

Masterarbeit ProTemp

Masterarbeit A – Data & Model	Masterarbeit B – Edge Deployment & MLOps
<p>„Few-Shot Foundation-Modelle für die präzise Regression des Ethylenglykol-Wasser-Verhältnisses“</p>	<p>„Edge-AI-Architektur und MLOps-Pipeline für self-learning Prozessthermostate“</p>
<p>Motivation Das Paper von Harfmann et al. zeigt eine sehr genaue Klassifikation lediglich zweier Mischungen (50/50, 60/40). Ziel ist nun die kontinuierliche Regression aller Mischverhältnisse in 1%-Schritten. Moderne Time-Series-Foundation-Modelle (TimesFM, TiRex / xLSTM) versprechen starke Generalisierung auch bei wenigen realen Samples.</p>	<p>LAUDA will das Modell schnell in Serie bringen. Dafür braucht es eine leichte, wartungsarme Edge-Runtime auf mehreren Gerätemodellen plus eine MLOps-Toolchain, die neue Daten automatisiert in Training, Tests und Over-the-Air-Updates überführt.</p>
<p>Haupt-Forschungsfragen</p>	<p>Haupt-Forschungsfragen</p>
<p>RQ A1: <i>Wie gut lassen sich TimesFM/TiRex mit wenigen gelabelten Samples auf eine $\pm 1\%$-Punkte-Genauigkeit für das vollständige EGW-Spektrum (40–60 % v/v) fein-tunen?</i></p>	<p>RQ B1: <i>Welche Runtime- und Kompressionsverfahren (TFLite, TVM, ONNX-INT8) erfüllen < 200 ms Inferenzzeit und < 400 MB RAM auf der bestehenden LAUDA-Elektronik?</i></p>
<p>RQ A2: <i>Welche Feature-Sets aus Sensorsignalen maximieren die Domänentransferfähigkeit auf weitere Thermostat-Baureihen?</i></p>	<p>RQ B2: <i>Wie muss eine CI/CD-Pipeline gestaltet sein, damit neue Sensor-Daten von der Edge bis zum retrainierten Modell binnen < 24 h ausgerollt werden können – unter Berücksichtigung von Versionierung, Drift-Monitoring und Rollback?</i></p>
<p>Kernaufgaben</p>	<p>Kernaufgaben</p>
<ol style="list-style-type: none"> 1. Aufsetzen einer automatisierten Datenerfassungsumgebung (MQTT/OPC-UA, Docker). 2. Few-Shot-Fine-Tuning von TimesFM und TiRex (LoRA/Adapter) vs. klassischen Regressoren (XGB, RF). 3. Explainability & Feature-Ranking zur Reduktion der Sensorik. 4. Validierung auf zusätzlichen LAUDA-Geräten (Cross-Device-Generalisation). 	<ol style="list-style-type: none"> 1. Design eines Edge-KI-Frameworks (C-Wrapper, REST/gRPC). 2. Vergleich Quantisierung, Pruning, Distillation; Benchmark auf Jetson Orin & Intel NUC. 3. Aufbau eines GitLab-CI-Templates: Auto-Retraining, Modell-Tests, Canary-Rollout. 4. Entwicklung eines Edge-Agenten für selbstständige Datenerfassung & Health-Metrics (CPU, Temp, Modell-Drift).
<p>Erwartete Artefakte</p>	<p>Erwartete Artefakte</p>

<ul style="list-style-type: none"> • Transfer-Learning-Pipeline (Jupyter + MLflow). • Modelle (.onnx, .tflite) + Vergleichsreport (MAE, inference cost). 	<ul style="list-style-type: none"> • Prototypische Edge-Runtime • End-to-End-CI/CD-Blueprint (Infrastructure-as-Code). • Dashboards (Grafana) für Modell- und Geräte-Telemetry.
<p>Synergie</p>	<ol style="list-style-type: none"> 1. Daten-Lieferung: Masterarbeit B nutzt die von A erhobenen Daten & Modelle. 2. Feedback-Loop: Edge-Performance-Metriken aus B fließen als Retraining-Trigger in A zurück.
<p>Vorkenntnisse</p>	<p>Python ≥ 3.10, TensorFlow / PyTorch, scikit-learn, TimesFM/TiRex, MLflow, Hyperparameter-Tuning</p>
<p>Aufwand & Logistik</p>	<p>Je 6 Monate, Start simultan; 1 Labortag/Woche in Lauda-Königshofen, Rest remote.</p>