Systems & Control Transactions

**Research Article - Peer Reviewed Conference Proceeding**
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

# A Comparative Analysis of Industrial MLOps prototype for ML Application Deployment at the edge devices

**Fatima Rani[a,b,*], Fenin Jose[a,b], Lucas Vogt[a,b], and Leonhard Urbas[a,b]**

[a] Faculty of Electrical and Computer Engineering, Chair of Process Control Systems & Process Systems Engineering Group, TUD Dresden University of Technology, 01069 Dresden, Germany

[b] Process-to-Order-Group, TUD Dresden University of Technology, 01069 Dresden, Germany
* Corresponding Author: fatima.rani@tu-dresden.de

## ABSTRACT

This paper introduces a prototype for constructing an edge AI system utilizing the contemporary Machine Learning Operations (MLOps) concept. By employing microcontrollers such as the Raspberry Pi as hardware, our methodology includes data scrubbing and machine learning model deployment on edge devices. Crucially, the MLOps pipeline is fully developed within the ecoKI platform, a research platform for ML/AI applications. In this study, we thoroughly investigate the performance of our ecoKI platform by comparing it with the established Edge Impulse platform. We deployed the ML model with different weight quantization methods, such as FP32 and INT8, to compare accuracy variations and inference speed between these two platforms and quantization strategies on edge devices. In our experiments, we identified that the average accuracy performance of the ecoKI platform is 3.61% better than the edge impulse. Moreover, real-time AI processing on edge devices enables microcontrollers, even those with limited resources, to effectively handle tasks in areas such as predictive maintenance, process optimization, quality assurance, and supply chain management.

**Keywords**: Artificial Intelligence, Industry 4.0, Energy Efficiency, Machine Learning, Big Data, Edge Intelligence.

## INTRODUCTION

Presently, Edge AI is a rapidly growing field combining edge computing principles with artificial intelligence (AI) to enable the execution of AI algorithms directly on edge devices. This eliminates the need to transmit and process data on remote servers. Thereby This significantly reduces latency and improves the efficiency of AI-driven applications. By processing data closer to its source, Edge AI supports faster decision-making and enhances real-time functionality. According to recent market analysis [1], the global Edge AI hardware market, estimated at $24.2 billion in 2024, is projected to reach approximately $55 billion by 2029. This growth is driven by the increasing adoption of Internet of Things (IoT) devices across various industries, including smart homes, industrial automation, healthcare, agriculture, and transportation.

While edge computing has existed for years, the integration of AI has recently fueled the development of Edge AI [2]. Many IoT devices lack the computational power to handle complex processing tasks and, therefore, often rely on cloud-based solutions that can be costly and time-intensive. Edge AI addresses this limitation by enabling on-device data processing and reducing dependence on external data centers. Additionally, industries increasingly focus on lightweight vision inspection systems designed for specific tasks, moving away from bulkier automated optical inspection (AOI) machines. These streamlined edge device-based solutions offer advantages such as real-time insights, reduced physical space requirements, and simplified setups. However, several challenges remain in fully realizing the potential of Edge AI. One major issue is the limited availability of detailed performance benchmarks, particularly for processing large datasets or high-resolution images.

## RELATED WORK

Recent studies have evaluated the performance of various Edge AI devices for deep learning applications, particularly object detection, highlighting how different

accelerators like GPUs and TPUs provide viable alternatives to traditional cloud-based processing [3]. Evaluations focused on throughput, power consumption, cost-effectiveness, accuracy, and latency [4,5]. Edge devices with AI chips demonstrated significant performance improvements compared to traditional edge devices. However, the choice of device depends on specific application requirements, as performance varies across different architectures and precision levels [6]. While some devices, like the Jetson AGX Orin and Intel NUC, showed promising results, others, like the Raspberry Pi, exhibited limitations in processing speed [7]. These comparative analyses provide valuable insights for selecting appropriate edge AI devices based on specific use cases and performance needs.

## .METHODOLOGY & DATASET

### A) Methodology

To derive our main insights, we perform a comparative analysis of industrial Edge MLOps platforms, specifically Edge Impulse and the ecoKI Platform. We follow contemporary MLOps practices [8,9], including model optimization and deployment on edge devices. "Lastly, we compare performance characteristics using an evaluation matrix consisting of the ROC curve, precision, recall, and F1 score. Based on this methodology, we elaborate on our findings about the pros and cons of both platforms in the next sections.

### B) Dataset

In line with the objectives of evaluation studies, aligning the experiment's goals with a suitable dataset is crucial. In this work, our initial task involved selecting an appropriate dataset and preparing a trained model. Given our primary focus on wearable devices, we selected a dataset tailored to this domain. To further enhance the analysis, we incorporated a dataset capable of detecting 12 distinct sleep positions with finer granularity under energy-efficient constraints [10]. Leveraging accelerometer data from wearable devices across various timeframes, this dataset supports detailed position inference, offering insights valuable for both sleep quality analysis and energy-efficient wearable solutions.

## EVALUATION

### A) Edge Impulse

Edge Impulse [11], a leading AI development platform, enables intelligence on any edge device. There are several steps to perform a complete inference in the Edge impulse platform and for the deployment.

#### i. Data acquisition

For the sleep dataset, a CSV wizard was used to load the data from a .csv file into the Edge Impulse platform

#### ii. Create Impulse

Creating an impulse is central to Edge Impulse, combining features, data pre-processors, neural network architectures, and desired outputs. Raw data undergoes signal processing to extract features, which are subsequently used by learning blocks for tasks such as classification or regression. Signal processing includes DSPs and FFTs to reduce noise and generate intermediate inputs for learning. Then, spectral analysis and classification were selected for accelerometer data, and raw data was converted into processed features using FFT to enhance model complexity and learning capability.

#### iii. Signal Processing Block

In Edge Impulse, spectral features processing is used to extract frequency, power, and signal characteristics from accelerometer data, leveraging FFT to decompose signals into frequency components.

This approach filters unwanted frequencies, analyzes patterns, and reduces model size for efficient and reliable machine learning. Here, the pre-processing block has been used spectral features for the sleep classification data. This SP block extracts a signal's frequency, power, and other characteristics.

#### iv. Learning Block

The learning block defines the neural network architecture, comprising a fully connected model with a 63-neuron input layer, two hidden layers (80 and 40 neurons), and a 12-class output layer as depicted in the Figure 1.

The result of the classifier for training data set for model FP32 is shown in Figure 2 and corresponding metric in table 1. The on-device performance can be seen in Figure 3 with accuracy and its loss.

Similarly, the result of the classifier for training data set for model int8 is depicted in Figure 4 and corresponding metric in table 2. The on-device performance can be seen in Figure 5 with accuracy and its loss.

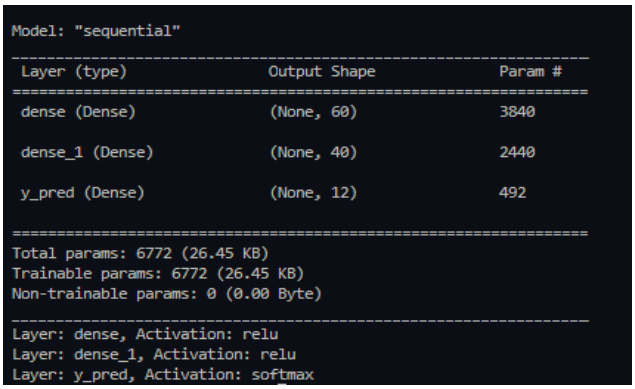



**Figure 1.** Deep Neural Network architecture used in Edge Impulse platform.

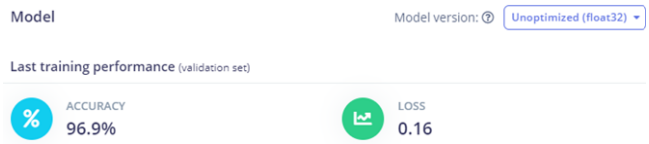



**Figure 2.** Validations result of DNN unoptimised (float32) model in Edge Impulse platform.

**Table 1:** Metrics **(validation set of unoptimized (float32) model).**

| METRIC | Value |
|---|---|
| Area under ROC Curve | 1.00 |
| Weighted avg. Precision | 0.98 |
| Weighted avg. Recall | 0.97 |
| Weighted avg. F1 score | 0.97 |


**Figure 3.** On-device performance of unoptimised (float32) model.

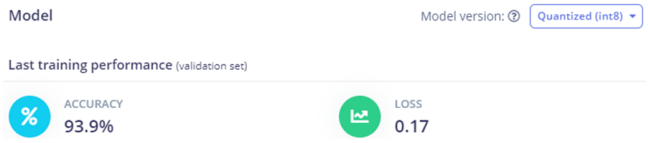



**Figure 4.** Validation result of DNN Quantized(int 8) model in Edge Impulse platform.

**Table 2:** Metrics **(validation set of quantised (int8) model).**

| METRIC | Value |
|---|---|
| Area under ROC Curve | 1.00 |
| Weighted avg. Precision | 0.96 |
| Weighted avg. Recall | 0.94 |
| Weighted avg. F1 score | 0.93 |


**Figure 5.** On device performance of Quantized(int 8) model.

## v. Model Test Result

Generally, quantization reduces model weight precision, potentially impacting performance, with options like quantized (INT8) and unoptimized (Float32) models offering varying accuracy.

In Edge impulse, model testing involves selecting the different versions, either unoptimized (float32) or quantized (int8) via the platform; the result of each has been presented in Tables 3 and 4 correspondingly.

**Table 3:** FP_32 accuracy, Metrics for Classifier test results.

| Accuracy: 90.39% | |
|---|---|
| **METRIC** | **Value** |
| Area under ROC Curve | 1.00 |
| Weighted avg. Precision | 0.98 |
| Weighted avg. Recall | 0.97 |
| Weighted avg. F1 score | 0.97 |

**Table 4:** int_8 accuracy, Metrics for Classifier test results.

| Accuracy: 88.21% | |
|---|---|
| **METRIC** | **Value** |
| Area under ROC Curve | 1.00 |
| Weighted avg. Precision | 0.96 |
| Weighted avg. Recall | 0.94 |
| Weighted avg. F1 score | 0.93 |

## vi. Model Deployment

The model offers versatile deployment options, including a standalone C++ application, board-specific binaries for devices like Nordic Thingy:91, or downloadable Keras (.h5) and TFLite formats. Testing confirms successful performance even on resource-constrained hardware such as the Raspberry Pi 2 (ARM V7).

**Table 5:** Metrics comparison for FP32 and Int8 quantization of ML model deployment result at Edge Impulse.

| Model quantization | File size | Ram usage | Inference time | Accuracy |
|---|---|---|---|---|
| FP 32 | 11MB* | 128KB | 0ms | 90.39 |
| INT_8 | 11MB* | 128KB | 0ms | 88.21 |

*11 MB is the file size of the build not the whole zip.

# B) ecoKI Platform

The ecoKI platform follows a streamlined workflow from data preparation to deployment: data loading, selection, train-test splitting, hyperparameter tuning, neural network training, and time series visualization. This systematic process ensures efficient and effective implementation of the ecoKI-Platform.

## i. ecoKI Data Reader

The ecoKI data reader building block enables seamless data integration through multiple input methods: local CSV files, MongoDB databases, or Dataverse integration [12]. This versatile approach ensures compatibility with diverse data sources, streamlining the entire workflow while maintaining optimal efficiency.

## ii. Data Selection

In ecoKI platform, the front end provides the functionality of data selection. This process not only enhances data quality by removing noise and redundancies

but also reduces computational overhead, enabling more efficient processing and analysis.

### iii. Split train and test data

In ecoKI, the front end provides the functionality of splitting data into training and testing because selecting input and output features is essential. This selection ensures ecoKI preprocesses the data into the required format for seamless analysis.

### iv. Train and predict NN multi-class

Then, ecoKI performs training and testing via the dashboard using a fully connected neural network with customizable activation functions, epochs, and neuron counts. Upon completion, it generates the model (.h5), weights, parameters, and a TFLite model for inference, as can be seen in Figure 6.

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 3)]               0

dense (Dense)                (None, 10)                40

dense_1 (Dense)              (None, 20)                220

dense_2 (Dense)              (None, 40)                840

dense_3 (Dense)              (None, 12)                492

dense_4 (Dense)              (None, 12)                156

=================================================================
Total params: 1748 (6.83 KB)
Trainable params: 1748 (6.83 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Layer: input_1, Activation: None
Layer: dense, Activation: relu
Layer: dense_1, Activation: relu
Layer: dense_2, Activation: relu
Layer: dense_3, Activation: relu
Layer: dense_4, Activation: softmax
```

**Figure 6.** Deep Neural Network architecture (classification) used in ecoKI platform.

### v. Time series visualizer

The final step involves generating data for analysis supported by visualization capabilities integrated into this module, as illustrated in Figure 7 (a). This feature allows for exploring key insights by presenting critical data points in an accessible and interpretable format. Through comprehensive visualization tools, users can effectively analyze trends, patterns, and relationships within the data, enabling more informed decision-making and a deeper understanding of the results.

### vi. Model Test Result

Model evaluation in ecoKI is conducted using comprehensive testing metrics, including a confusion matrix to assess classification accuracy and detailed performance measures such as Mean Squared Error (MSE), accuracy, precision, recall, and F1 score. The results can be seen in Figure 7. b and c. These metrics provide a holistic understanding of the model's performance, capturing its predictive accuracy and ability to balance false positives and negatives. This rigorous evaluation ensures the reliability and effectiveness of the trained model in real-world applications.

### vii. Model Deployment

The ecoKI platform provides a TFLite model in FP32 format, designed for seamless integration and deployment. The model can be wrapped using C++ or Python wrappers to enable inference, which serve as interfaces to invoke the TFLite interpreter; the model result is shown in Table 6.

**Table 6:** Model deployment result on ecoKI platform.

| Model quantization | File size | Ram usage | Inference time | Accuracy |
|---|---|---|---|---|
| FP 32 | 9.53KB | 128KB | 0.1ms | 94% |

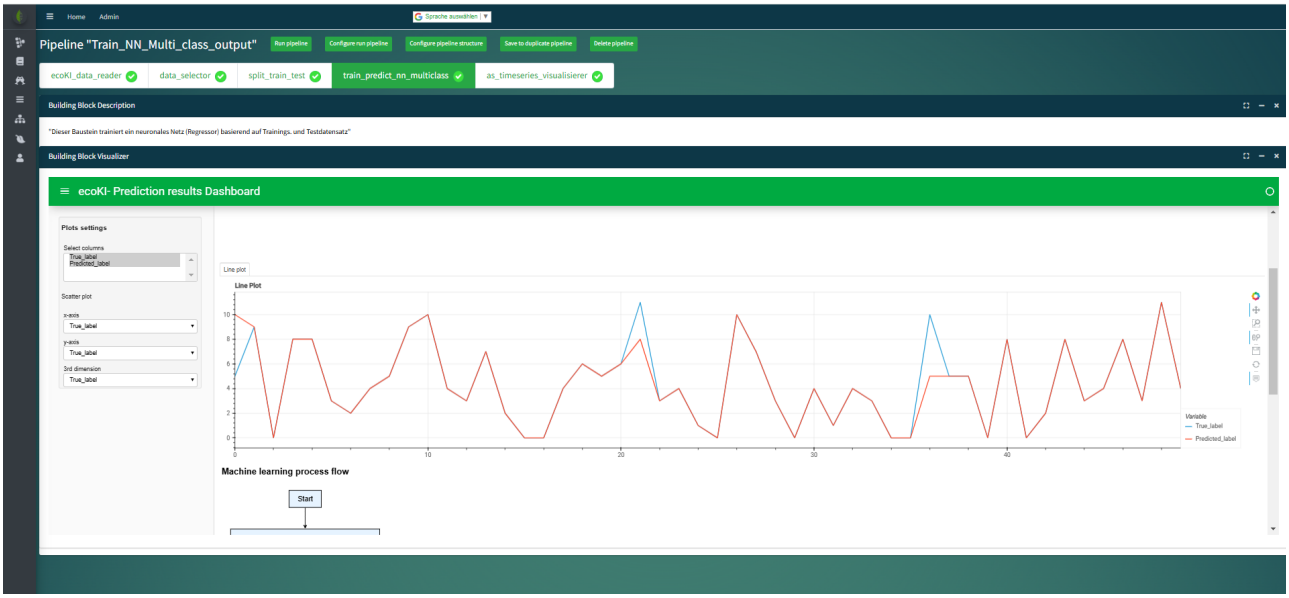
## DISCUSSION AND CONCLUSION

In the rapidly evolving field of machine learning at the edge, platforms like ecoKI and Edge Impulse provide powerful tools for efficient model development and deployment. Both platforms offer robust features tailored to specific needs, but they differ in their approaches, strengths, and limitations.

On the one hand, ecoKI excels in comprehensive data preprocessing, handling various data formats seamlessly, and offering advanced features such as ModelCheckpoint and dynamic learning rate adjustments.

*ModelCheckpoint* ensures the model or its weights are saved at specific intervals in checkpoint files, enabling training to resume from a saved state if needed. It offers flexibility by allowing users to save only the best-performing model based on a monitored metric or to save the model at the end of every epoch, regardless of performance. Users can define 'best performance' by selecting the metric to monitor and determining whether it should be maximized or minimized. In contrast, the dynamic learning rate adjustments allow for a dynamic reduction of the learning rate when the monitored metrics show no improvement for a specified number of epochs. This adaptive learning rate reduction, typically by a factor of 2–10, helps models to overcome stagnation and enhances convergence, ultimately improving training efficiency and model performance by 3.61% for the FP32 model compared to Edge impulse. ecoKI's flexible workflow also supports detailed hyperparameter tuning and real-time data processing. However, larger datasets may require more setup and computational resources.

On the other hand, Edge Impulse provides a user-




**(a)**

| index | pred0 | pred1 | pred2 | pred3 | pred4 | pred5 | pred6 | pred7 | pred8 | pred9 | pred10 | pred11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| True_0 | 576 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| True_1 | 0 | 598 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| True_2 | 0 | 0 | 611 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| True_3 | 0 | 0 | 0 | 599 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| True_4 | 0 | 0 | 0 | 0 | 569 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| True_5 | 0 | 0 | 0 | 0 | 0 | 146 | 0 | 0 | 0 | 0 | 469 | 0 |
| True_6 | 0 | 0 | 0 | 0 | 0 | 0 | 618 | 0 | 0 | 0 | 0 | 0 |
| True_7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 594 | 0 | 0 | 0 | 0 |
| True_8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 595 | 0 | 0 | 0 |
| True_9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 606 | 0 | 0 |
| True_10 | 0 | 0 | 0 | 0 | 0 | 41 | 0 | 0 | 0 | 0 | 583 | 0 |
| True_11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 592 |

**(b)**

| Index | Metric | 0.0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 | 7.0 | 8.0 | 9.0 | 10.0 | 11.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | MSE | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.07 | 0.0 | 0.0 | 0.0004 | 0.0 | 0.00088 | 0.0004 |
| 1 | Accuracy | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.929 | 1.0 | 1.0 | 0.9995 | 1.0 | 0.4411 | 0.9995 |
| 2 | Precision | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.78 | 1.0 | 1.0 | 0.9949 | 1.0 | 0.2727 | 1.0 |
| 3 | Recall | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.237 | 1.0 | 1.0 | 1.0 | 1.0 | 0.707 | 0.9949 |
| 4 | F1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.364 | 1.0 | 1.0 | 0.9974 | 1.0 | 0.0882 | 0.9974 |

**(c)**

**Figure 2:** a) The front-end dashboard of the EcoKI platform provides an intuitive interface for visualizing test data and b) test predictions, enabling comprehensive validation of model performance. It includes a confusion matrix offering insights into classification accuracy. c) Additionally, model metrics such as Mean Squared Error (MSE), accuracy, precision, recall, and F1 score are evaluated and displayed to ensure thorough performance assessment.

friendly interface for rapid model development and de ployment, which is especially suited for edge devices. Its support for lightweight quantized models and pre-built learning blocks enhances efficiency as it provides the standalone C++ model and TFlite interpreter. That is the reason it supports many hardware configurations. However, it has limitations in handling complex data pipelines and high-performance inference tasks. In summary, ecoKI offers flexibility and deep data integration, while Edge Impulse excels in ease of use and edge deployment efficiency. The ultimate choice depends on specific research needs and deployment requirements. The results of this study could be extended in future research by expanding the number of compared MLOps platforms and the number and variety of the used metrics. Furthermore, these platforms' scaling and reliability characteristics

require further investigation to make decisive statements about their capabilities.

## ACKNOWLEDGEMENTS

This work was funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant number 03EN2047C.

## DECLARATION OF AI-ASSISTED TECHNOLOGIES

While preparing this manuscript, the authors used AI tools to improve language and readability. They carefully reviewed all content, taking full responsibility for the research work and ideas presented in this publication.

## REFERENCES

1. Marketsandmarkets.https://www.marketsandmarkets.com/Market-Reports/edge-ai-hardware-market-158498281.html
2. Singh R, & Gill S. S. Edge AI: a survey. Internet of Things and Cyber-Physical Systems, 71-92 (2023).
3. Rohit C, Sant R, Nawade S, Vedhant V, & Gupta V. Exploring edge artificial intelligence: A comparative study of computing devices for deployment of object detection algorithm. INCET, pp. 1-5 (2023).
4. Su W, Li L, Liu F, He M, & Liang X. AI on the edge: a comprehensive review. Artificial Intelligence Review, 55(8), 6125-6183 (2022).
5. Amanatidis, P., Iosifidis, G., & Karampatzakis, D. (2021, November). Comparative evaluation of machine learning inference machines on edge-class devices. PCI,pp. 102-106,2(021).
6. Garcia-Perez A, Miñón R, Torre-Bastida A. I, & Zulueta-Guerrero E. Analysing edge computing devices for the deployment of embedded AI. *Sensors*, *23*(23), 9495 (2023).
7. Samsuri M. H, Yuen S. L, Lau P Y, Wong C. W, Kamarudin N A, Hussin Z, & Ho, H W. Comparative Performance Analysis of Edge-AI Devices in Deep Learning Applications. ICIEA, pp. 1-6 (2024). https://ieeexplore.ieee.org/document/10665079
8. Rani F, Chollet N, Vogt L, & Urbas L. Industrial Edge MLOps: Overview and Challenges. CACE, 53, 3019-3024 (2024). https://doi.org/10.1016/B978-0-443-28824-1.50504-4
9. Rani F, Khaydarov V, Bode D, Hasan I. H, & Urbas, L. MLOps Practice: Overcoming the Energy Efficiency Gap, Empirical Support Through ecoKI Platform in the Case of German SMEs. PAC- World Global Conference (2023).
10. IEEE Dataset. https://ieeedataport.org/documents/sleep-position-classification-using-single-abdominal-accelerometer
11. Janapa Reddi V, Elium A, Hymel S, Tischler D, Situnayake D, Ward C, & Quaye J. Edge impulse: An mlops platform for tiny machine learning. MLSys, 5. (2023).
12. Rani F, Wang X, Mutlu I, & Urbas L. A Centralized MongoDB-Based Repository Design for IIoT Data: The ecoKI Project. IFAC-PapersOnLine, 56(2), 3184-3189 (2023). https://doi.org/10.1016/j.ifacol.2023.10.1454