INFORMATION SYSTEM FOR RESEARCHING THE ECOLOGICAL STATE OF NATURE-RESERVED TERRITORIES

Lesya Uhryn, Senior Lecturer, Mariana Baran, PhD, As. Prof,
Lviv Polytechnic National University, Ukraine;
Tõnis Teppand, Lecturer of Analytical Selection of Building Structures, Researcher;
Institute of Forestry and Rural Engineering, Estonian University of Life Sciences, Tartu, Estonia,
e-mail: Lesya.e.uhryn@lpnu.ua;
e-mail: tonis.teppand@emu.ee

https://doi.org/10.23939/istcmtm2025.02.098

Abstract. The article is dedicated to monitoring and researching the environmental condition of areas that have the status of nature-reserved. The work describes the creation of an information system with elements of artificial intelligence for studying the environmental condition of nature-reserved territories. As a result of the analysis of monitoring data, primary data for work was identified, and a database was formed. Special attention is paid to the description of the information model of the object and the indicators selected for its adequate functioning. The built-in artificial intelligence system will be trained based on the formed model to improve the speed and quality of determining the state of the reserve area. Great attention will be paid to the ability to quickly and seamlessly receive data from sensors. The template-optimized code for sending data from sensors will easily help to quickly transmit large volumes of data, and the formed dynamic user interface model will allow administrators to increase the system's capabilities depending on their needs. The chosen technologies for creating the information system, microservices, Apache Kafka event streaming platform, and Confluent Platform are analyzed. The research results are presented in the form of a developed and tested information system.

Keywords: information system, artificial intelligence, microservices, nature-reserved territories.

1. Introduction

The assessment and monitoring of environmental conditions in the nature reserves of Ukraine are extremely important in modern times. The relevance of this issue is emphasized by the ongoing deterioration of natural ecosystems, which is exacerbated by factors such as industrial activity, climate change, and military actions in the country. There is a need to study and preserve the ecological identity that exists in nature reserves. In Ukraine, where the protection of diverse flora and fauna is critically important for national and ecological sustainability, the significance of this topic becomes even more apparent. The ecological challenges associated with the conflict, including missile strikes on critical infrastructure and the release of toxic substances, have increased the vulnerability of nature reserves. This underscores the urgent need for detailed research and continuous monitoring to assess the impact on biodiversity, air quality, and overall ecological balance. Information systems with artificial intelligence effectively collect, store, systematize, and process large amounts of information and obtain the results obtained in the study of nature-reserved territories.

2. Problems and difficulties

The urgency of this research is further heightened against the backdrop of the ongoing conflict in Ukraine, a circumstance that has exacerbated the ecological challenges faced by the nation. Missile strikes targeting vital infrastructure, including oil depots and power plants, have resulted in immediate environmental consequences and pose long-term threats by releasing various hazardous substances into the air. Consequently, this significantly impedes the effective execution of research and monitoring activities, intensifying the need for a robust and adaptable monitoring system. Access to sensors from which data is collected and transmitted to the monitoring system becomes a problem. Data collection from some areas becomes limited, and sometimes unavailable.

3. Goal

The aim of the work was to develop an information system with a trained neural network for data processing and assessment of the quality of the territory, which can be integrated into various environments, the priority being nature reserves. The information system is supplemented with programs for quickly sending data from sensors to the working environment, and a dynamic user interface will allow the system administrator to easily add new indicators for research.

4. Research and development of information system

The global discourse on environmental conservation and sustainable practices further amplifies the relevance of this theme. Investigating the ecological state of nature-reserved territories in Ukraine extends beyond regional concerns, encapsulating broader ecological, cultural, and global dimensions. This work addresses the critical need for a comprehensive understanding of the current ecological state and sets the stage for informed conservation practices and sustainable environmental policies. The focal point of this study lies in emphasizing the indispensable role played by the use of information systems in research and monitoring ecological state within nature reserve areas and its crucial contribution to ongoing conservation efforts dedicated to preserving Ukraine's rich biodiversity.

A safe environment not only aligns with legislative criteria but also complies with established standards and norms, encompassing various facets such as cleanliness, resource sustainability, ecological stability, sanitary conditions, species diversity, and the ability to meet the citizens' interests. The primary thrust of this research revolves around the comprehensive evaluation of the ecological state within Ukraine's nature reserves, driven by the overarching goal of establishing an accessible and user-friendly monitoring system. This system is meticulously designed to disseminate information to the public through an online platform, thus fostering greater environmental awareness and engagement.

After conducting a thorough analysis of known systems, advantages and disadvantages were highlighted with specific features strategically selected for integration into the developed information system.

As of January 1, 2022 (the beginning of a full-scale war) there were 4 biosphere reserves and 16 state nature reserves in Ukraine. The Nature Conservation Fund ensures the preservation of lands and waters, natural complexes, objects, maintains the general ecological balance, provides background monitoring of the natural environment. In Ukraine, nature conservation funds are protected as state property, special regimes of protection, reproduction and use are applied. Ukraine considers this Fund an integral part of the world system of natural territories and objects under special protection. The nature reserve fund of Ukraine includes natural territories and objects – nature reserves, biosphere reserves, national natural parks, local landscape parks, protected areas, natural monuments, reserves, man-made objects – botanical gardens, gardens, zoos and other parks – monuments of landscape art. In connection with this Law of Ukraine, reserve funds are protected as state property, special regimes for their protection, restoration and use are established. Ukraine considers this fund to be an integral part of the world system of natural territories and objects under special protection. Reserves are nature conservation and scientific research institutions of national importance, the task of which is to preserve typical or exceptional natural complexes with all their components in their natural state and preserve the nature that occurs in them, study processes and phenomena and develop scientific foundations of nature protection. Environmental protection, efficient use of natural resources, ecological safety. Land and waters belonging to the nature reserve fund are withdrawn from economic use. The reserve is the highest reserve and a natural laboratory where comprehensive scientific research is conducted. Each large natural complex has nature conservation areas. As of January 1, 2025, the share of the nature reserve fund has significantly decreased due to the war. About 20 % of protected areas are located in occupied territory, which creates difficulties in obtaining monitoring data and requires additional efforts for their preservation and restoration [1, 2].

Based on these data and the data of the analyzed existing systems, we will develop an up-to-date, easy-to-use system for research and monitoring of nature reserves in Ukraine. The monitoring system will check data in three directions. In particular, to assess the state of the air, indicators of fine dust will be used. These are solid particles of various diameters, they can be exhaust gases, soot, dust, dirt and others. PM 2.5 is a solid particle with a diameter of 2.5 microns or less. PM 10 is a solid particle with a diameter of 2.5 microns to 10 microns. To assess water, the water Ph indicator will be used. There will also be a temperature indicator [3, 4].

Therefore, a system will be developed, the main functions of which will be monitoring and control of protected areas. The system will be divided into several parts.

The first part is software for rapid data transfer to the storage environment and data transfer to the user.

The second part is software for using this data by the system and transmitting it for direct processing and evaluation by artificial intelligence based on a pre-prepared fuzzy logic model.

The third part is an artificial intelligence system trained on the basis of the formed model to determine the state of the protected area.

The fourth part is service software that will transmit this data to the user interface.

The fifth part is a user interface with which the user will be able to check and determine the state of the environment of the nature reserve.

An information model is used that describes the parameters, values, and relationships of an objects (Fig. 1).

This diagram will display the data processing chain: data from environmental sensors enters the system and the limit values of the parameters (minimum / maximum temperature, humidity) are determined. After that, the data is analyzed and the system integrates all the blocks. The results are presented in the form of statistical parameters and visualized for further analysis.

During the work, a UML use case diagram was developed, and actor specifications and use case specifications were written. UML is a standardized modeling language used in software development to visualize, design, and document systems (Table 1, 2).

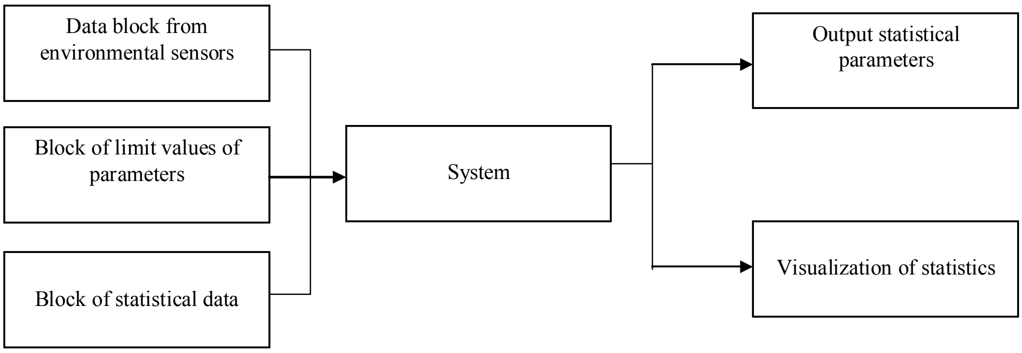

Fig. 1. Information model of the developed system

Table 1. Actor specification

Actor	Description
Admin	Admin has extended access to work with the system. Has the ability to work with sensors and the data reception and transmission system. Is responsible for loading data into the system. And training the neural network to obtain the result of the quality of the nature reserve area of Ukraine
User	Ordinary users of the system have the ability to open the system, search for data on a nature reserve for a certain period, and compare this data themselves. Artificial intelligence will analyze this data and return a valid result to the user

Table 2. Use case specifications

Usage option	Description
Data transmission system management	Administrators of protected areas have access to the admin panel, the data transfer environment, and the sensors themselves, which will upload data to the system
Working with information	Administrators of the protected area will be able to maintain the state of artificial intelligence (trained neural network), which will be responsible for checking the state of the natural protected area based on data received from the environment
Viewing the information	Viewing the information will be available to both the user and the administrators of the protected areas. Any user who wishes to view the data will be able to do so

To develop such a universal system, a significant list of technologies is required to make it as universal as possible and easy to integrate in the future. The development stages and tools can be divided into several levels: user interface development, backend system development, preparation and hosting of the data transfer environment, data consumption and processing [5].

The following technologies were chosen: Java as the primary programming language [6–8]. Apache Kafka is an event streaming platform used to collect, process, store, and integrate data at scale (Fig. 2).

It has numerous use cases, including distributed streaming, stream processing, data integration, and pub/sub messaging. The Confluent Platform is a full-scale streaming platform that makes it easy to access, store, and manage data as continuous, real-time streams.

Confluent extends the benefits of Kafka with enterprise-grade features while removing the burden of managing and monitoring Kafka. Today, over 80 % of the Fortune 100 are powered by streaming technology, and most of them use Confluent. Microservices are a popular architectural style for building systems that are resilient, highly scalable, independently deployable, and capable of rapid growth. But successful microservices architectures require a different approach to designing and building applications (Fig. 3).

A microservices architecture consists of a collection of small, autonomous services. Each service is self-contained and must implement a single business opportunity within a limited context. The limited context is a natural division in the business and provides a clear boundary within which the domain model exists. Microservices are small, independent, and loosely coupled. A single small team of developers can write and maintain a service. Each service is a separate codebase that can be managed by a small team of developers. Services can be deployed independently. A team can update an existing service without rebuilding and redeploying the entire application. Services are responsible for maintaining their own data or external state. This is different from the traditional model where a separate data layer handles data maintenance. Services communicate with each other using well-defined APIs. The internal implementation of each service is hidden from other services. Supports threaded programming and services do not need to share the same technology stack, libraries, or frameworks.

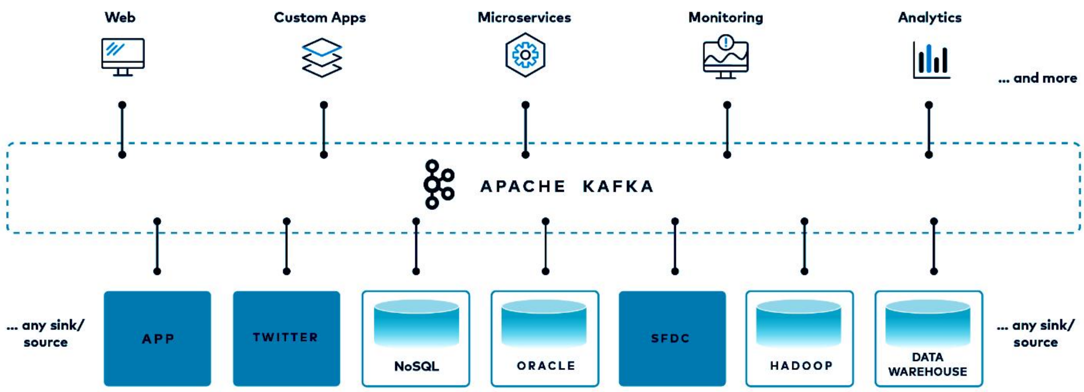

Fig. 2. Kafka streams model

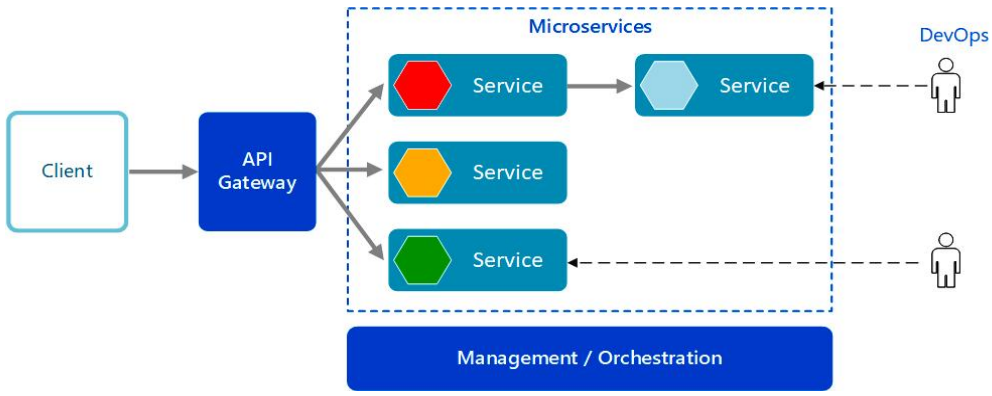

Fig. 3. Microservices architecture design

A program code has been written to send data that will be used on sensors. This code can easily be used for any sensor (Fig. 4).

An environment on the Confluent platform has been prepared for data transfer. For this purpose, a Kafka cluster has been deployed, in which topics or partitions for our data will be created (Fig. 5).

Service accounts were created to organize access to write and read data from topics in this cluster, and to create keys (passwords) and topics for receiving data from sensors. A separate topic was allocated for each sensor to maintain data integrity when reading it from the environment.

The following technologies were chosen for the development of the user interface: HTML, CSS, JavaScript

To fill the database, data from available sensors for monitoring air and water quality was used, other data was obtained from other verified resources [9]. The information system database initially consists of seven tables: quality_level, nature_territory, sensors, carbon_monoxide_level_sensor, water_polution_sensor, air_qulityindex_sensor, water_smell_sensor.

quality_level: this table will contain the quality level for a specific measured parameter (e. g. air quality level, water quality level, etc.), linked to other tables;

nature_territory: this table contains information about the natural territories where the sensors are located, linked to the sensors table;

sensors: a main table containing general information about all sensors. Contains data such as unique identifiers, locations, sensor types and other detailed information;

carbon_monoxide_level_sensor: a table containing specific data for carbon monoxide level sensors, linked to the sensors table;

water_pollution_sensor: A table with specific data for water pollution sensors, linked to the sensors table;

air_qulityindex_sensor: A table with specific data for sensors that measure the overall air quality index, linked to the sensors table;

water_smell_sensor: A table with specific data for sensors that monitor water odor, linked to the sensors table.

The interaction between these tables is provided by keys that refer to unique sensor identifiers and other relationships.

For example, the carbon_monoxide_level_sensor, water_pollution_sensor, air_qulityindex_sensor, and water_smell_sensor tables can have foreign keys that refer to identifiers in the sensors table, indicating which sensor is performing the measurement.

```
public class NatureProducer
{

    public static Properties loadConfig(final String configFile) throws IOException
    {
        if (!Files.exists(Paths.get(configFile)))
        {
            throw new IOException(configFile + " not found.");
        }
        final Properties cfg = new Properties();
        try (InputStream inputStream = new FileInputStream(configFile))
        {
            cfg.load(inputStream);
        }
        return cfg;
    }

    public static void sendData(String topic, String value) throws IOException
    {
        final Properties props = loadConfig( configFile: "client.properties");
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        final Producer<String, String> producer = new KafkaProducer<>(props);
        producer.send(new ProducerRecord<>(topic, UUID.randomUUID().toString(), value));
        producer.close();
    }
```

Fig. 4. Sensor data transmission code

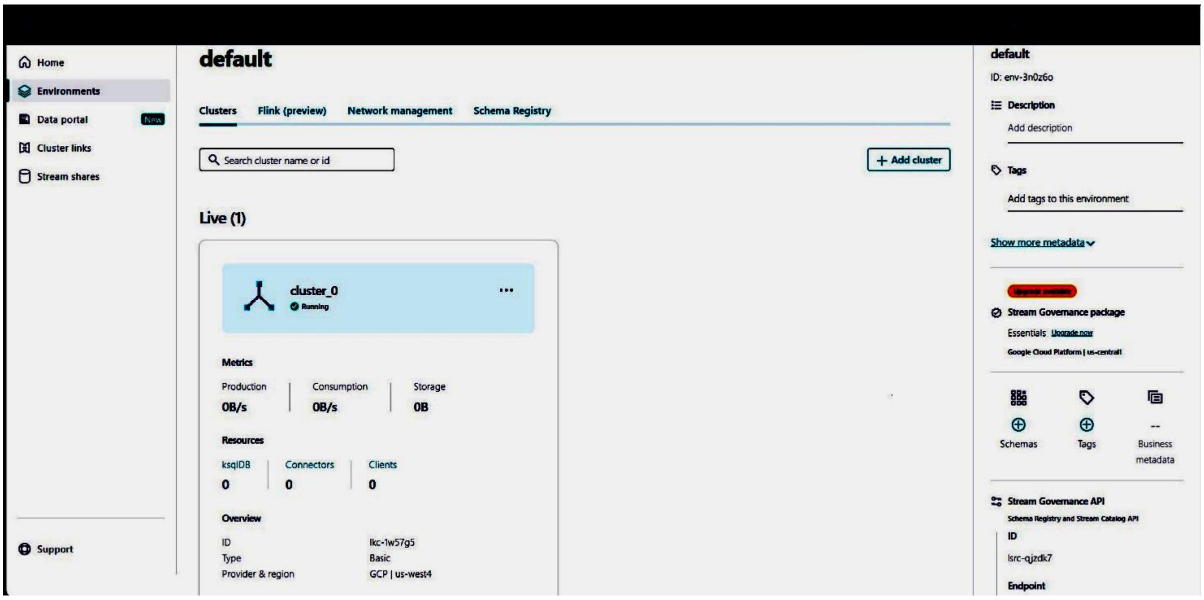

Fig. 5. Kafka cluster on the Confluent platform

```
public class NatureConsumer
{

    public static void consumeData(List<String> topics) throws IOException
    {
        final Properties props = loadConfig( configFile: "client.properties");
        props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        props.put(ConsumerConfig.GROUP_ID_CONFIG, "kafka-java-getting-started");
        props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "earliest");

        KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
        consumer.subscribe(topics);
        while (true)
        {
            ConsumerRecords<String, String> records = consumer.poll(Duration.ofMillis(100));
            for (ConsumerRecord<String, String> consumerRecord : records)
            {
                loadRecordToDb(consumerRecord);
            }
        }
    }
```

Fig. 6. Reading data from topics and loading data into the database

The designed database simplifies the administrator's work and provides the ability to quickly view the necessary information. In particular, the system assumes that the administrator will be able to easily add information about the indicators from the sensors to the database using our transmitter. The system is designed in such a way that the data from the sensors will fall into the topics, the system will use these topics to read the data and load it into the database. This system allows us, in the event of a database failure, to re-obtain all the data from our sensors by reading it from these topics. Since the platform allows us to store the data there for a long time. To read data from the topics, a consumer has been created that provides instant reading of data from the topics and constantly monitors the receipt of data in them (Fig. 6).

Also, when reading data by our system and writing it to the database, there will be a large set of logs.

During the work, it was determined that the data that is available to ordinary users on the sources from which we want to receive information will be entered into the database by the administrator. Thus, the system is adaptive and can be filled with data from verified sources for any regions.

The home page for users contains a header with the name of the program and a mini menu (Fig. 7).

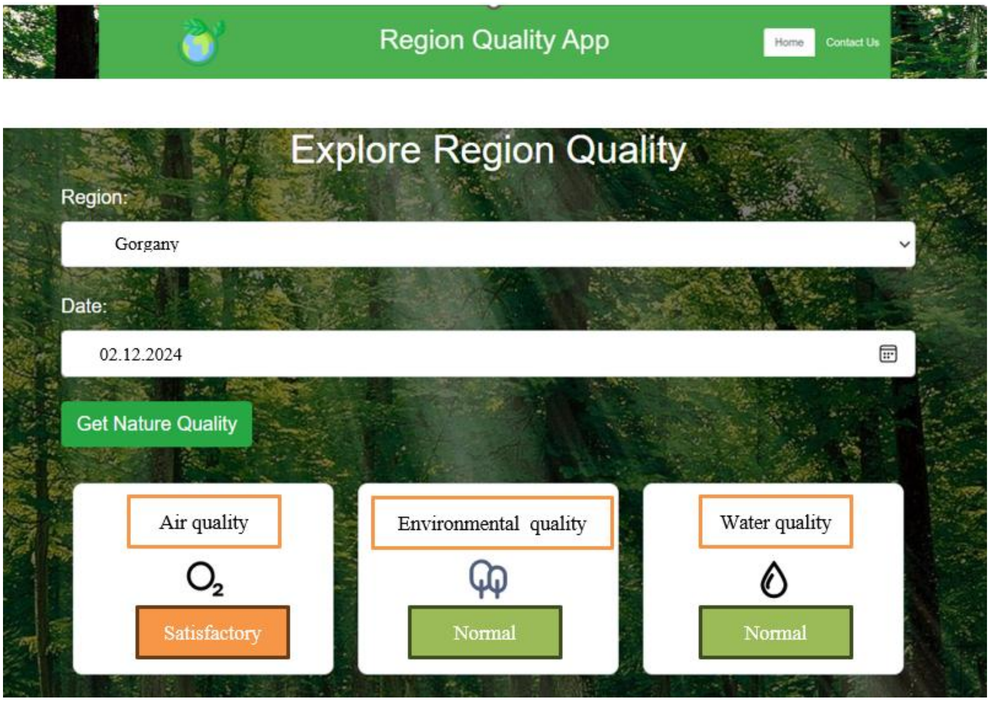

Fig. 7. The result of the monitoring system

```java
private static final int NUM_INPUTS = 5; // pm1, pm25, pm10, temperature, water
private static final int NUM_HIDDEN_NODES = 20;
private static final int NUM_OUTPUTS = 5; // Good, Moderate, Harmful, Very Harmful, Dangerous
private static final int LEARNING_RATE_SCHEDULE_EPOCHS = 1000;
private static final double INITIAL_LEARNING_RATE = 0.01;

private static MultiLayerConfiguration.Builder buildNeuralNetworkConfiguration()
{
    return new NeuralNetConfiguration.Builder()
        .seed(123)
        .updater(Adam.builder().learningRate(INITIAL_LEARNING_RATE).build())
        .weightInit(WeightInit.XAVIER)
        .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
        .gradientNormalizationThreshold(1.0)
        .list()
        .layer( 0, new DenseLayer.Builder()
            .nIn(NUM_INPUTS)
            .nOut(NUM_HIDDEN_NODES)
            .activation(Activation.RELU)
            .build())
        .layer( 1, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
            .nIn(NUM_HIDDEN_NODES)
            .nOut(NUM_OUTPUTS)
            .activation(Activation.SOFTMAX)
            .build())
        .backpropType(BackpropType.Standard);
}
```

Fig. 8. Neural network configuration

The system is designed in such a way that data from sensors will fall into topics, the system will use these topics to read data and load them into the database.

This system allows, in case of database failure, to re-obtain all data from our sensors by reading them from these topics. Since the platform allows us to store data there forever. To read data from topics, a consumer was created that provides instant data reading and constantly monitors the data flow in them. To assess the state of the nature reserve, a neural network was trained on 5 parameters such as water index, temperature, and three air indicators (Fig. 8).

In particular, this neural network was trained on prepared original data from open access.

5. Conclusions

Thus, a completely relevant and at the same time unique system for research and monitoring of nature reserves in Ukraine was developed. Based on analogue systems, the advantages and disadvantages of existing systems were found, which played a significant role in the development of the system. This system is divided into three parts: the user interface, the data transfer environment and the neural-server part of the system. The server part is responsible for data integrity and processing, in particular, here the administrator has the ability to add new sensors, update and delete data. The developed monitoring system is adaptive and with the help of this system, users will be provided with the opportunity to quickly and smoothly receive data, because the template optimized code for sending data from sensors will easily help to quickly transfer large volumes of data every second, and the formed dynamic user interface model will easily allow administrators to increase the capabilities of the system and adapt it for different territories and regions.

Conflict of Interest

The authors declare re no financial or other potential conflicts of interest regarding this work.

References

[1] Eco Rayon, 2024 [Online]. Available: https://eco.rayon.in.ua/news/578824-plan-zi-stvorennya-zapovidnikh-teritoriy-v-ukraini-vikonali-lishe-na-35

[2] Wow Nature, 2024 [Online]. Available: https://wownature.in.ua/en/

[3] Uhryn L. E. Mathematical modeling of the assessment of the conditions of the territories of the nature reserve fund of national importance// Information and innovative technologies of the 21st century: problems and prospects for development: materials of the VI International Scientific Conference (Katowice, September 19–20, 2023).

[4] Uhryn L. E. Monitoring the ecological safety of nature reserve regions, Innovative technologies in the development of modern society: abstracts of the VI International Scientific and Practical Conference, October 10–11, 2024.

[5] Backend Development: Skills, Salary, Roles & More, 2023 [Online]. Available: https://www.simplilearn.com/tutorials/programming-tutorial/what-is-backend-development

[6] Flanagan, David. "JavaScript: The Definitive Guide". O'Reilly Media, 2020 – 706pp.

[7] How Java Spring MVC Works: Spring MVC Request, 2023 [Online]. Available: https://tutorialspedia.com/how-java-spring-mvc-works-spring-mvc-request-flow-explained-step-by-step/

[8] Palmer, Boris. "Programming TypeScript: Making Your JavaScript Applications Scale". O'Reilly Media, 2019. 324 p.

[9] Beaulieu, Alan. "Learning SQL: Generate, Manipulate, and Retrieve Data". O'Reilly Media, 2020. 38 p.