

YU.O. SHELIKHOV, N.G. AXAK
Kharkiv National University of Radio Electronics

MICROSERVICE-BASED ADAPTIVE INDOOR MICROCLIMATE CONTROL SYSTEM USING LSTM FORECASTING

Maintaining stable indoor microclimate conditions is a critical challenge in smart buildings, precision agriculture, and industrial environments, where traditional centralized control systems often fail to provide sufficient flexibility, scalability, and resilience under dynamically changing conditions. This study proposes a microservice-based adaptive microclimate control system that integrates real-time IoT sensing, distributed service orchestration, and machine-learning-driven forecasting to enable proactive environmental management.

The research methodology combines system-level domain analysis, architectural decomposition into autonomous microservices, implementation of lightweight communication mechanisms based on REST and MQTT protocols, containerized deployment using Docker technologies, and empirical evaluation of a functional prototype. The proposed system architecture separates data acquisition, storage, analytics, decision-making, actuator control, and monitoring into independent services, ensuring loose coupling, fault isolation, and independent horizontal scaling. Such decomposition enables continuous system evolution and flexible adaptation to varying operational scenarios without service interruption.

A central contribution of the work is the integration of an analytical microservice utilizing Long Short-Term Memory (LSTM) neural networks for forecasting temperature and humidity parameters. Sensor streams undergo preprocessing, normalization, and aggregation before being used for predictive modeling, allowing the control subsystem to anticipate future environmental changes and execute preventive actions rather than reactive responses.

Experimental evaluation demonstrated stable system performance with end-to-end response latency below two seconds under realistic workloads. The forecasting module achieved strong predictive accuracy, with determination coefficients of approximately $R^2 = 0.90$ for temperature and $R^2 = 0.94$ for humidity, while maintaining low prediction errors (MAE of 0.21 °C and 0.56 %, respectively). These results confirm that integrating predictive analytics into a microservice architecture significantly improves environmental stability and energy efficiency.

The proposed approach demonstrates the feasibility and practical effectiveness of combining microservice-based software engineering principles with intelligent data-driven control strategies. The developed solution is suitable for deployment in smart buildings, indoor farms, and industrial facilities requiring scalable, resilient, and adaptive microclimate management systems.

Key words: microservice architecture, monitoring, neural network, distributed systems, machine learning, IoT.

Ю.О. ШЕЛІХОВ, Н.Г. АКСАК
Харківський національний університет радіоелектроніки

МІКРОСЕРВІСНА СИСТЕМА АДАПТИВНОГО КЕРУВАННЯ МІКРОКЛІМАТОМ ПРИМІЩЕНЬ НА ОСНОВІ LSTM-ПРОГНОЗУВАННЯ

Підтримання стабільних параметрів мікроклімату є критично важливим завданням для розумних будівель, систем закритого землеробства та промислових об'єктів, де традиційні централізовані системи керування часто не забезпечують необхідної гнучкості, масштабованості та відмовостійкості в умовах динамічних змін зовнішнього середовища. У роботі запропоновано мікросервісну систему адаптивного керування мікрокліматом, яка поєднує реальний збір даних із IoT-сенсорів, розподілену взаємодію сервісів та прогнозування на основі методів машинного навчання для реалізації проактивного керування параметрами середовища.

Методологія дослідження включає системний аналіз предметної області, архітектурну декомпозицію системи на автономні мікросервіси, використання протоколів REST та MQTT для обміну даними, контейнеризацію компонентів засобами Docker, а також експериментальне тестування реалізованого прототипу. Запропонована архітектура передбачає розділення функцій збору даних, зберігання, аналітики, формування керувальних рішень, керування виконавчими пристроями та моніторингу, що забезпечує слабку зв'язаність компонентів, локалізацію відмов і можливість незалежного масштабування сервісів.

Ключовим внеском роботи є реалізація аналітичного модуля на основі рекурентних нейронних мереж типу Long Short-Term Memory (LSTM), що дозволяє прогнозувати температуру та вологість на основі історичних часових рядів. Попередня обробка даних включає фільтрацію, нормалізацію та агрегування показників, після чого модель формує прогноз майбутніх значень, що забезпечує перехід від реактивного до проактивного керування мікрокліматом.

Експериментальна оцінка підтвердила стабільну роботу системи із середньою затримкою реакції менше дві секунди. Модель прогнозування продемонструвала високу точність: коефіцієнт детермінації $R^2 \approx 0,90$

для температури та $R^2 \approx 0,94$ для вологості при середній абсолютній похибці MAE 0,21 °C і 0,56 % відповідно. Отримані результати підтверджують, що використання прогнозної аналітики в поєднанні з мікросервісною архітектурою дозволяє підвищити стабільність параметрів середовища та енергоефективність системи.

Запропонований підхід демонструє практичну ефективність інтеграції принципів мікросервісної архітектури з інтелектуальними методами аналізу даних і може бути використаний у розумних будівлях, вертикальних фермах і промислових середовищах, де необхідне масштабоване та адаптивне керування мікрокліматом.

Ключові слова: мікросервісна архітектура, моніторинг, нейронна мережа, розподілені системи, машинне навчання, IoT.

Problem Statement

Ensuring stable indoor microclimate conditions is an essential requirement for modern intelligent environments such as smart buildings, vertical farms, industrial facilities, and controlled agricultural systems. Maintaining optimal environmental parameters – including temperature, humidity, carbon dioxide concentration, and lighting – directly affects human comfort, productivity, technological process stability, and energy efficiency.

Traditional automation systems are typically based on centralized architectures. While suitable for static or low-variability conditions, they often demonstrate limited adaptability, insufficient responsiveness to rapid environmental changes, poor scalability, and low fault tolerance. These limitations hinder their effectiveness in dynamic environments characterized by continuous sensor data streams and rapidly changing external conditions.

Recent technological advances – particularly in the Internet of Things (IoT), edge computing, cloud infrastructure, and artificial intelligence – have enabled the development of distributed intelligent control systems capable of real-time monitoring and autonomous decision-making. Among the architectural paradigms, microservice architecture has emerged as an effective approach for building scalable and modular distributed systems by decomposing functionality into loosely coupled services responsible for data acquisition, analytics, decision logic, and actuator control.

The relevance of applying a microservice-based approach to adaptive microclimate control is grounded in its flexibility, scalability, seamless integration with IoT and cloud/edge infrastructures, improved fault tolerance, and support for predictive analytics. These properties make it possible to transition from reactive regulation strategies toward proactive environmental management.

However, existing solutions often address either architectural design aspects or predictive modeling techniques separately, while integrated approaches that combine microservice-based system engineering with data-driven forecasting remain insufficiently explored. This creates a gap between scalable distributed system design and intelligent adaptive control mechanisms.

Therefore, the unresolved issue lies in the lack of an integrated solution that combines scalable microservice-based architecture with predictive time-series analytics to enable proactive environmental regulation. This gap defines the scientific and engineering challenge addressed in this study.

Analysis of Recent Research and Publications

Microservice architecture enables the decomposition of a system into independent, loosely coupled services with clearly defined responsibilities. In microclimate control, this architectural style allows separate services to manage data acquisition from sensors, process and analyze environmental data, make control decisions, and interact with actuators and users. Such decomposition increases modularity, scalability, and fault tolerance while facilitating continuous system evolution. For example, Haque et al. [1] implemented a climate control system where each functional unit—data collection, analytics, and user interface—operated as an independent service, significantly simplifying development and maintenance. Similarly, a BEMS refactored into microservices demonstrated the ability to manage over 3200 devices compared to only 400 in its monolithic predecessor, owing to improved load balancing [1].

Microservice-based architectures also improve system resilience. Failures of individual services, such as logging or analytics modules, do not compromise the operation of critical components

like device control services, thereby preventing cascading system-wide disruptions [1]. This architectural advantage is increasingly relevant in distributed and dynamic environments such as precision agriculture or smart buildings.

Several practical implementations confirm the feasibility of using microservices in intelligent control platforms. The IRRISENS system [2], designed for agricultural irrigation management, was built entirely on microservices. Each farm facility ran its services to gather soil, crops, and weather data, perform forecasting based on evapotranspiration models, and issue irrigation commands. Integration with FIWARE's ORION Context Broker enabled orchestration and interoperability across cloud services. The system was successfully tested on four commercial farms, demonstrating efficiency in water usage and data security through tenant isolation [2].

In smart indoor farming, machine learning methods have become integral for achieving adaptive climate control. A notable example is presented in [3], where Q-learning is employed to train an agent that optimizes temperature and humidity by interacting with the environment and receiving rewards based on performance. This approach allows the system to adapt dynamically to changing conditions without requiring a predefined analytical model. Another contribution [4] outlines a modular architecture with dedicated subsystems for data collection, decision-making, actuator control, and monitoring, which supports the integration of AI components, including Q-learning, and emphasizes system flexibility through detailed UML-based design.

Sensor data must undergo preprocessing to ensure the reliability and effectiveness of control decisions. Studies show that crucial techniques include noise filtering, data aggregation, and multi-sensor fusion. For instance, edge-level filtering using moving averages or Kalman filters reduces unnecessary data transmission and network load [5]. Aggregation methods combine readings over time or from multiple sensors, providing meaningful summaries without flooding the cloud with raw data [5]. Fusion techniques combine data types—such as temperature and humidity—to derive more informative indicators like vapor pressure deficit (VPD). In [6], such fusion is performed at the fog computing level in a real aeroponic greenhouse, with processed results stored in the cloud and made accessible via mobile applications. Another case, described by Rezvani et al. [7], presents a fuzzy inference system that combines multiple climate parameters into an Optimality Degree index, offering a clear assessment of greenhouse conditions for tomato cultivation.

Many modern systems adopt multi-level processing architectures (edge–fog–cloud) to balance speed and computational load. In a four-layer architecture described in [6], data collected by sensors is initially processed at fog nodes, enabling real-time responses, and then transmitted to the cloud for long-term analytics and visualization. This design ensures system resilience, allowing local decision-making even during temporary network outages.

Machine learning plays a central role in predictive and adaptive climate control. Recurrent neural networks such as LSTM have been used to forecast temperature and humidity accurately. Thwin et al. [8] developed a multi-step LSTM model for greenhouse climate prediction, achieving $R^2 \approx 0.98$ with RMSE ~ 0.49 , and maintaining optimal conditions even under external heat stress. A hybrid approach combining LSTM and SVM was also tested for simultaneous forecasting and classification of environmental conditions, demonstrating superior performance compared to classical algorithms [8]. In addition to forecasting, classification models help the system recognize states such as overheating or equipment failure, and trigger appropriate responses. Online learning and model adaptation techniques ensure long-term robustness by continuously updating parameters. For instance, Van Henten et al. applied an Extended Kalman Filter to identify changing system dynamics, improving model accuracy over time. Meanwhile, neural networks with transfer learning capabilities can adjust to new conditions during operation [9]. Reinforcement learning methods have also shown promising results in selecting optimal control strategies under uncertainty, balancing environmental comfort with energy efficiency [10].

These findings underscore the transformative potential of intelligent microclimate systems that combine IoT infrastructure with AI capabilities. Such systems reduce human intervention, enhance sustainability, and offer practical benefits in precision agriculture, green buildings, and indoor farming.

Ongoing research ranges from conceptual designs to applied solutions. Theoretical contributions often explore orchestration frameworks and adaptive architectures. One example is a three-layer design for green buildings that adapts its structure at runtime using ML-based rules, successfully tested at the CIESOL facility in Spain [11]. On the other hand, real-world deployments like IRRIS-ENS [2] and the Adaptive Greenhouse system [8] confirm the effectiveness of these technologies in diverse operational contexts, while revealing practical challenges in wireless communication, sensor autonomy, and cybersecurity. Nonetheless, these implementations consistently report measurable improvements in water conservation, energy efficiency, and crop productivity, demonstrating the real-world impact of microservice-based adaptive climate control.

Aim of the Study

The aim of this research is to develop and experimentally validate an integrated microservice-based adaptive microclimate control system that combines distributed IoT-driven data acquisition with LSTM-based predictive analytics in order to enable proactive environmental regulation, improve system scalability, and enhance energy efficiency.

To achieve this aim, the following research objectives were defined:

- to design a modular microservice architecture ensuring functional decomposition and fault isolation;
- to implement real-time data ingestion, storage, and monitoring services;
- to develop and integrate an LSTM-based forecasting model for temperature and humidity prediction;
- to evaluate forecasting accuracy and system latency under realistic operational conditions.

Designing a microservice system for adaptive microclimate control

In the system, each microservice is an independent software component that performs a clearly defined function, such as ventilation, heating, or lighting control. These services are built on the principle of autonomy: they have their logic and a separate API and do not depend on the internal implementation of other system parts. Their specialization allows you to focus on tasks such as temperature control, humidity monitoring, or CO₂ readings. All system components interact via lightweight protocols (REST, MQTT, WebSocket), which provide flexibility in integration and data exchange.

Thanks to the microservice approach, this system is easily scalable – new services can be added without stopping the main infrastructure, and existing services can be updated or replaced independently. The system's adaptability is manifested in each service's ability to dynamically respond to changing external conditions, such as air temperature or the presence of people. For example, a temperature control service regulates the operation of heaters by receiving up-to-date readings from thermal sensors. At the same time, a humidity module is responsible for humidifying or dehumidifying the air based on humidity data. At the same time, another microservice can monitor the CO₂ level, automatically activating ventilation when the permissible threshold is exceeded.

Another critical component is the energy efficiency module, which analyzes energy consumption and adapts the operating mode of devices based on the schedule or presence of users. Integration with IoT allows the system to control smart appliances, such as opening or closing blinds to regulate the lighting and heat load level. The advantages of this architecture are flexibility in development, high fault tolerance (an error in one service does not affect the operation of others), ease of scaling, and the ability to use different technologies within one system.

As a result, microservice architecture provides adaptive, reliable, and efficient microclimate control in smart buildings, greenhouses, or industrial facilities where precise response to environmental changes is essential.

A detailed development plan for a microservice adaptive climate control system covers all key stages of design, implementation, and deployment, allowing for a systematic approach to creating

an intelligent, flexible, and scalable system with high adaptability. At the initial stage, an analysis of functional requirements is performed, particularly determining the environmental parameters that need to be controlled: temperature, humidity, CO₂ content, and lighting level. In parallel, the types of actuators are established, including heaters, air conditioners, humidifiers, ventilation units, and lighting systems. Next, an appropriate architectural approach is selected, and in this context, microservice architecture is the most suitable, as it provides for the independence of individual system components. The main functional blocks are defined: data collection, storage, analytics, device management, API, and monitoring services. To visualize the interactions between them, a generalized data flow diagram is created, taking into account the types of communication – both synchronous (REST) and asynchronous (event queues, MQTT).

Microservices design involves the implementation of specialized components: a temperature service that processes data from temperature sensors and regulates the operation of heaters or air conditioners; a humidity service that interacts with the corresponding sensors and controls humidifiers or dehumidifiers; a CO₂ module that monitors carbon dioxide and activates ventilation when the threshold is exceeded; a lighting service that regulates artificial lighting or bright curtains; an analytical service that aggregates data, detects anomalies, and performs predictions based on machine learning algorithms; and a device controller that acts as an intermediary between analytics and performers.

Communication between services is organized through standardized REST APIs or gRPC interfaces. Data exchange uses HTTP/HTTPS, MQTT (for IoT devices), and RabbitMQ or Kafka for asynchronous event exchange. API protection is provided using JWT tokens, HTTPS encryption, and role-based access control (RBAC) mechanisms.

At the data processing level, the system collects information in real time with subsequent filtering and normalization. Historical data is stored in time-series databases (InfluxDB, TimescaleDB), while metadata is stored in the PostgreSQL relational database. Decision-making uses rule-based algorithms and machine learning models, including LSTM, regression, and classification methods.

Integration with sensor and executive infrastructure is carried out via MQTT brokers and standard protocols such as Modbus, Zigbee, or Wi-Fi. Device control uses HTTP requests, GPIO, or MQTT, with a response time of no more than two seconds.

Scalability and reliability of the system are achieved through containerization of all services in Docker, with the possibility of centralized orchestration using Docker Compose or Kubernetes. Prometheus and Grafana are used for monitoring, and an alert system is used to detect failures. Logging is carried out centrally via NLog or the ELK stack.

At the final stage, testing is carried out – both modular and integration, as well as load testing using tools such as JMeter. The system is deployed via CI/CD pipelines, separated into environments for testing and productive operation. This approach guarantees adaptability, reliability, and convenient integration of the developed system into existing infrastructures of an innovative environment.

Developing a microservice adaptive microclimate control system involves a precise formulation of functional and non-functional requirements that determine the system's architecture, operating logic, and expected behavior in real operating conditions. At the functional level, the system should collect automated data from temperature, humidity, CO₂, and light sensors, and store this data in a database with an accurate timestamp for further analysis. In addition to accumulating information, the system should be able to make decisions on turning on or off controlled devices, including ventilation equipment, heaters, or other actuators, based on the values obtained.

The user should be able to interact with the system via API, in particular, to receive the current state of the microclimate, view sensor values in real time, and analyze the dynamics of indicators over a specified period. This is implemented using a client application or web interface that allows access to historical data and the results of automatic decisions.

Regarding non-functional requirements, the system should guarantee data processing with minimal delay – no more than two seconds from when the measurement is received to when the corresponding

control decision is made. To achieve high availability and fault tolerance, all microservices are developed as isolated containers that run in a Docker environment and can be independently deployed.

Architecture of the adaptive microclimate control system

It is based on a microservice approach that provides modularity, flexibility, scalability, and high fault tolerance. Each service performs a clearly defined function, allowing individual components to be developed, deployed, and updated independently. Thanks to this approach, the system can easily adapt to changing requirements or expanding functionality, which is especially important in a dynamic environment.

At the center of the architecture is the Sensor Ingestion Service, responsible for receiving and processing data from temperature, humidity, CO₂, and light sensors. This data is transferred as streams or messages to the Analytics Service, which analyzes the indicators, determines deviations from the standard values, detects anomalies, and generates solutions for managing the environment.

The results of the calculations are transferred to the Actuator Control Service, which directly interacts with actuators such as fans, heaters, lighting, or humidifiers and sends commands to turn them on or off according to the current state of the environment and the specified logic. At the same time, all incoming and processed data is recorded in the Storage Service, which ensures that sensor values and system actions are saved with time stamps in the database.

All external requests to the system, including those from user applications or other services, go through the Gateway API, which performs routing, load balancing, authentication, and authorization of access to the relevant microservices. This allows you to control the entry points to the system centrally and ensures the security of communications.

To maintain the operability and diagnose the functioning of components in real time, the Monitoring Service is implemented, which collects technical metrics, records event logs, monitors the load, and allows for a quick response to failures. Thanks to this architecture, the system is functionally flexible and ready to scale and integrate new services or devices without compromising its stability (Fig. 1).

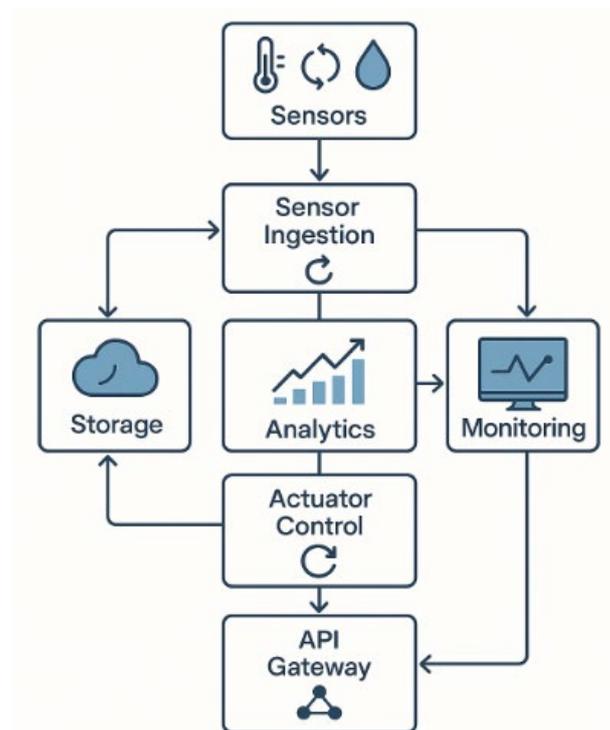


Fig. 1. Architecture of the adaptive climate control system.

Sensors – collect primary data on temperature, humidity, CO₂, and light.

Sensor Ingestion – receives data streams from sensors and redirects them to other services.

Storage – stores incoming data in a database (including for retrospective analysis).

Analytics – processes indicators, detects deviations, performs analysis, and generates decisions.

Actuator Control – controls actuators based on analytics solutions.

API Gateway – provides centralized access to the system's functionality from the outside.

Monitoring – monitors the status of the system and its components.

The arrows between the blocks represent the flow of data and commands, in particular: from sensors to ingestion; from ingestion to storage, analytics, and monitoring; from analytics to actuator control and monitoring; from actuator control to API gateway; and feedback from API gateway to actuator control and monitoring.

Designing a microservice system for adaptive microclimate control

It involves creating separate independent services, each performing a specific function in the overall control structure. This approach allows for high system flexibility, simplifies scaling, testing, and maintenance, and allows for gradual expansion of functionality.

A key role in interacting with physical devices is played by the Sensor Ingestion Service, which is responsible for receiving MQTT messages from IoT sensors that transmit temperature, humidity, light, and CO₂ levels. This service performs preliminary data processing, including normalizing values and checking the correctness of the message structure before transferring them for further analysis and storage.

The received data is transferred to the Storage Service, which is implemented based on the PostgreSQL relational database. All measurements are stored with a precise time reference, which allows tracking changes in parameters over time, conducting retrospective analysis, and using historical data to train predictive models.

The central element of intelligent processing is the Analytics Service, which calculates averages, detects anomalous conditions, and predicts climate parameters using models based on long-term short-term memory (LSTM). The resulting analytical findings form the basis for changing the system's operating modes.

The decisions made are transferred to the Actuator Control Service, which generates commands for environmental control devices, such as heating, ventilation, humidification, or lighting systems. This service is responsible for interacting with real actuators and ensuring that their behavior aligns with current conditions and the recommendations of the analytical module.

The Ocelot-based Gateway API is used to coordinate the interaction between external clients and microservices. It is responsible for authorizing requests via JWT tokens, routing to the corresponding REST API services, and providing load balancing between service instances in case of scaling.

To control the entire system's stability, the Monitoring Service has been implemented, which integrates Prometheus and Grafana monitoring tools. This service allows you to track the technical metrics of each component, create performance graphs, detect deviations, and respond quickly to critical events using an alert system.

These microservices form an interconnected, adaptive, and scalable system that can dynamically respond to microclimate changes and provide effective indoor environmental management.

Figure 2 shows a simplified block diagram of the design of a microservice system for adaptive microclimate control, which shows the main components of the system and the directions of their interaction.

Sensor Ingestion (MQTT → Validation). This microservice receives data from sensors (temperature, humidity, CO₂, etc.) via the MQTT protocol. After receiving the data, it is validated to check its correctness before being transmitted further to the system.

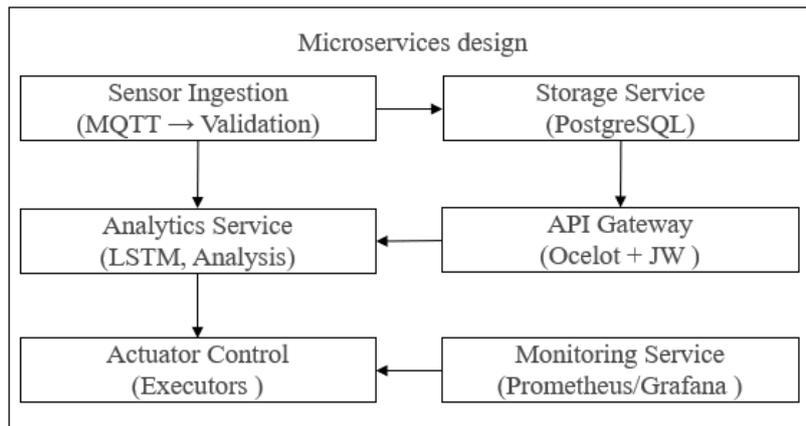


Fig. 2. Block diagram of the design of a microservice system for adaptive microclimate control

Storage Service (PostgreSQL). The received and validated data is transferred to the storage service based on the PostgreSQL relational database. Both current sensor values and historical records are stored here for further analysis.

Analytics Service (LSTM, analysis). This component analyzes the collected data and notably predicts climate parameters using machine learning algorithms (LSTM). Based on the study, a decision is made on actions to change the microclimate.

API Gateway (Ocelot + JWT). API Gateway acts as a gateway for external requests. It authorizes users (in particular, through JWT) and routes requests to the appropriate internal services.

Actuator Control. Based on forecasts or analytics recommendations, this service generates control commands for actuators, such as ventilation, lighting, heating, etc.

Monitoring Service (Prometheus/Grafana). Responsible for monitoring all services, collecting metrics (Prometheus), and visualizing indicators (Grafana). It also monitors the status of actuators and reports failures or anomalies.

The flow begins with receiving data from sensors. The data is stored and transferred to analytics and from there to actuators.

Designing data analysis modules

Data analytics in an adaptive microclimate control system is a key element that enables intelligent decision-making based on real-time sensor information and historical trends. The primary purpose of analytics is not just to process input data, but to identify patterns, predict changes in environmental parameters, and formulate practical management actions. The main functions of analytics. Data pre-processing and cleaning. The system receives the temperature, humidity, CO₂ level, and illumination values in the form of MQTT message streams. The data is filtered (for example, using a moving average or Kalman filter) to eliminate noise, outliers, or anomalous values.

Aggregation and statistical analysis. The processed indicators are aggregated as average, minimum, maximum, or variance values at specified intervals (for example, every minute or every hour). This allows you to reduce the amount of data and identify deviations from normal conditions.

Anomaly detection. The analytics service automatically identifies deviations from typical values based on thresholds or classification methods. For example, it can detect excessive humidity or CO₂ levels, which indicate a ventilation problem or the presence of people in the room.

Prediction of climate parameters. Proactive environmental management uses time-series temperature, humidity, or CO₂ forecasting using an LSTM (Long Short-Term Memory) model. This recurrent neural network is good at dealing with dynamic changes over time. For example, the model can predict a temperature increase in 30 minutes and activate ventilation or reduce lighting in advance.

Generating recommendations. Based on the analysis and forecast, the system creates control strategies as commands for actuators. For example, if the temperature threshold is exceeded, a command is activated to turn on the cooling or open the blinds.

Evaluation of control efficiency. After the commands are executed, the system continues to monitor changes in the environmental parameters, recording whether the desired effect has been achieved. This allows you to analyze the effectiveness of the selected action and use the feedback to train models in the future.

Figure 3 shows a schematic structure of a recurrent neural network of the LSTM (Long Short-Term Memory) type, which consists of serially connected LSTM cells. This type of network can store and update information over long intervals thanks to special memory management mechanisms.

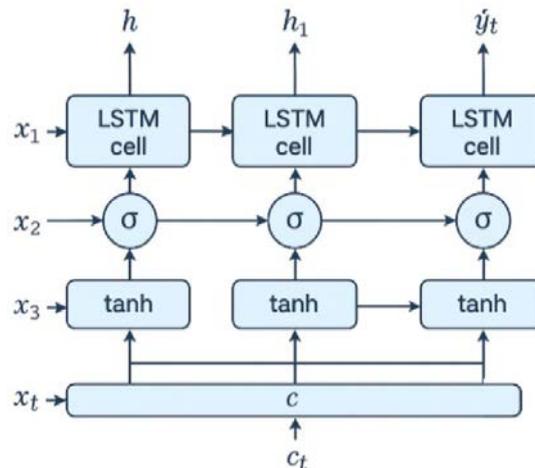


Fig. 3. Structure of a recurrent neural network of the LSTM (Long Short-Term Memory) type

The main elements in the diagram:

- LSTM cell – the central computational unit, which receives as input the feature vector x_t , the internal memory state c_t , and the previous hidden state h_{t-1} at each time step.
- $x_1, x_2, x_3, \dots, x_t$ – sequence of input vectors (e.g., temperature, humidity, etc. measurements in a time series).
- h, h_t, y_t – output hidden states or predicted values.
- c_t – internal memory state vector, which carries the main information through the entire sequence and is updated at each step.
- σ – sigmoidal activation function, used in the input, forget, and output gates to determine what information will be retained or discarded.
- \tanh – hyperbolic tangent, normalizing values added to a memory state or generated as output.
- Arrows represent information flows between layers: from left to right, successive time steps; vertical arrows, internal processing within a single cell.

The principle of operation is as follows:

At each time step t , the model receives input x_t , the previous memory state c_{t-1} , and the hidden state h_{t-1} . Sigmoidal gates (σ) determine which parts of the old state to keep, which to update, and which to pass on.

The hyperbolic function \tanh forms a new proposal to the state, which is combined with the old memory contents.

The updated memory state c_t is passed on to the next LSTM cell.

The output state h_t or y_t is passed to the next step as a prediction or feature.

As input, the LSTM model receives a time series of sensor data for the last 30 minutes, which are pre-normalized to a range from 0 to 1 for computational stability. The network architecture starts with an input layer, which accepts a fixed-length data fragment. The basis is one or more LSTM layers with a given number of neurons (for example, 64 or 128), which can save the sequence of initial states. To prevent overtraining, a dropout layer is added, and the final one is a fully connected output layer that generates a forecast for one or more environmental parameters at once.

Depending on the settings, the model can perform single-point forecasting (for example, values in 15 minutes) or multi-step prediction (the next 5–60 minutes ahead). The results are presented as numerical estimates for each parameter – such as predicted temperature or humidity values.

The primary function of such a model is to detect future threshold exceedances before they occur. This allows the system to act proactively, forming recommendations or directly generating control commands for actuators, such as fans, heaters, or automated blinds. As a result, the system responds to current conditions and prevents deterioration in advance, significantly increasing environmental comfort and management energy efficiency.

Python and the Keras, NumPy, and Pandas libraries were used to predict microclimate parameters (temperature and humidity).

Step 1. Data preparation.

Data collection from sensors in CSV format or via API. They contain the following fields: timestamp, temperature, humidity, CO₂, and light.

Several operations were performed for data preprocessing to make the data from the sensors suitable for feeding into the LSTM model. Initially, the data looks like this :

```
timestamp, temperature, humidity, CO2, light
2024-06-01 12:00:00,23.5,48.1,420,300
2024-06-01 12:01:00,23.7,48.0,419,305
```

...

We check that the data has a fixed time interval, for example, 1 minute. We remove anomalous values or noise if there are gaps or uneven distances between records. LSTM learns better when the input data is in the range [0, 1] (minmax-normalization). We save the scaled data set. Now, scaled_df is a uniformly interpolated, cleaned, and scaled array, ready to form sequences for LSTM.

Step 2. Forming training sequences is preparing the data in the format that the LSTM model expects: sequences of fixed length that are fed as input, and the corresponding "targets" (outputs) that the LSTM should predict.

LSTM is trained on sequences of values, not on individual points. For example, to predict the temperature one minute ahead, we feed the previous 30 minutes as input. This is called the observation window.

After scaling, we have an array of dimension: scaled_df → shape: (N, 2), where N is the number of time records, and 2 is the number of features (e.g., temperature + humidity).

Thus, we create X – a three-dimensional array for the model, with dimensions (number of sequences, length of sequence, number of features), and y – a two-dimensional array of target values, with dimensions (number of sequences, number of features)

Let's say we have 100 minutes of data (N = 100), and window_size = 30.

– Number of possible sequences: 100 – 30 = 70

– Each element X[i] is a 30×2 matrix (30 minutes, 2 parameters)

– Each y[i] is a vector of 2 values (at the 31st minute: temperature and humidity)

These X and y are used in model.fit(X, y, ...)

Multi-step forecasting means that instead of predicting a single next value (e.g., temperature in 1 minute), the LSTM model predicts several future values at once – for example, the next 60 minutes (i.e., y will have a size of 60 steps instead of 1).

There are two main approaches:

First. One-to-many (direct forecast)

The model is trained to output a vector of predictions for n next steps simultaneously.

Second. Recursive forecast (step-by-step)

The model forecasts only 1 step, and then uses its forecast as input for the next forecast (accumulating errors!).

The best approach for stability is one-to-many

The format of y changes:

Was: y[i] = data[i + window_size]

Now: y[i] = data[i + window_size : i + window_size + forecast_horizon]

As a result, we get:

X.shape = (N, 30, 2)

y.shape = (N, 60, 2) – that is, 60 future points for each of the two parameters

The graph in Figure 4 compares the actual temperature and the LSTM model forecast for 60 minutes. It allows us to assess the accuracy of the forecast and identify possible deviations in the model's behavior.

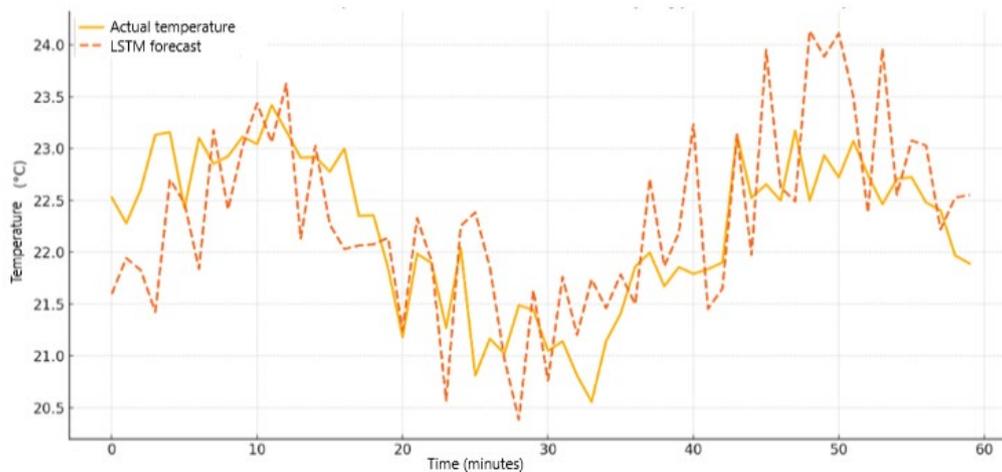


Fig. 4. Comparison between actual temperature and LSTM model forecast for 60 minutes.

The yellow curve shows the actual temperature values obtained from the sensors. The orange curve is the model forecast based on previous data. The LSTM model reproduces the overall temperature dynamics quite accurately. There are slight lags or smoothing of peaks – this is a typical effect for LSTM, which seeks to avoid sudden changes.

The forecast remains within the permissible fluctuations, which indicates good agreement with the real data.

The graph in Figure 5 compares the actual and forecasted humidity values (%) 60 minutes ahead. Such visualization allows us to assess the quality of the LSTM model in different microclimate parameters and identify potential forecast errors for further model improvement.

The yellow line shows the actual humidity values from the devices. The orange line is the model's forecast for 60 minutes ahead. The humidity forecast is slightly more volatile than the temperature forecast. This is because humidity often undergoes sudden changes (for example, turning on a humidifier).

The model tries to predict the general trend, but does not always accurately guess the moment of change. This means it needs to be trained with longer historical windows or supplemented with external features (for example, when a humidifier is turned on).

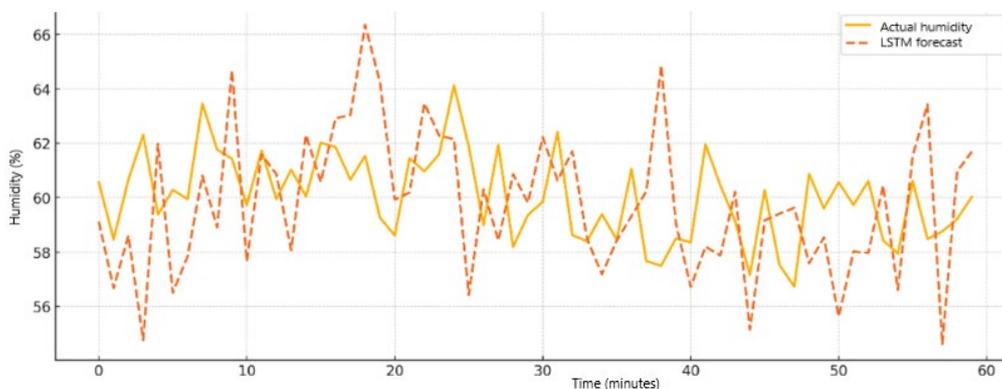


Fig. 5. Comparison between actual and forecasted humidity values (%) 60 minutes ahead

The LSTM model copes effectively with the forecast of trends and smooth changes in climate parameters. It can proactively control the system (for example, cool the air before it gets hot).

To increase the accuracy of the forecast, especially for humidity, it is worth adding other input features or optimizing the model architecture (for example, through attention or ensemble models).

The accuracy of the LSTM model in predicting microclimate parameters 60 minutes ahead was evaluated using standard regression metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the coefficient of determination (R^2).

The model achieved an MAE of 0.2077°C for temperature prediction, indicating a very low average deviation between predicted and actual values. The MSE was 0.0787, which suggests that large deviations were rare, as this metric penalizes significant errors more heavily. The R^2 score reached 0.9019, meaning that the model successfully captured over 90% of the variance in temperature measurements, demonstrating strong predictive performance.

In the case of humidity forecasting, the model performed even better. The MAE was only 0.5588%, and the MSE was 0.5190, both confirming high precision. The R^2 score of 0.9448 indicates that the model explained approximately 94.5% of the variation in humidity values. This highlights the model's ability to capture trends and fluctuations in environmental humidity accurately.

The results show that the LSTM model demonstrates high forecasting accuracy for both parameters. While the R^2 value for temperature is slightly higher, the MAE and MSE values for humidity are more favorable, suggesting greater robustness in humidity prediction. These findings confirm the suitability of LSTM-based forecasting for adaptive control in innovative microclimate systems.

The graph (Fig. 6) shows histograms of the distribution of forecast errors for temperature and humidity.

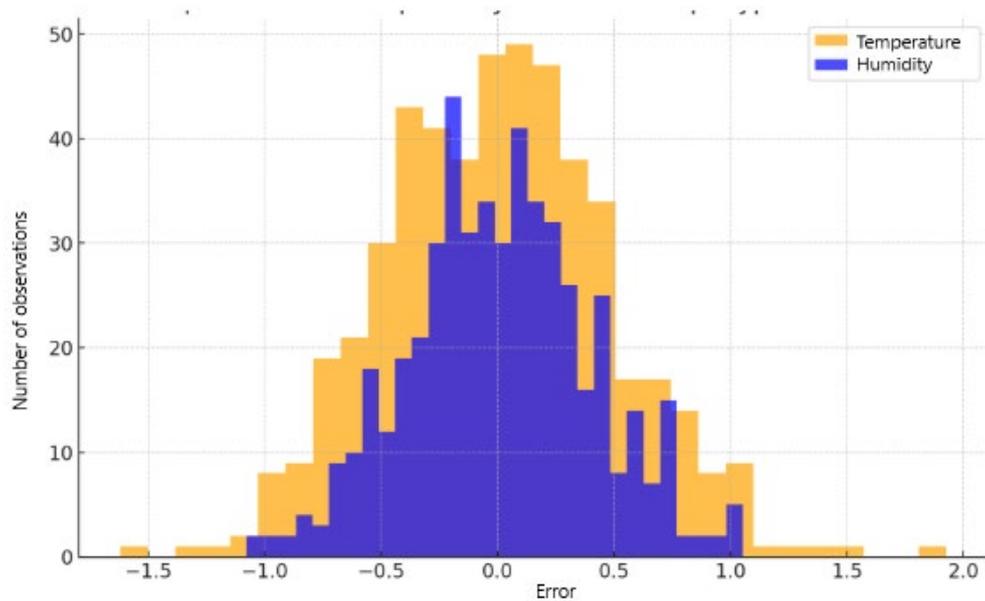


Fig. 6. Histogram of the distribution of forecast errors for temperature and humidity

This histogram shows the distribution of temperature and humidity forecast errors:

- The temperature forecast errors (orange fill) have a roughly normal distribution around zero, indicating that the model has no systematic bias in its temperature forecast.
- The humidity forecast errors (blue fill) also have a symmetric distribution, although slightly less variance, indicating a more stable forecast.

Both distributions are centered within ± 1 , which is a good indication of the model's accuracy.

Conclusions

In this paper, we have implemented a microservice system for adaptive microclimate control, which can automatically adjust environmental parameters (temperature, humidity) based on data obtained from IoT sensors. A modular architecture was used, where each microservice performs a highly specialized function: collection, storage, analytics, device management, and monitoring.

Particular attention was paid to developing an analytical service that predicts microclimate parameters using a recurrent neural network such as LSTM. This made it possible to increase the proactivity of the system: decisions to turn equipment on or off are made not only based on current values, but also taking into account expected changes in the next 60 minutes.

Evaluation of the accuracy of LSTM forecasts showed an acceptable quality of the models for use in practical control systems. The average MAE, MSE, and R^2 values confirm the effectiveness of the applied approaches for different climate parameters.

The system is implemented using containerization technologies (Docker), supports standard protocols for interacting with sensors (MQTT, HTTP), and has monitoring and logging tools, which makes it suitable for deployment in real-world settings such as greenhouses, smart buildings, or industrial premises.

This study demonstrates that a microservice-based adaptive micro-climate control system can proactively maintain environmental parameters by coupling real-time IoT sensing with LSTM-driven forecasting. The modular decomposition of data ingestion, analytics, decision logic, and actuator services guarantees high scalability and fault-tolerance while enabling hot redeployment and horizontal scaling.

Bibliography

1. Hadj Abdelkader O., Bouzebiba H., Pena D., Aguiar A. P. Energy-Efficient IoT-Based Light Control System in Smart Indoor Agriculture. *Sensors*. 2023. T. 23, № 18. C. 7670. DOI: 10.3390/s23187670. PMID: 37765728; PMCID: PMC10534542
2. Lork C., Cubillas M., Ng B. K. K., Yuen C., Tan M. Minimizing electricity cost through intelligent lighting control for indoor plant factories. *IECON 2020 – The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, 18–21 October 2020. C. 297–302. DOI: 10.1109/IECON43393.2020.9255061
3. Аксак Н., Кушнарьов М., Шеліхов Ю. Інтелектуальне керування мікрокліматом ситі-ферми на основі алгоритму Q-learning. *Advanced Information Technology*. 2024. Т. 1, № 3. С. 12–24. DOI: 10.17721/AIT.2024.1.02
4. Шеліхов Ю. О., Аксак Н. Г. Архітектура системи контролю мікроклімату у замкнутому приміщенні. *Grail of Science*. 2023. № 24 С. 296–301. DOI: <https://doi.org/10.36074/grail-of-science.17.02.2023.055>
5. Afzali S., Bao Y., Van Iersel M. W., Velni J. M. Optimal lighting control in greenhouses using Bayesian neural networks for sunlight prediction. *2022 European Control Conference (ECC)*, London, 12–15 July 2022. C. 1140–1145. DOI: <https://doi.org/10.48550/arXiv.2205.03733>
6. Afzali S., Mosharafian S., Van Iersel M. W., Velni J. M. Optimal lighting control in greenhouses equipped with high-intensity discharge lamps using reinforcement learning. *2021 American Control Conference (ACC)*, New Orleans, 25–28 May 2021. C. 1414–1419. DOI: 10.23919/ACC50511.2021.9482964
7. Mosharafian S., Afzali S., Weaver G. M., Van Iersel M., Velni J. M. Optimal lighting control in greenhouse by incorporating sunlight prediction. *Computers and Electronics in Agriculture*. 2021. T. 188. C. 106300. DOI: <https://doi.org/10.1016/j.compag.2021.106300>
8. Ajagekar A., Mattson N. S., You F. Energy-efficient AI-based control of semi-closed greenhouses leveraging robust optimization in deep reinforcement learning. *Advances in Applied Energy*. 2023. T. 9. C. 100119. DOI: <https://doi.org/10.1016/j.adapen.2022.100119>

9. Van Laatum B., Van Henten E. J., Boersma S. GreenLight-Gym: A reinforcement learning benchmark environment for greenhouse crop production control. *arXiv preprint*. 2024. arXiv:2410.05336. DOI: <https://doi.org/10.48550/arXiv.2410.05336>
10. Hati A. J., Singh R. R. Smart indoor farms: Leveraging technological advancements to power a sustainable agricultural revolution. *AgriEngineering*. 2021. T. 3, № 4. C. 728–767. DOI: <https://doi.org/10.3390/agriengineering3040047>
11. Lozano-Castellanos, L. F., Navas-Gracia, L. M., Lozano-Castellanos, I. C., Correa-Guimaraes, A. Technologies Applied to Artificial Lighting in Indoor Agriculture: A Review. *Sustainability*. 2025. T. 17, № 7. C. 3196. DOI: <https://doi.org/10.3390/su17073196>

References

1. Hadj Abdelkader, O., Bouzebiba, H., Pena, D., & Aguiar, A. P. (2023). Energy-efficient IoT-based light control system in smart indoor agriculture. *Sensors*, 23(18), 7670. <https://doi.org/10.3390/s23187670> [in English].
2. Lork, C., Cubillas, M., Ng, B. K. K., Yuen, C., & Tan, M. (2020). Minimizing electricity cost through intelligent lighting control for indoor plant factories. In *IECON 2020 – The 46th Annual Conference of the IEEE Industrial Electronics Society* (pp. 297–302). <https://doi.org/10.1109/IECON43393.2020.9255061> [in English].
3. Aksak, N., Kushnaryov, M., & Shelikhov, Y. (2024). Intelktualne keruvannia mikroklimatom siti-fermy na osnovi alhorytmu Q-learning [The intelligent control of the city-farm microclimate based on the Q-learning algorithm]. *Advanced Information Technology*, 1(3), 12–24. <https://doi.org/10.17721/AIT.2024.1.02> [in Ukrainian].
4. Shelikhov, Y. O., & Aksak, N. H. (2023). Arkhitektura systemy kontroliu mikroklimatu u zamknutomu prymishchenni [Architecture of a microclimate control system in an enclosed space]. *Grail of Science*, 24, 296–301. <https://doi.org/10.36074/grail-of-science.17.02.2023.055> [in Ukrainian].
5. Afzali, S., Bao, Y., Van Iersel, M. W., & Velni, J. M. (2022). Optimal lighting control in greenhouses using Bayesian neural networks for sunlight prediction. In *2022 European Control Conference (ECC)* (pp. 1140–1145). <https://doi.org/10.48550/arXiv.2205.03733> [in English].
6. Afzali, S., Mosharafian, S., Van Iersel, M. W., & Velni, J. M. (2021). Optimal lighting control in greenhouses equipped with high-intensity discharge lamps using reinforcement learning. In *2021 American Control Conference (ACC)* (pp. 1414–1419). <https://doi.org/10.23919/ACC50511.2021.9482964> [in English].
7. Mosharafian, S., Afzali, S., Weaver, G. M., Van Iersel, M., & Velni, J. M. (2021). Optimal lighting control in greenhouse by incorporating sunlight prediction. *Computers and Electronics in Agriculture*, 188, 106300. <https://doi.org/10.1016/j.compag.2021.106300> [in English].
8. Ajagekar, A., Mattson, N. S., & You, F. (2023). Energy-efficient AI-based control of semi-closed greenhouses leveraging robust optimization in deep reinforcement learning. *Advances in Applied Energy*, 9, 100119. <https://doi.org/10.1016/j.adapen.2022.100119> [in English].
9. Van Laatum, B., Van Henten, E. J., & Boersma, S. (2024). GreenLight-Gym: A reinforcement learning benchmark environment for greenhouse crop production control. arXiv:2410.05336. <https://doi.org/10.48550/arXiv.2410.05336> [in English].
10. Hati, A. J., & Singh, R. R. (2021). Smart indoor farms: Leveraging technological advancements to power a sustainable agricultural revolution. *AgriEngineering*, 3(4), 728–767. <https://doi.org/10.3390/agriengineering3040047> [in English].
11. Lozano-Castellanos, L. F., Navas-Gracia, L. M., Lozano-Castellanos, I. C., & Correa-Guimaraes, A. (2025). Technologies applied to artificial lighting in indoor agriculture: A review. *Sustainability*, 17(7), 3196. <https://doi.org/10.3390/su17073196> [in English].

Шеліхов Юрій Олександрович – аспірант кафедри інформаційних систем і технологій Харківського національного університету радіоелектроніки. E-mail: yurii.shelikhov@nure.ua, ORCID: 0009-0009-8970-6571.

Аксак Наталія Георгіївна – д.т.н., професор кафедри інформаційних систем і технологій Харківського національного університету радіоелектроніки. E-mail: nataliia.axak@nure.ua, ORCID: 0000-0001-8372-8432.

Shelikhov Yurii Oleksandrovych – Postgraduate Student at the Department of Information Systems and Technologies of the Kharkiv National University of Radio Electronics. E-mail: yurii.shelikhov@nure.ua, ORCID: 0009-0009-8970-6571.

Axak Nataliia Georgiivna – Doctor of Technical Sciences, Professor at the Department of Information Systems and Technologies of the Kharkiv National University of Radio Electronics. E-mail: nataliia.axak@nure.ua, ORCID: 0000-0001-8372-8432.



Дата надходження статті: 17.10.2025

Дата прийняття статті: 09.12.2025

Опубліковано: 30.12.2025