DOI: https://doi.org/10.36910/6775-2524-0560-2026-63-03
UDC 004.75:004.8:504.064
**Borodii Ivan**, Postgraduate Student
https://orcid.org/0009-0005-4986-4429
**Osukhivska Halyna**, Ph.D., Associate Professor
https://orcid.org/0000-0003-0132-1378
Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

# AIR QUALITY ANALYSIS BASED ON THRESHOLD VALUES AND ANOMALY DETECTION USING MACHINE LEARNING MODELS

**Borodii I., Osukhivska H. Air quality analysis based on threshold values and anomaly detection using machine learning models.** The article addresses two applied tasks in local air quality monitoring: threshold-based condition classification and anomaly detection in micro-batch sensor data. Air quality degradation has dynamic character shaped by particulate matter, $CO_2$, temperature and relative humidity. Data were collected from SCD41, BME688 and SPS30 sensors forming micro-batches of 10 sequential observations stored in DuckDB. Based on 33190 measurements, 3319 micro-batches were formed, each described by 60 statistical features. Logistic regression, decision tree, random forest and rule-based baseline were evaluated for threshold-based classification using thresholds for PM2.5, PM10, $CO_2$, temperature and humidity. Isolation forest, local outlier factor with and without PCA and one-class SVM were assessed against rule-based anomaly labels and on micro-batches with extreme feature values. The decision tree reproduced the labelling logic. Random forest achieved F1 = 0.857 under time-series cross-validation and F1 = 0.980 on test set. Isolation forest showed the highest F1 against rule-based anomaly labels. Local outlier factor showed stronger sensitivity to outliers in the feature space. Training time was below 500 ms and inference time below 109 ms. Results confirm both tasks can be implemented in an environmental monitoring system using edge computing and method selection should reflect the anomaly type and classification error costs.
**Keywords:** air quality monitoring, environmental monitoring, edge computing, machine learning, micro-batching processing, anomaly detection, internet of things.

**Бородій І.І., Осухівська Г.М. Аналіз стану якості повітря на основі порогових значень та виявлення аномалій з використанням моделей машинного навчання.** У статті розглядаються дві прикладні задачі в галузі моніторингу якості повітря на локальному рівні: класифікація стану на основі порогових значень та виявлення аномалій у мікропакетах сенсорних даних. Погіршення якості повітря має динамічний характер, що визначається вмістом твердих частинок, $CO_2$, температурою та відносною вологістю. Дані збиралися з сенсорів SCD41, BME688 та SPS30, які утворювали мікропакети з 10 послідовних спостережень, що зберігалися в базі даних DuckDB. На основі 33190 вимірювань було сформовано 3319 мікропакетів, кожен з яких описувався 60 статистичними ознаками. Для класифікації на основі порогових значень з використанням граничних значень для PM2.5, PM10, $CO_2$, температури та вологості було оцінено логістичну регресію, дерево рішень, випадковий ліс та базову модель на основі правил. Ізоляційний ліс, локальний фактор винятків з PCA та без нього, а також однокласовий SVM були оцінені щодо міток аномалій на основі правил та на мікропакетах з граничними значеннями ознак. Дерево рішень відтворило логіку мітки. Випадковий ліс досяг F1 = 0,857 під час перехресної валідації часових рядів та F1 = 0,980 на тестовому наборі. Ізоляційний ліс показав найвищий F1 щодо міток аномалій на основі правил. Локальний фактор винятків продемонстрував вищу чутливість до винятків у просторі ознак. Час навчання становив менше 500 мс, а час ухвалення рішення менш ніж 109 мс. Результати підтверджують, що обидва завдання можна реалізувати в екологічній системі моніторингу з використанням периферійних обчислень, а вибір методу повинен враховувати тип аномалії та витрати, пов'язані з помилками класифікації.
**Ключові слова:** моніторинг якості повітря, екологічний моніторинг, периферійні обчислення, машинне навчання, мікропакетна обробка, виявлення аномалій, інтернет речей.

**Scientific problem statement.** In modern environmental monitoring systems, it is becoming increasingly important not only to record current air quality but also to perform intelligent assessments of environmental quality without constant reliance on cloud infrastructure. The need for this arises from the fact that air quality degradation is often dynamic in nature and can develop gradually under the influence of a combination of factors, including ventilation patterns, $CO_2$ concentrations, changes in temperature and humidity, the presence of particulate matter, as well as the spatial and temporal distribution of data. Under these conditions, simple accumulation of measurements is no longer sufficient, and systems capable of locally assessing potential air quality risks, detecting anomalies in sensor data, and ensuring a quick response without the need to transmit all measurements to the cloud are becoming more valuable [1].

The transition from traditional monitoring to intelligent analysis is associated with the need to use Machine Learning (ML) models capable of processing multidimensional sensor data, considering the interrelationships between environmental indicators, and detecting potentially harmful or abnormal conditions. At the same time, the use of such models in centralised or fully cloud-based environments is not always the best solution, as it increases the system's dependence on the communication channel, increases the delay between the moment of measurement and the obtaining of the assessment result, and complicates continuous operation under conditions of unstable network infrastructure. That is why, edge

computing approaches are becoming increasingly relevant for tasks involving local threshold classification of air quality and anomaly detection, whereby part of the computation is transferred directly to a peripheral node located close to the data source [2].

The practical value of this approach is particularly high in cases where the system must operate autonomously, be cost-effective, ensure rapid response times, and utilise available computing resources. Under such conditions, the task of threshold classification of air quality and detection of abnormal changes in sensor data faces additional constraints. ML models must ensure acceptable classification quality and anomaly detection under the constraints of limited computational resources at the edge node, making their comparative evaluation particularly important. A particular challenge is that assessing air quality in a real-world environment cannot rely on a single parameter. A robust assessment of air quality typically requires consideration of a combination of different indicators that describe the microclimatic conditions of the environment and the presence of particulate matter. This combination makes it possible to evaluate air quality more accurately than using a single method. Under these conditions, scientific interest extends not only to the implementation of a monitoring system but also to the comparative evaluation of ML models capable of ensuring acceptable quality of sensor data analysis within limited time and computational resources.

The relevance of the research lies in the need for intelligent analysis of indoor air quality, the deterioration of which is dynamic in nature and is shaped by the simultaneous influence of several environmental indicators: particulate matter concentration, $CO_2$, temperature and relative humidity. Simply accumulating sensor measurements does not ensure the timely detection of potentially adverse conditions and abnormal changes, which makes it necessary to process the data directly at the edge node.

**Research analysis.** Recent research in the field of air quality monitoring indicates a gradual shift from the simple collection of sensor measurements towards the development of intelligent systems for local analysis, condition classification, anomaly detection and short-term forecasting. A systematic review [1] shows that Internet of Things technologies and Artificial Intelligence methods are increasingly being used for sensor calibration, anomaly detection, air quality index assessment, and short-term forecasting. This conclusion confirms the general relevance of the transition from observation to predictive analysis.

The architectural prerequisites for moving analytical computations closer to the data source are revealed in the review [2], which compares edge, cloud and hybrid approaches in the Internet of Things environment. It is shown that local processing provides lower latency, better control over data and a lower load on the network infrastructure.

The relationship between air quality and energy efficiency is investigated in [3], which is dedicated to the simultaneous optimisation of these indicators in educational buildings using ML. The authors demonstrate that GRU and LSTM models achieve an accuracy of over 92 %, enabling real-time control of ventilation systems, considering the dynamics of $CO_2$, temperature and air pollutants.

Paper [4] provides a systematic overview of low-cost sensor systems and Internet of Things technologies for air quality monitoring. Attention is given to the selection of sensors, data transmission arrangements, processing models, and the need for calibration and validation of measurements, which is essential for building reliable local monitoring systems.

The broader context of smart sensor use is presented in the review [5], where indoor air quality is considered one of the key components of indoor environmental quality. In this paper, the authors demonstrate that the integration of sensors with Internet of Things networks and building management systems is becoming the foundation of modern adaptive solutions.

A practical implementation of continuous air quality monitoring in real-life domestic conditions is presented in [6], which describes an Internet of Things architecture for monitoring $CO_2$ concentration and particulate matter levels during sleep. The authors obtained results demonstrating the environment's sensitivity to ventilation modes and confirmed the practical value of continuous measurements.

The study [7] justifies the use of low-cost PM2.5 sensors for indoor air quality monitoring. It is shown that such sensors can serve as a practical basis for mass monitoring, but require a reference sensor, periodic calibration and correction procedures.

Of particular interest to the topic of intelligent air quality analysis is the study [8], dedicated to the assessment and forecasting of the dynamics of atmospheric pollutants in the urban environment of Zhytomyr. It analyses temporal changes in the concentrations of CO, VOC ($H_2CO$), PM10, PM2.5, PM1.0, $NH_3$ and $NO_2$ for the years 2019–2024 and provides a forecast based on seasonal fluctuations and factors influencing the distribution of pollutants in urban air.

Article [9] proposes an Internet of Things-based system for monitoring and forecasting PM2.5 at both the edge and in the cloud. However, the results presented in study [10] showed that running the model locally on a Raspberry Pi requires not only building the model itself but also optimising it, specifically through compression and quantisation. Works [9] and [10] confirm that edge forecasting is technically feasible, but at the same time demonstrate that model comparisons must take into account not only the quality of classification or anomaly detection, but also the latency of local execution.

A decentralised approach to air quality forecasting is presented in [11], where Internet of Things technologies are combined with federated learning. It is shown that local pre-processing and training at the node level can reduce the need for raw data transmission and enhance privacy.

General architectural prerequisites for shifting computations to the periphery are discussed in [12] and [13], where edge computing is considered as a response to the limitations of cloud-centric approaches in the Internet of Things environment. These works demonstrate that bringing processing closer to the data source makes it possible to reduce latency and improve the practical suitability of distributed services.

In [14], signal processing and ML models deployed at the edge are combined to detect anomalies in real time in Internet of Things sensor networks. The authors demonstrated that local feature extraction and classification can simultaneously reduce communication overhead, latency and energy consumption.

The suitability of using the Raspberry Pi as an edge computing platform is also confirmed in [15], [16] and [17]. In [15], a scalable edge computing environment based on the Raspberry Pi kit is proposed. [16] demonstrates the possibility of running compact Edge AI models on the Raspberry Pi in a visual inspection application. [17] describes the practical implementation of an IoT-based air pollution monitoring system using the Raspberry Pi for real-time collection and transmission of sensor data.

An important factor in implementing a forecasting system is the choice of method for local data storage and the mode of data processing. In [18], it is shown that different approaches to data organisation can vary significantly in terms of performance and resource efficiency. In [19], it is established that the stream processing mode directly affects latency and throughput.

At the same time, the analysed sources do not sufficiently cover a holistic approach that would combine the assessment of air quality thresholds and the detection of abnormal changes in sensor data micro-batches within a single monitoring system. Attention should be paid to comparing different anomaly detection methods that assess the compliance of thresholds established on real data and the ability to detect statistically significant deviations in the feature space.

**Definition of the research goal.** The purpose of the study was to develop and validate a method for air quality analysis that combines threshold-based classification of environmental conditions with the detection of abnormal changes in sensor data micro-batches. To achieve the goal, ML methods have been used that are suitable for reproducing the threshold logic for determining a potentially unfavourable air quality condition and detecting abnormal changes in sensor data micro-batches in edge computing environments.

The main focus is on the ability of ML methods to replicate the logic for determining threshold values of potentially adverse air quality conditions and to detect abnormal changes in sensor readings, as well as on their suitability for local execution under the resource constraints of an edge node.

**Presentation of the main material and the justification of the results.** The experimental part of the study focused on implementing and verifying a threshold-based method for classifying air quality and detecting abnormal changes in sensor data micro-batches in a monitoring system using a Raspberry Pi 5 edge device. Within the scope of this work, the device was considered as a standalone node for the collection, storage, processing and analytical evaluation of sensor readings.

The hardware foundation of the experimental environment consisted of a Broadcom BCM2712 quad-core processor and 8 GB of LPDDR4X RAM. The system was implemented in Python 3.13.5 running on a Linux environment. The DuckDB database version 1.4.4 was used for the local storage of sensor measurements, providing built-in analytical data processing without the need to deploy a separate server component. Feature engineering, model training and result evaluation were performed using the pandas, NumPy, scikit-learn and psutil libraries. This set of technologies made it possible to combine sensor data collection, local storage, micro-batch generation, model training and resource performance recording within a single experimental environment. A general structural diagram of the implemented air quality monitoring and analysis system is shown in Fig. 1.

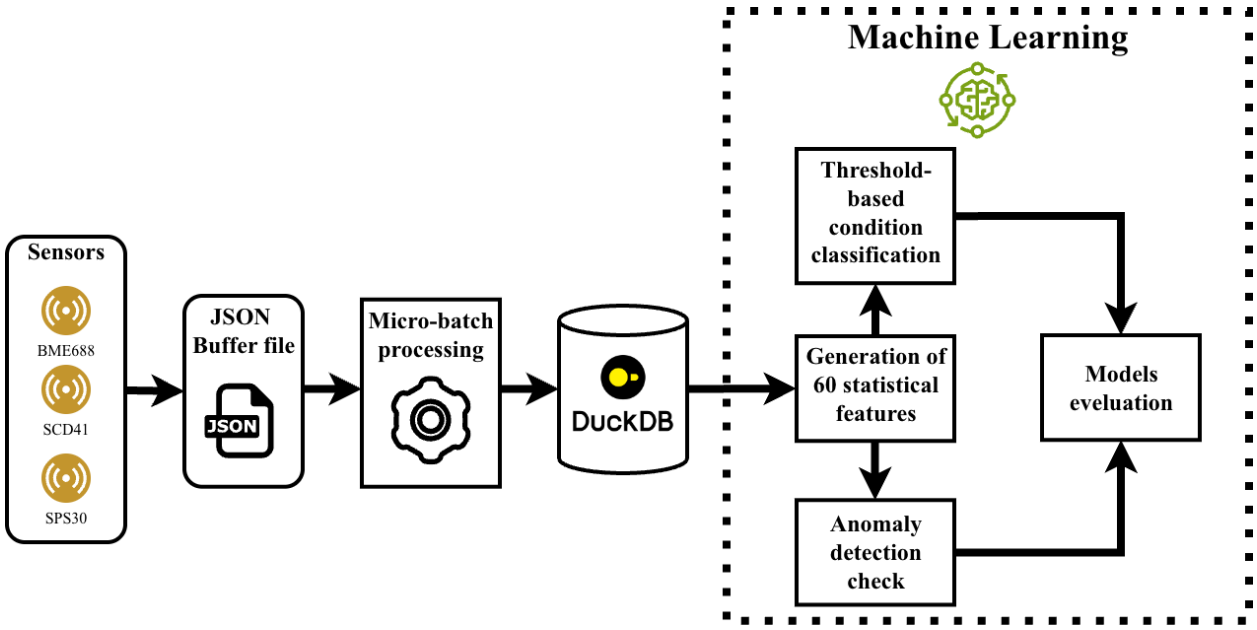


Fig. 1. General structural diagram of the local air quality monitoring and analysis system.

Data collection was performed under local air monitoring conditions using three sensor modules, each of which was responsible for a separate set of parameters. Specifically, the SCD41 sensor was used to measure $CO_2$ concentration, temperature and relative humidity. The BME688 was used to measure temperature, humidity, atmospheric pressure and gas resistance. The SPS30 measured the mass and number concentrations of PM1.0, PM2.5, PM4.0 and PM10 particulate matter, as well as the typical particle size. The system's workflow was organized into repetitive loops, with sensor measurements stored in the local DuckDB database. In addition to sensor readings, the system recorded performance metrics, including the duration of individual stages, sensor reading status and RAM usage.

Further processing involved converting individual sensor measurements into fixed-size micro-batches. A single micro-batch contained 10 consecutive readings and was used to describe short-term changes in air quality parameters. Sensor measurements were stored in a local DuckDB table, after which, during the ML data preparation stage, they were loaded from the database and aggregated by micro-batch identifier. For each micro-batch, a single row of a feature matrix was formed in RAM. At this stage, for each sensor field, the mean, minimum and maximum values, standard deviation, change between the first and last readings, and the range of variation were calculated. A total of 10 sensor fields were used, so the resulting vector contained 60 statistical features. The resulting feature matrix was not stored as a separate DuckDB table but was used as an intermediate data representation for label generation, sample time division, and the training of classification and anomaly detection models. The general description of the resulting experimental dataset is presented in Table 1.

Table 1. Description of the Experimental Dataset

| Indicator | Value |
|---|---|
| Number of primary observations | 33190 |
| Number of feature batches | 3319 |
| Number of features per batch | 60 |
| Training set | 2655 batches |
| Test set | 664 batches |
| Positive health-risk class | 993 batches (29.9 %) |
| Negative health-risk class | 2326 batches (70.1%) |

In the study, the generated micro-batches of data were used to define two distinct tasks: threshold classification of potentially adverse air quality and the detection of abnormal changes in sensor readings. The aim of the threshold classification task is to determine whether a micro-batch contains readings that may indicate potentially unfavourable air quality. These indicators include concentrations of PM2.5, PM10, $CO_2$, temperature and relative humidity. To generate a binary label for a potentially adverse air quality

condition, a threshold assessment scheme was used, whereby each micro-batch of sensor data was checked against the maximum or threshold values of key air quality indicators. A micro-batch was classified as potentially unfavourable if at least one of the monitored parameters exceeded the corresponding threshold value or fell outside the permissible range. The risk label in this study was not a medical diagnostic indicator but was used as an operational indicator of a potentially adverse air quality condition within the monitoring system.

The selection of threshold values was based on indoor air quality guidelines. In particular, the PM2.5 limit value of 15 μg/m³ was selected in accordance with the WELL Building Standard's requirements for indoor air quality, which stipulate that PM2.5 concentrations must be below 15 μg/m³ [20]; a $CO_2$ threshold of 1000 ppm was used as a practical threshold for potentially inadequate ventilation, consistent with the use of $CO_2$ as an indicator for controlling indoor ventilation [21]; temperature ranges of 19–24 °C and relative humidity ranges of 30–70 % were used as the operating limits of the indoor microclimate, as they are consistent with the permissible thermal comfort parameters for building interiors cited in the literature [22].

For the anomaly detection task, a label was generated based on the range of variation for each monitored parameter within a micro-batch, calculated as the difference between its maximum and minimum values. If at least one of the ranges exceeded the corresponding threshold value, the micro-batch was identified as abnormal. The threshold values for this task were determined empirically based on an analysis of the distribution of variation ranges in the training sample: the threshold values were selected to record only sharp and non-typical changes in indicators that fall outside the limits of normal system operation. The summary criteria for label formation for both tasks are presented in Table 2.

Table 2. Criteria for Establishing Air Quality Labels

| Sensor indicator | Risk condition | Anomaly condition |
|---|---|---|
| PM2.5 | max > 15 μg/m³ | range > 15 μg/m³ |
| PM10 | max > 45 μg/m³ | - |
| $CO_2$ | max > 1000 ppm | range > 200 ppm |
| Temperature | min < 19 °C or max > 24 °C | range > 1.5 °C |
| Relative humidity | min < 30 % or max > 70 % | range > 10 % |
| Pressure | - | range > 2 hPa |

The data was split into training and test sets regarding the time sequence. The first 80 % of micro-batches were used for training, and the remaining 20 % for testing. The experimental setup replicated a practical scenario where the model was trained on previously accumulated data and applied to new observations. Additionally, five-fold temporal cross-validation was used on the training set, which enabled the assessment not only of the overall quality of the models but also of their stability across different time intervals.

Three models were selected for the threshold classification task of air quality: logistic regression, decision tree and random forest. Logistic regression served as the baseline linear model, making it possible to assess the extent to which the task could be solved using a relatively simple linear boundary between classes. The decision tree was included due to its ability to reproduce threshold dependencies, corresponding to the nature of the generated labels. The random forest was applied as an ensemble model capable of accounting for the interaction of many features and reducing the instability typical of a single tree. In addition, the experiment utilised a basic threshold model that required no training and directly applied the rules by which the risk labels were formed. Table 3 presents the results of temporal cross-validation for the air quality threshold classification task, enabling the stability of the threshold labelling logic to be evaluated on different time segments of the training sample.

Table 3. Results of Time-Series Cross-Validation of Threshold-Based Air Quality State Classification Models

| Model | Mean F1 | F1 Standard Deviation | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| Rule-based baseline model | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Logistic Regression | 0.544 | 0.206 | 0.918 | 0.708 | 0.475 |
| Decision Tree | 0.605 | 0.360 | 0.940 | 0.798 | 0.528 |

| | | | | | |
|---|---|---|---|---|---|
| Random Forest | 0.857 | 0.205 | 0.967 | 0.976 | 0.817 |

The results of the baseline threshold model were as expected, as it applied the same rules used to generate the air quality labels. Therefore, this model was not considered a training model, but rather a deterministic baseline for comparison with ML models.

Following a temporal cross-validation, the threshold classification methods were evaluated on a separate test subset of data that had not been used during training. The results of this stage are interpreted as the ability of the methods to reproduce the specified operational logic of threshold classification of air quality status, rather than as confirmation of the independent medical or environmental validity of the classification. Additionally, the training time, the time taken to generate an assessment for a new micro-batch, and the size of the saved model were obtained. The results of the test evaluation are presented in Table 4.

Table 4. Classification Model Results on the Test Set

| **Model** | **F1** | **Accuracy** | **Precision** | **Recall** | **Training time** | **Inference time** | **Model size** |
|---|---|---|---|---|---|---|---|
| Rule-based baseline model | 1.000 | 1.000 | 1.000 | 1.000 | not applicable | not applicable | not applicable |
| Logistic Regression | 0.821 | 0.792 | 0.711 | 0.972 | 24 ms | 0.32 ms | 1.1 KB |
| Decision Tree | 1.000 | 1.000 | 1.000 | 1.000 | 69 ms | 0.31 ms | 1.4 KB |
| Random Forest | 0.980 | 0.980 | 1.000 | 0.960 | 322 ms | 4.2 ms | 56 KB |

A summarised relationship between the F1 score and the training time of threshold-based air quality classification models is shown in Fig. 2. Since the air quality label was generated using threshold rules, the high performance of the models should be interpreted as their ability to reproduce the specified evaluation logic.

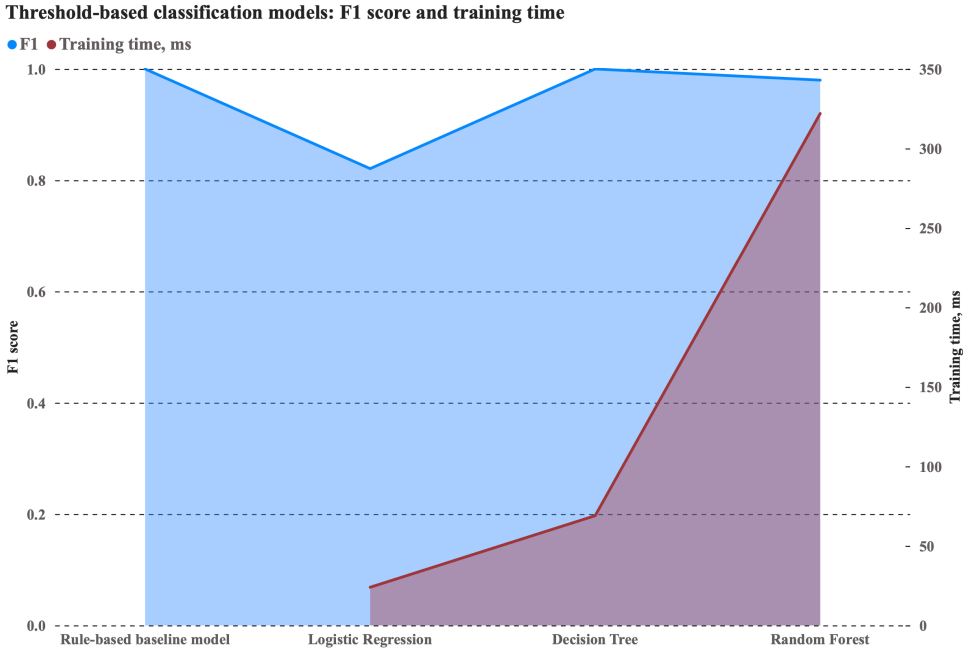


Fig. 2. F1 score and training time of threshold-based air quality condition classification models.

The complete reproduction of the threshold logic of the decision tree classification is explained by the threshold nature of the classification and the presence of minimum and maximum values of the monitored indicators in the feature vector. The results obtained indicate that the task of threshold classification of air quality can be successfully solved using ML methods in a local execution environment. Logistic regression and decision trees had the shortest estimation time, whereas random forest required more time but achieved the highest F1 score among the trained models.

Additionally, a statistical comparison of the models was performed using the McNemar's test, the purpose of which was to verify the differences between the predictions of the two models on the same test sample. The criterion analyses the confusion matrix between the pair of models, where b is the number of micro-batches that the first model classified correctly but the second incorrectly; c is the number of micro-batches that the second model classified correctly but the first incorrectly. Both metrics are integers and reflect the number of divergent predictions on the test sample of 664 micro-batches. The Statistic value reflects the calculated value of the Edwards-adjusted chi-square statistic and shows how significantly the two models diverge in their errors. The higher the Statistic value, the greater the divergence between the models' error profiles. The p-value reflects the probability that the observed difference between b and c arose by chance, rather than due to a real difference between the models. If the p-value is less than 0.05, the divergence between the models is considered statistically significant, meaning it is not accidental. The results of the comparison of air quality classification models based on threshold values are presented in Table 5.

Table 5. McNemar Test Results for Threshold-Based Air Quality State Classification Models

| **Model Pair** | **b** | **c** | **Statistic** | **p-value** | **Significant difference?** |
|---|---|---|---|---|---|
| Rule-based baseline vs Logistic Regression | 130 | 0 | 136.01 | ~0.000 | Yes |
| Rule-based baseline vs Decision Tree | 0 | 0 | 0.00 | 1.000 | No |
| Rule-based baseline vs Random Forest | 13 | 0 | 11.08 | 0.0009 | Yes |
| Logistic Regression vs Decision Tree | 0 | 138 | 136.01 | ~0.000 | Yes |
| Logistic Regression vs Random Forest | 13 | 138 | 101.83 | ~0.000 | Yes |
| Decision Tree vs Random Forest | 13 | 0 | 11.08 | 0.0009 | Yes |

The results of the McNemar's test confirmed that the basic threshold model and the decision tree produced identical predictions on the test sample. The absence of discrepancies between them is consistent with the earlier explanation that the decision tree reproduces the threshold logic of the classification. At the same time, statistically significant differences between other pairs of models indicated that logistic regression and random forest produce different error profiles.

Following the evaluation of threshold classification methods, the results of detecting abnormal changes in sensor readings were examined. Unsupervised learning methods were used to detect anomalies, the results being evaluated according to their correspondence with threshold anomaly labels generated based on real sensor data. Unsupervised learning models were used to detect anomalies: isolation forest, local outlier factor, local outlier factor following dimensionality reduction via principal component analysis, and one-class SVM. Their results were evaluated not as confirmation of the true nature of the anomalies, but as the degree of agreement with the threshold marking of short-term changes in sensor readings. The primary purpose of these models was to identify objects that differ from the general data structure. Therefore, they were evaluated based on the correspondence of their results to anomaly thresholds established based on real sensor data. The evaluation results are presented in Table 6.

Table 6. Results of Anomaly Detection Models Against Rule-Based Labels

| **Model** | **F1** | **Precision** | **Recall** | **Number of Detected Anomalies** | **Training time** | **Inference time** |
|---|---|---|---|---|---|---|
| Isolation Forest | 0.548 | 0.389 | 0.927 | 131 | 73 ms | 11 ms |
| Local Outlier Factor | 0.152 | 0.109 | 0.255 | 129 | 90 ms | 73 ms |
| Local Outlier Factor + PCA | 0.146 | 0.102 | 0.255 | 137 | 125 ms | 109 ms |
| One-Class SVM | 0.489 | 0.324 | 1.000 | 170 | 73 ms | 55 ms |

A summarised relationship between the F1 score and the training time of anomaly detection models using threshold labels is shown in Fig. 3. The obtained results showed that different anomaly detection methods reproduce the threshold logic for labelling anomalous changes in sensor data in different ways, confirming the need for their comparative evaluation when selecting a method for a specific monitoring system.

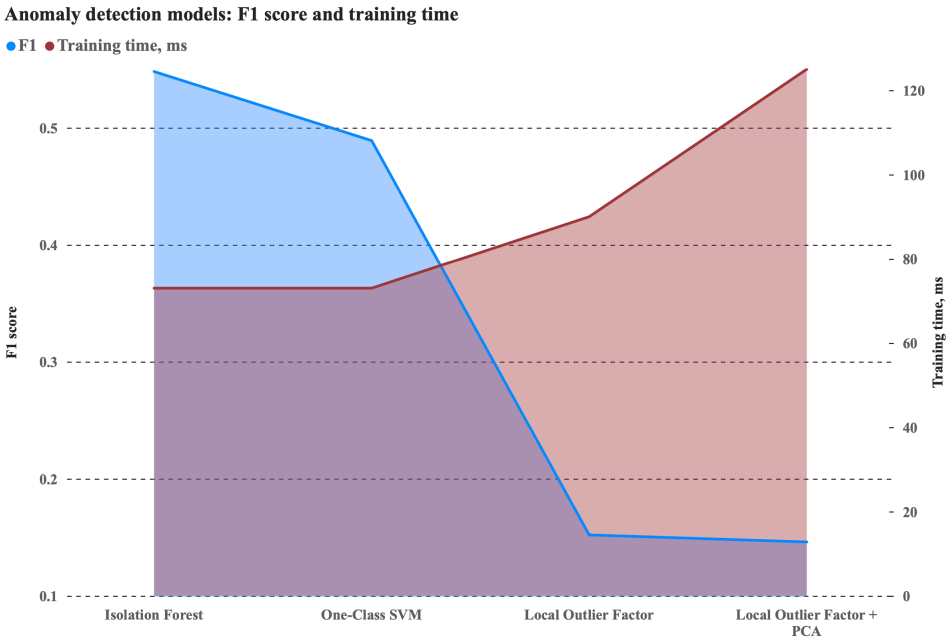


Fig. 3. F1 score and training time of anomaly detection models against rule-based labels.

The isolation forest achieved the highest F1 score among the considered models, indicating that its results best correspond to the rule-based labels. At the same time, a Precision of 0.389 indicated a significant number of false positives. One-Class SVM achieved a Recall of 1.000, meaning it did not miss a single micro-batch marked as abnormal according to the threshold rules. However, the low Precision value indicated an excessive number of false positives. The local outlier factor, both in the full feature space and after dimensionality reduction using principal component analysis, showed the lowest F1 scores, indicating a weaker correspondence of these models to the threshold-based anomaly logic.

Separately, outside the scope of the main evaluation based on threshold labels, an additional sensitivity check of the models was conducted on a subset of real micro-batches with clearly expressed deviations in the feature space. This subset included 66 of 3319 micro-batches in the range of which at least one feature's value exceeded three standard deviations of the corresponding feature across the entire sample. This test made it possible to assess whether the models can detect statistically significant deviations that do not necessarily correspond to the anomaly thresholds. The results of this assessment are presented in Table 7.

Table 7. Results of Anomaly Detection Models on Statistically Extreme Feature Batches

| Model | F1 | Precision | Recall | Number of Detected Anomalies |
|---|---|---|---|---|
| Isolation Forest | 0.040 | 0.030 | 0.061 | 132 |
| Local Outlier Factor | 0.468 | 0.317 | 0.894 | 186 |
| Local Outlier Factor + PCA | 0.236 | 0.166 | 0.409 | 163 |
| One-Class SVM | 0.114 | 0.078 | 0.212 | 180 |

A comparison of the results presented in Tables 6 and 7 showed that the choice of anomaly detection method depends significantly on the type of anomaly to be detected. If the monitoring system is designed to detect changes that meet the threshold logic for flagging, it is more appropriate to use isolation forest. However, when the priority is to detect statistically significant local deviations in the feature space, local outlier factor is preferable. These findings underscore that the task of detecting anomalies in sensor data does not have a single universal solution, and the choice of method must be determined by the specific requirements of the monitoring system.

**Conclusions and prospects for further research.** In this study, a method for analysing air quality based on micro-batch formation of a statistical feature set was developed and validated, covering two practical tasks: determining air quality status based on threshold values and detecting abnormal changes in sensor data. The experimental system collected data from SCD41, BME688 and SPS30 sensors, stored

measurements locally in DuckDB, and formed micro-batches comprising 10 consecutive readings. Based on 33,190 raw measurements, 3,319 micro-batches were generated, each described by 60 statistical features.

Verification of the method showed that threshold classification of air quality can be implemented using ML at the edge level. The most balanced method for this task turned out to be random forest, which achieved an F1 score of 0.857 in cross-validation and 0.980 on the test set. The decision tree fully reproduced the threshold logic for identifying micro-batches with potentially poor air quality, but this result should be interpreted not as an advantage of generalization, but because of the model structure matching the character of the formed rules.

Analysis of the threshold classification results confirmed that the choice of method must consider not only integral quality metrics but also the type of errors. Logistic regression more frequently produced false positives, whereas the random forest had no false positives but missed 13 micro-batches identified as potentially unfavourable according to the threshold rules. In early warning tasks regarding deteriorating air quality, the choice of classification method should be determined by which errors are more critical for a specific monitoring system: missed instances of adverse conditions or false positives.

Verification of the method for detecting abnormal changes showed that none of the methods considered is universal. Isolation forest is best suited to the threshold logic of anomaly labelling and achieved the highest F1 score for rule-based labels. Local outlier factor, on the other hand, proved to be more sensitive to statistically significant local deviations in the feature space. This confirms that the choice of method for detecting anomalies in sensor data should be determined by the specific type of anomaly that the monitoring system is required to detect.

An evaluation of practical performance confirmed that both tasks can be implemented within a single monitoring system without placing a critical load on the peripheral node. The training time for all methods did not exceed 500 ms, the evaluation time for classification methods did not exceed 4.2 ms, and for anomaly detection methods did not exceed 109 ms. Peak RAM usage was approximately 240 MB. Overall, the results obtained indicate that the choice of a specific method for each task should be determined not only by the F1 score, but also by the type of errors, the character of the anomalies, and the requirements of the specific monitoring system.

The scientific novelty of the study lies in the proposed procedure for air quality analysis based on the formation of statistical feature sets from short sequences of sensor measurements. The procedure combines threshold-based air quality condition classification with anomaly detection and evaluates abnormal changes according to two criteria: agreement with threshold-based labels and sensitivity to statistical deviations in the feature space. Unlike approaches in which classification and anomaly detection are considered separately, the proposed procedure supports their joint use within an autonomous monitoring system under edge computing constraints.

A further direction of research is the extension of the proposed air quality analysis method towards continuous operation in an environmental monitoring system. Future work will focus on integrating threshold-based air quality condition classification and anomaly detection into a single analytical workflow implemented directly on Raspberry Pi 5. This will make it possible to assess the current air quality condition based on threshold values for PM2.5, PM10, $CO_2$, temperature and relative humidity, while also detecting atypical deviations in the dynamics of sensor readings.

**References**

1. Garcia A., Saez Y., Harris I., Huang X., Collado E. Advancements in air quality monitoring: a systematic review of IoT-based air quality monitoring and AI technologies // Artificial Intelligence Review. 2025. Vol. 58. Art. 275. DOI: 10.1007/s10462-025-11277-9.

2. Andriulo F. C., Fiore M., Mongiello M., Traversa E., Zizzo V. Edge Computing and Cloud Computing for Internet of Things: A Review // Informatics. 2024. Vol. 11. Art. 71. DOI: 10.3390/informatics11040071.

3. Godasiaei S. H., Ejohwomu O. A., Zhong H., Booker D. Integrating experimental analysis and machine learning for enhancing energy efficiency and indoor air quality in educational buildings // Building and Environment. 2025. Vol. 276. Art. 112874. DOI: 10.1016/j.buildenv.2025.112874.

4. Lopes S. I., Orłowski C., Branco P. T. B. S., Karatzas K., Villena G., Saffell J., Marques G., Sousa S. I. V., Lenartz F., Bergmans B. et al. Low-Cost Sensor Systems and IoT Technologies for Indoor Air Quality Monitoring: Instrumentation, Models, Implementation, and Perspectives for Validation // Sensors. 2025. Vol. 25, no. 24. Art. 7567. DOI: 10.3390/s25247567.

5. Alongi A., Pacileo L., Shahrabani M. M. N., Spūdys P., Scoccia R., Mazzarella L. Smart sensors for Indoor Environmental Quality in residential smart buildings: a review // International Journal of Sustainable Energy. 2025. Vol. 44, no. 1. Art. 2578592. DOI: 10.1080/14786451.2025.2578592.

6. Mota A., Serôdio C., Briga-Sá A., Valente A. Implementation of an Internet of Things Architecture to Monitor Indoor Air Quality: A Case Study During Sleep Periods // Sensors. 2025. Vol. 25. Art. 1683. DOI: 10.3390/s25061683.

7. Morawska L., Asbach C., Patel H. Application of PM2.5 low-cost sensors for indoor air quality compliance monitoring // Aerosol Science and Technology. 2025. Vol. 59, no. 10. P. 1210–1220. DOI: 10.1080/02786826.2025.2457326.
8. Kahukina A., Patseva I. Assessment and forecast of atmospheric pollutant dynamics in the urban ecosystem of Zhytomyr // Technology Audit and Production Reserves. 2025. Vol. 2, no. 3(82). P. 36–42. DOI: 10.15587/2706-5448.2025.326893.
9. Moursi A. S., El-Fishawy N., Djahel S., Shouman M. A. An IoT enabled system for enhanced air quality monitoring and prediction on the edge // Complex & Intelligent Systems. 2021. Vol. 7. P. 2923–2947. DOI: 10.1007/s40747-021-00476-w.
10. Wardana I. N. K., Gardner J. W., Fahmy S. A. Optimising Deep Learning at the Edge for Accurate Hourly Air Quality Prediction // Sensors. 2021. Vol. 21. Art. 1064. DOI: 10.3390/s21041064.
11. Kulkarni V., Lakshmi A. S., Lakshmi C. B. N., Panneerselvam S., Kanan M., Flah A., Elnaggar M. F. Air Quality Decentralized Forecasting: Integrating IoT and Federated Learning for Enhanced Urban Environmental Monitoring // Engineering, Technology & Applied Science Research. 2024. Vol. 14, no. 4. P. 16077–16082. DOI: 10.48084/etasr.7869.
12. Cruz P., Achir N., Carneiro Viana A. On the Edge of the Deployment: A Survey on Multi-access Edge Computing // ACM Computing Surveys. 2022. Vol. 55, no. 5. Art. 99. DOI: 10.1145/3529758.
13. Kong L., Tan J., Huang J., Chen G., Wang S., Jin X. et al. Edge-computing-driven Internet of Things: A Survey // ACM Computing Surveys. 2023. Vol. 55, no. 8. Art. 174. DOI: 10.1145/3555308.
14. Reis M. J. C. S. Lightweight Signal Processing and Edge AI for Real-Time Anomaly Detection in IoT Sensor Networks // Sensors. 2025. Vol. 25. Art. 6629. DOI: 10.3390/s25216629.
15. Farrel G. E., Yahya W., Basuki A., Amron K., Siregar R. A. Scalable Edge Computing Cluster Using a Set of Raspberry Pi: A Framework // Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology (SIET 2023). Badung, Bali, Indonesia, 24–25 October 2023. DOI: 10.1145/3626641.3626936.
16. Okano M. T., Lopes W. A. C., Ruggero S. M., Vendrametto O., Fernandes J. C. L. Edge AI for Industrial Visual Inspection: YOLOv8-Based Visual Conformity Detection Using Raspberry Pi // Algorithms. 2025. Vol. 18. Art. 510. DOI: 10.3390/a18080510.
17. Palamar A., Karpinski M., Palamar M., Osukhivska H., Mytnyk M. Remote Air Pollution Monitoring System Based on Internet of Things // Proceedings of the 2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2022). 2022. P. 98–106. URL: https://ceur-ws.org/Vol-3309/paper14.pdf.
18. Borodii I., Osukhivska H. Research on the efficiency of data loading and storage in Data Lakehouse architectures for the formation of analytical data systems // Information Technology: Computer Science, Software Engineering and Cyber Security. 2025. No. 4. P. 28–36. DOI: 10.32782/IT/2025-4-4.
19. Fedorovych I., Osukhivska H., Lutsyk N. Performance Benchmarking of Continuous Processing and Micro-Batch Modes in Spark Structured Streaming // Proceedings of the 4th International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2024). 2024. P. 80–90. URL: https://ceur-ws.org/Vol-3896/paper5.pdf.
20. International WELL Building Institute. Air quality standards // WELL Feature Library - v2. URL: https://standard.wellcertified.com/v2/air/air-quality-standards.
21. ANSI/ASHRAE Standard 62.1-2022. Ventilation and Acceptable Indoor Air Quality. Peachtree Corners : ASHRAE, 2022. 90 p. ISSN 1041-2336.
22. Jokl M. V. Thermal Comfort and Optimum Humidity Part 1 // Acta Polytechnica. 2002. Vol. 42, no. 1. P. 12–24. DOI: 10.14311/302.

Історія статті:
Отримано: 19.05.2026 Доопрацьовано: 22.04.2026 Прийнято до друку: 23.05.2026 Опубліковано: 29.05.2026