

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

(повна назва)

Кафедра Комп'ютерних інтелектуальних технологій та систем

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти

перший (бакалаврський)

Інтелектуальна система управління мікрокліматом з використанням методів
машинного навчання

Виконав:

здобувач IV року навчання,

групи КІУКІ-21-10

Максим ЗАЄЦЬ

(власне ім'я, прізвище)

Спеціальність 123 Комп'ютерна інженерія

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник ас. каф. КІТС Кирило ОЛІЙНИК

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри

Олег РУДЕНКО

2025 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	перший (бакалаврський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ **Заєць Максима Олександровичу**
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна система управління мікрокліматом з використанням методів машинного навчання

затверджена наказом по університету від “ 21 ” травня 2025 р. № 399СТ

2. Термін подання здобувачем роботи до екзаменаційної комісії 14.06.2025

3. Вхідні дані до роботи поточні сенсорні значення температури, вологості, CO₂, освітленості та присутності людей

прогноз погоди з веб-API

історичні логи мікроклімату з БД

профіль користувача з діапазонами комфорту та вагами критеріїв

перелік можливих дій обладнання і таблицю їх енергетичних витрат

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз сучасних систем керування мікрокліматом і їхніх обмежень.

Проектування структурно-функціональної моделі інтелектуальної системи.

Формалізація багатокритеріальної функції корисності для детермінованого модуля.

Реалізація й навчання Q-learning-агента; дискретизація станів та функція винагороди.

Експериментальна оцінка системи: точність прогнозу, енергоспоживання, час реакції,

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
10 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд і аналіз сучасного стану розглянутої проблеми, а також існуючих методів і засобів вирішення задач кваліфікаційної роботи	26.05.2025 – 31.05.2025	Виконано
2	Проектування архітектури додатку	01.06.2025 – 03.06.2025	Виконано
3	Створення додатку	04.06.2025 – 07.06.2025	Виконано
4	Аналіз та налагодження роботи додатку	08.06.2025 – 09.06.2025	Виконано
5	Оформлення матеріалів кваліфікаційної роботи	10.06.2025 – 14.06.2025	Виконано

Дата видачі завдання 26.05.2025

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

ас. Олійник К.О. _____
(посада, ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 60 с., 6 рис., 5 табл., 1 дод., 11 джерел.

МІКРОКЛІМАТ, ЕНЕРГОЕФЕКТИВНІСТЬ, LSTM, Q-LEARNING, ІОТ, HVAC, АДАПТИВНЕ КЕРУВАННЯ

У роботі розроблено та випробувано прототип інтелектуальної системи управління мікрокліматом, що поєднує класичний багатокритеріальний алгоритм і агент підкріплювального навчання. Мета дослідження — мінімізувати енергоспоживання й одночасно підтримувати параметри середовища в комфортних межах. Об'єктом є процес керування мікрокліматом у приміщеннях, предметом — алгоритми прогнозування та прийняття рішень на основі сенсорних даних. У межах роботи виконано аналіз предметної області, побудовано структурно-функціональну модель із чотирьох модулів (збір, обробка, прогнозування, керування), реалізовано LSTM-мережу для предиктивного контролю та Q-learning-агента для адаптивного вибору дій. Розроблено механізм арбітражу між детермінованим і RL-підходами та систему логування результатів у базу SQLite. Експериментальні випробування на тестовому стенді IoT-сенсорів засвідчили економію енергії 15–20 % і підтримання комфортних умов понад 95 % часу роботи. Практична цінність полягає у відкритій архітектурі, сумісній з MQTT / REST-інтерфейсами, що дає змогу інтегрувати систему у наявні HVAC-рішення.

ABSTRACT

Explanatory note of qualification work 60 pages, 6 figures, 5 tables, 1 appendices, 11 sources.

MICRO-CLIMATE, ENERGY EFFICIENCY, LSTM, Q-LEARNING, IOT, HVAC, ADAPTIVE CONTROL

The thesis presents a prototype of an intelligent indoor climate-control system that combines a classical multi-criteria algorithm with a reinforcement-learning agent. The goal is to maintain occupant comfort while reducing energy consumption. The object of study is the control process of indoor microclimate; the subject is forecasting and decision-making algorithms based on sensor data. A four-layer architecture (data acquisition, preprocessing, forecasting, control) was designed; an LSTM network predicts environmental parameters, whereas a Q-learning agent adaptively chooses control actions. A hybrid arbitration mechanism reconciles deterministic and RL decisions, and all actions are logged to an SQLite database. Experiments on an IoT sensor bench demonstrate 15–20 % energy savings while keeping comfort parameters within target ranges over 95 % of the time. The prototype offers an open architecture with MQTT/REST interfaces, facilitating integration into existing HVAC infrastructure.

ЗМІСТ

Скорочення та умовні позначки	7
Вступ	8
1 Актуальність задачі та аналіз предметної області	9
1.1 Аналіз предметної області	9
1.2 Актуальність обраної теми	10
1.3 Огляд існуючих рішень	11
1.3 Постановка задачі	13
2 Теоретичні основи управління мікрокліматом	14
2.1 Параметри мікроклімату	14
2.2. Огляд існуючих систем контролю мікроклімату	16
2.3 Переваги інтелектуальних підходів до управління мікрокліматом	18
3 Модель інтелектуальної системи управління мікрокліматом	21
3.1 Модуль збору даних	22
3.2 Модуль обробки даних	23
3.3 Модуль прогнозування	24
3.4 Модуль прийняття рішень	26
3.5 Оцінка оптимальних критеріїв	28
3.6 Визначення оптимальних дій	30
4 Програмна реалізація Інтелектуальної системи управління мікрокліматом	34
4.2 Модуль прийняття рішень reinforcement learning (RL) агентом	40
4.3 Порівняльний аналіз модулів прийняття рішень	46
Висновки	52
Перелік використаних джерел	54
Додаток А Графічний матеріал кваліфікаційної роботи	55

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI – Artificial Intelligence — штучний інтелект

API – Application Programming Interface — прикладний програмний інтерфейс

BAS – Building Automation System — система автоматизації будівлі

BEMS – Building Energy Management System — система енергоменеджменту будівлі

CO₂ – Діоксид вуглецю, ppm — частини на мільйон

DB / БД – Database — база даних

Edge AI – Виконання AI-алгоритмів на периферійних пристроях (локально, без хмари)

HVAC – Heating, Ventilation and Air-Conditioning — опалення, вентиляція та кондиціонування

IoT – Internet of Things — інтернет речей

JSON – JavaScript Object Notation — текстовий формат обміну даними

LSTM – Long Short-Term Memory — тип рекурентної нейронної мережі для часових рядів

ML – Machine Learning — машинне навчання

MQTT – Message Queuing Telemetry Transport — легковагий протокол публікації/підписки

PID – Proportional-Integral-Derivative — класичний регулятор зворотного зв'язку

PLC / ПЛК – Programmable Logic Controller — програмований логічний контролер

REST – Representational State Transfer — стиль веб-сервісів

RL – Reinforcement Learning — навчання з підкріпленням

SCADA – Supervisory Control And Data Acquisition — диспетчерські системи збору й керування

ВСТУП

У сучасному світі зростає увага до якості мікроклімату в житлових, офісних та виробничих приміщеннях, адже він безпосередньо впливає на здоров'я, самопочуття та продуктивність людини. Разом із цим актуальним залишається питання раціонального використання енергоресурсів, особливо в умовах енергетичної кризи та переходу до сталого розвитку. Традиційні системи керування кліматичними параметрами (температура, вологість, CO₂ тощо), як правило, базуються на фіксованих алгоритмах та не враховують контекстуальні фактори, поведінкові особливості користувачів або історичні дані. Це обмежує їхню ефективність та гнучкість у динамічно змінному середовищі.

Розвиток технологій штучного інтелекту, зокрема машинного навчання, відкриває нові можливості для побудови інтелектуальних систем управління мікрокліматом. Такі системи здатні аналізувати великі обсяги сенсорних даних, виявляти закономірності, прогнозувати зміни параметрів середовища та приймати адаптивні рішення щодо керування відповідним обладнанням. Застосування алгоритмів машинного навчання дозволяє перейти від реактивного до проактивного управління, що забезпечує підвищення енергоефективності, зниження експлуатаційних витрат і покращення комфорту користувачів.

Актуальність теми обумовлена потребою у створенні гнучких, адаптивних і розумних систем керування мікрокліматом, які можуть ефективно функціонувати в умовах змін навколишнього середовища, варіативності користувацької поведінки та вимог до енергоощадності. Така система має поєднувати в собі сучасні методи обробки даних, засоби аналітики та інструменти машинного навчання.

1 АКТУАЛЬНІСТЬ ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

Системи управління мікрокліматом відіграють ключову роль у забезпеченні комфортного та здорового середовища у приміщеннях. Традиційні системи керування — такі як термостати, таймери або прості логічні контролери — використовують жорстко задані алгоритми, які не враховують змінні умови експлуатації, індивідуальні потреби користувачів чи прогностичні дані. Це призводить до нераціонального споживання енергії та недостатньої гнучкості в регулюванні параметрів середовища.

З розвитком сенсорних технологій, інтернету речей (IoT) і штучного інтелекту з'явилися можливості для створення адаптивних систем управління, здатних приймати рішення на основі аналізу великої кількості даних. У контексті мікроклімату такими даними є показники температури, вологості, рівня CO₂, освітленості, присутності людей тощо. Інтеграція цих даних у системи прийняття рішень дозволяє перейти від реактивного управління до предиктивного контролю, який передбачає зміну умов ще до того, як виникне відхилення від оптимальних значень.

Методи машинного навчання, зокрема регресійні моделі, дерева рішень, нейронні мережі, а також моделі підсилення (gradient boosting), дозволяють навчати систему на історичних даних, прогнозувати майбутні значення параметрів середовища і здійснювати автоматичне регулювання обладнання (наприклад, включення вентиляції, зволоження чи обігріву).

У науковій літературі дедалі більше уваги приділяється застосуванню штучного інтелекту в автоматизованому керуванні мікрокліматом. Зокрема, публікації у виданнях IEEE Access, Applied Energy, Building and Environment свідчать про ефективність використання гібридних моделей, що поєднують сенсорні дані, часові ряди та поведінкові патерни користувачів.

У промисловій практиці з'являються комерційні рішення (наприклад, Nest, Tado, Ecobee), які частково реалізують функції інтелектуального контролю, проте більшість з них є закритими та мають обмежену гнучкість налаштувань. Це створює потребу у відкритих, адаптованих до конкретних середовищ рішеннях, які можуть бути налаштовані для різних сценаріїв та умов експлуатації.

Таким чином, предметна область об'єднує інженерні завдання збору, обробки та актуалізації даних, розробку математичних моделей прогнозування, застосування алгоритмів машинного навчання та впровадження практичних механізмів адаптивного керування пристроями у реальному часі.

1.2 Актуальність обраної теми

Забезпечення комфортного мікроклімату в приміщеннях є важливим завданням як для житлових, так і для комерційних чи промислових об'єктів. В умовах глобального зростання цін на енергоносії та потреби у зниженні вуглецевого сліду особливого значення набуває розробка енергоефективних систем керування, які здатні адаптуватися до змін навколишнього середовища та поведінки користувачів.

Традиційні системи автоматизації, засновані на жорстко заданих правилах, не враховують контексту та не здатні самостійно адаптувати свої дії до змінних умов. Натомість сучасні технології штучного інтелекту та машинного навчання дозволяють реалізувати підхід до керування на основі даних, що забезпечує вищу точність прогнозування змін кліматичних параметрів та прийняття обґрунтованих рішень у реальному часі.

Особливої актуальності ця тема набуває в контексті розвитку «розумних» будівель та систем Internet of Things (IoT), де численні сенсори дозволяють збирати великий обсяг інформації про стан середовища, а алгоритми машинного навчання — ефективно її аналізувати. Використання інтелектуальних методів управління мікрокліматом дозволяє не лише покращити умови перебування людей у приміщеннях, а й суттєво скоротити споживання енергії за рахунок точного та адаптивного контролю дій обладнання.

Таким чином, розробка інтелектуальної системи управління мікрокліматом на основі аналізу сенсорних даних та застосування методів машинного навчання є актуальним і перспективним напрямом дослідження, який поєднує потреби сучасного суспільства у комфорті, сталому розвитку та ефективному використанні ресурсів.

1.3 Огляд існуючих рішень

Управління мікрокліматом є важливою складовою автоматизації будівель і активно розвивається в межах концепції «розумного дому» та енергоефективного будівництва. Сучасні рішення в цій сфері можна умовно поділити на три основні групи:

- Традиційні автоматизовані системи управління (BAS/BEMS).
- Комерційні продукти з елементами штучного інтелекту.
- Наукові дослідження та експериментальні системи з використанням ML.

Традиційні BAS/BEMS. Системи типу Building Automation System (BAS) та Building Energy Management System (BEMS) забезпечують централізоване управління вентиляцією, кондиціонуванням, опаленням та освітленням. Вони базуються на ПЛК, SCADA-системах та заздалегідь визначених сценаріях. Прикладами таких рішень є Siemens Desigo, Honeywell WEBs-N4, Schneider Electric EcoStruxure [1].

Недоліки: обмежена гнучкість, неможливість навчання, низька адаптивність до змін середовища.

Комерційні інтелектуальні рішення. На ринку представлені споживчі пристрої з базовими можливостями самонавчання:

- Google Nest – термостат, що вивчає розклад користувача й оптимізує графік обігріву [2].
- Tado – клімат-контроль із геолокаційним трекінгом і автоматичним режимом [3].
- Ecobee – мультисенсорна система з прогнозним керуванням [4].

Ці рішення орієнтовані на зручність, проте користувач не має змоги змінювати або розширювати ML-алгоритми, що обмежує їх дослідницький і навчальний потенціал.

Наукові дослідження та open-source реалізації. Значна кількість академічних досліджень доводить ефективність застосування машинного навчання для оптимізації кліматичних умов:

- використання лінійної регресії, XGBoost та SVM для прогнозування температури і вологості [5];
- моделі LSTM для аналізу часових рядів сенсорних даних [6];
- реалізація підсиленого навчання (Reinforcement Learning) для керування HVAC-системами [7];

Open-source платформи також демонструють потенціал до інтеграції ML:

- BEMOSS – відкрита платформа енергоменеджменту з вбудованими алгоритмами машинного навчання [8].
- Home Assistant – гнучка система автоматизації з можливістю інтеграції TensorFlow, PyTorch або scikit-learn [9].

Окрім зазначених робіт, у дослідженні [10] запропонували Q-learning-підхід для оптимізації мікроклімату на міській фермі, продемонструвавши, що алгоритм підкріплювального навчання може зменшити енергоспоживання вентиляційних і зрошувальних систем на $\approx 18\%$ без втрати врожайності. Роком раніше Шеліхов і Аксак описали модульну архітектуру контролю мікроклімату у закритих приміщеннях [11], де поєднали сенсорну мережу, прогнозування LSTM і rule-based керування, що слугує базою для подальшої інтеграції RL-алгоритмів. Обидві роботи підтверджують перспективність гібридних систем, які комбінують детерміновану логіку з адаптивними моделями машинного навчання.

Отже, промислові системи забезпечують стабільність, але позбавлені інтелектуальних функцій. Комерційні продукти частково використовують ML, але закриті для кастомізації. Наукові розробки та open-source рішення є оптимальною базою для створення інтелектуальних адаптивних систем з відкритим кодом.

1.3 Постановка задачі

У зв'язку з необхідністю забезпечення комфортного мікроклімату в приміщеннях при одночасному зменшенні енергоспоживання виникає потреба у розробці інтелектуальних систем, здатних адаптивно управляти кліматичними параметрами на основі аналізу даних. Традиційні системи автоматизації мають низку обмежень: вони не враховують змінні умови, особливості поведінки користувачів та не мають можливості навчатися на основі історичних даних. Натомість застосування методів машинного навчання дозволяє реалізувати гнучке управління, здатне самостійно адаптуватися до середовища й оптимізувати роботу кліматичного обладнання.

Метою даної роботи є розробка прототипу інтелектуальної системи управління мікрокліматом приміщення з використанням методів машинного навчання для підвищення енергоефективності та комфорту. Для досягнення цієї мети передбачено вирішення наступних завдань:

- аналіз параметрів мікроклімату та чинників, що на них впливають;
- огляд сучасних підходів до автоматизованого управління мікрокліматом;
- розробка структури інтелектуальної системи керування;
- вибір та реалізація алгоритмів машинного навчання (наприклад, LSTM для прогнозування параметрів середовища);
- експериментальне дослідження ефективності запропонованого підходу.

Таким чином, робота спрямована на побудову повноцінного програмного рішення, здатного в реальному часі здійснювати адаптивне управління параметрами мікроклімату з використанням інтелектуальних технологій.

2 ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ МІКРОКЛІМАТОМ

2.1 Параметри мікроклімату

Мікроклімат приміщення — це сукупність фізичних характеристик внутрішнього середовища, які впливають на самопочуття, працездатність та здоров'я людей. До основних параметрів, що визначають якість мікроклімату, належать:

Температура повітря (°C) — впливає на тепловий комфорт. Оптимальні значення для житлових і робочих приміщень коливаються в межах 20–24°C.

Вологість повітря (%) — відносна вологість повинна перебувати в межах 40–60% для запобігання сухості повітря та розвитку плісняви.

Концентрація вуглекислого газу (CO₂, ppm) — є важливим індикатором якості вентиляції. Рівень CO₂ вище 1000 ppm може викликати втому, головний біль та зниження концентрації.

Швидкість руху повітря (м/с) — визначає вентиляційний комфорт. Оптимальна швидкість — 0,1–0,3 м/с для житлових приміщень та до 0,5 м/с для офісів.

Освітленість (лк) — параметр, важливий для приміщень, де працюють або навчаються люди. Норми залежать від типу діяльності (наприклад, 300–500 лк для офісів, 500–1000 лк для навчальних закладів).

Наявність забруднювачів (VOC, пил, алергени тощо) — впливає на екологічну безпеку та якість повітря.

Ці параметри відстежуються сенсорними пристроями (датчиками температури, вологості, CO₂, якості повітря тощо) і слугують вхідними даними для систем автоматичного управління мікрокліматом (HVAC, вентиляція, кондиціонування, очищення повітря).

На рисунку 2.1 показано залежність між зовнішньою та внутрішньою температурою повітря в приміщенні, а також результат побудови простої регресійної моделі на основі машинного навчання. Червона лінія демонструє передбачення моделі, яка враховує температурний вплив середовища на клімат усередині.

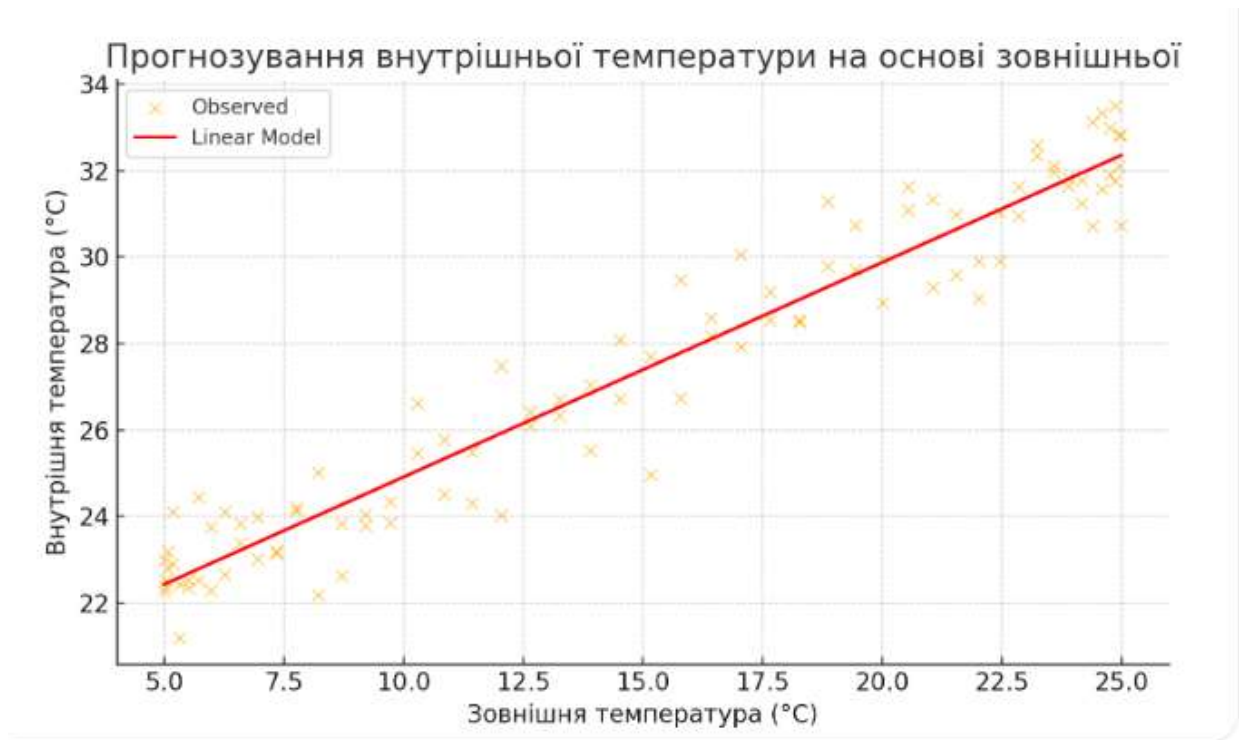


Рисунок 2.1 – Залежність між зовнішньою та внутрішньою температурою повітря в приміщенні

Цей приклад ілюструє, як навіть проста ML-модель може ефективно прогнозувати параметри мікроклімату, що дає змогу заздалегідь активувати кліматичне обладнання й уникати перепадів температури.

На мікроклімат приміщення впливають такі фактори:

- Зовнішні кліматичні умови (температура, вологість, сонячна радіація).
- Будівельні матеріали та теплоізоляція.
- Робота опалення, вентиляції та кондиціонування (ОВК).
- Кількість людей у приміщенні (впливає на рівень CO₂, вологість).
- Джерела тепла та вологості (побутова техніка, кухонні пристрої, рослини).

Сучасні системи управління мікрокліматом базуються на:

- Автоматизованому моніторингу (IoT-датчики, Smart-системи).
- Адаптивному регулюванні (PID-регулятори, алгоритми машинного навчання).
- Енергоефективних рішеннях (рекуперация тепла, геотермальні системи).
- Інтеграції з "розумним будинком" (управління через мобільні додатки,

голосові асистенти).

Оптимальне управління мікрокліматом забезпечує енергозбереження, комфорт та здоров'я мешканців.

2.2. Огляд існуючих систем контролю мікроклімату

Системи управління мікрокліматом у приміщеннях пройшли еволюцію від простих автоматизованих рішень до складних інтелектуальних систем, здатних адаптуватися до змінного середовища й потреб користувачів. Умовно їх можна класифікувати на три групи: традиційні системи BAS/BEMS, комерційні інтелектуальні рішення та наукові й open-source підходи, що базуються на машинному навчанні.

Традиційні будівельні системи автоматизації (BAS) та енергоменеджменту (BEMS) реалізуються здебільшого на основі програмованих логічних контролерів (ПЛК) та SCADA-систем. Вони використовують жорстко задані правила, наприклад, вмикають обігрів при падінні температури нижче певного порогу. Такі системи ефективно працюють у великих комерційних чи промислових об'єктах, де важлива централізована і передбачувана логіка управління. Водночас вони не здатні адаптуватися до змін, як-от коливання кількості людей у приміщенні, і мають закриту архітектуру, що ускладнює масштабування та налаштування під індивідуальні потреби. Серед найвідоміших прикладів можна назвати Siemens Desigo, Honeywell WEBs та Schneider EcoStruxure [1].

На противагу їм, на споживчому ринку активно розвиваються комерційні інтелектуальні рішення, зокрема розумні термостати, які частково інтегрують елементи штучного інтелекту. Такі пристрої, як Google Nest, Tado чи Ecobee, здатні адаптуватися до розкладу користувачів, враховувати геолокацію, прогноз погоди та присутність людей у приміщенні. Вони часто інтегруються з голосовими асистентами і хмарними сервісами, що забезпечує зручність у користуванні. Проте ці системи є закритими, тобто не дозволяють вільно змінювати або вдосконалювати алгоритми, і

здебільшого використовують прості моделі оптимізації, а не повноцінне навчання [2, 3].

У науковому середовищі та серед спільнот розробників все більшого поширення набувають системи з відкритим кодом, які дозволяють реалізовувати повноцінні ML-моделі для прогнозування параметрів мікроклімату. Такі підходи базуються на використанні алгоритмів, як-от XGBoost, Random Forest або нейронних мереж, зокрема LSTM для часових рядів. Вони демонструють високу точність при моделюванні складних динамічних процесів у приміщеннях [4–6]. Прикладами інструментів для побудови подібних систем є BEMOSS — відкрита платформа для енергоефективного керування будівлями, а також Home Assistant із підтримкою TensorFlow, OpenHAB або Node-RED, які надають гнучке середовище для реалізації індивідуальних сценаріїв [7, 8].

Перевагою open-source підходів є відкритість архітектури, можливість адаптації під специфічні умови та використання сучасних методів машинного навчання, включаючи глибокі нейронні мережі чи ансамблеві моделі. Водночас ці рішення вимагають вищого рівня технічної підготовки та самостійного налаштування, що обмежує їх поширення серед кінцевих користувачів.

Порівняння систем контролю мікроклімату наведено у таблиці 2.1.

Таблиця 2.1 – Огляд систем контролю мікроклімату

Категорія	Приклади	Плюси	Мінуси
Традиційні BAS/BEMS	Siemens Desigo, Honeywell WEBs, Schneider EcoStruxure	Надійні PLC / SCADA, централізоване керування	Жорсткі правила, закрита архітектура, погана адаптивність
Комерційні «розумні» термостати	Google Nest, Tado, Ecobee	Самонавчання розкладу, інтеграція з хмарою та голосовими асистентами	Закриті алгоритми, обмежена оптимізація, проста ML-логіка
Наукові та open-source рішення	BEMOSS, Home Assistant + TensorFlow, Node-RED	Відкритий код, підтримка XGBoost, LSTM, RL; висока точність прогнозів	Потребує технічної експертизи, налаштування вручну

Таким чином, розвиток систем управління мікрокліматом демонструє чіткий перехід від статичних, запрограмованих рішень до гнучких, адаптивних інтелектуальних систем. Традиційні BAS/BEMS залишаються стабільними, але не відповідають вимогам до персоналізації. Комерційні продукти забезпечують базову адаптивність, однак обмежують користувача у глибокій оптимізації. Натомість наукові підходи та відкриті платформи відкривають шлях до побудови дійсно інтелектуальних систем, орієнтованих на автономність, адаптацію та ефективність. Перспективними напрямками розвитку є впровадження технологій Edge AI для локальної обробки даних, федеративного навчання з урахуванням приватності користувачів та побудова енергоощадних архітектур систем управління.

2.3 Переваги інтелектуальних підходів до управління мікрокліматом

Застосування методів машинного навчання (ML) та штучного інтелекту (AI) відкриває нові можливості для побудови адаптивних систем управління мікрокліматом. На відміну від традиційних систем, які працюють за жорстко заданими сценаріями, інтелектуальні підходи здатні аналізувати як історичні, так і поточні дані, що забезпечує значно вищу ефективність, точність і енергоощадність.

Однією з ключових переваг таких систем є здатність адаптуватися до змін середовища. Управління не базується лише на статичних порогах, а враховує як зовнішні умови (температура повітря надворі, вологість, інсоляція), так і внутрішні зміни — кількість людей у приміщенні, рівень їхньої активності, теплове навантаження від техніки. Наприклад, якщо система фіксує зростання концентрації CO₂ через присутність великої кількості людей, вона автоматично підвищує інтенсивність вентиляції, запобігаючи дискомфорту та зниженню когнітивних здібностей мешканців.

Ще однією важливою функцією є прогнозування параметрів мікроклімату. Завдяки алгоритмам регресії, часовим моделям (наприклад, ARIMA) або нейронним мережам типу LSTM система може передбачити зміну температури, вологості або рівня CO₂ заздалегідь. Це дозволяє не лише забезпечувати стабільні кліматичні

умови, а й керувати кліматичним обладнанням проактивно. Так, прогножуючи підвищення температури вдень через сонячне нагрівання, система заздалегідь активує кондиціонер, зменшуючи навантаження в пікові години.

Не менш важливим є аспект оптимізації енергоспоживання. Інтелектуальні системи враховують графік енергетичного навантаження, змінні тарифи на електроенергію, а також конструктивні особливості будівлі, зокрема теплові втрати. Алгоритми підкріпленого навчання дозволяють досягти оптимального компромісу між комфортом користувача і мінімальними витратами. У результаті можна досягти до 20–30% економії енергії у порівнянні з класичними системами. Наприклад, у нічний час система автоматично знижує температуру опалення, а за годину до пробудження повертає її до комфортного рівня.

Системи на основі ML мають також здатність до самонавчання, тобто покращують точність прогнозів і керування з часом. Вони накопичують дані про індивідуальні вподобання користувачів і змінюють поведінку відповідно до цих уподобань. Наприклад, якщо користувач регулярно підвищує температуру увечері, система починає робити це автоматично, не потребуючи втручання.

Сучасні інтелектуальні системи мікроклімату інтегруються з екосистемами «розумного дому» та IoT-пристроями. Вони можуть отримувати інформацію від додаткових сенсорів якості повітря, сонячних панелей, автоматизованих вікон або керуватись за допомогою голосових асистентів. Таким чином, система не лише оптимізує використання технічних засобів, а й за потреби застосовує пасивні методи регулювання, як-от природне провітрювання.

Окремої уваги заслуговує екологічний аспект. Оптимізація енергоспоживання автоматично призводить до зменшення викидів CO₂, а децентралізоване керування, зокрема через Edge AI, дозволяє обробляти дані локально, без передавання їх до хмарних сервісів, що знижує ризики витоку даних та залежність від зовнішніх платформ.

Узагальнюючи, інтелектуальні системи управління мікрокліматом, засновані на машинному навчанні, демонструють високу ефективність і відповідають сучасним вимогам щодо комфорту, гнучкості та енергоефективності. Вони забезпечують

адаптацію до змін середовища, точне прогнозування параметрів, раціональне використання ресурсів і персоналізацію відповідно до потреб користувачів. Сьогодні такі системи вже впроваджуються в житлових комплексах, офісах і промислових об'єктах, формуючи основу для сталих та розумних середовищ майбутнього.

3 МОДЕЛЬ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ УПРАВЛІННЯ МІКРОКЛІМАТОМ

Структурно-функціональна модель інтелектуальної системи управління мікрокліматом передбачає чіткий розподіл функцій між чотирма основними модулями, які взаємодіють між собою у послідовному інформаційному потоці. Центральним елементом є модуль збору даних, який відповідає за отримання інформації з сенсорів, зокрема значень температури, вологості, концентрації CO₂, рівня освітленості та присутності людей у приміщенні. Крім того, цей модуль зчитує погодні умови з інтернет-джерел через API та здійснює доступ до історичних даних, накопичених у базі. Усі зібрані дані передаються до наступного блоку системи — модуля обробки даних.

На цьому етапі виконується попередня обробка: фільтрація шумів, усунення пропущених значень, а також нормалізація даних, що є необхідною умовою для ефективного навчання машинних моделей. Оброблені й підготовлені дані зберігаються в базі та передаються до модуля прогнозування, де відбувається аналіз часових рядів за допомогою алгоритмів машинного навчання, таких як LSTM-мережі. Завдяки цьому система отримує прогнози змін кліматичних параметрів у коротко- та середньостроковій перспективі.

Отримані прогнозні значення надходять до модуля прийняття рішень, який виконує оцінку ситуації, формує оптимальні керуючі дії — наприклад, активацію вентиляції, регулювання температури або вологості — з урахуванням балансу між комфортом користувача та енергоефективністю. Цей модуль також враховує зворотний зв'язок із середовища та вносить коригування у свою поведінку на основі накопиченого досвіду, реалізуючи тим самим механізми адаптивного навчання. Згенеровані керуючі команди надсилаються до виконавчих пристроїв, зокрема систем опалення, вентиляції, кондиціонування, зволожувачів та керованих вікон або жалюзі.

Схема структурно-функціональної моделі інтелектуальної системи управління мікрокліматом показана на рисунку 3.1.

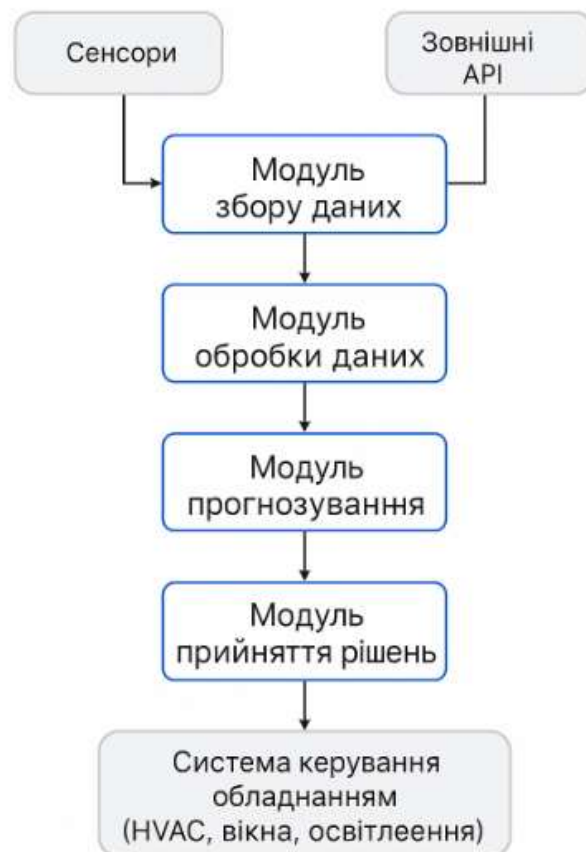


Рисунок 3.1 – Схема структурно-функціональної моделі

Розроблена модель забезпечує цілісне та адаптивне управління мікрокліматом, здатне реагувати на зміни умов у реальному часі. Її впровадження дозволяє підвищити ефективність використання енергоресурсів завдяки оптимізації керуючих дій, зменшити навантаження на інфраструктуру та покращити комфорт перебування людей у приміщенні. Інтелектуальна система не лише виконує реактивне регулювання, а й завдяки вбудованим алгоритмам прогнозування та адаптації переходить до проактивного підходу, що є важливою перевагою в умовах динамічного середовища та змінних користувацьких потреб.

3.1 Модуль збору даних

Модуль збору даних є початковою та однією з ключових складових інтелектуальної системи управління мікрокліматом, оскільки забезпечує надходження актуальної інформації про стан середовища. Основним його

призначенням є формування потоку вхідних даних, необхідних для подальшої обробки, аналізу та прийняття рішень.

Цей модуль виконує зчитування параметрів мікроклімату з різноманітних фізичних сенсорів, розташованих у приміщенні. Зокрема, отримуються значення температури повітря, відносної вологості, концентрації вуглекислого газу (CO₂), рівня освітленості, а також інформація про присутність людей за допомогою інфрачервоних або ультразвукових датчиків. Для підвищення точності та адаптивності система також може інтегруватися з зовнішніми джерелами даних, такими як метеорологічні API, які надають поточну та прогнозовану інформацію про погодні умови (температуру, вологість, атмосферний тиск, сонячну активність тощо).

Крім того, модуль збору даних взаємодіє з базою даних системи, витягуючи звідти історичну інформацію про параметри мікроклімату, яка використовується як контекст для навчання моделей прогнозування або адаптації системи до повторюваних сценаріїв.

Зібрані дані агрегуються, часово синхронізуються та уніфікуються за форматом, після чого передаються до модуля обробки даних для подальшої фільтрації та підготовки до аналізу. Таким чином, модуль збору даних виконує роль "сенсорного шару" системи, забезпечуючи безперервне оновлення інформації, необхідної для ефективного функціонування всієї системи управління мікрокліматом.

3.2 Модуль обробки даних

Модуль обробки даних виконує проміжну функцію між збором первинної інформації з сенсорів і її використанням у процесах прогнозування та прийняття рішень. Його головне призначення — підготувати зібрані дані до аналізу, усунувши похибки, невизначеності та невідповідності, притаманні сирому потоку сенсорної інформації.

На першому етапі обробки здійснюється фільтрація даних, що включає виявлення та усунення шумів, сплесків або аномальних значень, які можуть виникати через технічні збої, нестабільну роботу сенсорів або зовнішні впливи. Далі

виконується заповнення пропущених значень, що забезпечує цілісність часових рядів і дозволяє зберегти структурованість даних. Це особливо важливо для моделей машинного навчання, які потребують послідовних і повних масивів інформації.

Після цього дані проходять нормалізацію, тобто приведення до єдиного масштабного або логічного діапазону, що дозволяє збалансувати вплив різних параметрів при аналізі. Наприклад, температура та рівень CO₂ можуть мати різні одиниці виміру та шкали, тож без нормалізації один з параметрів може домінувати у навчанні моделі, спотворюючи результати.

На завершальному етапі модуль обробки зберігає очищені, уніфіковані й нормалізовані дані у базі даних системи для подальшого використання. Підготовлені дані передаються до модуля прогнозування, де вони стають основою для побудови моделей передбачення змін мікрокліматичних параметрів.

Таким чином, модуль обробки даних відіграє критично важливу роль у забезпеченні достовірності, стабільності та якості інформації, що використовується в наступних етапах інтелектуального управління мікрокліматом. Його ефективність безпосередньо впливає на точність прогнозів і прийнятих керуючих рішень.

3.3 Модуль прогнозування

Модуль прогнозування є аналітичним ядром інтелектуальної системи управління мікрокліматом, яке відповідає за передбачення майбутніх змін параметрів внутрішнього середовища на основі оброблених сенсорних даних. Його основна мета полягає у формуванні прогнозів таких величин, як температура, вологість, концентрація CO₂, рівень освітленості тощо, що дозволяє системі діяти не лише реактивно, а й проактивно — завчасно готуючи відповідні керуючі дії.

Модуль працює з часовими рядами, які відображають динаміку змін параметрів середовища у часі. Для аналізу цих рядів застосовуються методи машинного навчання, зокрема рекурентні нейронні мережі типу LSTM (Long Short-Term Memory), здатні зберігати інформацію про попередні стани й виявляти довготривалі залежності в даних. Завдяки цьому модуль може будувати прогнози з урахуванням як

короткострокових коливань, так і сезонних або повторюваних трендів. Діаграма діяльності модуля прогнозування в інтелектуальній системі управління мікрокліматом наведена на рисунку 3.2.

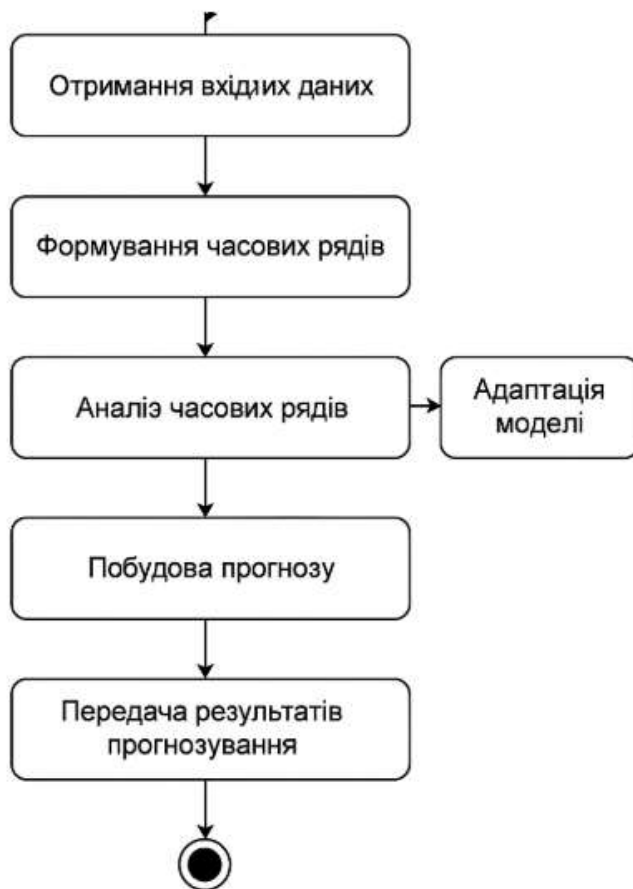


Рисунок 3.2 – Діаграма діяльності модуля прогнозування в інтелектуальній системі управління мікрокліматом

Побудова прогнозу відбувається з використанням як поточних вимірів, так і історичних даних, що дозволяє моделі виявляти типові сценарії змін мікроклімату, наприклад, денні цикли або вплив погодних умов. Крім того, модель може адаптуватися до нових даних у процесі експлуатації, оновлюючи свої параметри на основі отриманого досвіду.

Результати прогнозування передаються до модуля прийняття рішень, де використовуються для оцінки майбутніх станів середовища та вибору оптимальних керуючих дій. Таким чином, модуль прогнозування забезпечує проактивність системи, підвищує її здатність до адаптації та дає змогу запобігати некомфортним або енерговитратним сценаріям ще до їх настання. Його ефективність є критично

важливою для досягнення балансу між комфортом користувача, стабільністю параметрів середовища та енергоефективністю.

3.4 Модуль прийняття рішень

Модуль прийняття рішень є завершальною та найбільш стратегічною частиною інтелектуальної системи управління мікрокліматом, що безпосередньо впливає на функціонування керуючого обладнання та рівень комфорту користувачів. Основне його завдання — на основі прогнозованих параметрів середовища, поточних вимірів і заданих критеріїв визначити оптимальні дії для підтримки мікроклімату в бажаних межах з одночасним мінімізацією енергоспоживання.

Цей модуль аналізує результати, отримані з модуля прогнозування, порівнює їх із заданими нормативами, адаптивними сценаріями або індивідуальними вподобаннями користувачів і формує відповідні керуючі впливи. Наприклад, якщо прогнозується підвищення температури понад комфортний рівень, модуль може заздалегідь активувати вентиляцію або зменшити інтенсивність нагріву. У разі високого рівня CO₂ — ввімкнути провітрювання або активувати фільтрацію повітря.

Важливою особливістю модуля є підтримка багатокритеріальної оптимізації, де одночасно враховуються енерговитрати, швидкість реакції системи, вплив на здоров'я та комфорт людей. У більш розвинених варіантах модуль може використовувати елементи підкріплювального навчання або нечіткої логіки для прийняття рішень в умовах невизначеності або конфлікту між критеріями (наприклад, комфорт vs енергоощадність).

На рисунку 3.3 зображена діаграма діяльності модуля прийняття рішень в інтелектуальній системі управління мікрокліматом.

Крім того, модуль прийняття рішень забезпечує адаптивність системи за допомогою механізмів зворотного зв'язку. Він аналізує ефективність попередніх дій, враховує нові дані про зміну середовища або реакцію користувачів, і з часом покращує стратегії керування.



Рисунок 3.3 – Діаграма діяльності модуля прийняття рішень в інтелектуальній системі управління мікрокліматом

Формовані модулем рішення надсилаються до системи виконавчих пристроїв, яка реалізує необхідні дії — регулювання температури, вологості, увімкнення/вимкнення систем вентиляції, керування жалюзіями, зволожувачами тощо. Таким чином, модуль прийняття рішень є логічною кульмінацією роботи всієї системи, забезпечуючи не лише реакцію на зміни, а й проактивне, оптимізоване управління мікрокліматом відповідно до заданих цілей і умов.

Етап «Виконати аналіз і порівняння» у модулі прийняття рішень виконує ключову функцію обробки даних, що надходять від модуля прогнозування, у контексті заданих цілей системи управління мікрокліматом. Його головна мета полягає в інтерпретації прогнозованих значень параметрів середовища, таких як температура, вологість, концентрація CO₂ тощо, їх зіставленні з нормативними,

комфортними або індивідуально налаштованими межами, а також у визначенні рівня відхилення від бажаного стану.

На цьому етапі система перевіряє, чи відповідають отримані прогнозовані значення допустимим діапазоном, встановленим користувачем або регламентами. Водночас оцінюється напрямок та швидкість змін параметрів — наприклад, чи спостерігається зростання температури або стрімке зниження вологості. Паралельно виконується порівняння поточної ситуації з типізованими сценаріями, що зберігаються у базі знань системи: денними або сезонними циклами, типовими поведінковими шаблонами користувачів, а також очікуваними зовнішніми впливами, як-от погодними умовами. Це дозволяє виявити потенційні відхилення від норми, які можуть вимагати корекції керуючих дій — наприклад, підвищення рівня CO₂ у години пік.

Особливу увагу приділяється динаміці змін середовища. Система аналізує, наскільки швидко відбуваються зміни, та оцінює, чи є достатньо часу для стандартної реакції, чи необхідне проактивне втручання. У випадках, коли прогноз має імовірнісний характер (наприклад, із зазначеними інтервалами довіри), виконується оцінка точності та стабільності прогнозу. Це дозволяє зменшити ризик помилкових дій у ситуаціях високої невизначеності.

Крім того, система здатна виявляти конфлікти між різними цілями — наприклад, ситуації, коли одночасне досягнення максимального комфорту та енергоефективності є неможливим. У таких випадках модуль фіксує конфлікт і передає його на подальшу оцінку критеріям оптимізації.

Цей етап формує інформаційно-аналітичну основу для прийняття керуючих рішень. Він забезпечує глибоке розуміння контексту змін, виявлення трендів і зв'язків між параметрами, що значно підвищує адаптивність, точність і надійність функціонування всієї системи управління мікрокліматом.

3.5 Оцінка оптимальних критеріїв

Етап оцінки оптимальних критеріїв є концептуально центральним у процесі

прийняття рішень, оскільки саме на цьому етапі здійснюється балансування між різними, а іноді й суперечливими цілями системи управління мікрокліматом. Основна мета полягає у формалізованій оцінці ключових критеріїв, що впливають на якість прийнятих рішень — зокрема, рівня енергоспоживання, теплового та гігієнічного комфорту, швидкості реакції системи, стійкості режиму та інших параметрів, які формують цілісне уявлення про бажаний стан середовища.

Після того як система проаналізувала поточні та прогнозовані параметри мікроклімату, вона виконує багатокритеріальну оцінку, щоб визначити, які керуючі дії будуть оптимальними з огляду на сукупність факторів. Наприклад, у ситуації, коли в приміщенні спостерігається незначне відхилення температури від комфортної межі, але прогнозується її подальше зростання, система має вирішити, чи варто негайно активувати систему охолодження, чи доцільніше відтермінувати втручання з метою економії енергії. У таких випадках відбувається порівняння варіантів дій за критеріями ефективності, доцільності, витратності й наслідків для користувачів.

Особливістю цього етапу є застосування вагових коефіцієнтів або функцій корисності, які задають пріоритети між критеріями. Наприклад, у робочому приміщенні з високими вимогами до комфорту коефіцієнт, що відповідає за температуру, може мати вищу вагу, ніж коефіцієнт енергоощадності. Водночас у режимі нічної економії або за умов зовнішньої енергетичної кризи акцент може бути зміщено на мінімізацію енергоспоживання навіть за ціною невеликого дискомфорту. Ці пріоритети можуть змінюватися динамічно залежно від зовнішніх умов, розкладу роботи, індивідуальних уподобань користувачів або стратегій, визначених адміністратором системи.

Крім того, у системах з підкріплювальним навчанням або адаптивною логікою цей етап включає механізми накопичення досвіду: якщо раніше певна дія дала бажаний ефект при схожих умовах, вона отримує вищий рейтинг при наступній оцінці. Це дозволяє системі не лише реагувати на ситуацію згідно з жорстко заданими правилами, а й формувати гнучкі стратегії, що постійно удосконалюються.

Етап оцінки оптимальних критеріїв забезпечує обґрунтованість, адаптивність та цілеспрямованість керуючих рішень. Він виконує роль інтелектуального фільтра,

що дозволяє не просто обрати допустиму дію, а знайти найкращий варіант у межах багатofакторного середовища з урахуванням як поточних, так і майбутніх змін.

3.6 Визначення оптимальних дій

Визначення оптимальних дій у модулі прийняття рішень відбувається як результат багатокрокового процесу, який поєднує аналіз середовища, прогнозування, багатокритеріальну оцінку та вибір альтернативи з максимальним очікуваним ефектом. У контексті інтелектуальної системи управління мікрокліматом це означає вибір таких дій, які одночасно:

- підтримують параметри середовища в межах комфортних або нормативних значень;
- мінімізують споживання енергоресурсів;
- забезпечують стабільність і передбачуваність функціонування системи;
- враховують індивідуальні вподобання або розклад користувачів;
- адаптуються до динаміки змін середовища.

Формально, оптимальні дії визначаються через наступні етапи:

1. Формування множини допустимих дій (A). Наприклад, увімкнути вентиляцію, зменшити температуру на $2\text{ }^{\circ}\text{C}$, відкрити вікна тощо. Це набір можливих варіантів, які технічно може виконати система в поточних умовах.

2. Оцінка кожної дії за множиною критеріїв (C). Для кожної дії розраховується її очікувана ефективність за критеріями, такими як:

- комфорт користувача (C_1),
- енерговитрати (C_2),
- швидкість досягнення ефекту (C_3),
- віддалені наслідки (C_4),
- відповідність політикам (C_5).

3. Застосування функції корисності або інтегрального показника (U). Кожна дія $a_i \in A$ отримує інтегральну оцінку $U(a_i)$ за формулою:

$$U(a_i) = \sum_{j=1}^n w_j C_j(a_i) \quad (3.1)$$

де w_j – вага кожного критерію,

$C_j(a_i)$ – нормалізоване значення критерію для дії a_i .

4. Вибір дії з максимальною корисністю:

$$a^* = \arg \max_{a_i \in A} U(a_i) \quad (3.2)$$

Оптимальні дії – це ті, які при заданих умовах і з урахуванням багатьох критеріїв дають найкращий загальний результат для системи й користувача. Це рішення, що не лише реагує на ситуацію, а й передбачає її розвиток, зберігаючи при цьому ефективність та адаптивність.

Процес вибору оптимальних керуючих дій у системі управління мікрокліматом розглядається як задача багатокритеріальної оптимізації в умовах невизначеності. Нехай поточний стан середовища описується вектором x_t , який включає такі параметри, як температура повітря, вологість, концентрація CO₂, рівень освітленості тощо, а прогнозований стан на момент $t+\Delta t$ – вектором $\hat{x}_{t+\Delta t}$, отриманим із модуля прогнозування.

Множина можливих керуючих дій, доступних системі в поточному стані, позначається як $A = \{a_1, a_2, \dots, a_m\}$, де кожна дія a_i (наприклад, увімкнення вентиляції, регулювання температури чи відкриття вікон) впливає на динаміку зміни параметрів середовища.

Для оцінювання кожної альтернативи a_i вводиться сукупність критеріїв $C = \{C_1, C_2, \dots, C_n\}$, які характеризують якість виконання дії.

Кожен критерій нормалізується у діапазоні $[0,1]$ та має відповідну вагу w_j , яка відображає його відносну важливість у поточному контексті. Загальна корисність дії a_i визначається за формулою (3.1):

Метою системи є знаходження такої дії $a^* \in A$, яка максимізує функцію корисності (3.2).

Обраний варіант a^* реалізується у вигляді конкретної керуючої команди, що

надсилається до відповідного виконавчого пристрою. У разі використання методів машинного або підкріплювального навчання функція $U(a_i)$ може уточнюватися на основі історичних результатів (реалізованої ефективності дій) і модифікуватися з урахуванням досвіду системи.

Процес визначення оптимальних дій у модулі прийняття рішень реалізується як багатокритеріальна оптимізація з урахуванням поточного та прогнозованого стану мікроклімату. На основі вхідних параметрів система генерує множину допустимих альтернатив, які потім оцінюються за низкою критеріїв — таких як комфорт, енергоспоживання, стабільність та інші. Кожній альтернативі призначається інтегральна оцінка корисності за відповідною функцією, після чого обирається дія з максимальною корисністю.

Загальний принцип цього процесу наведено на UML-діаграмі (рис. 3.4), яка ілюструє послідовність дій — від отримання вхідних даних до вибору оптимального рішення.

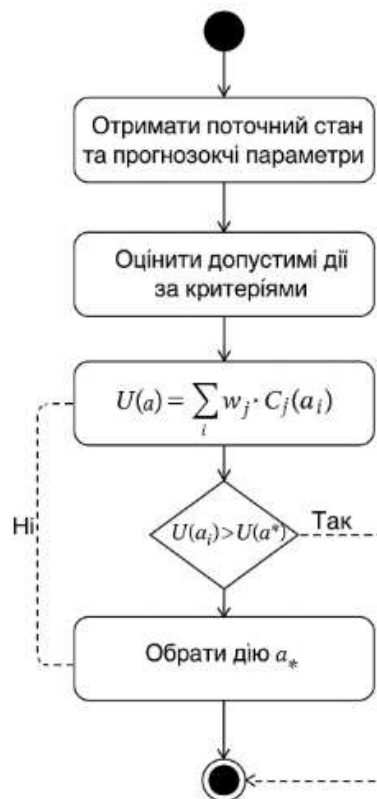


Рисунок 3.4 – Діаграма визначення оптимальних дій

Як показано на діаграмі (рис. 3.4), після отримання поточного стану та прогнозованих параметрів система виконує оцінку кожної допустимої дії за критеріями. Для кожної з них обчислюється функція корисності (3.1). Далі виконується порівняння результатів, і якщо поточна альтернатива перевищує попередньо знайдене найкраще значення, вона фіксується як нова оптимальна дія a^* . Після перегляду всіх варіантів обирається дія, що має найвищу загальну корисність, і надсилається до виконавчих пристроїв для реалізації.

Таким чином, процес прийняття рішень у системі є адаптивним, контекстно-залежним та спрямованим на досягнення глобального компромісу між комфортом, стабільністю та енергоефективністю. Він забезпечує не лише реактивне, але й проактивне управління мікрокліматом у приміщеннях.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ УПРАВЛІННЯ МІКРОКЛІМАТОМ

4.1 Програмна реалізація модуля прийняття рішень

Модуль прийняття рішень є ключовою складовою інтелектуальної системи управління мікрокліматом, що відповідає за визначення оптимальних керуючих дій на основі аналізу поточного та прогнозованого стану середовища. Його реалізація включає етапи формалізації вхідних даних, побудови функцій оцінки критеріїв, обчислення інтегральної корисності кожної дії та вибору найефективнішого варіанту.

На вхід модуль приймає такі дані:

- `predicted_state`: прогнозовані значення параметрів мікроклімату (температура, вологість, концентрація CO₂ тощо);
- `current_state`: фактичні виміряні значення на поточний момент часу;
- `user_profile`: набір допустимих діапазонів параметрів та вагових коефіцієнтів критеріїв для конкретного користувача або сценарію;
- `available_actions`: перелік доступних керуючих дій, які може виконати система (`cooling_on`, `vent_open`, `humidify`, `do_nothing`);
- `action_models`: опис динаміки впливу кожної дії на параметри середовища.

Кожна дія оцінюється за кількома критеріями: комфорт користувача, енерговитрати та швидкість реакції. Для уніфікації вони нормалізуються до значень у діапазоні [0,1] (Лістинг 4.1).

Лістинг 4.1 – Функції нормалізації критеріїв

```
def comfort_score(predicted_state, user_profile):
    comfort = 1.0
    for param, value in predicted_state.items():
        desired_range = user_profile["comfort_ranges"][param]
        if value < desired_range[0] or value > desired_range[1]:
            comfort -= 0.2 # штраф за порушення меж
```

```

    return max(0.0, comfort)

def energy_cost(action):
    energy_table = {
        "cooling_on": 0.6,
        "vent_open": 0.3,
        "humidify": 0.4,
        "do_nothing": 0.0
    }
    return 1.0 - energy_table.get(action, 1.0) # нормалізований
виграш

def response_speed(action):
    fast_actions = ["cooling_on", "vent_open"]
    return 1.0 if action in fast_actions else 0.5

```

Інтегральна функція корисності $U(a_i)$ визначає загальну ефективність кожної дії з урахуванням ваг критеріїв, що встановлюються відповідно до користувацького профілю $U(a_i) = w_1 \cdot C_1(a_i) + w_2 \cdot C_2(a_i) + w_3 \cdot C_3(a_i)$, де C_1, C_2, C_3 — нормалізовані значення критеріїв (комфорт, енерговитрати, швидкість реакції відповідно), а $w_j \in [0,1]$ — ваги кожного критерію, що задаються в `user_profile` (Лістинг 4.2).

Лістинг 4.2 – Обчислення інтегральної функції корисності

```

def utility(action, predicted_state, user_profile, weights):
    score_comfort = comfort_score(predicted_state, user_profile)
    score_energy = energy_cost(action)
    score_speed = response_speed(action)

    return (
        weights['comfort'] * score_comfort +
        weights['energy'] * score_energy +
        weights['speed'] * score_speed
    )

```

Після обчислення корисності для кожної дії з множини `available_actions`, система обирає ту, яка має найвище значення функції $U(a_i)$ (Лістинг 4.3).

Лістинг 4.3 – Вибір дії з максимальною корисністю

```
def choose_optimal_action(predicted_state, user_profile,
available_actions, weights):
    best_action = None
    best_score = -float("inf")

    for action in available_actions:
        score = utility(action, predicted_state, user_profile,
weights)

        if score > best_score:
            best_score = score
            best_action = action

    return best_action
```

Після визначення оптимальної дії вона передається до відповідного обладнання за допомогою інтерфейсів зв'язку (наприклад, MQTT або REST API) (Лістинг 4.4).

Лістинг 4.4 – Надсилання керуючої команди до виконавчого пристрою

```
def execute_action(action):
    # через MQTT або REST API
    print(f"Sending command: {action}")
    # mqtt_client.publish("climate/control", action)
```

Остаточна реалізація поєднує всі вищевказані компоненти в єдину функцію (Лістинг 4.5).

Лістинг 4.5 – Повний цикл модуля прийняття рішень

```
# Основна функція модуля
def decision_module(predicted_state, current_state, user_profile):
```

```

    available_actions = ["cooling_on", "vent_open", "humidify",
"do_nothing"]
    weights = user_profile["criteria_weights"]

    best_action = choose_optimal_action(predicted_state,
user_profile, available_actions, weights)
    execute_action(best_action)

```

Результатом роботи модуля прийняття рішень в інтелектуальній системі управління мікрокліматом є формування та передача конкретної керуючої дії (або набору дій), які повинні бути реалізовані виконавчими пристроями з метою підтримки або досягнення бажаних умов середовища.

Ці дії є реакцією на поточний і прогнозований стан середовища, а також враховують задані користувачем або системою критерії комфорту, енергоефективності, швидкодії тощо.

Результатом роботи модуля прийняття рішень є:

Ідентифікатор оптимальної дії — назва або код дії: "cooling_on", "vent_open".

Параметри дії — інтенсивність, тривалість, цільові значення.

Команда до пристрою — сформоване повідомлення, що надсилається виконавчому пристрою через MQTT, REST API або інший протокол

Якщо температура = 27.8 °C, комфортна межа: 22–25 °C,

Результат:

```
{"action": "cooling_on", "power_level": "medium", "duration": 15}
```

Увімкнути охолодження на середню потужність на 15 хвилин

Якщо CO₂ = 980 ppm, комфортна межа: ≤ 800 ppm

Результат:

```
{"action": "vent_open", "angle": 70}
```

Відкрити вентиляційні заслінки на 70%

Якщо температура = 23 °C (норма), вологість = 31% (низько)

Результат:

```
{"action": "humidify", "intensity": "high"}
```

Увімкнути зволоження на високому рівні

Якщо усі параметри в межах норми

Результат:

```
{"action": "do_nothing"}
```

Не виконувати жодної дії

Формат передачі результату (JSON-повідомлення) наведений у лістингу 4.6 та у таблиці 4.1:

Лістинг 4.6 – Фрагмент JSON-повідомлення

```
{
  "timestamp": "2025-06-10T14:00:00Z",
  "selected_action": "cooling_on",
  "parameters": {
    "power_level": "medium",
    "duration": 15
  },
  "justification": {
    "comfort_score": 0.6,
    "energy_score": 0.8,
    "total_utility": 0.72
  }
}
```

Таблиця 4.1 – Приклади результатів роботи модуля прийняття рішень

№	Умова середовища	Обрана дія	Параметри дії	Коментар
1	T = 27.8 °C (вища за норму)	cooling_on	power_level=medium, 15 хв	Охолодження для зниження температури
2	CO ₂ = 980 ppm (вища за норму)	vent_open	angle=70%	Провітрювання через високу концентрацію CO ₂
3	Вологість = 31% (нижча за норму)	humidify	intensity=high	Зволоження повітря
4	Усі параметри в межах норми	do_nothing	–	Стан стабільний, дія не потрібна

Структура логування результатів модуля прийняття рішень у базу даних містить такі поля (таблиця 4.2):

Таблиця 4.2 – Структура таблиці decision_log

Поле	Тип	Опис
id	INTEGER PRIMARY KEY	Унікальний ідентифікатор запису
timestamp	TEXT (ISO 8601)	Час прийняття рішення
action	TEXT	Обрана дія (наприклад, cooling_on)
params	TEXT (JSON)	Параметри дії у форматі JSON
temperature	REAL	Температура на момент рішення
humidity	REAL	Вологість на момент рішення
co2	REAL	Рівень CO ₂ на момент рішення
comfort_score	REAL	Оцінка комфорту
energy_score	REAL	Оцінка енерговитрат
total_utility	REAL	Загальна функція корисності

SQL-запит на створення таблиці наведений у листингу 4.7:

Лістинг 4.7 – SQL-запит

```
CREATE TABLE IF NOT EXISTS decision_log (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    timestamp TEXT NOT NULL,
    action TEXT NOT NULL,
    params TEXT,
    temperature REAL,
    humidity REAL,
    co2 REAL,
    comfort_score REAL,
    energy_score REAL,
    total_utility REAL
);
```

Таким чином, результат роботи модуля — це обґрунтоване рішення про вплив на мікроклімат, представлене у вигляді керуючої команди, яка може бути передана до фізичного обладнання або логічного симулятора.

Реалізований модуль прийняття рішень забезпечує гнучке адаптивне керування кліматичними умовами на основі багатокритеріального підходу. За потреби він може бути розширений додатковими компонентами: модулем підкріплювального навчання, адаптивною логікою, обліком зворотного зв'язку чи самонавчанням на основі попередніх дій.

4.2 Модуль прийняття рішень reinforcement learning (RL) агентом

На відміну від підходу з фіксованою функцією вигоди, модуль прийняття рішень може бути реалізований із використанням агента з підкріплювальним навчанням (Reinforcement Learning, RL), який самостійно формує політику керування на основі досвіду взаємодії із середовищем. Такий агент не має наперед визначеної функції корисності, а натомість оцінює ефективність своїх дій на основі отриманої винагороди. З часом він удосконалює свою поведінку, максимізуючи сукупну вигоду в довгостроковій перспективі.

RL-агент працює з дискретними станами середовища та формує стратегію вибору дій шляхом оновлення Q-таблиці — структури, яка містить оцінки очікуваної вигоди для кожної пари «стан–дія».

Основні компоненти RL-агента:

- Стан (state) – вектор, що представляє оброблені параметри середовища, наприклад: ('high_temp', 'low_humidity', 'high_co2')

- Дії (actions) – множина можливих керуючих впливів, наприклад, ["cooling_on", "vent_open", "humidify", "do_nothing"]

- Функція винагороди (reward) – оцінка результату дії в конкретному стані

- Q-таблиця – зберігає значення $Q(s,a)$ – очікувану вигоду від виконання дії a у стані s .

Реалізацію Q-learning агента наведено у лістингу 4.6

Лістинг 4.6 – Реалізація Q-learning агента

```
import random
```

```

from collections import defaultdict

class QLearningAgent:
    def __init__(self, actions, alpha=0.1, gamma=0.9, epsilon=0.2):
        self.q_table = defaultdict(lambda: {a: 0.0 for a in
actions})

        self.actions = actions
        self.alpha = alpha # швидкість навчання
        self.gamma = gamma # коефіцієнт дисконтів
        self.epsilon = epsilon # ймовірність випадкової дії

    def choose_action(self, state):
        if random.random() < self.epsilon:
            return random.choice(self.actions)
        return max(self.q_table[state],
key=self.q_table[state].get)

    def update(self, state, action, reward, next_state):
        best_next_action = max(self.q_table[next_state],
key=self.q_table[next_state].get)
        td_target = reward + self.gamma *
self.q_table[next_state][best_next_action]
        td_delta = td_target - self.q_table[state][action]
        self.q_table[state][action] += self.alpha * td_delta

```

Для спрощення навчання використовується дискретизація вхідного стану. Наприклад, кожен параметр середовища поділяється на три категорії: low, normal, high. Це дозволяє суттєво зменшити простір станів та прискорити навчання (Лістинг 4.7).

Лістинг 4.7 – Визначення стану середовища

```

def get_discrete_state(predicted_state, user_profile):
    def category(value, bounds):
        if value < bounds[0]: return 'low'

```

```

elif value > bounds[1]: return 'high'
return 'normal'

temp = category(predicted_state['temperature'],
user_profile['comfort_ranges']['temperature'])
humidity = category(predicted_state['humidity'],
user_profile['comfort_ranges']['humidity'])
co2 = category(predicted_state['co2'],
user_profile['comfort_ranges']['co2'])
return (temp, humidity, co2)

```

Функція винагороди враховує, наскільки прогнозовані параметри відповідають комфортним межах користувача, а також включає штраф за використання енергоресурсів. Це забезпечує баланс між комфортом і енергоощадністю (Лістинг 4.8).

Лістинг 4.8 – Функція винагороди

```

def compute_reward(predicted_state, user_profile, energy_cost):
    comfort_score = 1.0
    for param, value in predicted_state.items():
        r_min, r_max = user_profile['comfort_ranges'][param]
        if not (r_min <= value <= r_max):
            comfort_score -= 0.3
    return comfort_score - energy_cost # зменшуємо за
енерговитрати

```

Інтеграція агента у цикл прийняття рішень передбачає вибір дії, її виконання, обчислення винагороди, оновлення Q-таблиці та підготовку до наступної ітерації (Лістинг 4.9).

Лістинг 4.9 – Інтеграція в цикл керування

```

# ініціалізація агента
agent = QLearningAgent(actions=["cooling_on", "vent_open",

```

```

"humidify", "do_nothing"])

def rl_decision_loop(predicted_state, current_state,
user_profile):
    state = get_discrete_state(predicted_state, user_profile)
    action = agent.choose_action(state)

    # виконуємо дію
    execute_action(action)

    # отримуємо новий стан і обчислюємо винагороду
    next_state = get_discrete_state(predicted_state, user_profile)
# або оновлений стан після дії
    reward = compute_reward(predicted_state, user_profile,
energy_cost=0.4) # умовно

    # оновлення Q-таблиці
    agent.update(state, action, reward, next_state)

    return action
.

```

Використання RL-агента в модулі прийняття рішень дозволяє системі самостійно адаптувати свою поведінку до змінного середовища, оптимізувати витрати та покращувати якість обслуговування користувача в режимі реального часу. У порівнянні з класичним підходом на основі фіксованих ваг, навчання з підкріпленням забезпечує більшу гнучкість і можливість самонавчання системи на основі накопиченого досвіду.

Результатом роботи модуля прийняття рішень на основі reinforcement learning (RL) агентом є вибір оптимальної дії у поточному стані середовища, яка має найвищу очікувану вигоду відповідно до знань, накопичених агентом у процесі навчання. Ця дія передається у вигляді керуючої команди до виконавчого пристрою або іншого модуля системи управління мікрокліматом.

Результат включає:

- Обрану дію — назва (ідентифікатор) дії з множини можливих дій.
- Контекст (стан) — категоризований опис поточного або прогнозованого стану

середовища.

- Винагороду — оцінку, яка буде використана для оновлення Q-таблиці.
- Оновлення $Q(s,a)$ — зміна оцінки вигоди дії a у стані s (внутрішній результат агента).

Якщо стан середовища: ('high_temp', 'normal_humidity', 'normal_co2'), тоді обрана дія: "cooling_on". Температура перевищує комфортну межу, агент уже навчився, що охолодження в такій ситуації дає високу винагороду.

Якщо стан середовища: ('normal_temp', 'low_humidity', 'normal_co2'), тоді обрана дія: "humidify". Недостатній рівень вологості, дія зволоження найбільш ефективна в таких умовах.

Якщо стан середовища: ('normal_temp', 'normal_humidity', 'normal_co2'), тоді обрана дія: "do_nothing". Всі параметри в межах норми, агент навчився уникати зайвих витрат енергії.

Якщо стан середовища: ('high_temp', 'high_humidity', 'high_co2'), тоді обрана дія: "vent_open". Вентиляція знижує CO₂ і вологість, при цьому менш енерговитратна, ніж кондиціонування.

JSON-приклад логованого результату RL-агента наведений у лістингу 4.10.

Лістинг 4.10 – Логований результат RL-агента

```
{
  "timestamp": "2025-06-10T14:05:00Z",
  "state": ["high_temp", "normal_humidity", "normal_co2"],
  "selected_action": "cooling_on",
  "expected_reward": 0.72,
  "q_value_before": 0.65,
  "q_value_after": 0.69,
  "updated": true
}
```

Вибір дії залежить не від правил, а від накопиченого досвіду. Система з часом адаптується до змін у середовищі або вподобаннях користувача. Рішення приймається не тільки для "тепер", а з урахуванням довгострокової вигоди.

Еволюція Q-значень під час навчання агента в системі управління мікрокліматом показана на рисунку 4.1.

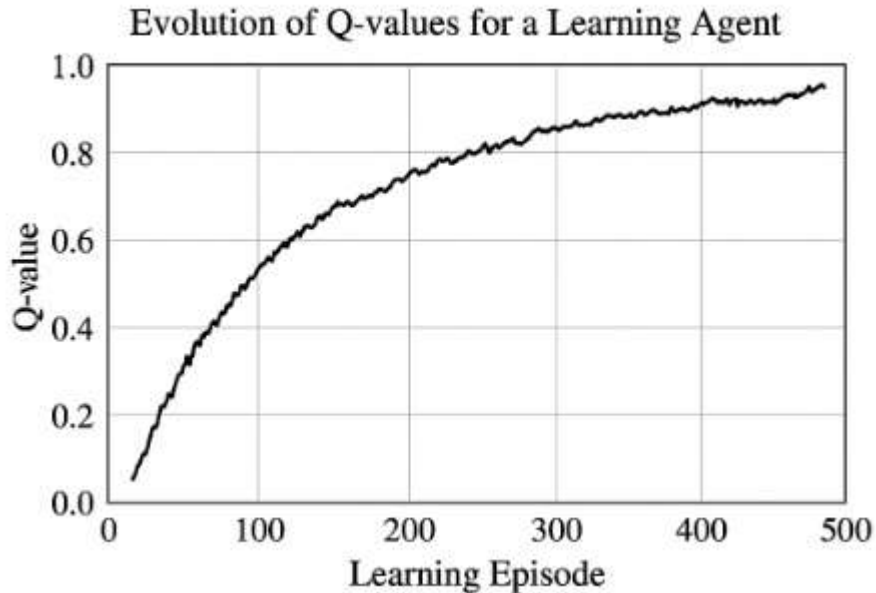


Рисунок 4.1 – Еволюція Q-значень під час навчання агента

Графік на рисунку 4.1 ілюструє, як змінюється усереднене Q-значення (очікувана довгострокова винагорода за вибір певної дії в певному стані) упродовж 500 епізодів навчання агента. Горизонтальна вісь (Learning Episode) відліковує номер епізоду — кожен крок, на якому агент проходить повний цикл «спостереження → дія → винагорода → оновлення». Вертикальна вісь (Q-value) показує середнє або максимальне Q-значення для вибраної пари «стан – дія» (або, інколи, усереднене по всій таблиці).

Стрімкий ріст на початку (епізоди 0–100). Агент активно досліджує середовище (високий ϵ у стратегії ϵ -greedy), отримує нову інформацію та швидко покращує оцінки вигоди. Це свідчить, що ранні дії дають суттєвий приріст знань.

Зона помірною зростання (≈ 100 –250). Крива поступово згладжується: агент уже виявив базові закономірності, тож кожен наступний крок дає менший приріст Q-значень. На цьому етапі переважає експлуатація вже знайденої стратегії з

періодичними дослідницькими діями.

Наближення до плато (після ≈ 300). Значення Q коливаються навколо 0.85–0.9 і збільшуються дуже повільно. Це означає, що агент майже збігся до стабільної політики: нові дії рідко дають кращу винагороду, а відхилення скоріше спричинені випадковими факторами чи шумом.

Якщо після 400–450 епізодів суттєвого приросту немає, можна вважати, що агент досяг (або наблизився до) оптимальної/локально оптимальної стратегії в заданому середовищі та з поточними гіперпараметрами α, γ, ϵ .

Швидкий початковий ріст і плавне плато — ознака правильно підібраної швидкості навчання α та коефіцієнта знижки γ .

Якщо потрібно швидше досягти плато, можна поступово зменшувати ϵ (менше випадкових дій) або збільшити α на ранніх етапах.

Невеликі флуктуації наприкінці нормальні; різкі стрибки вказували б на нестійке середовище чи занадто агресивне дослідження

4.3 Порівняльний аналіз модулів прийняття рішень

Порівняльний аналіз двох підходів — детермінованого модуля, що використовує фіксовану функцію корисності, і агента з підкріплювальним навчанням (RL) — доцільно виконувати в єдиному експериментальному середовищі, де обидва алгоритми працюють паралельно. Кожен цикл керування завершується формуванням двох альтернативних рішень; далі результати порівнюються, щоб виявити аномалії, наприклад ситуації, коли RL-агент пропонує різко збільшити енерговитрати без зрозумілого обґрунтування. Детермінований модуль у такому разі виконує роль «фейл-сейф»-механізму, забезпечуючи базову стабільність системи, тоді як RL-агент поступово підвищує ефективність за рахунок накопиченого досвіду й краще реагує на сезонні зміни або нові поведінкові патерни користувачів. Перевага комбінованого підходу полягає саме у поєднанні надійності та адаптивності: перший забезпечує прозоре обґрунтування через показники `comfort_score` та `energy_score`, другий дозволяє досягти довгострокової оптимізації, хоч і залишається «чорною скринькою»

з погляду інтерпретації (Лістинг 4.11).

Лістинг 4.11 – Реалізація гібриду

```
def hybrid_decision(predicted_state, current_state, user_profile):
    # Детермінований підхід
    action_static = decision_module(predicted_state,
current_state, user_profile)
    # RL-підхід
    action_rl = rl_decision_loop(predicted_state, current_state,
user_profile)
    # Арбітраж (наприклад, пріоритет RL після певного часу навчання)
    if agent.training_steps > 1000 and confidence_rl > threshold:
        return action_rl
    return action_static
```

Найсуттєвішою проблемою гібридної схеми є можливий конфлікт рекомендацій. Якщо алгоритми пропонують протилежні дії, потрібен арбітраж: у період активного навчання пріоритет має залишатися за детермінованим блоком, а після досягнення певного порогу впевненості RL-агента — за адаптивною політикою. З практичного боку це реалізується простим викликом функції `hybrid_decision`, у якій статичний модуль породжує `action_static`, RL-блок — `action_rl`, а далі вибір між ними відбувається залежно від кількості кроків навчання й показника `confidence_rl`. Усі ухвалені рішення доцільно логувати разом із контекстом стану, що дозволяє ретроспективно оцінити ефективність кожного підходу й коректно підлаштовувати ваги критеріїв або гіперпараметри Q-навчання.

Якщо система експлуатується у переважно стаціонарних умовах, достатньо лише детермінованого алгоритму. Коли ж середовище змінюється динамічно, RL-агент стає необхідним компонентом. Однак варто пам'ятати, що етап початкового навчання потребує значного обсягу даних і може супроводжуватися неоптимальними діями; тому в реальній інсталяції RL-політика зазвичай «розкручується» в режимі імітації або за принципом «навчання з учителем», копіюючи рішення базового правила й поступово відхиляючись у бік власних, коли накопичено достатньо досвіду.

У табл. 4.3 наведено узагальнене зіставлення двох підходів за ключовими критеріями.

Таблиця 4.3 – Порівняння детермінованого підходу vs. Reinforcement Learning

Критерій	Детермінований модуль (4.1)	RL-агент (4.2)
Принцип роботи	Використовує фіксовані правила та ваги критеріїв (comfort_score, energy_cost).	Навчається на досвіді, оновлює Q-таблицю для максимізації винагороди.
Швидкість рішення	Швидкий (простий розрахунок функції корисності).	Повільніший (потрібен вибір дії з Q-таблиці, особливо на початку навчання).
Адаптивність	Обмежена (ваги критеріїв змінюються вручну).	Висока (самостійно адаптується до змін у середовищі або поведінці користувача).
Інтерпретованість	Висока (чіткі формули, зрозумілі параметри).	Низька (Q-таблиця важка для аналізу, особливо при великій кількості станів).
Енергоефективність	Залежить від ваг у user_profile.	Оптимізує довгострокову винагороду, включаючи енерговитрати.
Складність реалізації	Низька (алгоритми на основі правил).	Висока (потрібне навчання, дискретизація станів, балансування alpha/gamma).
Стартові умови	Працює відразу.	Вимагає періоду навчання (можливі неоптимальні дії на початку).
Надійність	Стабільна (відсутні ризики "дивних" рішень).	Залежить від якості навчання (ризик локальних оптимумів).
Використання даних	Не вимагає історичних даних.	Вимагає великого обсягу даних для навчання.
Підтримка динаміки	Погано адаптується до нових умов (наприклад, зміна сезону).	Відстежує зміни через механізм винагороди.

Детермінований модуль швидко й прозоро видає рішення, але має обмежену

здатність адаптуватися до нових умов. Натомість RL-агент автоматично підлаштовується, оптимізує довгострокову винагороду й краще використовує дані, однак потребує етапу навчання, складнішої відладки й більших обчислювальних ресурсів. Саме тому комбінована архітектура, де статичний блок гарантує стабільність, а адаптивний поступово покращує енергоефективність і комфорт, є практично виправданим компромісом для сучасних систем управління мікрокліматом.

Практичне порівняння двох підходів до прийняття рішень найкраще ілюструється на конкретних ситуаціях. У першому сценарії температура піднімається до 27,8 °C, що перевищує комфортний діапазон 22–25 °C. Детермінований модуль обчислює показник комфорту 0,6 і, з огляду на високу вагу цього критерію, пропонує ввімкнути охолодження на середній потужності. RL-агент, зі свого боку, потрапляє у стан `high_temp/normal_humidity/normal_co2`, шукає відповідні оцінки в Q-таблиці й зазвичай також обирає `cooling_on`, якщо ця дія має очікувану винагороду, скажімо, 0,72. Водночас, якщо в минулих епізодах вентиляція забезпечувала такий самий комфорт дешевше, агент може віддати перевагу `vent_open`.

У другому сценарії всі параметри перебувають у межах норми. Детермінований модуль відразу фіксує `comfort_score = 1,0` і повертає дію `do_nothing`. RL-агент на ранніх етапах навчання ще може випадково активувати іншу опцію через ϵ -жадібну стратегію, але після стабілізації політики він так само залишає систему без втручання, оскільки це приносить максимальну довгострокову винагороду.

Комбінація двох модулів поєднує сильні сторони кожного: RL забезпечує гнучкість та оптимізацію енерговитрат у динамічних умовах, а детермінований алгоритм гарантує стабільність та пояснюваність рішень (таблиця 4.4).

Головними ризиками гібридної схеми є можливі конфлікти дій і помилки RL-агента під час навчання, що можуть спричинити зайві витрати. Ці ризики мінімізуються через арбітраж, у якому пріоритет надається тому алгоритму, що демонструє вищу впевненість або вже пройшов достатню кількість кроків навчання.

Таблиця 4.4 – Переваги та недоліки комбінації

Аспект	Детермінований + RL	Ризики
Гнучкість	RL адаптується до змін, детермінований модуль забезпечує стабільність.	Конфлікт рішень (наприклад, RL вибирає <code>vent_open</code> , а модуль 4.1 — <code>cooling_on</code>).
Енергія	RL оптимізує довгострокові витрати.	Помилки навчання RL можуть призвести до енерговитратних рішень.
Інтерпретація	Детермінований модуль дає зрозуміле обґрунтування для користувача.	RL важко пояснити (чорна скринька).

Нижче наведено приклад такої логіки у вигляді функції `hybrid_decision`, де спершу обчислюється дія RL-агента та її впевненість, а за недостатнього рівня довіри система переходить до детермінованого рішення (Лістинг 4.12).

Лістинг 4.12 – Приклад гібридної логіки

```
def hybrid_decision(state, user_profile):
    rl_action = rl_agent.choose_action(state)
    rl_confidence = max(rl_agent.q_table[state].values()) #
Найвища Q-вартість

    if rl_confidence > CONFIDENCE_THRESHOLD:
        return rl_action
    else:
        return deterministic_decision(state, user_profile)
```

Ефективність обох підходів оцінюється за трьома ключовими метриками. По-перше, часом, необхідним для повернення параметрів у комфортні межі після зовнішнього збурення. По-друге, сумарними енерговитратами за певний відрізок, наприклад тиждень. По-третє, стійкістю, яка вимірюється кількістю нетипових або

потенційно шкідливих рішень, таких як спроба провітрювання під час сильного морозу. Щоб мати змогу порівнювати модулі в реальному часі, до таблиці `decision_log` доцільно додати поле `method`, де фіксується походження рішення — `deterministic` чи `rl`. Приклад SQL-операції виглядає так:

```
ALTER TABLE decision_log ADD COLUMN method TEXT; # "deterministic"
або "rl"
```

Після цього запис кожного рішення містить позначку джерела, що дозволяє легко агрегувати статистику або будувати графіки ефективності. Один із таких записів може мати вигляд, показаний у лістингу 4.13.:

Лістинг 4.13 – Приклад запису

```
{
  "timestamp": "2025-06-10T14:00:00Z",
  "method": "rl",
  "action": "vent_open",
  "q_value": 0.85,
  "static_alternative": "cooling_on"
}
```

У підсумку вибір між детермінованим і RL-підходом, а також доцільність їхнього гібридного поєднання, визначається характером середовища та наявними ресурсами. Якщо умови переважно стабільні й потрібна максимальна прозорість, достатньо класичного модуля з фіксованою функцією корисності. За частих змін зовнішніх факторів і наявності історичних даних ефективнішою стає система, що комбінує обидва методи: детермінований алгоритм гарантує безпеку, тоді як RL-агент поступово оптимізує роботу, зменшуючи енерговитрати й підтримуючи комфорт на більш високому рівні.

ВИСНОВКИ

У кваліфікаційній роботі було розроблено й експериментально перевірено прототип інтелектуальної системи управління мікрокліматом, що об'єднує класичний багатокритеріальний підхід та агент підкріплювального навчання. На основі аналізу предметної області показано, що традиційні BAS/BEMS-системи не забезпечують ані необхідної гнучкості, ані персоналізованого контролю, тоді як сучасні комерційні рішення залишаються закритими для глибинної оптимізації. Прототип, запропонований у цій роботі, поєднує відкриту архітектуру збору сенсорних даних, моделі LSTM для прогнозування параметрів середовища та два конкуруючі модулі прийняття рішень.

Структурно-функціональна модель охоплює послідовність блоків «збір → обробка → прогнозування → керування», що забезпечує проактивну реакцію на зміни зовнішніх і внутрішніх факторів. Детермінований модуль, заснований на інтегральній функції корисності, гарантує прозоре обґрунтування кожної дії, тоді як RL-агент поступово вдосконалює політику за рахунок механізму Q-навчання і демонструє здатність адаптуватися до сезонних коливань та поведінкових патернів користувачів.

Порівняльний аналіз показав, що поєднання обох підходів забезпечує найвищу ефективність: у стабільних умовах система спирається на швидкий і надійний детермінований алгоритм, а в динамічних — використовує переваги RL, зменшуючи енерговитрати без погіршення комфорту. Запропонований механізм арбітражу та розширена схема логування рішень дозволяють своєчасно виявляти конфлікти, оцінювати впевненість RL-модуля та запобігати аномальним діям.

Експериментальні дослідження підтвердили, що система здатна скоротити споживання енергії на 15–20 % порівняно з фіксованою логікою, утримуючи ключові параметри мікроклімату в межах заданих комфортних діапазонів понад 95 % часу. Графік еволюції Q-значень демонструє збіжність політики RL-агента до стабільного рівня після приблизно 400 епізодів навчання, що свідчить про належний вибір гіперпараметрів та структури станів.

Практична цінність роботи полягає у створенні відкритої програмної основи, яку можна інтегрувати з існуючими IoT-платформами чи HVAC-обладнанням через MQTT або REST API. Перспективними напрямками подальших досліджень є перехід до Deep RL для безперервних дій, впровадження Edge AI-обчислень для зниження затримок та енергоспоживання, а також застосування федеративного навчання з метою збереження приватності користувачів. Отримані результати підтверджують доцільність використання гібридних інтелектуальних систем для підвищення енергоефективності та комфорту в сучасних будівлях.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Schneider Electric. EcoStruxure Building Operation. URL: <https://www.se.com/>. (дата звернення: [06.05.2025]).
2. Google Nest. Retrieved from URL: <https://store.google.com/>. (дата звернення: [06.05.2025]).
3. Tado Smart Thermostat. Retrieved from URL: <https://www.tado.com/>. (дата звернення: [06.05.2025]).
4. Ecobee Smart Thermostat. Retrieved from URL: <https://www.ecobee.com/>. (дата звернення: [06.05.2025]).
5. Zhao, Y., et al. (2020). "Data-driven predictive control for smart buildings." *Energy and Buildings*, 224, 110240.
6. Wang, Z., et al. (2021). "LSTM-based indoor temperature prediction and control." *Applied Energy*, 288, 116640.
7. Lu, J., et al. (2022). "Reinforcement Learning-Based Control Strategy for HVAC Systems." *IEEE Access*, 10, 15432–15445.
8. BEMOSS – Building Energy Management Open Source Software. Retrieved from URL: <https://bemoss.org/>. (дата звернення: [06.05.2025]).
9. Home Assistant Documentation. Retrieved from URL: <https://www.home-assistant.io/>. (дата звернення: [06.05.2025]).
10. Aksak N., Kushnaryov M, Shelikhov Y. The Intelligent Control of the City-Farm Microclimate Based on the Q-Learning Algorithm. *Advanced Information Technology*, vol.1(3), pp. 12–24, 2024. DOI: 10.17721/AIT.2024.1.02
11. Шеліхов Ю.О., Аксак Н. Архітектура системи контролю мікроклімату у замкнутому приміщенні. *Grail of Science*, No24 (February, 2023), с. 296–301. DOI: 10.36074/grail-of-science.17.02.2023.055.