network-based surrogate model for fast prediction of hydrogen leak consequences in hydrogen refueling station	25
9	Literaturquelle 7: Integrated CFD, Bayesian optimization, and deep learning for hydrogen leak detection on offshore renewable-energy platforms.....	31
10	Hyperparameter	35
11	Literatur	37



1 Einführung

Mithilfe der durchgeführten Literaturrecherche sollte untersucht werden, in welchem Umfang KI-Modelle im Zusammenhang mit Wasserstoff-Leckagen verwendet werden. Bei einer Wasserstoff-Leckage entsteht eine Gefahrensituation, für welche eine schnelle Analyse von Daten (z. B. Messergebnisse der H₂-Konzentration) dazu beitragen kann, die Gefährdung besser einzuschätzen, um geeignete Gegenmaßnahmen zu ergreifen.

Es wurden 7 Literaturquellen ausgewertet, bei denen mit Hilfe von Simulationen und/oder Messungen der H₂-Ausbreitung KI-Modelle erstellt wurden.

Die nachfolgenden Auswertungen beschreiben:

- Was wurde untersucht
- Eingangs- und Ausgangsparameter der KI-Modelle
- Aufbau der KI-Modelle
- mathematische Methoden, um die KI-Modelle zu erzeugen
- eventuelle Besonderheiten

Die Auswertungen erheben keinen Anspruch auf Vollständigkeit. Die in den Literaturquellen verwendeten mathematischen Methoden können nur vereinfacht bzw. ansatzweise beschrieben werden.

Im vorletzten Kapitel 10 erfolgt eine Auflistung der wichtigsten Hyperparameter, welche in KI-Modellen verwendet werden.



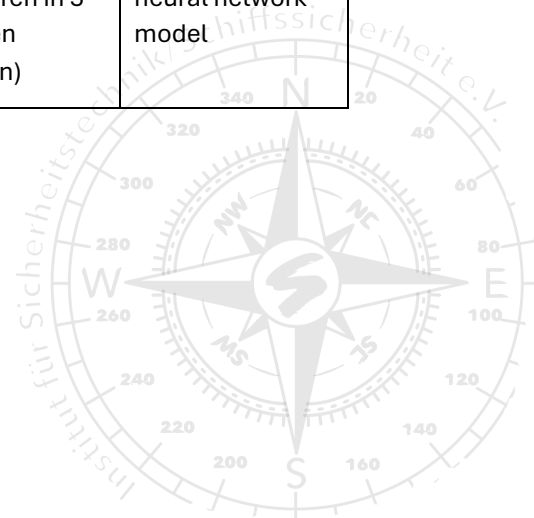
2 KI-Modelle bei H₂-Leckage – ein Überblick der Literaturquellen in Tabellenform

Tabelle 1: Überblick zu den betrachteten Szenarien, der Datenerzeugung (Simulation und/oder Messung), H₂-Massen- oder Volumenstrom, Zeitdauer Einspritzen /Versuch, H₂-Menge

Literatur- quelle	Szenario	Simulation und/oder Messung	H ₂ -Massen- oder Volumenstrom	Zeitdauer Einspritzen/ Versuch	H ₂ - Menge
1	geschlossener Raum 2.6 m ³ mit 2 Lüftungsöffnungen	CFD-Simulation (Fluent)	56.6 L/min (20 °C, 1 atm)	1100 s	1 m ³
2	Parkgarage für BZ-Fahrzeuge (80 Parkplätze), 60 m x 36 m x 5.8 m	Messungen He- Konz. in 1/24 Modell	3.4 L/min	220 s	12.47 L
3	H ₂ -Tankstelle 141 m x 123 m x 80 m	CFD-Simulation (Fluent)	1.5589 kg/s bis 4.5589 kg/s,	30 s	46.8 kg bis 136.8 kg
4	geschlossener Raum 30 m ³ bis 200 m ³ mit 2 Lüftungsöffnungen	Simulation mit HyRAM+ (Hydrogen Risk Assessment Models Plus)	keine direkte Angabe, 250, 350, 450 bar H ₂ - Druck, Düsendurchmesser 0.068 mm bis 6.8 mm	keine Angabe	keine Angabe
5	H ₂ -Tankstelle 29 m x 28 m x 6.4 m	Simulation mit FLACS (Flame Acceleration Simulator)	0.46 kg/s bis 1.46 kg/s	70 s	32.2 kg bis 102.2 kg
6	H ₂ -Tankstelle, 80 m x 40 m x 10 m	CFD-Simulation (Fluent), Experimente mit He mit skaliertem Modell	0.60 kg/s für den Betankungsschlauch und 2.15 kg/s für die Betankungsdüse	30 s/40 s	18 kg, 64.5 kg
7	Offshore Plattform zur Erzeugung von grünem H ₂ , Methanol, Ammoniak	CFD-Simulation (Fluent)	0.282 kg/s bis 1.128 kg/s	30 s	8.46 kg bis 33.84 kg

Tabelle 2: Überblick zur Datenerzeugung (Simulation und/oder Messung), Anzahl der Varianten, Eingangs- und Ausgangsparameter des KI-Modells und der verwendeten mathematischen Methoden zur Erstellung des KI-Modells

Literatur	Simulation und/oder Messung	Anzahl Varianten	Eingangsparameter	Ausgangsparameter	mathematische Methoden
1	CFD-Simulation (Fluent)	29	Abstände R ₂ , R ₄	H ₂ -Konzentration im Raum	XGBoost MLP K-NN
2	Messungen He-Konz.	154	gemessene He-Konz.	Leckageort (welches Fahrzeug)	ANN K-DTW (K-NN+ DTW)
3	CFD-Simulation (Fluent)	240	H ₂ -Massenstrom, H ₂ -Ausströmzeit, Windgeschwindigkeit	horizontale und vertikale Ausdehnung der H ₂ -Wolke	BP ANN
4	Simulation mit HyRAM+	61888	floor area, release pressure, vent area, flow rate, distance to wall, orifice diameter, release height, room height	Max. overpressure, max. concentration, Time to overpressure	VAE (Variational Autoencoder), DL-based model (ANN), baseline models (RF, GB, SVM, KNN, XGB)
5	Simulation mit FLACS	70	H ₂ -Freisetzungsrate, Windgeschwindigkeit 1 m/s bis 7 m/s, Wind in Ausströmrichtung oder dazu entgegengesetzt	H ₂ -Konzentrationsfeld/ Ausdehnung	ANN Monte Carlo dropout
6	CFD-Simulation (Fluent), Experimente mit He	32 (Simulation) 16 (Experiment)	Leckagerate, Leckageort, Einspritzen nach oben oder unten	2D Verteilung H ₂ -Konzentration, zeitabhängige He-Konzentration	GAN, LSTM, DNN, SSIM
7	CFD-Simulation (Fluent)	420	Leckageort, Leckagedurchmesser, Windgeschwindigkeit, Windrichtung	optimale Verteilung von Sensoren in 5 Ebenen (Höhen)	Transformer-LSTM hybrid neural network model



3 Literaturquelle 1: Machine learning regression CFD models for predicting hydrogen dispersion in a naturally ventilated area

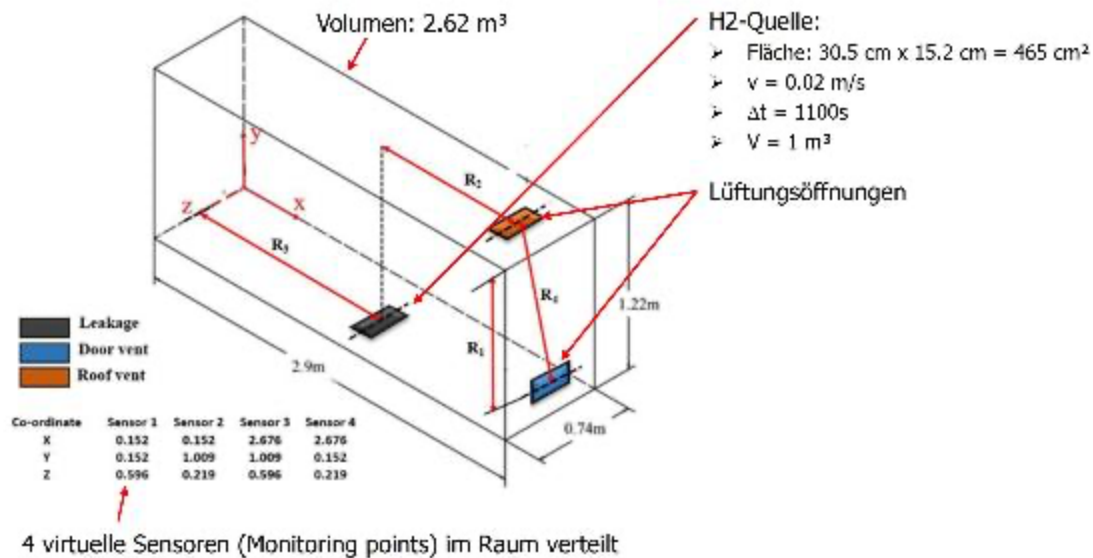


Abbildung 1: Literaturquelle 1 - CFD-Modell zur Simulation der H₂-Ausbreitung in einem Raum mit 2 Lüftungsöffnungen

Es wurden mit dem in Abbildung 1 dargestellten CFD-Modell 29 Varianten der H₂-Ausbreitung mit Variation der Maße R_1 , R_2 , R_3 und R_4 simuliert. Durch die beiden Lüftungsöffnungen erfolgt ein Gasaustausch mit der Umgebung. Der aufwärts strömende Wasserstoff vermischt sich mit Luft. Durch die obere Lüftungsöffnung kann ein H₂/Luft-Gemisch in die Umgebung ausströmen und Umgebungsluft durch die untere Lüftungsöffnung in den Raum einströmen.

Zielstellung: Optimierung der Maße R_1 , R_2 , R_3 und R_4 zum Erreichen einer möglichst geringen H₂-Konzentration im Raum.

Die Parameter R_2 und R_4 wurden als Eingangsparameter für das maschinelle Lernen verwendet. Ausgangsparameter des KI-Modells ist die H₂-Konzentration im Raum.

Durch Anwendung des KI-Modells soll eine schnelle Vorhersage der H₂-Konzentration im Raum möglich sein.

Die in Abbildung 2 dargestellte Methodik zeigt, dass 75 % der Daten (22 Varianten) zum Training und 25 % der Daten (7 Varianten) zum Test des entwickelten KI-Modells verwendet wurden. Hierzu ist anzumerken, dass Trainings- und Testdaten für die Entwicklung eines KI-Modells strikt voneinander getrennt werden sollten, weil die Testdaten für das KI-Modell die „reale Welt“ wiedergeben, d. h. Daten, welche das KI-Modell vorher noch nie gesehen hat. Die „reale Welt“ ist dann so real wie die Simulationsdaten diese wiedergeben. Zwischen Simulation und der „realen Welt“ sind im allgemeinen immer Abweichungen möglich, welche durch unterschiedliche Randbedingungen der Simulation zur „realen Welt“ entstehen können.

Für das KI Modell wurden 3 Modelle (Methoden) verwendet:

- **XGBoost** (Extreme Gradient Boosting): Es erfolgt eine Kombination von mehreren einfachen Modellen (mit entsprechender Ungenauigkeit) zu einem relativ genauen Gesamtmodell. Ein Optimierungsalgorithmus zur Minimierung der Verlustfunktion (z. B. mittlerer quadratischer Fehler) kann in Python programmiert werden. Die verwendete Lernrate (Schrittweite) ist 0.1 (s. Tabelle 3). Es wurden 50 Entscheidungsbäume (No. Of estimators in Tabelle 3) verwendet. Die Bäume werden nicht gleichzeitig, sondern nacheinander (sequentiell) aufgebaut. Jeder neue Baum versucht gezielt, die Fehler der vorherigen Bäume zu korrigieren. Der „Fehler“ (Residuum) ist der Unterschied zwischen dem tatsächlichen Wert und der aktuellen Vorhersage des Modells. Bei XGBoost lernt Baum (n) den Fehler von Baum (n-1) zu bestimmen. Baum (n+1) lernt dann diesen reduzierten Fehler wiederum von Baum (n) und kann so den Fehler des vorhergesagten Wertes immer weiter verringern. Ein Entscheidungsbaum (Decision Tree) ist ein weit verbreitetes Modell im Machine Learning, dass Entscheidungen wie ein Flussdiagramm darstellt. Er zerlegt eine komplexe Fragestellung Schritt für Schritt in einfache Ja/Nein-Fragen, um am Ende eine Vorhersage zu treffen.
- **MLP** (Multi-Layer-Perceptron): Neuronales Netz, bestehend aus Eingabeschicht/verborgenen Schichten (eigentliche Rechenleistung)/Ausgabeschicht. Es wurden zwei verborgene Schichten (Hidden Layers) mit jeweils 150 Neuronen verwendet (s. Tabelle 3). Die Schrittweite ist ein Hyperparameter, der eingestellt werden muss. Die Schrittweite bestimmt, wie groß die Schritte sind, die das Modell beim Lernen macht, um seine Fehler zu korrigieren. Die in Tabelle 3 angegebene Schrittweite hat einen Startwert von 0.9 (sehr großer Wert) und wird auf $\alpha = 0.001$ reduziert. Es wird eine dynamische Schrittweite verwendet, welche im Verlauf des Trainings reduziert wird. Eine zu hohe Schrittweite macht den Trainingsprozess instabil (vergleichbar mit einem zu großen Zeitschritt bei der Simulation eines zeitabhängigen Prozesses). Eine zu niedrige Schrittweite erhöht die erforderliche Rechenzeit.
- **K-NN** (k-Nächste-Nachbarn): Ein Algorithmus sucht nach dem „k-ähnlichsten“ Nachbar. Der Wert k bestimmt, wie viele der nächsten Nachbarn betrachtet werden sollen. Bei k = 9 sucht die KI die neun ähnlichsten Datenpunkte. In der Praxis wählt man bei einer Ja/Nein-Entscheidung (binäre Klassifikation) fast immer eine ungerade Zahl für die Nachbarn. Bei k = 9 kann es niemals zu einem Gleichstand (Unentschieden) bei der Abstimmung kommen. Es gibt immer eine klare Mehrheit.



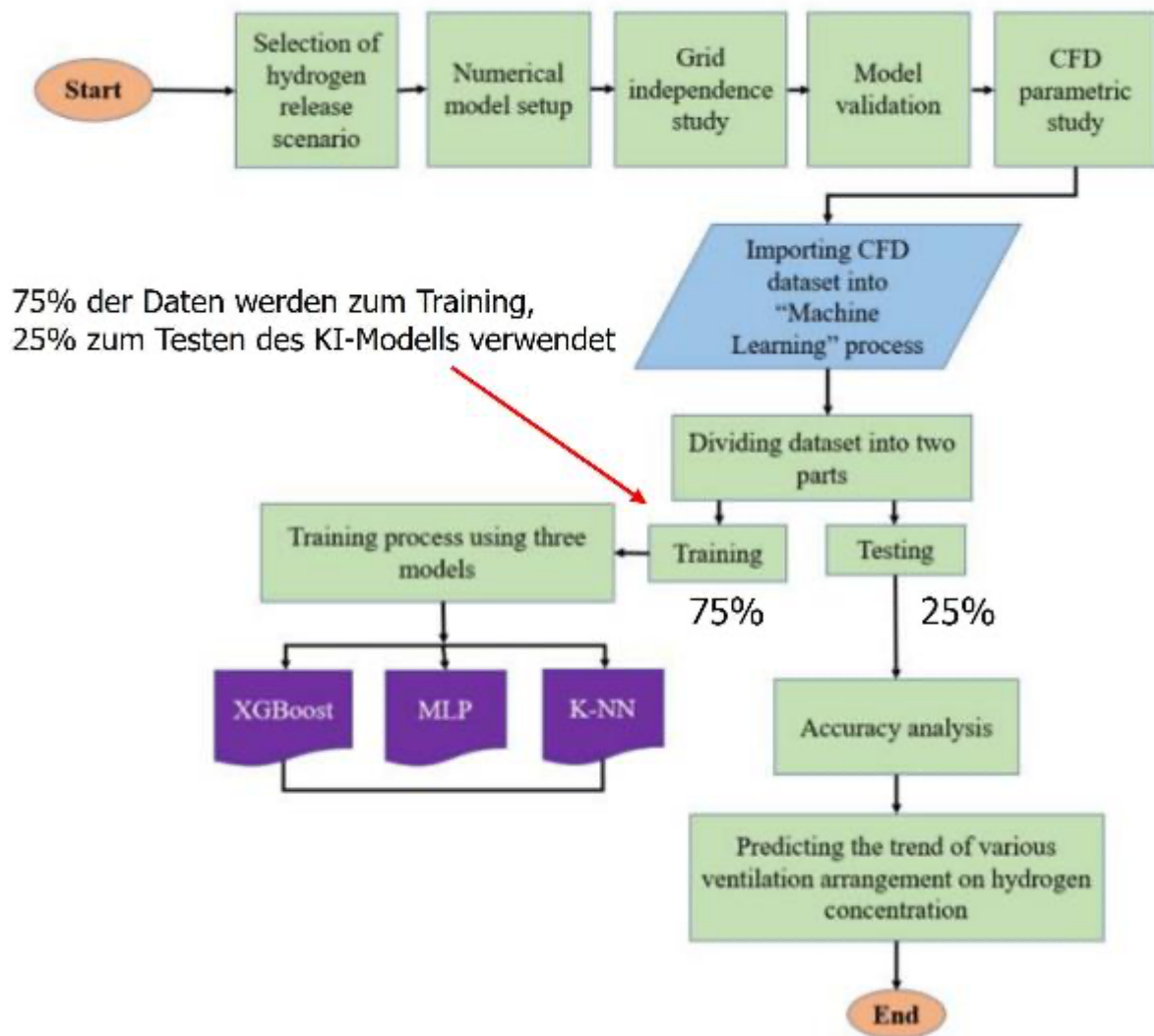


Abbildung 2: Literaturquelle 1 - Beschreibung der Methodik

Tabelle 3: Literaturquelle 1 - verwendete Parameter der Regressionsmodelle

No.	MLAs	Key parameters
1	XGBoosting	Learning rate = 0.1, No. of estimators = 50
2	MLP	$\alpha = 0.001$, Hidden layer = (150,150), Learning rate = 0.9
3	K-NN	No. of neighbours = 9



4 Literaturquelle 2: Leak localization using distributed sensors and machine learning for hydrogen releases from a fuel cell vehicle in a parking garage

Zielstellung ist die Detektion der H₂-Leckage von einem Brennstoffzellen-Fahrzeug in einer Parkgarage mit 80 Parkplätzen (s. Abbildung 3).

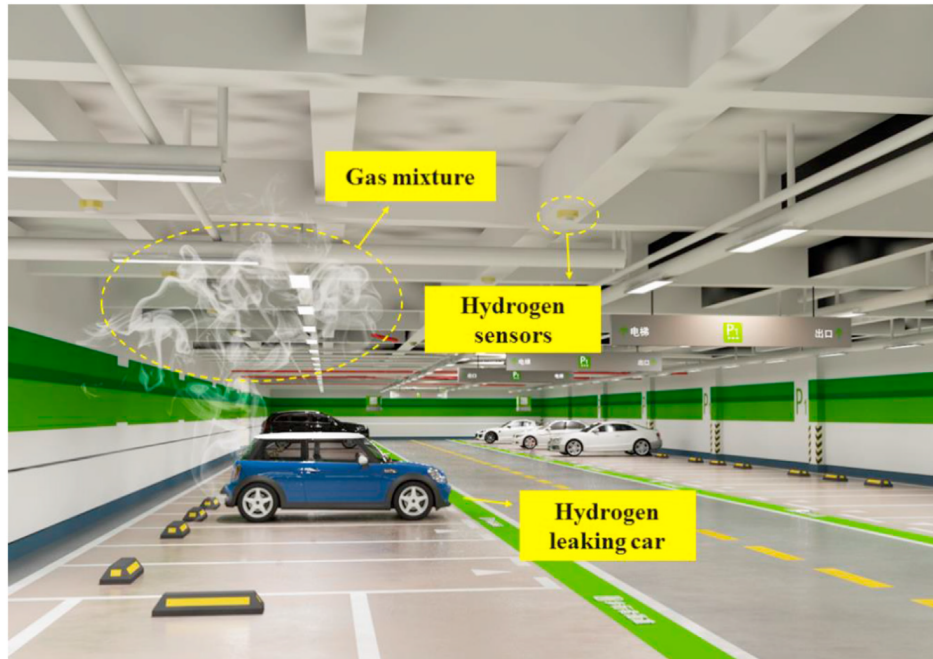


Abbildung 3: Literaturquelle 2 - H₂-Leckage eines Brennstoffzellen-Fahrzeugs und H₂-Ausbreitung in einer Parkgarage

Als Versuchsanlage wird ein 1/24 Modell (Volumen: 0.9 m³) der Parkgarage verwendet (s. Abbildung 4).

Anstelle H₂ wird bei den Experimenten Helium eingesetzt:

- 3.4 L/min
- $v = 11.5 \text{ m/s}$ (2.5 mm Schlauch)
- Einspritzdauer 220 s
- $V = 12.47 \text{ L}$
- Versuchsdauer 600 s



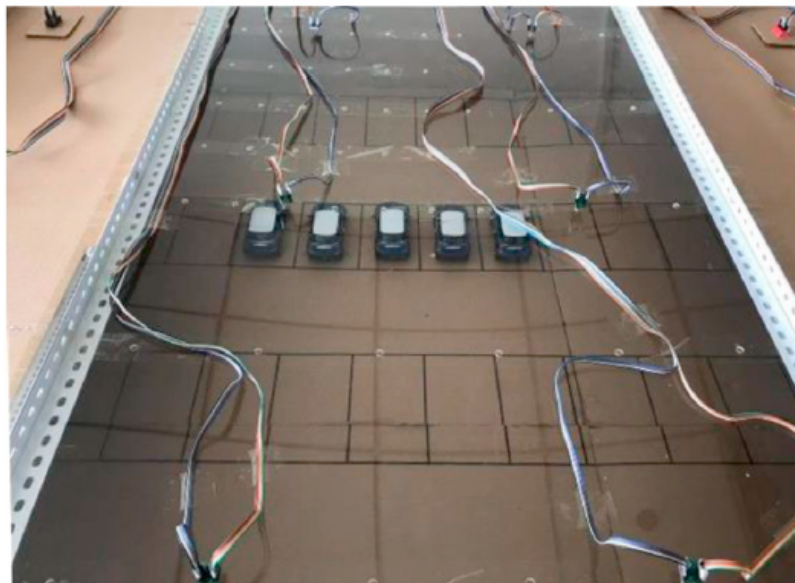
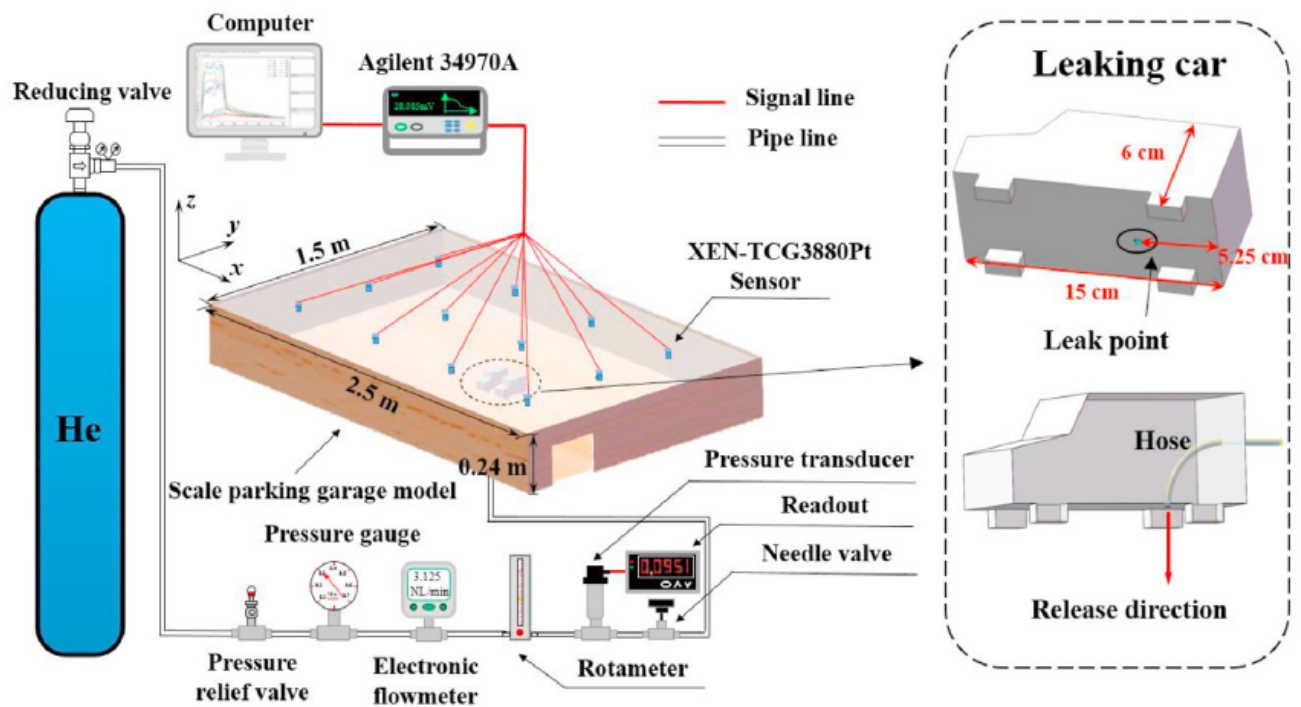


Abbildung 4: Literaturquelle 2 – Versuchsanlage

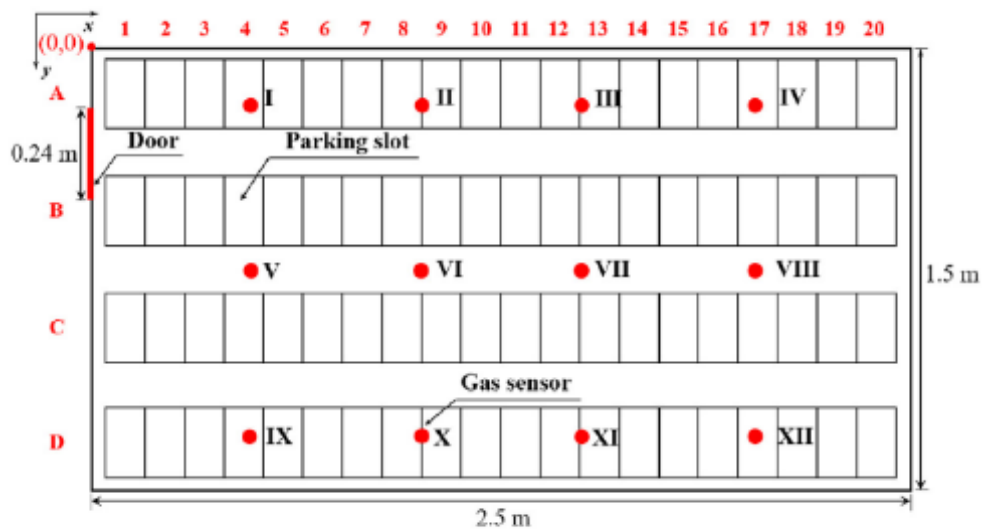


Abbildung 5: Literaturquelle 2 – Schema für die Bezeichnung der Parkplätze und Positionierung der Sensoren (Wärmeleitungssensoren)

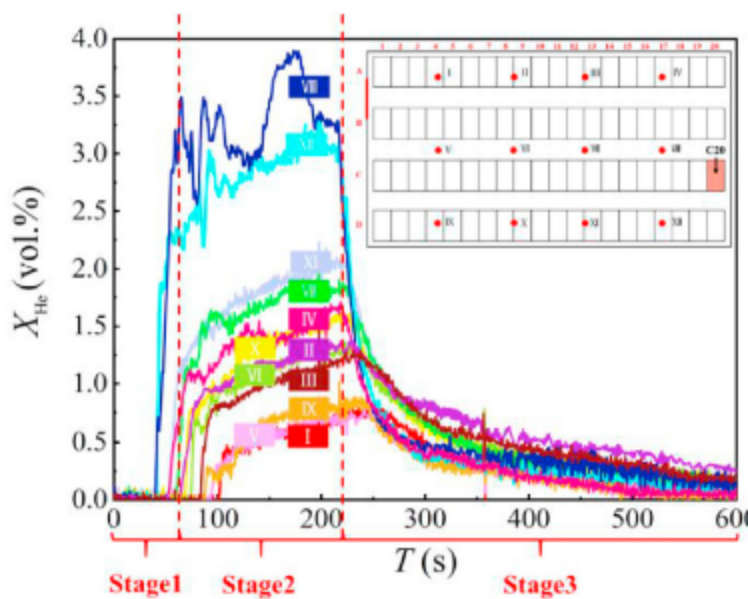
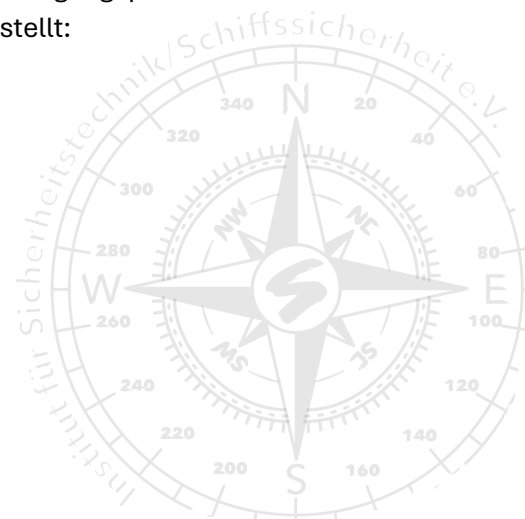


Abbildung 6: Literaturquelle 2 – Beispiel für gemessene zeitabhängige He-Verläufe bei einer He-Leckage bei Parkplatz C20

Die gemessenen He-Konzentrationen von 154 Versuchen werden als Eingangsparameter für ein KI-Modell verwendet. Es werden zwei unterschiedliche KI-Modelle erstellt:

- K-DTW (K-NN + DTW)
- artificial neural network ANN



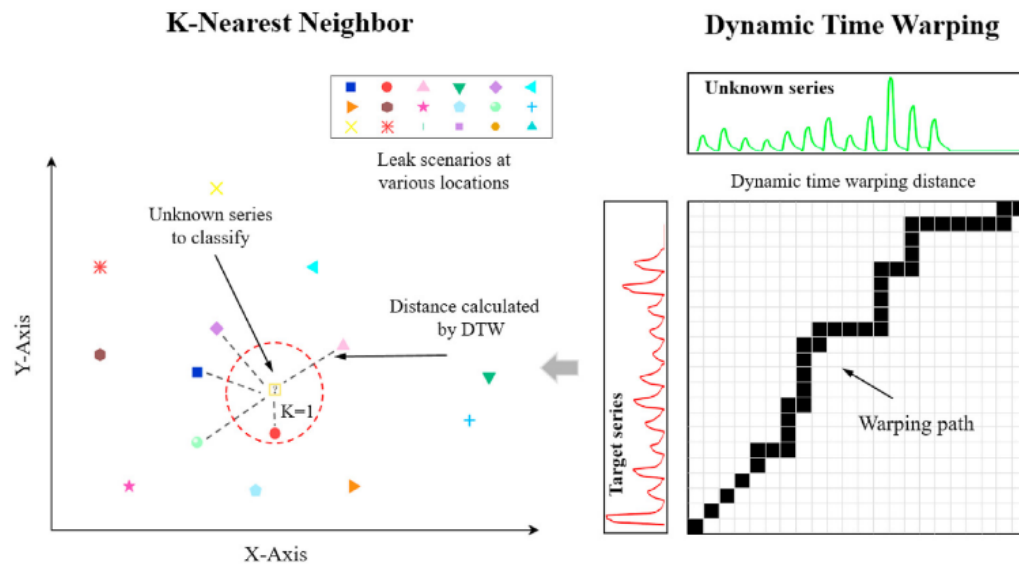


Abbildung 7: Literaturquelle 2 – Schema zur Erklärung des K-DTW Algorithmus

DTW: Dynamic Time Warping, misst die Ähnlichkeit oder den Abstand zwischen zwei zeitlichen Datenreihen oder Kurven, die unterschiedlich schnell ablaufen oder zeitlich verschoben sind. Es ist eine Mustererkennung in Zeitreihen möglich.

Die Vorgehensweise beim K-DTW Algorithmus ist wie folgt:

1. Berechnung der „minimalen DTW-Distanz“ zwischen den Eingabedaten und jeder Konzentrationsreihe unter Verwendung von Versuchsdaten mit bekannten Leckstellen
2. Das KNN-Verfahren vergleicht den minimalen DTW-Abstand der Zeitreihen mit den anderen bekannten Leckageszenarios. Auswahl eines am ehesten vergleichbaren Szenario und Zuordnung der Leckagestelle.

Zu dem verwendeten ANN siehe Abbildung 8 und Abbildung 9. Zu der in Abbildung 9 dargestellten Fehlermatrix folgende ergänzenden Anmerkungen:

- Abszisse (x-Achse) entspricht den tatsächlichen Leckageorten, die Ordinate (y-Achse) zeigt die vorhergesagten Leckageorte
- Die Einträge auf der Diagonalen geben die Anzahl der korrekt lokalisierten Testproben an
- Werte, welche nicht auf der Diagonalen liegen, sind fehlerbehaftet, d. h. entsprechen nicht exakt der korrekten Leckagestelle
- Die Test-Lokalisierungsgenauigkeit beträgt 78.4 %



In der Literaturquelle 2 ist nicht zu erkennen, ob der K-DTW Algorithmus mit dem erstellten ANN gekoppelt wird. Zu Abbildung 8 ist ergänzend zur Ermittlung der Modellunsicherheit mit einem „Monte-Carlo-Dropout-Verfahren (MC Dropout)“ folgendes anzumerken:

- Normalerweise wird Dropout nur während des Trainings genutzt, um eine Überanpassung zu verhindern. Beim MC Dropout lässt man den Dropout-Mechanismus jedoch auch während der Testphase aktiv. Dadurch wird die Modellunsicherheit mathematisch bestimmbar.
- Beim Dropout werden in jedem Durchgang zufällig (z. B. mit einer Wahrscheinlichkeit von 30 %) bestimmte Neuronen im Netzwerk abgeschaltet. Das bedeutet, dass sich bei jedem einzelnen Durchlauf die Architektur des Netzwerks minimal verändert.
- Anstatt einen Datensatz nur einmal durch das fertige Modell zu schicken, schickt man dieselbe Eingabe viele Male hintereinander (z. B. 100-mal) durch das Netzwerk, während Dropout aktiv bleibt. Da bei jedem der z. B. 100 Durchgänge andere Neuronen blockiert sind, verhält sich das Modell jedes Mal wie ein leicht anderes „Sub-Netzwerk“. Man verwendet dadurch quasi ein ganzes Ensemble aus verschiedenen KI-Modellen.
- Am Ende erhält man z. B. 100 verschiedene Vorhersagen für ein und dieselbe Eingabe. Aus der Verteilung dieser Antworten lässt sich die Unsicherheit mit statistischen Methoden bestimmen (Varianz, Standardabweichung, Entropie etc.).

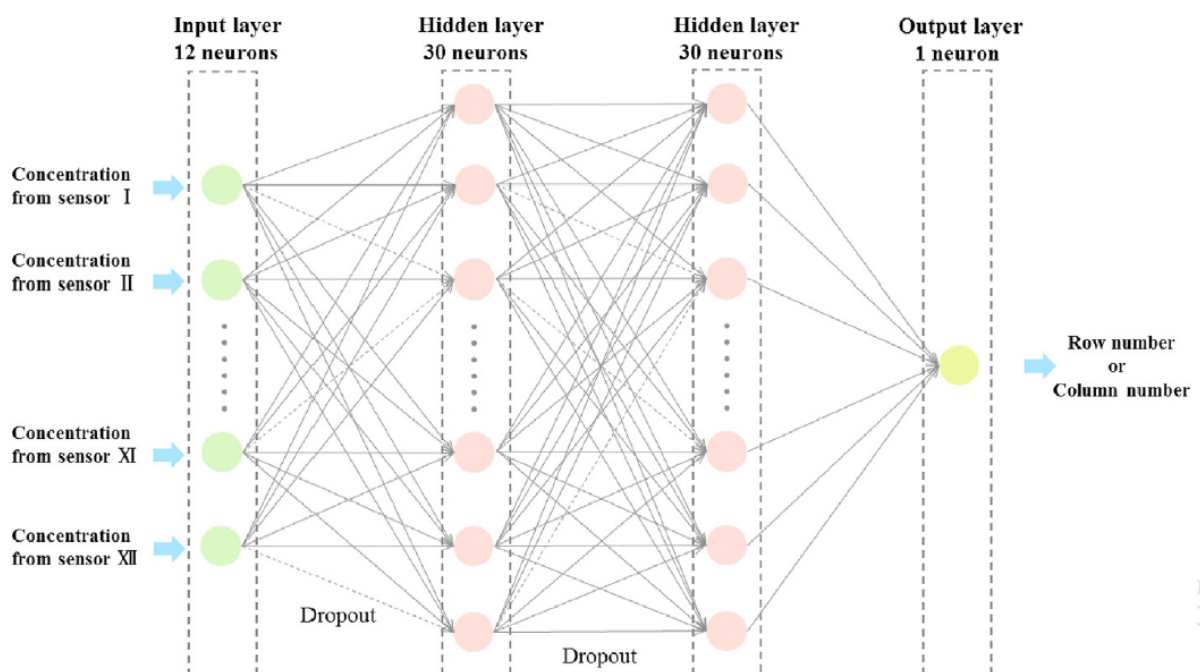


Abbildung 8: Literaturquelle 2 – Struktur des ANN, Anmerkung: Die Dropout Rate (ein Wert zwischen 0 und 1) bestimmt, welcher Prozentsatz an Neuronen während eines Durchgangs zufällig deaktiviert wird, um die Modellunsicherheit zu ermitteln.

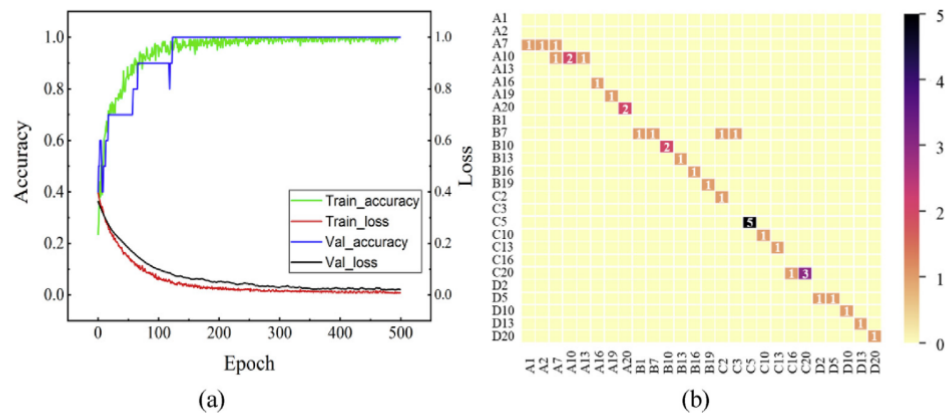


Abbildung 9: Literaturquelle 2 – ANN-Modelltrainingsprozess und Lokalisierungsgenauigkeit: Konvergenz (Loss) und Genauigkeit der Trainings- und Validierungsvarianten (a) Fehlermatrix der Leckage-Lokalisierung der Testvarianten (b)

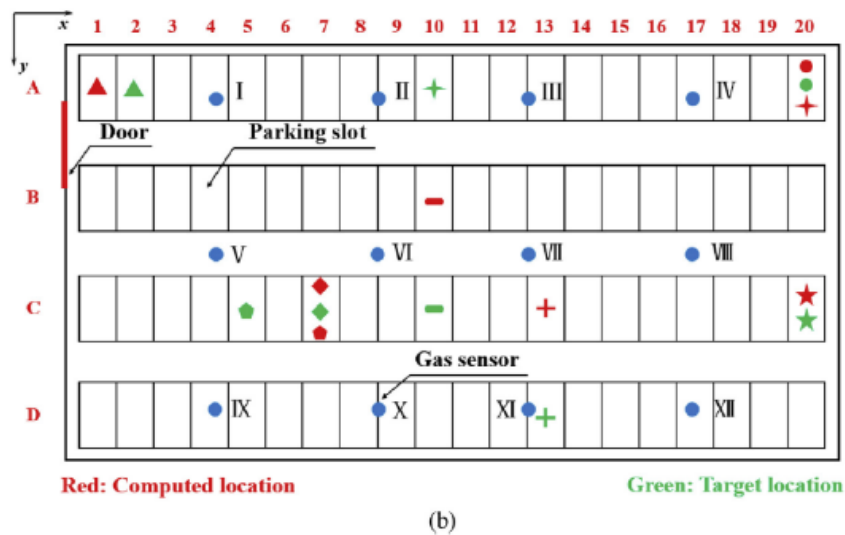
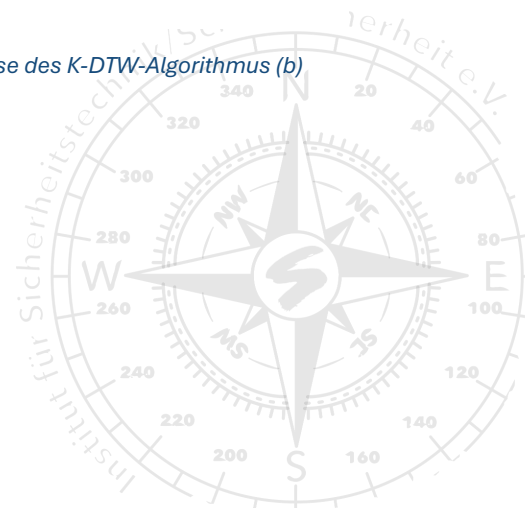


Abbildung 10: Literaturquelle 2 – Vorhersageprozess (a) und Lokalisierungsergebnisse des K-DTW-Algorithmus (b)

Tabelle 4: Literaturquelle 2 – Lokalisierungsergebnisse des K-DTW Algorithmus



Classifier	Testing sets								Accuracy
Target location	A1	C7	D13	C10	A20	A10	C20	C5	87.5%
Computed location	A2	C7	C13	B10	A20	A20	C20	C7	
Result ^a	C	C	C	C	C	IC	C	C	
Error ^b	1	0	1	1	0	10	0	2	

^a Correctly identified (C), incorrectly identified (IC).
^b Numbers represent the number of parking spaces from the target leak location.

In Abbildung 10 und Tabelle 4 sind der Vorhersageprozess und die Lokalisierungsergebnisse des K-DTW Algorithmus dargestellt. Die grünen Blöcke stellen Parkplätze mit bekannten experimentellen Daten dar, während die roten Blöcke auf fehlende Daten hinweisen. Das Leckszenario in A2 wurde verwendet, um die Lokalisierungsgenauigkeit zu testen, wobei die Daten für A2 nicht in den Trainingsdatensätzen enthalten waren. Der K-DTW-Algorithmus vergleicht dann die Testdaten (A2) und die Trainingsdaten (A4, B2, B4), um den Leckageort als A4, B2 oder B4 zu klassifizieren, die im Trainingsdatensatz enthalten waren. Dabei wurde das Klassifizierungsergebnis als korrekt akzeptiert, wenn der Fehler weniger als 2 Parkplätze in x-Richtung und 1 Parkplatz in y-Richtung betrug. Der K-DTW-Algorithmus identifizierte die Leckstellen mit einer Genauigkeit von 87,5 %.

Zu der in Literaturquelle 4 beschriebenen Vorgehensweise zur Lokalisierung der Leckagestelle (s. vorherigen Absatz) folgende Anmerkung: Es wird davon ausgegangen, dass das Sensormuster an A2 ähnlich ist wie bei seinen Nachbarn, was durchaus kritisch ist. Spannend wäre hier die Kopplung mit einem PINN (physical informed neural network) welches die an unbekannten Positionen zu erwartenden Sensorkonzentrationen aus den Daten der bekannten Positionen auf Basis physikalischer Modelle vorhersagt und dann erst die Entscheidung trifft.

5 Literaturquelle 3: CFD Simulation and ANN Prediction of Hydrogen Leakage and Diffusion Behavior in a Hydrogen Refuelling Station

Der Grundriss und das 3D Geometriemodell der H₂-Tankstelle sind in Abbildung 11 dargestellt.

Es wurden 240 Varianten mit einer H₂-Leckagemenge von 46.8 kg bis 136.8 kg H₂ in 30 s (1.5589 kg/s, 4.5589 kg/s) simuliert. Die relativ hohe Leckrate entsteht beim Ausströmen aus einem 450 bar H₂-Behälter.

Bei den Varianten wurde der Leckagemassenstrom, die Ausströmzeit und die Windgeschwindigkeit variiert. Diese Parameter sind die Eingangsparameter des ANN (artificial neural network). Die horizontale und vertikale Ausdehnung der H₂-Wolke sind die Ausgabewerte des ANN (s. Abbildung 12).

Es wurde ein BP ANN verwendet: Ein BP Artificial Neural Network (Backpropagation-Netzwerk) ist ein künstliches neuronales Netz, das mit dem Backpropagation-Algorithmus (Rückwärtsgenerierung des Fehlers) trainiert wird. Es ist ein mehrschichtiges Vorwärtsnetzwerk (Multi-Layer Feedforward Network), das Fehler von der Ausgabe zur Eingabe zurückleitet, um die Gewichtung der Signale anzupassen.

Die Simulationsdaten der 240 Varianten werden zufällig 6:2:2 aufgeteilt, d. h. 144 Varianten zum Training, 48 Varianten zur Validierung und 48 Varianten zum Test. Die Aufgaben dieser 3 Anteile können wie folgt beschrieben werden:

Datensatz	Anteil (Typisch)	Analogie aus der Schule	Aufgabe im Detail
1. Training Set	60 % – 80 %	Das Schulbuch und die Übungsaufgaben	Mit diesen Daten lernt das Modell. Es passt seine internen Gewichte (oder die Äste der Entscheidungsbäume) an, um die Muster in den Daten zu verstehen.
2. Validation Set	10 % – 20 %	Die Zwischenprüfungen während des Schuljahres	Diese Daten werden genutzt, um die Hyperparameter zu tunen (z. B. um die optimale Lernrate, die Anzahl der Bäume bei XGBoost oder das beste k bei KNN zu finden). Das Modell lernt nicht direkt daraus, aber Sie passen das Modell basierend auf diesen Ergebnissen an.
3. Test Set	10 % – 20 %	Die finale Abschlussprüfung	Diese Daten bleiben bis zum Schluss streng unter Verschluss (Hold-out Data). Erst wenn das Modell komplett fertig entwickelt ist, prüft man damit die finale Genauigkeit. So sieht man, wie gut das Modell auf <i>völlig unbekannte</i> Praxisdaten reagiert.

Zu Abbildung 13 folgende Anmerkung: Die Zielwerte (tatsächliche Werte) sind nur so genau wie von der Simulation vorhergesagt. Die „reale Welt“ ist so real wie die Simulationsdaten diese wiedergeben. Zwischen Simulation und der „realen Welt“ sind im allgemeinen immer Abweichungen möglich, welche durch unterschiedliche Randbedingungen der Simulation zur „realen Welt“ entstehen können.

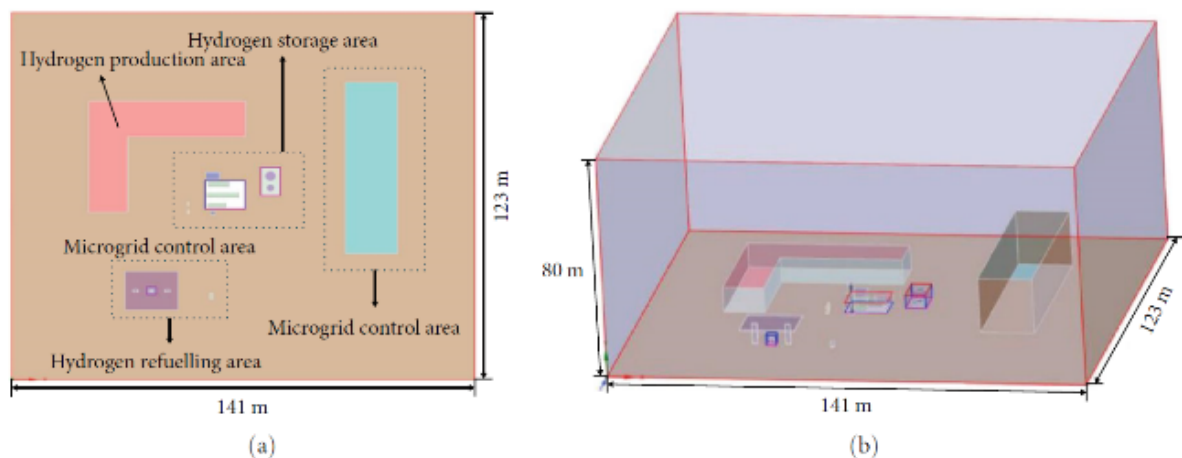
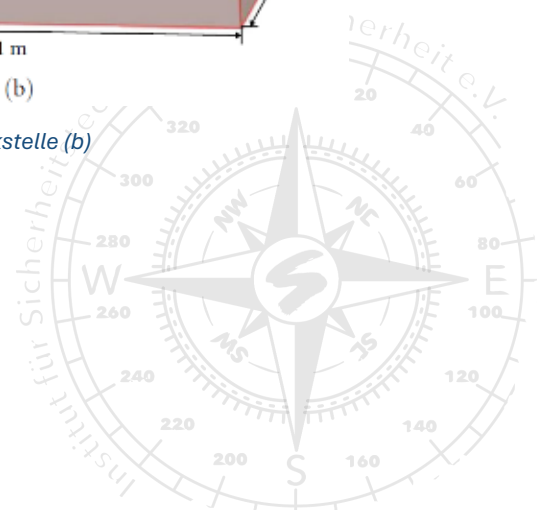
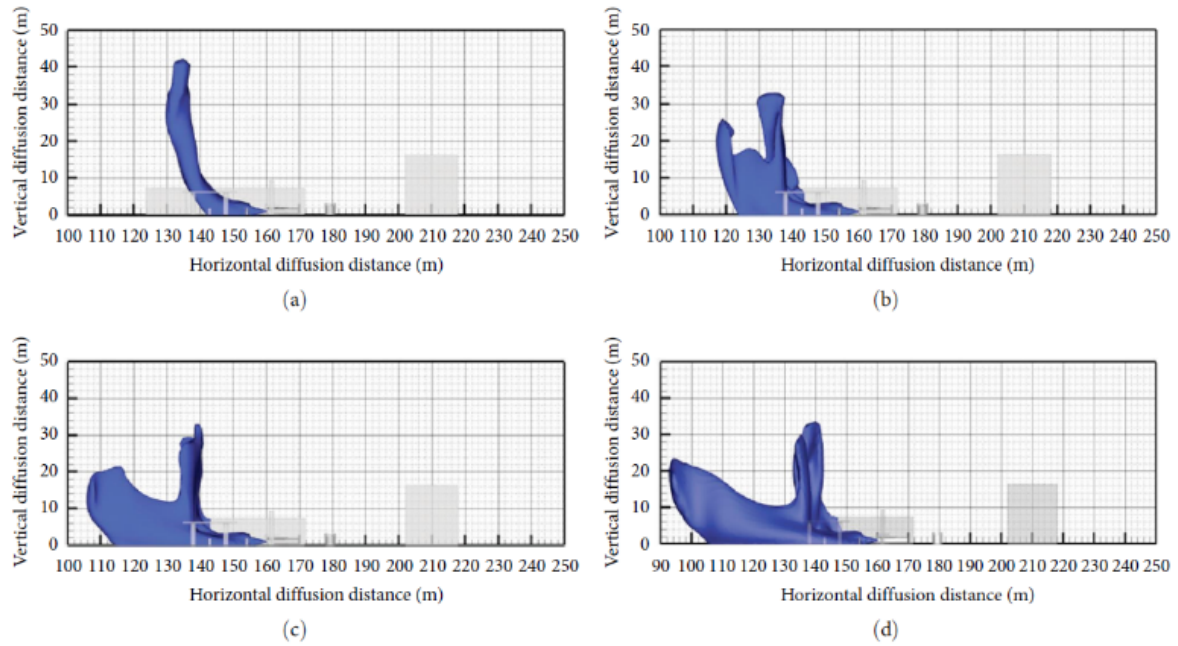


Abbildung 11: Literaturquelle 3 – Grundriss (a) und 3D Geometriemodell der H₂-Tankstelle (b)





Mass flow rate (kg/s)	Horizontal diffusion distance (m)	Vertical diffusion distance (m)	Volume of flammable hydrogen cloud (m ³)
1.5589	30.417	42.41	2,277.48
2.5589	43.092	33.13	3,664.88
3.5589	54.283	33.08	4,876.71
4.5589	67.2	34	6,140.46

Abbildung 12: Literaturquelle 3 – horizontale und vertikale Ausdehnung der H₂-Wolke nach 30s für unterschiedliche H₂-Leckage-Massenströme: 1.5589 kg/s (a), 2.5589 kg/s (b), 3.5589 kg/s (c) und 4.5589 kg/s (d)



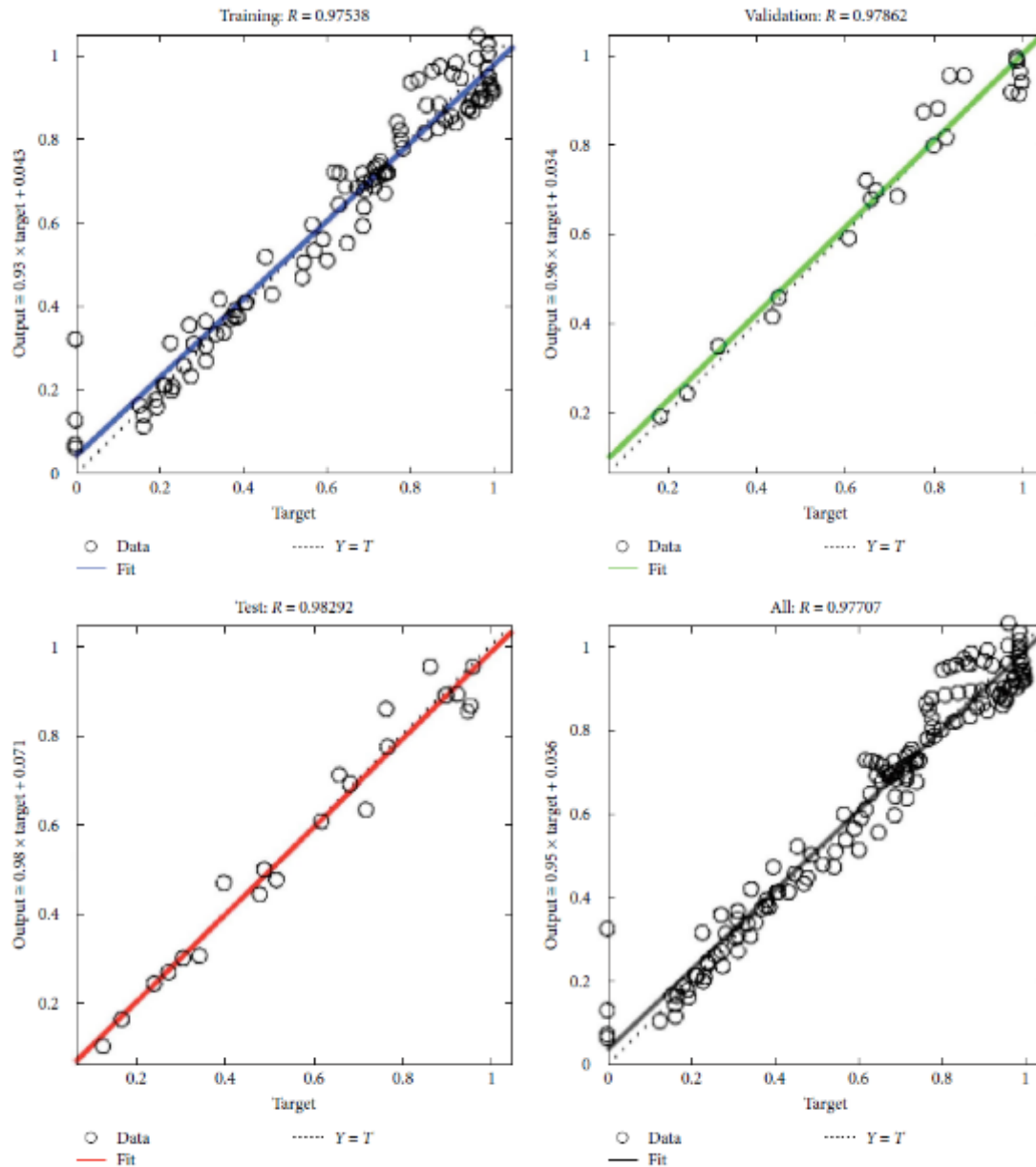


Abbildung 13: Literaturquelle 3 – Korrelationskoeffizienten zwischen dem Zielwert (tatsächlicher Wert, Target) und dem berechneten Wert (Output)



6 Literaturquelle 4: Data-augmented deep learning for hazard assessment of hydrogen accumulation in confined spaces: Multitask prediction and sensitivity analysis

Der betrachtete Raum (30 m³ bis 200 m³) mit 2 Lüftungsöffnungen, sowie ein Schema mit Parametern zu den durchgeführten Untersuchungen sind in Abbildung 14 dargestellt.

Zur Simulation wurde die Software HyRAM+ verwendet. HyRAM+ („Hydrogen Risk Assessment Models Plus“) ist ein robustes Werkzeug für die Wasserstoffsicherheitsanalyse und Risikobewertung, um vielfältige experimentelle Szenarien zu simulieren und umfangreiche Datensätze zu generieren (s. <https://h2tools.org/hyram>).

Die statischen und dynamischen Einstellungen (Randbedingungen) für die mit HyRAM+ durchgeführten Simulationen sind in Tabelle 5 enthalten.

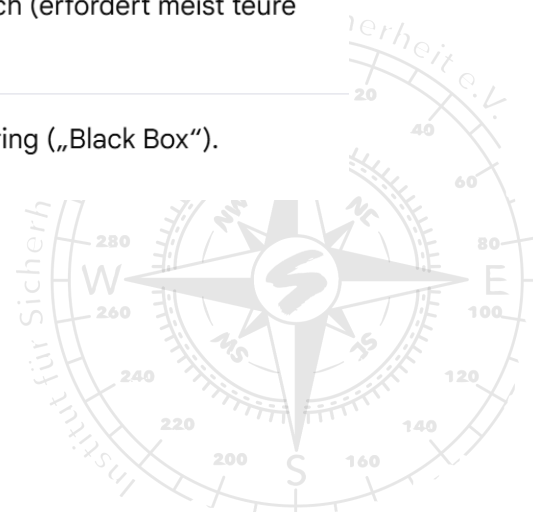
Schema für die Modellierung der Wasserstoffanreicherung und das Multitask-Learning ist in Abbildung 15 dargestellt.

Nach dem Training und der Validierung des Modells wurde eine Sensitivitätsanalyse durchgeführt, um den Einfluss der wichtigsten Parameter auf die Vorhersage des Wasserstoffrisikos zu bewerten.

In Abbildung 15 erfolgt eine Unterscheidung in DL-based model (deep learning) und Baseline models. DL-based model entspricht einem künstlichen neuronalen Netzwerk (ANN). Bei den Baseline models werden mathematische Methoden wie KNN und XGB (s. Literaturquelle 2), RF (Random Forest), GB (Gradient Boosting) und SVM (Support Vector Machine) verwendet. Diese Basismodelle können dazu verwendet werden, um erste Ausgabewerte sowie Vergleichswerte für das ANN des KI-Modells zu erhalten.

Die erwähnten Baseline-Modelle werden als Maßstab verwendet, den ein neues, oft komplexeres Modell (wie ein tiefes neuronales Netzwerk) erst einmal schlagen muss. XGBoost simuliert keine Neuronen und „denkt“ nicht wie ein Large Language Model, weshalb es von Laien manchmal abgewertet wird. In der Datenwissenschaft ist es jedoch eines der am häufigsten genutzten und angesehensten Werkzeuge. Nachfolgend ein Vergleich der Eigenschaften von XGBoost und ANN:

Eigenschaft	XGBoost (Klassisches ML)	Neuronale Netze (Deep Learning)
Datentyp	Perfekt für tabellarische Daten (Zahlen, Kategorien).	Perfekt für unstrukturierte Daten (Bilder, Audio, Text).
Rechenaufwand	Gering (läuft super auf einer normalen CPU).	Sehr hoch (erfordert meist teure GPUs).
Erklärbarkeit	Relativ hoch (Feature Importance zeigt, was wichtig war).	Sehr gering („Black Box“).



Die Hyperparameteroptimierung erfolgte mit der Software Optuna, einem Framework, das effiziente Stichproben- und Suchstrategien nutzt, um die optimale Modellkonfiguration zu ermitteln. Das vorgeschlagene DL-basierte Modell wurde eingehend mit Basismodellen verglichen und zeigte dabei eine gute Leistung bei der genauen Klassifizierung von Risikostufen und der Vorhersage wichtiger Kennzahlen zur Wasserstoffansammlung.

Bei Literaturquelle 4 sind in Bezug auf das KI-Modell einige Besonderheiten zu erkennen, welche bei den anderen Literaturquellen nicht vorkommen:

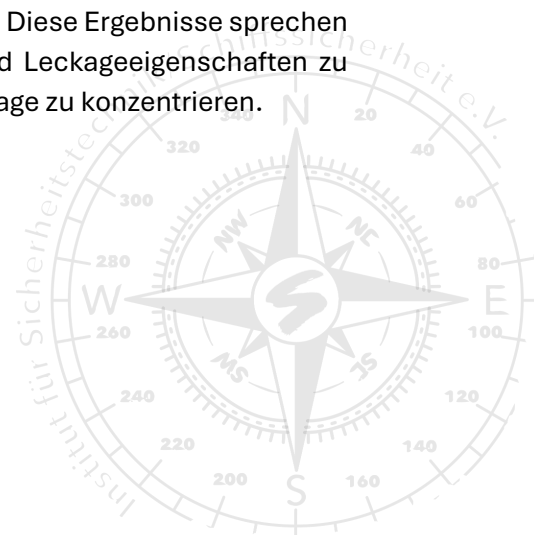
- Zur Erhöhung der Datenmenge (data augmentation) wurde ein VAE (Variational Autoencoder) eingesetzt. Das bedeutet, dass die Anzahl der Varianten künstlich vergrößert wurde (auf 61888 Varianten). VAE ist ein KI-Werkzeug, welches versucht, aus den echten Datensätzen Grundregeln zu erlernen. Danach erfindet VAE neue, ähnliche Daten, die es vorher so noch nicht gab. Hierbei werden die Eingangsparameter gezielt verändert (ansonsten würde der VAE im Grunde nur die bereits bekannten Trainingsdaten rekonstruieren).
- Es wurde eine 5-fache Kreuzvalidierung durchgeführt, d.h. 5 x 20 % als Testdatensatz und die jeweils restlichen 80% sind die Trainingsdaten. Eine solche Kreuzvalidierung ist nicht üblich, Trainings- und Testdaten sollten nicht vermischt werden.

Zur Bewertung des Einflusses der Eingangsparameter (s. Tabelle 5) auf die Ausgabewerte (s. Abbildung 16) wurden folgende Analysemethoden angewendet:

- MI_Analyse (Mutual Information): Bestimmt die Stärke der Abhängigkeit zwischen Eingangsparametern und Ausgabewerten. Diese Analyse hilft dabei, die Merkmale zu identifizieren, die die Modellvorhersagen am stärksten beeinflussen.
- SHAP-Analyse (SHapley Additive exPlanations) ist eine Methode aus dem Bereich der künstlichen Intelligenz, um die Entscheidungen von KI-Modellen verständlich und nachvollziehbar zu machen. Die Methode nutzt sogenannte *Shapley-Werte* aus der Spieltheorie. Dabei wird jedes Merkmal (Feature) des Datensatzes als ein „Spieler“ betrachtet, der gemeinsam mit anderen zu einem Ergebnis (der Vorhersage) kommt. SHAP berechnet für jedes Merkmal einen konkreten Wert, der angibt, wie stark dieses Merkmal die Vorhersage im Vergleich zum Durchschnittswert erhöht oder verringert.

Schlussfolgerungen:

Im Gegensatz zu herkömmlichen Hypothesen, die die Bedeutung der Lüftungsdurchflussrate betonen, identifizierten die durchgeführten Merkmalsempfindlichkeitsanalysen (s. vorheriger Absatz) durchweg den Leckdurchmesser, die Gebäudefläche und den Betriebsdruck als die wichtigsten Einflussfaktoren für die Dynamik des Wasserstoffrisikos. Diese Ergebnisse sprechen dafür, den Schwerpunkt stärker auf räumliche Konfigurationen und Leckageeigenschaften zu legen, anstatt sich ausschließlich auf die Auslegung der Lüftungsanlage zu konzentrieren.



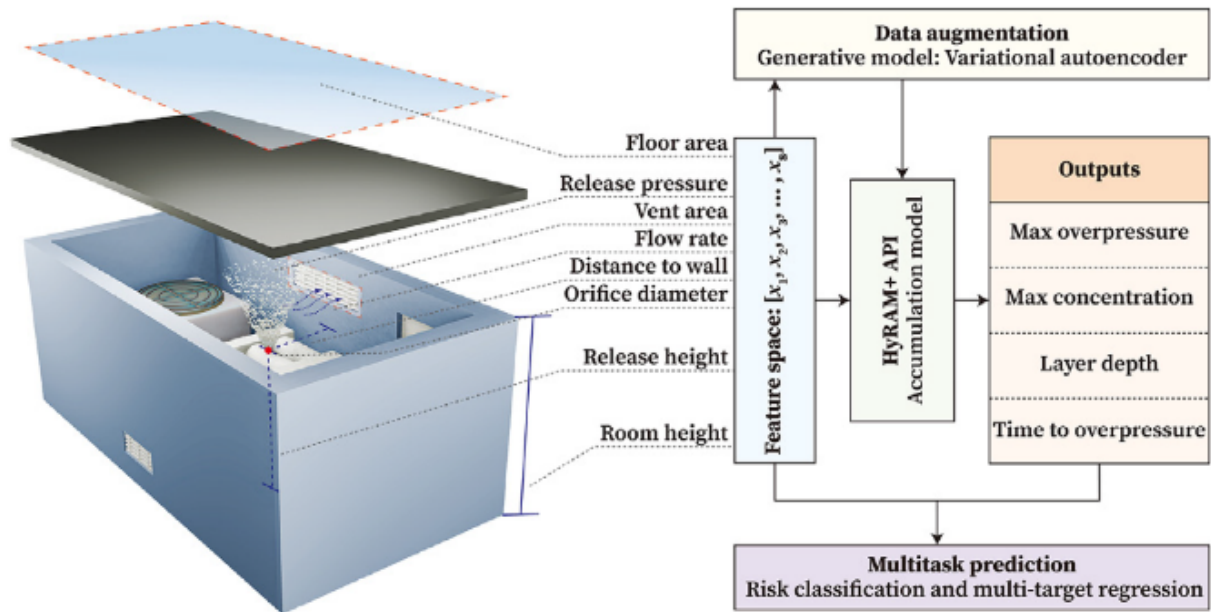


Abbildung 14: Literaturquelle 4 – Geometrie und Schema mit Parametern zu den durchgeführten Untersuchungen

Tabelle 5: Literaturquelle 4 – statische und dynamische Einstellungen (Randbedingungen) für die mit HyRAM+ durchgeführten Simulationen

Dynamic settings for hydrogen accumulation simulation with HyRAM+.

Parameter	Description	Value	Unit
OD	Nozzle diameter	0.068, 0.215, 0.680, 2.150, 6.800	mm
ReH	Height of hydrogen release point	0.0, 1.0, 2.0	m
ReP	Release pressure	25, 35, 45	MPa
RH	Room height	3.0, 3.5, 4.0	m
FA	Floor area of the room	10, 30, 50	m ²
DW	Distance from wall	0.5, 1.0, 2.0, 2.5	m
VA	Ventilation area	0.05, 0.10, 0.15, 0.20	m ²
FR	Airflow rate through the vent	VA 0.05 m ² : 0.0 to 0.1 (step: 0.025)	m ³ /s
		VA 0.10 m ² : 0.0 to 0.2 (step: 0.050)	
		VA 0.15 m ² : 0.0 to 0.3 (step: 0.075)	
		VA 0.20 m ² : 0.0 to 0.4 (step: 0.100)	

Static settings for hydrogen accumulation simulation with HyRAM+.

Parameter	Description	Value	Unit
Tank volume	Total volume of the hydrogen tank	0.00363	m ³
Floor vent area	Area of the floor vent opening	0.00762	m ²
Floor vent height	Height of the floor vent from the base	0.044	m
Ceiling vent height	Height of the ceiling vent from the base	$RmH - 0.3$	m
Nozzle type	Type of nozzle used for hydrogen release	Yuceil/Otugen	–
Release angle	Angle of hydrogen release relative to horizontal	0.0	°
Simulation time	Total time duration for the simulation	30.0	s



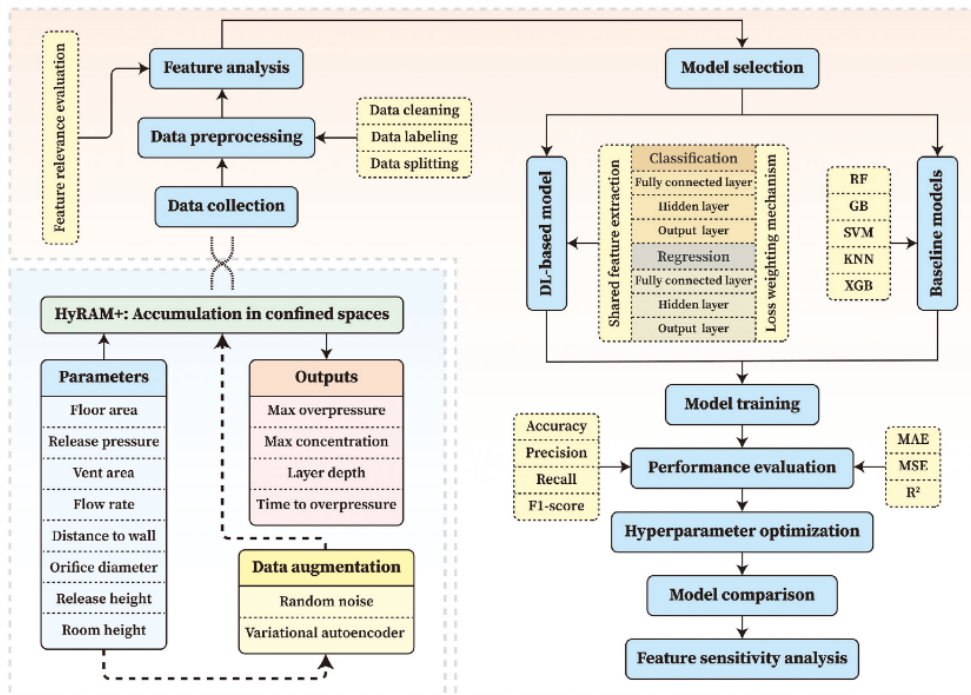


Abbildung 15: Literaturquelle 4 – Schema für die Modellierung der Wasserstoffanreicherung und das Multitask-Learning

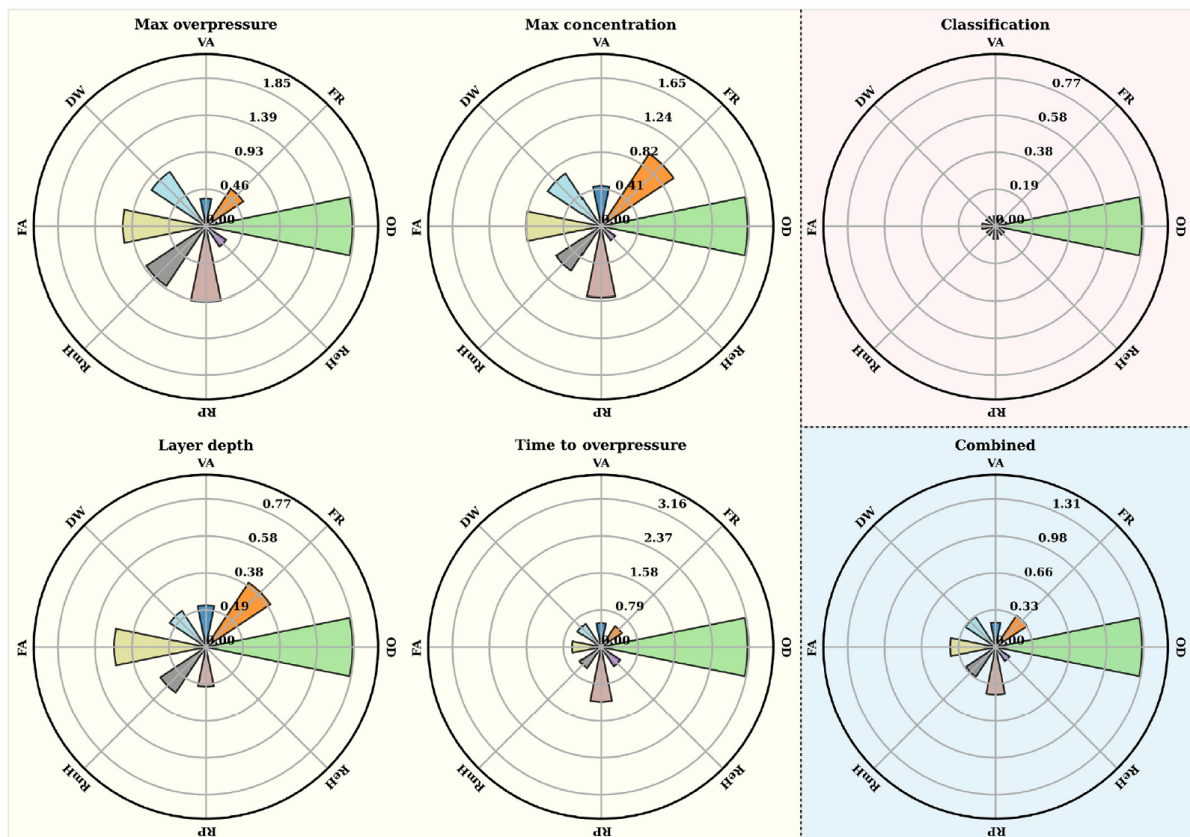


Abbildung 16: Literaturquelle 4 – Bewertung des Einflusses der Eingangsparameter (s. Tabelle 5) auf die Ausgabewerte mit Hilfe von MI-Analysen

7 Literaturquelle 5: Real-time hydrogen release and dispersion modelling of hydrogen refuelling station by using deep learning probability approach

Es wird die Leckage und Ausbreitung von H₂ an einer Tankstelle mit der Software FLACS simuliert:

- 70 Varianten (49 für Training, 21 für Test)
- Einspritzzeit / Versuchszeit: 70 s
- Massenstrom: 0.46 bis 1.46 kg/s
- Wind: 1 m/s bis 7 m/s, in und entgegengesetzt zur Ausbreitungsrichtung
- ANN: 5 Schichten, 1. und 2. Schicht haben 8 Neuronen, 3. Schicht 16, 4. Schicht 32, 5. Schicht 64 Neuronen

Die Architektur des entwickelten KI-Modells ist in Abbildung 17 dargestellt. Das verwendete Monte Carlo dropout Verfahren hat folgende Merkmale:

- Die Unsicherheit der mit dem ANN vorhergesagten H₂-Verteilung wird berechnet
- Die Dropout Rate (ein Wert zwischen 0 und 1) bestimmt, welcher Prozentsatz an Neuronen während eines Durchgangs zufällig deaktiviert wird, um die Modellunsicherheit zu ermitteln.
- Normalerweise schaltet man Dropout nach dem Training ab. Beim MC-Dropout bleibt es aktiv, wenn eine neue Modellabfrage erfolgt.

Ein Beispiel für eine mit dem entwickelten KI-Modell vorhergesagte H₂-Verteilung mit Angaben zur Unsicherheit s. Abbildung 18

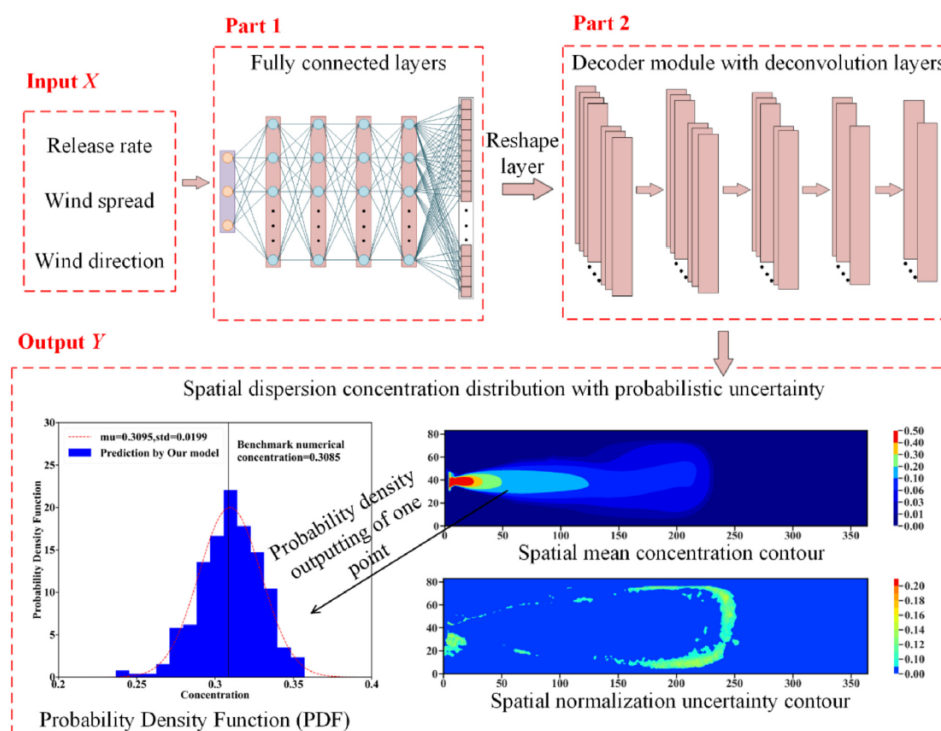


Abbildung 17: Literaturquelle 5 – Architektur des entwickelten KI-Modells (DPL_H₂ Plume model)

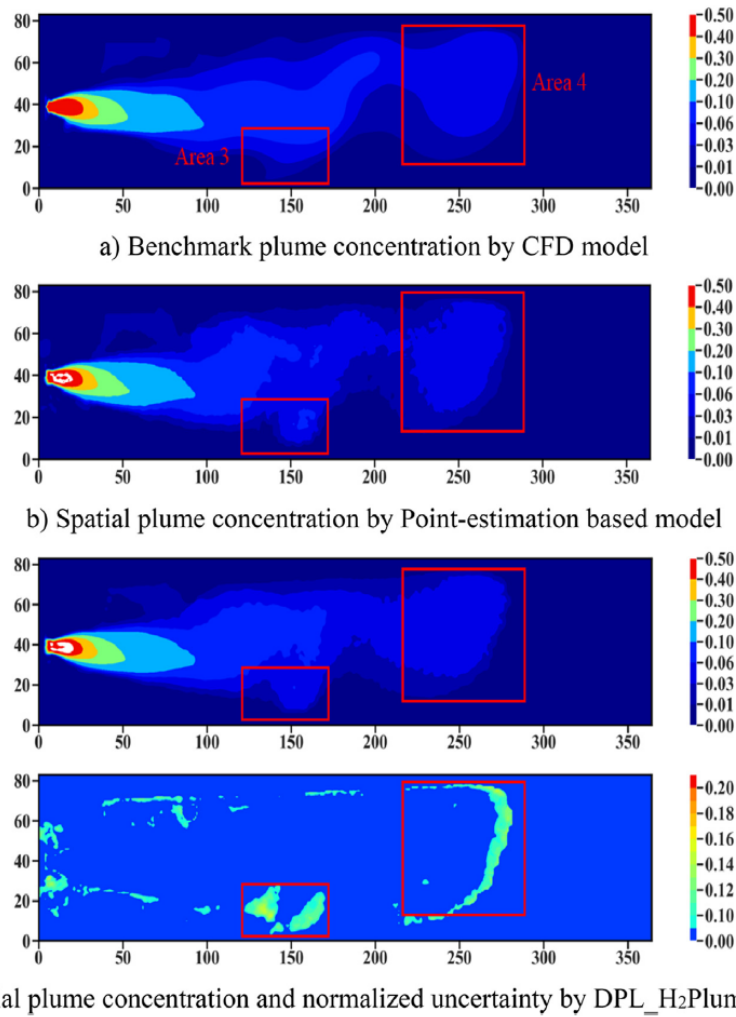


Abbildung 18: Literaturquelle 5 – Vergleich der räumlichen Verteilung der H₂-Wolke: CFD (a), Point-estimation based model (b) und DPL_H₂ Plume model, H₂-Leckagerate 1.46 kg, Windgeschwindigkeit 3 m/s in Einspritzrichtung, beim DPL_H₂ Plume model. Angabe der Unsicherheit (in Bezug auf die Vorhersage der CFD-Simulationsergebnisse), welche hauptsächlich im Randbereich der H₂-Wolke entsteht.



8 Literaturquelle 6: Hybrid neural network-based surrogate model for fast prediction of hydrogen leak consequences in hydrogen refueling station

Es wird die Leckage und Ausbreitung von H₂ an einer Tankstelle mit der Software Fluent simuliert:

- H₂-Quelle: 0.60 kg/s für den Betankungsschlauch und 2,15 kg/s für die Betankungsdüse,
- Einspritzzeit 30 s

Die Geometrie und das CFD-Modell der H₂-Tankstelle mit Darstellung der Leckagestellen s. Abbildung 19. Bei den Experimenten wurde anstelle H₂ Helium He verwendet. Der skalierte Versuchsaufbau ist in Abbildung 20 dargestellt.

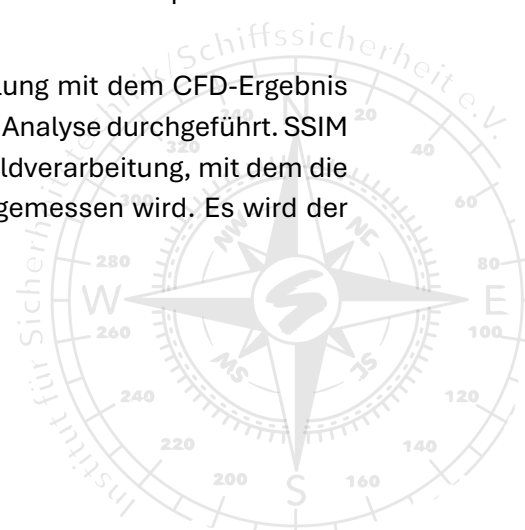
Ein Schema zur Entwicklung des KI-Modells mit den angewendeten mathematischen Methoden s. Abbildung 21. Die Methoden GAN, LSTM und DNN werden gekoppelt, s. Abbildung 22. Hierzu eine Kurzbeschreibung der Funktionen der einzelnen Module:

- GAN (Generative Adversarial Network): Konzept aus dem Maschinellen Lernen, bei dem zwei neuronale Netze (Generator und Diskriminator) im Wettbewerb zueinander stehen, um täuschend echte Daten zu erzeugen. In diesem hybriden Modell besteht die Hauptaufgabe des GAN darin, einen Generator bereitzustellen, der eine Wasserstoff-Konturkarte erzeugen kann. Während des Trainingsprozesses wird der Generator des GAN mit einem zufälligen Rauschen gefüttert, und auf der Grundlage dieser Daten wird ein falsches Bild erzeugt, das der Diskriminator bewerten soll. Der Diskriminator wird mit den von CFD generierten Trainingsbeispielen und den von Generatoren erzeugten gefälschten Bildern gefüttert, um zu bestimmen, welche Daten echt und welche gefälscht sind.
- LSTM (Long-Short-Term-Memory-Netzwerk): Ist in der Lage, sowohl periodische als auch zufällige Verhaltensmuster von Zeitreihendaten zu erfassen. Im Hybridmodell besteht die Hauptaufgabe des LSTM darin, die zeitliche Charakterisierung der Veränderung der Wasserstoffwolkenkonzentration zu erlernen und Vorhersagen zu treffen.
- DNN (Deep Neural Network): Die Hauptaufgabe des DNN im Hybridmodell besteht darin, die physikalischen Randbedingungen auf Dimensionen abzubilden, die dem Eingangsrauschen des GAN-Generators entsprechen.

Das KI-Modell kombiniert die generative Fähigkeit von GAN mit der zeitlichen Vorhersagefähigkeit von LSTM. Die Eingangsparameter (Leckageort, Leckagerichtung, Leckagerate) werden in Form von Labels über ein DNN hinzugefügt.

Ein Vergleich von vorhergesagten H₂-Konzentrationen mit CFD-Ergebnissen und experimentellen Ergebnissen (Referenzwerte) s. Abbildung 23 und Abbildung 24.

Zur Bewertung der Übereinstimmung der vorhergesagten H₂-Verteilung mit dem CFD-Ergebnis (s. Abbildung 23) wird eine SSIM (Structural Similarity Index Measure) Analyse durchgeführt. SSIM ist ein weit verbreitetes, hochentwickeltes Metrik-Verfahren in der Bildverarbeitung, mit dem die optische Qualität eines Bildes im Vergleich zu einem Referenzbild gemessen wird. Es wird der Ähnlichkeitsgrad zwischen 2 Bildern bestimmt.



Die Structural Similarity Index Measure (SSIM) ist ein wahrnehmungsbasiertes Modell, das die Ähnlichkeit zwischen zwei Bildern (einem Originalbild x und einem Testbild y) berechnet. Im Gegensatz zu rein mathematischen Pixel-Fehlermessungen wie dem MSE (Mean Squared Error), orientiert sich SSIM an der menschlichen visuellen Wahrnehmung, die besonders sensibel für strukturelle Veränderungen und Muster im Raum ist.

SSIM zerlegt das Problem der Bildbewertung in drei Kernkomponenten:

1. Luminanz (Helligkeit): Misst die Annäherung der mittleren Helligkeit beider Bilder über den Mittelwert
2. Kontrast: Vergleicht die Intensitätsunterschiede und Musterstrukturen anhand der Standardabweichung
3. Struktur: Bewertet die Ausrichtung lokaler Pixelmuster. Dadurch wird geprüft, ob Kanten und Formen in beiden Bildern identisch verlaufen.

Der Ablauf einer SSIM-Analyse:

1. Sliding-Window-Verfahren: Die Analyse wird nicht auf das gesamte Bild auf einmal angewendet. Stattdessen wandert ein kleines Analysefenster (meist eine 8×8 oder 11×11 Pixel große Gauß-Matrix) Schritt für Schritt über beide Bilder.
2. Lokale Berechnung: Für jedes dieser Fenster werden die Helligkeit, der Kontrast und die Struktur berechnet und miteinander verglichen.
3. Erstellung der Quality Map: Aus den lokalen Werten entsteht eine Zwischenkarte (Quality Map), die genau zeigt, an welchen Stellen des Bildes die stärksten Abweichungen existieren.
4. Mittelwertbildung (MSSIM): Der finale, globale SSIM-Wert für das gesamte Bild ergibt sich aus dem Durchschnitt (Mean) aller lokalen Fensterwerte.

Ergebniswerte und Interpretation:

- Der Wert liegt in der Regel im Bereich von 0 bis 1 (theoretisch -1 bis 1, wobei negative Werte eine völlige Antikorrelation beschreiben).
- Ein Wert von 1.0 bedeutet, dass die Bilder visuell absolut identisch sind.
- Ein Wert nahe 0 signalisiert, dass keinerlei strukturelle Ähnlichkeit vorhanden ist.



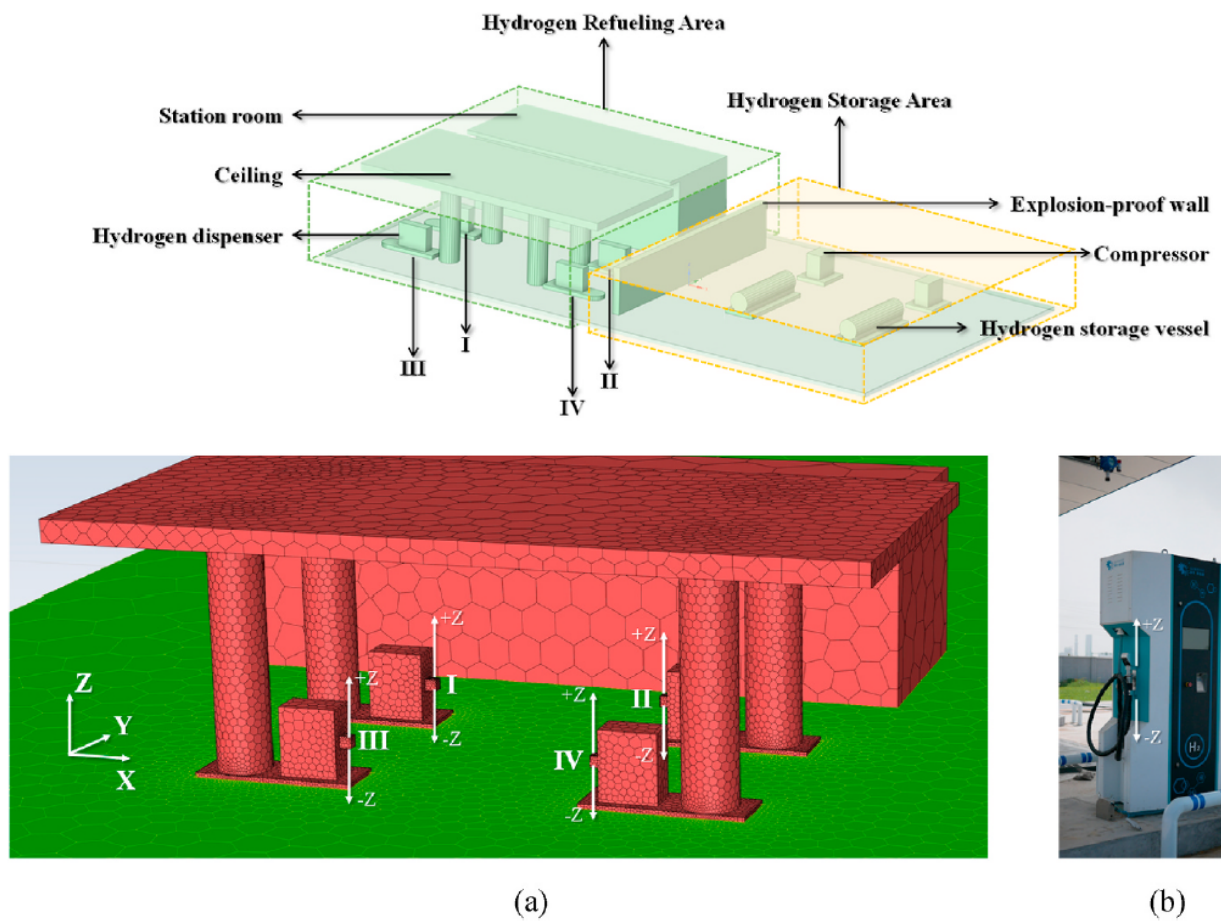
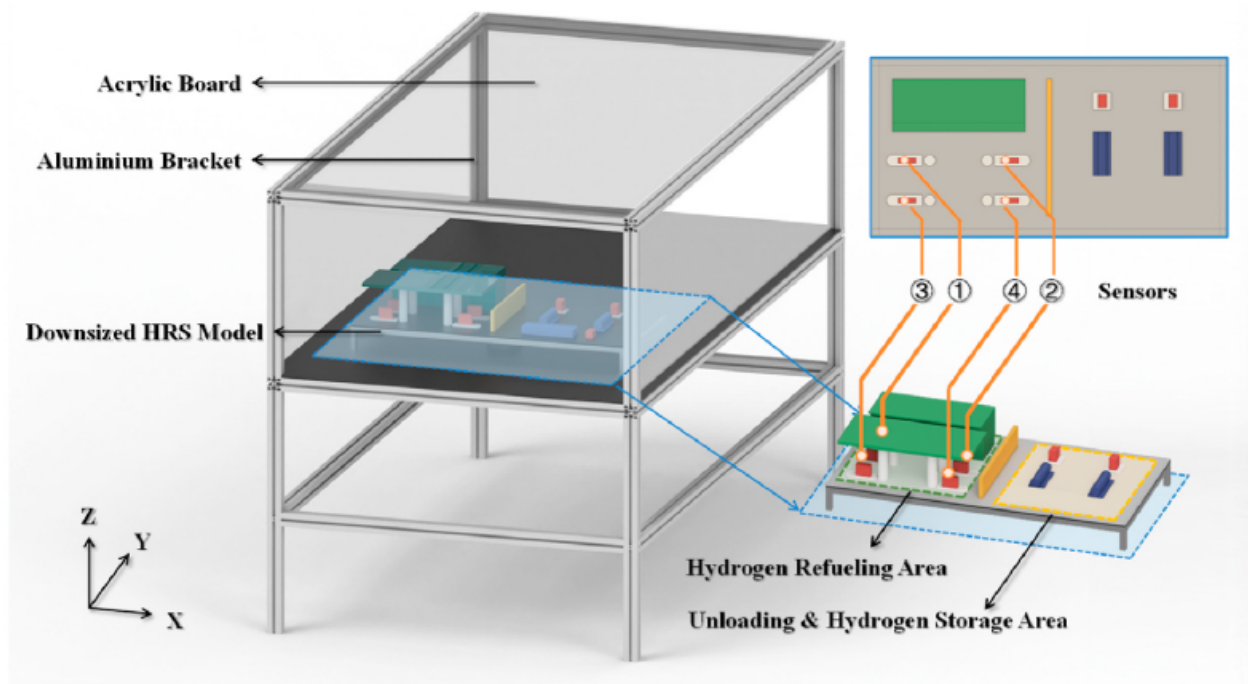
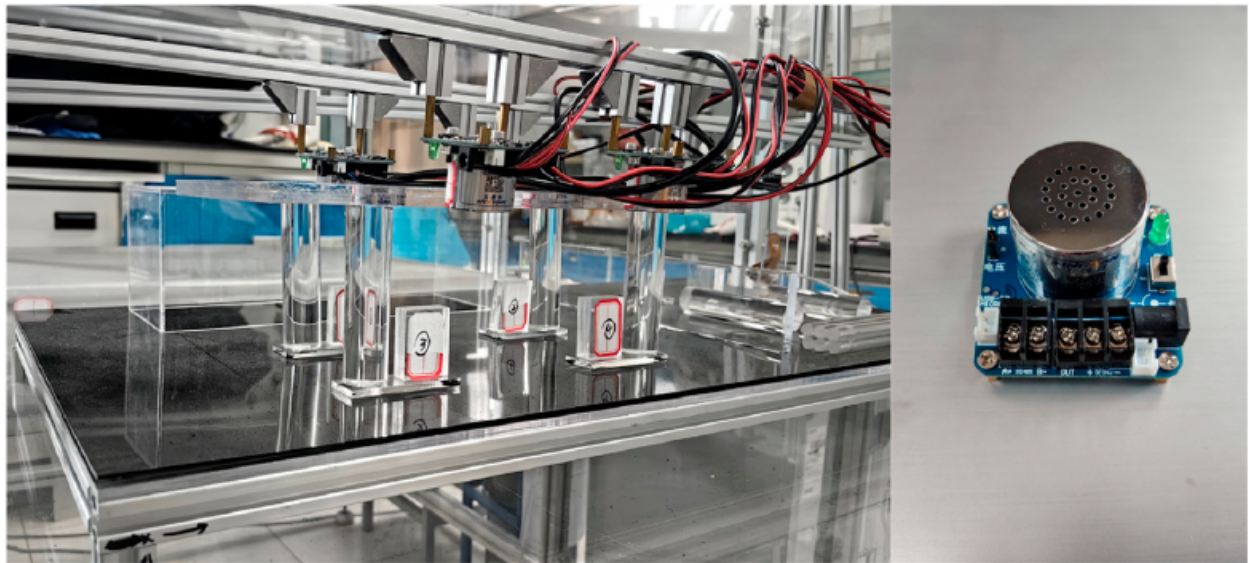


Abbildung 19: Literaturquelle 6 –Geometrie (oberes Bild) und CFD-Modell der H₂-Tankstelle mit Darstellung der Leckagestellen (untere Bilder)





(a) The rendering of experimental apparatus and sensors location



(b) The figure of experimental setup and He sensor used in these experiments

Abbildung 20: Literaturquelle 6 –skalierter Versuchsaufbau, He-Sensoren (Wärmeleitungssensoren, unten rechts)



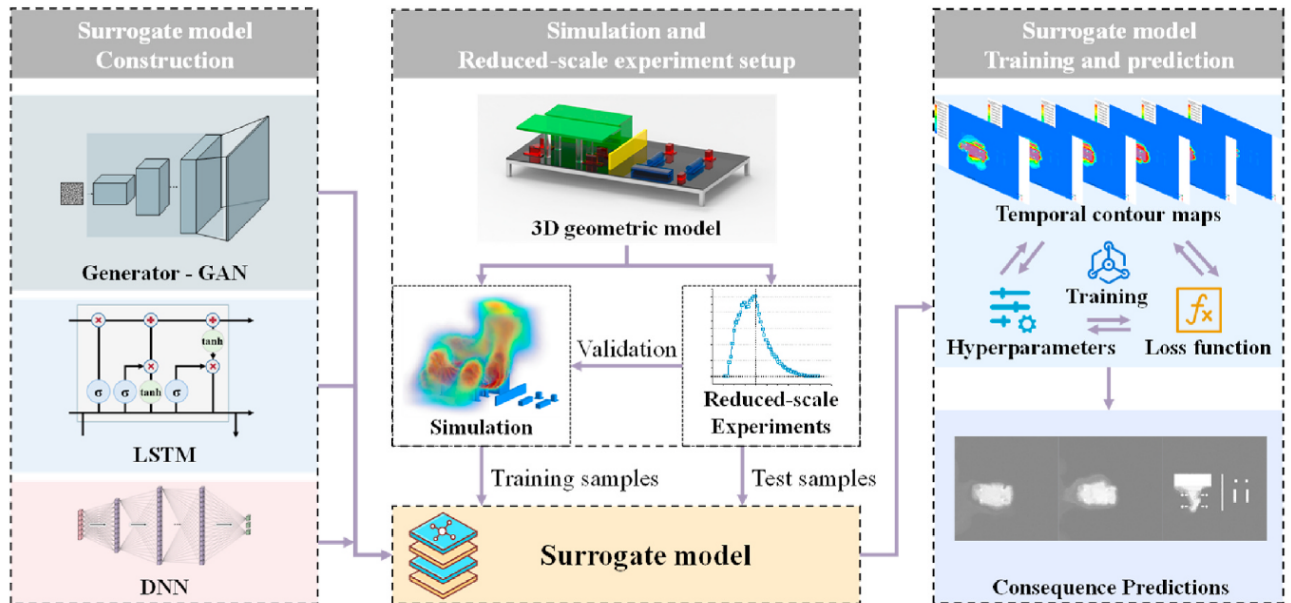


Abbildung 21: Literaturquelle 6 –Schema zur Entwicklung des KI-Modells (surrogate model)

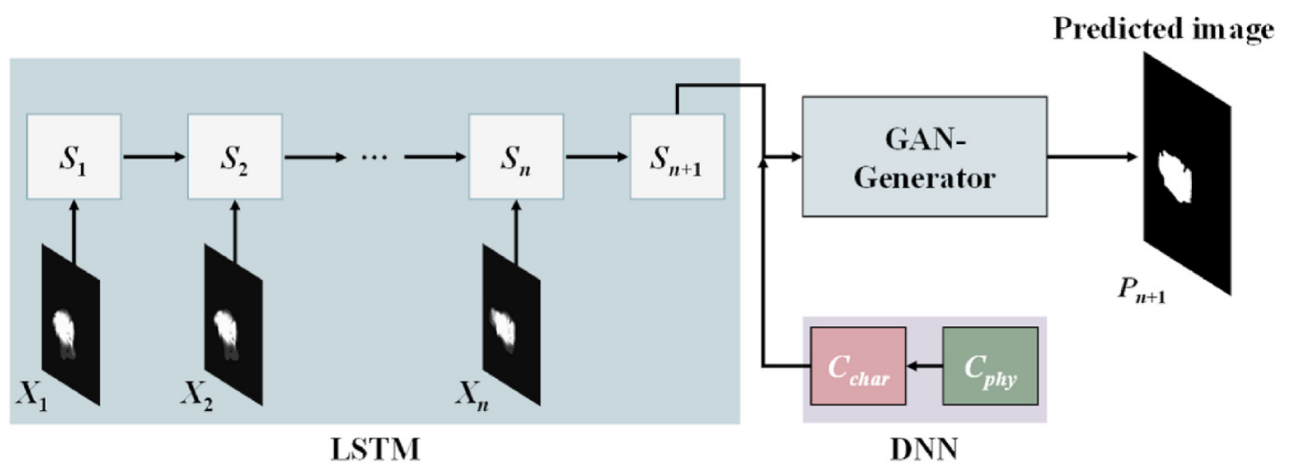


Abbildung 22: Literaturquelle 6 –Struktur des KI-Modells (hybrid neural network surrogate model)



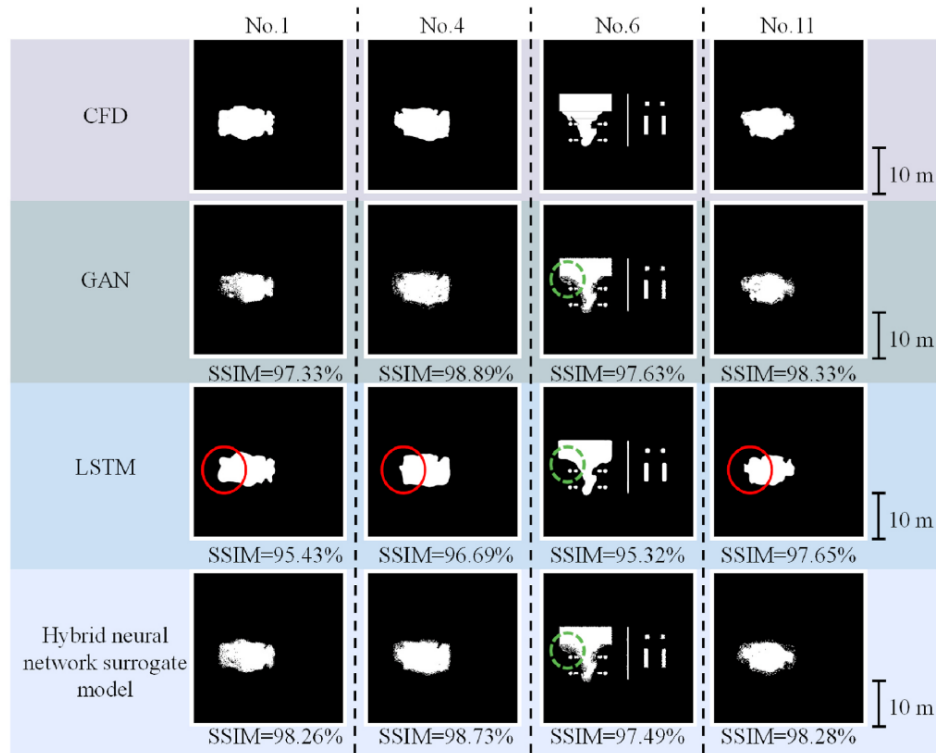


Abbildung 23: Literaturquelle 6 –Beispiele für vorhergesagte H₂-Konzentrationsverteilungen 10s nach der H₂-Freisetzung, Vergleich mit CFD-Ergebnissen (obere Bilder)

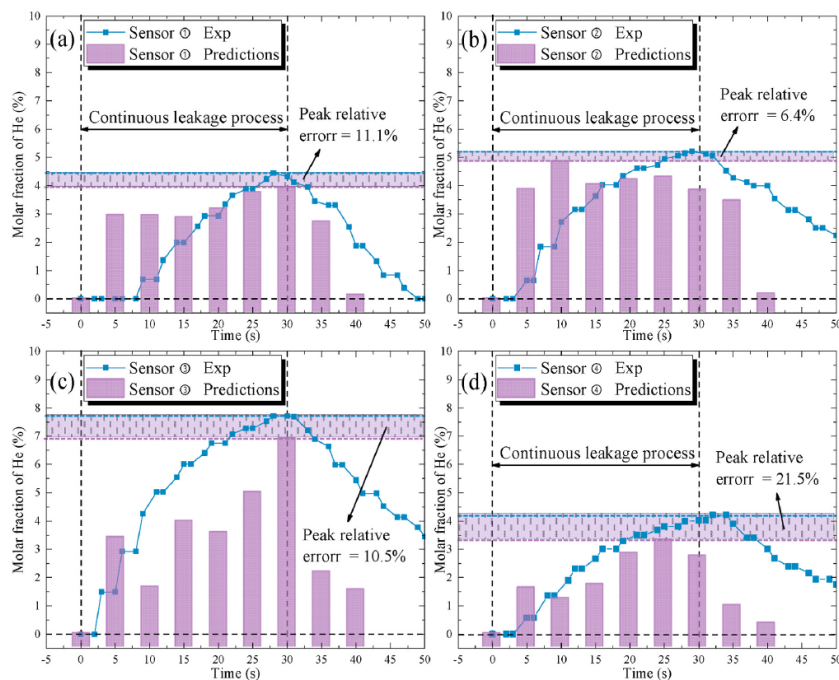
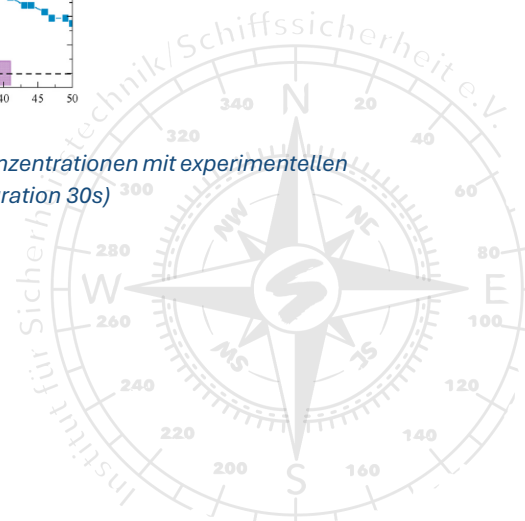


Abbildung 24: Literaturquelle 6 – Vergleich von vorhergesagten zeitabhängigen H₂-Konzentrationen mit experimentellen Ergebnissen (Szenario No.3, Leak point III, Leak direction +Z, Leak rate 6.9 L/min, Duration 30s)



9 Literaturquelle 7: Integrated CFD, Bayesian optimization, and deep learning for hydrogen leak detection on offshore renewable-energy platforms

Es wurden 420 Varianten einer H₂-Leckage auf einer offshore-Plattform mit der Software Fluent simuliert. Es wurden 140 virtuelle Sensoren verwendet, um die den zeitabhängigen Verlauf der H₂-Konzentration zu erfassen. Die Sensoranzahl wurde mit Hilfe eines erstellten KI-Modells von 140 auf 38 reduziert. Die Einsatzkosten sinken dadurch um 73 %, bei einer Erfassungsrate von 97.8 % (im Vergleich zu 140 Sensoren).

Das CFD-Modell der offshore-Plattform (100 m x 50 m x 25m) mit Leckageorten und den Leckagebedingungen s. Abbildung 25. Aus den 5 Windgeschwindigkeiten, 4 Windrichtungen, 3 Leckage-Durchmessern und 7 Leckageorten ergibt sich eine Variantenanzahl von 420. Die H₂-Massenströme für die 3 Leckage-Durchmesser 10 mm, 15 mm und 20 mm sind: 0.282 kg/s, 0.635 kg/s und 1.128 kg/s. Die simulierte Versuchszeit beträgt 30 s. Die Ergebnisse der optimierten Sensor-Anordnung sind in Abbildung 26 dargestellt.

Simulationsergebnisse des zeitabhängigen Volumens einer entzündbaren H₂-Wolke (≥ 4 Vol%) und einer H₂-Wolke mit ≥ 1 Vol% sind, in Abhängigkeit von der Windrichtung, der Windgeschwindigkeit und des Leckage-Durchmessers in Abbildung 27 dargestellt. Die H₂-Wolke mit ≥ 1 Vol% erreicht nach 30 s ein Volumen von bis zu 1600 m³, die entzündbare H₂-Wolke von bis zu ca. 43 m³ (upwind).

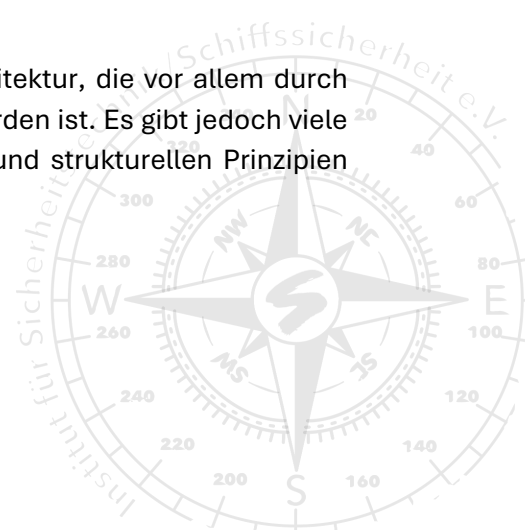
Der Arbeitsablauf des KI-Modells für die Bestimmung der Leckagestelle und der optimalen Sensoranordnung, basierend auf den CFD-Simulationsergebnissen, ist in Abbildung 28 enthalten.

Das KI-Modell ist ein Transformer-LSTM-Hybrid-Netzwerk, welches eine Kombination aus einem Transformer und einem LSTM (Long Short-Term Memory) Netzwerk. Das LSTM verarbeitet Daten Schritt für Schritt (sequenziell). Ist in der Lage, sowohl periodische als auch zufällige Verhaltensmuster von Zeitreihendaten zu erfassen (s. auch Literaturquelle 6). Bei extrem langen Sequenzen kann ein „Flaschenhals“ entstehen. Hier hilft der Transformer, welcher eine Selbstaufmerksamkeit besitzt, um alle Datenpunkte gleichzeitig bzw. globale Zusammenhänge des KI-Modells zu verstehen. Er benötigt bei großen Datenmengen entsprechend viel Speicherplatz. Es gibt verschiedene Möglichkeiten, die beiden Netzwerke miteinander zu koppeln (Kaskade, Encoder-Decoder-Ansatz, Cross-Attention-Interaktion).

Die Variation der verwendeten Hyperparameter beim Modelltraining ist in Tabelle 6 enthalten.

Nachfolgend einige ergänzende Erklärungen zum Begriff der Selbstaufmerksamkeit von KI-Modellen.

Die Selbstaufmerksamkeit ist das Herzstück der Transformer-Architektur, die vor allem durch moderne Sprachmodelle wie GPT, BERT oder Gemini bekannt geworden ist. Es gibt jedoch viele andere KI-Anwendungen, die auf völlig anderen mathematischen und strukturellen Prinzipien basieren.



Viele klassische und hochspezialisierte KI-Systeme kommen komplett ohne Selbstaufmerksamkeit aus:

- Faltungsnetzwerke (CNNs): Diese werden primär in der Bildverarbeitung und Computer Vision (z. B. für Gesichtserkennung oder medizinisches Screening) eingesetzt. Sie arbeiten mit lokalen Filtern, die Pixel für Pixel nach Mustern und Kanten absuchen, statt das gesamte Bild auf einmal in Relation zu setzen.
- Klassische Rekurrente Netze (RNNs/LSTMs): Ältere Sprachmodelle oder Systeme für Zeitreihenanalysen (z. B. Wettervorhersagen) verarbeiten Daten sequenziell (Schritt für Schritt). Sie merken sich Vergangenes in einem internen Speicher, statt – wie die Selbstaufmerksamkeit – alle Elemente gleichzeitig zu vergleichen.
- Klassisches Maschinelles Lernen: Algorithmen wie Random Forests, Support Vector Machines (SVM) oder die lineare Regression, die für Tabellendaten, Kreditrisikoanalysen oder Vorhersagen genutzt werden, basieren auf rein statistischen Mustern und nutzen keine neuronalen Aufmerksamkeitsmechanismen.

Die Selbstaufmerksamkeit kommt dort zum Einsatz, wo der Kontext und die Beziehungen aller Elemente untereinander entscheidend sind:

- Große Sprachmodelle (LLMs): Bei der Textgenerierung und Übersetzung sorgt die Selbstaufmerksamkeit dafür, dass das Modell versteht, worauf sich ein Wort wie „es“ in einem langen Satz bezieht.
- Vision Transformer (ViTs): Moderne Bild-KIs (wie sie oft in generativen Modellen wie Midjourney oder stabilen Diffusionsmodellen stecken) nutzen mittlerweile ebenfalls Transformer, um globale Zusammenhänge in Bildern besser zu verstehen als klassische CNNs.

Die Selbstaufmerksamkeit ist ein mächtiges Werkzeug für generative KI und komplexe Kontextanalysen. Für einfachere Mustererkennung, klassische Datenanalysen oder rein lokale Bildverarbeitungen nutzen Entwickler jedoch nach wie vor andere, recheneffizientere KI-Architekturen.



Tabelle 6: Literaturquelle 7 – Hyperparameter Analyse, mit ergänzenden Anmerkungen zu der Bedeutung von einigen Hyperparametern

Transformer-LSTM model hyperparameter analysis.

Parameters	Value
learning_rate	0.0001, 0.0005, 0.001, 0.005, 0.01
batch_size	8, 16, 32, 64, 128
weight_decay	1E-2, 1E-3, 1E-4, 1E-5
d_model	64, 128, 256, 512
nhead	2, 4, 8
trans_layer	2, 4, 6, 8
lstm_hidden	64, 128, 256
lstm_layer	1, 2, 3

learning_rate: Schrittweite

batch_size: Wie viele Datenpunkte ein Modell in einem einzigen Schritt verarbeitet

weight_decay: Steuergröße im maschinellen Lernen, die das Wachstum von Modellgewichten (Parametern) während des Trainings bremst, um Overfitting zu verhindern

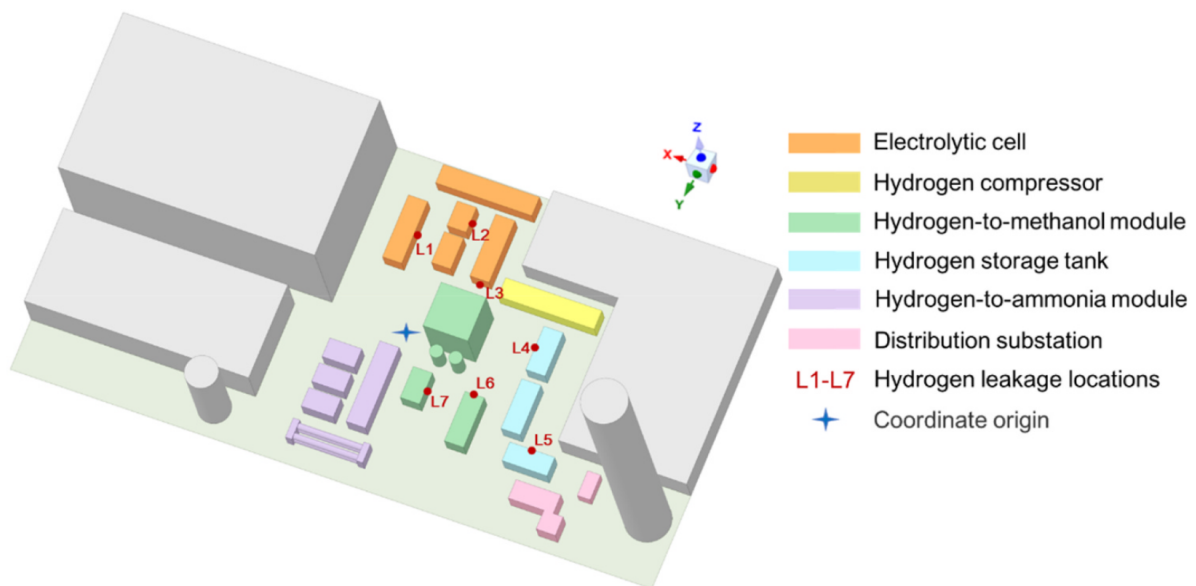
d_model: Vektorlänge eines Wortes / Token

nhead: Anzahl der parallelen Aufmerksamkeits-Kanäle in einem Transformer-Modell

trans_layer: Transformer-Schicht / Verarbeitungsschicht

lstm_hidden: verborgene Schicht, Kurzzeitgedächtnis des Modells, ein Vektor voller Zahlen, der sich bei jedem Zeitschritt verändert

lstm_layer: LSTM-Schicht, übergeordnete Struktur und Logik von lstm_hidden



Composition of leakage conditions.

Parameters	Value
Wind speed (m/s)	5, 7, 9, 25, 5-9
Wind direction	N, NW, S, SE
Leakage aperture (mm)	10, 15, 20
Leaking location	L1, L2, L3, L4, L5, L6, L7

Abbildung 25: Literaturquelle 7 – CFD-Modell der offshore Plattform mit Leckageorten und den Leckagebedingungen (untere Tabelle)



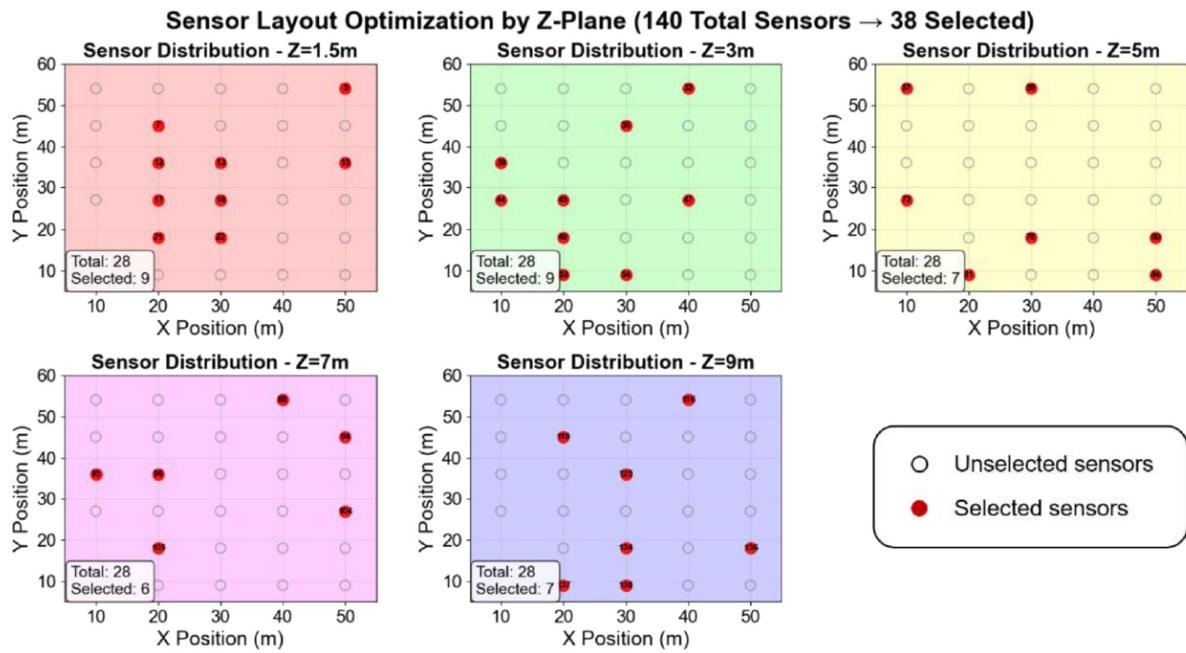
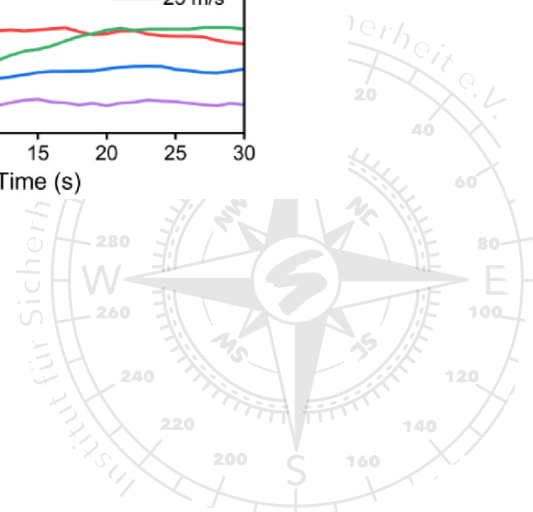
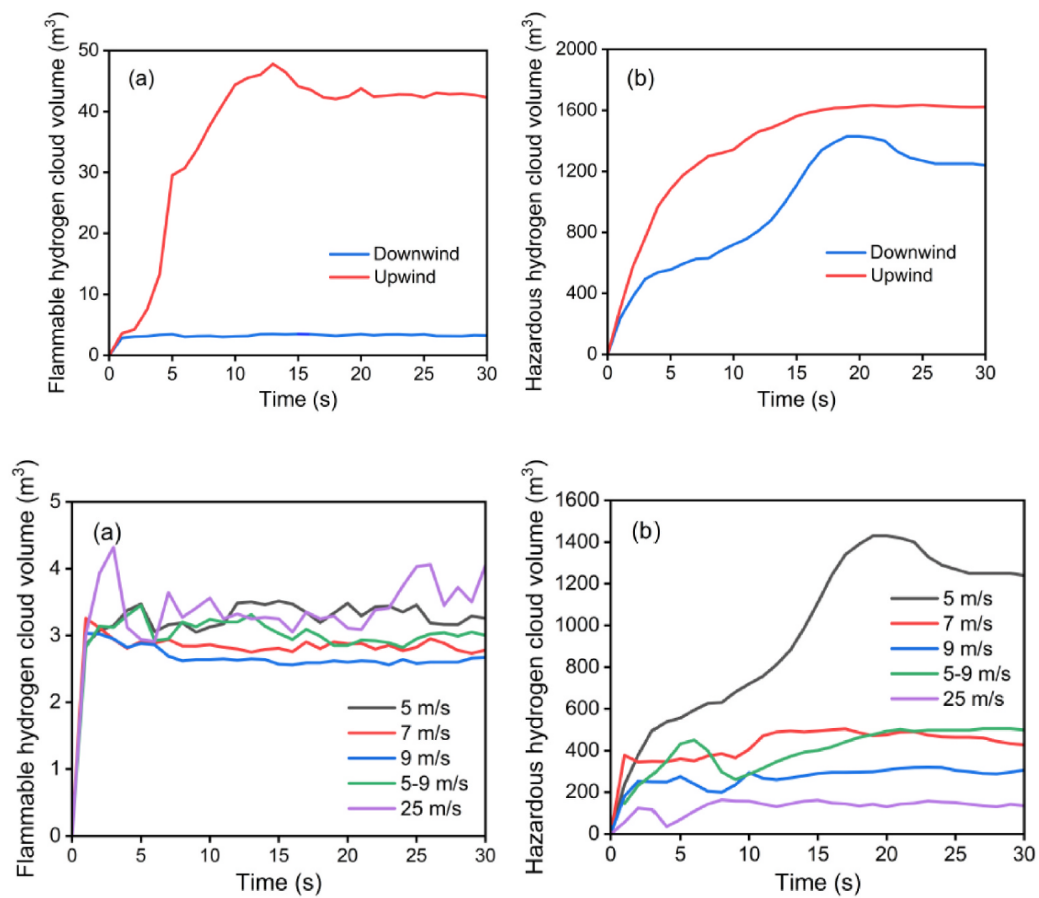


Abbildung 26: Literaturquelle 7 – optimierte Sensor-Anordnung



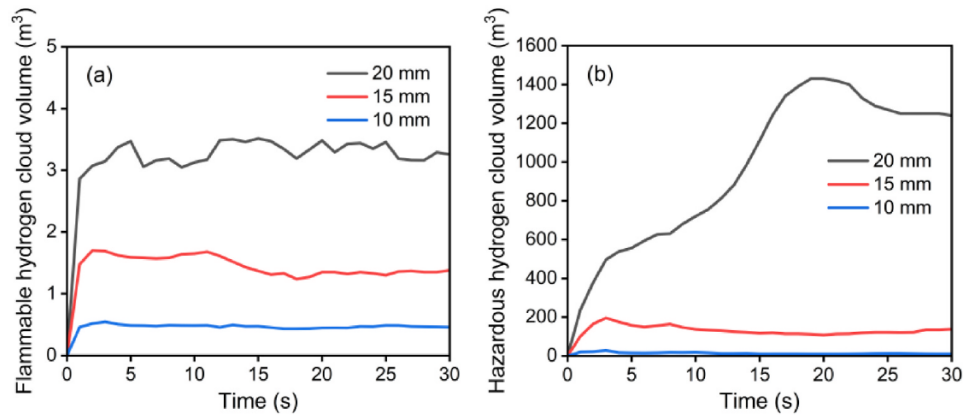


Abbildung 27: Literaturquelle 7 – zeitabhängiges Volumen einer entzündbaren H₂-Wolke (a, ≥ 4 Vol%, flammable cloud) und einer H₂-Wolke mit ≥ 1 Vol% (b, hazardous cloud) in Abhängigkeit von der Windrichtung (oberes Bild), der Windgeschwindigkeit (mittleres Bild) und des Leckage-Durchmessers (unteres Bild)

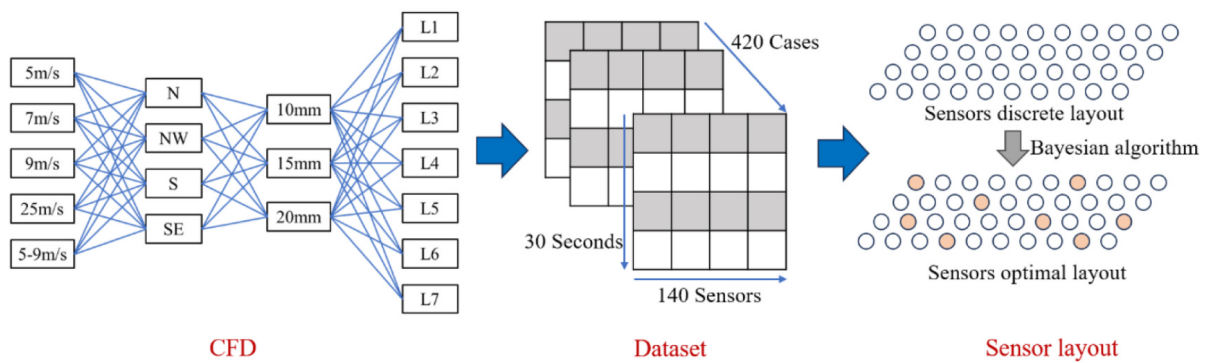


Abbildung 28: Literaturquelle 7 – Arbeitsablauf des KI-Modells für die Bestimmung der Leckagestelle und der optimalen Sensoranordnung (Bayes'sche Optimierung), basierend auf den CFD-Simulationsergebnissen



10 Hyperparameter

Es folgt eine Aufzählung der wichtigsten Hyperparameter, eingeordnet in 3 Kategorien:

Hyperparameter für die Architektur des ANN (Artificial Neural Network):

- Schichten- und Neuronenanzahl
- Aktivierungsfunktionen (Activation Functions): Bestimmen, wie das Signal eines Neurons umgewandelt wird (z. B. ReLU, Sigmoid oder GELU). Sie bringen die nötige Nicht-Linearität in das Modell.
- Gewichts-Initialisierung (Weight Initialization): Die Methode, wie die internen Gewichte ganz zu Beginn zufällig gewürfelt werden (z. B. He- oder Xavier-Initialisierung). Eine falsche Initialisierung kann dazu führen, dass das Netz gar nicht erst anfängt zu lernen.

Hyperparameter des ANN für den Lernprozess:

- Lernrate (Schrittweite)
- Batch-Größe: Anzahl der Trainingsläufe, bevor die Gewichte (interne mathematische Verknüpfungen) des KI-Modells geändert werden
- Der Optimizer (Optimierungsalgorithmus): Das Verfahren, mit dem das Netz die Gewichte anpasst (z. B. Adam, SGD oder RMSprop).
- Lernraten-Verfall (Learning Rate Decay / Schedule): Bestimmt, ob und wie stark die Lernrate im Laufe des Trainings automatisch kleiner werden soll (z. B. damit das Modell kurz vor dem Ziel feinfühler wird).
- Momentum: Hilft dem Algorithmus, „Schwung“ aufzubauen, um flache Regionen oder kleine Hügel in der Fehlerlandschaft schneller zu überwinden.

Hyperparameter des ANN gegen das Auswendiglernen, für den Umgang mit neuen unbekannten Daten:

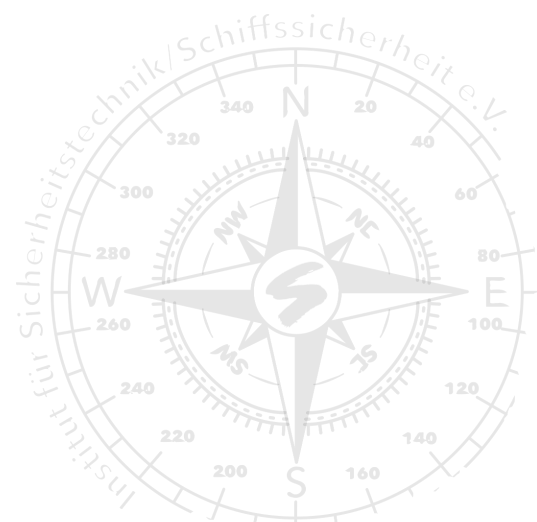
- Dropout Rate: Der Prozentsatz an Neuronen, die pro Trainingsschritt zufällig abgeschaltet werden.
- Weight Decay (L2-Regularisierung): Bestraft das Modell mathematisch dafür, wenn interne Gewichte zu extrem oder zu groß werden. Es zwingt das Modell zu einer „einfacheren“ und robusteren mathematischen Formel.
- Batch-Normalisierung (Batch Normalization): Stabilisiert das Training, indem die Ausgaben der Neuronen nach jeder Schicht neu skaliert (normalisiert) werden.

Die Bestimmung der Hyperparameter erfolgen z. B. mit einer Bayesischen Optimierung. In der Praxis wählt man 3 bis 5 der wichtigsten Hyperparameter aus. Das sind meistens die Lernrate, die Batch-Größe, die Architektur-Größe und einen Regularisierer wie dropout. Für die restlichen Parameter werden Standardwerte verwendet.

Bei der Bayesischen Optimierung wird ein mathematisches Wahrscheinlichkeitsmodell (meist über Gauß-Prozesse) aufgebaut. Der Algorithmus lernt mit relativ wenigen Trainingsdurchläufen die Einstellung der Hyperparameter. Nachteile sind die schwierige Implementierung, die Verwendung von relativ vielen Parametern und die Rechenzeit.

Die Optimierung der Hyperparameter kann z. B. mit der Software Optuna (Open-Source-Software-Framework in Python) erfolgen (wurde auch in Literaturquelle 4 angewendet).

Bei der Software Stochos (von der Firma PI Probaligence GmbH) erfolgt eine Bayesische Optimierung der Hyperparameter durch Anwendung von Gauß-Prozessen. Der verwendete Algorithmus nennt sich DIM-GP (*Deep Input Mixture of Gaussian Processes*). Die Integration der Gauß-Prozesse in das neuronale Netzwerk ermöglicht eine Unsicherheitsbewertung.



11 Literatur

- 1: Parth Patel, Vikram, Garaniya Til, Baalisam pang, Ehsan Arzaghi, Javad Mohammadpour, Rouzbeh Abbassi, Fatemeh Salehi: MACHINE LEARNING REGRESSION-CFD MODELS FOR PREDICTING HYDROGEN DISPERSION IN A NATURALLY VENTILATED AREA, Proceedings of the ASME 2023 42nd International Conference on Ocean, Offshore and Arctic Engineering, June 11-16, 2023, Melbourne, Australia
- 2: Zhao M et al., Leak localization using distributed sensors and machine learning for hydrogen releases from a fuel cell vehicle in a parking garage, International Journal of Hydrogen Energy, <https://doi.org/10.1016/j.ijhydene.2020.09.218>
- 3: Jinsheng Xiao, Nianfeng Xu, Yaze Li, Guodong Li, Min Liu, Liang Tong, Chengqing Yuan, Xuefang Li and Tianqi Yang: CFD Simulation and ANN Prediction of Hydrogen Leakage and Diffusion Behavior in a Hydrogen Refuelling Station, Wiley, International Journal of Energy Research Volume 2024, Article ID 8910533, 17 pages, <https://doi.org/10.1155/2024/8910533>
- 4: Wei Dong, Yuichi Sugai, Ying Shi, Theodora Noely Tambaria, Takehiro Esaki: Data-augmented deep learning for hazard assessment of hydrogen accumulation in confined spaces: Multitask prediction and sensitivity analysis, Elsevier, Fuel 394 (2025) 135056, <https://doi.org/10.1016/j.fuel.2025.135056>
- 5: Junjie Li a, Weikang Xie, Huihao Li, Xiaoyuan Qian, Jihao Shi, Zonghao Xie, Qing Wang, Xinqi Zhang, Guoming Chen: Real-time hydrogen release and dispersion modelling of hydrogen refuelling station by using deep learning probability approach, i n t e r n a t i o n a l j o u r n a l o f hydrogen energy 5 1 (2 0 2 4) 7 9 4 e8 0 6, <https://doi.org/10.1016/j.ijhydene.2023.04.126>
- 6: Xu He, Depeng Kong, Guodong Yang, Xirui Yu, Gongquan Wang, Rongqi Peng, Yue Zhang, Xinyi Dai: Hybrid neural network-based surrogate model for fast prediction of hydrogen leak consequences in hydrogen refueling station, International Journal of Hydrogen Energy 59 (2024) 187–198, <https://doi.org/10.1016/j.ijhydene.2024.01.328>
- 7: Zhihui Jin, Xingning Guo, Hao Fang, Zifei Zhang, Weijun Zhang, Cunwen Tang, Zuojun Wie: Integrated CFD, Bayesian optimization, and deep learning for hydrogen leak detection on offshore renewable-energy platforms, International Journal of Hydrogen Energy 231 (2026), ://doi.org/10.1016/j.ijhydene.2026.154917

