Journal of Agricultural Faculty of Gaziosmanpasa University
Gaziosmanpaşa Üniversitesi Ziraat Fakültesi Dergisi
https://dergipark.org.tr/tr/pub/gopzfd

Research Article

JAFAG (2026) 43(1), 83-93
ISSN: 1300-2910
E-ISSN: 2147-8848
DOI: 10.55507/gopzfd.1681251

Development of a low-cost IoT and machine learning-based air quality monitoring system for dairy barns

Yasemin Betül ÇİÇEK[1], Umut MUCAN[1], Ünal KIZIL[1*]

[1]Çanakkale Onsekiz Mart Üniversitesi, Ziraat Fakültesi, Tarımsal Yapılar ve Sulama Bölümü, Çanakkale, Türkiye

***Corresponding author e-mail:** unal@comu.edu.tr

Received: 21.04.2025 Accepted: 29.11.2025

Abstract:
Advances in Internet of Things (IoT) and machine learning have enabled the development of cost-effective systems for real-time air quality monitoring in livestock environments. In this study, a low-cost Internet of IoT based device that can be used for air quality monitoring and evaluation in dairy barns was developed. Various machine learning algorithms were used to model the sensor data. The BME 680 air quality sensor data were collected in the ThingSpeak cloud database, and subsequent analysis was conducted. The performance of Random Forest (RF), Support Vector Regression (SVR), Gradient Boosting (GB), k-nearest neighbors (KNNs), Decision Trees (DTs), and Artificial Neural Networks (ANNs) algorithms was tested. The RF and GB algorithms yielded the best results in estimating air quality index values. However, it was stated that the performance of the ANN algorithm should be re-evaluated using large-volume datasets. In addition, it was emphasized that incorporating additional environmental parameters could improve the system's performance. As a result, it was shown that low-cost digital agricultural applications can be developed using the capabilities of current technology.

Keywords Digital agriculture, environmental data modeling, IoT-based air quality monitoring, machine learning algorithms.

Süt sığırcılığı ahırları için düşük maliyetli Iot ve makine öğrenmesine dayalı bir hava kalitesi izleme sisteminin geliştirilmesi

Öz:
Nesnelerin İnterneti (IoT) ve makine öğrenimi alanındaki gelişmeler, hayvancılık yapılan alanlarda gerçek zamanlı hava kalitesi izlemesi için uygun maliyetli sistemlerin geliştirilmesine imkan sağlamıştır. Bu çalışmada, süt sığırcılığı işletmelerinde hava kalitesinin izlenmesi ve değerlendirilmesinde kullanılabilecek düşük maliyetli bir (IoT) tabanlı cihaz geliştirilmiştir. Ayrıca, toplanan verilerin modellenmesinde farklı makine öğrenmesi algoritmaları kullanılmıştır. Veri seti ağırlıklı olarak BME 680 hava kalitesi sensöründen elde edilen verilere dayanmaktadır. Sensör verileri ThingSpeak bulut veritabanında toplanmış ve ardından ileri analizler gerçekleştirilmiştir. Rastgele Ormanlar (RF), Destek Vektör Regresyonu (SVR), Artırmalı Öğrenme (GB), En Yakın Komşular (KNNs), Karar Ağaçları (DTs) ve Yapay Sinir Ağları (ANNs) algoritmalarının performansları test edilmiştir. Hava kalitesi indeks değerlerinin tahmininde en iyi sonuçlar RF ve GB algoritmalarıyla elde edilmiştir. Bununla birlikte, ANNs algoritmasının performansının daha büyük hacimli veri kümeleriyle yeniden değerlendirilmesi gerektiği belirtilmiştir. Ayrıca, sisteme diğer çevresel parametrelerin dahil edilmesinin performansı artırabileceği vurgulanmıştır. Sonuç olarak, günümüz teknolojisinin sunduğu imkânlarla düşük maliyetli dijital tarım uygulamalarının geliştirilebileceği ortaya konmuştur.

Anahtar kelimeler: Dijital tarım, çevresel veri modelleme, IoT tabanlı hava kalitesi izleme, makine öğrenme algoritmaları,

1. Introduction

In recent years, technological advances have led to revolutionary developments across nearly every field, including agriculture and animal husbandry. The use of technological developments to improve animal welfare and worker health is a high priority. However, the environmental conditions in livestock barns to which workers are exposed are considered inadequate. More than 250,000 workers are

To cite this article: Çiçek, Y. B., Mucan, U., & Kızıl, Ü. (2026). Development of a low-cost iot and machine learning-based air quality monitoring system for dairy barns. *Journal of Agricultural Faculty of Gaziosmanpasa University*, 43(1), 83-93. https://doi.org/10.55507/gopzfd.1681251

employed in animal husbandry enterprises alone in the United States. It has been determined that high ammonia and carbon dioxide levels in barns affect the health of both animals and workers. For example, Kilic and Yaslioglu (2014) reported that ammonia (NH_3) levels in the summer months were as high as 8 ppm, and carbon dioxide (CO_2) levels reached 732 ppm.

In addition to NH_3 and (CO_2), another pollutant that should be monitored is particulate matter (PM2.5 and PM10). It has been shown that elevated particulate matter levels in animal barns may pose health risks to animals and workers (Shen et al., 2019). Some studies have reported that NH_3 emissions from barns have a major impact on the formation of particulate matter (PM2.5) and constitute a major component of it (Kim et al., 2024). Therefore, it is essential to maintain air quality using modern technologies. In this context, sensors, the Internet of Things (IoT), and cloud data systems can enable highly useful solutions. In IoT-based monitoring systems, sensor data can be converted into a real-time air quality index (AQI) and presented in an easily understandable format (Truong et al., 2021). In addition, by monitoring parameters such as PM, CO_2, temperature, and humidity with IoT devices and sending the data to a cloud-based system, air quality automation systems can be developed (Taştan & Gökozan, 2019). In their study, Doğan and Özyurt (2025) developed weather forecasts using SVM, KNN, LSTM, and XGBoost algorithms, based on approximately 600,000 meteorological data points obtained from IoT-based sensors. They reported that LSTM (99%) and XGBoost (100%) achieved the highest accuracy.

Microcontrollers are used extensively in such technologies. Microcontrollers provide cost-effective solutions for data measurement, processing, and transmission. In addition, these modules, which can be integrated with OLED displays, allow the development of very useful devices (Babiuch et al., 2020; Haroon P S et al., 2024). It has been reported that microcontrollers are also suitable for developing educational prototypes in robotic applications that use Bluetooth and Wi-Fi (Netzahual et al., 2019).

Open-source, cloud-based data collection and analysis systems used in IoT platforms, such as ThingSpeak, facilitate prototype development. These platforms, which store, analyze, and visualize data collected from various sensors, are widely employed, particularly in environmental monitoring applications such as air quality monitoring. Such platforms enable users to readily assess air quality by presenting data in a graphical interface in real time (Kelechi et al., 2022). When air quality exceeds a specified threshold, automated email and SMS alerts can be sent (Murad et al., 2021). Hence, the aim of this study is to develop a prototype air quality monitoring and data collection system using ThingSpeak, a cloud-based platform that provides data collection, processing, analysis, and automation with a low-cost air quality sensor. It is also aimed at estimating air quality sensor values by modeling indoor temperature and relative humidity using major machine learning algorithms.

2. Materials and Methods

2.1. Experimental dairy barn

The developed prototype device was tested at the Aşkın Dairy Cattle Farm in Dümrek village, Çanakkale Province. The farm housed 80 milking Simmental cows, and the barn was designed for a free-stall system. The developed device was mounted in the middle of the barn as much as possible, in a position where the animals cannot reach it and where air currents will not directly contact the device (Figure 1). Since the barn is a small structure with a floor area of approximately 25×40 m2, only one prototype was developed. Furthermore, due to the principle that heated polluted air expands and rises, measurements were taken from only one level.

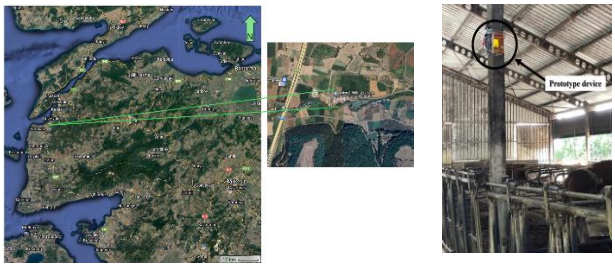

Figure 1. Location of the experimental barn and the device

2.2. Prototype design

The I2C BME680 (Bosch Sensortec GmbH, Reutlingen, Germany) sensor was used to determine the air quality index values. Due to its compact structure and high accuracy, it is preferred in environmental data collection studies. This environmental sensor is used in modeling and other

academic studies because it can measure parameters such as temperature, humidity, and barometric pressure, as well as air quality index values. It is easily connected to microcontrollers via the I2C protocol. It is also preferred in portable devices due to the low cost and low energy requirements (Zhang et al., 2022).

This sensor helps to comment on air quality by expressing the cumulative effect of pollutants, gases and volatile organic compounds (VOCs) affecting air quality in a single index value. Using the obtained index values, 6 air quality classes were determined, and these values are given in Table 1.

Table 1. Air quality index values and corresponding quality classes

Air quality index	Air quality class
0-50	Good (1)
51-100	Average (2)
101-150	Somewhat bad (3)
151-200	Bad (4)
201-300	Worse (5)
301-500	Very bad (6)

The ESP32 (Espressif, Shanghai, China) microcontroller was used to transmit the sensor data to the ThingSpeak platform. This processor is a low-cost, high-performance microcontroller board and is widely used in IoT projects. The dual-core Tensilica LX6 processor supports Wi-Fi and Bluetooth Low Energy (BLE) connectivity (Hercog et al., 2019). To ensure communication between the ESP32 and the digital display, the code was written in the Arduino Integrated Development Environment (IDE), which uses the C/C++ programming languages. The code was then uploaded to the ESP32 microcontroller via a USB connection. A Nextion NX4832T035_011 (ITEAD Intelligent Systems Co., Ltd., Shenzhen, China) touchscreen digital display was integrated into the device to enable simultaneous data monitoring. This 3.5-inch screen is widely preferred in HMI (Human-Machine Interface) projects. It can be used for both data visualization and sending data or commands to the system, which is integrated with a touchscreen. The screen's user interface can be easily designed and tested using Nextion Editor software. The device transfers data via Wi-Fi connection. There is a Wi-Fi connection in the experimental barn where the device is installed. However, due to the distance between the modem providing internet and the device, data loss sometimes occurred. To address this problem, a mobile phone was placed in the box in which the device was integrated. Data transfer was enabled via the mobile phone's internet connection. The developed prototype is shown in Figure 2.

2.3. Cloud data storage platform – thingspeak

ThingSpeak is a data collection, analysis, and visualization platform developed by MathWorks that offers cloud-based solutions for IoT projects. This platform uses protocols such as MQTT, HTTP, and REST API in data transmission. It also supports MATLAB integration, enabling certain analyses to be performed on the platform. ThingSpeak also provides cost-effective and efficient automation capabilities by enabling multiple devices to communicate with one another (Dang et al., 2020). The platform, which offers free service for up to five channels, generates 2 API keys that it uses to send and receive data, ensuring reliable data transmission. By referencing these keys in the necessary coding, communication can be established between the device and ThingSpeak. In addition, the data recorded at the specified time interval can be plotted as graphs on the platform. By examining these graphs, it is possible to determine whether data communication is functioning properly and whether the device is experiencing malfunctions.

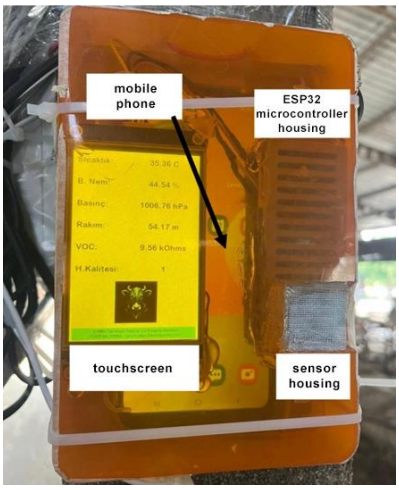

Figure 2. The prototype device and its components

2.4. Cloud data storage platform – thingSpeak

ThingSpeak is a data collection, analysis, and visualization platform developed by MathWorks that offers cloud-based solutions for IoT projects. This platform uses protocols such as MQTT, HTTP, and REST API in data transmission. It also supports MATLAB integration, enabling certain analyses to be

performed on the platform. ThingSpeak also provides cost-effective and efficient automation capabilities by enabling multiple devices to communicate with one another (Dang et al., 2020). The platform, which offers free service for up to five channels, generates 2 API keys that it uses to send and receive data, ensuring reliable data transmission. By referencing these keys in the necessary coding, communication can be established between the device and ThingSpeak. In addition, the data recorded at the specified time interval can be plotted as graphs on the platform. By examining these graphs, it is possible to determine whether data communication is functioning properly and whether the device is experiencing malfunctions.

2.5. Machine learning algorithms used

In the study, the objective was to model the air quality sensor's index values using hourly data on temperature, relative humidity, and atmospheric pressure recorded by the device. In this context, the performance of the 6 most used machine learning algorithms was tested. The algorithms used were Random Forest (RF), Support Vector Regression (SVR), Gradient Boosting (GB), K-Nearest Neighbors (KNN), Decision Trees (DT), and Artificial Neural Networks (ANN). RF, an ensemble learning algorithm, combines multiple decision trees to improve estimation accuracy and mitigate overfitting. It works by aggregating predictions from various trees through majority voting or averaging. The most important advantage of the RF algorithm is its robustness to noise. It also yields strong results on large datasets (Ao et al., 2019). SVR is a type of Support Vector Machine (SVM) used for regression. It constructs a hyperplane in a high-dimensional space by estimating continuous values to minimize estimation error. SVR is effective at detecting nonlinear relationships using kernel functions on small to medium-sized datasets (Deniz, 2024).

GB sequentially builds models and optimizes the loss function by adding new models to correct errors from previous models. It is widely used in structured data problems, offering high accuracy and the ability to capture nonlinear relationships (Singh, 2022).

KNNs estimate a target by considering the K-nearest data points in the feature space. It is a simple, example-based learning algorithm. This algorithm, which makes no assumptions about data distribution, is nonparametric. Although it exhibits strong estimation performance, particularly on small datasets, its most significant disadvantages are its high computational demands and poor performance on multidimensional data (Joshi et al., 2023).

DTs are commonly used as the base learner in ensemble learning methods such as RF and GB. It divides the dataset into subsets based on feature values , following a tree-like structure in the estimation process. Although this algorithm is intuitive and easy to interpret, it can reduce prediction accuracy due to overfitting on noisy data (Le Fort, 2018). ANN, an important class of machine learning algorithms, was developed by mimicking biological neural networks. It is highly effective at detecting nonlinear, complex relationships. It uses activation functions when transforming data input features. They also consist of interconnected layers of nodes (neurons). They yield higher performance in image recognition, natural language processing and other areas of use. However, they require large data volumes for the trained model to make accurate predictions. This may require high-capacity processors for calculations (Mahammad Rafi et al., 2023).

2.6. Data processing and modeling

The data processing was performed on Google Colab, a cloud-based platform provided by Google. This platform uses Python for calculations. It also provides access to the necessary machine learning libraries. The data was prepared in MS Excel format, uploaded to Google Colab and the necessary processing and analysis were performed.

First, principal component analysis (PCA) was performed. PCA is a statistical technique widely used for dimension reduction and feature extraction in data analysis. The purpose of this analysis is to transform the original variables into smaller, independent components called principal components. The complexity of these data is reduced, and the maximum variance in the data set is captured (Jolliffe 2002).

Seventy percent of the data obtained in the modeling phase was used for model development (training), and the remaining 30% was used to test model performance. Hyperparameter tuning was performed by evaluating different configurations to identify the parameter that yields the highest coefficient of determination (R2), the lowest root

mean square error (RMSE), and the lowest mean absolute error (MAE) for each algorithm.

3. Results and Discussion

Hourly measurements of temperature, relative humidity, atmospheric pressure, air quality index, and corresponding air quality class were collected between July and December 2024. Data could not be collected during certain periods due to power outages at the enterprise. A total of 2744 lines of data were recorded for modeling. There is a linear relationship between NH_3 emissions and ambient temperature in barns. Studies have shown that a 1 °C increase in barn temperature is associated with a 1.47 g increase in NH_3 emissions (Sanchis et al., 2019). A similar relationship has been demonstrated between temperature and CO_2 (Hempel et al., 2016). High relative humidity creates a favorable environment for microorganism reproduction and the decomposition of organic matter (Kaya, 2007). An increase in the microbial population results in greater urea hydrolysis by the urease enzyme. This increases the NH_3 concentration within the barn (Toprak et al., 2016). In addition, microbial respiration accelerates with increasing relative humidity, thereby increasing CO_2 emissions (Gough, 2011). As shown, temperature and relative humidity significantly affect air quality. Other factors affecting air quality in barns include ventilation, animal density, inadequate manure cleaning and bedding materials used. However, because all these factors are continuous, it is difficult to determine their hourly changes and to treat them as parameters in the model. Yet their effects can be determined by observing changes in intensity and amount over the long term. Therefore, in this study, the effectiveness of temperature and relative humidity in determining air quality index values was primarily investigated. In addition, the sensor also measures ambient barometric pressure. Therefore, it was considered a third parameter to determine whether it affects modeling. The PCA analysis was performed by considering these 3 basic parameters. Of the 3 basic components calculated, those with a total variance explained percentage of 95% or higher were used in the modeling. The variance explanation rate of each component is given in Table 2.

The first two principal components explain 98% of the total variance. Therefore, they were used as inputs in the models. Necessary hyperparameters should be tuned to optimize the performance of all algorithms. In hyperparameter tunings the GridSearchCV method offered by the Scikit-learn library was used. It optimizes the model's performance by evaluating all combinations of hyperparameters within specified limits.

Table 2. Variance explanation ratios of principal components

	PC1	PC2	PC3
Proportion of explained variance	0.76	0.22	0.02

PC: principal component

One of the basic parameters of RF is the number of trees in the forest. Another important parameter is the maximum depth, which indicates how deeply each tree will grow. Limiting the depth to values such as 10, 20, or 30 in the model prevents overfitting, whereas specifying None allows trees to grow to a depth less than the minimum number of samples per split. The number of samples required to split an internal node is determined by the minimum number of samples required to split the parameters. The minimum samples leaf parameter is used to determine the number of samples in each leaf node. The high values of the minimum sample split and minimum sample leaf parameters prevent overfitting. The maximum features parameter determines how many features are used to split at each node. Among the alternatives, 'auto' uses all the features, while 'sqrt' and 'log2' select a subset. The last two options make training faster and reduce overfitting (Yuliana, 2024). The hyperparameters used for RF, the tested values, and the selected ones are given in Table 3.

Table 3. Random Forest hyperparameters and selected values

Parameter	Tested value	Selected value
n_estimators	50, 100, 200, 300, 400, 500	500
max_depth	None, 10, 20, 30, 40, 50	20
min_samples_split	2, 4, 6, 8, 10	2
min_samples_leaf	2, 4, 6, 8, 10	2
max_features	auto, sqrt, log2	sqrt

The RMSE values calculated for the training and test data were 5.62 and 9.72, respectively; the MAE

values were 2.93 and 5.15, respectively. The R2 values are given in Figure 3.

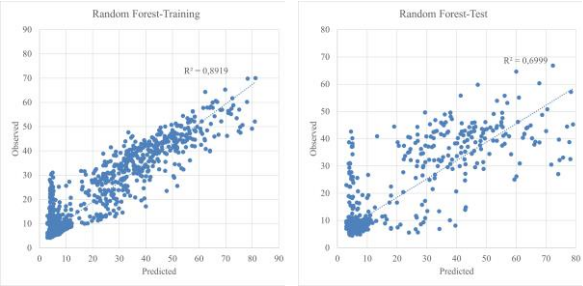

Figure 3. Performance of training and test phases of the Random Forest algorithm.

One of the most important parameters of the SVR algorithm is the kernel, which ensures that the data can be linearly separable (Ma et al., 2015). The linear option yields a linear decision boundary; poly yields polynomial relationships; rbf yields radial basis functions; and sigmoid yields sigmoid-based transformations. Another important hyperparameter, C, balances the reduction in error margin during the model's training phase with the need to maintain a simpler decision boundary. While lower values provide simplicity, higher values minimize error (Wu et al., 2013). Gamma is a parameter used to control the influence of individual data points in nonlinear kernels such as RBF and poly. While scale automatically adjusts gamma according to the number of features, auto uses the inverse of the number of samples. The degree parameter determines the polynomial degree of polynomial kernels. The Epsilon parameter determines the margin of error in the estimates. Small values allow the model to capture fine details, whereas large values ignore small deviations (Han et al., 2012). The parameters used in hyperparameter tuning for the SVR algorithm, along with the values tested, are presented in Table 4.

Table 4. Support Vector Regression hyperparameters and selected values

Parameter	Tested value	Selected value
Kernel	linear, poly, rbf, sigmoid	rbf
C	0.1, 1, 10, 100	100
Gamma	scale, auto	auto
Degree	2, 3, 4, 5	2
Epsilon	0.01, 0.1, 0.2, 0.5	0.5

C: penalty parameter

The RMSE values for the training and test data were 11.81 and 12.53, respectively; the MAE values were 6.23 and 6.54, respectively. The R2 values are given in Figure 4.

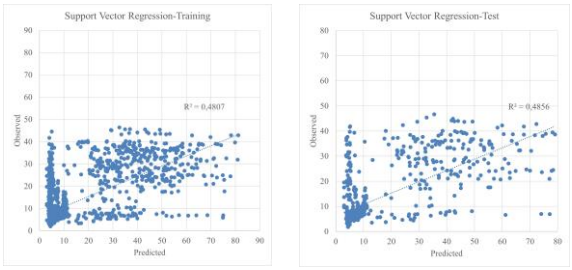

Figure 4. Performance of training and test phases of the Support Vector Regression algorithm.

The parameters of the Gradient Boosting algorithm are similar to those of the RF algorithm (Yuliana, 2024). In addition, the learning rate parameter adjusts the contribution of each tree to the model. Smaller values require more trees while improving the model's generalization ability. The subsample parameter specifies the proportion of the randomly selected training data used to train each tree in the ensemble (Duroux & Scornet, 2018). The parameters used in the hyperparameter tuning of the GB algorithm, along with the values tested, are presented in Table 5.

The RMSE values calculated for the training and test data were 5.97 and 10.35, respectively; the MAE values were 3.43 and 5.56, respectively. The R2 values given in Figure 5.

Figure 5. Performance of training and test phases of the Gradient Boosting algorithm.

One of the most important parameters used in the KNNs algorithm is n neighbors. This parameter indicates the number of neighbors to be considered in estimating a sample. Choosing small values causes overlearning, whereas large values reduce the model's sensitivity. The weights parameter determines the contribution of each neighbor to the estimate during the estimation phase. If a uniform is used, the multiple rates of all neighbors are taken

equally. In the distance option, the contribution rate varies with the neighbor's proximity. The algorithm parameter specifies the algorithm used to identify neighbors. The auto option automatically selects the method used to find neighbors, whereas ball tree and kd tree are preferred for multidimensional data. The brute-force method is effective for small datasets. The leaf size parameter is used to determine the leaf size for a ball tree or a KD tree. Small values increase the calculation time. The parameter p, which is the nearest parameter, determines the method used for distance calculation. If this value is selected as 1, the Manhattan distance is used; if selected as 2, the Euclidean distance is used (Scikit-learn, 2024). The hyperparameter parameters used in hyperparameter tuning for the K-Nearest Neighbors algorithm, along with the values tested, are presented in Table 6.The RMSE values calculated for the training and test data were 0.03 and 9.06, respectively; the MAE values for training and test data were 0.00 and 4.71, respectively. The R2 values are given in Figure 6.

Table 5. Gradient Boosting hyperparameters and selected values

Parameter	Tested value	Selected value
n_estimators	50, 100, 200, 300, 400, 500	50
Learning_rate	0.01, 0.05, 0.1, 0.2, 0.3	0.1
Max_depth	3, 4, 5, 6, 7, 8	7
Min_samples_split	2, 4, 6, 8, 10	2
Min_samples_leaf	1, 2, 4, 6, 8	2
Subsample	0.6, 0.8, 1.0	0.6
Max_features	auto, sqrt, log2, None	sqrt

Table 6. K-Nearest Neighbors hyperparameters and selected values

Parameter	Tested value	Selected value
n_neighbors	3, 5, 7, 9	7
Weights	uniform, distance	distance
Algorithm	auto, ball_tree, kd_tree, brute	brute
Leaf_size	20, 30, 40, 50	20
p	1, 2	2

The criterion parameter in the DT algorithm specifies the statistical method used to evaluate the success of the splits. The alternatives are squared error (mean square error), Friedman mse (MSE optimized for gradient boosting), absolute error (mean absolute error), and Poisson (for numerical data). The splitter parameter determines whether the split is optimized for accuracy or speed; it can take the values best or random. The max leaf nodes parameter determines the maximum number of leaf nodes to be in the tree (Scikit-learn, 2024). Other parameters are explained above (Yuliana, 2024). The parameters used in hyperparameter tuning for the DT algorithm, along with the values tested, are presented in Table 7.

Table 7. Decision Trees hyperparameters and selected values

Parameter	Tested value	Selected value
Criterion	squared_error, friedman_mse, absolute_error, poisson	poisson
Splitter	best, random	random
Max_depth	None, 10, 20, 30, 40	20
Min_samples_split	2, 5, 10	10
Min_samples_leaf	1, 2, 4, 6	2
Max_features	None, sqrt, log2	None
Max_leaf_nodes	None, 10, 20, 50	None

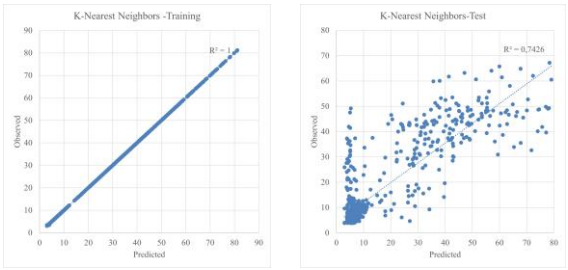

Figure 6. Performance of the training and test phases of the K-Nearest Neighbors algorithm.

The RMSE values for the training and test data were 8.49 and 11.01, respectively; the MAE values were 4.35 and 5.79, respectively. The R2 values are given in Figure 7.

In ANN, the hidden-layer size parameter determines the number of hidden layers and the number of neurons per layer. The activation parameter specifies the function used to model the relationship between neurons. These are the identity, logistic, tanh, and ReLU functions, respectively. The solver parameter specifies the algorithm used to optimize the weights. These algorithms are determined as lbfgs (Newton-based), SGD (stochastic gradient descent), and Adam (adaptive). Learning rate init determines the initial learning rate (Scikit-learn, 2024). The

hyperparameter parameters used in the Artificial Neural Networks algorithm, along with the tested values , are presented in Table 8.

Figure 7. Performance of training and test phases of the Decision Trees algorithm.

The RMSE values calculated for the training and test data were 10.24 and 10.74, respectively; the MAE values for training and test data were 5.51 and 5.79, respectively. The R2 values are given in Figure 8.

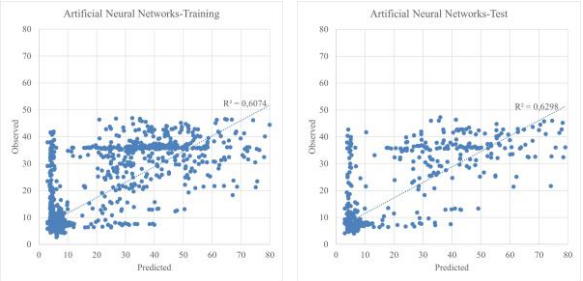

Figure 8. Performance of training and test phases of the Artificial Neural Networks algorithm.

The ability of common algorithms, including Random Forest (RF), Gradient Boosting (GB), Support Vector Regression (SVR), k-Nearest Neighbors (KNN), Decision Trees (DT), and Artificial Neural Networks (ANN), to predict air quality index values was evaluated. The input parameters, consisting of ambient temperature, relative humidity, and atmospheric pressure, were first reduced to two main components using principal component analysis (PCA) and provided as input to the models.

Table 8. Artificial Neural Networks hyperparameters and selected values

Parameter	Tested value	Selected value
Hidden_layer_sizes	(50,), (100,), (50, 50), (100, 50, 25)	(100, 50, 25)
Activation	identity, logistic, tanh, relu	tanh
Solver	lbfgs, sgd, adam	sgd
Learning_rate_init	0.001, 0.01, 0.1	0.01

According to the findings, the performance of the RF and GB algorithms was higher than that of the others. The R^2 value obtained for the test data in the RF algorithm was high, with an RMSE of 9.72 and an MAE of 5.15. This performance can be explained by RF's ability to capture complex relationships and its noise-resistant architecture. Similarly, the model's greater generalization ability can be explained by optimizing parameters such as the learning rate and subsampling. However, the significantly higher RMSE values for both algorithms during the test phase indicate a degree of overfitting.

The SVR algorithm yielded highly low R^2 value. Although SVR is known for its superior performance in modeling nonlinear relationships, the size of the dataset used in this study may not have captured complex relationships. Similarly, the ANN algorithm underperformed. This is because ANN require large, balanced datasets to function effectively. The study used only 2744 rows of data, limiting the ANN's potential performance. It should also be noted that the ANN's low performance may be attributable not only to the data volume but also to factors such as overfitting, parameter optimization, and the suitability of the data type.

The KNNs algorithm, achieved nearly perfect predictions on the training data. However, it showed significantly lower performance on the test data. This may be explained by KNN's high dependence on the training data and its poor generalization ability on large datasets. Another disadvantage of this algorithm is the high computational cost in high-dimensional datasets. This situation may limit the practical use of KNNs algorithm.

In such studies, when evaluating algorithm performance, sensitivity to data volume should be assessed separately. For algorithms such as SVR and ANNs, the dataset may be considered "small," whereas for KNNs it may be considered "large." This is due to the differences in the architecture of machine learning algorithms.

ANNs demonstrate very high performance in learning patterns with multiple parameters and complex relationships. However, for the model to recognize this complexity, relatively large datasets are required. The risk of overfitting increases with small datasets, and the model's identification performance may decline (Ou Yang et al., 2021). A similar situation applies to SVR. Because of its

kernel-based structure, it can achieve the desired performance only by appropriately tuning the hyperparameters. This, again, is directly related to data volume (Lima et al., 2013).

KNNs, an example-based algorithm, consider the entire training dataset when making predictions. This increases the computational load, especially for large datasets, reduces generalization ability, and may cause overfitting. Therefore, KNNs are more effective on small datasets because neighborhood relationships are more pronounced, thereby reducing computational load (Rahman, 2020). Therefore, a 2744-row dataset may be insufficient for ANNs and SVRs, whereas it may be considered large for KNNs.

The DT algorithm provides an easy-to-understand and interpretable method. However, it is susceptible to high variance and overfitting. With the tested parameters it has been noticed that the performance of DT algorithm is quite low.

As a result, the RF and GB algorithms achieved higher predictive performance with a limited number of environmental parameters. Hence, they can be used to predict air quality index. On the other hand, larger and more diverse datasets are required to improve the performance of more ANNs and SVR algorithms.

The BME 680 sensor used in the study not only provides air quality index values but also provides the air quality class at the time of measurement. During the study, only 164 hours of air quality class were recorded as average (2) (Table 1). This means that the air quality inside the barn was classified as good (1) for the remaining 2580 hours. This indicates that both mechanical and natural ventilation are functioning effectively.

In this study, the BME680 sensor was considered to have long-term limitations. Due to its metal-oxide (MO) structure, the sensor is prone to saturation and calibration drift, particularly when exposed to high concentrations of pollutants such as volatile organic compounds (VOCs) and ammonia. The system employs a self-calibration algorithm developed by Bosch Sensortec. This algorithm enables the sensor to use the best air-quality conditions as the "clean air" reference. In our study, we observed that the sensor was regularly exposed to good air quality (class 1) due to effective mechanical and natural ventilation. This situation limited the sensor's calibration drift. However, it should be noted that in barns where the sensor is exposed to higher pollutant concentrations, there is a risk of long-term stability issues. On the other hand, our findings demonstrate that this limitation does not eliminate the overall reliability of the results.

The lack of parameters, such as changes in ventilation conditions, manure management practices, and animal numbers, limited the models' performance. In addition, failures in the electrical system and the device also caused data losses. However, it should not be forgotten that the main purpose of this study is to observe the air quality inside the barn and to design a low-cost IoT-based system for this purpose. The results showed that it is straightforward and cost-effective to measure and incorporate additional parameters that can improve the method's modeling performance.

4. Conclusion

In this study, an IoT-based, low-cost system for monitoring air quality in dairy barns has been developed. The cost of the developed system was around $150, including sensors, other electronics, and parts. The data obtained has been modeled using major machine learning algorithms. The study results indicate that a low-cost device that integrates with cloud data systems can be further developed into a modern digital agriculture application. In addition, it has been observed that more accurate estimates can be obtained by including additional environmental parameters in the dataset. Moreover, the developed prototype has the potential to be used in other studies evaluating animal welfare and worker health. Our future studies will use larger, more comprehensive datasets to improve model performance.

Acknowledgement

The project was supported by the TÜBİTAK 2209-A University Students Research Projects Support Program under project number 1919B012321640.

Authorship contribution statement

E.D.: Data collection, laboratory work, article writing. O.S.: Planning, editing, article writing. N.K.: Planning, laboratory work, editing, article writing.

Conflict of interest

The authors declare no conflicts of interest.

Ethical Statement

There is no need to obtain ethics committee approval for this study.

References

Ao, Y., Li, H., Zhu, L., Ali, S., & Yang, Z. (2019). The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. Journal of Petroleum Science and Engineering, 174, 776–789. https://doi.org/10.1016/j.petrol.2018.11.067

Babiuch, M., & Postulka, J. (2020). Smart home monitoring system using ESP32 microcontrollers. In Internet of Things. IntechOpen. https://doi.org/10.5772/intechopen.94589

Dang, C., Nguyen, K., & Dao, H. (2020). Apply Matlab in Thingspeak server to build the system measure and analyze data using IoT gateway technology. Journal of Mining and Earth Sciences, 61(5), 10. https://doi.org/10.46326/jmes.2020.61(5).10

Dang, C., Nguyen, K., & Dao, H. (2020). Apply Matlab in Thingspeak server to build the system measure and analyze data using IoT gateway technology. Journal of Mining and Earth Sciences, 61(5), 88 – 95. https://doi.org/10.46326/jmes.2020.61(5).10

Deniz, E. (2024). Evaluating the effectiveness of machine learning models in predicting student academic achievement. ADBA Computer Science, 1(1), 8–13. https://doi.org/10.69882/adba.cs.2024072

Doğan, N., & Özyurt, F. (2025). IoT destekli hava durumu verileri ile yapay zekâ tabanlı hava tahmin sisteminin geliştirilmesi. KSÜ Mühendislik Bilimleri Dergisi, 28(1), 524 – 535.

Duroux, R., & Scornet, E. (2018). Impact of subsampling and tree depth on random forests. ESAIM: Probability and Statistics, 22, 96–128. https://doi.org/10.1051/ps/2018008

Gough, N. (2011). Avoiding CO_2. Science Signaling, 4(155), ec12. https://doi.org/10.1126/scisignal.4155ec12

Han, S., Qubo, C., & Meng, H. (2012, June). Parameter selection in SVM with RBF kernel function. In Proceedings of the World Automation Congress 2012 (pp. 1–4). IEEE.

Haroon, A. L., Shafiulla, M., Naveed, S. M., Ahmed, S., Nawaz, S. M., & Kumar, U. (2024). Home automation using Wi-Fi: ESP32-based system for remote control and environmental monitoring. In Proceedings of the 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). IEEE. https://doi.org/10.1109/ICDCECE60827.2024.10549726

Hempel, S., Saha, C., Fiedler, M., Berg, W., Hansen, C., Amon, B., & Amon, T. (2016). Non-linear temperature dependency of ammonia and methane emissions from a naturally ventilated dairy barn. Biosystems Engineering, 145, 10–21. https://doi.org/10.1016/j.biosystemseng.2016.02.006

Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). Design and Implementation of ESP32-Based IoT Devices. Sensors, 23(15), 6739. https://doi.org/10.3390/s23156739

Le Fort, E. (2018). A comparative study of machine learning algorithms (Master's thesis, McMaster University). McMaster University Archive.

Jolliffe, I. T. (2002). Principal component analysis (2nd ed.). Springer. https://doi.org/10.1007/978-1-4757-1904-8

Joshi, K., Aher, N., Arora, S., Mulay, V., Suryawanshi, R., Pise, N., & Gutte, V. (2023). Comparison of different machine learning and self-learning methods for predicting obesity on generalized and gender-segregated data. International Journal on Recent and Innovation Trends in Computing and Communication, 11(10), 464–471. https://doi.org/10.17762/ijritcc.v11i10.8510

Kaya, N. (2007). Overwintering with an empty super in beekeeping and its effects on relative humidity, temperature inside the hive, and colony survival (Publication No. 193820) [Master's thesis, Ankara University, Graduate School of Health Sciences, Department of Animal Science]. Council of Higher Education Thesis Center (YÖK Tez Merkezi)

Kelechi, A., Alsharif, M., Agbaetuo, C., Ubadike, O., Aligbe, A., Uthansakul, P., Kannadasan, R., & Aly, A. (2022). Design of a low-cost air quality monitoring system using Arduino and ThingSpeak. Computers, Materials & Continua, 70(1), 151–169. https://doi.org/10.32604/cmc.2022.019431

Kilic, I., & Yaslioglu, E. (2014). Ammonia and carbon dioxide concentrations in a layer house. Asian-Australasian Journal of Animal Sciences, 27, 1211–1218. https://doi.org/10.5713/ajas.2014.14099

Kim, J., Seo, S., Woo, S., Lee, J., Lee, Y., & Park, J. (2024). Distribution of air pollutant concentration and ammonium conversion rate in livestock areas: Focusing on Boryeong. Journal of Korean Society of Environmental Engineers, 46(10), 559–569. https://doi.org/10.4491/ksee.2024.46.10.559

Lima, A., Cannon, A., & Hsieh, W. (2013). Nonlinear regression in environmental sciences by support vector machines combined with evolutionary strategy. Computers and Geosciences, 50, 136-144. https://doi.org/10.1016/J.CAGEO.2012.06.023.

Ma, X., Zhang, Y., & Wang, Y. (2015). Performance evaluation of kernel functions based on grid search for support vector regression. In 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM) (pp. 283–288). IEEE. https://doi.org/10.1109/ICCIS.2015.7274635

Mahammad Rafi, D., Chandrashekar, R., Veena, C., Dutt, A., Waghmare, S. P., & Supriya, B. Y. (2023). Prediction of drug-target interactions through the application of supervised machine learning algorithms. In Proceedings of the 2023 International Conference on Emerging Research in Computational Science (ICERCS). IEEE. https://doi.org/10.1109/ICERCS57948.2023.10434245

Murad, S., Bakar, F., Azizan, A., & Shukri, M. (2021). Design of Internet of Things based air pollution monitoring system using ThingSpeak and Blynk application. Journal of Physics: Conference Series, 1962(1), 012062. https://doi.org/10.1088/1742-6596/1962/1/012062

Netzahual, J. E. T., Bautista, H. N., & Sánchez Hernández, M. J. (2019). Real-time control of robotic arm using Bluetooth Low Energy and Wi-Fi with the module board ESP32 and Android application. International Journal of Science and Research (IJSR), 8(4), 775–780.

Ou Yang, W. Y., Lai, C. C., Tsou, M. T., & Hwang, L. C. (2021). Development of machine learning models for prediction of osteoporosis from clinical health examination data. International Journal of Environmental Research and Public Health, 18(14), 7635. https://doi.org/10.3390/ijerph18147635

Rahman, F. (2020). Short term traffic flow prediction using machine learning - KNN, SVM and ANN with weather information. International Journal for Traffic and Transport Engineering, 10(3), 371 – 389. https://doi.org/10.7708/ijtte.2020.10(3).08.

Sanchis, E., Calvet, S., Prado, A., & Estellés, F. (2019). A meta-analysis of environmental factor effects on ammonia emissions from dairy cattle houses. Biosystems Engineering, 178, 176 – 183. https://doi.org/10.1016/j.biosystemseng.2018.11.017

Scikit-Learn. (2024). Machine learning in Python. https://scikit-learn.org/stable/

Shen, D., Wu, S., Li, Z., Tang, Q., Dai, P., Li, Y., & Li, C. (2019). Distribution and physicochemical properties of particulate matter in swine confinement barns. Environmental Pollution, 250, 746–753. https://doi.org/10.1016/j.envpol.2019.04.086

Singh, G. (2022). Machine learning models in stock market prediction. arXiv. https://doi.org/10.35940/ijitee.C9733.0111322

Taştan, M., & Gökozan, H. (2019). Real-time monitoring of indoor air quality with internet of things-based e-nose. Applied Sciences, 9(16), 3435. https://doi.org/10.3390/app9163435

Toprak, N. N., Öztürk, H., Yurdakök Dikmen, B., & Ünler, F. M. (2016). Sınırlı ve serbest kaba yemle beslenen kuzu yemlerine malat ilavesinin performans ve rumen fermantasyonu üzerine etkileri. In I. Uluslararası Hayvan Besleme Bilim Kongresi (pp. 384–386).

Truong, T., Nguyen, D., & Truong, P. (2021). Design and deployment of an IoT-based air quality monitoring system. International Journal of Environmental Science and Development, 12, 139–145. https://doi.org/10.18178/IJESD.2021.12.5.1331

Wu, C., Yao, H., Du, J., & Jiang, J. (2013). Research on parameter selection of support vector regression. Applied Mechanics and Materials, 344, 219–225. https://doi.org/10.4028/www.scientific.net/AMM.344.219

Yuliana, H. (2024). Hyperparameter optimization of random forest for 5G coverage prediction. Buletin Pos dan Telekomunikasi.

Yuliana H., Iskandar, Hendrawan, Basuki S., Hidayat M. R., Charisma A., & Vidyaningtyas H. (2024). Hyperparameter optimization of random forest algorithm to enhance performance metric evaluation of 5G coverage prediction. Buletin Pos dan Telekomunikasi, 22(1), 75–90. https://doi.org/10.17933/bpostel.v22i1.390

Zhang, L., Zhao, X., Sun, X., Wu, Q., Fu, H., Hu, M., & Chen, M. (2022). The design and implementation of I2C driver for wireless sensor terminal equipment. Proceedings of SPIE, 12455, 124550B–124550B-5. https://doi.org/10.1117/12.2655410