https://doi.org/10.31891/2219-9365-2026-87-3
UDC 004.77+631.4

BESHTA Liliya
Dnipro University of Technology
https://orcid.org/0000-0001-5041-0962
e-mail: Beshta.l.v@nmu.one
SERGIEIEVA Kateryna
Dnipro University of Technology
https://orcid.org/0000-0001-7345-2209
e-mail: sergieieva.k.l@nmu.one

MODULAR ARCHITECTURE AND SERVICE STACK OF LORAWAN SYSTEM FOR TELEMETRY AND MONITORING OF GREENHOUSE ENVIRONMENT

The article presents the development and experimental testing of a local web monitoring system based on LoRaWAN technology, designed for use in greenhouses and similar infrastructure facilities. A microservice architecture deployed in a Docker environment on a single-board computer or server is proposed, providing autonomy, local data storage, and independence from cloud platforms. The system includes ChirpStack as the LoRaWAN server, InfluxDB for time-series storage, and Grafana for data visualization. Key parameters for assessing radio channel quality (RSSI, SNR, fCnt) have been systematized, and their acceptable values for the indoor environment have been determined. Experimental studies have confirmed the stability of communication at distances of up to 300 m in the presence of interference, the absence of packet loss, and the correctness of telemetry processing. The results obtained confirm the suitability of the proposed architecture for the practical implementation of autonomous IoT monitoring systems.

Keywords: LoRaWAN, IoT system, ChirpStack, RSSI, SNR, SF, Grafana, InfluxDB, Docker, Docker stack, Docker Compose, Portainer, local architecture, greenhouse, microclimate monitoring.

БЕШТА Лілія, СЕРГЄЄВА Катерина
Національний технічний університет «Дніпровська політехніка»

МОДУЛЬНА АРХІТЕКТУРА ТА СТЕК СЕРВІСІВ LORAWAN-СИСТЕМИ ДЛЯ ТЕЛЕМЕТРІЇ ТА МОНІТОРИНГУ ТЕПЛИЧНОГО СЕРЕДОВИЩА

У статті представлено розробку та експериментальну перевірку локальної системи веб-моніторингу на базі технології LoRaWAN, орієнтованої на використання в тепличних та подібних інфраструктурних об'єктах. Запропоновано мікросервісну архітектуру, розгорнуту в середовищі Docker на одноплатному комп'ютері або сервері, що забезпечує автономність, локальне зберігання даних і незалежність від хмарних платформ. Система включає ChirpStack як LoRaWAN-сервер, InfluxDB для зберігання часових рядів та Grafana для візуалізації даних. Систематизовано ключові параметри оцінки якості радіоканалу (RSSI, SNR, fCnt) та визначено їх допустимі значення для внутрішнього середовища. Експериментальні дослідження підтвердили стабільність зв'язку на відстанях до 300 м за наявності перешкод, відсутність втрат пакетів і коректність обробки телеметрії. Отримані результати засвідчують придатність запропонованої архітектури для практичного впровадження автономних IoT-систем моніторингу.

Ключові слова: LoRaWAN, IoT-система, ChirpStack, RSSI, SNR, SF, Grafana, InfluxDB, Docker, Docker-стек, Docker Compose, Portainer, локальна архітектура, теплиця, моніторинг мікроклімату.

Стаття надійшла до редакції / Received 30.04.2026
Прийнята до друку / Accepted 14.06.2026
Опубліковано / Published 10.09.2026

PROBLEM STATEMENT

Monitoring the microclimate in greenhouse farms is a key factor in ensuring stable yields and predictable growth rates for agricultural crops. A greenhouse as a technological facility operates in a closed environment, where the parameters of temperature, relative humidity, ventilation, and air exchange change dynamically and significantly affect the biological processes of plant growth.

Traditional control methods based on manual measurements or the use of autonomous household sensors do not provide sufficient temporal and spatial detail of data. They do not allow for the formation of continuous time series, the detection of hidden patterns, or rapid response to changes in environmental conditions. In this regard, there is a pressing need to develop automated telemetry data-collection systems for continuous monitoring and long-term storage of information [1].

The specific operating conditions of greenhouses – high humidity, metal frames, and large structures – significantly limit the use of wired systems and short-range wireless technologies. Wired systems are complex to install, sensitive to high humidity, corrosion, and mechanical damage, and require significant maintenance costs. Short-range wireless technologies, such as Wi-Fi, Bluetooth, or ZigBee, are characterized by limited range, unstable

operation in high humidity and the presence of metal structures, and increased energy consumption by end devices. Mobile communication technologies (NB-IoT, LTE-M) depend on telecom operators, have unstable coverage in rural areas, and are often economically unfeasible for small farms. In this context, LoRaWAN technology [2, 3] is one of the most promising solutions for greenhouse monitoring. It enables the transmission of small amounts of data over long distances with minimal energy consumption, enabling the use of autonomous sensors with a battery life of several years. However, the effectiveness of LoRaWAN systems in real agricultural conditions is determined not only by the physical characteristics of the radio channel, but also by the architecture of data exchange between sensor nodes, gateways, and server infrastructure [4].

A significant challenge is choosing a LoRaWAN network management platform. Currently, there are commercial full-cycle LoRaWAN platforms available on the market, including Actility ThingPark, Kerlink Wanesy, Loriot, and Tata Communications LoRa Platform. These solutions provide comprehensive network-level support, device management, integration with cloud services, and scalability at the national or global level. At the same time, such platforms are commercial, require licensing, are designed to work in a cloud environment, and provide limited user access to internal data processing mechanisms. For local greenhouse farms, this creates dependence on external infrastructure, increases operating costs, and complicates the system's adaptation to the facility's specific conditions [5].

An alternative approach is to use free global LoRaWAN platforms, such as The Things Network (TTN) [6]. TTN is an open community network that provides free device connectivity and management through a convenient web interface. The advantages of TTN include support for the LoRaWAN 1.0.x standard, a built-in MQTT broker, and a low entry threshold for prototype deployment. However, the use of TTN in agricultural applications has several limitations: no offline mode, complete reliance on external server infrastructure, and a fair use policy that imposes limits on the number and frequency of transmitted packets. This makes it impossible to use TTN in scenarios where guaranteed autonomy and complete control over data are required.

In this regard, it is essential to build a local LoRaWAN infrastructure based on the open ChirpStack platform, which provides complete control over all stages of data exchange. ChirpStack allows you to deploy a network server, application server, and integration mechanisms within a local computing platform without using external cloud services. This is fundamentally important for greenhouse farms operating under unstable or no Internet access [7].

The interaction between sensors and the monitoring system in such architecture follows the standard LoRaWAN scheme (Fig. 1). At the first stage, the sensor node forms a radio packet and transmits it to the LoRaWAN gateway. Transmission is carried out using Chirp Spread Spectrum modulation, which ensures high noise immunity and communication stability in high-humidity conditions and in the presence of signal reflections from metal structures. LoRaWAN uses a "star with multiple receivers" topology. This means that several gateways can hear each sensor.

In the second stage, the gateway transmits the received packet to the ChirpStack network server via the IP network. The gateway does not interpret the data but only adds service information and metadata. The next stage is the network server's processing of the packet, which authenticates the device, processes MAC commands, decodes the payload, and aggregates metadata.

The data is then transmitted via an MQTT broker to a time-series database, which allows large arrays of telemetry data to be stored efficiently. The final level is a visualization system that displays data in the form of graphs and tables in near real time [8].

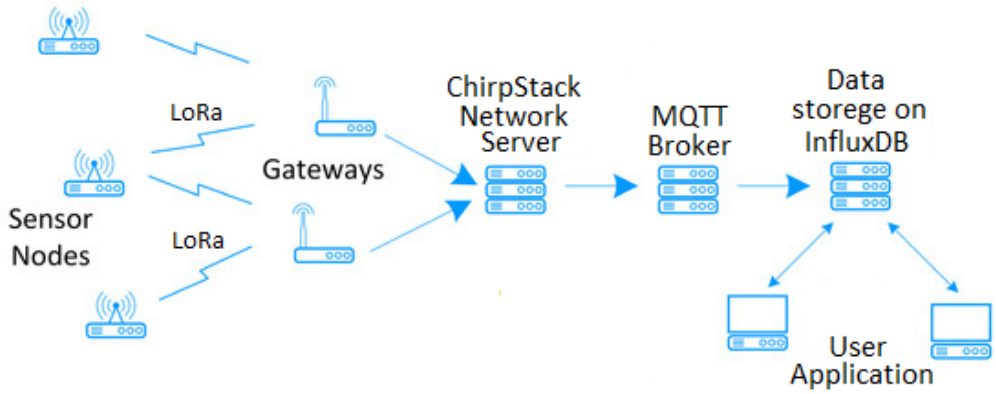

Fig. 1. Data processing workflow

Of particular scientific and practical value is the fact that LoRaWAN packets contain not only the values of the measured parameters, but also an extended set of metadata, including raw PHY payload data, device identifier (DevEUI), connection status, signal strength (RSSI), number of transmissions (SNR), as well as information about frequency and spreading factor (SF). This allows analyzing not only the microclimate conditions, but also the quality

of the radio channel in real time, assessing the impact of the greenhouse's design features on signal propagation, and optimizing network parameters.

Thus, the scientific problem lies in the lack of sufficiently researched autonomous LoRaWAN systems for greenhouse farms that would combine complete control over data exchange processes, the ability to analyze radio channel metadata in detail, and independence from external cloud platforms. Solving this problem requires developing a unified method for deploying the server component of the web monitoring system, enabling reproducible deployment of the microservice architecture using Docker containers. The proposed approach allows the creation of an entirely local monitoring system that operates independently of Internet access in rural areas and stores telemetry data locally at the greenhouse owner's premises. This, in turn, increases the reliability, manageability, and practical value of IoT systems for agricultural applications and lays the foundation for further automation of greenhouse processes.

RESEARCH ANALYSIS

LoRaWAN in greenhouse monitoring has proven its effectiveness across both traditional data-collection architectures and more innovative approaches. Research by Singh et al. [4] has shown that the technology can provide long-term monitoring of the microclimate in greenhouses, and radio channel data, in particular RSSI, can be used not only for network diagnostics but also for predicting biological parameters such as plant growth. This opens up new opportunities for analytics without additional sensors [9]. Practical implementations using open tools such as Grafana and InfluxDB confirm that LoRaWAN can be easily integrated into modern visualization platforms, and systems for multiple greenhouses demonstrate scalability and the possibility of centralized management [4].

From a technical standpoint, LoRaWAN demonstrates resistance to interference in complex environments, particularly in greenhouses with metal and glass structures. Optimizing modulation parameters, such as spreading factor or bandwidth, is crucial for ensuring stable data transmission in rural areas, where distances are long and radio interference is common. In a broader agricultural context, LoRaWAN is distinguished by its cost-effectiveness and energy efficiency. In precision farming, this technology enables the creation of private infrastructure for monitoring soil parameters, reducing dependence on external services, and making the system more autonomous [1].

OBJECTIVE

The goal of this work is to create and experimentally test a local system for monitoring greenhouse environment parameters using LoRaWAN technology, based on an open, scalable software solution. It is necessary to develop a data collection system architecture based on the ChirpStack → InfluxDB → Grafana stack and to investigate radio channel properties for telemetry transmission in greenhouse complexes. To achieve the goal, it is necessary to develop a unified method for deploying a web monitoring architecture for containerized services in Docker, integrating InfluxDB and Grafana for data collection and analysis. Perform experimental field measurements in an open environment and in conditions close to those in a greenhouse.

TASKS

To achieve the research goal, it is planned to select the parameters of the wireless network, build the server infrastructure for the monitoring system based on ChirpStack, InfluxDB, and Grafana, and implement decoding of LoRaWAN packet payloads using proprietary codecs. A separate task is to deploy the system as a completely local solution, independent of cloud services. As part of the experimental part of the research, it is planned to measure the quality of radio communication at different distances, analyze RSSI and SNR indicators, check the stability of transmission, and the absence of packet loss. The final stage is to evaluate the suitability of the developed system for use in greenhouses and similar agricultural facilities.

COMPONENTS OF SMART FARMING TECHNOLOGY

Assessing the quality of LoRaWAN communication in greenhouse conditions requires accounting for the environment's specific characteristics, including high humidity, the presence of metal structures, and variable plant density. Such factors cause additional signal loss and changes in the noise level, so the analysis must be based on a set of physical and operational parameters of the radio channel.

The main indicators of the physical layer are RSSI and SNR. RSSI characterizes the level of the received signal, while SNR reflects the signal's resistance to interference and determines the likelihood of correct packet decoding. For LoRaWAN, it is essential that the system can operate even at negative SNR values, a critical advantage in greenhouse environments [11].

The settings for Spreading Factor, Bandwidth, and Coding Rate affect the connection's sensitivity and stability and should be considered when optimizing the radio channel (Table 1). Their selection determines the trade-off between range, transmission speed, and energy efficiency of the nodes.

Table 1

Key metrics for evaluating LoRaWAN communication quality in a greenhouse

Parameter	Optimal level	Acceptable level	Critical level
RSSI (dBm)	–40…–90	–90…–115	< –120
SNR (dB)	+5…+15	–5…+5	< –10
Packet Loss (%)	< 1	1–5	> 10

In a greenhouse environment, RSSI values up to –115 dBm and SNR values up to –10 dB are acceptable, provided that packet loss does not exceed 5%. If there is a simultaneous decrease in RSSI, negative SNR, and an increase in the packet loss rate above 10%, the connection should be considered unstable and require parameter correction (increasing SF, increasing TX Power, or optimizing gateway placement).

A comprehensive assessment should be based not on a single indicator, but on a combination of RSSI, SNR, and Packet Loss Rate, as this combination allows for the correct determination of the boundary conditions for the stable operation of the LoRaWAN network in a greenhouse environment.

LoRaWAN architecture of the greenhouse monitoring system

The system under development is a local multi-level LoRaWAN architecture designed to collect, transmit, process, and visualize telemetry data in a greenhouse environment. The architecture is built on the principle of a complete information processing cycle – from the sensor node to the user web interface (Fig. 2). All server components are deployed locally on a Raspberry Pi single-board computer in a Docker environment, which ensures autonomous operation, service isolation, and independence from external cloud platforms.

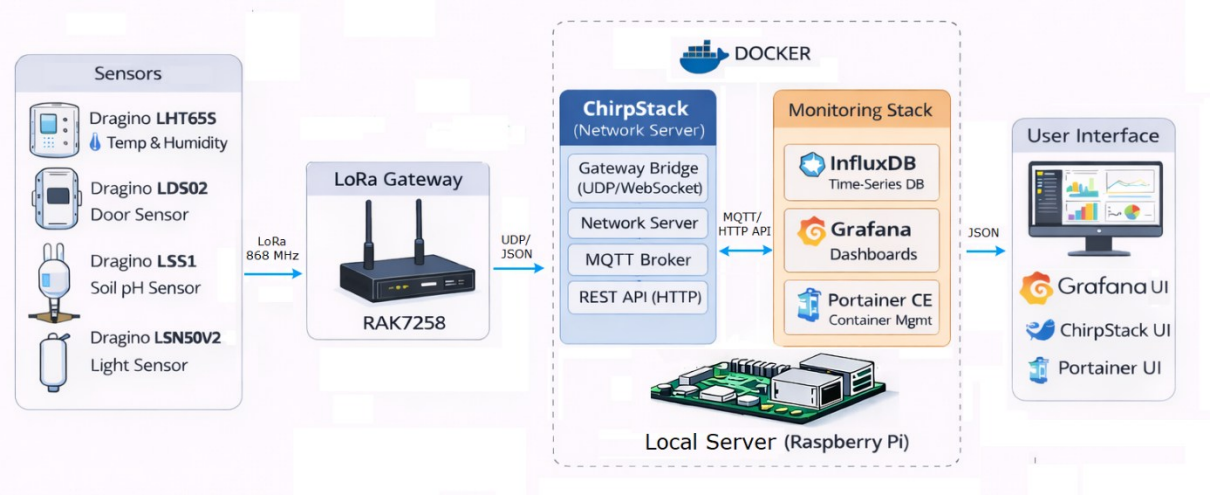

Fig. 2. Topology of the monitoring system based on the Chirpstack framework

The sensor level is represented by LoRaWAN nodes that measure microclimate parameters and form PHY payloads in accordance with the LoRaWAN specification. Data transmission occurs in the EU868 band using Chirp Spread Spectrum modulation, which ensures energy-efficient communication and resistance to interference in the complex conditions of a greenhouse, where metal structures and high humidity are present. The nodes operate in Class A mode with OTAA activation and transmit data periodically or on an event basis.

The communication layer is provided by a LoRaWAN gateway, which acts as a bridge between the radio network and the IP infrastructure. The gateway receives radio packets along with radio channel metadata, including RSSI, SNR, frequency, and gateway ID. Then it transmits them to the network server via the Semtech UDP Packet Forwarder protocol. No logical data processing is performed at this stage, which is consistent with the standard LoRaWAN model.

The ChirpStack network server implements protocol packet processing, including integrity checking, MAC command processing, adaptive data rate (ADR) control, and downlink message formation (Fig. 3). After network processing, the data is transferred to the application server, where the frm-payload is decoded using proprietary codecs, converted into a structured JSON format, and prepared for storage.

Telemetry data is stored in the InfluxDB time-series database, which is optimized for working with streaming time series. The database operates locally, ensuring the owner's control over the information and the system's ability to operate without Internet access. Visualization is implemented using Grafana, which receives data directly from InfluxDB and provides dashboards for monitoring both microclimate parameters and radio channel technical characteristics.

The system's administrative level is implemented using containerized components in a Docker environment, forming a microservice architecture. Within it, the network server, application server, database, and visualization system function as isolated but logically interconnected services. This approach provides centralized service management, simplifies deployment and scaling, facilitates system technical support, and ensures configuration reproducibility during platform updates or migrations.

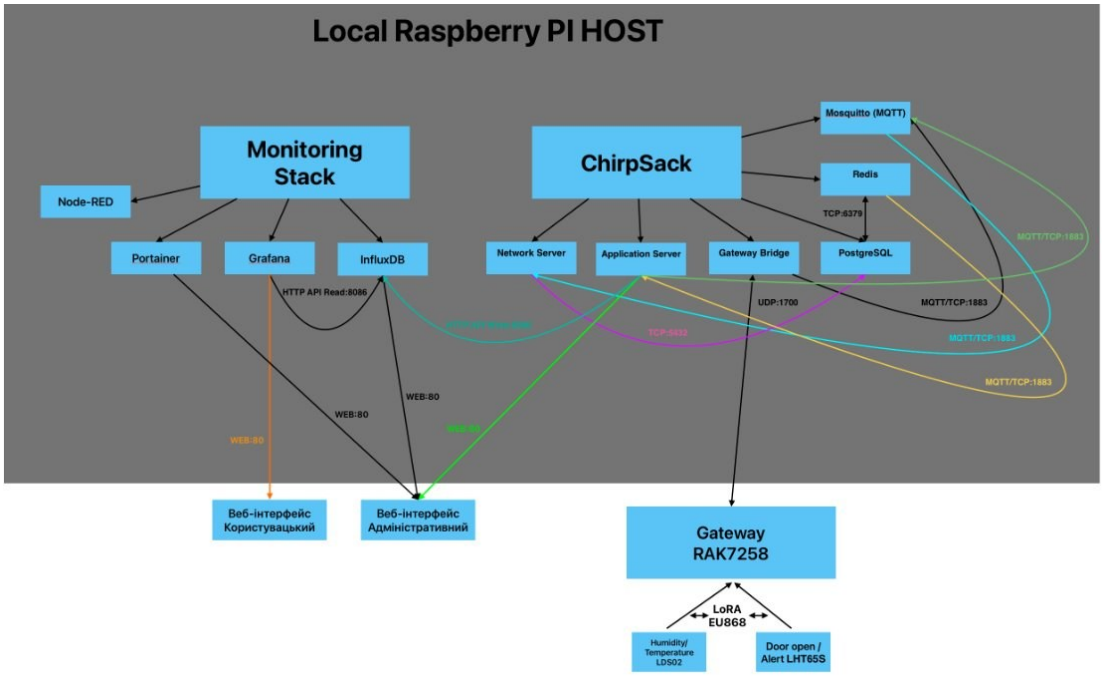

Fig. 3. Functional structure diagram of the greenhouse monitoring system

Development of an administrative-level Bash script for Docker Stack Control Center

The software part of the system is implemented as a containerized architecture that can be deployed on a single-board computer (Raspberry Pi), a virtual machine, or a physical server. It is based on two independent Docker stacks: the ChirpStack Stack, which ensures the operation of the LoRaWAN server, and the Monitoring Stack, which handles data storage, visualization, and administration.

ChirpStack Stack implements the full cycle of LoRaWAN message processing. Network Server receives uplink/downlink packets, manages device sessions, performs MIC verification, and cryptographic processing. Gateway Bridge or Basic Station provides interaction with gateways via UDP (port 1700) or WebSocket. Internal messaging between services is performed via an MQTT broker, and the REST API (port 8090) provides integration with external systems. The stack configuration is defined in the `chirpstack.yml` file, which is generated by a Bash script. The file specifies container images (`chirpstack/chirpstack:4`, `postgres:14-alpine`, `redis:7-alpine`), ports (8080 for the web interface, 8090 for the REST API, 1883 for MQTT, 1700/udp for the gateway), and named volumes (`postgresqldata, redisdata`) that store data between container restarts.

The Monitoring Stack includes InfluxDB, Grafana, and Portainer CE. InfluxDB is used as a time-series database to store telemetry with timestamps, device tags, and measured parameters. Grafana provides dashboards to display climate indicators, sensor status, and radio channel parameters. Portainer simplifies container management via a web interface. Service settings are described in the `monitoring.yml` file, which is generated by a Bash script. Data is stored in `/home/docker-data/monitoring`, with separate subdirectories for each component (`influxdbdata, grafanadata, portainerdata, configuration/`). For Grafana, the script automatically sets access rights.

This approach ensures modularity, configuration reproducibility, and the ability to deploy a local LoRaWAN monitoring system autonomously without depending on cloud infrastructure.

To automate deployment and system management, a universal Bash script, ChirpStack_Monitoring_Docker.sh, has been developed, providing a unified mechanism for installing, configuring, and maintaining two Docker stacks: the LoRaWAN server (ChirpStack Stack) and the monitoring (Monitoring Stack). The Bash script automates the creation of directories, the generation of Docker Compose files, and the launch, update, backup, and rollback of Docker stacks. The script is designed for use on both the Raspberry Pi single-board computer and on Linux virtual or physical servers.

At the initial stage, an automatic check of system dependencies (Docker, Docker Compose v2, unzip, jq) is performed. If they are missing, the user is prompted to install the necessary components, which minimizes the risk of incorrect deployment. Next, a standardized directory structure is established to store configurations, container data, and backups, ensuring service isolation and simplifying system migration.

The script automatically creates the following base directories:

- ✓ /home/docker-data/monitoring – for Monitoring Stack;
- ✓ /home/docker-data/chirpstack – for ChirpStack Stack;
- ✓ /home/docker-backup – for stack backups.

This way, all stack data is stored in a single location, making it easier to back up and migrate the system.

If the directory with ChirpStack configurations (/home/docker-data/chirpstack/configuration/chirpstack) does not yet exist, the script downloads it from the official chirpstack/chirpstack-docker repository on GitHub, unpacks the archive, and copies the configurations to the working directory.

The script automatically generates `Docker Compose files for both stacks, including` container image definitions, network settings, open ports, and named volumes. Separate directories with the correct access rights are created for the Monitoring Stack (Fig. 4). For Grafana, a .stack_env file is created to store the login and password, which are generated automatically (using openssl rand -base64 12). For ChirpStack Stack, an .env file is created with the path to the base directory (CHIRP_BASE) fixed.

An interactive control menu is provided, allowing you to start, restart, update, or delete each stack separately, view their current status, and perform maintenance without manually editing configurations. A separate "status bar" displays the server's local and external IP addresses, the URLs of the web interfaces (Grafana, ChirpStack UI, Portainer, REST API), the Grafana login and password, and the container status summary.

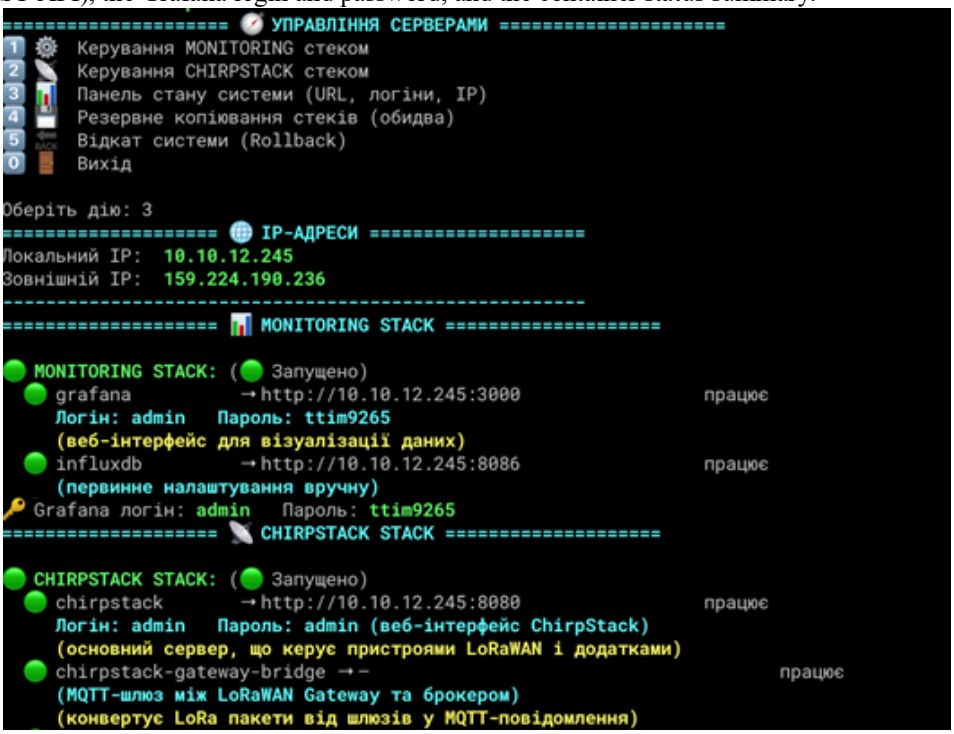

Fig. 4. Example of system status panel output

The script implements a backup mechanism that includes saving Docker images to the images.tar file, archiving named volumes (in particular, PostgreSQL and Redis databases via a temporary busybox container), and copying working directories. Automatic limitation of the number of archives and a rollback function to the previous system state are supported.

This approach ensures deployment reproducibility, increases operational reliability, and simplifies the administration of the local LoRaWAN infrastructure.

DISCUSSION

The experimental part of the work aimed to comprehensively test the built LoRaWAN system under conditions close to real operation. The purpose of the study was not only to confirm the fact of data transmission from sensors, but also to evaluate the behavior of the radio channel at different distances and in the presence of interference, analyze the continuity of packet transmission, the correctness of `frm_payload` decoding, as well as the stability of the server infrastructure during continuous telemetry collection.

The study used LDS02 (door sensor) and LHT65S-E3 (temperature and humidity sensor) end devices, a RAK7258 (EU868, Semtech UDP Packet Forwarder) gateway, and a locally deployed server architecture based on ChirpStack v4, InfluxDB 2.x, and Grafana in a Docker environment (Fig. 5). Payload decoding was performed using proprietary JavaScript codecs. Radio parameters: 868 MHz, 125 kHz bandwidth, SF7, Class A.

Three scenarios were modeled:

1. 15–40 m without obstacles (conditionally open space);
2. 60–150 m with one wall and heavy equipment;
3. 200–300 m deep inside a building (4–5 brick walls and doors).

In each case, the RSSI, SNR, and frame counter (fCnt) values were recorded.

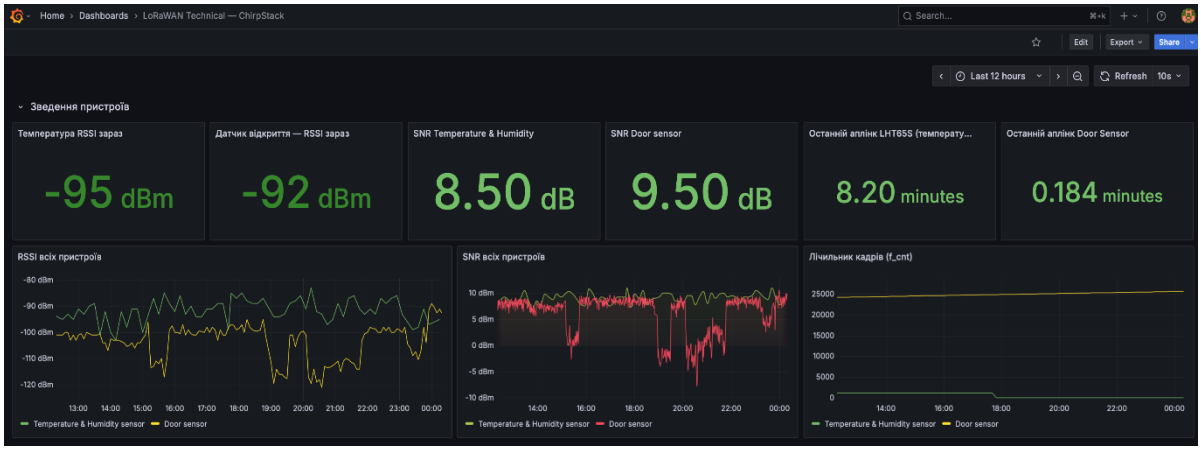

Fig. 5. Example of a Grafana dashboard with technical data from LoRaWAN sensors

In the first scenario, RSSI of –88 dBm and SNR of 9.8 dB were obtained, which correspond to the "comfort zone" for LoRaWAN and indicate a significant margin of channel stability. In the second scenario, RSSI decreased to –97 dBm and SNR to 8.5 dB, as expected due to the signal passing through a wall and additional obstacles. In the third scenario, with the maximum number of barriers, RSSI –99 dBm and SNR 6.8 dB were recorded.

The decrease in signal level and signal-to-noise ratio is gradual and does not exceed the values considered acceptable for LoRaWAN in indoor environments in the literature. Even in the worst-case scenario, the parameters obtained remain within the range that ensures stable communication at SF7. This confirms the technology's ability to maintain a reliable channel in greenhouses or warehouses with structural obstacles.

The behavior of the fCnt frame counter was analyzed separately. A consistent increase in values (32 → 36 → 40) without gaps was recorded, indicating no packet loss during the transition between scenarios. The absence of "jumps" or resets also confirms the devices' stability and the absence of restarts. Thus, even with deteriorating radio conditions, the system ensures full delivery of uplink messages to the server.

In parallel with the radio channel assessment, the correct interpretation of telemetry was verified. Temperature and humidity values changed smoothly and corresponded to the environment's physical conditions; the door status and opening counter responded appropriately to mechanical actions; battery parameters showed gradual dynamics without abnormal jumps. This confirms the correctness of the implemented `frm_payload` decoding codecs and the correctness of data transmission in JSON format.

The data was continuously stored in InfluxDB, and the time series in Grafana did not contain any "gaps," which further indicates the absence of packet loss and the stability of the ChirpStack → InfluxDB → Grafana server chain. The load created by the experimental transmission did not cause any malfunctions in the containerized infrastructure.

Comparison with literature data

The obtained RSSI (–88…–99 dBm) and SNR (6.8–9.8 dB) values are consistent with published LoRaWAN results for distances up to 100 m in rooms with moderate interference. The absence of frame loss at these parameters also corresponds to the expected characteristics of the technology when using SF7. Thus, the experimental data do not contradict the theoretical models of signal propagation in the 868 MHz band.

CONCLUSIONS

The local LoRaWAN web monitoring system designed for use in greenhouses and similar facilities has been developed and experimentally tested. The proposed architecture is built on a microservice principle and deployed in a Docker environment on a Raspberry Pi or server, ensuring autonomy, scalability, and independence from external cloud platforms.

The system includes ChirpStack as the LoRaWAN server, InfluxDB for time-series storage, and Grafana for data visualization. A unified automated stack-deployment method with backup and recovery capabilities has been implemented, increasing operational reliability and configuration reproducibility.

Key parameters for assessing radio channel quality (RSSI, SNR, fCnt) have been systematized, and their acceptable ranges for the greenhouse environment have been determined. Experimental results across various conditions (open space, presence of obstacles, building depth) showed stable communication at distances of up to 300 m at SF7, with no packet loss, and compliance of the obtained indicators with data reported in scientific sources.

Thus, the proposed local LoRaWAN infrastructure has proven its suitability for practical implementation, ensuring reliable collection, storage, and analysis of telemetry data without dependence on external services.

It is also essential that the result is not just software. The system can be scaled to accommodate more greenhouses or expanded with additional sensors. The proposed local architecture does not require external services

and can operate autonomously, a significant advantage for agricultural facilities where the internet is either unstable or unavailable.

References

1. Dragoș-Ioan Săcăleanu, D.-I., Matache, M.-G., Roșu, Ș.-G., & Florea, B.-C. (2024). IoT-enhanced decision support system for real-time greenhouse microclimate monitoring and control. *Technologies, 12*, 230. https://www.mdpi.com/2227-7080/12/11/230
2. LoRa Alliance. (n.d.). *LoRaWAN specification 1.0.4*. https://lora-alliance.org/resource_hub/lorawan-104-specification-package/
3. Bor, M., Vidler, J. E., & Roedig, U. (2016). *LoRa for the Internet of Things*. https://eprints.lancs.ac.uk/id/eprint/77615/
4. Singh, R. K., et al. (2020). Leveraging LoRaWAN technology for precision agriculture in greenhouses. *Sensors, 20*(7), 1827. https://doi.org/10.3390/s20071827
5. CropX. (n.d.). *Farm management platform overview*. https://www.cropx.com/
6. The Things Network. (n.d.). *EU868 regional parameters*. https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/eu868/
7. ChirpStack. (n.d.). *ChirpStack documentation – Network server*. https://www.chirpstack.io/docs/
8. Muladi, Wirawan, I. M., Lestari, D., El Raka, S. C., Leksono, A. B., Qodri, F. A., & Prasetyo, S. D. (2025). ChirpStack-based LoRaWAN platform for land-sliding monitoring system. *Instrumentation Mesure Métrologie, 24*(1), 73–79. https://doi.org/10.18280/i2m.240108
9. Singh, R. K., Rahmani, M. H., Weyn, M., & Berkvens, R. (2022). Joint communication and sensing: A proof of concept and datasets for greenhouse monitoring using LoRaWAN. *Sensors, 22*(4), 1326. https://doi.org/10.3390/s22041326
10. Shilpa, B., Gupta, H. P., Jha, R. K., & Hashmi, S. S. (2024). LoRa interference issues and solution approaches in dense IoT networks: A review. *Telecommunication Systems, 87*(2), 517–539. https://doi.org/10.1007/s11235-024-01192-9
11. Hakala, I. (2015). Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks. In *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*. https://doi.org/10.15439/2015F241