

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «СИСТЕМА ОБРОБКИ ДАНИХ ТА ПРОГНОЗУВАННЯ
ЕНЕРГОСПОЖИВАННЯ ІОТ-ПРИСТРОЇВ»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

_____ Дмитро ТРОХИМЧУК

Виконав:
здобувач вищої освіти
група ІСД-42

Дмитро ТРОХИМЧУК

Керівник:
науковий ступінь,
вчене звання

Оксана ТКАЛЕНКО
к.т.н., доцент

Рецензент:
науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

_____ Каміла СТОРЧАК
«_____» _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ СТУДЕНТУ

Трохимчуку Дмитру Сергійовичу
(*прізвище, ім'я, по батькові здобувача*)

1. Тема кваліфікаційної роботи: «Система обробки даних та прогнозування енергоспоживання IoT-пристроїв»

керівник кваліфікаційної роботи Оксана ТКАЛЕНКО, к.т.н., доцент
(*ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання*)

затверджені наказом вищого навчального закладу від «24» лютого 2025 року
№ 56.

2. Строк подання кваліфікаційної роботи: 30 травня 2025 року.

3. Вихідні дані до кваліфікаційної роботи: протоколи DDS, CoAP, XMPP, MQTT, STOMP, SOAP, AMQP;
бібліотеки Pandas, NumPy, Sklearn;
np.random.seed = 42;
науково-технічна література з питань, пов'язаних з наукою про дані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд технологій та організацій у сфері Інтернету речей.

2. Система обробки даних та прогнозування енергоспоживання IoT-пристроїв.
3. Інтеграція IoT-технологій та штучного інтелекту.
4. Прогнозування на основі IoT-даних за допомогою штучного інтелекту.
5. Перелік ілюстративного матеріалу: *презентація*
 1. Принципи роботи Інтернету речей.
 2. Стек IoT-протоколів.
 3. Інтеграція IoT-технологій та штучного інтелекту.
 4. Алгоритми для використання в IoT.
 5. Типи машинного навчання. Етапи збору даних.
 6. Вибір середовища розробки моделі машинного навчання.
 7. Вибір алгоритму для навчання моделі ML.
 8. Оцінка якості ML-моделі.
6. Дата видачі завдання: 24 лютого 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	24.02 – 28.02.25	
2	Дослідження організацій у сфері IoT	03.03 – 07.03.25	
3	Дослідження особливостей машинного навчання та принципів обробки даних	10.03 – 14.03.25	
4	Технології штучного інтелекту	17.03 – 28.03.25	
5	Вивчення особливостей бібліотек Sklearn, Pandas, NumPy	31.03 – 10.04.25	
6	Реалізація ML-моделі	11.04 – 24.04.25	
7	Оформлення роботи: вступ, висновки, реферат	25.04 – 14.05.25	
8	Розробка демонстраційних матеріалів	15.05 – 30.05.25	

Здобувач вищої освіти

(підпис)

Дмитро ТРОХИМЧУК
(Ім'я, ПРІЗВИЩЕ)

Керівник роботи
кваліфікаційної роботи

(підпис)

Оксана ТКАЛЕНКО
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 63 стор., 23 рис., 10 табл., 28 джерел.

Мета роботи – розробка системи обробки даних, прогнозування на основі IoT-даних за допомогою штучного інтелекту.

Об'єкт дослідження – процес збору, обробки та аналізу даних для прогнозування температури, та енергоспоживання IoT-пристроїв.

Предмет дослідження – методи машинного навчання та алгоритми обробки даних.

Короткий зміст роботи: Здійснено огляд технологій та організацій у сфері Інтернету речей, характеристику сенсорних пристроїв, які збирають дані. Визначені елементи IoT-систем, приведено рекомендації щодо застосування протоколів IoT на ділянках IoT-системи. Досліджено особливості інтеграції IoT-технологій та штучного інтелекту, визначено роль машинного навчання в системі Інтернету речей. Проведений аналіз ML-алгоритмів для IoT. Розглянуті етапи збору даних в системах IoT. Виконане прогнозування на основі IoT-даних засобами машинного навчання. Розроблено модель машинного навчання та виконано оцінку якості моделі.

КЛЮЧОВІ СЛОВА: ДАНІ, ТЕХНОЛОГІЯ, МОВА ПРОГРАМУВАННЯ PYTHON, JUPYTER NOTEBOOK, АЛГОРИТМ, ПРОТОКОЛ, MACHINE LEARNING, МОДЕЛЬ, ШТУЧНИЙ ІНТЕЛЕКТ, IoT-ПРИСТРОЇ.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ОГЛЯД ТЕХНОЛОГІЙ ТА ОРГАНІЗАЦІЙ У СФЕРІ ІНТЕРНЕТУ РЕЧЕЙ.....	10
1.1 Організації та альянси, які працюють в області Інтернету речей.....	10
1.2 Принцип роботи Інтернету речей.....	14
1.3 Характеристика сенсорних пристроїв.....	16
1.4 Характеристика IoT-технологій.....	21
РОЗДІЛ 2. СИСТЕМА ОБРОБКИ ДАНИХ ТА ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ IoT-ПРИСТРОЇВ.....	24
2.1 Визначення елементів системи.....	24
2.2 Технології передавання даних.....	25
2.3 Протоколи обміну даними.....	31
РОЗДІЛ 3. ІНТЕГРАЦІЯ IoT-ТЕХНОЛОГІЙ ТА ШТУЧНОГО ІНТЕЛЕКТУ.....	37
3.1 Концепція AIoT. Машинне навчання для IoT.....	37
3.2 Аналіз ML-алгоритмів для IoT.....	39
3.3 Збір даних IoT.....	41
РОЗДІЛ 4. ПРОГНОЗУВАННЯ НА ОСНОВІ IoT-ДАНИХ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ.....	44
4.1 Підготовка середовища для створення та оцінки моделі машинного навчання.....	44
4.2 Створення набору даних.....	48
4.3 Створення матриці вхідних ознак.....	51
4.4 Вибір алгоритму для навчання моделі ML.....	53
4.5 Оцінка якості моделі регресії.....	55
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ.....	61
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	64

ВСТУП

Актуальність теми. З кожним роком кількість IoT-пристроїв стрімко зростає. Вони використовуються у «розумних будинках», містах, промисловості, охороні здоров'я та в багатьох інших популярних сферах. Це призводить до підвищеного енергоспоживання та необхідності його оптимізації. Багато IoT-пристроїв працюють на акумуляторах або отримують живлення від обмежених джерел. Ефективне прогнозування споживання енергії дозволяє продовжити час їх автономної роботи, зменшити витрати на обслуговування та підвищити загальну продуктивність систем. Оптимізація енергоспоживання сприяє зниженню навантаження на електромережі. Застосування методів машинного навчання (ML) та штучного інтелекту (AI) для аналізу енергоспоживання дозволяє робити точні прогнози, автоматично коригувати режими роботи пристроїв та мінімізувати втрати енергії. Сучасні енергетичні системи переходять на «розумні мережі», де важливу роль відіграє оптимізація споживання електроенергії у реальному часі. IoT-пристрої можуть інтегруватися в ці системи для більш ефективного управління. У промисловості прогнозування енергоспоживання допомагає оптимізувати виробничі процеси, зменшити експлуатаційні витрати та покращити ефективність обладнання. Системи аналізу споживання енергії можуть виявляти аномальну активність, що може бути ознакою кіберзагроз або несправностей у мережі.

Дослідження та впровадження систем прогнозування енергоспоживання IoT-пристроїв є важливим напрямом у технологічному розвитку. Вони сприяють зменшенню енергетичних витрат, підвищенню ефективності та підтримці сталого розвитку, що робить тему бакалаврської роботи досить актуальною.

Метою роботи є розробка системи обробки даних, прогнозування на основі IoT-даних за допомогою штучного інтелекту.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити IoT-технології;
- дослідити особливості інтеграції IoT-технологій та штучного інтелекту;

- визначити особливості машинного навчання в системі Інтернету речей;
- виконати прогнозування на основі IoT-даних засобами машинного навчання.

Об'єкт дослідження - процес збору, обробки та аналізу даних для прогнозування температури, та енергоспоживання IoT-пристроїв.

Предметом дослідження є методи машинного навчання та алгоритми обробки даних.

Методи дослідження: аналіз літературних джерел та огляд існуючих рішень; метод моделювання; методи математичної статистики; метод комп'ютерного моделювання.

Наукова новизна отриманих результатів полягає в оптимізації енергоспоживання IoT-пристроїв, поєднанні методів аналізу та обробки даних з алгоритмами штучного інтелекту для покращення ефективності прогнозування.

Практична значущість кваліфікаційної бакалаврської роботи. Тема "Система обробки даних та прогнозування енергоспоживання IoT-пристроїв" має високу практичну значущість, оскільки результати дослідження можуть бути застосовані у реальних умовах для підвищення ефективності використання енергії в IoT-системах. Автоматизоване керування режимами роботи IoT-пристроїв допоможе зменшити витрати енергії без втрати функціональності. Виробничі IoT-системи зможуть автоматично адаптувати режими роботи, виходячи з прогнозованих навантажень. Отримані результати можуть бути корисними для розробників IoT-систем, компаній у сфері енергетики, міських інфраструктур та промислових підприємств.

Апробація результатів бакалаврської роботи:

1. Трохимчук Д. С. «Використання алгоритмів пошуку в Python». Тези доповіді на VI Міжнародній науково-технічній конференції «Сучасний стан та перспективи розвитку IoT». – Київ, 15 квітня 2025 року.

1 ОГЛЯД ТЕХНОЛОГІЙ ТА ОРГАНІЗАЦІЙ У СФЕРІ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Організації та альянси, які працюють в області Інтернету речей

Інтернет речей (Internet of Things, IoT) набуває широкої популярності у сучасному світі завдяки своїм широким можливостям для автоматизації, оптимізації процесів та покращення якості життя.

Завдяки розвитку бездротових технологій (5G, Wi-Fi 6, LPWAN) підключення пристроїв стало простішим і ефективнішим. Очікується, що до 2030 року кількість IoT-пристроїв у світі перевищить 30 мільярдів. IoT дозволяє компаніям та приватним користувачам автоматизувати рутинні процеси, що зменшує витрати та підвищує продуктивність. Наприклад, у промисловості розумні датчики контролюють роботу обладнання та попереджають про несправності, що запобігає дорогим простоям.

Ключовим аспектом у сучасному світі є розвиток штучного інтелекту (Artificial Intelligence, AI) та аналізу даних. Поєднання IoT та AI надасть можливість аналізувати великі обсяги даних у реальному часі та приймати оптимальні рішення. Наприклад, у «розумних містах» датчики IoT зможуть керувати дорожнім рухом для зменшення заторів.

Популярність «розумних будинків», переносних пристроїв (фітнес-трекерів, смарт-годинників) та інтелектуальних побутових приладів постійно зростає. Смарт-технології допомагають людям економити час, енергію та підвищувати комфорт. IoT сприяє зменшенню витрат енергії та води, що особливо важливо в умовах глобального потепління та енергетичної кризи. Розумні лічильники та системи енергоменеджменту дають змогу користувачам контролювати та оптимізувати споживання ресурсів.

Виробництва впроваджують IoT для автоматизації, контролю якості та прогнозного обслуговування обладнання. Розвиток цифрових двійників (Digital Twins) дозволяє створювати віртуальні копії заводів і аналізувати їх роботу у реальному часі. IoT допомагає моніторити стан пацієнтів на відстані, що особливо актуально під час пандемій та для літніх людей. «Розумні медичні пристрої»

можуть автоматично відправляти дані лікарям, що покращує якість медичного обслуговування.

IoT-камери, «розумні замки» та системи контролю доступу підвищують рівень безпеки у будинках, офісах та містах. У сфері логістики IoT-датчики дозволяють відстежувати місцезнаходження та стан вантажів у реальному часі.

Популярність Інтернету речей зростає завдяки його здатності підвищувати ефективність, автоматизувати процеси, знижувати витрати та покращувати якість життя. Подальший розвиток технологій (5G, AI, хмарні обчислення) ще більше прискорить впровадження IoT у всі сфери життя.

Досліджено організації та альянси, які працюють в області Інтернету речей.

AIOTI - альянс з іновацій Інтернету речей, який був затверджений Єврокомісією з метою підтримки розвитку європейської екосистеми IoT, включаючи розробку стандартів та норм.

Альянс AllSeen - індустріальна група AllSeen, яка об'єднує 180 учасників, просуває ідею широкого розповсюдження протоколу взаємодії AllJoyn між пристроями та системами через IoT.

ETSI - Європейський інститут телекомунікаційних стандартів (ETSI) займається розробкою стандартів безпеки даних, управління даними, передавання та обробки даних для підключення мільярдів «розумних речей» в одну комунікаційну мережу.

IEC 62443/ISA99 - комітет промислової автоматизації та контролю систем безпеки, який розробляє стандарти, технічні звіти та процедури для впровадження безпечних систем промислової автоматизації та контролю. IEEE (включаючи P2413) - Інститут інженерів з електротехніки (IEEE) здійснює спеціальну ініціативу IoT та має інформаційну службу для технічного співтовариства, яке займається дослідженням, реалізацією та використанням IoT-технологій. IERC - Європейська рада по дослідженню Інтернету речей, яка координує поточну діяльність в області IoT по всій Європі.

Цільова інженерна група Інтернету (IETF) - основний орган затвердження Інтернет-стандартів, який має директорат IoT, який координує відповідні зусилля

серед своїх робочих груп, оцінює специфікації з точки зору відповідності і відстежує діяльність, яка пов'язана з IoT, в інших групах стандартизації. ІІС - консорціум промислового Інтернету (ІІС) поєднав зусилля з ОІС, щоб прискорити впровадження промислового архітектурного шаблону для IoT. У 2015 році ІІС випустив еталонну архітектуру для IoT [7, 8].

Форум з управління використанням Інтернету (IGF) спонсорує динамічну коаліцію IoT, яка проводить відкриті збори для обговорення проблем використання IoT. Консорціум Інтернету речей - промислова група, яка займається вивченням споживачів та просвітою учасників ринку для прискорення визнання IoT-продуктів та послуг [9-11].

Альянс ІР для «розумних об'єктів» IPSO, який займається розвитком Інтернету речей, просуває використання протоколу ІР для підключення «розумних об'єктів» шляхом інформування, досліджень та реклами. У 2014 році організація ISO/IEC/JTC-1 випустила Попередній звіт про Інтернет речей, а також Звіт про «розумні міста». Група має постійні комітети в обох сферах.

Група «Інтернет та харчові продукти» ISOC - спеціальна група, яка веде дискусію про стандарти технічної інфраструктури, які будуть актуальними для харчової галузі у майбутньому. Міжнародний телекомунікаційний союз (ITU) заснував Глобальну ініціативу по стандартам Інтернету речей, яка завершила свою роботу у липні 2015 року, після чого була сформована нова Дослідницька група 20 (Study Group 20), яка займається, головним чином, IoT-додатками. Альянс MAPI - альянс промисловців для продуктивності та інновацій (MAPI). Працює над стратегією Індустрія 4.0 у сфері промислового застосування Інтернету речей.

Компанія OASIS розробляє відкриті протоколи для забезпечення інтероперабельності IoT-пристроїв. Група вибрала протокол MQTT (Message Queuing Telemetry Transport) в якості основного протоколу обміну повідомленнями між IoT-пристроями та оптимізувала протокол MQTT-S-N для бездротових сенсорних мереж. OASIS заснувала три технічних комітети Інтернету речей, які займаються питаннями MQTT та двох інших стандартів: AMQP

(Advanced Message Queuing Protocol) та oBIX (OASIS Open Building Information Exchange).

Мультивендорна компанія oneM2M займається розробкою архітектури та стандартів міжмашинної комунікації, головним чином, у сферах телемедицини, а також промислової та побутової автоматизації. Мета компанії - розробка загальної платформи управління та контролю M2M для впровадження в апаратне та програмне забезпечення. Online Trust Alliance - група постачальників систем безпеки, яка запропонувала початковий варіант платформи для IoT-додатків у сферах безпеки, конфіденційності та стійкості систем.

Консорціум Open Interconnection Consortium (OIC) займається розробкою загальної комунікаційної платформи, яка базується на промислових стандартах для бездротового підключення та управління потоком даних між IoT-пристроями. OIC спонсорує проєкт IoTivity - систему відкритих вихідних кодів для взаємного підключення пристроїв.

The Open Management Group - консорціум по розробці технічних стандартів, який займається декількома IoT-стандартами, включаючи протокол DDS (Data Distribution Service), мову IFML (Interaction Flow Modeling Language), а також моделюванням надійності, загроз та уніфікованих компонентів для систем реального часу та вбудованих систем.

Open Web Application Security Project - відкритий проєкт забезпечення безпеки веб-додатків (OWASP), який спонсорує проєкт IoT Топ-10 з наданням списку найбільш вразливих атак областей IoT з метою роз'яснення проблем безпеки виробникам, розробникам та користувачам.

Комітет Smart Grid Interoperability Panel (SGIP) заснував ініціативу EnergyIoT, яка концентрується на нових можливостях Інтернету речей в області енергетики. Енергетичний проєкт OpenFMB об'єднує загальні енергетичні моделі даних та передавання даних в IoT-сфері для створення системи Open Field Message Bus.

Група постачальників рішень «розумного будинку» Thread Group займається розробкою загальних мережних протоколів для побутових IP-пристроїв (побутових приладів, засобів освітлення, систем безпеки) [2, 3].

1.2 Принцип роботи Інтернету речей

Принцип роботи Інтернету речей включає 4 етапи (рис. 1.1):

- 1) Зчитування інформації за допомогою датчиків.
- 2) Передавання даних від датчиків до хмарних сховищ.
- 3) Обробка даних, які отримані за допомогою датчиків.
- 4) Передавання інформації на інтерфейс користувача.

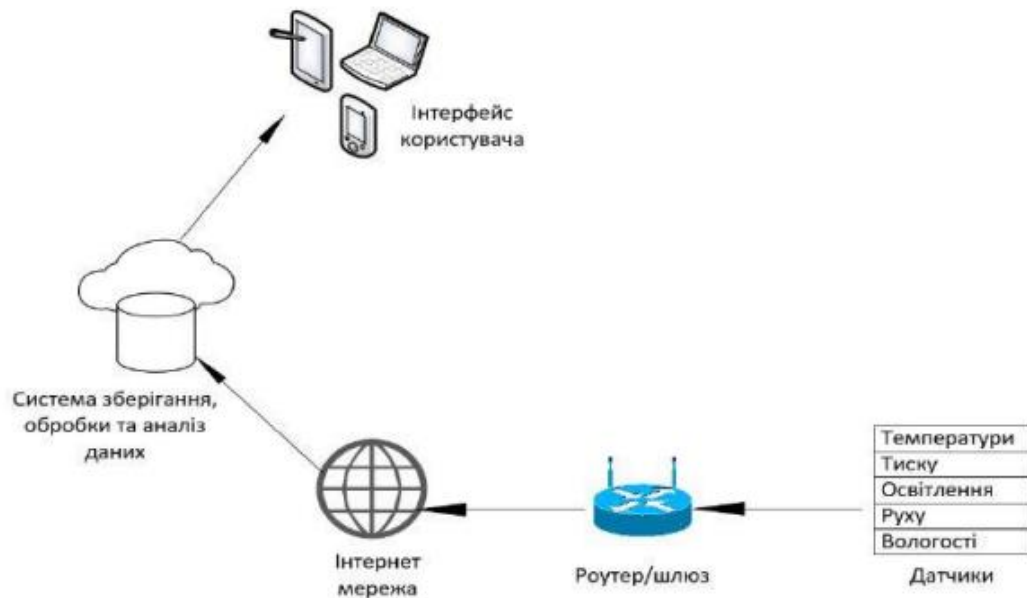


Рис. 1.1 Принцип роботи Інтернету речей

ІоТ-системи функціонують за принципом збору, передавання, обробки та представлення даних. Кожен з цих етапів відіграє важливу роль у забезпеченні ефективної роботи ІоТ-рішень.

На етапі зчитування інформації за допомогою датчиків ІоТ-пристрої збирають дані із зовнішнього середовища за допомогою різних сенсорів.

Датчики вимірюють фізичні параметри, такі як температура, вологість, рівень освітленості, тиск, рух, рівень газів тощо. Активатори (електронні пристрої, що реагують на дані) можуть виконувати дії на основі цих показників, наприклад, відкривати клапани, вмикати освітлення або регулювати температуру. Датчики можуть бути підключені до мікроконтролерів (Arduino, ESP8266, Raspberry Pi), які збирають та попередньо обробляють інформацію. Наприклад, у “розумному

будинку” датчики руху визначають, коли в кімнаті є люди, і вмикають світло, у сільському господарстві вологоміри контролюють рівень вологи у ґрунті, щоб система автоматично зрошувала рослини.

Після збору даних вони передаються для подальшої обробки. IoT-пристрої передають дані через дротові (Ethernet) або бездротові мережі (Wi-Fi, Bluetooth, Zigbee, LoRa, 5G). Дані надходять до локального шлюзу (router/gateway), який може попередньо фільтрувати інформацію. Далі дані відправляються у хмарне середовище (AWS IoT, Google Cloud, Microsoft Azure) або на локальний сервер для зберігання та обробки.

На етапі обробки даних, які отримані за допомогою датчиків, відбувається аналіз та інтерпретація отриманих даних. Зібрані дані проходять попередню фільтрацію та очищення (усунення шумів, перевірка коректності). Дані аналізуються за допомогою алгоритмів машинного навчання (AI/ML) для виявлення трендів та прогнозування. У разі виявлення відхилень або загроз система може автоматично надсилати оповіщення або виконувати дії.

Оброблена інформація відображається на інтерфейсі користувача, де люди можуть переглядати дані та керувати пристроями. Дані передаються на мобільні додатки, веб-інтерфейси або панелі керування. Користувач може отримувати повідомлення, графіки, аналітику та керувати пристроями дистанційно. У разі критичних ситуацій (наприклад, витоку газу або перевищення температури) система може автоматично надсилати тривожні повідомлення або запускати аварійні протоколи [1, 4].

Інтернет речей працює за циклічним принципом, де дані постійно збираються, передаються, обробляються та надаються користувачу у зрозумілому вигляді. Система IoT складається з датчиків, які взаємодіють із хмарою за допомогою спеціального типу зв'язку. Після того, як дані передаються у хмару, програмне забезпечення обробляє їх, а потім забезпечує виконання певних дій, наприклад, надіслати попередження або автоматично регулювати роботу датчика без потреби користувача.

1.3 Характеристика сенсорних пристроїв

Інтернет речей починається з джерел даних та виконуючих пристроїв – це датчики, а дані, які вони передають, утворюють потоки даних, які розподілені в часі. Для цих потоків необхідно забезпечити можливість передавання інформації, аналіз та зберігання інформації. Кількість датчиків та виконуючих пристроїв, які об'єднані в єдину мережу, значно зростає, тому важливо розуміти, яким чином вони взаємодіють. Датчики є основними елементами IoT-систем, оскільки вони виконують функцію збору інформації про навколишнє середовище та стан пристроїв. Вони є першою ланкою в IoT-інфраструктурі, забезпечуючи зв'язок між фізичним світом та цифровими технологіями.

Датчики дозволяють пристроям взаємодіяти з реальним світом. Завдяки їм стає можливою автоматизація, моніторинг та прогнозування у різних сферах - від «розумних будинків» до промисловості та медицини.

Основні функції датчиків IoT:

- збір даних - вимірювання фізичних параметрів (температура, вологість, освітлення, рух, тиск, рівень газів тощо);
- передавання інформації - надсилання зібраних даних через бездротові або дротові канали (Wi-Fi, Bluetooth, Zigbee, LoRa, NB-IoT);
- аналіз та взаємодія - використання даних для прийняття рішень, автоматизації процесів або активації інших IoT-пристроїв;
- моніторинг і контроль - безперервне спостереження за параметрами системи та реагування на зміни у режимі реального часу;
- забезпечення автономної роботи - разом із мікроконтролерами та AI/ML алгоритмами дозволяють автоматично регулювати процеси без втручання людини.

Важливим є розуміння, який тип датчика або кінцевого пристрою потрібно використовувати для вирішення задачі.

Датчики температури є найбільш розповсюдженими типами датчиків. Термопара – пристрій для вимірювання температури, який не потребує джерела живлення, тому що він сам генерує сигнал малої амплітуди (зазвичай

мікровольти). Це два провідники, які виготовлені з двох різних матеріалів, що з'єднані у точці вимірювання температури. На рис. 1.2 приведена блок-схема вимірювання температури за допомогою термопари.

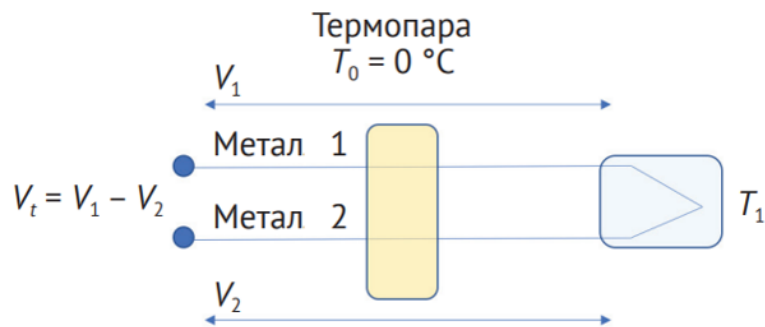


Рис. 1.2 Блок-схема вимірювання температури за допомогою термопари

Так як термопари часто використовуються у промислових умовах, високотемпературні середовища протягом часу можуть погіршувати точність датчиків, тому IoT-рішення повинні враховувати зміни, які відбуваються з датчиками у процесі їх експлуатації.

Резистивні датчики температури (Resistance Temperature Detectors - RTD) працюють у вузькому діапазоні температур (нижче 600°C), але дозволяють виконувати вимірювання з більшою точністю, ніж термопари (рис. 1.3).

RTD рідко використовуються у діапазоні вище $600\text{ }^{\circ}\text{C}$, що обмежує їх застосування у промисловості. При високих температурах платина може забруднюватися, що призводить до помилкових показників, але при вимірюваннях у межах заданого діапазону, RTD демонструють досить точні та стабільні результати.

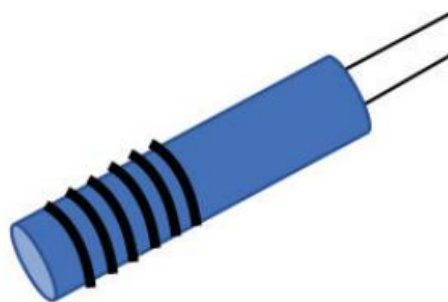


Рис. 1.3 Резистивний датчик температури

Термістор – датчик температури, електричний опір якого залежить від його температури. Цей вид датчиків забезпечує більш високу точність вимірювань порівняно з RTD. Є 2 види термісторів: NTC (їх опір зменшується при збільшенні температури) та PTC (їх опір збільшується з підвищенням температури). Основна відмінність від RTD полягає в тому, що термістори виготовляються з кераміки або полімерів, тоді як основою RTD завжди є метал.

Термістори знаходять застосування у медичному та науковому обладнанні, у харчовій промисловості, інкубаторах та в термостатах.

Вибір датчика залежить від умов експлуатації, точності та температурного діапазону, необхідного для задачі. Термопари підходять для високих температур, швидкі та міцні, але мають низьку точність. RTD - найточніші, але дорожчі та чутливі до механічних впливів. Термістори - чутливі та дешеві, але працюють лише у вузькому температурному діапазоні. Порівняння датчиків температури: термопари, RTD та термісторів приведене у табл. 1.1.

Таблиця 1.1

Порівняння датчиків температури

№ п/п	Характеристика	Термопара	Резистивний датчик температури (RTD)	Термістор
1	Принцип роботи	Генерація електрорушійної сили (ЕРС) при зміні температури	Зміна електричного опору металевого провідника при зміні температури	Зміна опору напівпровідникового матеріалу при зміні температури
2	Діапазон температур (°C)	-200 ... +1800 (залежно від типу)	-200 ... +850	-50 ... +150
3	Точність вимірювань	Середня ($\pm 1 \dots \pm 5$ °C)	Висока ($\pm 0,05 \dots \pm 0,5$ °C)	Висока, але нелінійна ($\pm 0,1 \dots \pm 1$ °C)
4	Лінійність залежності	Нелінійна	Лінійна	Дуже нелінійна
5	Час реакції	Швидкий (мілісекунди)	Середній (секунди)	Швидкий (мілісекунди)

6	Стійкість до впливу середовища	Висока (витримує екстремальні умови)	Середня (чутливий до механічних впливів)	Середня (чутливий до вологості)
7	Живлення	Не потребує живлення	Потребує живлення	Потребує живлення
8	Вартість	Низька (простий дизайн)	Висока (складні матеріали)	Низька (залежить від типу)
9	Типові сфери застосування	Високотемпературні процеси, промисловість, двигуни, печі	Лабораторії, медицина, авіація, високоточні вимірювання	Побутові пристрої, HVAC, електроніка

Датчик Холла – це смужка металу, через яку пропущений електричний струм. Ефект Холла застосовується у датчиках струму, для вимірювання змінного і постійного струму (рис. 1.4). Є 2 типи таких датчиків: із замкнутим та з розімкнутим контуром. Датчики із замкнутим контуром більше коштують, їх часто використовують у схемах із живленням від батарей.

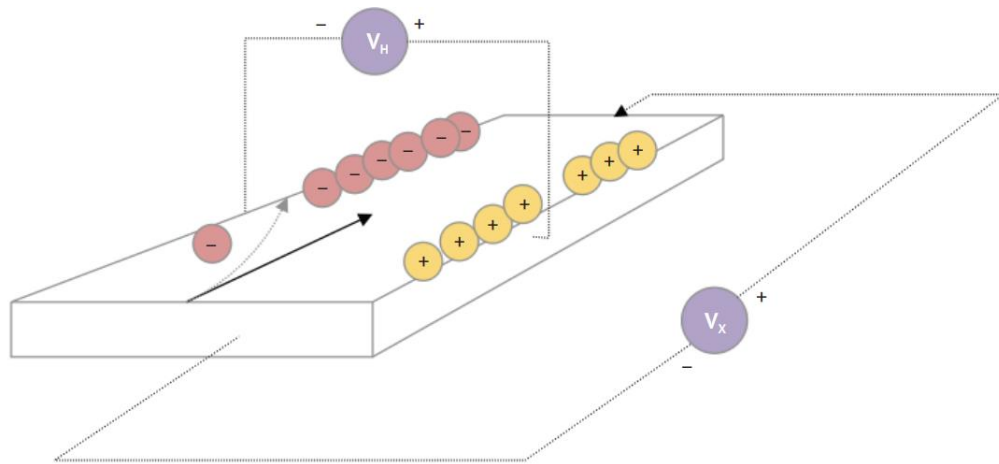


Рис. 1.4 Використання ефекту Холла

Застосування датчиків Холла: датчики положення, магнітometri, високонадійні перемикачі та вказівники рівня води. Використовуються також для вимірювання швидкості обертання різних вузлів та механізмів. Крім цього, ці датчики недорогі у виготовленні, невимогливі до умов експлуатації й можуть працювати у сурових умовах.

Фотоелектричні датчики – це датчики для виявлення світла або визначення його інтенсивності. Ці пристрої необхідні у системах безпеки, інтелектуальних комутаторах або в системах управління вуличним освітленням. Є 2 види датчиків такого типу: фоторезистор та фотодіод. Фоторезистор змінює опір в залежності від інтенсивності світла, а фотодіод перетворює світло в електричний струм.

Піроелектричний інфрачервоний (Pyroelectric Infrared, PIR) датчик складається з двох слотів, заповнених матеріалом, який реагує на інфрачервоне випромінювання та тепло. Це теплові датчики руху систем безпеки.

Датчики світлового виявлення та вимірювання (Light Detecting and Ranging, LiDAR) – тип датчика, який вимірює відстань до цілі шляхом вимірювання відображених від об'єкту лазерних імпульсів. Якщо датчик PIR лише виявляє рух в межах свого діапазону, LiDAR здатний вимірювати кількісні характеристики руху. На теперішній час широко використовується у сільському господарстві, автоматизованих транспортних засобах та транспортних засобах, що рухаються самі, робототехніці, при спостереженнях та дослідженнях оточуючого середовища. Цей вид активних вимірювальних пристроїв здатний аналізувати об'єкти будь-якого виду. Вони використовуються для аналізу газів, атмосфери, хмарних утворень та композицій, частин, швидкості об'єктів, що рухаються. Системи LiDAR здатні сканувати дуже великі простори і можуть працювати навіть із супутників (рис. 1.5).

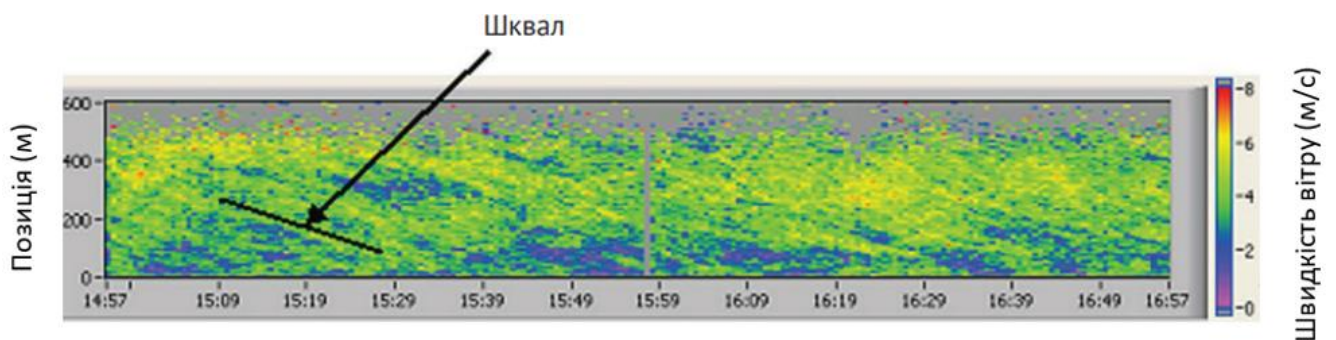


Рис. 1.5 Зображення, отримане за допомогою LiDAR

Акселерометри та гіроскопи широко застосовуються у різних мобільних пристроях для позиціонування та відслідковування руху, наприклад, у крокомірах та фітнес-трекерах.

Основні причини необхідності датчиків у IoT: моніторинг і контроль навколишнього середовища - датчики дозволяють вимірювати параметри (температуру, вологість, рівень CO₂, освітленість тощо) у «розумних містах», сільському господарстві, промисловості та екологічному моніторингу; автоматизація та оптимізація процесів - у «розумних будинках», на виробництві та в логістиці датчики допомагають зменшити витрати енергії, попередити аварії та підвищити ефективність; забезпечення безпеки - датчики руху, диму, газу дозволяють створювати «розумні системи» безпеки та захисту від загроз; інтерактивність і персоналізація - у «розумній електроніці» та переносних пристроях (фітнес-трекерах, «розумних годинниках») датчики дозволяють отримувати персоналізовані дані про здоров'я, активність та стан користувача.

Важливим аспектом необхідності застосування датчиків у сфері IoT є збір і аналіз великих даних (Big Data), адже сенсори IoT генерують великі обсяги даних, що використовуються для прогнозування, машинного навчання (AI/ML) та прийняття рішень у реальному часі.

1.4 Характеристика IoT-технологій

IoT-системи потребують ефективних технологій зв'язку для передавання даних від датчиків до серверів або хмарних платформ. Вибір технології залежить від дальності зв'язку, енергоспоживання, швидкості передавання даних та вартості.

Основні технології зв'язку в IoT:

- короткодіапазонні бездротові технології;
- мобільні мережі;
- LPWAN (Low-Power Wide-Area Network) - енергоефективні мережі;
- дротові технології зв'язку.

Вибір технології передавання даних в IoT залежить від:

- дальності зв'язку - для малих відстаней підходять Wi-Fi, Zigbee, Bluetooth, для великих - LoRaWAN, 4G/5G;
- енергоспоживання - для автономних пристроїв доцільно використовувати LPWAN (LoRa, NB-IoT, Sigfox);

- швидкості передавання - якщо потрібна висока швидкість, доцільно обирати Wi-Fi, 5G, Ethernet.

Короткодіапазонні бездротові технології використовуються для зв'язку між пристроями на малій відстані (до 100 м) (табл. 1.2).

Таблиця 1.2

Короткодіапазонні бездротові технології

№ п/п	Технологія	Діапазон	Швидкість передавання	Особливості
1	Wi-Fi (802.11)	50-100 м	11 Мбіт/с – 9,6 Гбіт/с	Висока швидкість, високе енергоспоживання
2	Bluetooth (BLE)	10-50 м	1-3 Мбіт/с	Низьке енергоспоживання, підходить для переносних пристроїв
3	Zigbee (802.15.4)	10-100 м	250 кбіт/с	Енергоефективна технологія для «розумного будинку»
4	Z-Wave	30-100 м	9,6-100 кбіт/с	Використовується в автоматизації будинків

Приклади застосування: Wi-Fi – “розумні камери”, домашні IoT-пристрої; Bluetooth - фітнес-трекери, “розумні годинники”; Zigbee/Z-Wave – “розумне освітлення”, датчики руху.

Мобільні мережі використовуються для передавання даних на великі відстані (до кількох км) через мобільні оператори. Технології мобільних мереж для використання в системах IoT приведені у табл. 1.3.

Таблиця 1.3

Технології мобільних мереж зв'язку для IoT

№ п/п	Технологія	Діапазон	Швидкість передавання	Особливості
1	2G (GSM, GPRS, EDGE)	до 10 км	10-100 кбіт/с	Використовується для M2M-комунікацій
2	3G (UMTS, HSPA)	до 10 км	200 кбіт/с – 42 Мбіт/с	Помірна швидкість, витісняється 4G
3	4G LTE	до 10 км	1-100 Мбіт/с	Висока швидкість, підходить для відеонагляду
4	5G	до 10 км	1-10 Гбіт/с	Мінімальна затримка, придатне для розумних міст

Приклади застосування: 2G/3G - системи моніторингу та телеметрії; 4G/5G - автомобілі, відеоспостереження, промисловий IoT.

Мережі LPWAN розроблені для пристроїв із низьким енергоспоживанням, що працюють у віддалених місцях (наприклад, “розумне сільське господарство”) (табл. 1.4).

Таблиця 1.4

Технології енергофактивних мереж LPWAN

№ п/п	Технологія	Діапазон	Швидкість передавання	Особливості
1	LoRaWAN	до 15 км	0,3-50 кбіт/с	Дуже низьке енергоспоживання
2	NB-IoT	до 10 км	20-250 кбіт/с	Підтримується мобільними операторами
3	Sigfox	до 50 км	100 біт/с – 1 кбіт/с	Низька вартість, підходить для датчиків

Прикладами застосування є: LoRaWAN - смарт-лічильники, екологічний моніторинг; NB-IoT – «розумні парковки», контроль водопостачання; Sigfox - моніторинг логістики та відстеження активів.

Проводові технології зв'язку використовуються там, де необхідна висока надійність та мінімальні затримки (табл. 1.5).

Таблиця 1.5

Проводові технології зв'язку

№ п/п	Технологія	Діапазон	Швидкість передавання	Особливості
1	Ethernet (LAN, PoE)	до 100 м	100 Мбіт/с – 10 Гбіт/с	Висока швидкість, стабільність
2	Modbus (RS-485, RS-232)	до 1200 м	1,2-115 кбіт/с	Використовується у промисловості
3	CAN Bus	до 40 м	1 Мбіт/с	Автомобільні системи, транспорт

Приклади застосування: Ethernet - серверні IoT-рішення; Modbus - промислові датчики та контролери; CAN Bus - автомобільні та транспортні системи.

IoT-комунікації об'єднують різні технології для створення ефективних та енергозберігаючих рішень у промисловості, розумних містах та побуті.

2 СИСТЕМА ОБРОБКИ ДАНИХ ТА ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ ІОТ-ПРИСТРОЇВ

2.1 Визначення елементів системи

Система обробки даних та прогнозування енергоспоживання ІоТ-пристроїв призначена для збору, аналізу та прогнозування енергоспоживання ІоТ-пристроїв, що дозволяє ефективно керувати використанням електроенергії та оптимізувати витрати. Вона базується на технологіях Інтернету речей (ІоТ), великих даних (Big Data) та штучного інтелекту (AI/ML).

Основними елементами системи є: датчики та ІоТ-пристрої, шлюзи передавання даних, хмарне або локальне сховище даних, аналітичний модуль (Big Data + AI/ML), інтерфейс користувача (UI/UX).

Датчики збирають інформацію про енергоспоживання ІоТ-пристроїв. Шлюз передає ці дані до хмарного або локального сервера. Аналітичний модуль обробляє отриману інформацію та прогнозує майбутнє енергоспоживання. Інтерфейс користувача відображає результати та дозволяє контролювати пристрої.

Датчики та ІоТ-пристрої зчитують параметри електроспоживання та інші фактори, що впливають на витрати енергії. Прикладами можуть бути: «розумні лічильники» електроенергії, які вимірюють напругу, струм, споживану потужність; датчики температури та вологості, які враховують зовнішні умови, що впливають на енергоспоживання; «розумні розетки» та реле, які відстежують роботу підключених пристроїв та дозволяють їх вимикати дистанційно.

Шлюз передавання даних забезпечує зв'язок між датчиками та серверною частиною системи. Технологіями передавання даних є: Wi-Fi, Zigbee, LoRaWAN, NB-IoT - для бездротового передавання даних від ІоТ-пристроїв. Розповсюдженими протоколами обміну даними між пристроями та сервером є наступні: MQTT, HTTP, CoAP.

Хмарне або локальне сховище даних зберігає великі обсяги даних про споживання електроенергії. Хмарні платформи, такі як AWS IoT, Google Cloud,

Microsoft Azure зберігають дані, обробляють їх та візуалізують. Локальні сервери або бази даних (MySQL, InfluxDB, MongoDB) використовуються у випадках, коли хмарні рішення не підходять.

Аналітичний модуль (Big Data + AI/ML) використовує алгоритми штучного інтелекту (AI) та машинного навчання (ML) для аналізу та прогнозування споживання електроенергії. До методів аналізу відносяться: машинне навчання (регресійні моделі, нейронні мережі, методи глибокого навчання), обробка великих даних (Big Data), прогнозування. Обробка Big Data включає: аналіз історичних даних, виявлення закономірностей, прогнозування пікових навантажень. Прогнозування на основі часових рядів визначає майбутнє енергоспоживання та можливі перевантаження.

Інтерфейс користувача (UI/UX) відображає аналітику енергоспоживання, дозволяє управляти пристроями. Можливі варіанти:

- веб-додатки (React, Angular, Vue.js) - панелі керування для моніторингу та аналізу даних;
- мобільні додатки (Android/iOS) - дають змогу користувачам переглядати дані та керувати IoT-пристроями дистанційно;
- графіки та звіти - інтерактивна візуалізація споживання енергії (наприклад, Power BI, Grafana, Tableau).

Система обробки даних та прогнозування енергоспоживання IoT-пристроїв дозволяє оптимізувати використання електроенергії, знижувати витрати та підвищувати ефективність роботи IoT-систем. Вона є ключовим елементом у «розумних будинках», «розумних містах», промисловості 4.0 та енергетичних системах нового покоління.

2.2 Технології передавання даних

Основними функціями технологій передавання даних у IoT є:

- забезпечення зв'язку між пристроями;
- передавання інформації на малі та великі відстані;

- оптимізація енергоспоживання;
- підвищення безпеки та надійності;
- масштабованість та адаптивність.

IoT складається з великої кількості датчиків, контролерів та кінцевих пристроїв. Технології зв'язку дозволяють їм обмінюватися даними в режимі реального часу або за певними алгоритмами. Для локальних IoT-систем («розумні будинки», переносні пристрої) використовуються Bluetooth, Wi-Fi, Zigbee. Для масштабних розгортань («розумні міста», промислові системи) необхідні LPWAN (LoRaWAN, NB-IoT), 4G/5G.

Деякі IoT-пристрої (наприклад, «розумні лічильники», екологічні датчики) повинні працювати роками без підзарядки. Технології на кшталт LoRaWAN, NB-IoT, Sigfox дозволяють передавати дані енергоефективно. Технології зв'язку в IoT передбачають шифрування та захист даних під час передавання (TLS, AES, VPN). Це критично важливо для фінансових операцій, систем відеоспостереження та медичних IoT-рішень.

Різні технології дозволяють масштабувати IoT-системи відповідно до потреб бізнесу. Наприклад, Wi-Fi підходить для локальних рішень, тоді як 5G та LPWAN - для глобальних проєктів.

Wi-Fi 6 - це маркетингова назва стандарту 802.11ax, який був представлений Wi-Fi Alliance у 2018 році. Його попередник Wi-Fi 5 (802.11ac) був запущений у 2013 році. Так історично склалось, що людям легше асоціювати рівень/якість технологій за простими назвами. Наприклад, всі знають, що 4G краще за 3G, мало хто розбирається у технічних деталях. Так само склалося і з Wi-Fi.

Одне з покращень технології - перехід до більш щільної модуляції — 1024-QAM (квадратно-амплітудна модуляція). Як цей параметр змінювався в технології Wi-Fi представлена на рис. 2.1.

802.11n (Wi-Fi 4)	64-QAM
802.11ac (Wi-Fi 5)	256-QAM
802.11ax (Wi-Fi 6)	1024-QAM

PHY	Bandwidth (as number of data subcarriers)	Data bits per subcarrier	Time per OFDM symbol (800ns GI)	1 SS	3 SS	4 SS	8 SS
802.11ac	234 (80 MHz)	$5/6 \times \log_2(256)$ ≈ 6.67	4 μs	390 Mbps	1.17 Gbps	1.56 Gbps	-
	2 × 234 (160 MHz)	X	/	= 780 Mbps	-	3.12 Gbps	-
802.11ax	980 (80 MHz)	$5/6 \times \log_2(1024)$ ≈ 8.33	13.6 μs	600 Mbps	1.8 Gbps	2.4 Gbps	4.8 Gbps
	2 × 980 (160 MHz)			1.2 Gbps	3.6 Gbps	4.8 Gbps	-

Рис. 2.1 Обчислення швидкості 802.11ac та 802.11ax

Крім того, Wi-Fi 6 використовує технологію OFDMA (ортогональний частотний множинний доступ), яка також застосовується в мобільному зв'язку LTE (4G). Завдяки цьому методу точка доступу може одночасно передавати дані кільком клієнтам, розподіляючи для кожного окрему частоту. Це забезпечує більш ефективне використання доступного спектра та часу, що підвищує продуктивність передавання даних. Зокрема, точка доступу може гнучко регулювати обсяг спектру, що використовується, оптимізуючи швидкість передавання відповідно до потреб кожного клієнта.

В IoT є відомі протоколи та технології, для рівня квартири/будинку: Z-Wave, ZigBee, BLE; для рівня району/міста: Weightless, LoRaWAN. На відміну від попередніх технологій, Wi-Fi використовується із значно меншою кількістю датчиків або в тих місцях, де можна обійтися без автономної роботи датчиків (заводи, склади, ферми). Це обумовлено тим, що наявність Wi-Fi модулю значно зменшує час роботи автономного джерела живлення датчика і в більшості випадків потребує зовнішнього джерела живлення. Також варто враховувати, що процес розводки кабелів живлення тягне за собою фінансові та часові витрати.

Вендори також виходили на цей конкурентний ринок і випускали точки доступу з підтримкою не тільки Wi-Fi, а і Zigbee, BLE та Thread. Завдяки економному використанню енергії період експлуатації деяких пристроїв (датчиків) може коливатися від року до п'яти, в залежності від частоти передавання та обсягу даних та кліматичних умов.

Стандарт 802.11ah також надає можливість значної економії електроенергії у підключеного пристрою. Замість того, щоб радіо IoT-пристрою перебувало весь час для прослуховування пакету, через точку доступу девелопер може встановити для клієнта параметр Target-Wakeup Time (максимальний час - до 5 років). Таким чином, IoT-пристрій може перевести радіомодуль у сплячий режим до цього часу.

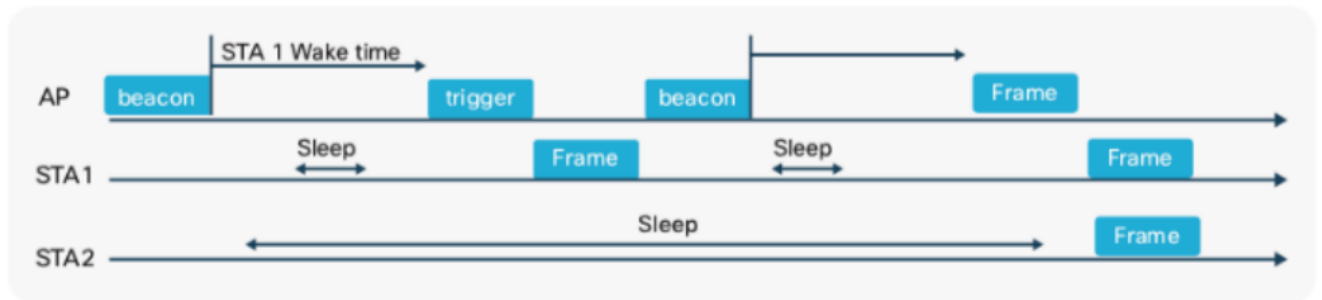


Рис. 2.2 STA-IoT пристрій/станція

Таким чином, Wi-Fi 6 робить серйозну заявку на розширення своєї частки в IoT-мережах. Зокрема є потенціал розширення шляхом використання інфраструктури, яка вже працює, - комутаційного обладнання, мережі PoE. Проникнення Wi-Fi інфраструктури в цільових ринках (підприємства/організації, заводи, торговельні/офісні центри, HoReCa, житлові комплекси) значно більше, ніж інших IoT-технологій. Зараз перевага на боці вендорів та виробників датчиків. Якщо вони швидко підхоплять цей тренд і почнуть випускати автономні датчики з Wi-Fi модулями, то ми скоро зможемо побачити нові стартапи в цій сфері. Розробникам, у свою чергу, відкриваються можливості із створення застосунків для IoT з використанням популярної технології та управління параметром Target-Wakeup Time.

Пристрій Wi-Fi 6 повинен підтримувати попередні режими Wi-Fi. Точка доступу Wi-Fi 6 також може зв'язуватися з попередньою версією пристроїв Wi-Fi. Тому Wi-Fi 6 може співіснувати з попередніми версіями та пристроями, що їх підтримують. Але попередні версії пристроїв не можуть скористатися перевагами Wi-Fi 6. Наразі виробники почали працювати над підтримкою нового стандарту у кінцевих пристроях.

Wi-Fi 6 - програмований, а це означає, що відкривається цілий новий світ застосунків та бізнес-можливостей. Ці зміни відбудуться швидко, тому розробникам, мережним та IT-провайдерам потрібно зараз розпочати підготовку до нових можливостей та викликів. Зокрема у сфері IoT розробникам треба бути готовими до роботи та взаємодії з датчиками та сенсорами, використовуючи стандарт 802.11ax, і враховувати, в тому числі, можливість управління параметрами Target-Wake Time. Новий стандарт значно покращує ефективність технології Wi-Fi, зокрема через збільшення швидкості, зменшення затримки та зростання кількості клієнтів, з якими може взаємодіяти точка доступу.

LPWAN (Low-Power Wide-Area Network) - технологія енергоефективних мереж дальнього радіусу дії.

LoRa (що означає Long Range) - це запатентована технологія бездротового зв'язку, яка поєднує в собі наднизьке споживання енергії і ефективний дальній зв'язок. При цьому потрібно враховувати, що дальність дії сильно залежить від навколишнього середовища і можливих перешкод (LOS або N-LOS).

LoRa зазвичай має діапазон від 5 до 15 км і один шлюз LoRa може забезпечувати покриття для всього міста. Технологія була розроблена Cycleo у Франції і вийшла на перший план, коли компанія була придбана Semtech у 2012 році. У Semtech використовували модулі LoRa з Arduino і Raspberry Pi і результат роботи виявився достатньо хорошим.

LoRa – технологія і метод модуляції.

LoRaWAN – відкритий протокол для мереж.

LPWAN – енергоефективна мережа дальнього радіусу дії.

Модуляція LoRa відповідає за фізичний рівень, LoRaWAN є протоколом (MAC-рівень) для мереж з високою ємністю, великим радіусом дії та низьким енергоспоживанням, розробленим організацією LoRa Alliance для мереж LPWAN.

У результаті досліджень були визначені переваги та недоліки технології LPWAN. Перевагами LPWAN є: низьке енергоспоживання, великий радіус дії, можливість підключення великої кількості пристроїв, низька вартість експлуатації, гарна проникаюча здатність сигналу.

LPWAN-пристрої можуть працювати роками на одній батареї, що робить їх ідеальними для віддалених або автономних рішень. Діапазон покриття LPWAN досягає 10-50 км у сільській місцевості та 1-10 км у міських умовах, що значно перевищує традиційні бездротові технології, такі як Wi-Fi або Bluetooth. LPWAN підтримує тисячі підключених пристроїв в одній мережі, що робить її ефективною для масштабних IoT-рішень (наприклад, смарт-лічильники, системи моніторингу довкілля). Використання LPWAN зменшує витрати на обслуговування пристроїв, адже їх не потрібно часто заряджати або замінювати батареї. Також, для деяких протоколів (LoRaWAN, Sigfox) відсутні щомісячні плати за мобільний зв'язок. LPWAN може передавати сигнал через стіни, підвали та інші перешкоди, що робить її ефективною для підключення датчиків у важкодоступних місцях.

Недоліками LPWAN є: низька швидкість передавання даних, обмежений розмір пакета даних, висока затримка, обмежена двостороння комунікація, залежність від провайдерів (для деяких технологій). NB-IoT і LTE-M працюють через мобільних операторів, що означає необхідність укладати контракти та залежність від покриття оператора.

LPWAN - ідеальне рішення для IoT-додатків, де потрібне дальнє передавання даних, низьке енергоспоживання і невеликий обсяг трафіку. Вона підходить для «розумних лічильників», агротехнологій, логістики та екологічного моніторингу.

NB-IoT (Narrowband Internet of Things) - технологія зв'язку для IoT-пристроїв, що працює у стільникових мережах (LTE) і оптимізована для передавання невеликих обсягів даних на великі відстані з низьким енергоспоживанням (рис. 2.3).

NB-IoT працює на ліцензованих частотах, що використовуються операторами мобільного зв'язку (LTE, 4G). Це забезпечує стабільність, безпеку та широкий радіус покриття. Технологія має кращу проникаючу здатність, ніж традиційні мобільні мережі, тому може передавати дані навіть із підвалів, тунелів, підземних стоянок або віддалених сільських районів. NB-IoT підтримує режими PSM (Power Saving Mode) і eDRX (Extended Discontinuous Reception), що дозволяють IoT-пристроєм працювати до 10 років на одній батареї.

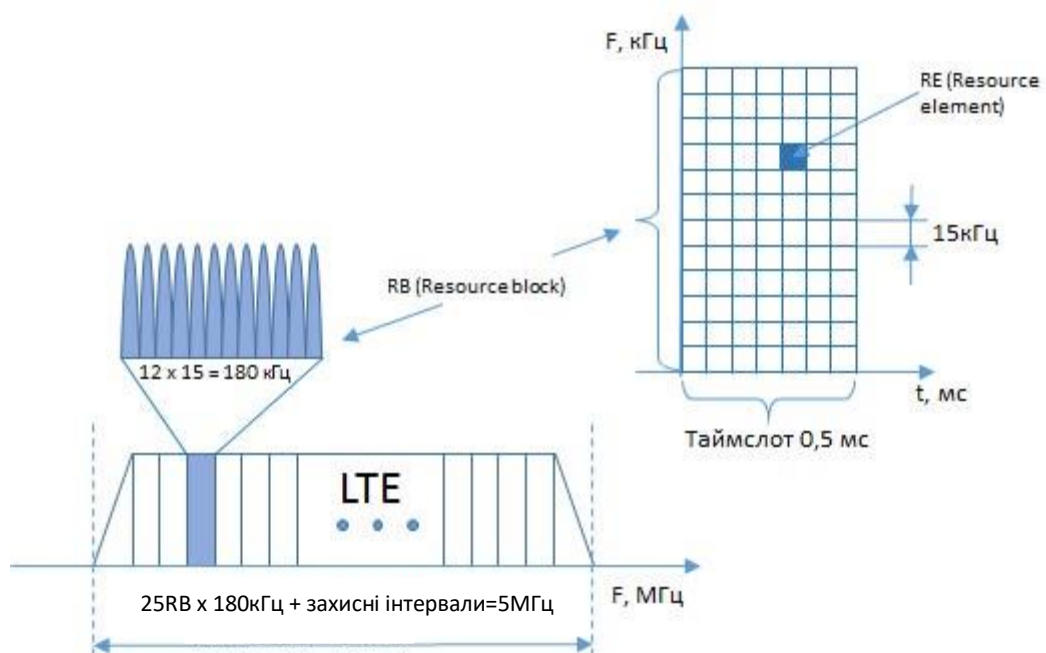


Рис. 2.3 Принцип роботи технології NB-IoT

Максимальна швидкість - до 250 кбіт/с, що підходить для сенсорів, розумних лічильників, GPS-трекерів, але не для відеоспостереження чи великих обсягів даних. NB-IoT підтримує тисячі пристроїв на одній базовій станції, що робить її ідеальною для масштабних IoT-мереж (розумні міста, екологічний моніторинг, сільське господарство). NB-IoT підтримує обмежений двосторонній зв'язок, що дозволяє не лише передавати дані від датчиків, а й отримувати керуючі команди (наприклад, вмикати або вимикати пристрій). Завдяки простоті реалізації у мобільних мережах та дешевим модулям NB-IoT є економічно вигідним рішенням для IoT-додатків. NB-IoT - це ефективна технологія для IoT-пристроїв, які потребують далекого радіусу дії, низького енергоспоживання та роботи у мобільних мережах.

2.3 Протоколи обміну даними

Протоколи в IoT забезпечують обмін даними між пристроями, шлюзами, серверами та користувачами. Без них елементи системи не змогли б “розуміти” одне одного. Протокол IP - це набір правил, що визначають спосіб відправлення

даних в Інтернет. Протоколи Інтернету речей гарантують, що інформація з одного пристрою або датчика буде зчитуватися і правильно інтерпретуватися іншим. Так як існує багато пристроїв Інтернету речей, важливо використовувати потрібний протокол у потрібному контексті.

Тип протоколу Інтернету речей, який використовується, залежить від рівня архітектури системи, на якому повинні передаватися дані. Модель OSI надає карту різних рівнів, на яких виконується відправлення і отримання даних. Кожний протокол в архітектурі системи Інтернету речей забезпечує взаємодію:

- між пристроями,
- між пристроєм і шлюзом,
- між шлюзом і центром обробки даних або між шлюзом і хмарою,
- обмін даними між центрами обробки даних.

Основні функції протоколів в IoT:

- передавання даних;
- сумісність між пристроями;
- забезпечення безпеки;
- оптимізація енергоспоживання;
- управління та контроль;
- передавання у реальному часі.

Протоколи відповідають за стандартизоване та надійне передавання інформації від датчиків до хмарного середовища або до інтерфейсу користувача. Завдяки єдиним протоколам різні виробники можуть створювати пристрої, що взаємодіють між собою. Наприклад, «розумні лампи», термостати та дверні замки можуть працювати в одній системі. Протоколи можуть включати механізми шифрування, автентифікації та захисту даних, що критично важливо для запобігання зломам та витокам інформації.

Деякі протоколи створені з урахуванням низького енергоспоживання, що дозволяє IoT-пристроєм працювати роками без заміни батареї. Протоколи також дозволяють керувати IoT-пристроєм: надсилати команди, оновлювати прошивку, змінювати налаштування. У критично важливих сферах (наприклад, медицина,

безпека) протоколи забезпечують мінімальну затримку для миттєвої реакції системи.

Розповсюдженими протоколами, які забезпечують обмін даними між пристроями, пристроєм і шлюзом, шлюзом і центром обробки даних або між шлюзом і хмарою, між центрами обробки даних є: IP, TCP, UDP, HTTP/HTTPS, TLS, DTLS, CoAP, DDS, XMPP, MQTT, STOMP, SOAP. Доцільно виконати дослідження шести останніх протоколів, так як вони є затребуваними та перспективними у сфері IoT станом на теперішній час. Для того, щоб дослідити вищевказані протоколи, доцільно представити топологію IoT у вигляді рис. 2.4, з якої видно, що зручним рішенням є розбиття даної топології на сегменти, на яких використовується той чи інший тип протоколу.

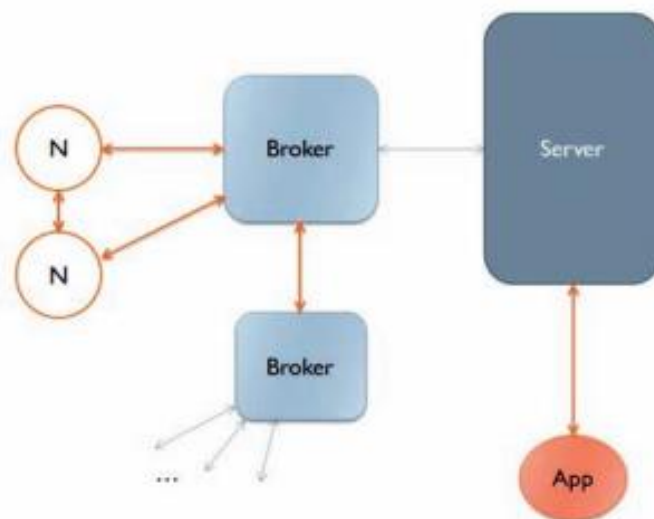


Рис. 2.4 Вихідна топологія

На ділянці між сенсорними вузлами функціонує протокол DDS (рис. 2.5). DDS (Data Distribution Service) - це високопродуктивний, масштабований протокол обміну даними в реальному часі, який активно використовується в системах IoT, особливо там, де важлива оперативність, надійність і передавання даних між великою кількістю пристроїв. DDS призначений для публікації/підписки (publish/subscribe) на дані між пристроями без потреби в централізованому сервері. Його головна мета - автоматично поширювати дані між усіма зацікавленими вузлами в мережі, які мають на них підписку.



Рис. 2.5 Ділянка між сенсорними вузлами

DDS дозволяє сенсорним вузлам безпосередньо обмінюватися даними один з одним, без необхідності звертатися до центрального хмарного сервера. Це знижує затримки та мережеве навантаження. Він підтримує низьку затримку та високу швидкість передавання, що робить його ідеальним для критичних систем (наприклад, автономного транспорту, промислового контролю, робототехніки). DDS забезпечує гарантовану доставку, впорядкування повідомлень та виявлення втрати даних, що особливо важливо при роботі з сенсорами в умовах промислового або розподіленого середовища.

Протокол автоматично знаходить інші пристрої, які “цікавляться” певним типом даних. Це зручно, коли кількість сенсорів змінюється або система масштабується. DDS дозволяє налаштовувати параметри передавання: наприклад, як часто оновлювати дані, наскільки вони критичні і чи потрібно дублювати їх, якщо вузол був тимчасово недоступний.

На ділянці від сенсорного вузла до брокера доцільно використовувати протоколи CoAP, XMPP (рис. 2.6).

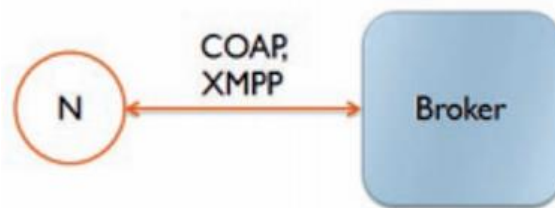


Рис. 2.6 Ділянка від сенсорного вузла до брокера

У системах Інтернету речей (IoT) ділянка "від сенсорного вузла до брокера" - це критичний етап, де необхідно економно, швидко й надійно передати дані від пристрою до системи, яка їх обробляє або перенаправляє далі. Саме тут CoAP та XMPP є ефективними рішеннями.

CoAP (Constrained Application Protocol) - легкий протокол, розроблений спеціально для IoT-пристроїв з обмеженими ресурсами. Побудований на основі UDP, підтримує модель запит-відповідь, подібну до HTTP. Підходить для пристроїв з низькою обчислювальною потужністю та обмеженим енергоживленням. Має вбудовану підтримку спостереження за ресурсами - вузол може надсилати оновлення лише при зміні даних.

XMPP (Extensible Messaging and Presence Protocol) - це протокол, що використовується для обміну повідомленнями у режимі реального часу. Його основа - формат XML. Підтримує асинхронний обмін повідомленнями. Підходить для двосторонньої комунікації - сенсор не лише надсилає дані, а й може отримувати команди.

Досліджуючи ділянку брокер – сервер, визначено, що протоколами на ній є MQTT, STOMP. Основна функція на цій ділянці - передавання, обробка та зберігання структурованих IoT-даних для подальшого аналізу, відображення або прийняття рішень.

MQTT (Message Queuing Telemetry Transport) працює за моделлю “публікація–підписка” (publish/subscribe) і спеціально розроблений для пристроїв з обмеженими ресурсами та нестабільним з’єднанням. MQTT забезпечує надійне передавання повідомлень від пристроїв (публікаторів) до інших пристроїв або систем (підписників) через брокер. MQTT має мінімальний розмір заголовка (всього 2 байти), що дозволяє ефективно передавати дані навіть в умовах низької пропускної здатності. Якщо пристрій неочікувано відключається, MQTT може автоматично надіслати останнє повідомлення, яке вказує на його стан - корисно для виявлення збоїв. Брокер може зберігати останнє повідомлення по темі, щоб нові підписники одразу отримували актуальні дані. Гнучкість і адаптивність даного протоколу до ресурсних обмежень роблять його одним із найпопулярніших стандартів у сфері Інтернету речей.

STOMP (Simple Text Oriented Messaging Protocol) - простий текстовий протокол для обміну повідомленнями між клієнтом і сервером через брокери повідомлень. Його ще називають "HTTP для обміну повідомленнями", адже він

використовує звичайний текстовий формат і легко читається. Він не залежить від конкретної мови програмування або середовища, тому його легко інтегрувати з багатьма системами.

Основне призначення STOMP - забезпечити простий, текстовий та кросплатформенний інтерфейс для клієнтів, які взаємодіють з брокерами повідомлень, такими як ActiveMQ, RabbitMQ, Apollo, HornetQ.

Ділянка “сервер — додаток (інтерфейс користувача)” є завершальним етапом обробки даних і відповідає за візуалізацію, доступ, аналітику і взаємодію користувача з усією системою. Основними задачами на цій ділянці є: передавання оброблених даних додатку, візуалізація та моніторинг, керування IoT-пристроями, реєстрація/аутентифікація користувачів, отримання сповіщень, інтеграція з іншими сервісами, користувач може аналізувати історичні дані - це важливо для прогнозування, аналітики та звітності.

Ділянка «сервер-додаток» є інтерфейсом між цифровим середовищем IoT і реальними діями користувача (рис. 2.7).

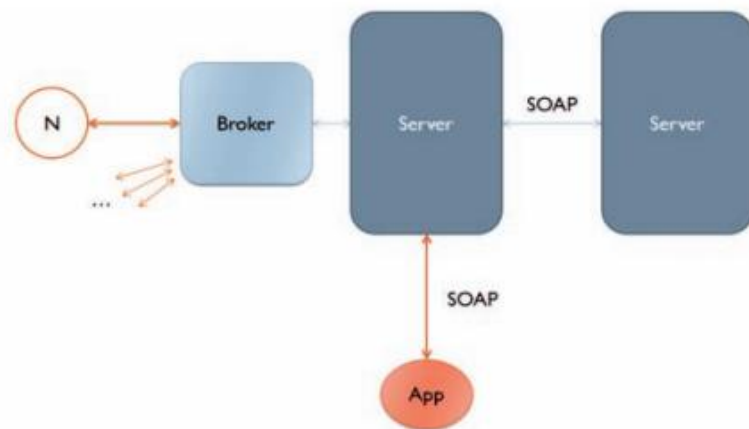


Рис. 2.7 Сервер – додаток

В якості висновку проведеного дослідження можна зазначити, що всі вищевказані протоколи мають різні можливості, різні архітектури та різні задачі. Особливості протоколів залежать від їхніх застосувань. В результаті до вибору для свого додатку оптимального протоколу слід вибирати обґрунтований підхід, враховуючи негативні та позитивні властивості та потреби кожного з протоколів.

З ІНТЕГРАЦІЯ ІoT-ТЕХНОЛОГІЙ ТА ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Концепція AIoT. Машинне навчання для IoT

AIoT (Artificial Intelligence of Things) - це концепція, яка поєднує Інтернет речей (IoT) та штучний інтелект (Artificial Intelligence, AI) з метою створення інтелектуальних, автономних і адаптивних систем. Це коли пристрої, підключені до Інтернету, не просто збирають і передають дані, а аналізують, "розуміють" і приймають рішення самостійно.

У попередніх розділах бакалаврської роботи були досліджені такі компоненти IoT, як: датчики, комунікаційні модулі, мережі та платформи обміну даними.

До компонентів штучного інтелекту відносяться: машинне навчання (Machine Learning, ML), глибоке навчання (DL), які забезпечують обробку великих обсягів даних, прогнозування, автоматичні рішення. IoT-системи реагують на події, але не можуть їх прогнозувати або інтерпретувати. AIoT замість пасивного збору даних визначає активне розуміння ситуації, автоматизовані інтелектуальні дії, що забезпечує прогнозування майбутніх подій, реальне прийняття рішень на основі даних, перетворюючи звичайні "розумні" пристрої на "самомислячі" системи. Завдяки поєднанню IoT і AI пристрої можуть аналізувати дані, навчатися з досвіду та діяти без участі людини, що відкриває нові горизонти в автоматизації та цифровій трансформації (рис. 3.1).

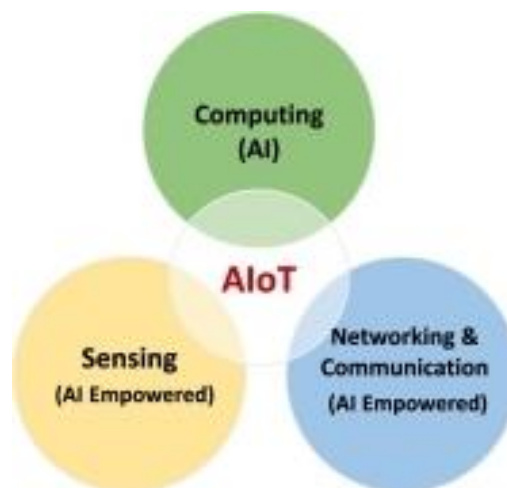


Рис. 3.1 Огляд AIoT

Машинне навчання є важливою частиною екосистеми Інтернету речей (IoT), оскільки допомагає обробляти великі обсяги даних, які збирають IoT-пристрої, для отримання корисної інформації та автоматизації процесів. Провівши дослідження, можна визначити основні напрямки застосування ML в IoT, а саме: аналіз даних та виявлення закономірностей; прогнозування; розпізнавання аномалій; створення автономних систем; створення персоналізованих рішень; забезпечення безпеки IoT-пристроїв; оптимізація енерговикористання (рис. 3.2).

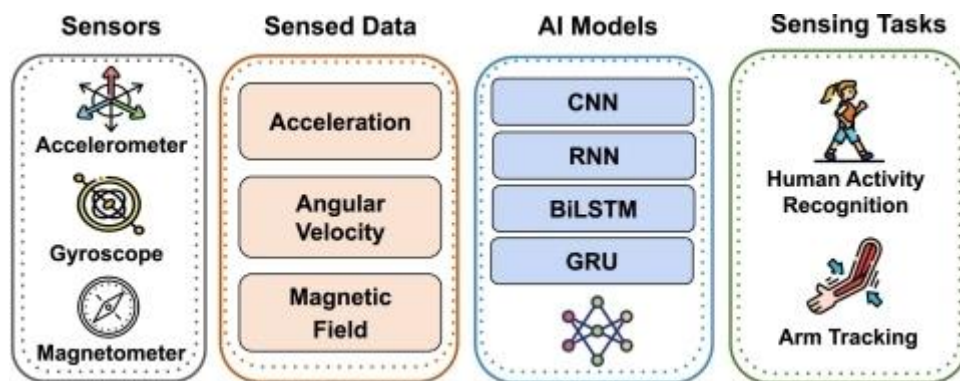


Рис. 3.2 Принцип дії датчика руху на основі штучного інтелекту

IoT-пристрої збирають дані в реальному часі і ML можна використовувати для виявлення прихованих закономірностей у цих даних, виявлення трендів, які допоможуть спрогнозувати події або оптимізувати процеси (наприклад, у смарт-містах або виробництві), наприклад, у системах "розумного будинку" ML може аналізувати дані з сенсорів і прогнозувати, коли потрібно увімкнути опалення або освітлення в залежності від звичок користувачів.

ML можна використовувати для створення моделей прогнозування на основі даних IoT, наприклад, на виробничих підприємствах сенсори на обладнанні разом із ML можуть прогнозувати необхідність технічного обслуговування.

Важливим аспектом є застосування машинного навчання в системі Інтернету речей для розпізнавання аномалій. ML моделі допомагають виявляти аномалії в даних, які можуть свідчити про несправності, збої або потенційні загрози безпеці. Наприклад, в енергомережах IoT-пристрої передають дані про енергоспоживання.

ML може виявляти аномалії, які свідчитимуть про несанкціоноване підключення або аварії.

IoT у поєднанні з ML дозволяє створювати системи, що працюють автономно. Такий підхід можна використовувати у галузі сільського господарства. У сільському господарстві дрони або поливні системи, керовані ML, дозволять оптимізувати використання ресурсів (води, добрив). У автомобілях використовується ML для систем автономного керування. Наприклад, смарт-трактори можуть адаптувати свою роботу залежно від стану ґрунту, отримуючи дані від сенсорів [3-5].

ML в IoT дозволяє створювати персоналізовані рішення для користувачів. “Розумні переносні пристрої” (наприклад, фітнес-браслети) використовують ML для аналізу даних про фізичну активність та здоров'я, щоб пропонувати індивідуальні рекомендації.

ML використовується для захисту IoT-пристроїв від кібератак: виявлення аномалій у поведінці пристроїв, аналізу мережевого трафіку для запобігання вторгненням, наприклад, фільтрація трафіку у “розумних будинках” для виявлення підозрілих дій.

Досліджуючи оптимізацію енерговикористання, слід зазначити, що IoT-пристрої часто мають обмежені ресурси, такі як батареї. ML оптимізує енергоспоживання, дозволяючи пристроям працювати довше. Наприклад, смарт-лічильники використовують ML для оптимального розподілу електроенергії між пристроями в будинку.

Поєднання IoT та ML значно розширить можливості автоматизації та інтелектуальної обробки даних, створюючи інноваційні рішення для різних галузей.

3.2 Аналіз ML-алгоритмів для IoT

В області Інтернету речей (IoT) доцільно використовувати різні алгоритми машинного навчання (ML) залежно від задачі, обмежень пристроїв та доступу до

обчислювальних ресурсів. Найбільш актуальними алгоритми для використання в IoT є:

1) Лінійна регресія / Поліноміальна регресія, які доцільно використовувати для прогнозування енергоспоживання, температури та вологості, так як цей алгоритм легкий у реалізації, добре працює на малих пристроях. IoT-приклад: прогнозування енергоспоживання будинку з «розумним лічильником».

2) Древа рішень (Decision Trees) та Random Forest застосовуються для класифікації станів пристрою, виявлення несправностей, наприклад, для розпізнавання типів споживання або відмов пристроїв. Цей алгоритм інтерпретований, добре підходить для невеликих даних.

3) Алгоритм К-ближчих сусідів (K-Nearest Neighbors, KNN) доцільно використовувати для класифікації сигналів від сенсорів.

4) Ще одним актуальним алгоритмом є метод опорних векторів (Support Vector Machines, SVM). IoT-прикладом є безпека систем - виявлення вторгнень або аномалій у мережі.

5) Алгоритм кластеризації (K-Means, DBSCAN) доцільно використовувати для сегментації даних.

6) Для класифікації простих сигналів, тексту (NLP) у чатботах IoT рекомендовано використовувати алгоритм Наївного Байєсу (Naive Bayes). Прикладом є автоматичне класифікування повідомлень датчиків.

7) Наприклад, для обробки послідовностей даних - прогнозування, тимчасових рядів, для прогнозування температури або навантаження сервера використовують рекурентні нейронні мережі (RNN, LSTM, GRU).

8) Глибокі нейронні мережі (DNN, CNN) знаходять своє застосування в обробці зображень, відео, складних сенсорних даних, наприклад, розпізнавання об'єктів на камерах відеоспостереження IoT.

9) Є алгоритми виявлення аномалій: Isolation Forest, One-Class SVM, Autoencoders, які знаходять застосування для виявлення підозрілих подій або несправностей. Можуть застосовуватися для виявлення витoku води, несправностей обладнання.

10) Для автономного виявлення звуку, руху або падіння пристрою можливе використання алгоритмів глибинного навчання на "краю" (Edge AI), де машинне навчання застосовується прямо на сенсорному пристрої (без хмари).

Вибір алгоритму залежить від задачі (класифікація, регресія, виявлення аномалій), доступних обчислювальних ресурсів (хмара або edge) та особливостей даних (структуровані, часові ряди, зображення).

3.3 Збір даних IoT

У технологічному середовищі, що швидко розвивається, підприємства стикаються з серйозною проблемою: ефективно збирати та використовувати дані з підключених пристроїв. Оскільки галузі стикаються з великою кількістю суперечливих і неструктурованих даних, вони часто опиняються перевантаженими й не в змозі використати весь їхній потенціал. Ця неефективність пригнічує інновації та перешкоджає процесам прийняття рішень.

Потрібно знаходити способи оптимізувати збір даних IoT і перетворювати ці фрагментовані набори даних у корисну інформацію. Тобто потрібно розробити чіткий, структурований підхід до ефективного використання даних, для підвищення ефективності ведення бізнесу.

Розвиток Інтернету речей (IoT) змінив спосіб збору, обробки та використання даних. Завдяки взаємопов'язаним пристроям IoT дозволяє збирати, передавати й аналізувати великі обсяги даних, що веде до розумніших рішень і більш ефективної роботи у різних галузях.

Збір даних IoT – це процес збору даних із мережі взаємопов'язаних пристроїв і датчиків, які спілкуються один з одним через мережу Інтернет. Ці пристрої, часто вбудовані в об'єкти, відстежують і вимірюють дані у реальному часі щодо різних параметрів [15-17].

Зібрані дані IoT передаються в центральні системи, де їх можна обробляти та аналізувати для отримання корисної інформації. Ці дані можна зберігати та відновлювати у будь-який момент часу.

Етапи збору даних представлені на рис. 3.3.



Рис. 3.3 Етапи збору даних

Збір даних в системах IoT полягає у фіксації інформації з автономно працюючих процесів та пристроїв без участі людини. Такий підхід є особливо важливим у промисловості, де автоматизовані машини й виробничі лінії потребують постійного контролю для досягнення максимальної ефективності. Завдяки даним про технічний стан обладнання, робочі параметри та продуктивність, компанії можуть своєчасно виявляти недоліки, прогнозувати техобслуговування, оптимізувати процеси й уникати простоїв.

Збір даних з IoT-обладнання охоплює отримання інформації з різноманітних пристроїв, машин та технічних систем. До зібраних параметрів належать шаблони використання, технічний стан і ключові показники ефективності. Такий моніторинг у режимі реального часу сприяє безперебійній роботі обладнання та мінімізації простоїв.

Збір екологічних даних в IoT-системах передбачає отримання інформації про фізичні умови довкілля, зокрема температуру, вологість, якість повітря та рівень освітленості. Такий тип збору є надзвичайно важливим для галузей, таких як «розумне сільське господарство», де потрібно контролювати вологість ґрунту та погодні умови, а також для систем «розумного будинку».

Збір даних про місцезнаходження включає отримання інформації від IoT-пристроїв і сенсорів, оснащених GPS або іншими технологіями позиціонування.

Ці дані в режимі реального часу дозволяють відстежувати географічне положення, рух осіб або активів, а також здійснювати моніторинг працівників та виконувати інші управлінські завдання. Технології збору місцезнаходження IoT активно застосовуються у логістиці, управлінні активами, відстежуванні майна, а також у ланцюгах постачання і навігаційних системах.

Генерація даних передбачає збір даних з машин, датчиків і пристроїв, які встановлені у різних середовищах. Ці дані включають показники датчиків, інформацію про місцезнаходження, оновлення стану пристрою та інші показники.

Після збору IoT-даних, їх потрібно *передати* до центральної системи або на хмарний сервер, де відбувається подальшої обробки та аналіз даних. Передавання здійснюється за допомогою трансляторів – IoT-шлюзів. IoT-шлюз отримує повідомлення та інформацію від “розумних датчиків”, виконує попередню фільтрацію та обробку даних, а потім передає дані через проводові або бездротові мережі. IoT-шлюз забезпечує вибір ефективного протоколу передавання для конфіденційності даних та їх цілісності, забезпечуючи таким чином їх безпеку.

Для отримання необхідної інформації отримані дані повинні *оброблятися* та *аналізуватися*. До обробки даних належать: їх очищення, впорядкування аналітичних даних, додавання метаданих до даних IoT та інше.

Після цього оброблені дані потрібно *зберігати* у сховищах і базах даних для аналізу та візуалізації. Вибране рішення для зберігання даних повинне бути здатним обробляти великі обсяги IoT-даних, бути масштабованим та безпечним.

Методи *аналізу* даних, такі як машинне навчання, інтелектуальний аналіз даних, прогнозна аналітика можна використовувати для отримання корисної інформації на основі зібраних даних і збережених IoT-даних. Інформаційні панелі та *інструменти візуалізації* дозволяють організаціям інтерпретувати та передавати результати аналізу даних IoT.

4 ПРОГНОЗУВАННЯ НА ОСНОВІ ІoT-ДАНИХ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ

4.1 Підготовка середовища для створення та оцінки моделі машинного навчання

Машинне навчання (ML) - один із ключових інструментів для прогнозування на основі даних ІoT, оскільки воно дозволяє виявляти закономірності у потоках даних, що генеруються сенсорами, пристроями та системами. Машинне навчання у цьому контексті використовується для прогнозування енергоспоживання, передбачення технічного обслуговування, визначення аномалій та несправностей, оптимізації логістики, прогнозу погоди або кліматичних умов.

Основні типи машинного навчання приведені на рис. 4.1.

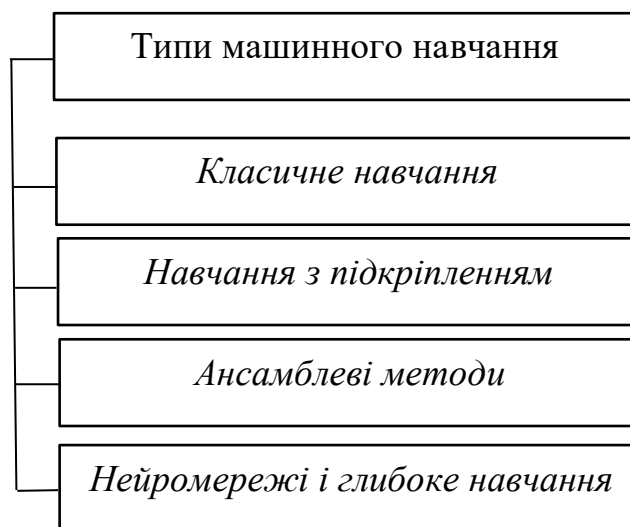


Рис. 4.1 Типи машинного навчання

Розглянемо використання машинного навчання для прогнозування температури у системі ІoT. Його можна адаптувати для реальних даних із сенсорів, замінивши згенеровані дані на реальні. Розглянемо приклад прогнозування температури у «розумному будинку» на основі даних із сенсорів. Є дані з сенсорів ІoT, які включають: час доби (години), температуру зовні,

вологість, поточну температуру у приміщенні. Потрібно спрогнозувати температуру у приміщенні через 1 годину.

```
In [1]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

Рис. 4.2 Імпорт необхідних бібліотек

На першому етапі необхідно імпортувати бібліотеки, такі як Pandas, NumPy, Sklearn. Бібліотека Pandas використовується для роботи з таблицями даних. Для спрощення запису коду будемо використовувати скорочення: *as pd*. Основні можливості бібліотеки Pandas:

- зчитування/запис CSV, Excel, JSON;
- робота з табличними даними;
- фільтрація, групування, агрегація даних;
- обробка відсутніх значень.

Бібліотека NumPy дозволяє працювати з масивами чисел і виконувати числові обчислення. Скорочення, яке дозволяє звертатися до NumPy – *as np*.

Основні можливості бібліотеки NumPy:

- робота з багатовимірними масивами;
- математичні операції (сума, середнє, стандартне відхилення та інші);
- випадкова генерація даних (наприклад, *np.random*);
- лінійна алгебра.

Бібліотека Scikit-learn (Sklearn) надає зручні інструменти для створення, тренування, оцінки та використання моделей машинного навчання. Дана бібліотека є повним інструментарієм для побудови моделей машинного навчання від початку (обробки даних) до кінця (передбачення, оцінка, налаштування).

Розділяємо набір даних на тренувальну та тестову вибірки. Розділення набору даних на тренувальну та тестову вибірки допомагає перевірити, наскільки добре модель навчилася працювати не лише на відомих, але й на нових, невідомих даних.

Тренувальна вибірка використовується для навчання моделі. На основі тренувальних даних модель підбирає внутрішні параметри (наприклад, ваги у лінійній регресії).

Тестова вибірка використовується для оцінки того, наскільки добре модель робить передбачення на даних, які вона не бачила під час навчання. Це імітує реальну ситуацію, коли модель буде працювати з новими даними.

Навчити й виконати оцінку моделі на одних і тих же даних не можна, тому що модель може завчити дані (overfitting) і давати добрі результати тільки на них, але погано працювати на нових даних. Типовим розподілом є: 80% - тренувальні дані, 20% - тестові дані (або 70/30 чи 60/40 – в залежності від задачі та обсягу даних).

Розділення даних - це спосіб перевірити, чи модель справді розуміє дані, а не просто завчила їх. Це запорука створення надійної, універсальної моделі.

На наступному етапі необхідно створити об'єкт лінійної регресії - математичну модель для знаходження залежності між однією або кількома незалежними змінними (X) та залежною змінною (y).

Далі необхідно імпортувати функція з бібліотеки Scikit-learn для обчислення середньоквадратичну помилку (MSE) між реальними значеннями та передбаченими. MSE - популярна метрика для оцінки якості регресійної моделі. Чим менше MSE, тим точніша модель.

Отже, блок коду імпорту (рис. 4.2) готує середовище для створення та оцінки моделі лінійної регресії з використанням табличних даних, які обробляються бібліотеками Pandas, NumPy, а моделювання та метрики - через Scikit-learn.

Середовищем розробки було вибране Jupyter Notebook, яке пов'язане з машинним навчанням та аналізом IoT-даних. Воно є інтерактивним, дозволяє виконувати візуалізацію даних у режимі реального часу, забезпечуючи зручну роботу з наборами даних.

В Jupyter Notebook можна виконувати код покроково, блок за блоком. У нього легко інтегруються відомі Python-бібліотеки, що особливо корисно при роботі з IoT-даними. В середовищі дуже зручно працювати з таблицями, особливо, коли необхідно швидко опрацьовувати фрагменти даних.

Можна зберігати ноутбук, ділитися ним з колегами або публікувати як звіт/дослідження. Якщо порівнювати середовище Jupyter Notebook з Google Colab, яке також є потужним інструментом для розробки, аналітики, машинного навчання та роботи з IoT-даними, можна виділити їх принципові відмінності (табл. 4.1).

Таблиця 4.1

Порівняння Google Colab та Jupyter Notebook

<i>№ n/n</i>	<i>Критерій</i>	<i>Google Colab</i>	<i>Jupyter Notebook</i>
1	Місце запуску	У хмарі (через браузер)	Локально на комп'ютері користувача
2	Доступ до GPU/TPU	Доступний безкоштовно	Не має (можна вручну налаштувати CUDA)
3	Необхідність встановлення	Не потребує встановлення	Потрібна інсталяція через Anaconda, pip
4	Швидкість доступу до даних	Обмежений доступ до локальних файлів	Миттєвий доступ до файлів на комп'ютері
5	Робота без Інтернету	Потрібне з'єднання	Повністю автономна робота
6	Права доступу/контроль	Залежить від Google (може втратити сесію)	Повний контроль над середовищем і файлами
7	Налаштування середовища	Частково обмежені	Можна кастомізувати ядро, бібліотеки, версії
8	Підключення до пристроїв IoT	Ускладнено або неможливо напряму	Можна напряму взаємодіяти з пристроями через порти
9	Підключення баз даних	Через API/бібліотеки	Напряму через драйвери (MySQL, PostgreSQL тощо)
10	Збереження та історія змін	Автоматично в Google Drive	Файли .ipynb можна зберігати, контролювати через Git
11	Спільна робота	Ідеальна для співпраці (коментарі, доступ)	Потрібно додатково налаштовувати через Git або сервер

4.2 Створення набору даних

Інтеграція IoT (Internet of Things) із штучним інтелектом (AI) трансформує традиційне виробництво, впроваджуючи інтелектуальні процеси завдяки комплексному моніторингу, аналізу даних у реальному часі, оптимізації та підвищенню ефективності контролю виробництва. Інтеграція IoT та штучного інтелекту (AI) відбувається через поєднання сенсорних технологій, технологій збору даних, аналітики та автоматизованих рішень. До основних етапів та механізмів цієї інтеграції належать:

- збір даних IoT-пристроями;
- передавання та зберігання даних;
- обробка та аналіз даних за допомогою AI;
- прийняття рішень та автоматизація;
- зворотний зв'язок і самонавчання.

Інтеграція AI та IoT відкриває величезні можливості для автоматизації та оптимізації процесів у різних сферах.

IoT-системи складаються з «розумних пристроїв», обладнаних датчиками (температури, вологості, тиску, руху, відеокамер тощо). Вони безперервно збирають дані з навколишнього середовища або технічних об'єктів.

Дані передаються через мережі (Wi-Fi, 4G, 5G, LPWAN, Bluetooth, MQTT, LoRaWAN) на центральні сервери або у хмару. Обробка може здійснюватися як на периферії (edge computing), так і в дата-центрах.

Штучний інтелект аналізує великі масиви даних, використовуючи: машинне навчання (ML) - для виявлення закономірностей і прогнозування; глибоке навчання (DL) - для розпізнавання образів, мови, аналізу відео; обробку природної мови (NLP) - для голосових команд та текстового аналізу.

AI може автоматично приймати рішення та надсилати команди пристроям IoT для виконання дій, наприклад: регулювання температури у "розумному будинку"; контроль за виробничими процесами на підприємствах; оптимізація трафіку в "розумних містах".

Чим більше система працює, тим краще вона навчається, коригує моделі та покращує ефективність прийняття рішень.

В якості прикладів інтеграції IoT та AI можна привести: «розумні міста» - наприклад, адаптивне керування освітленням, трафіком; автономний транспорт - аналіз руху, виявлення перешкод; охорона здоров'я - моніторинг стану пацієнтів у реальному часі; промисловість 4.0 - передбачуване обслуговування обладнання.

Штучний інтелект (AI) відіграє ключову роль в інтеграції з IoT, надаючи можливість аналізувати великі обсяги даних, автоматизувати процеси та приймати розумні рішення в реальному часі. Основні аспекти ролі AI в IoT включають: аналіз та обробку даних, прогнозу аналітику, автоматизацію та прийняття рішень, розпізнавання та ідентифікацію, оптимізацію роботи IoT-пристроїв, підвищення безпеки.

AI обробляє великі масиви даних, які отримані від IoT-пристроїв, і допомагає виявляти закономірності, тренди та аномалії. Машинне навчання дозволяє прогнозувати майбутні події, наприклад, можливі поломки обладнання (predictive maintenance) або зміни в поведінці споживачів. AI може автономно приймати рішення на основі аналізованих даних, наприклад, автоматично регулювати енергоспоживання в будівлі або оптимізувати виробничі процеси.

Завдяки глибокому навчанню AI може розпізнавати зображення, мову та жести, що корисно для безпеки, контролю доступу, управління голосовими командами. AI допомагає ефективно використовувати ресурси, оптимізувати трафік даних та зменшувати затримки обробки за допомогою edge computing. AI-алгоритми можуть виявляти кіберзагрози, аномальну активність та реагувати на потенційні атаки на IoT-мережі. Таким чином, штучний інтелект виступає як "мозок" IoT-систем, перетворюючи дані в корисні рішення та забезпечуючи автономність і ефективність інтегрованих технологій.

Поєднання AI, ML, IoT створює потужну синергію, яка відкриває новий рівень інтелектуальної автоматизації, аналітики та адаптації систем у реальному часі. IoT-пристрої генерують велику кількість даних (наприклад, сенсори, камери,

трекери). ML-алгоритми виявляють закономірності, а AI приймає рішення - автоматично, без участі людини.

AI у поєднанні з ML та IoT є автономними, адаптивними, «розумними системами», що вчаться на досвіді, приймають рішення та допомагають людині діяти ефективніше, безпечніше і швидше.

```
In [2]: np.random.seed(42)
data = {
    "Година": np.arange(0, 24), # Від 0 до 23, години доби.
    "Температура_зовні": np.random.uniform(10, 35, 24), # Випадкові значення температури від 10°C до 35°C.
    "Вологість": np.random.uniform(40, 90, 24), # Випадкові значення вологості від 40% до 90%.
    "Поточна_температура": np.random.uniform(18, 28, 24), # Випадкові значення температури в приміщенні від 18°C до 28°C.
}
df = pd.DataFrame(data) # Створення DataFrame з цими даними.
df["Цільова_температура"] = df["Поточна_температура"] + np.random.uniform(-1, 2, 24)
```

Рис. 4.3 Створення набору даних

Для генератора випадкових чисел NumPy встановлюємо параметр `np.random.seed(42)`. Це основа для генератора випадкових чисел NumPy, щоб забезпечити відтворюваність: кожен раз, коли запускається код, отримуємо однакові випадкові числа. Це важливо для тестування та наукових експериментів. `np.random.seed(42)` фіксує генератор випадкових чисел для відтворюваності результатів. Це означає, що кожного разу будуть генеруватися однакові випадкові значення.

`data = { ... }` - це словник, який створює набір даних із чотирьох змінних:

- година - створює масив чисел від 0 до 23 (тобто 24 години доби). Кожне число - це окрема година доби.
- температура зовні - генерує 24 випадкових значення температури зовнішнього середовища від 10°C до 35°C. Це імітація температури на вулиці щогодини.
- вологість - генерує 24 випадкових значення вологості повітря від 40% до 90%. Це імітація зміни вологісних умов упродовж доби.
- поточна температура - генерує 24 випадкових значення температури всередині приміщення від 18°C до 28°C. Це, наприклад, те, що показує термостат або внутрішній IoT-сенсор.

$df = pd.DataFrame(data)$ - створює таблицю (DataFrame) з усіх згенерованих даних, де кожен рядок - це окрема година доби. Кожен рядок - це година, кожен стовпець - одна з ознак (features) [19, 20].

$df["Цільова_температура"] = \dots$ - додає до таблиці новий стовпець - "Цільова_температура", яка дорівнює: Поточна_температура + випадкове число від -1 до 2. Це імітує ситуацію, коли система хоче трохи підвищити або знизити температуру, наприклад, залежно від зовнішніх умов або потреб користувача. До кожного значення поточної температури додається випадкове число в діапазоні від -1 до +2. Таким чином, цільова температура - це дещо відкориговане значення поточної, яке може бути трохи нижчим або вищим. Це може моделювати бажану температуру, задану системою клімат-контролю або користувачем на основі зовнішніх умов. Приклад результату приведений у табл. 4.2.

Таблиця 4.2

Приклад результату

Година	Температура зовні	Вологість	Поточна температура	Цільова температура
7 ⁰⁰	15.0	80	20.0	20.9
8 ⁰⁰	16.0	78	21.0	21.9
9 ⁰⁰	18.0	75	22.0	22.1
11 ⁰⁰	25.4	67.2	21.3	22.1
13 ⁰⁰	27.1	72.8	22.5	23.9

Код створює синтетичний набір даних, який можна використовувати для:

- моделювання управління температурою в IoT-системі;
- навчання моделі прогнозування цільової температури;
- візуалізації змін кліматичних параметрів протягом доби.

4.3 Створення матриці вхідних ознак

На наступному етапі створюється матриця вхідних ознак (features). У неї входять чотири колонки з DataFrame df , які будуть використовуватися для передбачення, а саме: година, температура зовні, вологість, поточна температура.

Це ті вхідні параметри, на основі яких модель буде вчитись прогнозувати цільове значення. Цільове значення (або цільова змінна) - це те, що ми хочемо передбачити або класифікувати за допомогою моделі машинного навчання. Є таблиця з параметрами температури, вологості та часу, і потрібно передбачити "Цільову_температуру". Ознаками (вхідними даними) в завданні є: година, температура зовні, вологість, поточна температура. Цільове значення – це цільова температура, яку ми хочемо отримати в результаті прогнозу. Цільове значення - це як правильна відповідь, яку модель має навчитися визначати на основі наданих їй прикладів.

```
In [3]: X = df[["Година", "Температура_зовні", "Вологість", "Поточна_температура"]] # Ознаки (фактори).
y = df["Цільова_температура"] # Цільова змінна.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Рис. 4.4 Створення матриці вхідних ознак

Далі відбувається розділення даних на тренувальні та тестові. *train_test_split()* - функція із *sklearn.model_selection*, яка розбиває дані на тренувальний та тестовий набір:

- тренувальний набір (train): 80% даних;
- тестовий набір (test): 20% даних.

test_size=0.2 - означає, що 20% всіх даних підуть на тест (перевірку), а решта - для навчання.

random_state=42 - фіксує випадковість, щоб при кожному запуску розподіл був однаковий.

Вміст змінних відображений у табл. 4.3.

Таблиця 4.3

Вміст змінних

№ п/п	Змінна	Вміст
1	X_train	Вхідні ознаки для навчання (80%)
2	X_test	Вхідні ознаки для тестування (20%)
3	y_train	Відповіді (цільові значення) для навчання
4	y_test	Відповіді для оцінки точності моделі

4.4 Вибір алгоритму для навчання моделі ML

Для вирішення поставленого завдання використовується класичне навчання.

Класичне навчання (у контексті машинного навчання) - це процес, при якому модель навчається на основі історичних (навчальних) даних, щоб згодом робити передбачення або класифікацію для нових, раніше не бачених даних. Якщо потрібно передбачити температуру, то модель навчається на прикладах, де відомо, яка температура була при певних умовах. Дані діляться на:

- *training set* - для навчання моделі;
- *test set* - для перевірки, як добре модель працює.

Метою навчання є мінімізація помилки. При прогнозі температури - це середньоквадратична помилка (MSE). Модель буде намагатися знайти таку залежність, щоб різниця між передбаченням і реальним значенням була якнайменшою.

Класичне навчання є основою багатьох систем AI, особливо у прогнозуванні на основі IoT-даних.

Класичне навчання поділяється на навчання з вчителем, та навчання без вчителя.

Навчання з вчителем (Supervised Learning) - це тип машинного навчання, при якому модель навчається на основі розмічених даних, тобто таких, де кожному прикладу відповідає правильна відповідь (мітка або цільове значення). Модель отримує на вхід приклади (набір ознак) та правильні відповіді (цільову змінну) і вчиться знаходити залежність між ними. Після навчання вона може передбачати результат для нових прикладів, де правильна відповідь ще невідома.

Наприклад, є IoT-система, яка збирає дані (година доби, температура, вологість, поточна температура в приміщенні), має цільове значення: бажана (оптимальна) температура, яку треба досягти. Мета навчання з вчителем - навчити модель передбачати бажану температуру в залежності від вхідних даних.

Навчання без вчителя (Unsupervised Learning) - це тип машинного навчання, при якому модель працює з нерозміченими даними, тобто без готових правильних

відповідей або міток. Її завдання - знайти закономірності, структуру або схожість у даних самостійно.

У рамках виконання бакалаврської кваліфікаційної роботи було використане навчання із вчителем. Навчання із вчителем вирішує такі задачі як класифікація та регресія. Регресія - це тип завдання у машинному навчанні (зокрема, навчанні із вчителем), який використовується для передбачення числових значень на основі вхідних даних (ознак). Простіше кажучи, регресія допомагає відповісти на питання: "Скільки?" або "Яке буде значення?"

Вибираючи алгоритм, який підходить для поставленого у роботі завдання, було визначено, що найбільш популярними алгоритмами для вирішення задач регресії є: лінійна регресія, поліноміальна регресія, Ridge/Lasso Regression, метод найближчих сусідів (K-NN), градієнтний бустинг (XG Boost Light GBM).

Для прогнозування було вибрано алгоритм лінійної регресії (рис. 4.5).

```
In [4]: model = LinearRegression()  
        model.fit(X_train, y_train)
```

```
Out[4]: LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Рис. 4.5 Лінійна регресія

Лінійна регресія використовується для передбачення числового значення на основі однієї або кількох вхідних змінних (факторів/ознак), припускаючи, що між ними є лінійна залежність. Порівнюючи лінійну регресію з іншими алгоритмами машинного навчання, було визначено наступне.

Поліноміальна регресія може враховувати нелінійні зв'язки, добре працює з не прямими трендами, але має гіршу швидкодію, ніж у лінійної регресії. Ridge/Lasso регресія може обирати найважливіші ознаки, але працює тільки з лінійними залежностями. Метод К-ближчих сусідів (K-NN) простий у реалізації, адаптивний до локальних змін, але повільний при передбаченні, особливо на великих даних. Градієнтний бустинг має високу точність, враховує нелінійні зв'язки, при цьому має таку складність як довге навчання та необхідність налаштувань.

Алгоритм лінійної регресії доцільно використовувати при наявності задачі з лінійною залежністю, якщо потрібна швидкість навчання моделі ML.

```
In [5]: y_pred = model.predict(X_test)
```

Рис. 4.6 Прогнозування за допомогою ML-моделі

Команда `y_pred = model.predict(X_test)` виконує прогнозування за допомогою вже навченої моделі машинного навчання. `model` - це навчена модель. Перед цим модель лінійної регресії була навчена на тренувальних даних.

`X_test` - це тестові входні дані, тобто ті дані, яких модель раніше не бачила. Вони використовуються для перевірки точності прогнозу.

`model.predict(X_test)` - повертає прогнозовані значення. Модель бере на вхід `X_test` і видає набір передбачених значень - прогнозів.

`y_pred` - масив прогнозованих значень цільової змінної, які визначила модель. Їх можна порівняти з `y_test` (реальні значення) для оцінки точності моделі.

4.5 Оцінка якості моделі регресії

Оцінка якості регресійної моделі - це процес, під час якого визначається, наскільки точно модель передбачає вихідні значення (цільову змінну), базуючись на входних даних (ознаках). Оцінка проводиться за допомогою метрик MSE, MAE, R^2 . Оцінка дозволяє зрозуміти, наскільки добре модель навчається на даних, чи не перенавчилася (overfitting) або недонавчилася (underfitting) ML-модель і чи відповідають передбачення реальності.

Оцінка якості моделі регресії дозволяє:

- визначити ефективність ML-моделі;
- уникнути перенавчання/недонавчання;
- покращити передбачення;
- зменшити ризики у реальних застосуваннях;
- обрати найкращий підхід для задачі.

Основні метрики для оцінки регресії приведені у табл. 4.4.

Таблиця 4.4

Основні метрики для оцінки регресії

№ п/п	Метрика	Опис
1	MSE (Mean Squared Error)	Середньоквадратична помилка - чутлива до великих помилок.
2	MAE (Mean Absolute Error)	Середня абсолютна помилка - проста та стабільна.
3	R ² (коефіцієнт детермінації)	Від 0 до 1 - показує, скільки варіації цільової змінної пояснює модель.

```
In [6]: mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
```

Mean Squared Error: 2.26

Рис. 4.7 Оцінка якості ML-моделі

Середньоквадратична помилка (MSE) - один з основних показників якості моделі регресії. MSE підносить кожену помилку до квадрата, тому великі відхилення впливають набагато більше, ніж малі. Якщо для нашої задачі більші помилки несуть серйозні наслідки - MSE є кращим вибором. Також, якщо важливим є математично точно виміряти відхилення значень від істинних, з урахуванням впливу великих відхилень – підходить MSE, так як добре працює в аналітичних, статистичних задачах.

Функція `mean_squared_error()` порівнює:

- `y_test` - реальні значення (правильні відповіді);
- `y_pred` - прогнозовані значення моделі.

Функція обчислює середнє значення квадратів різниці між реальними та прогнозованими значеннями:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{\text{test}}^{(i)} - y_{\text{pred}}^{(i)})^2 \quad (1.1)$$

Чим менше значення MSE, тим краща точність моделі. Оцінку якості моделі регресії для даної задачі можна визначити за шкалою, яка приведена у табл. 4.5.

Якість моделі

№ п/п	MSE	Якість моделі
1	0 - 1	Дуже добре
2	1 - 2.5	Добре
3	2.5 - 3	Задовільно/добре
4	> 3 -5	Середньо
5	> 5	Слабка модель

Модель показала результат: $MSE = 2.26$, який по шкалі якості ML-моделі відповідає значенню «добре».

Нижче представлений код, який виконує прогноз майбутньої температури через 1 годину на основі нових даних за допомогою раніше навченої моделі (рис. 4.8).

```
In [7]: new_data = pd.DataFrame({
    "Година": [15], # 15-та година (3-тя дня).
    "Температура_зовні": [25], # Температура зовні 25°C.
    "Вологість": [60], # Вологість 60%.
    "Поточна_температура": [22], # Поточна температура 22°C.
})
future_temp = model.predict(new_data)
print(f"Прогнозована температура через 1 годину: {future_temp[0]:.2f} °C")
```

Прогнозована температура через 1 годину: 22.63 °C

Рис. 4.8 Прогноз температури через 1 годину

Створюється датафрейм. Модель, яка раніше навчена, отримує значення на вхід і видає прогнозовану цільову температуру, тобто значення температури, яке має бути на думку моделі через деякий час (у випадку поставленого завдання – через 1 годину).

Тобто модель вважає, що при вищевказаних умовах температура у приміщенні стане 22.63°C.

Прогнозуючи температуру у приміщенні через 1 годину, можна оптимізувати енергоспоживання. Наприклад, якщо модель прогнозує підвищення температури, система може автоматично знизити потужність кондиціонера або бойлера, щоб

зменшити енергоспоживання. Це підходить для «розумного будинку», який активно використовує технології машинного навчання для управління енергією.

Прогнозування температури може бути частиною більш комплексної моделі, де одночасно прогнозується температура та енергоспоживання, що дозволить побудувати ефективну систему управління енергією, що планується втілити у подальших дослідженнях та магістерській кваліфікаційній роботі.

Прогнозування температури у приміщенні та енергоспоживання можуть бути об'єднані в єдину систему для «розумного будинку». Таким чином, система обробки даних може включати не лише прогнозування енергоспоживання, але й регулювання температури для оптимізації енергетичних витрат.

Практичне застосування подібної моделі машинного навчання може бути використане у системах “розумного клімату”, наприклад, для автоматичного керування кондиціонерами та вентиляцією, а також у промисловості - для оптимізації енергоспоживання з метою підтримки стабільного мікроклімату.

ВИСНОВКИ

У дипломній роботі розглянемо використання машинного навчання для прогнозування температури у системі IoT. IoT генерує великі обсяги даних, які можна аналізувати для прийняття обґрунтованих рішень, прогнозування трендів і розв'язання проблем. Завдяки можливості взаємодії між пристроями та обміну даними IoT відкриває нові горизонти для інновацій у багатьох сферах життя.

Штучний інтелект може обробляти великі обсяги даних, зібраних від IoT-пристроїв та аналізувати їх у реальному часі для прийняття обґрунтованих рішень. Це дає змогу автоматизувати процеси, покращуючи ефективність і точність без необхідності втручання людини. Інтеграція IoT і штучного інтелекту дозволяє оптимізувати використання ресурсів (енергії, часу, матеріалів), зменшуючи витрати та підвищуючи продуктивність. Наприклад, «розумні будівлі» можуть автоматично налаштовувати освітлення або кондиціонування, базуючись на поведінці користувачів. Штучний інтелект здатний обробляти дані з датчиків IoT для виявлення аномалій та загроз. IoT-пристрої можуть збирати дані про вподобання та звички користувачів, а штучний інтелект обробляє ці дані для створення персоналізованих рекомендацій або налаштувань.

Інтеграція IoT та штучного інтелекту дає змогу створювати системи, які є не лише більш ефективними та адаптивними, але й здатними до самовдосконалення, що відкриває нові можливості для розвитку технологій у багатьох сферах.

У роботі визначені можливості бібліотек Pandas, NumPy, Sklearn, які використовувалися у розробці моделі машинного навчання.

В якості середовища розробки було вибрано Jupyter Notebook, яке пов'язане з машинним навчанням та аналізом IoT-даних. Воно є інтерактивним, дозволяє виконувати візуалізацію даних у режимі реального часу, забезпечуючи зручну роботу з наборами даних.

У рамках виконання бакалаврської кваліфікаційної роботи було використане навчання із вчителем. Навчання із вчителем вирішує такі задачі як класифікація та регресія. Регресія - це тип завдання у машинному навчанні (зокрема, навчанні із

вчителем), який використовується для передбачення числових значень на основі вхідних даних (ознак).

В якості алгоритму для навчання моделі ML був вибраний алгоритм лінійної регресії, так як лінійна регресія використовується для передбачення числового значення на основі однієї або кількох вхідних змінних (факторів/ознак), припускаючи, що між ними є лінійна залежність. Алгоритм лінійної регресії доцільно використовувати при наявності задачі з лінійною залежністю, якщо потрібна швидкість навчання моделі ML.

Визначені основні метрики для оцінки регресії, такі як MSE (Mean Squared Error), MAE (Mean Absolute Error), R^2 (коефіцієнт детермінації).

Оцінка якості ML-моделі була виконана на базі метрики MSE – середньоквадратичної помилки, як одного з основних показників якості моделі регресії.

Модель показала результат: $MSE = 2.26$, який по шкалі якості ML-моделі відповідає значенню «добре».

Практичне застосування подібної моделі машинного навчання може бути використане у системах “розумного клімату”, наприклад, для автоматичного керування кондиціонерами та вентиляцією, а також у промисловості - для оптимізації енергоспоживання з метою підтримки стабільного мікроклімату.

Поєднання IoT та ML значно розширить можливості автоматизації та інтелектуальної обробки даних, створюючи інноваційні рішення для різних галузей. Завдяки інтеграції штучного інтелекту в IoT-системи можна не тільки збирати дані, але й отримувати з них цінну інформацію для прогнозування, що відкриває нові можливості для автоматизації та покращення різних сфер діяльності.

ПЕРЕЛІК ПОСИЛАНЬ

1. <http://realpython.com/python3-object-oriented-programming/>
2. <http://leetcode.com/problemset/all/>
3. <http://perso.limsi.fr/pointal/>
4. <http://media/python:cours:mementopython3-english.pdf>
5. <https://numpy.org/>
6. <https://matplotlib.org/>
7. <https://pytorch.org/>
8. <https://www.tensorflow.org/>
9. Jake VanderPlas Python Data Science Handbook: Essential Tools for Working with Data, 2022, 551p.
10. Heller, Nicholas et al. (2020). The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes. arXiv: 1904. 00445 [q-bio.QM].
11. Hollo, Kaspar (2019). Exploring the Value of Weakly-Supervised Deep Learning Approaches for Artefact Segmentation in Brightfield Microscopy Images. URL: https://comserv.cs.ut.ee/home/files/hollo_softwareengineering_2021.
12. Wang, Haofan et al. (2020). Score-CAM: Score-Weighted Visual Explanations for Convo-lutional Neural Networks. arXiv: 1910.01279 [cs.CV].
13. Yang, Guanyu et al. (2020). “Weakly-supervised convolutional neural networks of renal tumor segmentation in abdominal CTA images”. In: BMC Medical Imaging 20.1, p. 37. ISSN: 1471-2342. DOI: 10.1186/s12880-020-00435-w. URL: <https://doi.org/10.1186/s12880-020-00435-w>.
14. Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., Debbah, M. A tutorial on UAVs for wireless networks: Applications challenges and open problems. IEEE Commun. Surveys Tuts., 21(3), 2019. — 2334-2360 c.
15. Zhang, J., Zhou, L., Tang, Q., Ngai, E. C.-H., Hu, X., Zhao, H., et al. Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing. IEEE Internet Things J., 6(2), 2019. — 3688-3699 c.

16. Zhou, F., Hu, R. Q., Li, Z., Wang, Y. Mobile edge computing in unmanned aerial vehicle networks. *IEEE Wireless Commun.*, 27(1), 2020. — 140-146 c.
17. Mishra, D., Natalizio, E. A survey on cellular-connected UAVs: Design challenges enabling 5G/B5G innovations and experimental advancements. *Comput. Netw.*, 182, Dec. 2020.
18. Hua, M., Wang, Y., Li, C., Huang, Y., Yang, L. UAV-aided mobile edge computing systems with one by one access scheme. *IEEE Trans. Green Commun. Netw.*, 3(3), 2019. — 664-678 c.
19. Yao, H., Mai, T., Jiang, C., Kuang, L., Guo, S. AI Routers & Network Mind: A Hybrid Machine Learning Paradigm for Packet Routing. *IEEE Computational Intelligence Magazine*, 14(4), 2019. — 21-30 c.
20. Li, L., Wen, X., Lu, Z., Pan, Q., Hu, W. J. A. Z. Energy-efficient UAV-enabled MEC system: Bits allocation optimization and trajectory design. *Sensors*, 19(20), 2019. — 4521 c.
21. Wu, G., Miao, Y., Zhang, Y., Barnawi, A. Energy efficient for UAV-enabled mobile edge computing networks: Intelligent task prediction and offloading. *Comput. Commun.*, 150, Jan. 2020. — 556-562 c.
22. Selvaraju, Ramprasaath R. et al. (2019). “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *International Journal of ComputerVision* 128.2, pp. 336–359. DOI: 10 . 1007 / s11263 - 019 - 01228 - 7. URL: <https://doi.org/10.1007/s11263-019-01228-7>.
23. Грас Д. Data Science. Наука про дані з нуля: Пер. з англ. – 2-е видання., перероб. та доп. – 2021. – 416 с.
24. Лі, Кай-Фу Штучний інтелект: 10 передбачень для майбутнього / Кай-Фу Лі, Чень Цюфань; пер. з англ. А. Райчук. – Київ: Форс Україна, 2022. – 464 с.
25. Зінченко О.В., Фесенко М.А., Березівський М.Ю., Кисіль Т.М. Технології смарт-систем. – Навчальний посібник. – К.: ДУІКТ, 2023. – 162 с.
26. Золотухіна О.А., Негоденко О.В., Резник С.Ю., Разіна С.Я. Якість та тестування інформаційних систем. Навчальний посібник підготовлено до друку

для самостійної роботи студентів вищих навчальних закладів. Київ: ННІТ ДУТ, 2020. – 128 с.

27. Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина Системи штучного інтелекту: навчальний посібник. – Львів: «Магнолія-2006», 2024. – 279 с.

28. Лосєв М.Ю. Програмування мовою Python: навчальний посібник / М. Ю. Лосєв, В. М. Федорченко. – Харків, - Львів: Видавництво ПП «Новий Світ - 2000», 2023. – 178 с.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ
(Презентація)

