



УДК 663.5:004.896

# ХМАРНА АРХІТЕКТУРА ПЕРЕДАЧІ ДАНИХ ДЛЯ ВПРОВАДЖЕННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ В СПИРТОВУ ПРОМИСЛОВІСТЬ

Омельченко О.С.<sup>1</sup>, Луцька Н.М.<sup>2</sup><sup>1,2</sup> Національний університет харчових технологій, м. Київ, УкраїнаORCID: <https://orcid.org/0000-0001-8593-0431>E-mail: alexmir\_98@ukr.net<sup>1</sup>, lutkanm2017@gmail.com<sup>2</sup>

Copyright © 2021 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>DOI: <https://doi.org/10.15673/atbp.v15i4.2589>

**Анотація.** Впровадження концепції Індустрії 4.0 у сфері спиртової промисловості передбачає інтеграцію передових цифрових технологій та підходів у виробничий процес. Однією з інновацій є використання моделей машинного навчання в технологічній системі для забезпечення точного контролю та оптимізації процесів виробництва та якості продукції. Однак для реалізації таких моделей необхідний доступ до обширного архіву технологічних даних, а також відповідні інтерфейси для взаємодії з ними, що породжує проблему розробки програмно-інформаційної структури, яка задовольняє ці потреби і відповідає стандартам Промислового Інтернету Речей (IIoT).

У даному дослідженні розглядаються питання ефективного збору, передачі та збереження технологічних даних виробничого підприємства. Представлений механізм доступу до накопичених даних для їх подальшої обробки моделями інтелектуального аналізу. Також сформована базова архітектура, побудована на хмарному підході, для практичної реалізації системи обробки виробничої інформації. Цей підхід базується на результативних вітчизняних та міжнародних дослідженнях у цій області.

Для втілення хмарної архітектури використано такі програмні засоби: симулятор промислового контролера Unity Pro, інструмент потокової розробки Node-RED для передачі даних, база даних часових рядів InfluxDB та сервіс машинного навчання Grafana Machine Learning. Подано детальний опис налаштування взаємодії між цими компонентами на прикладі збору даних з об'єкта спиртової промисловості та досягнення прогнозування майбутньої поведінки системи.

Запропонована архітектура базується на гнучких програмних ресурсах, які надають широкий спектр API для взаємодії з різноманітними сервісами та додатками. Це дає можливість використовувати запропоновану структуру як основу для проектування подібних систем.

**Abstract.** The implementation of Industry 4.0 in the ethanol industry involves the integration of advanced digital technologies and approaches into the manufacturing process. One of the innovations is the usage of machine learning models in the technological system to ensure precise control and optimization of production processes and product quality. However, the realization of such models requires access to an extensive archive of technological data, along with corresponding interfaces for interaction, giving rise to the challenge of developing a software-information structure that meets these requirements and aligns with the standards of the Industrial Internet of Things (IIoT).

This study addresses the issues of effective collection, transmission, and storage of technological data of a production enterprise. A mechanism for accessing accumulated data is presented for their further processing using intelligent analysis models. Additionally, a foundational architecture is formulated, built upon a cloud-based approach, for the practical implementation of a production information processing system. This approach is based on effective domestic and international research in this area.

To implement the cloud architecture, the following software tools were employed: the Unity Pro industrial controller simulator; the Node-RED streaming development tool for data transmission, the InfluxDB time series database, and the Grafana Machine Learning service for machine learning support. A detailed description of configuring interaction among these components is provided, using the example of data collection from a ethanol object and achieving predictive behavior modeling for the system.

The proposed architecture is based on flexible software resources that offer a wide range of APIs for interaction with various services and applications. This allows for the utilization of the suggested structure as a foundation for designing similar systems.



**Ключові слова:** Індустрія 4.0, промисловий інтернет речей, хмарна архітектура, спиртова промисловість, машинне навчання.

**Key words:** Industry 4.0, industrial internet of things, cloud-based architecture, ethanol industry, machine learning.

## Вступ

Розвиток Індустрії 4.0 привів до впровадження нових можливостей в області обробки даних, задіявши нові підходи та поняття, які раніше мало або взагалі не використовувалися при аналізі виробничих процесів, таких як Data Mining, Big Data та Machine Learning. Початок застосування моделей машинного навчання у промисловому виробництві підняв питання про необхідність розробки та впровадження нової архітектури управління даними, яка б задовольняла новим вимогам Industrial Internet of Things (IIoT), та вирішувала наступні проблеми [1, 2]:

- стандартизація протоколів та програмних інтерфейсів;
- можливість відсіювання, управління та обробки великих даних;
- безпека передачі даних;
- механізми селекції збережених даних;
- забезпечення механізму неперервної передачі даних;
- ефективний формат збереження з часовими мітками.

Реалізація наведених вимог дозволяє забезпечити ефективну обробку великих об'ємів даних, що являється одним з дев'яти кроків на шляху до впровадження Індустрії 4.0 та її переваг, таких як можливість залучення машинного навчання для технологічного процесу. При цьому основні проблеми впровадження пов'язані зі збором, передачею, перетворенням та шифруванням даних [3].

Спиртовий завод являє собою складний технологічний об'єкт з великою кількістю взаємозалежних, в тому числі невимірювальних та прихованих технологічних змінних. Одним з варіантів подальшого підвищення контролю та продуктивності цього виробництва є впровадження методів машинного аналізу, що призводить до актуалізації проблем наведених вище. При цьому методи вирішення задач обробки, передачі та зберігання виробничих даних залишаються відкритими для пошуку ефективних технологічних рішень в умовах постійного розвитку інтелектуальних технологій, сервісів та програм [1].

## Аналіз літературних даних і постановка проблеми

Передача та збереження технологічних даних займає одне з ключових значень на шляху до впровадження інтелектуальних методів обробки інформації на виробництві в рамках реалізації основ Індустрії 4.0. Це стало можливим завдяки інтеграції принципів IIoT в промислове виробництво – нової парадигми взаємодії виробничої інфраструктури [4, 5].

Наявні нині стандарти побудови архітектури IIoT не мають чітко вираженої структури та реалізації, що з однієї сторони розширює гнучкість та застосування інноваційних підходів при проектуванні систем, проте створює додаткові складнощі у вигляді пошуку рішень із забезпечення сумісності, синхронізації та взаємодії між складовими спроектованої системи [6, 7].

В роботі [8] пропонується приклад реалізації хмарної IIoT архітектури для покращення розгортання інтелектуальних промислових систем, що заснована на дистанційному моніторингу та управлінні. Використовуючи асинхронний двигун як об'єкт дослідження, авторам вдалося впровадити розрізнення штатної та аномальної поведінки за допомогою обробки в хмарі даних його роботи. Для передачі даних роботи двигуна в хмару IBM Bluemix, де надавали оцінку графіки отриманих даних, було використано Arduino з Node-RED. Ціллю дослідження була демонстрація можливості створення хмарної основи для прогнозування та управління автоматизованими системами в рамках Індустрії 4.0.

Більш детально з точки зору аналізу структури IIoT дану тематику було розглянуто в роботі [9]. Автори пропонують концептуальну модель архітектури програмного забезпечення для IIoT, відштовхуючись від основних складнощів при проектуванні та розробці рішень в IIoT, таких як сумісність, різноманітність та взаємодія між підсистемами. Відповідно до цих труднощів формуються основні вимоги до розробки – модульність, масштабованість та сумісність. Концептуальна модель розглядає розділення архітектури на чотири рівня із забезпеченням взаємодії між ними. На різних рівнях пропонується розглянути використання різноманітних стандартів зв'язку, таких як CAN, Narrowband IIoT та LTE Cat-M.

Ширший перегляд мережевих комунікацій та протоколів що використовуються в IIoT та їхніх особливостей розглядається в статті [10]. В [11] фокус дослідження зосереджений навколо питання забезпечення безпеки обміну інформацією.

В роботах [11 – 13] розглядається практична реалізація архітектури передачі даних для конкретної специфічної задачі. Для збору, передачі та збереження даних було використано подібні підходи, а саме концепція збору даних від об'єкта до вузла передачі даних, транспортування їх до хмарного сховища, яким виступає InfluxDB, та візуалізація інформації використовуючи можливості Grafana. Ціль вузла передачі даних полягає у відборі необхідної інформації та переведення її в JSON-формат для подальшого збереження у базі даних числових рядів InfluxDB. Завдяки сумісності інтерфейсів Grafana може зв'язуватися з InfluxDB для візуалізації збережених даних, та додаткових можливостей. Наприклад, в роботі [13] розраховано засобами Grafana виконувати сповіщення.



Центральною частиною інформаційної архітектури є сховище даних. Головними вимогами до промислової бази даних (БД) є надійність та можливість обробляти величезні об'єми інформації, які генеруються пристроями ІоТ. Так як дані ІоТ по своїй природі являють собою часові ряди, то для їх збереження доцільно використовувати БД з схожою структурою, вибір з-поміж яких залежить від конкретних вимог, особливостей інформації та наявних програмних засобів [14].

В роботі [15] проведено детальне порівняльне дослідження БД, в тому числі БД часових рядів з InfluxDB, за результатами якого InfluxDB показує значні конкурентоспроможні можливості по низкам параметрів, таких як пропускна здатність, продуктивність та обсяг дискового простору.

Автори [16] на додаток до вже запропонованої архітектури з InfluxDB та Grafana, пропонують використання мови візуального програмування Node-RED для створення програмного вузла передачі даних до InfluxDB. Завдяки підтримці безлічі API-інтерфейсів, мультиплатформеності, гнучкості та фокусуванні на підтримці взаємодії з онлайн-сервісами як частиною ІоТ він забезпечує зручну взаємодію з хмарними БД. Разом з тим Node-RED чудово справляється із завданням комунікації з керуючими пристроями, такими як промисловий контролер [17], що робить його зручним та ефективним інструментом на роль вузла передачі даних.

Підсумовуючи аналіз даних літературних джерел можна зробити висновки про поширеність та актуальність завдань побудови хмарної архітектури обробки даних в різних галузях, в тому числі переробної промисловості. Наявність прикладів впровадження робочої структури передачі даних та дослідження ефективностей варіативності підходів формують підґрунтя для реалізації дієвої архітектури обробки даних в спиртовому виробництві.

**Мета і завдання дослідження:** демонстрація впровадження хмарної архітектури попередньої обробки, передачі та збереження виробничих даних відповідно до вимог ІоТ на об'єктах спиртової промисловості. Дана підсистема спроектована у складі автоматизованої системи управління виробництвом та направлена на подальше використання інтелектуального аналізу, такого як опрацювання отриманих даних методами машинного навчання. Останнє призводить до підвищення ресурсоефективності виробництва у цілому.

#### Методи і матеріали досліджень

З аналізу публікацій встановлено ключові особливості побудови архітектури передачі та збереження даних, переваги та недоліки підходів та програмних засобів реалізації подібних завдань. Беручи до уваги результати розглянутих досліджень була спроектована та розроблена архітектура передачі даних для об'єктів спиртового виробництва.

Так як кінцевою ціллю дослідження є підвищення продуктивності та якості спиртового виробництва шляхом забезпечення впровадження інтелектуальних методів аналізу вже до наявної системи автоматизації, відправною точкою збору даних та структурних модифікації є промисловий контролер, через який вже проходять всі необхідні дані для обробки. Наступним кроком є налагодження зв'язку між контролером та вузлом передачі даних, в якості якого виступає Node-RED. Дані з Node-RED надходять до БД часових рядів InfluxDB, де вони можуть зберігатися та бути використаними для подальшого аналізу, наприклад машинного навчання. В статті наведений приклад використання цих даних засобами Grafana Machine Learning.

Запропонована архітектура передачі даних (Рис. 1) складається з трьох структурних складових зв'язків, між якими відбувається обмін даними:

- Unity Pro – Node-RED;
- Node-RED – InfluxDB;
- InfluxDB – Grafana.

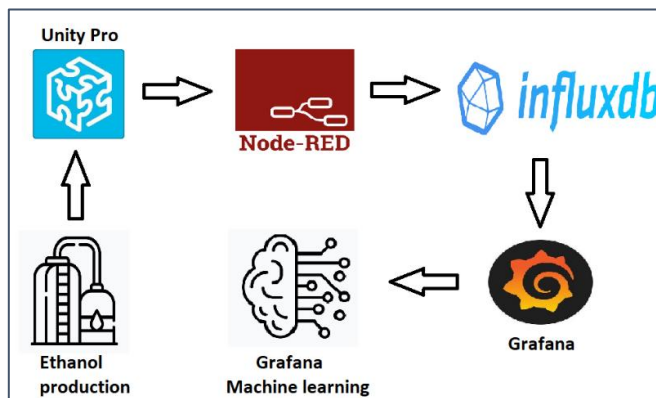


Рис. 1. Схема програмної архітектури передачі даних  
Fig. 1. Scheme of software architecture of data transfer

Для побудови архітектури було задіяно наступні програмні засоби:

**Unity Pro.** Unity Pro – це середовище для конфігурування, програмування, налагодження та діагностики промислових контролерів Modicon Quantum, Premium, Atrium та M340. Однією з використаних в цій роботі можливостей Unity Pro є наявність програмного імітатора контролера та можливості доступу до нього з інших програмно-технічних засобів по протоколу Modbus/TCP.



**Node-RED.** Node-RED – це інструмент для візуального програмування потоків даних, розроблений працівниками компанії IBM для поєднання різноманітних пристроїв, API та онлайн-сервісів як складових частин Інтернету речей. Легке середовище виконання (runtime) побудоване на Node.js, в повній мірі використовуючи переваги його подіє-орієнтованої неблокуючої моделі. Це робить його ідеальним для роботи на краю (Edge) мережі на недорогих апаратних засобах, таких як Raspberry Pi, а також у хмарі. В даній роботі можливості Node-RED використовуються для побудови вузла передачі даних між Unity Pro та InfluxDB. Вибір цього програмного засобу в тому числі зумовлений наявністю величезної кількості різноманітних модулів та використанням потоків, що базуються на JSON, що дозволяє легко імпортувати та експортувати їх для спільного використання.

**InfluxDB.** InfluxDB – база даних часових рядів з відкритим кодом. Написана на мові Go і оптимізована для швидкого запису та отримання даних часових рядів в галузях моніторингу операцій, метрик програмного забезпечення, даних сенсорів інтернету речей та аналітики в реальному часі. База даних часових рядів оптимізована для швидкого прийому даних. Такі системи використовують індексацію даних, об'єднаних з часом. Як наслідок, швидкість завантаження не зменшується з часом і залишається досить стабільною. InfluxDB не має зовнішніх залежностей і надає SQL-подібну мову запитів з вбудованими функціями для роботи з часом і структурою даних, що складається з вимірів, серій, та точок даних. Кожна точка складається з кількох пар ключ-значення які називаються множиною полів та мітку часу. Набір точок об'єднаних однаковою множиною пар ключ-значення які називаються множиною тегів, задають ряд. Нарешті, ряди групуються разом спільним ідентифікатором і формують вимірювання.

**Grafana.** Grafana – це мультиплатформний веб-додаток для аналітики та інтерактивної візуалізації з відкритим кодом. Grafana підтримує безліч платформ, в тому числі часто використовується як засіб візуалізації даних з БД часових рядів. Використання Grafana в рамках даної роботи зумовлене наявністю в її арсеналі плагіну машинного навчання – Grafana Machine Learning. Grafana Machine Learning дає користувачам Grafana Cloud можливість створювати прогнози щодо поточного чи майбутнього стану своїх систем. Це дозволить практично перевірити та продемонструвати працездатність та втілити кінцеву ціль спроектованої архітектури.

#### Результати досліджень.

##### 1) Зв'язок Unity Pro – Node-RED – InfluxDB.

Для демонстрації роботи передачі даних, вхідні дані було згенеровано в середовищі імітації Unity Pro. Об'єктом симуляції є модель спиртового зброджувача (Рис. 2, а).

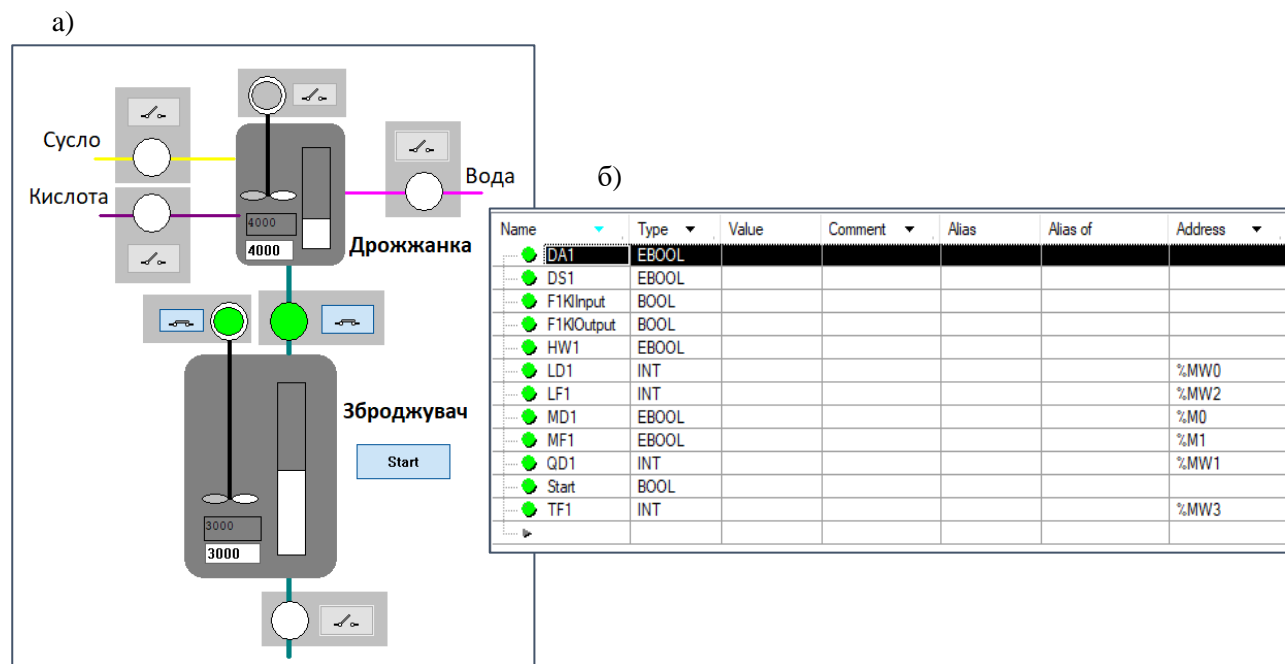


Рис. 2. Екран оператора Unity Pro (а) та таблиця змінних PLC (б)

Fig. 2. Unity Pro operator screen (а) and PLC variable table (б)

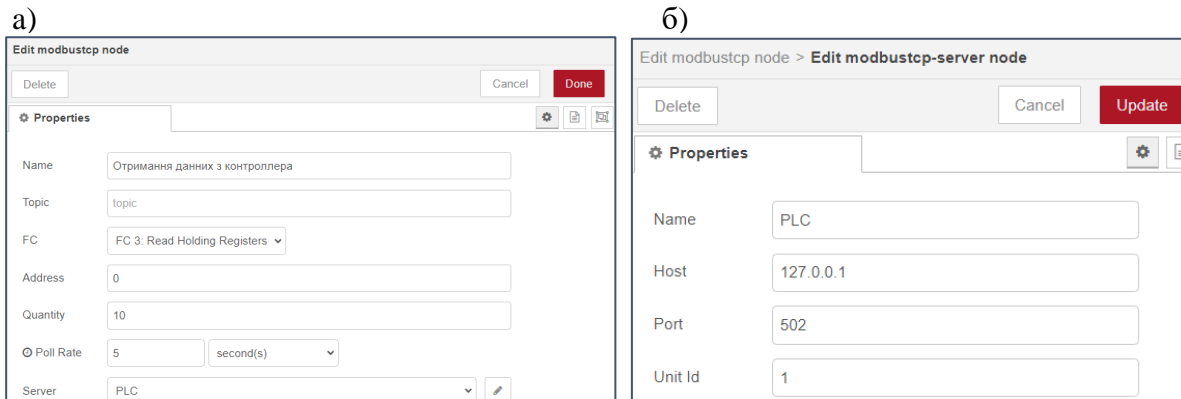
Змінні, необхідні для передачі, мають мати визначений адрес для їх отримання в Node-RED (Рис. 2, б). Це дає змогу налаштувати зв'язок між значеннями PLC та їх зчитуванням з Holding Registers в Node-RED (Рис. 3, а) через підключення через протокол Modbus-TCP (Рис. 3, б).

Програма в Node-RED (Рис. 4) являє собою структуру взаємопов'язаних вузлів, що маніпулюють даними під час їх проходження в потоці. Структурно програму можна поділити на такі функціональні блоки:

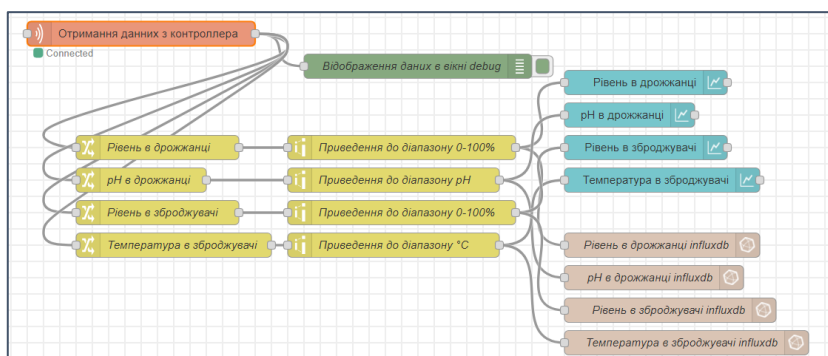
- Отримання даних з PLC (Рис. 3);
- Виділення конкретної змінної з масиву (Рис. 5, а);
- Приведення змінної до її діапазону (Рис. 5, б);
- Локальна візуалізація отриманих даних (Рис. 6);



- Передача відмасштабованих даних до InfluxDB (Рис. 8).



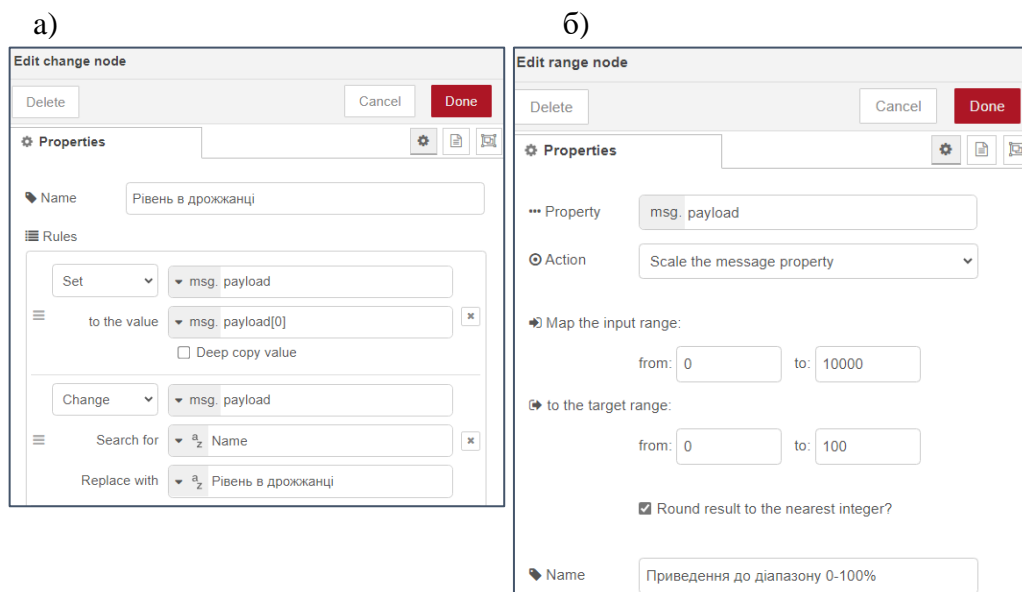
**Рис. 3. Конфігурація node для отримання даних (а) та з'єднання (б) з PLC**  
**Fig. 3. Node configuration for receiving data (a) and connecting (b) to the PLC**



**Рис. 4. Програма в Node-RED**  
**Fig. 4. Program in Node-RED**

Отримані від контролера дані являють собою масив, в який входять всі змінні, тому перш ніж працювати з цими даними їх необхідно розділити. Для цього в Node-RED використовується change node, де з масиву відбирається потрібна змінна (Рис. 5, а).

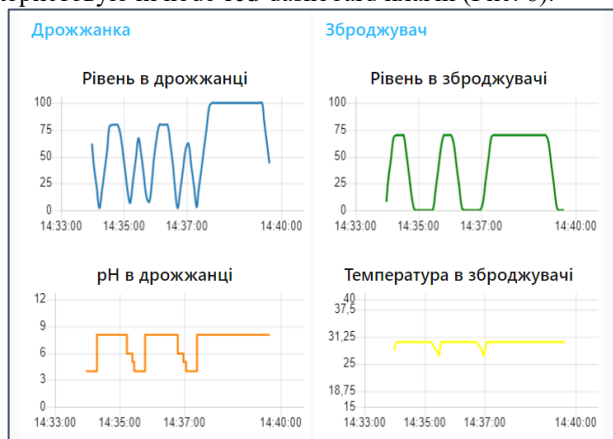
Отримані дані після обробки в change node дані є розділеними змінними, які обрані для передачі та аналізу, проте вони все ще знаходяться в діапазоні контролера, який за замовчуванням складає 0-10000. Тому для подальшого використання цих даних необхідне їх приведення до відповідного їм природнього діапазону. Для цього було використано range node, який дозволяє масштабувати числові значення під вказані діапазони (Рис. 5, б).



**Рис. 5. Приклад конфігурації change node (а) та range node (б)**  
**Fig. 5. Example of configuration of change node (a) and range node (b)**



На даному етапі закінчується обробка даних і вони є готовими для передачі. Попередньо їх можна візуалізувати в Node-RED використовуючи node-red-dashboard плагін (Рис. 6).



**Рис. 6. Візуалізація даних в Node-RED**

**Fig. 6. Data visualization in Node-RED**

Наступним кроком є встановлення зв'язку між Node-RED та InfluxDB, який конфігурується з обох сторін. З боку InfluxDB на сторінці організації в даних профілю необхідно дізнатися Cluster URL та Organization ID (Рис. 7), які використовуються для конфігурації підключення. Не менш важливим кроком є створення API Token для надання дозволів на запис чи читання з Buckets InfluxDB (Рис. 7) та виділення Bucket для зберігання даних.

**Рис. 7. Дані для підключення зі сторони InfluxDB**

**Fig. 7. Connection data from the InfluxDB side**

Отримані дані з InfluxDB використовуються для конфігурації підключення з боку Node-RED: Cluster URL та API Token необхідні для налаштування підключення до серверу (Рис. 8, а), Organization ID та дані Bucket – для налаштування зв'язку з спеціально виділеним Bucket (Рис. 8, б), при чому для кожної змінної можна робити вибір окремо.

**Рис. 8. Налаштування підключення до серверу InfluxDB (а)**

**та загальні налаштування взаємодії з InfluxDB (б)**

**Fig. 8. Setting up the connection to the InfluxDB server (a) and general settings for interaction with InfluxDB (b)**

## 2) Інтерфейс InfluxDB



Після успішної конфігурації, дані з PLC почнуть надходити у вказані в налаштування контейнери – Buckets. В свою чергу Buckets можуть зберігати значення багатьох змінних одночасно.

InfluxDB дозволяє переглядати збережені дані як в Data Explorer, у вигляді графу чи табличному вигляді (Рис. 9), так і підтримує створення користувацьких панелей візуалізації – Dashboard, де можна налаштувати перегляд необхідних змінних (Рис. 10).

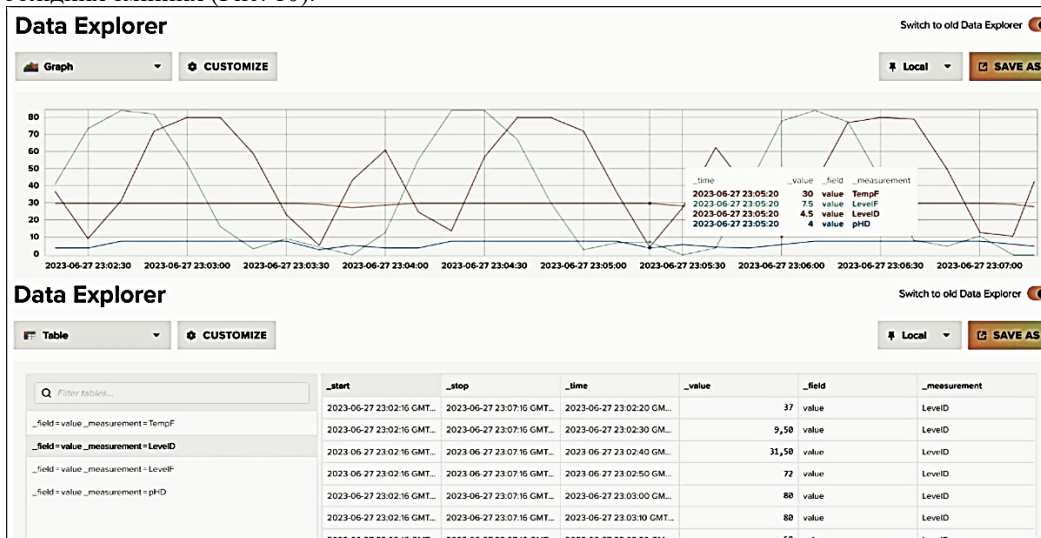


Рис. 9. Відображення даних за допомогою Data Explorer InfluxDB  
Fig. 9. Displaying data using the InfluxDB Data Explorer

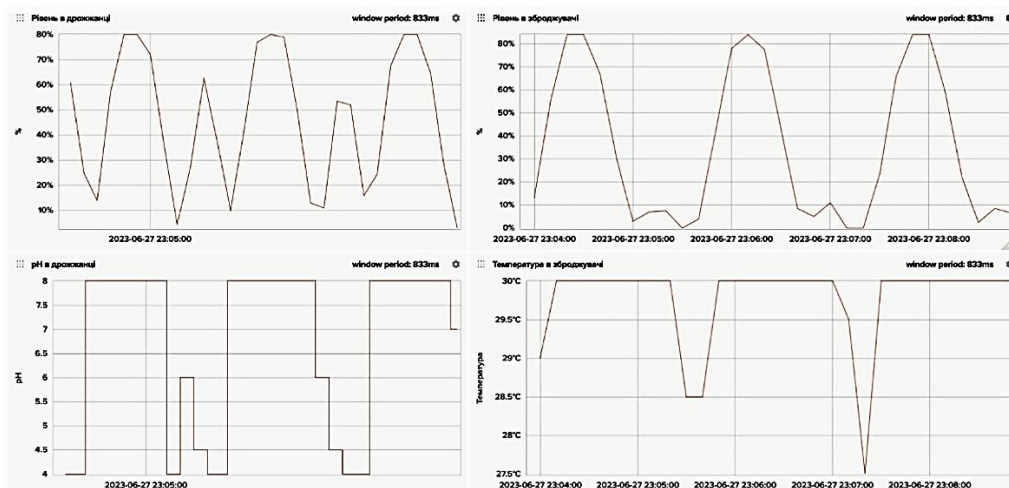


Рис. 10. Відображення даних за допомогою Dashboard InfluxDB  
Fig. 10. Displaying data using the InfluxDB Dashboard

### Обговорення результатів

В цьому розділі наведена демонстрація взаємодії з сервісом машинного навчання Grafana Machine Learning.

Однією з переваг бази даних InfluxDB є зручна API взаємодія з безліччю інших сервісів. Одним з таких сервісів, що задовольняє цілі дослідження та дозволяє продемонструвати реалізацію машинного навчання, базованому на запропонованій архітектурі є Grafana Machine Learning.

Grafana Machine Learning дозволяє прогнозувати поведінку системи, навчаючись на даних про попередню роботу. Для налагодження роботи з цим сервісом необхідно встановити з'єднання між InfluxDB та Grafana, використовуючи дані для підключення зі сторони InfluxDB (Рис. 7), та Grafana Data source конфігурацію (Рис. 11).

За умови успішного налаштування Grafana отримає доступ до збережених у InfluxDB даних (Рис. 12), дозволивши подальшу обробку цієї інформації.

Використання функції прогнозування в Grafana Machine Learning потребує створення та налаштування відповідної прогнозованої метрики – Metric Forecast. Для цього формується запит для відбору необхідної для навчання змінної із відповідного Bucket (Рис. 13).

Створений Metric Forecast проходить навчання, використовуючи алгоритм Prophet, та по завершенню надає прогноз поведінки системи в майбутньому в залежності від актуальних даних (Рис. 14).

Інтерфейси Grafana Machine Learning надають можливість детальніше вивчити прогнозовані масиви даних та створити користувацькі запити для них (Рис. 15).



The image shows the configuration interface for an InfluxDB data source in Grafana. The 'Name' field is set to 'InfluxDBData'. The 'Query Language' is set to 'Flux'. Under the 'HTTP' section, the 'URL' is 'https://westeurope-1.azure.cloud2.influxdat...'. The 'Allowed cookies' field contains 'New tag (enter key to add)' and the 'Timeout' is 'Timeout in seconds'. The 'InfluxDB Details' section includes: 'Organization' (redacted), 'Token' (configured), 'Default Bucket' (fermentation), 'Min time interval' (10s), and 'Max series' (1000).

Рис. 11. Налаштування Grafana Data source для зв'язку з InfluxDB  
Fig. 11. Configuring the Grafana Data source to communicate with InfluxDB

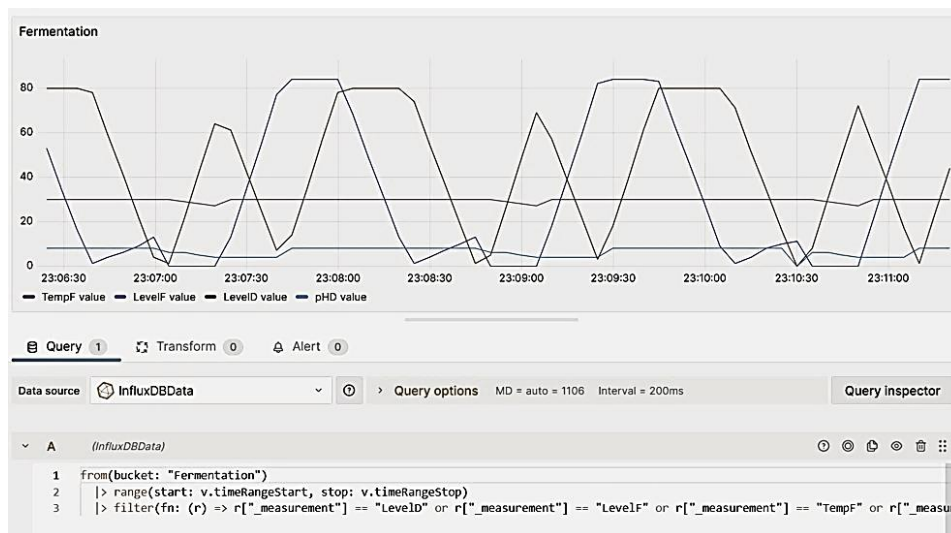


Рис. 12. Відображення збережених даних за допомогою Grafana  
Fig. 12. Displaying saved data using Grafana



Рис. 13. Створення Metric Forecast в Grafana Machine Learning  
Fig. 13. Creating a Metric Forecast in Grafana Machine Learning

Опираючись на результати передбачень Grafana Machine Learning можна відслідковувати зміни довгострокових трендів та отримувати попередження про майбутні зміни технологічних показників процесу спиртового виробництва. Таким чином, зменшується вірогідність виникнення передаварійних та аварійних ситуацій, які пов'язані з технологічною складовою виробництва, своєчасно змінюється технологічний режим та



збільшується ритмічність технологічних процесів. Все це призводить до зменшення матеріальних та енергетичних втрат, зменшення браку та збільшення продуктивності.

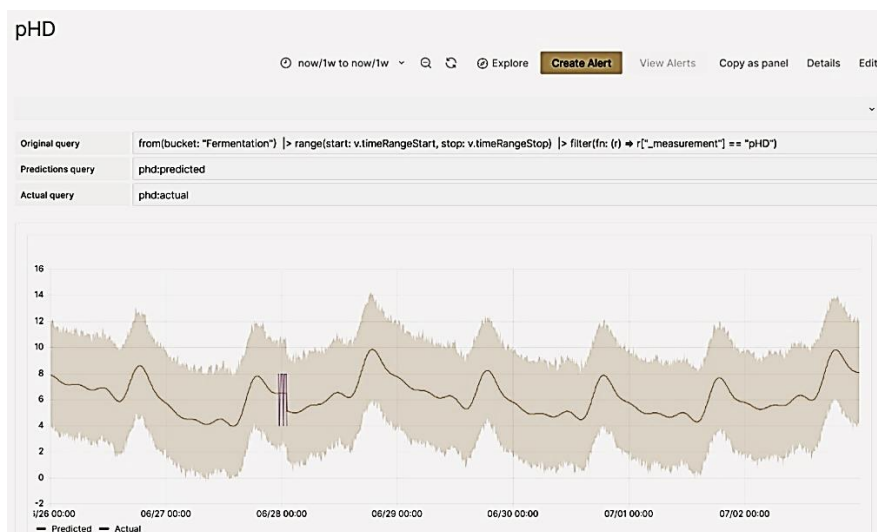


Рис. 14. Прогноз зміни поведінки змінної в Grafana Machine Learning

Fig. 14. Prediction of changes in the behavior of a variable in Grafana Machine Learning

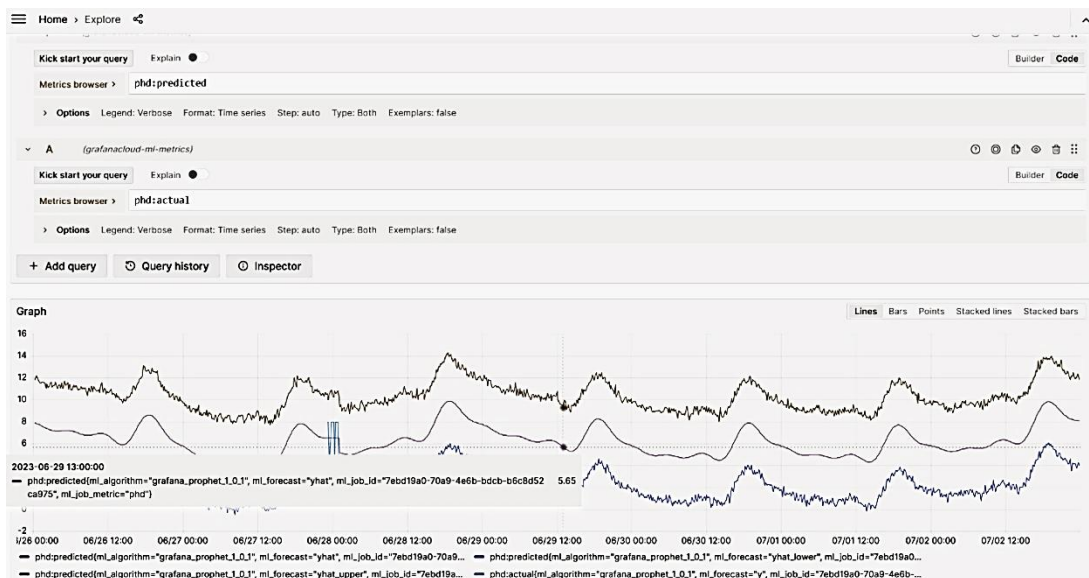


Рис. 15. Вікно дослідження прогнозу в Grafana Machine Learning

Fig. 15. Forecast study window in Grafana Machine Learning

## Висновки

Збереження даних промислового виробництва на рівні БД, розташованої у хмарі чи хмарних додатках, дозволяє вивести систему на принципово новий рівень інформаційної гнучкості. Дані, збережені таким чином, стають доступними для обробки множиною інтелектуальних сервісів та програмних продуктів, які у свою чергу реалізують нові функції, пов'язані з підвищенням якості, продуктивності та економічних складових виробництва шляхом різностороннього аналізу великих масивів виробничої інформації.

Об'єкти спиртової промисловості містять у своєму складі безліч прихованих та невимірюваних параметрів, починаючи від процесів бродіння, закінчуючи процесами ректифікації. Традиційний аналіз промислових даних також ускладнений значною інерційністю технологічних процесів. Це сприяє пошуку нових методів дослідження перебігу технологічних процесів, одним з яких є збір та концентрація виробничих даних для їх обробки методами машинного аналізу.

У цій статті розглянуто приклад побудови повноцінної архітектури передачі даних, яка забезпечує можливість впровадження методів машинного навчання для об'єктів спиртової промисловості. Для розробки запропонованої структури були проаналізовані актуальні дослідження в цій сфері та враховано досвід сучасних підходів.

Розроблена архітектура позиціонується як базова структура, відкрита для подальших модифікацій та змін. Використані програмні засоби Node-RED, InfluxDB та Grafana мають високу інтеграційну гнучкість, та, за потреби, можуть легко налаштуватися на взаємодію з іншими сервісами, залежно від наявного програмного забезпечення чи цілей побудови.

**Список використаних джерел**

1. M. Tavana, V. Hajipour, S. Oveisi, "IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions," *Internet of Things*, vol. 11, pp. 100262, 2020. <https://doi.org/10.1016/j.iot.2020.100262>
2. I. Ungurean, N.C. Gaitan, "A software architecture for the Industrial Internet of Things – A conceptual model," *Sensors*, vol. 20, no. 19, pp. 5603, 2020. <https://doi.org/10.3390/s20195603>
3. C. E. Cotet, et al., "An innovative industry 4.0 cloud data transfer method for an automated waste collection system," *Sustainability*, vol. 12, no. 5, pp. 1839, 2020. <https://doi.org/10.3390/su12051839>
4. E. Manavalan, K. Jayakrishna, "A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Computers & Industrial Engineering*, vol. 127, pp. 925-953, 2019. <https://doi.org/10.1016/j.cie.2018.11.030>
5. M. T. Okano, "IOT and industry 4.0: the industrial new revolution," in *International Conference on Management and Information Systems*, vol. 25, 2017.
6. S. Krčo, B. Pokrić, F. Carrez, "Designing IoT architecture (s): A European perspective," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 79-84, 2014.
7. S. A. Al-Qaseemi, et al., "IoT architecture challenges and issues: Lack of standardization," in *2016 Future Technologies Conference (FTC)*, IEEE, 2016. <https://doi.org/10.1109/FTC.2016.7821686>
8. A. F. da Silva, et al., "A cloud-based architecture for the internet of things targeting industrial devices remote monitoring and control," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 108-113, 2016. <https://doi.org/10.1016/j.ifacol.2016.11.137>
9. I. Ungurean, N. C. Gaitan, "A software architecture for the Industrial Internet of Things—A conceptual model," *Sensors*, vol. 20, no. 19, pp. 5603, 2020. <https://doi.org/10.3390/s20195603>
10. T. M. Tukade, R. Banakar, "Data transfer protocols in IoT—An overview," *Int. J. Pure Appl. Math*, vol. 118, no. 16, pp. 121-138, 2018.
11. M. D. Mudaliar, N. Sivakumar, "IoT based real time energy monitoring system using Raspberry Pi," *Internet of Things*, vol. 12, pp. 100292, 2020. <https://doi.org/10.1016/j.iot.2020.100292>
12. K. Chooung, K. Meekul, "Design of an IoT energy monitoring system," in *2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE)*, IEEE, 2018. <https://doi.org/10.1109/ICTKE.2018.8612412>
13. S. Gangadhar, "The real time environmental time series data analysis using Influx DB," *International Journal of Advanced Scientific Innovation*, vol. 1, no. 1, 2020. <https://doi.org/10.5281/zenodo.4641703>
14. M. Nasar, M. A. Kausar, "Suitability of influxdb database for iot applications," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 1850-1857, 2019.
15. S. N. Zehra Naqvi, S. Yfantidou, E. Zimányi, "Time series databases and influxdb," *Studienarbeit, Université Libre de Bruxelles*, vol. 12, 2017.
16. M. Cicioğlu, A. Calhan, "Internet of Things-based firefighters for disaster case management," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 612-619, 2020. <https://doi.org/10.1109/JSEN.2020.3013333>
17. A. Gavlas, J. Zwierzyna, J. Koziorek, "Possibilities of transfer process data from PLC to Cloud platforms based on IoT," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 156-161, 2018. <https://doi.org/10.1016/j.ifacol.2018.07.146>

**References**

1. M. Tavana, V. Hajipour, S. Oveisi, "IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions," *Internet of Things*, vol. 11, pp. 100262, 2020. <https://doi.org/10.1016/j.iot.2020.100262>
2. I. Ungurean, N.C. Gaitan, "A software architecture for the Industrial Internet of Things – A conceptual model," *Sensors*, vol. 20, no. 19, pp. 5603, 2020. <https://doi.org/10.3390/s20195603>
3. C. E. Cotet, et al., "An innovative industry 4.0 cloud data transfer method for an automated waste collection system," *Sustainability*, vol. 12, no. 5, pp. 1839, 2020. <https://doi.org/10.3390/su12051839>
4. E. Manavalan, K. Jayakrishna, "A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Computers & Industrial Engineering*, vol. 127, pp. 925-953, 2019. <https://doi.org/10.1016/j.cie.2018.11.030>
5. M. T. Okano, "IOT and industry 4.0: the industrial new revolution," in *International Conference on Management and Information Systems*, vol. 25, 2017.
6. S. Krčo, B. Pokrić, F. Carrez, "Designing IoT architecture (s): A European perspective," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 79-84, 2014.
7. S. A. Al-Qaseemi, et al., "IoT architecture challenges and issues: Lack of standardization," in *2016 Future Technologies Conference (FTC)*, IEEE, 2016. <https://doi.org/10.1109/FTC.2016.7821686>
8. A. F. da Silva, et al., "A cloud-based architecture for the internet of things targeting industrial devices remote monitoring and control," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 108-113, 2016. <https://doi.org/10.1016/j.ifacol.2016.11.137>
9. I. Ungurean, N. C. Gaitan, "A software architecture for the Industrial Internet of Things—A conceptual model," *Sensors*, vol. 20, no. 19, pp. 5603, 2020. <https://doi.org/10.3390/s20195603>



10. T. M. Tukade, R. Banakar, "Data transfer protocols in IoT—An overview," *Int. J. Pure Appl. Math*, vol. 118, no. 16, pp. 121-138, 2018.
11. M. D. Mudaliar, N. Sivakumar, "IoT based real time energy monitoring system using Raspberry Pi," *Internet of Things*, vol. 12, pp. 100292, 2020. <https://doi.org/10.1016/j.iot.2020.100292>
12. K. Chooruang, K. Meekul, "Design of an IoT energy monitoring system," in *2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE)*, IEEE, 2018. <https://doi.org/10.1109/ICTKE.2018.8612412>
13. S. Gangadhar, "The real time environmental time series data analysis using Influx DB," *International Journal of Advanced Scientific Innovation*, vol. 1, no. 1, 2020. <https://doi.org/10.5281/zenodo.4641703>
14. M. Nasar, M. A. Kausar, "Suitability of influxdb database for iot applications," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 1850-1857, 2019.
15. S. N. Zehra Naqvi, S. Yfantidou, E. Zimányi, "Time series databases and influxdb," *Studienarbeit, Université Libre de Bruxelles*, vol. 12, 2017.
16. M. Cicioğlu, A. Calhan, "Internet of Things-based firefighters for disaster case management," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 612-619, 2020. <https://doi.org/10.1109/JSEN.2020.3013333>
17. A. Gavlas, J. Zwierzyna, J. Koziorek, "Possibilities of transfer process data from PLC to Cloud platforms based on IoT," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 156-161, 2018. <https://doi.org/10.1016/j.ifacol.2018.07.146>

Отримана в редакції 01.11.2023. Прийнята до друку 04.12.2023. Received 01 November 2023. Approved 12 December 2023. Available in Internet 03 January 2024

УДК 004.021

## АПАРАТНО-ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ МОНІТОРИНГУ ТЕРОРИСТИЧНОЇ АКТИВНОСТІ

Старченко В'ячеслав Володимирович

Чорноморський національний університет імені Петра Могили, м. Миколаїв, Україна

ORCID: <https://orcid.org/0000-0001-8593-0431>

E-mail: [viacheslav.starchenko@chmnu.edu.ua](mailto:viacheslav.starchenko@chmnu.edu.ua)

Copyright © 2021 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v15i4.2718>

**Анотація.** У сучасному світі кількість смертей та інцидентів у результаті тероризму вже перевищило число жертв будь-якої іншої форми та продовжує збільшуватися. У зв'язку з можливою легалізацією в Україні стрілецької зброї виникає нагальна необхідність у автоматизованих системах моніторингу терористичної активності. Головна задача таких систем – за допомогою акустичних датчиків оперативно виявляти факти пострілів та вибухів, визначати їх координати та повідомляти служби забезпечення безпеки населення. Розроблений апаратно-програмний комплекс має зіркову архітектуру, складається з декількох сенсорних кластерів та головного серверу. Кожен сенсорний кластер включає до себе три звукових сенсори розташованих на місцевості, контролер керування на базі мікропроцесору Arduino та кластерний сервер. До складу кожного сенсору входить мікрофон із низьким рівнем шуму, підсилювач та лінійний перетворювач цифрового сигналу. Головною задачею кластерного сервера є класифікація та локалізація у просторі джерела гучного імпульсного звуку. У випадку виявлення факту пострілу або вибуху, інформація про нього оперативно передається до відповідних служб забезпечення безпеки населення. Локалізація у просторі джерела гучного імпульсного звуку виконується методом геометричної триангуляції за результатами аналізу даних від акустичних сенсорів. У результаті попередніх випробувань прототипу системи була досягнута точність локалізації місця пострілу біля 10 м при рознесенні сенсорів на відкритій місцевості більш ніж на 800 м.

**Abstract.** Recently, the number of deaths and incidents resulting from terrorism has already exceeded the number of victims of any other form and continues to increase. In connection with the possible legalization of small arms in Ukraine, there is an urgent need for automated systems for monitoring terrorist activity. The main task of such systems is to quickly detect the facts of shots and explosions with the help of acoustic sensors, determine their coordinates and notify public security services. The developed hardware and software complex has a star architecture, consisting of several sensor clusters and the main server. Each sensor cluster includes three sound sensors located on the ground, a control controller