

Міністерство освіти і науки України
Львівський національний університет природокористування
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему:

«Інтелектуальний сервіс підбору параметрів управління
твердопаливними котлами»

Виконав: студент групи Іт-41сп
спеціальності 126
«Інформаційні системи та технології»

Мілян Д.І.

Керівник:

Лиса О.В.

ЛЬВІВ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Освітній ступінь «Бакалавр» за спеціальністю –
126 – „Інформаційні системи та технології”

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 202_ р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту
Мілянчу Дмитру Івановичу

1. Тема роботи

«Інтелектуальний сервіс підбору параметрів управління твердопаливними котлами»

Керівник роботи: Лиса Ольга Володимирівна, к.т.н., доцент.

затверджена наказом по університету від “27” листопада 2023 р., № 641/к-с.

2. Строк подання студентом роботи: 10.06.2024 р.

3. Початкові дані до роботи:

Технологічні вимоги та обмеження для побудови інтелектуальних сервісів для систем “розумний будинок”; ДСТи, СНіПи; документація середовищ розробки

4. Зміст розрахунково-пояснювальної записки:

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Класифікація інтернет-сервісів

1.2 Сервіс-орієнтована архітектура

1.3 Компонування Web-сервісів

1.4 Інтелектуальні веб-сервіси

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Інтернет речей

2.2 Етапи створення інтелектуального сервісу

2.3 Порівняння та вибір технології зв'язку IoT

2.4 Вибір інструментарію реалізації системи

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Моделювання роботи твердопаливного котла

3.2 Реалізація інтелектуальної моделі та оцінка її роботи

3.3 Розробка системи для роботи пристроїв IoT

4 ОХОРОНА ПРАЦІ

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
Інтелектуальні веб-сервіси. Вибір апаратного і програмного забезпечення.

Моделювання роботи котла. Результати роботи інтелектуальної моделі. Серверний процес. Графічний інтерфейс. Висновки.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	<i>Лиса О.В., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри управління проектами та безпеки виробництва</i>		

7. Дата видачі завдання _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Написання першого розділу та означення головних завдань роботи</i>	10.02 - 21.03.24	
2	<i>Виконання другого розділу та формування початкових даних</i>	22.03 - 11.04.24	
3.	<i>Виконання третього розділу та узагальнення отриманих результатів роботи</i>	12.04 - 11.05.24	
4.	<i>Написання розділу: «Охорона праці»</i>	12.05 - 17.05.24	
5.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	18.05 - 24.05.24	
6.	<i>Завершення роботи в цілому</i>	24.05 - 10.06.24	

Студент _____ Мілян Д.І.
(підпис)

Керівник роботи _____ Лиса О.В.
(підпис)

РЕФЕРАТ

УДК 004.896 : 644.1

Інтелектуальний сервіс підбору параметрів управління твердопаливними котлами

Мілян Д.І. Кафедра ІТ – Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 53 с. текст. част., 10 рис., 1 табл. 12 арк. ілюстраційного матеріалу, 15 джерел.

Об’єкт дослідження – процес управління пристроями Інтернету речей за допомогою веб-сервісів.

Мета роботи – розробка інтелектуального сервісу підбору параметрів управління твердопаливними котлами.

Проведено аналіз предметної області та проведена класифікація інтернет-сервісів, розглянуті їх основні функції, переваги та недоліки. Розглянуті можливості інтелектуальних веб-сервісів, зокрема, для задач інтернету речей. Обрані апаратне та програмне забезпечення створення "розумного будинка". Проаналізована задача моделювання температурного режиму твердопаливного котла, визначені параметри пошуку рішення методами машинного навчання. Створена інтелектуальна модель та перевірена її робота. Обрана структура програмного забезпечення контролера для IoT, наведений приклад роботи системи при підключенні нових пристроїв та вигляд графічного інтерфейса. Розглянуті питання охорони праці та техніки безпеки.

Ключові слова: інтернет речей, інтелектуальний веб-сервіс, модель штучного інтелекту, програмне забезпечення контролера "розумного будинку".

Keywords: Internet of Things, intelligent web service, artificial intelligence model, smart home controller software.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Класифікація інтернет-сервісів	8
1.2 Сервіс-орієнтована архітектура	11
1.3 Компонування Web-сервісів	15
1.4 Інтелектуальні веб-сервіси	18
РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ	21
2.1 Інтернет речей	21
2.2 Етапи створення інтелектуального сервісу	25
2.3 Порівняння та вибір технології зв'язку IoT	28
2.4 Вибір інструментарію реалізації системи	31
РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	34
3.1 Моделювання роботи твердопаливного котла	34
3.2 Реалізація інтелектуальної моделі та оцінка її роботи	38
3.3 Розробка системи для роботи пристроїв IoT	40
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	47
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52

ВСТУП

Сучасний етап розвитку промисловості характеризується інтенсивним впровадженням інформаційних технологій у всі сфери виробництва. Однією з найважливіших галузей, що піддається значним трансформаціям, є енергетика, зокрема виробництво теплової енергії. Котли, як ключові компоненти систем опалення, відіграють важливу роль у забезпеченні енергоефективності та економічної доцільності підприємств. Саме тому розробка інтелектуальних сервісів для підбору параметрів котлів є надзвичайно актуальною темою, що заслуговує на детальне вивчення та аналіз.

Однією з основних проблем сучасного виробництва є оптимізація енергетичних витрат та підвищення ефективності використання паливно-енергетичних ресурсів. У цьому контексті особливої ваги набуває правильний підбір параметрів роботи котлів, адже від цього залежить як економічна ефективність, так і екологічна безпека експлуатації систем опалення. Традиційні методи налаштування котлів часто не відповідають сучасним вимогам, оскільки не враховують усієї складності і багатогранності процесів, що відбуваються в них.

Застосування інтелектуальних сервісів для підбору параметрів котлів дозволяє значно підвищити точність налаштувань, знизити витрати на паливо, зменшити викиди шкідливих речовин в атмосферу, а також підвищити загальну надійність систем опалення. Такі сервіси базуються на використанні сучасних методів аналізу даних, машинного навчання та штучного інтелекту, що забезпечує високий рівень адаптивності та автоматизації процесу налаштування.

Інтелектуальні сервіси підбору параметрів котлів мають важливе значення не тільки для окремих підприємств, але і для економіки країни в цілому. Енергоефективність є ключовим фактором сталого розвитку, адже зниження енергетичних витрат сприяє зменшенню собівартості продукції, підвищенню

конкурентоспроможності національної економіки та зниженню залежності від імпорту енергоресурсів.

Крім того, використання інтелектуальних систем дозволяє зменшити негативний вплив на навколишнє середовище. Сучасні підходи до екологічної безпеки вимагають мінімізації викидів шкідливих речовин, і правильне налаштування параметрів роботи котлів є одним із ефективних способів досягнення цієї мети. Отже, впровадження інтелектуальних сервісів підбору параметрів котлів сприяє досягненню глобальних цілей сталого розвитку.

Метою даного дослідження є розробка та впровадження інтелектуального сервісу для підбору параметрів котлів, що забезпечить підвищення ефективності їх роботи та зниження енергетичних витрат.

Практичне значення дослідження полягає у можливості впровадження розробленого сервісу на підприємствах, що експлуатують котли, з метою підвищення їх енергоефективності та зниження експлуатаційних витрат. Це дозволить підприємствам зменшити витрати на паливо, підвищити надійність роботи обладнання та зменшити екологічний вплив на навколишнє середовище.

Таким чином, дана дипломна робота присвячена актуальній проблемі підвищення ефективності та надійності роботи твердопаливних котлів за допомогою інтелектуальних сервісів. Впровадження таких технологій дозволить досягти значних успіхів у сфері енергетики, забезпечуючи сталий розвиток та збереження навколишнього середовища.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Класифікація інтернет-сервісів

Сьогодення немислиме без залучення глобальної інформаційної мережі Інтернет та її різноманітних ресурсів. Інтернет-ресурси можна класифікувати за різними ознаками, такими як призначення, тип контенту, цільова аудиторія тощо. Ось деякі основні класифікаційні критерії:

1. Призначення:

- інформаційні ресурси: включають веб-сайти, блоги, новинні портали, енциклопедії та інші ресурси, які містять інформацію з різних областей;
- розважальні ресурси: це веб-сайти, відеохостинги, ігрові платформи, стрімінгові сервіси та інші ресурси, які надають розважальний контент;
- комерційні ресурси: включають інтернет-магазини, послуги онлайн-торгівлі, рекламні платформи та інші ресурси, які мають комерційну спрямованість.

2. Тип контенту:

- текстові: веб-сайти, блоги, новинні портали;
- мультимедійні: відеохостинги, аудіо- та відеострімінгові платформи, фотогалереї тощо;
- інтерактивні: веб-додатки, ігрові платформи, соціальні мережі.

3. Цільова аудиторія:

- загальна: веб-сайти, які призначені для широкої аудиторії та мають загальну спрямованість;
- спеціалізована: ресурси, які адресовані певним категоріям користувачів або відображають інформацію з певних областей інтересів.

4. Формат доступу:

- веб-сайти: веб-сторінки, які можна переглядати через веб-браузер;

- веб-додатки: інтерактивні програми, що працюють через веб-браузер та надають різноманітну функціональність.

Окремо можна ще сказати про веб-сервіси, які покликані обробляти інформацію згідно закладених в них цільових програм та створені в основному, для забезпечення обробки запитів не людини, а програмних засобів. Веб-сайти, веб-сервіси та веб-додатки є різними за своїми призначенням та функціоналом.

Веб-сайти - це колекція веб-сторінок, які зазвичай мають статичний або малопомітний динамічний зміст. Веб-сайти використовуються для представлення інформації, реклами, комунікації з користувачами тощо. Вони можуть включати такі розділи, як домашня сторінка, "Про нас", контактна інформація, блог тощо. Веб-сайти можуть мати обмежену взаємодію з користувачем і не вимагати авторизації.

Веб-сервіси - це програмне забезпечення, яке надає функціональність для виконання конкретних завдань через мережу Інтернет. Веб-сервіси можуть виконувати різноманітні функції, такі як обмін даними між різними системами, обробка запитів, робота з базами даних тощо. Вони надають стандартизовані інтерфейси для взаємодії між різними програмними системами.

Веб-додатки - це програмне забезпечення, яке виконується на веб-сервері і може бути доступне через веб-браузер. Веб-додатки можуть мати різний функціонал, від простих інтерактивних інтерфейсів до складних систем управління контентом, електронної комерції, соціальних мереж тощо. Вони можуть включати взаємодію з базами даних, авторизацію користувачів, обмін даними з веб-сервером. Веб-додатки можуть бути створені як для користувачів, так і для адміністраторів.

Хоча веб-сайти, веб-сервіси та веб-додатки можуть використовуватися в онлайн-середовищі та мають спільні особливості, вони відрізняються за своїми функціями та призначенням.

1. Веб-сайт:

- онлайн-платформа або ресурс, який складається з статичних або динамічних веб-сторінок;

- зазвичай призначений для публічного доступу та забезпечення інформацією про компанію, продукти, послуги, контакти тощо;

- інтерфейс зазвичай зорієнтований на користувача та призначений для перегляду інформації, може містити статичний контент, блоги, фотогалереї тощо.

2. Веб-сервіс:

- програмне забезпечення або функціональність, яка надається через мережу Інтернет і доступна для інтеграції в інші програмні продукти або системи;

- зазвичай працює за протоколами HTTP або HTTPS та використовується для обміну даними між різними програмними продуктами;

- призначений для автоматизації певних процесів, спрощення взаємодії між різними системами або побудови розподілених додатків.

3. Веб-додаток:

- програмне забезпечення, яке розроблене для роботи через веб-браузер та доступне через Інтернет;

- зазвичай має складніший інтерфейс користувача та може забезпечувати широкий спектр функціональності, включаючи авторизацію, обробку даних, взаємодію з базами даних тощо;

- призначений для виконання різноманітних завдань, від управління бізнес-процесами до розваг та освіти.

Отже, веб-сайти призначені для публічного представлення інформації, веб-сервіси для обміну даними між системами, а веб-додатки - для різноманітних завдань, включаючи взаємодію з користувачем та роботу з даними.

Веб-сервіси можуть бути створені за різними підходами та архітектурами, але дуже часто використовують підхід, який має назву сервіс-орієнтована архітектура.

1.2 Сервіс-орієнтована архітектура

Центральним елементом сервіс-орієнтованої архітектури (COA, англ. SOA) є здатність розробляти нові додатки за допомогою компонування функціональності підприємства у вигляді послуг, як в рамках однієї організації, так і між безліччю підприємств.

Семантичні описи послуг, включаючи анотації функціональних і нефункціональних атрибутів, спрямовані на автоматизацію цього процесу і створення більш якісних рішень. Існує необхідність описувати можливості сервісів унікальною інтерпретованою, комп'ютерно читабельною мовою та підвищити якість та стійкість існуючих завдань, таких як веб-сервіси та виклики процедур.

Однією з цілей такого проекту було би створення засобу для автоматичного вибору, використання, створення та відстеження таких веб-сервісів. Онтології, які в широкому сенсі визначаються як спільні концепції формальної предметної області (ПрО), можуть відігравати вирішальну роль у налагодженні зв'язків, обміні даними та обміні знаннями між додатками.

Сервіс-орієнтована архітектура - це парадигма проектування, розробки та управління функціональними модулями (сервісами), кожен з яких доступний через мережу і здатний виконувати певні дії. SOA створює комунікаційне середовище для модулів, які реалізують прикладну бізнес-логіку. Під сервісом розуміють веб-сайт, який надає не просто статичну інформацію, а й дозволяє виконувати певні дії або змінювати світ, наприклад, продавати товар або керувати фізичним пристроєм [1].

Веб-сервіси надають високорівневі абстракції для використання у великомасштабних відкритих середовищах. Веб-сервіс - це автономний додаток, що надає засіб доступу до інформації зовнішнім клієнтам за допомогою набору послуг, що надаються їм.

В [2] Web-сервіс визначається як самокерований модуль, здатний обробляти і управляти даними; Він має можливість взаємодіяти зі своїм оточенням за допомогою повідомлень. Один модуль поєднує в собі функції зберігання та обробки даних; сервіс створюється і впроваджується незалежно від

інших, тому його можна змінювати, не впливаючи на партнерські сервіси; Сервіс має чітко визначені межі і подається ззовні тільки за допомогою повідомлень. Іншими словами, сервіс можна визначити як ІТ-компонент, який забезпечує виконання взаємозалежних функцій і здійснює зв'язок із зовнішньою інфраструктурою за допомогою обміну повідомленнями. Основне завдання — згрупувати функції та забезпечити їх механізмами обміну повідомленнями.

Сервіси можуть бути простими, або «примітивними», в тому сенсі, що вони активують тільки одну доступну в Інтернеті програму, датчик або пристрій, не використовуючи інші веб-служби, і тому між користувачем і службою немає ніякої взаємодії, крім простого дзвінка.

Наприклад, служба, яка повертає поштовий індекс або широту та довготу на вказану адресу. Вони також можуть бути складовими, тобто складатися з декількох примітивів. Такі сервіси, як правило, вимагають взаємодії або діалогу між користувачем і сервісами, тобто користувач може здійснювати вибір або нав'язувати певні умови. Прикладом може служити покупка книги через www.amazon.com: користувач шукає книги за різними критеріями, потім вирішує, чи варто їх купувати, а потім надає кредитну картку або поштову адресу.

Веб-сервіси базуються на трьох основних веб-стандартах [3]: SOAP (Simple Object Access Protocol) - протокол для відправки повідомлень по протоколу HTTP та іншим інтернет-протоколам; WSDL (Web Services Description Language) - мова для опису інтерфейсів програмування веб-сервісів; UDDI (Universal Description, Discovery and Integration) – це стандарт для індексації веб-сервісів.

Коли говорять про концепцію веб-сервісів, часто думають про SOA. Важливо зазначити, що SOA не є синонімом веб-сервісів, так само як і веб-сервіси не є єдиним способом реалізації SOA. Сервіс-орієнтована архітектура - це абстрактне уявлення про реалізацію ІС з використанням сервісів незалежно від технології. Як бачимо, SOA містить елементи об'єктного підходу до побудови інформаційних систем: декомпозиція додатків на окремі функції та інкапсуляція

(сервіси як «чорні скриньки»). Слід підкреслити, що термін «об'єктно-орієнтований» не є коректним по відношенню до SOA — правильніше називати SOA концепцією, яка використовує об'єктний підхід.

З іншого боку, веб-сервіси є лише технологіями, які можуть ефективно реалізувати сервіс-орієнтовану архітектуру. Дана архітектура має такі характеристики:

- є розподіленою - функціональні модулі додатку (системи) можуть бути розподілені по різних обчислювальних системах і здатні взаємодіяти по мережі;
- інтерфейс функціональних модулів такий, що їх використання не залежить від технології або платформи, в рамках якої вони реалізовані;
- можливий динамічний пошук і підключення потрібних модулів;
- архітектура базується на загальноприйнятих галузевих стандартах.

Архітектура, в якій всі функції додатків є незалежними сервісами з чітко визначеними інтерфейсами, які можуть бути викликані в потрібному порядку для формування бізнес-процесів, називається сервіс-орієнтованою.

SOA – це засіб представлення бізнесу як сукупності взаємопов'язаних послуг. [4] Однією з його переваг є гнучкість і простота розробки нових додатків. SOA створює комунікаційне середовище для модулів, які реалізують прикладну бізнес-логіку. Інформація про модулі публікується в такій формі, що їх використання не вимагає знань про рішення і технології, що використовуються в них.

SOA скорочує час і витрати на реалізацію проекту. Компанії, які вже впровадили SOA і планують це зробити, очікують від цієї архітектури наступних переваг:

- швидші та гнучкіші зміни в бізнес-процесах;
- скоротити витрати на ІТ;
- безпечне та надійне обслуговування;
- оперативне впровадження оновлень і додаткових можливостей програмних продуктів;

- безкоштовне використання рішень різних розробників та/або програм, розроблених на замовлення.

Концепція сервіс-орієнтованої архітектури була розроблена при розробці концепції веб-сервісів. Однак існують і інші підходи до реалізації SOA: Java RMI (Sun), CORBA (OMG), DCOM (Microsoft), DCE (Open Group) та ін.

Сервіс-орієнтовані обчислення (SDC) – це обчислювальна парадигма, яка використовує сервіси як фундаментальні елементи для розробки додатків. [5] IDS базується на SOA і забезпечує операції з управління послугами. Розробка IDS – це процес пошуку, вибору та створення сервісів, які відповідають вимогам користувача.

Побудова комерційних додатків на базі веб-сервісів вимагає встановлення відповідного взаємозв'язку між ними. Веб-сервіси підтримують ряд парадигм обміну повідомленнями:

- дистанційний виклик процедур (RPC);
- асинхронний обмін;
- односпрямована передача;
- мовлення;
- видавництво/передплата.

Розробники програмного забезпечення зазвичай надають можливість прив'язки веб-сервісів до своїх програмних систем (СУБД, серверів додатків та інтеграційних брокерів). Для клієнта взаємодія з веб-сервісами може проявлятися в інтерактивній або пакетній формі, що підтримує синхронні та асинхронні моделі зв'язку, а також користувальницький інтерфейс, написаний за допомогою Java, VB, офісних додатків або «товстих» клієнтів СУБД.

Тепер, коли основні технології отримали широке визнання, наступним кроком є визначення передової архітектури для впровадження додаткових технологій (безпека, потік процесів, надійна доставка повідомлень, координація транзакцій). Цим займається W3C Web Services Architecture Working Group.

Оскільки веб-служби вирішують проблему інтеграції в масштабах усього підприємства, постачальники інтеграційних брокерів використовують веб-

служби як будівельні блоки адаптерів та інтеграційних компонентів. Розробники баз даних реалізують доступ до інтерфейсів веб-сервісів безпосередньо до СУБД. Розробники ERP та CRM також підтримують веб-сервіси для інтеграції з іншим програмним забезпеченням. І, нарешті, ряд виробників націлені на реалізацію безпосередньо шару веб-сервісів.

1.3 Компонування Web-сервісів

Можливість компонування часто розглядається як одна з головних переваг веб-сервісів. Верстка веб-сервісу полягає в знаходженні набору атомарних сервісів, необхідних для реалізації запиту користувача, і визначенні порядку їх виконання. [6]

Компонування веб-сервісів схоже на проблему планування, яка була досліджена в штучному інтелекті. Але класичні планувальники не підходять для складання веб-сервісів, тому що вони розраховані на роботу в статичному середовищі (а інтернет-середовище динамічне) і не володіють повною інформацією про систему, з якою працюють (наприклад, про можливості різних сервісів).

Для автоматичного компонування програми повинні вміти вибирати потрібні їм веб-сервіси і комбінувати їх для досягнення поставлених цілей. Таким чином, можна будувати абсолютно нові сервіси, об'єднуючи вже доступні в мережі сервіси. Інформації, що міститься в реєстрі UDDI, недостатньо для автоматичного створення веб-сервісів, оскільки для інтерпретації семантики цих сервісів потрібні людські зусилля. Тому необхідно розробити механізми відображення семантики послуг, запитів користувачів цих сервісів та їх автоматизованого порівняння з урахуванням специфіки предметної області (PrO), що цікавить користувача. Створення веб-сервісів у динамічному середовищі вимагає більше синтаксичної та статичної мета-інформації про них: веб-сервіси в гетерогенному середовищі часто визнаються недійсними або замінюються іншими службами.

Функціональність кожного веб-сервісу визначається його входами, виходами, передумовами та діями, які називаються IOPEs (входи, виходи, передумови та ефекти). Сервіс IOPE міститься в описі WSDL.

За своїм виконанням веб-сервіси можна розділити на атомарні, які не поділяються на підпроцеси і можуть викликатися користувачем через Інтернет безпосередньо, і складові, які мають підпроцеси, пов'язані керуючими конструкціями. Як правило, бізнес-завдання користувача реалізуються складеними сервісами, які можуть бути складені з існуючих атомарних.

Концепція веб-сервісів має на увазі, що окремі сервіси мають певний обмежений функціонал, в той час як рішення більш-менш складних завдань вимагає використання функціоналу декількох сервісів. Тому в ході розвитку архітектури веб-сервісів з'явилися такі поняття, як композиція і потік, оркестровка і хореографія. Вони відображають взаємодію служб і послідовність їх виконання. Додатки, створені за допомогою веб-сервісів, вважаються заснованими на робочому процесі.

У добре спроектованому веб-сервісі всі процеси атомарні. Для інтеграції бізнес-процесів повинні бути дотримані наступні ключові вимоги [7-8]:

1. Послідовність процесів. При використанні веб-сервісу можна викликати операції в будь-якій послідовності. Наприклад, клієнт може отримати доступ до операції «грошовий переказ» перед викликом операції «оформлення замовлення».

2. Взаємозв'язок повідомлень. Коли бізнес-процеси взаємодіють, для обміну повідомленнями використовується якийсь загальний елемент даних. WSDL не підтримує стан між операціями (якщо ордер змінився, його необхідно розмістити знову).

3. Одиниця роботи. Для того, щоб веб-сервіс був частиною довготривалої розсилки, важливо вміти чітко встановлювати час фактичного початку та закінчення обміну, а також визначати, яку частину цього процесу можна вважати транзакцією. Наприклад, операції з оформлення та підтвердження замовлення на купівлю, а також процес виведення коштів повинні виконуватися як єдина

одиниця роботи: Веб-сервіс повинен виконати всі три операції успішно, інакше користувачеві доведеться відновлювати операції до стану, що відповідає початку їх виконання. Ці операції не можуть бути використані для розподілених транзакцій, виражених за допомогою WSDL.

4. Обробка винятків. Наприклад, якщо клієнт не підтверджує ордер на купівлю або продаж протягом хвилини, веб-сервіс генерує виняток тайм-ауту.

5. Контекст. Коли бізнес-процеси взаємодіють, важливо, щоб вони мали загальний контекст для обміну інформацією, яким може бути набір декларацій, виняткових подій або властивостей транзакцій. WSDL не має можливості для процесів запускатися в контексті.

Більшість складених служб створюються вручну з використанням описів на основі WSDL. Для автоматичної перелінковки програми повинні вміти підбирати необхідні веб-сервіси і комбінувати їх. У цьому випадку результатом має стати прийнятне вирішення проблеми. Інформації, що міститься в UDDI (Universal Description, Discovery, and Integration), недостатньо для автоматичного створення сервісів, оскільки вона не дозволяє інтерпретувати їх семантику. Ні WSDL, ні UDDI не дають зрозуміти програмі, що робить сервіс з точки зору клієнта. Саме тому так важливі механізми відображення семантики послуг та їх автоматизованого порівняння з семантикою запитів клієнтів. Задачі верстки можуть бути вирішені шляхом пов'язування параметрів сервісу з предметно-специфічними поняттями і їх смисловим обґрунтуванням.

1.4 Інтелектуальні веб-сервіси

Інтелектуальні веб-сервіси є складною та розвинутою категорією веб-технологій, які використовують штучний інтелект (ШІ) та інші передові технології для надання розумних функцій і можливостей. Ось деякі особливості інтелектуальних веб-сервісів [9-11]:

1. Інтелектуальні веб-сервіси можуть використовувати технології обробки природної мови (NLP) та машинного навчання для автоматичного розпізнавання та розуміння текстової та голосової інформації, що надходить від користувачів.

2. Інтелектуальні веб-сервіси можуть аналізувати дані про користувачів, їхні уподобання та поведінку, щоб надавати персоналізовані рекомендації, забезпечуючи кращий користувацький досвід.

3. Такі веб-сервіси можуть використовувати алгоритми машинного навчання для аналізу даних та прийняття автоматичних рішень на основі цієї аналітики. Наприклад, системи управління запасами можуть автоматично поповнювати запаси на основі прогнозу попиту.

4. Інтелектуальні веб-сервіси можуть використовувати продуктивні алгоритми пошуку, які враховують контекст, семантику та інші фактори, щоб надавати користувачам більш точні та релевантні результати пошуку.

5. Сервіси можуть автоматично аналізувати великі обсяги даних та генерувати динамічні звіти та візуалізації для допомоги в прийнятті рішень.

6. Деякі інтелектуальні сервіси можуть навчатися на основі взаємодії з користувачами та змінюватися відповідно до нової інформації або змінних умов.

7. Інтелектуальні веб-сервіси можуть легко інтегруватися з іншими інформаційними системами та джерелами даних для отримання ширшого обсягу інформації та функціональності.

Ці особливості роблять інтелектуальні веб-сервіси потужними інструментами для надання високорівневої функціональності та автоматизації веб-додатків.

Онтології є загальним засобом представлення семантики послуг, тобто інтегральних структур понять певної предметної області в рамках єдиної системи взаємопов'язаних компонентів. Це формалізований опис термінів, їх визначень та ознак, а також того, як вони пов'язані один з одним.

Онтології полегшують складання послуг. Наявність чітких знань про Про, до якого належить служба, дозволяє формулювати запити в контекстно-залежний спосіб і вести переговори про можливості цього сервісу. Ви можете

автоматизувати семантичну схожість запиту та опису послуги, незважаючи на синтаксичні відмінності між ними. Для цього запит узгоджується на основі ієрархії понять ПрО, відображених в онтології.

Для семантичного обґрунтування параметрів веб-сервісів використовуються онтології різних рівнів: додатки, ПрО, верхнього рівня. Наведемо основні класифікації онтологій для явного опису семантики джерел інформації:

1. Підходи, засновані на єдиній онтології, використовують одну глобальну онтологію зі спільним словником для конкретизації семантики. Всі джерела інформації пов'язані з глобальною онтологією. Такі підходи використовуються в тих випадках, коли всі джерела інформації, які необхідно об'єднати, використовують дуже схожі погляди на предметну область.

2. У підходах, заснованих на множинних онтологіях, кожне джерело інформації описується власною онтологією. Кожна з цих прикладних онтологій може бути комбінацією декількох інших, але ви не можете очікувати, що кілька прикладних онтологій використовують одну і ту ж лексику. А відсутність загального словника ускладнює їх порівняння.

3. Гібридні підходи схожі на підходи, засновані на множинних онтологіях, в тому, що семантика кожного вихідного тексту описується власною онтологією, але для того, щоб зробити локальні онтології порівнянними між собою, формується загальний глобальний словник.

На сьогоднішній день найбільш поширеними є підходи, засновані на єдиній онтології.

Появу сервісно-орієнтованого програмного забезпечення часто називають черговою революцією в індустрії програмного забезпечення. Розробники постійно нарощують можливості веб-сервісів, переходячи від простого обміну повідомленнями до повнофункціональних додатків.

Зробимо деякі висновки з аналізу предметної області. Програмне забезпечення, яке треба розробити в даній роботі, відноситься до веб-сервісів і буде виконувати досить обмежений клас задач - оцінювати методами машинного

навчання різні режими котлового обладнання та намагатись знаходити оптимальні параметри управління котлом для виконання поставлених користувачем задач.

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ

Якщо деталізувати завдання на кваліфікаційну роботу, то виявиться, що фактично потрібно створити сервіс, який буде забезпечувати обмін інформацією між пристроями з мінімальною участю людини (налаштування, тестування, перевірки, збір звітів тощо). Це завдання цілком можна віднести до умовного класу інформаційних систем, яке має назву інтернету речей. Зробимо короткий опис цієї концепції.

2.1 Інтернет речей

Інтернет речей (Internet of Things або IoT) є однією з найсучасніших і найважливіших концепцій у сфері інформаційних технологій. [12-13] Це явище змінило спосіб взаємодії фізичних об'єктів, створивши нові можливості для автоматизації, моніторингу та управління в різних галузях життя та економіки. IoT дозволяє об'єктам спілкуватися між собою та з іншими системами через Інтернет, що відкриває безліч нових можливостей для бізнесу та побуту.

Інтернет речей можна визначити як мережу фізичних об'єктів, які мають вбудовані технології для взаємодії та обміну даними через Інтернет. Ці об'єкти можуть бути різними пристроями: від простих побутових приладів до складних промислових систем. Основною метою IoT є створення інтегрованих і взаємопов'язаних систем, що забезпечують нові рівні автоматизації, ефективності та зручності.

Ключовими компонентами IoT є пристрої та сенсори, комунікаційні мережі, обробка даних та аналітика, інтерфейси користувача та системи безпеки. Пристрої та сенсори є фундаментом IoT, оскільки вони забезпечують збір даних про навколишнє середовище. Сенсори вимірюють різні параметри, такі як температура, вологість, світло, рух тощо, і передають ці дані на обробку.

Виконавчі пристрої або актори, своєю чергою, здійснюють певні дії у відповідь на отримані дані, наприклад, увімкнення або вимкнення приладів.

Комунікаційні мережі забезпечують зв'язок між пристроями та передавання даних. Різні мережеві протоколи, такі як Wi-Fi, Bluetooth, Zigbee та інші, використовуються для забезпечення зв'язку в локальних і глобальних мережах. Гейтвеї є пристроями, що забезпечують зв'язок між локальними мережами IoT та Інтернетом, а також здійснюють попередню обробку даних.

Обробка даних та аналітика є важливими аспектами IoT, оскільки саме вони забезпечують зберігання, обробку і аналіз великих обсягів даних. Хмарні сервіси відіграють ключову роль у цьому процесі, оскільки вони надають платформи для зберігання та обробки даних. Технології машинного навчання і штучного інтелекту використовуються для аналізу даних, виявлення закономірностей та прийняття рішень на основі отриманої інформації.

Інтерфейси користувача дозволяють користувачам взаємодіяти з IoT системами. Це можуть бути веб-інтерфейси або мобільні додатки, що надають можливість отримувати інформацію в режимі реального часу, керувати пристроями та налаштовувати їх. Нарешті, безпека є критично важливим аспектом IoT, оскільки вона забезпечує захист даних від несанкціонованого доступу та гарантує безпечний доступ до пристроїв.

IoT має широкий спектр застосувань у різних галузях, включаючи розумні дома, інтелектуальні міста, промисловість, охорону здоров'я та сільське господарство. У сфері розумних домів IoT дозволяє автоматизувати управління освітленням, клімат-контролем, системами безпеки та побутовими приладами. Це забезпечує зручність, економію енергії та підвищення рівня безпеки.

Інтелектуальні міста використовують IoT для моніторингу та управління міським транспортом, вуличним освітленням, системами водопостачання та енергоспоживання. Це дозволяє підвищити ефективність міської інфраструктури та покращити якість життя мешканців.

У промисловості IoT допомагає оптимізувати виробничі процеси, моніторити стан обладнання та здійснювати предиктивне технічне

обслуговування. Це дозволяє зменшити витрати на обслуговування та підвищити продуктивність. У сфері охорони здоров'я IoT використовується для віддаленого моніторингу пацієнтів, управління медичними приладами та автоматизації процесів у лікарнях. Це забезпечує покращення якості медичних послуг та зниження витрат на охорону здоров'я.

У сільському господарстві IoT дозволяє моніторити умови вирощування, керувати іригаційними системами та автоматизувати процеси вирощування і збору урожаю. Це сприяє підвищенню врожайності та ефективності сільськогосподарських операцій.

Незважаючи на численні переваги, IoT також стикається з низкою викликів та проблем. Перш за все, це питання безпеки. Велика кількість підключених пристроїв створює значні ризики кібербезпеки, оскільки кожен пристрій може стати потенційною точкою входу для зловмисників. Важливо забезпечити захист даних і надійність систем IoT, щоб запобігти несанкціонованому доступу та втраті даних.

Іншою важливою проблемою є питання сумісності та стандартизації. Існує велика кількість різних пристроїв і платформ IoT, що використовують різні протоколи і стандарти зв'язку. Це може створити труднощі при інтеграції різних систем і забезпеченні їх сумісності. Важливо розробляти загальні стандарти та протоколи, які забезпечать ефективну взаємодію між різними компонентами IoT.

Збір та обробка великих обсягів даних також є серйозним викликом. IoT генерує величезну кількість даних, які потрібно зберігати, обробляти та аналізувати. Це вимагає потужних обчислювальних ресурсів та ефективних методів обробки даних. Хмарні сервіси та технології великих даних є ключовими компонентами для вирішення цих задач.

Інше питання стосується енергоефективності. Багато пристроїв IoT працюють на батареях і вимагають тривалого часу автономної роботи. Важливо розробляти енергоефективні технології та алгоритми, що забезпечують тривалу роботу пристроїв без необхідності частого обслуговування.

Майбутнє IoT виглядає дуже перспективним, з багатьма новими можливостями та напрямками розвитку. Одним із ключових напрямків є розвиток 5G технологій, що забезпечить швидкісний зв'язок та зниження затримок передачі даних. Це дозволить реалізувати нові сценарії використання IoT, такі як автономні транспортні засоби, розумні міста та інтелектуальні промислові системи.

Іншим важливим напрямком є інтеграція штучного інтелекту та машинного навчання у IoT системи. Це дозволить здійснювати більш точний аналіз даних, виявлення закономірностей та автоматичне прийняття рішень на основі отриманої інформації. Такі системи зможуть самостійно адаптуватися до змін умов і покращувати свою ефективність з часом.

Розвиток технологій безпеки та захисту даних також буде важливим напрямком. Зокрема, будуть розроблятися нові методи аутентифікації та шифрування, що забезпечать високий рівень захисту IoT систем від кіберзагроз.

Зростання кількості підключених пристроїв та збільшення обсягів даних вимагатимуть нових підходів до обробки та зберігання інформації. Розвиток технологій великих даних, розподілених обчислень та хмарних сервісів стане ключовим фактором у вирішенні цих задач.

У сфері розумних домів і міст буде продовжуватися розвиток систем автоматизації та управління, що забезпечать підвищення комфорту, безпеки та ефективності використання ресурсів. Нові технології, такі як інтелектуальні системи освітлення, клімат-контролю та безпеки, будуть активно впроваджуватися в побут і міську інфраструктуру.

У промисловості IoT сприятиме подальшій автоматизації виробничих процесів, зниженню витрат і підвищенню продуктивності. Інтелектуальні системи моніторингу та управління дозволять оптимізувати роботу обладнання та здійснювати предиктивне технічне обслуговування, що зменшить ризики простоїв і поломок.

В галузі охорони здоров'я IoT продовжуватиме впроваджуватися для покращення якості медичних послуг та зниження витрат. Віддалений моніторинг

стану пацієнтів, інтелектуальні медичні прилади та системи автоматизації медичних процесів стануть невід'ємною частиною сучасної медицини.

У сільському господарстві IoT сприятиме подальшому розвитку технологій точного землеробства, що дозволять підвищити врожайність та ефективність використання ресурсів. Інтелектуальні системи моніторингу умов вирощування, управління іригаційними системами та автоматизація сільськогосподарських операцій стануть ключовими елементами успішного розвитку аграрного сектору.

Таким чином, Інтернет речей є однією з найперспективніших і найважливіших технологій сучасності, яка відкриває нові можливості для автоматизації, моніторингу та управління у різних галузях життя та економіки. Незважаючи на численні виклики, такі як безпека, сумісність та енергоефективність, IoT має величезний потенціал для покращення якості життя, підвищення ефективності та створення нових бізнес-моделей. Майбутнє IoT обіцяє бути ще більш захоплюючим, з новими технологіями та інноваціями, які змінять наш світ на краще.

2.2 Етапи створення інтелектуального сервісу

Створення інтелектуального сервісу для Інтернету речей (IoT) є складним і багатоступеневим процесом, який вимагає ретельного планування, проектування, розробки, тестування та впровадження. [14] Нижче наведені основні етапи, які зазвичай включає цей процес.

На першому етапі визначаються цілі та вимоги до сервісу. Необхідно з'ясувати, які завдання має вирішувати сервіс, які дані він повинен обробляти, які функції виконувати. Проведення аналізу ринку і дослідження існуючих рішень допоможе зрозуміти, які функціональні можливості та технології вже використовуються і що можна поліпшити.

На основі зібраних вимог створюється загальна архітектура системи. Це включає визначення основних компонентів системи, їх взаємодію, вибір

технологій і платформ. Важливо врахувати аспекти масштабованості, безпеки, надійності та продуктивності. Зазвичай архітектура складається з декількох шарів, включаючи сенсорний рівень, рівень передачі даних, рівень обробки та аналізу даних, а також рівень інтерфейсу користувача.

Наступний крок - вибір апаратного забезпечення та сенсорів. На цьому етапі вибираються сенсори та пристрої, які будуть використовуватися для збору даних. Важливо врахувати характеристики сенсорів, такі як точність, надійність, енергоефективність та сумісність з іншими компонентами системи. Також необхідно забезпечити можливість підключення сенсорів до мережі через відповідні протоколи (Wi-Fi, Bluetooth тощо).

Далі йде розробка програмного забезпечення. Цей етап включає створення програмного забезпечення для всіх компонентів системи. Необхідно створити програму для збору та передачі даних з сенсорів, програму для обробки та аналізу даних, а також інтерфейс користувача. Важливо забезпечити високу якість коду, безпеку та ефективність роботи програмного забезпечення. На цьому етапі можуть використовуватися різні мови програмування та платформи залежно від вимог проекту.

Більшість сучасних IoT-сервісів використовують хмарні платформи для зберігання, обробки та аналізу даних. Необхідно вибрати відповідну хмарну платформу (наприклад, AWS, Azure, Google Cloud) та налаштувати інтеграцію з нею. Це включає налаштування баз даних, сервісів обробки даних, а також налаштування безпеки та доступу до даних.

Обробка та аналіз даних є ключовими етапами створення інтелектуального сервісу. Для цього використовуються технології машинного навчання та штучного інтелекту, які дозволяють здійснювати аналіз великих обсягів даних, виявляти закономірності та робити прогнози. Важливо розробити алгоритми та моделі, які забезпечать високу точність та ефективність аналізу.

Після цього треба розробити інтерфейс користувача, який дозволяє користувачам взаємодіяти з IoT-сервісом, отримувати інформацію в режимі реального часу, керувати пристроями та налаштовувати їх. Необхідно створити

зручний та інтуїтивно зрозумілий інтерфейс, який забезпечить легкість використання сервісу. Інтерфейс може бути реалізований у вигляді веб-додатку або мобільного додатку.

Перед впровадженням системи необхідно провести ретельне тестування всіх її компонентів. Це включає функціональне тестування, тестування продуктивності, безпеки та надійності. Важливо виявити та усунути всі можливі помилки та недоліки, щоб забезпечити стабільну та ефективну роботу сервісу.

Після завершення тестування сервіс можна впроваджувати у реальне середовище. На цьому етапі важливо забезпечити безперервний моніторинг роботи системи, щоб вчасно виявляти та усувати можливі проблеми. Необхідно також забезпечити підтримку та оновлення системи для подальшого вдосконалення та розширення її функціональних можливостей.

Після впровадження сервісу важливо забезпечити його подальшу підтримку та оновлення. Це включає виправлення помилок, додавання нових функцій, оптимізацію продуктивності та забезпечення безпеки. Регулярні оновлення допоможуть підтримувати сервіс на високому рівні та відповідати змінюваним вимогам користувачів і ринку.

Таким чином, створення інтелектуального сервісу для Інтернету речей є комплексним процесом, що включає аналіз вимог, проектування архітектури, вибір апаратного забезпечення, розробку програмного забезпечення, інтеграцію з хмарними сервісами, обробку даних, розробку інтерфейсу користувача, тестування, впровадження та подальшу підтримку. Кожен з цих етапів вимагає ретельного планування та виконання, щоб забезпечити успішне створення та функціонування IoT-сервісу, що відповідає вимогам користувачів та ринку.

2.3 Порівняння та вибір технології зв'язку IoT

Вибір технології зв'язку є однією з ключових задач при проектуванні систем Інтернету речей (IoT). [15] Різні технології мають свої переваги та

недоліки, і їх вибір залежить від конкретних вимог проекту. Розглянемо детально різні технології зв'язку, їх характеристики та підходи до вибору.

LPWAN - це категорія технологій зв'язку, що характеризуються низьким енергоспоживанням і широким покриттям. До LPWAN належать такі технології, як LoRaWAN і NB-IoT.

LoRaWAN є однією з найбільш популярних LPWAN-технологій. Вона забезпечує велику дальність передачі даних, що робить її ідеальною для моніторингу великих територій, таких як сільське господарство чи міські інфраструктури. LoRaWAN використовує незахищені частоти, що дозволяє уникнути витрат на ліцензії, але може спричинити проблеми з перешкодами та перевантаженням мережі.

NB-IoT (Narrowband IoT) також забезпечує велике покриття і низьке енергоспоживання. На відміну від LoRaWAN, NB-IoT працює на ліцензованих частотах, що забезпечує більш стабільний і захищений зв'язок. Ця технологія ідеально підходить для застосувань, що вимагають надійності та безпеки передачі даних, наприклад, у промислових умовах.

Wi-Fi є поширеною технологією зв'язку, що забезпечує високу швидкість передачі даних і зручність підключення. Вона використовується в багатьох IoT-застосуваннях, особливо в умовах, де доступ до електромережі не є проблемою. Висока швидкість передачі робить Wi-Fi ідеальним для додатків, що потребують великої пропускної здатності, таких як відеоспостереження чи мультимедійні сервіси. Однак високе енергоспоживання обмежує використання Wi-Fi для автономних пристроїв, що живляться від батарей.

Bluetooth - це технологія короткого діапазону, що використовується для з'єднання пристроїв на невеликій відстані. BLE, або Bluetooth Low Energy, є його енергоефективною версією. BLE ідеально підходить для пристроїв, що потребують періодичної передачі невеликих обсягів даних, наприклад, фітнес-трекери або датчики здоров'я. Низьке енергоспоживання дозволяє використовувати BLE для пристроїв з тривалим терміном роботи від батареї. Однак обмежений діапазон робить його непридатним для великих мереж.

Zigbee - це технологія зв'язку, що використовує мережеву топологію "mesh" для забезпечення надійного зв'язку між пристроями. Вона добре підходить для додатків домашньої автоматизації, де важливою є стабільність та надійність з'єднань. Завдяки підтримці "mesh" топології, Zigbee дозволяє створювати мережі з великою кількістю пристроїв, де кожен пристрій може ретранслювати дані інших пристроїв, що збільшує загальний радіус дії мережі. Енергоспоживання Zigbee є низьким, що робить його придатним для автономних пристроїв.

Ще один варіант (Z-Wave) дуже схожий на Zigbee в тому, що це малопотужні мережі, які працюють на протоколі mesh-мережі і призначені для обміну даними на коротких і середніх відстанях. ZigBee використовує стандарт персональної мережі IEEE 802.15.4 для підключення пристроїв. Завдяки 128-бітному симетричному шифруванню AES дані під час передавання захищені. Він працює на частоті 2,4 ГГц, що досить непогано, але може створювати перешкоди для інших пристроїв, що працюють на цій частоті. Z-Wave також використовує 128-бітне симетричне шифрування AES. Діапазон радіочастот у Європі (та Україні) становить 868,42 МГц. На такій частоті є менше шансів перешкод. Z-Wave також має трохи більшу відстань зв'язку (до 100 м).

Технології мобільного зв'язку, такі як 2G, 3G, 4G, і 5G, забезпечують широке покриття і високу швидкість передачі даних. Ці технології ідеально підходять для мобільних IoT-застосувань, де пристрої можуть переміщуватися між різними зонами покриття. Наприклад, 4G і 5G забезпечують високу пропускну здатність та низьку затримку, що робить їх придатними для критичних додатків у режимі реального часу. Однак високе енергоспоживання і вартість використання мобільних мереж можуть бути обмежуючими факторами для деяких проектів.

Sigfox - це ще одна технологія LPWAN, яка забезпечує дуже низьке енергоспоживання і велике покриття. Вона використовується для передачі невеликих обсягів даних на великі відстані, що робить її придатною для моніторингу віддалених об'єктів, таких як датчики в екологічному моніторингу

чи розумних містах. Однією з переваг Sigfox є низька вартість обслуговування, оскільки вона використовує незахищені частоти. Однак обмежена пропускна здатність може бути недоліком для додатків, що вимагають передавання великих обсягів даних.

Отже, вибір технології зв'язку для IoT залежить від багатьох факторів, включаючи тип додатку, вимоги до енергоспоживання, діапазон покриття, швидкість передачі даних, безпеку та вартість. Наприклад, для додатків домашньої автоматизації, де важлива енергоефективність та надійність з'єднань, Zigbee може бути оптимальним вибором. Для промислових додатків, де важлива надійність та безпека, доцільно використовувати NB-IoT або LTE-M. Для додатків з високими вимогами до пропускної здатності та низькою затримкою, таких як відеоспостереження, краще використовувати Wi-Fi або 5G.

Кожна з технологій підходить для певних типів додатків і сценаріїв використання. Тому важливо ретельно аналізувати потреби вашого IoT-проекту та вибирати найбільш підходящу технологію, що забезпечить ефективну та надійну роботу вашої системи.

З огляду на те, що хоча веб-сервіс не залежить від технологій підключення кінцевих пристроїв, але йому необхідно їх враховувати, то на погляд автора найбільш прийнятним для приватних господарств або невеликих промислових об'єктів вважається вибір технологій Zigbee або Z-Wave. В підсумку було вирішено зупинитись на останній.

2.4 Вибір інструментарію реалізації системи

Реалізація інтелектуального веб-сервісу вимагає використання широкого спектру інструментів і технологій, які забезпечують інтеграцію, аналіз даних, машинне навчання та надання користувацького інтерфейсу. Спочатку розглянемо вимоги до такого інструментарію.

Для побудови інтелектуального веб-сервісу необхідно використовувати платформу або середовище розробки, яке підтримує масштабованість та високу продуктивність.

Популярними виборами є платформи з підтримкою мікросервісної архітектури, що дозволяє розробляти модульні та легко масштабовані системи. Ці платформи надають можливість розподіляти різні частини системи на окремі сервіси, що взаємодіють між собою через API.

Обробка та аналіз даних є центральним аспектом інтелектуальних веб-сервісів. Для цього використовуються інструменти та бібліотеки машинного навчання, які забезпечують можливість побудови моделей для прогнозування, класифікації та кластеризації даних. Ці моделі можуть бути інтегровані в веб-сервіс для надання користувачам інтелектуальних рекомендацій або аналізу поведінкових даних.

Одним з ключових аспектів є забезпечення безпеки та конфіденційності даних користувачів. Для цього використовуються методи шифрування, управління доступом та протоколи безпечного з'єднання. Важливим є також забезпечення відповідності сервісу з вимогами щодо захисту персональних даних.

Користувацький інтерфейс є важливою складовою, що визначає зручність використання веб-сервісу. Для його реалізації використовуються сучасні фреймворки, які забезпечують створення адаптивних, інтерактивних та зручних для користувачів веб-додатків. Ці фреймворки підтримують інтеграцію з бекенд-сервісами, що дозволяє відображати динамічні дані та забезпечувати інтерактивну взаємодію з користувачами.

Для управління життєвим циклом розробки та розгортання інтелектуальних веб-сервісів використовуються системи безперервної інтеграції та розгортання. Вони забезпечують автоматизацію процесів тестування, збірки та деплою, що сприяє швидкому випуску нових версій та виправленню помилок.

Моніторинг та аналітика є невід'ємною частиною експлуатації інтелектуальних веб-сервісів. Вони забезпечують збір та аналіз метрик

продуктивності, виявлення аномалій та надання інформації про використання сервісу користувачами. Це дозволяє своєчасно реагувати на проблеми та оптимізувати роботу сервісу.

Інтелектуальні веб-сервіси часто інтегруються з іншими системами та сервісами через API. Це забезпечує можливість розширення функціональності та надання більш комплексних рішень користувачам. Інтеграція з іншими сервісами може включати обмін даними, використання зовнішніх обчислювальних ресурсів або підключення до сторонніх аналітичних інструментів.

В результаті аналізу та співставлення різних мов програмування, платформ та підходів автор вирішив зупинитись на алгоритмічній мові Python, платформі Anaconda та відповідних пакетах машинного навчання та інших фреймворках, необхідних для створення інтелектуального сервісу.

Треба додати, що мова Python є однією з найпопулярніших мов обробки даних і машинного навчання, має величезну кількість бібліотек з відкритим кодом, підтримується розгалуженою спільнотою та може бути використаною навіть розробниками, які не мають дуже великого досвіду. Все це додатково вплинуло на вибір автора роботи.

Таким чином, в даному розділі розглянуті особливості концепції Інтернету речей, зроблено огляд етапів розробки інтелектуального сервісу, обрані технологія зв'язку між пристроями та інструментарій побудови програмної частини системи.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Моделювання роботи твердопаливного котла

Твердопаливні котли відрізняються від інших пристроїв теплогенерації тим, що мають велику інерційність. Газові, електричні та котли на рідкому паливі - потужність всіх цих приладів можна досить легко регулювати (навіть до повного виключення), просто обмежуючи подачу палива або електроенергії. З твердопаливними котлами так робити неможливо.

Весь необхідний об'єм палива, як правило, вже знаходиться в котлі. І навіть при умові неповного його займання - все одно ніхто його не витягує з топки. Обмежити можна тільки другий компонент теплової реакції - повітря. Завдяки різному додатковому обладнанню (заслінкам, клапанам, повітряним турбінам тощо) можна регулювати або обмежувати потік повітря до камери згоряння.

Однак, теплова реакція, навіть при повному обмеженні повітря, все одно продовжиться. В умовах нестачі кисню змінюються протікання реакції та хімічний склад димових газів. Паливо згоряє неповністю, замість вуглекислого газу CO_2 на виході маємо більше угарного газу CO . Крім того, температура розпеченого палива досягає декількох сотень градусів, і завдяки тому, що воно залишається в котлі - продовжує нагрівати теплоносій. Все це накладає свої обмеження та особливості управління роботою твердопаливних котлів.

Розглянемо усереднений графік температурного режиму такого котла в якості прикладу (рис. 3.1).

На цьому графіку умовна нульова відмітка температури відповідає температурі котла, коли вмикається повітряна турбіна. Фактично - це може бути температура 60-70 °C, як рекомендують спеціалісти по твердопаливним котлам. Цьому моменту на графіку відповідає точка 1. До цієї точки котел працює в режимі охолодження (ділянка графіка зліва від точки 1).

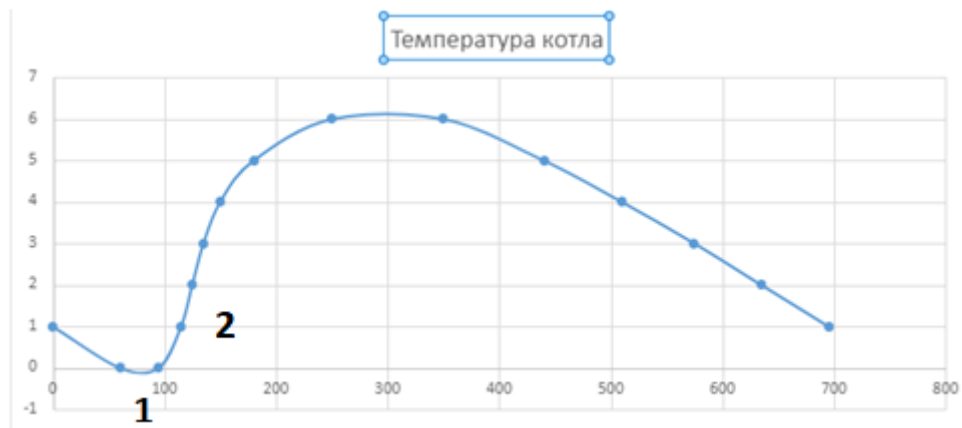


Рисунок 3.1 - Температурний режим твердопаливного котла

Після включення турбіни котел починає нарощувати свою потужність. Падіння температури сповільнюється, потім зупиняється, і починається її зростання. Контролер котла відстежує температуру теплоносія і в момент початку швидкого нагріву (точка 2) виключає повітряну турбіну.

Завдяки інерційності котла потужність нагріву різко не змінюється, і температура продовжує зростати, спочатку швидко, потім повільніше, поки не досягне максимуму. Після цього котел починає повільно втрачати температуру.

Як бачимо, з приблизно 700 с циклу - всього тільки приблизно 60 с (або фактично менше, ніж 10%) турбіна включена, а котел розігрівається. Інші 90% часу - він працює по інерції.

В залежності від наявних потреб системи опалення потужності котла може виявитись забагато, особливо в осінній або весняний період. І тоді перерегулювання (відстань від точки 2 до максимуму) може виявитись набагато більшою, ніж допустимо, що означає можливість перегріву.

Щоб цього не допустити - можна програмним способом вирахувати час виключення турбіни, не очікуючи, поки температура досягне точки 2. Від точки перетину нульової осі знизу вгору до точки 2 - приблизно 20 с. Спробуємо змодельовати ситуації, коли ми вимикаємо турбіну раніше. Ми зробимо це з кроком 1 с і отримаємо 20 різних графіків температурних режимів.

Результати такого моделювання наведені на рис. 3.2.

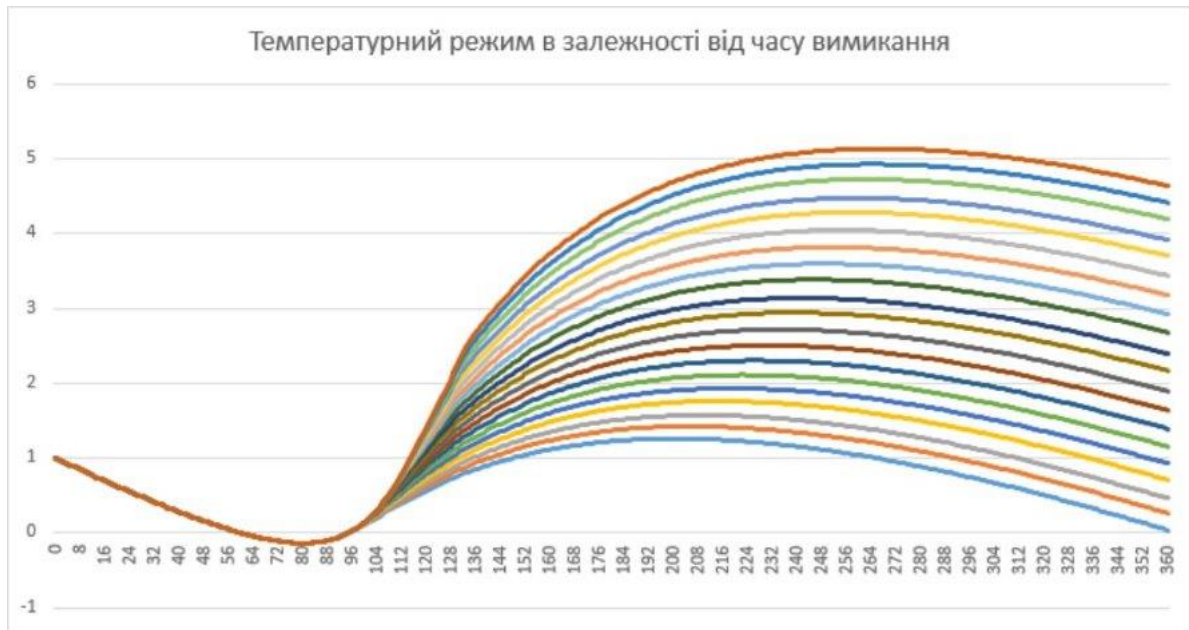


Рисунок 3.2 - Результати моделювання завчасного вимикання турбіни

Завдяки цьому моделюванню ми маємо дані про максимальну температуру в кожному випадку.

Тут важливо врахувати те, що параметри системи опалення та теплогенераторні здібності котла мають випадкову природу і залежать від дуже багатьох факторів (щільності та вологості палива, температурного режиму зовнішнього середовища, теплових втрат опалювального об'єкту і багатьох інших). Всі ці фактори неможливо врахувати і побудувати математичну модель з повним описом. В такому випадку здійснюють або імітаційне моделювання, або приймають рішення, які мають достатній рівень ймовірності, щодо отримання необхідних характеристик.

Є ще один шлях - спробувати вирішити задачу за допомогою машинного навчання, коли система відстежує деяку статистику, а потім намагається по ній збудувати модель штучного інтелекту, яка подібна до об'єкта, який спостерігає дослідник. Зробимо так і ми.

Даних для машинного навчання з 20 графіків явно недостатньо. Крім того, це моделювання робилось на усереднених даних, тому треба в них ввести ще елемент випадковості. Було вирішено зробити масштабування отриманих 20 графіків в двох вимірах: часі та температурі. Масштабування передбачає зміни

показників в той чи інший бік (збільшення або зменшення). Було прийнято рішення здійснити таке масштабування на 25% в кожну сторону і отримати по 10 відповідних значень в кожен бік. Так ми отримаємо по 20 додаткових значень параметрів на кожну вісь. Це означає, що загалом ми отримаємо $20 \times 20 \times 20 = 8000$ варіантів даних. Це цілком достатньо для машинного навчання.

Щоб інтелектуальна модель не перенавчилася на штучно сформованому наборі даних - ми будемо використовувати його не весь, а тільки випадковим чином відібрані 1000 варіантів. З цього набору 70% (700 випадкових варіантів) будемо використовувати для навчання моделі. А 30% (300 варіантів) - для тестування або валідації, щоб перевірити роботу моделі на даних, які вона “не бачила”. Таким чином, зможемо впевнитись, що модель підбрала необхідні параметри і працює адекватно.

Що стосується кількості даних для навчання моделі, коли вона працює в реальному часі - то тут можна приблизно вирахувати і цей показник.

Припустимо, що котел працює в приблизно такому режимі, який ми тільки що побачили. Тоді час циклу - приблизно 11-12 хвилин. Тобто за 1 годину ми отримаємо дані про приблизно 5 циклів, а за добу - приблизно за $24 \times 5 = 120$ циклів. Тобто, буквально за 2-3 доби початку роботи котла модель вже буде мати достатньо даних (декілька сотень) для свого навчання. Після цього в модель можна вже вводити додаткові ознаки (наприклад, час доби, день тижня і так далі) - для того, щоб більш адекватно моделювати ситуацію.

3.2 Реалізація інтелектуальної моделі та оцінка її роботи

Розробимо інтелектуальну модель в середовищі Python. Спочатку імпортуємо необхідні бібліотеки (рис. 3.3). Це відповідні бібліотеки для роботи за даними - numpy та pandas, бібліотека машинного навчання sklearn, а також графічна бібліотека для візуалізації seaborn.

```

import numpy as np
import pandas as pd
import sklearn

import seaborn as sns

from sklearn.linear_model import (
    Ridge,
    Lasso,
    LinearRegression,
)

from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline

from matplotlib import pyplot as plt

```

Рисунок 3.3 - Імпорт необхідних бібліотек

Вхідними даними для інтелектуальної моделі будуть такі:

- швидкість втрат температури в режимі охолодження (час втрати 1°C - відстань від часу 0 до точки 1 на графіку рис.3.3);
- час знаходження температури нижче умовного “нульового” рівня (відстань від точки 1 до точки перетину нульового рівня знизу вгору);
- швидкість нагріву або час спрацювання сигналу на відключення турбіни (відстань від точки перетину нульового рівня знизу вгору до точки 2).

Параметр, який ми намагаємось моделювати (або цільова змінна) - температура максимуму.

В якості моделі візьмемо регресійну модель з бібліотеки sklearn.

Трьох ознак навряд чи буде достатньо для отримання повноцінної моделі. Крім того, регресійна модель працює з лінійними залежностями і погано

відноситься до різного роду нелінійностей. Щоб зменшити негативність цього недоліку можна застосувати різні підходи. Один з них - штучне утворення нових ознак з нелінійностями - так зване створення поліноміальних ознак. Бібліотека `sklearn` має відповідні інструменти для цього.

Оберемо максимальну степінь поліному - 3. В звичайній лінійній моделі маємо 3 ознаки та вільний член - всього 4 ознаки. В поліноміальній моделі 3 степені - маємо вже 19 ознак (3 лінійні ознаки або першої степені, 6 ознак другої степені, 9 ознак третьої степені та вільний член). Ми отримали 15 додаткових нелінійних ознак, цього мусить бути цілком достатньо.

Крім того, застосуємо різні види регресійних моделей - Ridge, Lasso, LinearRegression. Який виявиться найкращим - будемо з'ясовувати в ході підбору гіперпараметрів.

Метод оцінки - метод найменших квадратів або середня помилка в оцінці від тих значень, що будемо мати в навчальній вибірці.

Для розуміння того, що модель дійсно навчається, нам потрібно з чимось порівнювати її результати. Спочатку спробуємо усереднити показники і просто давати середню відповідь. Оцінимо її середню помилку (рис. 3.4).

```
# constant prediction
mean_y = np.repeat(np.mean(y), y.shape[0])
constant_error = mean_squared_error(y, mean_y)
constant_error
```

6.1917104

Рисунок 3.4 - Середня помилка без навчання

Після цього застосуємо машинне навчання і порівняємо результати роботи моделі (рис. 3.5)

```
lr = LinearRegression()
lr_error = mean_squared_error(y_test_num, lr.predict(X_test_num))
lr_error
```

1.7815645

Рисунок 3.5 - Середня помилка з навчанням

Бачимо зменшення помилки більше, ніж в 3 рази. Це означає, що модель знайшла закономірності і навчилася. Тепер спробуємо ще покращити результат. Для цього використаємо пошук гіперпараметрів та зміну так званого альфа-параметру. Задіємо також візуалізацію, щоб побачити, де буде знаходитись мінімум помилки. Результати такого пошуку наведені на рис. 3.6.

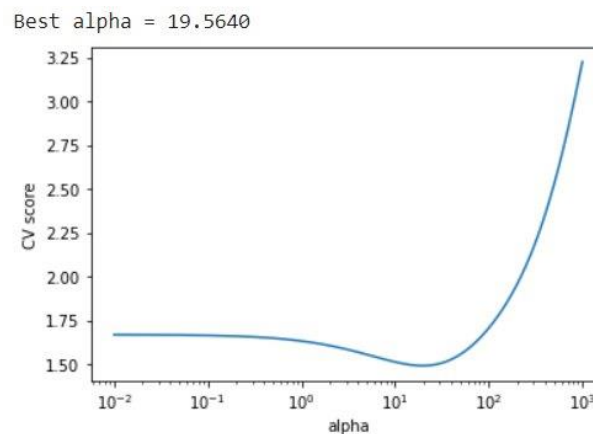


Рисунок 3.6 - Пошук гіперпараметрів моделі

В результаті маємо оцінку середньої помилки в мінімумі близько 1,50 (що в 4 рази краще, ніж просте усереднення). Це означає, що ми додатково покращили роботу інтелектуальної моделі.

3.3 Розробка системи для роботи пристроїв IoT

Розглянемо питання розробки системи для пристроїв “розумного будинку”, які будуть кінцевими в інформаційній мережі для побудови автоматизованих систем. Датчики і виконавчі механізми, як правило, підключаються через бездротовий інтерфейс до контролера розумного будинку - спеціалізованого мікрокомп'ютера, який об'єднує всі ці пристрої в єдину мережу і управляє ними.

Взагалі, деякі рішення можуть поєднувати в собі датчик, виконавчий механізм і контролер одночасно. Наприклад, розумну розетку можна запрограмувати на увімкнення або вимкнення за розкладом, а камера хмарного спостереження — на сигнал датчика руху. У найпростіших випадках можна обійтися без окремого контролера, але для створення гнучкої системи з безліччю сценаріїв потрібен один.

Для підключення контролера розумного будинку до глобальної мережі можна використовувати звичайний інтернет-роутер, який давно став звичним побутовим приладом в будь-якому будинку. Тут є ще один аргумент на користь контролера розумного будинку – якщо зникне інтернет-з'єднання, то розумний будинок продовжить нормально працювати завдяки логічному блоку, що зберігається всередині контролера, а не в хмарному сервісі.

Контролер для хмарної системи розумного будинку, про яку йде мова в цій роботі, заснований на одноплатному мікрокомп'ютері Raspberry Pi 3 model B+ і має достатні ресурси і продуктивність для завдань розумного будинку. Він включає в себе 4-ядерний процесор Cortex-A53 на 64-бітній архітектурі ARMv8-A, з тактовою частотою 1,4 ГГц, а також 1 ГБ оперативної пам'яті, Wi-Fi 802.11ac, Bluetooth 4.2 і Ethernet-адаптер, що працює по шині USB 2.0.

Програмне забезпечення контролера розумного будинку засноване на операційній системі Linux Raspbian Stretch. Вона складається з наступних основних підсистем (рис. 3.7):

- серверного процесу взаємодії з обладнанням розумного будинку та хмарою;
- графічного інтерфейсу користувача для налаштування конфігурації та робочих параметрів контролера;
- бази даних для зберігання конфігурації контролера.

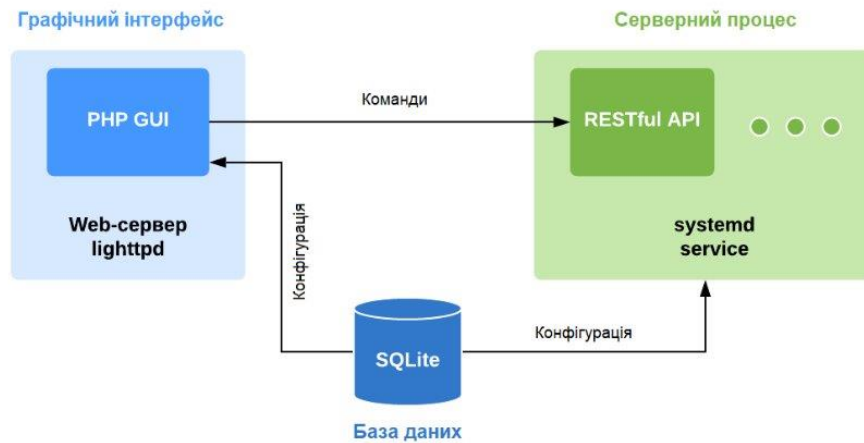


Рисунок 3.7 - Структура програмного забезпечення для контролера IoT

База даних контролерів розумного будинку реалізована на базі вбудованої СУБД SQLite і являє собою файл на SD-карті з системним програмним забезпеченням. Він служить сховищем конфігурації контролера — інформації про підключене обладнання та його поточний стан, блок логічних правил виробництва, а також інформація, що потребує індексації (наприклад, імена файлів у локальному відеоархіві). При перезавантаженні контролера ця інформація зберігається, що дає можливість відновити працездатність контролера в разі збою живлення.

Графічний інтерфейс (рис. 3.8) контролера розумного будинку розроблений на PHP з використанням мікрофреймворку Slim. Додаток працює на веб-сервері lighttpd, який часто використовується у вбудованих пристроях завдяки хорошій продуктивності та низьким вимогам до ресурсів.

Основною функцією графічного інтерфейсу є підключення обладнання розумного будинку до контролера. Веб-додаток зчитує конфігурацію та поточний стан контролера та підключених до нього пристроїв з бази даних SQLite. Для зміни конфігурації контролера він відправляє керуючі команди у форматі JSON через RESTful API серверного процесу.

The screenshot shows the 'IoT gateway' web interface. It features a dark sidebar on the left with navigation options: 'Панель состояния', 'Облако', 'Устройства', and 'Вывод'. The main content area is titled 'Панель состояния' and contains two tables.

И-клиенты (Clients):

Статус	IP-адрес	Производитель	Модель	Серийный номер	ID устройства
●	192.168.1.57	SONOFF	DS-114W	1751114920170814A499624187151	1b1b43d1-30c3-44b7-9e9d-...
●	192.168.1.60	Microdigital	MDC-M6290FTD-1	21980270136179000923	0ee9117-12d3-1095-be9d-...
●	192.168.1.64	Dalmeida	DN-IPC-H10W4231-HP-AS-03639	380169-FAA20020	6a1d35e-1d10-4118-b88e-...

Z-Wave устройства (Z-Wave devices):

HomeID	NodeID	Тип	Модель	ID устройства
0xeff0cta7	1	Stack PC Controller	ZMCC-UZB1-USB-Stack	b139180c-0210-4274-6505-...
0xeff0cta7	18	Water Alarm Sensor	Unknown: type=2025, id=7085	da1a0cb3-315e-4e94-807b-...
0xeff0cta7	19	Home Security Sensor	PST02-A.4 in 1 Multi-Sensor	c6e09e37-d301-15f1-aa01-...
0xeff0cta7	20	Credif Power Switch	PAN16 Smart energy plug in switch	7086c6bd-df8e-4213-a17b-...
0xeff0cta7	21	Smoke Alarm Sensor	PI-PS501 Smoke Sensor	bd3776c-1df8-1c9a-8680-...
0xeff0cta7	22	Home Security Sensor	Unknown: type=2025, id=7085	0ba20a2c-bd1f-43db-b082-...

At the bottom, it shows system information: 'И-клиенты: 6 / 48620 Oct6 4069 3968' and 'Версия прошивки: 3.0.0'.

Рисунок 3.8 - Графічний інтерфейс

Серверний процес є ключовим компонентом, який виконує всі основні роботи з автоматизації інформаційних процесів, що складають основу розумного будинку: прийом і обробку даних датчиків, видачу керуючих дій в залежності від закладеної логіки. Метою серверного процесу є взаємодія з обладнанням розумного будинку, виконання логічних правил, прийом і обробка команд з графічного інтерфейсу і хмари. Серверний процес реалізований у вигляді багатопотокового додатку, розробленого на C++ і запущеного як окремий сервіс systemd операційної системи Linux Raspbian.

Основними блоками серверного процесу є (рис. 3.9):

- диспетчер повідомлень;
- сервер IP-пристроїв;
- сервер пристроїв Z-Wave;
- сервер продукційної логіки;
- база даних конфігурації блоків контролерів і логічних правил;
- RESTful API сервер для взаємодії з графічним інтерфейсом;
- MQTT клієнт для взаємодії з хмарою.

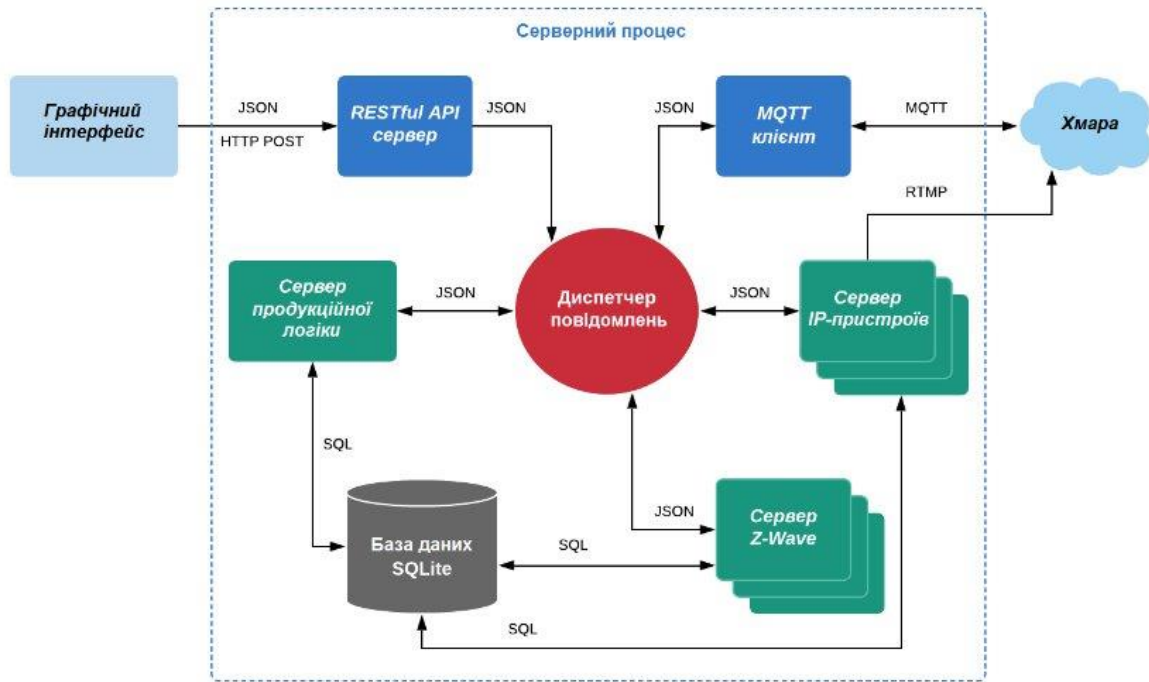


Рисунок 3.9 - Серверний процес

Серверні блоки процесів реалізуються у вигляді окремих потоків, інформація між якими передається у вигляді JSON-повідомлень (або структур даних, що представляють цей формат в пам'яті процесу).

Основним компонентом серверного процесу є менеджер повідомлень, який маршрутизує повідомлення у форматі JSON для всіх блоків серверного процесу. Типи інформаційних полів повідомлень JSON і значення, які вони можуть приймати, перераховані в табл. 3.1.

Таблиця 3.1 - Типи інформаційних полів JSON-повідомлень

<u>deviceType</u>	<u>protocol</u>	<u>messageType</u>	<u>deviceState</u>	<u>command</u>
camera	onvif	sensorData	on	streaming
sensor	zwave	command	off	recording
effector	mqtt	businessLogicRule	streaming	device
	businessLogic	configurationData	recording	
	bluetooth	deviceState	error	

Головна перевага розумного будинку полягає в тому, що він може самостійно вимірювати різні параметри зовнішнього середовища і виконувати корисні функції в залежності від ситуації. Для цього до контролера розумного будинку підключаються датчики і виконавчі механізми. У поточній версії це бездротові пристрої, які працюють за протоколом Z-Wave на спеціально виділеній частоті.

Для своєї роботи вони об'єднані в сітчасту мережу, в якій присутні ретранслятори сигналу для збільшення площі покриття. Також пристрої мають спеціальний режим енергозбереження - більшу частину часу вони проводять в сплячому режимі і відправляють інформацію тільки при зміні їх стану, що значно продовжує термін служби вбудованого акумулятора.

В серверному процесі контролера розумного будинку сервер Z-Wave відповідає за зв'язок з пристроями Z-Wave. Для отримання інформації від датчиків використовується бібліотека OpenZWave C++, яка надає інтерфейс для взаємодії з USB-контролером мережі Z-Wave і працює з безліччю датчиків і виконавчих механізмів. Значення параметра зовнішнього середовища, виміряне датчиком, записується сервером Z-Wave у вигляді JSON-повідомлення.

Потім він пересилається до диспетчера повідомлень серверного процесу, щоб потоки передплатників могли його отримати. Основним абонентом є сервер Production Logic, який зіставляє значення полів повідомлень у предках логічних правил. Вихідні результати, що містять керуючі команди, відправляються назад в диспетчер повідомлень, а звідти на сервер Z-Wave, який декодує їх і відправляє на мережевий контролер Z-Wave USB. Потім вони потрапляють в виконавчий пристрій, який змінює стан об'єктів навколишнього середовища, і розумний будинок, таким чином, виконує корисну роботу.

Пристрої Z-Wave підключаються в графічному інтерфейсі контролера розумного будинку. Для цього перейдіть на сторінку зі списком пристроїв і натисніть кнопку «Додати». Команда add через RESTful API входить у процес сервера, а потім перенаправляється менеджером повідомлень на сервер Z-Wave, який переводить USB-контролер Z-Wave у спеціальний режим додаткового

пристрою. Далі на пристрої Z-Wave потрібно зробити серію швидких натискань (3 натискання протягом 1,5 секунди) сервісної кнопки. USB-контролер підключає пристрій до мережі та надсилає інформацію про нього на сервер Z-Wave. Останній, у свою чергу, створює новий запис у базі даних SQLite з параметрами нового пристрою. Через заданий інтервал часу графічний інтерфейс повертається на сторінку списку пристроїв Z-Wave, зчитує інформацію з бази даних і відображає новий пристрій у списку. Кожен пристрій отримує свій унікальний ідентифікатор, який використовується в правилах виробничого логічного виводу та під час роботи у хмарі.

Робота цього алгоритму показана на діаграмі UML (рис. 3.10).

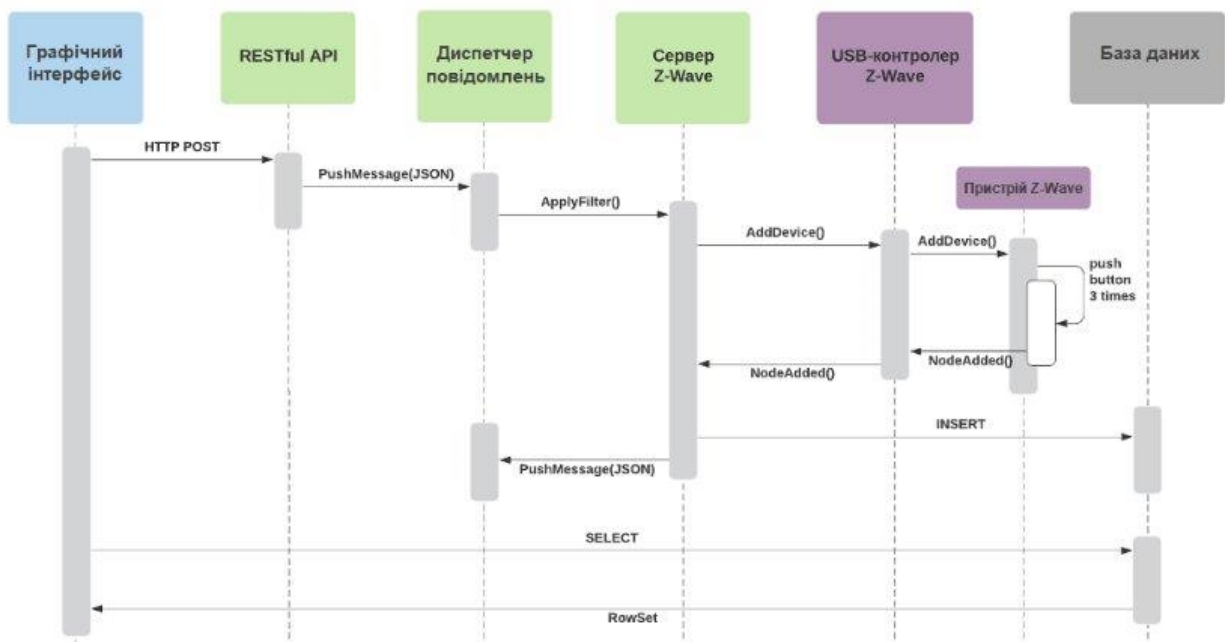


Рисунок 3.10 - Підключення пристроїв Z-Wave

Таким чином, була побудована система розумного будинку з виходом в хмарний сервіс, який може виконувати різні інтелектуальні задачі, в тому числі розглянуту вище задачу управління режимами твердопаливного котла.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ

В сучасних умовах розвитку промислових технологій охорона праці стає важливою складовою успішної діяльності будь-якого підприємства. При розробці і впровадженні інтелектуальних сервісів підбору параметрів котлів необхідно враховувати всі аспекти безпеки для забезпечення здоров'я та життя працівників, які будуть експлуатувати це обладнання.

Інтелектуальні системи, що використовуються для налаштування параметрів котлів, можуть значно підвищити ефективність і безпеку роботи. Проте, це не знімає необхідності в дотриманні основних принципів охорони праці, які включають правильну організацію робочого процесу, використання захисних засобів, а також регулярне навчання та інструктаж персоналу.

Перш за все, робочі місця, де здійснюється експлуатація та обслуговування котлів, повинні бути організовані відповідно до вимог діючих нормативних документів. Це включає правильне освітлення, вентиляцію, а також відповідні умови мікроклімату. Важливо забезпечити, щоб приміщення були обладнані системами аварійного освітлення та сигналізації на випадок надзвичайних ситуацій.

Особлива увага повинна бути приділена технічному стану обладнання. Перед початком роботи з котлом необхідно провести детальний огляд та переконатися в його справності. Це включає перевірку цілісності конструкції, відсутності витоків, справності датчиків і систем автоматичного контролю. Всі виявлені несправності повинні бути негайно усунені перед початком експлуатації.

Персонал, який працює з котлами, повинен бути забезпечений засобами індивідуального захисту. Це включає спеціальний одяг, рукавиці, захисні окуляри та засоби для захисту органів дихання. Використання цих засобів є обов'язковим під час виконання робіт, пов'язаних з обслуговуванням та налаштуванням котлів.

Важливою складовою охорони праці є навчання та інструктаж працівників. Всі працівники повинні пройти початкове навчання з питань охорони праці, а також регулярні повторні інструктажі. Це допоможе їм правильно реагувати на можливі небезпечні ситуації та мінімізувати ризики. Навчання повинно охоплювати не тільки теоретичні аспекти, але й практичні справи з використанням засобів захисту та дій у випадку аварійних ситуацій.

При роботі з інтелектуальними сервісами підбору параметрів котлів необхідно враховувати специфіку програмного забезпечення та можливі ризики, пов'язані з його використанням. Програмне забезпечення повинно бути регулярно оновлюваним для забезпечення його безперебійної роботи та захисту від кібератак. Персонал повинен бути навченим правильному використанню цього програмного забезпечення, включаючи дії в разі його несправностей.

Дотримання правил безпеки під час роботи з інтелектуальними сервісами включає перевірку всіх налаштувань перед запуском котла, а також регулярний моніторинг його роботи. Важливо забезпечити наявність резервних систем контролю та можливість дистанційного відключення котла в разі необхідності.

У разі аварійних ситуацій важливо діяти за заздалегідь розробленим планом. Всі працівники повинні бути ознайомлені з цим планом та вміти швидко евакуюватися з небезпечної зони. Приміщення повинні бути обладнані необхідними засобами для ліквідації пожежі та надання першої допомоги постраждалим.

Одним з ключових аспектів охорони праці є профілактичні заходи, спрямовані на попередження нещасних випадків і професійних захворювань. Це включає регулярні медичні огляди працівників, оцінку їх фізичного та психологічного стану, а також проведення санітарно-гігієнічних заходів.

Система вентиляції і кондиціонування повітря має відповідати нормативам і забезпечувати комфортні умови праці. Важливо регулярно перевіряти стан вентиляційної системи та фільтрів, щоб уникнути накопичення шкідливих речовин у повітрі.

Окрім фізичних умов праці, важливо також забезпечити психологічний комфорт працівників. Це включає підтримку сприятливого психологічного клімату у колективі, запобігання конфліктним ситуаціям та надання психологічної підтримки у разі необхідності.

Кожне підприємство повинно мати розроблений план дій на випадок надзвичайних ситуацій, таких як пожежа, вибух, або витік небезпечних речовин. Цей план повинен бути доступний для всіх працівників, а також регулярно оновлюватись і перевірятись на предмет актуальності. Важливо проводити регулярні тренування і навчання персоналу щодо дій у надзвичайних ситуаціях.

Регулярне технічне обслуговування та своєчасний ремонт обладнання є невід'ємною частиною охорони праці. Всі роботи повинні виконуватись кваліфікованими спеціалістами, які мають відповідну підготовку і досвід. Необхідно використовувати тільки сертифіковані запчастини і матеріали, щоб забезпечити надійність і безпеку експлуатації котлів.

Експлуатація котлів повинна здійснюватись з урахуванням екологічних вимог. Це включає мінімізацію викидів шкідливих речовин в атмосферу, правильну утилізацію відходів, а також зниження рівня шуму та вібрацій. Важливо також проводити регулярний моніторинг екологічних показників та впроваджувати заходи для покращення екологічної безпеки.

Всі заходи з охорони праці повинні бути задокументовані. Це включає ведення журналів обліку технічного обслуговування, протоколів навчання і інструктажів, а також звітів про проведені профілактичні заходи. Наявність повної і достовірної документації дозволяє забезпечити контроль за виконанням вимог охорони праці та своєчасно виявляти і усувати порушення.

Таким чином, охорона праці під час експлуатації інтелектуальних сервісів підбору параметрів котлів є комплексним завданням, що включає організацію робочого місця, технічне обслуговування обладнання, забезпечення персоналу засобами захисту, навчання та регулярний інструктаж, а також готовність до дій у випадку аварійних ситуацій.

Тільки при дотриманні всіх цих заходів можна забезпечити безпечні умови праці та мінімізувати ризики для здоров'я та життя працівників.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи був розроблений інтелектуальний сервіс підбору параметрів управління твердопаливними котлами. В результаті були виконані такі задачі:

1. Проведений аналіз предметної області дозволив провести класифікацію інтернет-сервісів, розглянути їх основні функції, переваги та недоліки.

2. Розглянуті можливості інтелектуальних веб-сервісів, зокрема, для задач інтернету речей.

3. Обрані апаратне та програмне забезпечення створення "розумного будинка".

4. Проаналізована задача моделювання температурного режиму твердопаливного котла, визначені параметри пошуку рішення методами машинного навчання.

5. Створена інтелектуальна модель та перевірена її робота.

6. Обрана структура програмного забезпечення контролера для IoT, наведений приклад роботи системи при підключенні нових пристроїв та вигляд графічного інтерфейса.

7. Розглянуті питання охорони праці та техніки безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cowles P. Web Services and the Semantic Web. URL: <http://www.sys-con.com/webservices/article.cfm?id=419>.
2. Christensen E., Curbera F., Meredith G. Web services description language (WSDL) 1.1. URL: www.w3.org/TR/wsdl.
3. Структура і принципи WEB. WEB-сервери та принципи їх роботи з користувачем. URL: <https://dl.sumdu.edu.ua/drafts/1920/413008/index.html?1683123584>.
4. Порівняння типів архітектури систем сервісів. URL: <http://dspace.nbuu.gov.ua/bitstream/handle/123456789/123541/05-Petrenko.pdf?sequence=1>.
5. Кучеров Д. П. Інженерія програмного забезпечення: навч. посібник / Д. П. Кучеров, Є. Б. Артамонов. – К. : НАУ, 2017. – 388 с.
6. Збіркове програмування компонентних і сервіс-орієнтованих прикладних програмних систем. URL: <http://dspace.nbuu.gov.ua/bitstream/handle/123456789/144494/03-Andon.pdf?sequence=1>.
7. Нетепчук В.В. Управління бізнес-процесами: Навч. посібник. - Рівне: НУВГП, 2014. - 158 с.
8. Бізнес-процеси на підприємстві: аналіз та оптимізація. URL: <http://dspace.wunu.edu.ua/bitstream/316497/46449/1/ОБАзм-21%20Олексій%20В..pdf>
9. Розробка системи пошуку Web-сервісів в мережі Інтернет на основі OWL-S онтологічної моделі подання знань. URL: <https://ena.lpnu.ua:8443/server/api/core/bitstreams/09ec9a1e-4323-4b92-ad8a-6251413df9dd/content>.
10. Особливості web-додатків. URL: <https://sites.znu.edu.ua/webprog/lect/1191.ukr.html>.
11. Щербак Д.В. Інтелектуальні системи створення адаптивних веб-ресурсів. URL:

[https://krs.chmnu.edu.ua/jspui/bitstream/123456789/270/1/АВТОРЕФЕРАТ%20Ще рбак.pdf](https://krs.chmnu.edu.ua/jspui/bitstream/123456789/270/1/АВТОРЕФЕРАТ%20Ще%20рбак.pdf).

12. Інтернет речей. URL: [https://uk.wikipedia.org/wiki/Інтернет речей](https://uk.wikipedia.org/wiki/Інтернет_речей).

13. Що таке IoT простими словами? URL: <https://www.atiko.com.ua/articles-ua/chto-takoe-iot-prostymi-slovami/>

14. Розробка IoT додатків: Як розробити додаток для інтернету речей. URL: <https://stfalcon.com/uk/blog/post/what-is-the-internet-of-things-and-where-is-it-used>.

15. Огляд протоколів IoT і як вибрати найкращий протокол IoT. URL: <https://www.dusuniot.com/uk/blog/best-wireless-protocol-for-your-iot-project/>