

# **Projekt: Edge AI DLBAIPEAI01\_D**

Fallstudie

Studiengang: Angewandte Künstliche Intelligenz

Sven Behrens

Matrikelnummer: 42303511

Prof. Dr. Bertram Taetz

12. Februar 2026

# Inhaltsverzeichnis

|  |            |
|--|------------|
| <b>Abbildungsverzeichnis</b>                           | <b>III</b> |
| <b>Tabellenverzeichnis</b>                             | <b>IV</b>  |
| <b>Abkürzungsverzeichnis</b>                           | <b>V</b>   |
| <b>1 Einleitung</b>                                    | <b>1</b>   |
| <b>2 Hauptteil</b>                                     | <b>2</b>   |
| 2.1 Projektumgebung . . . . .                          | 2          |
| 2.2 Datengrundlage . . . . .                           | 3          |
| 2.3 Datenvorverarbeitung . . . . .                     | 3          |
| 2.4 Modelltraining . . . . .                           | 3          |
| 2.4.1 Train/Test-Split . . . . .                       | 3          |
| 2.4.2 PM2.5-Modelle . . . . .                          | 3          |
| 2.4.3 NO2-Modelle . . . . .                            | 3          |
| 2.5 Feature Importance und Feature-Reduktion . . . . . | 3          |
| 2.6 Edge-Optimierung . . . . .                         | 3          |
| 2.6.1 ONNX-Export . . . . .                            | 3          |
| 2.6.2 Benchmark . . . . .                              | 3          |
| 2.7 Alert-System . . . . .                             | 3          |
| 2.8 Edge-Inference-Pipeline . . . . .                  | 3          |
| <b>3 Fazit</b>   | <b>3</b>   |
| 3.1 Zielerreichung und Projektergebnisse . . . . .     | 3          |
| 3.2 Kritische Reflexion . . . . .                      | 3          |
| 3.3 Verbesserungspotenziale . . . . .                  | 3          |
| 3.4 Ausblick . . . . .                                 | 3          |

|                                |          |
|--------------------------------|----------|
| <b>Projektrepository</b>       | <b>3</b> |
| <b>Literaturverzeichnis</b>    | <b>4</b> |
| <b>Verzeichnis der Anhänge</b> | <b>5</b> |
| <b>Anhang</b>                  | <b>5</b> |

**Abbildungsverzeichnis**

**Tabellenverzeichnis**

## **Abkürzungsverzeichnis**

**GRU** Gated Recurrent Unit

**MAE** Mean Absolute Error

**ONNX** Open Neural Network Exchange

**RMSE** Root Mean Squared Error

**WHO** World Health Organization

# 1 Einleitung

Die Luftqualität in städtischen Gebieten stellt ein wachsendes Gesundheitsrisiko dar. Insbesondere Feinstaub (PM<sub>2.5</sub>) und Stickstoffdioxid (NO<sub>2</sub>) werden mit Atemwegserkrankungen, Herz-Kreislauf-Beschwerden und einer erhöhten Mortalität in Verbindung gebracht (Lelieveld et al., 2021; World Health Organization, 2016). Besonders gefährdet sind vulnerable Gruppen wie Kinder, ältere Menschen und Personen mit Vorerkrankungen (European Environment Agency, 2023). An stark befahrenen Straßen in der Nähe von Schulen und Wohngebieten können die Schadstoffkonzentrationen kurzfristig kritische Werte erreichen, ohne dass rechtzeitig Gegenmaßnahmen eingeleitet werden.

Klassische Umweltüberwachungssysteme erfassen zwar kontinuierlich Messdaten, leiten daraus jedoch keine unmittelbaren Handlungsempfehlungen ab. Die Daten werden zentral gesammelt und erst mit Verzögerung ausgewertet. Für zeitkritische Anwendungen wie die dynamische Verkehrssteuerung bei Schadstoffspitzen ist dieser Ansatz unzureichend. Edge AI bietet hier einen vielversprechenden Lösungsansatz: Durch die lokale Ausführung von Vorhersagemodellen direkt an der Messstation können Prognosen in Echtzeit erstellt und Maßnahmen ohne Umweg über zentrale Server ausgelöst werden.

Vor diesem Hintergrund wurde im Rahmen des Moduls „Projekt: Edge AI“ an der IU Internationalen Hochschule ein System entwickelt, das die Luftqualität an der Messstation Hamburg Habichtstraße vorhersagt. Die Station befindet sich an einer stark befahrenen Straße in unmittelbarer Nähe einer Schule und eignet sich daher besonders als Anwendungsfall. Das System prognostiziert sowohl die PM<sub>2.5</sub>- als auch die NO<sub>2</sub>-Konzentration eine Stunde im Voraus und leitet daraus ein dreistufiges Ampelsystem ab, das bei erhöhter Belastung automatisch Verkehrsmaßnahmen wie Geschwindigkeitsbegrenzungen oder Umleitungen auslösen kann.

Die methodische Vorgehensweise umfasst mehrere aufeinander aufbauende Schritte. Zunächst werden stündliche Messdaten der OpenAQ-Plattform vorverarbeitet und durch Feature Engineering angereichert. Anschließend werden vier Modelltypen, lineare Regression, Random Forest, Gradient Boosting und Gated Recurrent Unit (GRU), trainiert und verglichen. Durch eine systematische Feature-Importance-Analyse wird das Feature-Set von 28 auf 17 Merkmale reduziert. Die besten Modelle werden in das Open Neural Network Exchange (ONNX)-Format exportiert und hinsichtlich Inferenzzeit und Modellgröße für den Edge-Einsatz evaluiert. Abschließend wird eine Edge-Inference-Pipeline implementiert, die den gesamten Ablauf von der Sensorablesung bis zur Ampelentscheidung abbildet.

Das vorliegende Projekt gliedert sich wie folgt: Nach der Beschreibung der Projektumgebung und der Datengrundlage werden die Vorverarbeitung und das Modelltraining erläutert. Anschließend werden die Edge-Optimierung, das Alert-System und die Inference-Pipeline vorgestellt. Abschließend werden die Ergebnisse kritisch reflektiert und Verbesserungspotenziale aufgezeigt.

## 2 Hauptteil

### 2.1 Projektumgebung

Zu Beginn des Projekts wurde ein GitHub-Repository<sup>1</sup> angelegt, um eine nachvollziehbare Versionsverwaltung zu gewährleisten. Als Implementierungssprache wurde Python gewählt. Für die Reproduzierbarkeit wurde eine virtuelle Umgebung mit `venv` eingerichtet, in der alle Abhängigkeiten über eine `requirements.txt` installiert werden. Zu den zentralen Bibliotheken zählen `pandas` und `numpy` für die Datenverarbeitung, `scikit-learn` für die klassischen Machine-Learning-Modelle, `torch` (CPU-Version) sowie `onnxruntime` und `skl2onnx` für den ONNX-Export und die Edge-Inferenz.

Die gesamte Pipeline, von den Rohdaten bis zur Edge-Inference-Demo, lässt sich mit einem einzigen Befehl (`python run_all.py`) ausführen. Das Skript ruft alle Verarbeitungsschritte sequenziell auf und stellt sicher, dass jedes Teilskript im korrekten Arbeitsverzeichnis ausgeführt wird. Dadurch kann ein Tutor die Ergebnisse ohne manuelle Konfiguration reproduzieren.

---

<sup>1</sup> <https://github.com/svenb23/EdgeAI>



## **2.2 Datengrundlage**

## **2.3 Datenvorverarbeitung**

## **2.4 Modelltraining**

### **2.4.1 Train/Test-Split**

### **2.4.2 PM2.5-Modelle**

### **2.4.3 NO2-Modelle**

## **2.5 Feature Importance und Feature-Reduktion**

## **2.6 Edge-Optimierung**

### **2.6.1 ONNX-Export**

### **2.6.2 Benchmark**

## **2.7 Alert-System**

## **2.8 Edge-Inference-Pipeline**

## **3 Fazit**

### **3.1 Zielerreichung und Projektergebnisse**

### **3.2 Kritische Reflexion**

### **3.3 Verbesserungspotenziale**

### **3.4 Ausblick**

## **Projektrepository**

Der vollständige Quellcode ist im GitHub-Repository verfügbar: <https://github.com/svenb23/EdgeAI>

## Literatur

- European Environment Agency. (2023, 24. April). *Air pollution and children's health*. <https://www.eea.europa.eu/en/analysis/publications/air-pollution-and-childrens-health>
- Lelieveld, J., Hahad, O., Daiber, A., & Münzel, T. (2021). Luftverschmutzung und Herz-Kreislauf-Erkrankungen. *Aktuelle Kardiologie*, 10(06), 510–515. <https://doi.org/10.1055/a-1546-7355>
- World Health Organization. (2016, 12. Mai). *Air pollution levels rising in many of the world's poorest cities*. <https://www.who.int/news/item/12-05-2016-air-pollution-levels-rising-in-many-of-the-world-s-poorest-cities>

## **Verzeichnis der Anhänge**

### **Anhang**