ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра інформаційних технологій

Пояснювальна записка

до магістерської роботи

на ступінь вищої освіти магістр

на тему: «**ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖ ДО ЗАДАЧ МАШИННОГО НАВЧАННЯ СИСТЕМИ SMART CITY**»

Виконав: студент 6 курсу, групи ІСДМ–61
спеціальності
122 _Інформаційні системи і технології__
(шифр і назва спеціальності)
Трамбовецький О.А_____
(прізвище та ініціали)
Керівник Ткаленко О.М._____
(прізвище та ініціали)
Рецензент_____
(прізвище та ініціали)

Київ - 2022

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення автоматизованих систем
Ступінь вищої освіти - «Магістр»
Спеціальність підготовки 126 Комп'ютерні науки та інформаційні технології
Освітня програма «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри ІСТ
_____ К.П.Сторчак
"____" _____2021 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Трамбовецький Олександр Андрійович
(прізвище, ім'я, по батькові)

1.Тема роботи: «Дослідження та застосування нейромереж до задач машинного навчання системи Smart City»
Керівник роботи: Ткаленко Оксана Миколаївна, доцент, кандидат технічних наук.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від _____року №
2. Строк подання студентом роботи _____
3. Вхідні дані до роботи :
1. Науково-технічна література
2. Існуючі методології проектування IIoT

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1.

5. Перелік графічного матеріалу
1. Титульний слайд
2. Постановка завдання
3. Методології методології проектування IIoT
6. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Приміт ка
1	Підбір науково-технічної літератури		
2	Вивчення матеріалів для подальшої взаємодії з ними		
3	Огляд протоколів передачі даних		
4	Визначення технічного завдання		
5	Розробка сценарію взаємодії		
6	Розробка бібліотеки		
7	Вступ, висновки, реферат		
8	Розробка демонстраційних матеріалів		
9	Попередній захист роботи		

Студент_____ Трамбовецький О.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Ткаленко О.М.
(підпис) (прізвище та ініціали

РЕФЕРАТ

Текстова частина магістерської роботи: 70 сторінок, 10 рисунків, 1 таблиця, 15 джерел.

Об'єкт дослідження - процес машинного навчання

Предмет дослідження – моделі та процеси навчання нейронної мережі для систем «Розумного міста»

Мета роботи – аналіз навчання та створення нейромережі для концепції «smart city»

Методи дослідження. Для вирішення поставленої мети у роботі використовувалися метод аналізу побудови нейронних мереж, також були підібрані усі необхідні бібліотеки для написання власного програмного забезпечення.

У дипломній роботі зроблена робоча модель нейронної мережі, котра вміє розпізнавати обличчя та впізнавати окремих людей завдяки комп'ютерному зору.

У першому розділі розповідається про різні типи нейронних мереж та їх складові.

У другому розділі йдеться мова про «розумні» міста, різновиди технології їх імплементації, тощо.

Третій розділ аналізу технологій, які я використав для побудови нейромережі.

У практичній частині побудована нейронна мережа, яка розпізнає обличчя. Виконано експериментальний і теоретичний аналізи досліджуваної моделі.

Галузь використання – телекомунікаційні мережі зв'язку.

ЗМІСТ

Стор.

Вступ……..9
1 АНАЛІЗ АРХІТЕКТУР ПОБУДОВИ НЕЙРОМЕРЕЖІ……...11
1.1 Штучна нейронна мережа……...11
1.1.1 Багатошарові перцептрони……12
1.1.2 Згорткові нейронні мережі…….……...14
1.1.3 Рекурсивні нейронні мережі……..15
1.1.4 Рекурентні нейронні мережі……...16
1.1.5 Мережа довгої короткострокової пам’яті……16
1.1.6 Sequence-to-sequence модель……18
1.1.7 Неглибокі (shallow) нейронні мережі……...18
1.2 Переваги штучних нейронних мереж……..18
1.2.1 Розв’язання задач при невідомих закономірностях……19
1.2.2 Стійкість до шумів у вхідних даних……19
1.2.3 Адаптування до змін довкілля……..19
1.2.4 Потенційна надвисока швидкодія……20
1.2.5 Відмовостійкість при апаратній реалізації нейронної мережі……...20
1.3 Недоліки штучних нейронних мереж……..21
1.4 Сфера застосування нейронних мереж……...22
1.5 Нейромережеві алгоритми……24
1.6 Кластеризація……30
2 РОЗУМНЕ МІСТО – РОЗВИТОК ТА РОЗБУДОВА МІСТ МАЙБУТНЬОГО……32
2.1 «Розумне» майбутнє міст……32
2.2 Переваги розумних міст для муніципалітетів та жителів міста……...36
2.3 Інтернет-речей……...37
2.3.1 Принцип роботи Інтернету-речей……41

2.3.2 Стандарт LoraWAN……45
2.3.3 Стандарт Narrowband-IoT……..53
2.3.4 Технологія ZigBee……..59
3 РОЗРОБКА ПРОГРАМИ З РОЗПІЗНАВАННЯ ОБЛИЧ ВИКОРИСТОВУЮЧИ ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ……63
3.1 Мова програмування Python……63
3.2 Бібліотека комп’ютерного зору OpenCV……66
3.3 Бібліотека NumPy……..68
3.4 Пакетний менеджер Anaconda……69
3.5 Бібліотека dlib……70
ВИСНОВОК……..72
ПЕРЕЛІК ПОСИЛАНЬ……...73
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ...……..75
ДОДАТОК……..79

ВСТУП

Актуальність. Інтенсивний розвиток технологій бездротової передачі даних та нейронних мереж впливає на життя громадян великих міст по всьому світу щоденно. У деяких містах у світі вже давно впровадили технології з розпізнавання обличчя. Такі як камери спостереження з розпізнаванням обличчя для безпеки, або оплату обличчям у магазинах або навіть у підземці.

Згорткова СР має спеціальну архітектуру, яка дозволяє їй максимально ефективно розпізнавати образи. Сама ідея CHP ґрунтується на чергуванні згорткових та субдискретизуючих шарів (pooling), а структура є односпрямованою. CHP отримала свою назву від операції згортки, яка передбачає, що кожен фрагмент зображення буде помножений на ядро згортки поелементно, при цьому отриманий результат повинен підсумовуватися і записатися в схожу позицію вихідного зображення. Така архітектура забезпечує інваріантність розпізнавання щодо зсуву об'єкта, поступово укрупнюючи «вікно», на яке «дивиться» згортка, виявляючи дедалі більші структури та патерни в зображенні.

Метою даної роботи є аналіз навчання та створення нейромережі для концепції «smart city».

Об’єкт дослідження - процес машинного навчання

Предмет дослідження – моделі та процеси навчання нейронної мережі для систем «Розумного міста»

Методи дослідження Для вирішення поставленої мети у роботі використовувалися метод аналізу побудови нейронних мереж, також були підібрані усі необхідні бібліотеки для написання власного програмного забезпечення.

Основою всіх архітектур для відеоспостереження є аналіз, першою фазою якого буде розпізнавання зображення (об'єкта). Потім штучний інтелект за допомогою машинного навчання розпізнає дії та класифікує їх.

Для того щоб розпізнати зображення, нейронна мережа має бути раніше навчена на даних. Це дуже схоже на нейронні зв'язки в людському мозку — ми маємо певні знання, бачимо об'єкт, аналізуємо його та ідентифікуємо.

Нейромережі вимогливі до розміру та якості датасету, на якому вона навчатиметься. Датасет можна завантажити з відкритих джерел або зібрати самостійно.

1 АНАЛІЗ АРХІТЕКТУР ПОБУДОВИ НЕЙРОМЕРЕЖІ

1.1 Штучна нейронна мережа

Штучна нейронна мережа (скорочено "нейромережа") - це комп'ютерна програма, що працює за принципом людського мозку: вона проганяє вхідні дані через систему "нейронів" - більш простих програм, що взаємодіють між собою, після чого видає певний результат обчислень на основі цієї взаємодії, враховуючи досвід та помилки минулих запусків програми. Це означає, що будь-яка нейромережа є самонавченою системою «штучного інтелекту». Саме слово "нейрон" у перекладі з грецької означає "нерв". В цілому це математична модель функціонування традиційних для живих організмів нейромереж, які є мережею нервових клітин. Як і в біологічному аналогу, у штучних мережах основним елементом виступають нейрони, з'єднані між собою і утворюють шари, число яких може бути різним залежно від складності нейромережі та її призначення (розв'язуваних задач).

Робота із зображеннями – важлива сфера застосування технологій Deep Learning. Глобально усі зображення з усіх камер світу становлять бібліотеку неструктурованих даних. Задіявши нейромережі, машинне навчання та штучний інтелект, ці дані структурують та використовують для виконання різних завдань: побутових, соціальних, професійних та державних, зокрема, забезпечення безпеки.

Мабуть, найпопулярнішим завданням нейромереж є розпізнавання візуальних образів. Сьогодні створюються мережі, в яких машини здатні успішно розпізнавати символи на папері та банківських картках, підписи на офіційних документах, детектувати об'єкти тощо. Ці функції дозволяють суттєво полегшити працю людини, а також підвищити надійність та точність різних робочих

процесів за рахунок відсутності можливості припущення помилки через людський фактор.

Звичайною часто називають повнозв'язкову нейронну мережу. У ній кожен вузол (крім вхідного та вихідного) виступає як входом, так і виходом, утворюючи прихований шар нейронів, і кожен нейрон наступного шару з'єднаний з усіма нейронами попереднього. Входи подаються з терезами, які в процесі навчання налаштовуються і не змінюються надалі. При цьому кожен нейрон має поріг активації, після проходження якого він приймає одне з двох можливих значень: -1 або 1, або 0 або 1.

Будь-яка архітектура ШНМ складається з штучних нейронів - елементів обробки, що мають структуру 3 пов'язаних один з одним шарів: вхідним, що складається з одного або більше шарів прихованим і вихідним.

Вхідний шар складається з вхідних нейронів, які передають інформацію до прихованих слів. Прихований шар, у свою чергу, передає інформацію у вихідний. Кожен нейрон має входи з вагами – синапсами, функцію активації, що визначає вихідну інформацію при заданій вхідній, та один вихід. Синапси - регульовані параметри, що конвертують нейронну мережу в параметризовану систему.

Зважена сума входів — активаційний сигнал — проходить через функцію активації виведення даних з нейрона. Є кілька видів функції активації: лінійна, ступінчаста, сигмоїдна, тангенціальна, випрямна (Rectified linear unit, ReLu).

Типи функцій активації:

- Лінійна функція;

  $f(x)=ax$

- Ступінчаста функція;

  $$f(x) = \begin{cases} 0 for x < 1 \\ 1 for x \geq 0 \end{cases}$$

- Сигмоїда;

$$f(x) = \frac{1}{1+e^{-x}}$$

- Функція гіперболічного тангенса;

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

- Функція лінійного випрямляча;

$$f(x) = \begin{cases} 0 \; for \; x < \; 0 \\ x \; for \; x \; \geq 0 \end{cases}$$

Навчання (або тренування) — процес оптимізації ваги, в якому мінімізується помилка передбачення, і мережа досягає необхідного рівня точності. Найбільш використовуваний метод визначення вкладу в помилку кожного нейрона — зворотне поширення помилки, з допомогою якого обчислюють градієнт. Це одна з модифікацій методу градієнтного спуску.

За допомогою додаткових прихованих шарів можна зробити систему більш гнучкою і потужною. ШНМ із багатьма прихованими шарами називаються глибокими нейронними мережами (deep neural network, DNN); вони створюють складні нелінійні зв'язки.

1.1.1 Багатошарові перцептрони

Багатошаровий перцептрон складається з трьох або більше шарів. Він використовує нелінійну функцію активації, часто тангенціальну або логістичну, яка дозволяє класифікувати лінійно нероздільні дані. Кожен вузол у шарі з'єднаний з кожним вузлом у наступному шарі, що робить мережу повністю пов'язаною. Така архітектура знаходить застосування у завданнях розпізнавання мови та машинному перекладі.

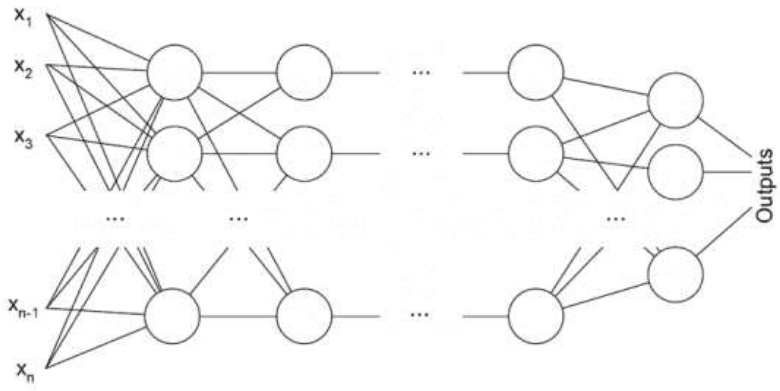


Рисунок 1.1 Структура багатошарового перцептрону

1.1.2 Згорткові нейронні мережі

Згорткова нейронна мережа (Convolutional neural network, CNN) містить один або більше об'єднаних або з'єднаних згорткових шарів. CNN використовує варіацію багатошарового перцептрону, розглянутого вище. Згорткові шари використовують операцію згортки для вхідних даних і передають результат наступного шару. Ця операція дозволяє мережі бути глибшими з меншою кількістю параметрів.

Згорткові мережі показують видатні результати у додатках до картинок та мови. У статті Convolutional Neural Networks for Sentence Classification автор описує процес та результати завдань класифікації тексту за допомогою CNN. У роботі представлена модель на основі word2vec, яка проводить експерименти, тестується на кількох бенчмарках та демонструє блискучі результати.

У роботі Text Understanding from Scratch автори показують, що мережа згортки досягає видатних результатів навіть без знання слів, фраз речень і будь-яких інших синтаксичних або семантичних структур властивих людській мові. Семантичний аналіз, пошук парафраз, розпізнавання мови - теж додатки CNN.

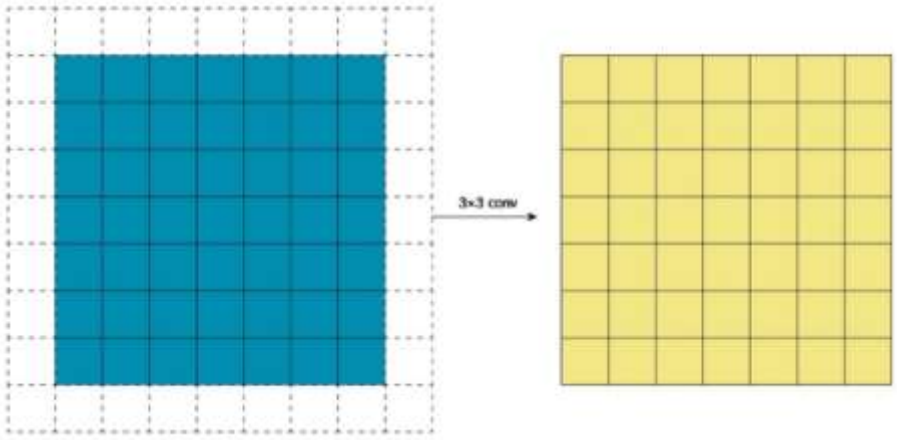


Рисунок 1.2 – Згорткова нейронна мережа

1.1.3 Рекурсивні нейронні мережі

Рекурсивна нейронна мережа — тип глибокої нейронної мережі, сформований при застосуванні тих самих наборів терезів рекурсивно над структурою, щоб зробити скалярне чи структуроване передбачення над вхідний структурою змінного розміру через активацію структури в топологічному порядку. У найпростішій архітектурі нелінійність, така як тангенціальна функція активації, і матриця терезів, що розділяється всією мережею, використовуються для об'єднання вузлів батьківські об'єкти.

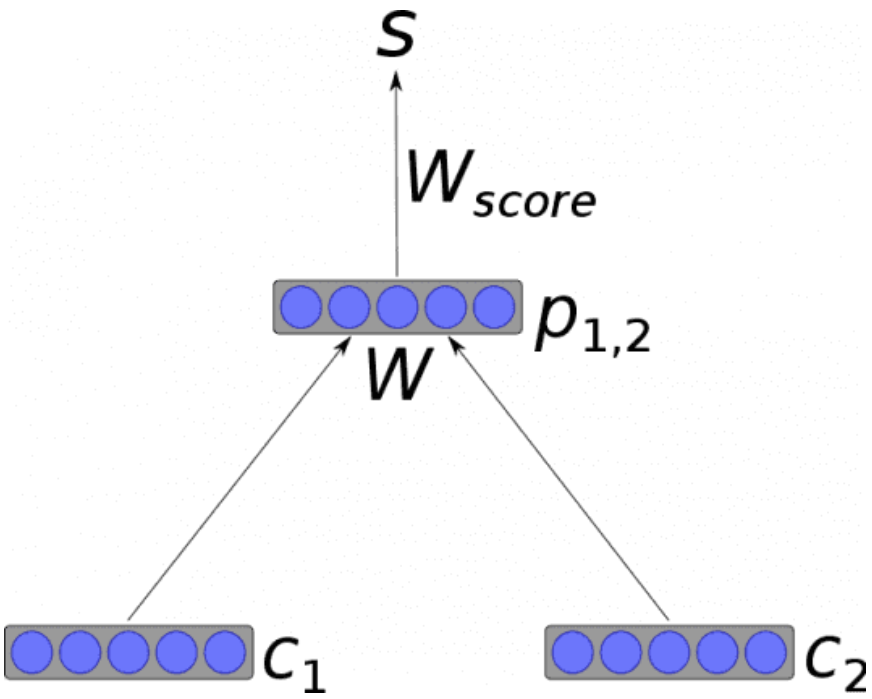


Рисунок 1.3 – Схема рекурсивної нейронної мережі

1.1.4 Рекурентні нейронні мережі

Рекурентна нейронна мережа, на відміну від прямої нейронної мережі, є варіантом рекурсивної ШНМ, де зв'язки між нейронами — спрямовані цикли. Останнє означає, що вихідна інформація залежить не тільки від поточного входу, але також станів нейрона на попередньому кроці. Така пам'ять дозволяє користувачам вирішувати завдання NLP: розпізнавання рукописного тексту чи мови. У статті Natural Language Generation, Paraphrasing та Summarization of User Reviews with Recurrent Neural Networks автори показують модель рекурентної мережі, яка генерує нові пропозиції та короткий зміст текстового документа.

1.1.5 Мережа довгої короткострокової пам’яті

Мережа довгої короткострокової пам'яті (Long Short-Term Memory, LSTM) — різновид архітектури рекурентної нейромережі, створена для точного

моделювання часових послідовностей та його довгострокових залежностей, ніж традиційна рекурентна мережу. LSTM-мережа не використовує функцію активації в рекурентних компонентах, збережені значення не модифікуються, а градієнт не прагне зникнути під час тренування. Часто LSTM застосовується у блоках по кілька елементів. Ці блоки складаються з 3 або 4 затворів (наприклад, вхідного, вихідного та гейту забування), які контролюють побудову інформаційного потоку за логістичною функцією.

У Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling автори показують архітектуру глибокої LSTM рекурентної мережі, яка досягає хороших результатів для великомасштабного акустичного моделювання.

У роботі Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network представлена модель автоматичної морфологічної розмітки. Модель показує точність 97.4 % у розмітці. Apple, Amazon, Google, Microsoft та інші компанії впровадили у продукти LSTM-мережі як фундаментальний елемент.

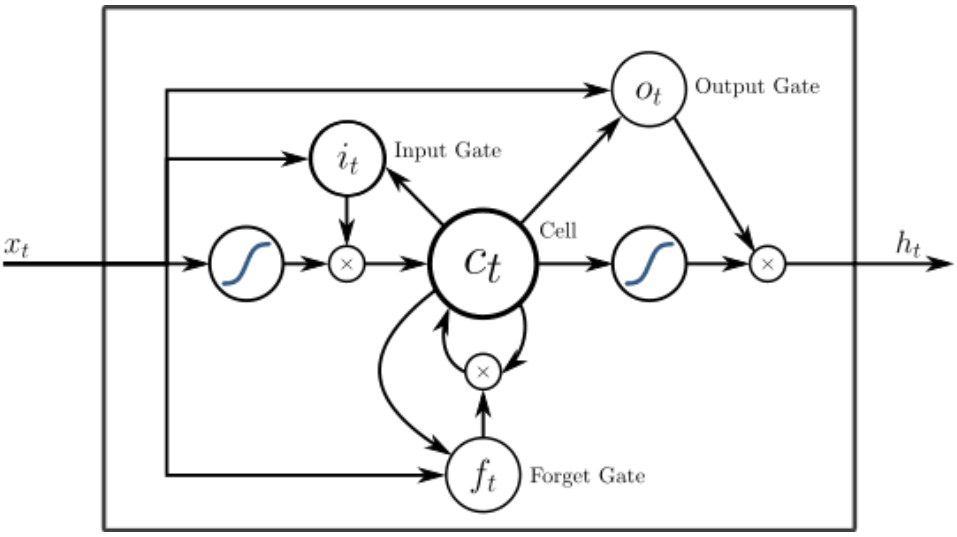


Рисунок 1.4 – Схема мережі довгої короткострокової пам’яті

1.1.6 Sequence-to-sequence модель

Часто Sequence-to-sequence моделі складаються з двох рекурентних мереж: кодувальника, який обробляє вхідні дані, та декодера, який здійснює виведення. Sequence-to-Sequence моделі часто використовуються в питаннях-відповідях системах, чат-ботах і машинному перекладі. Такі багатошарові комірки успішно використовувалися в sequence-to-sequence моделях для перекладу статті Sequence to Sequence Learning with Neural Networks study.
У Paraphrase Detection Using Recursive Autoencoder представлена нова рекурсивна архітектура автокодувальника, в якій уявлення — вектора в n-мірному семантичному просторі, де фрази зі схожими значеннями близькі одна до одної.

1.1.7 Неглибокі (shallow) нейронні мережі

Неглибокі моделі, як і глибокі нейронні мережі, також популярні та корисні. Наприклад, word2vec - група неглибоких двошарових моделей, які використовуються до створення векторних уявлень слів (word embeddings). Представлена в Efficient Estimation of Word Representations in Vector Space, Word2vec приймає на вході великий корпус тексту і створює векторний простір. Кожному слову цього корпусу приписується відповідний вектор у цьому просторі. Відмінна властивість - слова із загальних текстів у корпусі розташовані близько один до одного у векторному просторі.

1.2 Переваги штучних нейронних мереж

Очевидно, що свою силу нейронні сітки черпають, по-перше, від розпаралелювання обробки інформації, і, по-друге, з можливості самонавчати, тобто створювати узагальнення. Узагальнення – це вміння отримати достовірний

висновок з урахуванням наявних даних, які були отримані під час навчання. Ці властивості дозволяють нейронним сіткам вирішувати складні (масштабні) проблеми, які на сьогоднішній день вважаються важкорозв'язними. Але практично автономна робота нейронних мереж неспроможна забезпечити готові рішення. Для цього їх необхідно поєднати зі складними системами.

1.2.1 Розв'язання задач при невідомих закономірностях

Використовуючи здатність навчання на безлічі прикладів, нейронна мережа здатна вирішувати завдання, у яких невідомі закономірності розвитку ситуації та залежності між вхідними та вихідними даними. Традиційні математичні методи та експертні системи в таких випадках пасують.

1.2.2 Стійкість до шумів у вхідних даних

Можливість роботи за наявності великої кількості неінформативних, шумових вхідних сигналів. Немає необхідності робити їх попереднє відсівання, нейронна мережа сама визначить їх малопридатність для вирішення завдання і відкине їх.

1.2.3 Адаптування до змін довкілля

Нейронні мережі мають здатність адаптуватися до змін навколишнього середовища. Зокрема, нейронні мережі, навчені діяти у певному середовищі, може бути легко перенавчані до роботи за умов незначних коливань параметрів середовища. Більше того, для роботи в нестаціонарному середовищі (де статистика змінюється з часом) можуть бути створені нейронні мережі, що переучуються в реальному часі. Чим вище адаптивні можливості системи, тим

стійкішою буде її робота в нестаціонарному середовищі. У цьому слід зазначити, що адаптивність який завжди веде до стійкості; іноді вона призводить до абсолютно протилежного результату. Наприклад, адаптивна система з параметрами, що швидко змінюються в часі, може також швидко реагувати на сторонні збудження, що спричинить втрату продуктивності. Для того, щоб використовувати всі переваги адаптивності, основні параметри системи повинні бути достатньо стабільними, щоб можна було не враховувати зовнішні перешкоди, і досить гнучкими, щоб забезпечити реакцію на суттєві зміни середовища.

1.2.4 Потенційна надвисока швидкодія

Нейронні мережі мають потенційну надвисоку швидкодію за рахунок використання масового паралелізму обробки інформації. Комп'ютер виконує команди послідовно. Однак у голові людини кожен нейрон є маленьким процесором (який приймає сигнал, перетворює його і подає на вихід). І таких процесорів у нас у голові мільярди. Отримуємо гігантську мережу розподілених обчислень. Сигнал обробляється нейронами одночасно.

Ця властивість потенційно проявляється і в штучних нейронних мережах. Якщо у вас багатоядерний комп'ютер, то ця властивість виконуватиметься. Для одноядерних комп'ютерів жодної різниці не буде помітно.

1.2.5 Відмовостійкість при апаратній реалізації нейронної мережі

Нейронні мережі потенційно стійкі до відмови. Це означає, що за несприятливих умов їхня продуктивність падає незначно. Наприклад, якщо пошкоджено якийсь нейрон або його зв'язок, вилучення запам'ятованої інформації не може. Проте, зважаючи на розподілений характер зберігання

інформації в нейронній мережі, можна стверджувати, що тільки серйозні пошкодження структури нейронної мережі суттєво вплинуть на її працездатність. Тому зниження якості роботи нейронної мережі відбувається повільно.

Окремі переваги були названі при описі можливостей нейромереж - наприклад, нейромережа одночасно може вирішувати декілька завдань (за наявності у неї кількох виходів).

1.3 Недоліки штучних нейронних мереж

Незважаючи на широкий спектр можливостей ШНМ, вирішення завдань за їх допомогою супроводжує ряд недоліків:

- більшість підходів для проектування ШНМ є евристичними та часто не призводять до однозначних рішень;
- для побудови моделі об'єкта на основі ШНМ потрібно виконання багатоциклового налаштування внутрішніх елементів та зв'язків між ними;
- проблеми, що виникають під час підготовки навчальної вибірки, пов'язані з труднощами знаходження достатньої кількості навчальних прикладів;
- навчання мережі в ряді випадків призводить до тупикових ситуацій;
- тривалі тимчасові витрати на виконання процедури навчання часто не дозволяють застосовувати ШНМ у системах реального часу;
- поведінка навченої ШНМ не завжди може бути однозначно передбачувано, що збільшує ризик застосування ШНМ для управління дорогими технічними об'єктами;
- більшість відомих комерційних продуктів схемотехнічної реалізації нейронних мереж, що виконуються у вигляді надвеликих інтегральних схем (СВІС), які сьогодні важко назвати широкодоступними та ін.

Пошук оптимального співвідношення параметрів нейромережевих моделей та його характеристик у кожному даному випадку є однією з ключових завдань, для ефективного вирішення якої необхідний широкий спектр методів, алгоритмів та програм синтезу, що різняться за обсягом обчислень, якістю одержуваних результатів, часом пошуку рішення, способами представлення даних.

Ці обставини дозволяють відзначити актуальність розробки нових алгоритмів для підбору навчальних прикладів, орієнтованих на мінімізацію апаратурних витрат за подальшої реалізації ШНМ. Враховуючи перераховані вище недоліки, можна зробити висновок про те, що вибір типу нейронної мережі та створення нейромережевої моделі для будь-якої задачі є трудомістким процесом і часом може не принести задовільних результатів. Як вирішення подібної досить поширеної проблеми пропонується розробка алгоритмів, інструментальних програмних засобів та методики для автоматичного отримання формального опису апроксимуючих нейронних мереж.

1.4 Сфера застосування нейронних мереж

Сфера застосування нейронних мереж багато в чому збігається з колом завдань, які вирішуються традиційними статистичними методами. Тому вкажемо переваги нейронних мереж перед кількома класичними методами статистики. Порівняно з лінійними методами статистики (лінійна регресія, авторегресія, лінійний дискримінант) нейромережі дозволяють ефективно будувати нелінійні залежності (відображення, що розділяють поверхні), які більш точно описують набори даних. З нелінійних методів класичної статистики поширений, мабуть, тільки байєсовський класифікатор, який будує квадратичну поверхню, що розділяє, а штучна нейронна мережа може побудувати поверхню вищого порядку. Висока нелінійність розділяючої поверхні наївного байєсівського класифікатора (він не використовує коварійні матриці класів, як класичний байєс,

а аналізує локальні щільності ймовірності) вимагає значного сумарного числа прикладів для можливості оцінювання ймовірностей при кожному поєднанні інтервалів значень змінних - тоді як нейромережа навчається не фрагментуючи її, що підвищує адекватність налаштування мережі.

Розпізнавання мови – одна з найпопулярніших галузей застосування нейронних мереж.

Ще одна область - управління цінами та виробництвом (втрати від неоптимального планування виробництва часто недооцінюються). Оскільки попит та умови реалізації продукції залежать від часу, сезону, курсів валют та багатьох інших факторів, то і обсяг виробництва повинен гнучко варіюватися з метою оптимального використання ресурсів (нейромережна система виявляє складні залежності між витратами на рекламу, обсягами продажів, ціною, цінами конкурентів, днем тижня, сезоном і т. д.). Через війну використання системи здійснюється вибір оптимальної стратегії виробництва з погляду максимізації обсягу продажу чи прибутку.

При аналізі споживчого ринку (маркетинг), коли звичайні (класичні) методи прогнозування відгуку споживачів може бути недостатньо точні, використовується прогнозуюча нейромережна система з адаптивної архітектурою нейросимулятора.

Дослідження попиту дозволяє зберегти бізнес компанії в умовах конкуренції, тобто підтримувати постійний контакт із споживачами через «зворотний зв'язок». Великі компанії проводять опитування споживачів, дозволяють з'ясувати, які чинники є вирішальними при купівлі даного товару чи послуги, чому у деяких випадках перевагу надається конкурентам і які товари споживач хотів би побачити у майбутньому. Аналіз результатів такого опитування – досить складне завдання, оскільки існує велика кількість корелейованих параметрів. Нейросетельна система дозволяє виявляти складні залежності між факторами попиту, прогнозувати поведінку споживачів при зміні

маркетингової політики, знаходити найбільш значущі фактори та оптимальні стратегії реклами, а також окреслювати сегмент споживачів найбільш перспективний для даного товару.

У медичній діагностиці нейронні мережі використовуються, наприклад, для діагностики слуху у немовлят. Система об'єктивної діагностики обробляє зареєстровані «викликані потенціали» (відгуки мозку), які у вигляді сплесків на електроенцефалограмі, у відповідь звуковий подразник, синтезований у процесі обстеження. Зазвичай для впевненої діагностики слуху дитини досвідченому аудіологу-експерту необхідно провести до 2000 тестів, що займає близько години. Система на основі нейронної мережі здатна з тією ж вірогідністю визначити рівень слуху вже по 200 спостереженнях протягом декількох хвилин, причому без участі кваліфікованого персоналу.

Нейронні мережі застосовуються також прогнозування короткострокових і довгострокових тенденцій у різних галузях (фінансової, економічної, банківської та інших.).

1.5 Нейромережеві алгоритми

Нейромережевим алгоритмом розв'язання задач називається обчислювальна процедура, повністю або здебільшого реалізована як нейронної мережі тієї чи іншої структури (наприклад, багатошарова нейронна мережу з послідовними чи перехресними зв'язками між шарами формальних нейронів) з відповідним алгоритмом налаштування вагових коефіцієнтів. Основою розробки нейромережевого алгоритму є системний підхід, у якому процес вирішення завдання представляється як функціонування у часі певної динамічної системи. Для її побудови необхідно визначити: об'єкт, який виступає у ролі вхідного сигналу нейронної мережі; об'єкт, який виступає у ролі вихідного сигналу нейронної мережі (наприклад, безпосередньо рішення або деяка його

характеристика); бажаний (необхідний) вихідний сигнал нейронної мережі; структуру нейронної мережі (кількість шарів, зв'язки між шарами, об'єкти, що служать ваговими коефіцієнтами); функцію помилки системи (що характеризує відхилення бажаного вихідного сигналу нейронної мережі від реального вихідного сигналу); критерій якості системи та функціонал її оптимізації, що залежить від помилки; значення вагових коефіцієнтів (наприклад, що визначаються аналітично безпосередньо з постановки задачі, за допомогою деяких чисельних методів або процедури налаштування вагових коефіцієнтів нейронної мережі).

Кількість і тип формальних нейронів у шарах, а також кількість шарів нейронів вибираються виходячи зі специфіки розв'язуваних завдань та необхідної якості рішення. Нейронна мережа у процесі налаштування на вирішення конкретного завдання розглядається як багатовимірна нелінійна система, яка в ітераційному режимі цілеспрямовано шукає оптимум деякого функціоналу, який кількісно визначає якість рішення поставленого завдання. Для нейронних мереж, як багатовимірних нелінійних об'єктів управління, формуються алгоритми налаштування безлічі вагових коефіцієнтів. Основні етапи дослідження нейронної мережі та побудови алгоритмів налаштування (адаптації) їх вагових коефіцієнтів включають: дослідження характеристик вхідного сигналу для різних режимів роботи нейронної мережі (вхідним сигналом нейронної мережі є, як правило, вхідна оброблювана інформація та вказівка так званого «вчителя» нейронної мережі) ; вибір критеріїв оптимізації (за ймовірнісної моделі зовнішнього світу такими критеріями можуть бути мінімум середньої функції ризику, максимум апостеріорної ймовірності, зокрема за наявності обмежень на окремі складові середньої функції ризику); розробку алгоритму пошуку екстремумів функціоналів оптимізації (наприклад, для реалізації алгоритмів пошуку локальних та глобальних екстремумів); побудова алгоритмів адаптації коефіцієнтів нейронної мережі; аналіз надійності та методів діагностики

нейронної мережі та ін. Необхідно відзначити, що введення зворотних зв'язків і, як наслідок, розробка алгоритмів налаштування їх коефіцієнтів у 1960-80 роки мали суто теоретичний сенс, тому що не було практичних завдань, адекватних таким структурам. Лише наприкінці 1980-х – початку 1990-х років стали з'являтися такі завдання і найпростіші структури з зворотними зв'язками для їх вирішення (так звані рекурентні нейронні мережі). Розробники в галузі нейромережевих технологій займалися не тільки створенням алгоритмів налаштування багатошарових нейронних мереж та нейромережевими алгоритмами вирішення різних завдань, але й найбільш ефективними (на поточний момент розвитку технології електроніки) апаратними емуляторами (особливі програми, які призначені для запуску однієї системи в оболонці іншої) нейромережевих алгоритмів. У 1960-ті роки, до появи мікропроцесора, найефективнішими емуляторами нейронних мереж були аналогові реалізації розімкнених нейронних мереж з розробленими алгоритмами налаштування на універсальних ЕОМ (іноді системи на адаптивних елементах з аналоговою пам'яттю). Такий рівень розвитку електроніки робив актуальним запровадження перехресних зв'язків у структури нейронних мереж. Це призводило до значного зменшення кількості нейронів у нейронній мережі за збереження якості розв'язання задачі (наприклад, дискримінантної здатності при вирішенні задач розпізнавання образів). Дослідження 1960-70-х років в області оптимізації структур нейронних мереж з перехресними зв'язками напевно знайдуть розвиток при реалізації мемристорних нейронних систем. в залежності від заряду, що протікав через нього], з урахуванням їх специфіки в частині аналого-цифрової обробки інформації і дуже значної кількості настроюваних коефіцієнтів.

Специфічні вимоги прикладних завдань визначали деякі особливості структур нейронних мереж за допомогою алгоритмів налаштування: континуум (від латів. continuum – безперервне, суцільне) числа класів, коли вказівка «вчителя» системи формується у вигляді безперервного значення функції у

певному діапазоні зміни; континуум рішень багатошарової нейронної мережі, який формується вибором континуальної функції активації нейрона останнього шару; континуум числа ознак, що формується переходом у просторі ознак від уявлення вихідного сигналу у вигляді N-мірного вектора речових чисел до речової функції у деякому діапазоні зміни аргументу; континуум числа ознак, як наслідок, потребує специфічної програмної та апаратної реалізації нейронної мережі; варіант континууму ознак вхідного простору був реалізований у задачі розпізнавання періодичних сигналів без перетворення їх за допомогою аналого-цифрового перетворювача (АЦП) на вході системи, та реалізацією аналого-цифрової багатошарової нейронної мережі; континуум числа нейронів у шарі; Реалізація багатошарових нейронних мереж із континуумом класів та рішень проводиться вибором відповідних видів функцій активації нейронів останнього шару.

У таблиці показаний систематизований набір варіантів алгоритмів налаштування багатошарових нейронних мереж у просторі "Вхідний сигнал - простір рішень". Подано безліч варіантів характеристик вхідних та вихідних сигналів нейронних мереж, для яких справедливі алгоритми настроювання коефіцієнтів, розроблених російською науковою школою у 1960–70 роках. Сигнал на вхід нейронної мережі описується кількістю класів (градацій) образів, що становлять вказівки «вчителя». Вихідний сигнал нейронної мережі є кількісним описом простору рішень. У таблиці дана класифікація варіантів функціонування нейронних мереж для різних видів вхідного сигналу (2 класи, K класів, континуум класів) та різних варіантів кількісного опису простору рішень (2 рішення, Kр рішень, континуум рішень). Цифрами 1, 7, 8 представлені конкретні варіанти функціонування нейронних мереж.

<table>
<tr><td rowspan="2">Простір (число) рішень</td><td colspan="5">Вхідний сигнал</td></tr>
<tr><td colspan="2">2 класи</td><td colspan="2">K класів</td><td>Континуум класів</td></tr>
<tr><td>2</td><td colspan="2">1</td><td colspan="2">7</td><td>8</td></tr>
<tr><td rowspan="3">K_p</td><td rowspan="2">$K_p = 3$</td><td rowspan="2">3а</td><td>$K < K_p$</td><td>9</td><td rowspan="3">10</td></tr>
<tr><td>$K = K_p$</td><td>2</td></tr>
<tr><td>$K_p = \text{const}$</td><td>3б</td><td>$K > K_p$</td><td>4</td></tr>
<tr><td>Континуум</td><td colspan="2">5</td><td colspan="2">6</td><td>11</td></tr>
</table>

Таблиця 1 - Набір варіантів алгоритмів налаштування

Основними перевагами нейронних мереж як логічного базису алгоритмів розв'язання складних завдань є: інваріантність (незмінність, незалежність) методів синтезу нейронних мереж від розмірності простору ознак; можливість вибору структури нейронних мереж у значному діапазоні параметрів залежно від складності та специфіки розв'язуваного завдання з метою досягнення необхідної якості рішення; адекватність поточним та перспективним технологіям мікроелектроніки; відмовостійкість у сенсі його невеликого, а не катастрофічного зміни якості розв'язання задачі в залежності від кількості елементів, що вийшли з ладу.

При побудові нелінійних моделей (наприклад, поліноміальних) у статистичних програмах іноді потрібне ручне введення-опис моделі в символьному вигляді з точністю до значень параметрів: при N=10 незалежних змінних поліном другого ступеня міститиме N*(N-1)/2=45 коефіцієнтів при попарних творах змінних, 10 за самих змінних, 10 за квадратів значень змінних, тобто. 66 (з урахуванням неоднорідного доданку) коефіцієнтів. За двадцяти змінних у вираз увійде вже 231 доданок. Вводити вручну такі довгі формули довго, великий ризик друкарської помилки. Нейросеть ж створюється шляхом вказівки виду структури, числа шарів та числа нейронів у кожному шарі, що

набагато швидше. А алгоритми побудови нейромереж, що ростуть, і зовсім не вимагають початкового завдання розміру нейронної мережі. Альтернативою нейронної мережі при побудові складних нелінійних моделей є лише метод групового обліку аргументів та, напевно, машини опорних векторів (support vector machines).

Для стиснення та візуалізації даних у статистиці розроблено метод лінійних основних компонентів. Нейромережі-автоасоціатори дозволяють ефективніше стискати дані за рахунок побудови нелінійних відображень та візуалізувати дані у просторі меншої кількості нелінійних головних компонентів. У порівнянні з методами непараметричної статистики (на основі парзенівських оцінок) мережа з радіальними базовими функціями нейронів (її структура показана на малюнку) дозволяє скорочувати кількість ядер, оптимізувати координати та розмитість кожного ядра. Це дозволяє за збереження парадигми локальної ядерної апроксимації прискорювати роботу побудованої моделі (бо число нейронів у результаті виявляється менше, ніж кількість прикладів навчальної вибірки) і, відповідно, прискорювати процес прийняття рішення.

При навчанні нейромережі замість критерію якості як найменших квадратів можна використовувати робастні критерії, додатково вести оптимізацію та інших властивостей нейронної мережі (наприклад, додаючи вимоги регуляризації рішення, оптимізації структури чи приведення значень адаптивних параметрів до потрібних величин). Алгоритми навчання нейронної мережі у своїй залишаються незмінними.

Необхідність вирішення прямої та зворотної задач зазвичай вимагає побудови двох моделей. При використанні нейронних мереж можна обійтися однією мережею, навченою вирішувати пряме завдання.

1.6 Кластеризація

Кластеризація (самонавчання, навчання «без учителя») – це приватний режим роботи багатошарових нейронних мереж, коли системі не повідомляється інформація про належність зразків до того чи іншого класу. Нейронної мережі пред'являються лише вхідні сигнали, а виходи мережі формуються самостійно з урахуванням лише вхідних і похідних них сигналів. Незважаючи на численні прикладні досягнення, навчання "з учителем" критикувалося за біологічну неправдоподібність. Важко уявити навчальний механізм у природному людському інтелекті, який порівнював би бажані та дійсні значення виходів, виконуючи корекцію за допомогою зворотного зв'язку. Якщо допустити подібний механізм у людському мозку, то звідки виникають бажані виходи? Навчання без вчителя є більш правдоподібною моделлю навчання в біологічній системі. Вона не потребує цільового вектора для виходів і, отже, не вимагає порівняння з певними ідеальними відповідями. Навчальна множина складається лише з вхідних векторів. Навчальний алгоритм підлаштовує ваги нейронної мережі так, щоб виходили узгоджені вихідні вектори, тобто пред'явлення досить близьких вхідних векторів давало однакові виходи. Процес навчання, отже, виділяє статистичні властивості навчальної множини та групує подібні вектори до класів. Пред'явлення на вхід вектора з даного класу дасть певний вихідний вектор, але до навчання неможливо передбачити, який буде вироблятися даним класом вхідних векторів. Отже, виходи такої мережі мають трансформуватися на деяку зрозумілу форму, зумовлену процесом навчання. Це не є серйозною проблемою. Зазвичай не складно ідентифікувати зв'язок між входом та виходом, встановлений мережею.

Кластеризації присвячено безліч наукових праць. Основне завдання кластеризації полягає в обробці множини векторів у багатовимірному просторі

ознак з виділенням компактних підмножин (підмножин, близько розташованих один до одного), їх кількості та властивостей. Найбільш поширеним методом кластеризації є метод K-means, практично не пов'язаний з методами зворотного поширення і не узагальнюється на архітектури типу багатошарових нейронних мереж.

2 РОЗУМНЕ МІСТО – РОЗВИТОК ТА РОЗБУДОВА МІСТ МАЙБУТНЬОГО

2.1 "Розумне" майбутнє міст

Основна мета smart-концепції – оптимізувати функції мегаполісів і, як наслідок, підвищити рівень комфорту їхніх мешканців за рахунок використання сучасних технологій. Традиційно "розумне" місто включає шість smart-складників: уряд, суспільство, економіку, навколишнє середовище, спосіб життя і мобільність.

Насамперед, йдеться про комплекс технологій smart-будинку, що включає просунуті системи управління інженерними мережами, і "розумну" транспортну інфраструктуру. Наприклад, вирішення питання транспортної мобільності є для багатьох мегаполісів ключовим. Так, у низці американських міст з метою зменшити пробки на дорогах встановлюють "розумні" світлофори. Вони отримують інформацію від датчиків, що реагують на автомобілі, та регулюють тривалість свого світлового сигналу. Інший актуальний порядок денний міст — раціональне використання водних ресурсів. За даними Всесвітньої організації охорони здоров'я, до 2025 року половина населення світу проживатиме в районах, які зазнають нестачі води, а до 2050 року глобальна урбанізація призведе до зростання споживання ресурсу на 55%. Тому в містах США, Канади, Японії, Німеччини вже ухвалюють "розумні" рішення у цьому напрямку.

Досвід життя в умовах самоізоляції, очевидно, надовго зробить затребуваними рішення, які допомагають безконтактно взаємодіяти з навколишнім світом, витримувати санітарну дистанцію з випадковими сусідами та дотримуватися суворих гігієнічних вимог. Двері, що відкриваються за датчиком руху або за допомогою технології розпізнавання осіб; камери із тепловізором, що визначають температуру людини з точністю до 0,3 градуси;

обов'язкові для офісів та підприємств відеоаналітичні системи, які відстежують, чи миє персонал руки.

У США вже з'явився попит на програмні рішення на основі штучного інтелекту, які дозволяють наявним системам відеоспостереження фіксувати порушення людьми соціальної дистанції та відсутність масок. Власники магазинів запроваджують нове ПЗ, побоюючись штрафів та закриття торгових точок контролюючими органами. Купівля та використання таких рішень виявляється дешевшим, ніж виділення персоналу для постійного контролю санітарних вимог.

На великих територіях відеоспостереження забезпечує робототехніка та дрони. По Мережі швидко розлетілося відео, на якому робот-собака у пілотному режимі патрулює парк у Сінгапурі та нагадує перехожим про дистанцію. «Електронну собаку» також можна використовувати для оцінки приблизної кількості людей у парку.

Запитуючи «що таке розумне місто», ви вже робите припущення: розумне місто — це єдина певна сутність — щось, що можна взяти цілком і запустити у вигляді повного деталізованого списку. Щоб місто могло називатися розумним, все обладнання має бути підключено до Інтернету речей, все програмне забезпечення має бути взаємопов'язане, а кожен окремий елемент має бути миттєво перевизначений та приведений у відповідність.

Однак насправді все не зовсім так, тільки якщо ви не починаєте з чистого листа. У більшості міст неможливо реалізувати підхід до будівництва з чистого аркуша, тому що немає двох міст з однаковою структурою, статистикою, історією, інфраструктурою чи базовими потребами.

Отже, коли ви думаєте про те, як створити розумне місто, вам необхідно змінити міські стратегії: замість одноразового використання застосовувати багаторівневий підхід, який адаптує всі доступні технології відповідно до широти, глибини, життя та індивідуальності кожного існуючого міста.

В Амстердамі (населення 821 000 осіб), наприклад, більшість жителів їздять велосипедом або користуються громадським транспортом. У Талсі (населення 652 тисячі осіб) більшість жителів їздять на роботу на машині. Ці міста значно відрізняються тим, як люди щодня використовують транспорт, дороги та мости. Однак це не єдина міська служба, на яку так сильно впливає подібна статистика: вона створює величезну розбіжність у сплесках енергоспоживання і навіть визначає, де розмістити лікарні та системи охорони здоров'я (і які фахівці у них потрібні). При раціональному плануванні інтелектуального міста відомості, аналогічні порівняльній статистиці Амстердама та Талси, мають впливати на ієрархію, розподіл бюджету та навіть терміни.

Нам потрібен індивідуальний, адаптований підхід до міського проектування майбутнього і, отже, ми маємо перейти від поняття розумного міста до питання — «як зробити місто розумнішим»?

Коли ви ставите питанням «як виглядають служби розумного міста», ви повинні починати з питання «як виглядає розумна мережа електропостачання».

Можливо, ви знаєте, що слово "мережа" використовується для всіх компонентів розрізненої енергетичної інфраструктури по всьому світу. Це не єдина і безпроблемна мережа підключень, однак у всіх компонентів є одна функція: надавати електрику. Розумна мережа – це яскравіша і блискуча версія цієї електричної інфраструктури. Розумна мережа – це мережа із вбудованою енергоефективністю.

Розумна мережа вже працює у вашому місті. Будь то встановлення розумних лічильників або планування двостороннього підключення сонячних батарей або зарядних станцій для електромобілів — міське проектування, необхідне багатьом ІТ-елементам (від синхрофазорів до термостатів на основі ІІ) — невід'ємна частина повсякденного робочого процесу енергетичної компанії.

Деякі міста основною метою своєї smart-стратегії обрали вуглецеву нейтральність. Так, у Норвегії націлені до 2030 року скоротити викиди вуглецю на 95% шляхом управління освітленням, обігрівом та охолодженням у будинках, а також переходом на електромобілі.

Сучасні технології не тільки роблять міське життя комфортнішим, а й оптимізують транспортну мобільність, а також забезпечують стійкість та ефективне використання ресурсів.

Основною ідеєю розумного міста є спільне використання інфокомунікаційних технологій та Інтернету речей. Використання датчиків та сенсорних мереж дозволяє здійснювати моніторинг у реальному часі. Інформація надходить від самих громадян чи громадських устроїв, потім обробляються та аналізуються у ЦОД.

- Виділяють такі елементи інфраструктури розумного міста:
- Управління водопостачанням та енергопостачанням – розумна ЖКГ;
- Управління відходами - розумне сміття;
- Забезпечення мобільності громадян усередині міста – підвищення ефективності використання доріг як особистим, так і громадським транспортом, запровадження розумного транспорту та розумних паркувань;
- Цифровізація та забезпечення надійного зв'язку – створення середовища для легкого взаємозв'язку та обміну інформацією між громадянами;
- Участь громадян в управлінні містом – електронний уряд;
- Охорона довкілля – контроль рівня забруднення та шуму, створення «зелених» кварталів;
- Безпечне місто – забезпечення безпеки громадян;
- Доступна електронна освіта та охорона здоров'я – розумна охорона здоров'я, телемедицина, дистанційне навчання.

Рішення розумних міст використовують різні платформи, автоматизовані сенсорні мережі та центри обробки даних. Також концепція розумного міста нерідко має на увазі наявність смарт-карток, що видаються кожному мешканцю. Такі смарт-карти є особистим ключем до всіх об'єктів у місті: метро, парковок, кінотеатрів тощо.

2.2 Переваги розумних міст для муніципалітетів та жителів міста

Система смарт-сіті має свої переваги і для влади, і для мешканців. Якщо мерія докладає зусиль до розвитку міста, враховує потреби мешканців, робить його зручнішим та безпечнішим, то отримує:

- зростання чисельності населення;
- інвестиції;
- податки.

Завдяки децентралізації міська влада стала вкладати більше коштів у розвиток інфраструктури. Тепер рівень надходжень до міського бюджету залежить переважно від ефективності роботи влади, тому чиновники змушені шукати способи залучення до проектів бізнесу та іноземних інвесторів. Інвестиції, робота кафе та ресторанів, магазинів та сервісів приносять місту податки, а отже – ресурси для розвитку.

Для жителів міст використання смарт-сіті теж приносить тільки плюси. Насамперед, це безпека — одна з базових потреб кожної людини. Їй сприяє наявність вуличного освітлення, і регульовані світлофори, і камери спостереження. Все це складається в одну систему, яка дає відчути себе захищеним поза домом. Відсутність пробок, чисті двори, швидке вирішення проблем із комунікаціями — це приваблює мешканців та підвищує рівень життя.

Смарт-Сіті дозволяє правильно організувати роботу комунальників. Тому вони можуть оперативно реагувати на аварійні ситуації та швидко з ними

справлятися. Ефективна робота міської влади викликає довіру у мешканців, а за їх підтримки можна розраховувати на залучення іноземних інвестицій. Іноземні компанії вважають за краще базуватися на надійних локаціях, а інвестори завжди вибирають перспективні проекти.

2.3 Інтернет-речей

Інтернет речей, або IoT — це система взаємопов’язаних обчислювальних пристроїв, механічних і цифрових машин, об’єктів, тварин або людей, які мають унікальні ідентифікатори (UID) і можливість передавати дані через мережу без потреби спілкування від людини до людини. взаємодія людини або людини з комп’ютером.

В Інтернеті речей може бути людина з імплантатом серцевого монітора, сільськогосподарська тварина з транспондером біочіпа, автомобіль, який має вбудовані датчики для попередження водія, коли тиск у шинах низький, або будь-які інші природні чи штучні. об’єкт, якому можна призначити адресу Інтернет-протоколу (IP) і який може передавати дані по мережі.

Організації в різних галузях все частіше використовують Інтернет речей, щоб працювати ефективніше, краще розуміти клієнтів, щоб забезпечити покращене обслуговування клієнтів, покращити прийняття рішень і підвищити цінність бізнесу.

Термін "інтернет речей" (IoT) - означає взаємодію пристроїв, підключених до Інтернету та "спілкуються" між собою. Ви, ймовірно, зараз подумали про такі речі, як ноутбук або розумний телевізор, але IoT охоплює значно різноманітніші пристрої, наприклад, електроніку, яка ніколи раніше не була підключена до мережі – домашні холодильники, кавомашини, пральні машини тощо. Інтернет речей – це все (навіть незвичайні пристрої), які можуть підключатися до

Інтернету. У наші дні майже будь-який пристрій із вимикачем потенційно може підключатися до Інтернету, що робить його частиною IoT.

чому всі сьогодні говорять про "інернет речей?"

"Інернет речей" - актуальна тема на сьогоднішній день, тому що ми розуміємо, скільки речей може бути підключено до Інтернету і як це може вплинути на повсякденне життя. Розвиток технологій за останнє десятиліття сприяло таким факторам:

- способи створення технічно «розумних» пристроїв стали доступнішими
- зросла кількість Wi-Fi - сумісних продуктів
- кількість користувачів смартфонами у всьому світі різко збільшилася
- з'явилися можливості використовувати смартфон як пристрій, який керує іншими пристроями

Саме тому IoT – більше не професійний IT-сленг. Це термін, який має знати перш за все кожен власник бізнесу.

Дослідження показують, що IoT-пристрої сприяють підвищенню ефективності роботи підприємства. Продуктивність праці співробітників, дистанційний моніторинг та налагоджені процеси є одними з головних переваг IoT для компаній.

Але як же виглядає інтернет речей усередині компанії? Кожна компанія має свої особливості, але кілька прикладів використання IoT на роботі:

- Розумні замки дозволяють керівнику підприємства відчиняти двері зі свого смартфона, пропускаючи працівника на робоче місце у суботу.
- Інтелектуальні термостати та ліхтарі включаються та вимикаються у потрібний час, щоб заощадити на витратах на електроенергію.
- Голосові помічники, такі як Siri або Alexa, відкривають програми, які дозволяють, наприклад, робити нотатки, встановлювати нагадування, отримувати доступ до свого календаря або надсилати електронні листи.

- Підключені датчики всередині принтерів визначають низький рівень тонера та автоматично розміщують замовлення на його поповнення.
- Камери відеоспостереження, які дають змогу транслювати контент через Інтернет.

Для передачі інформації на звичайний сервер або хмару, де вона буде оброблена та використана далі, датчики оснащують передавальним модулем. У IoT це зазвичай модуль бездротового зв'язку, наприклад: Bluetooth, NFC, RF або Wi-Fi. Іноді кілька датчиків підключають до одного модуля, що передає.

Для швидкого об'єднання датчиків у мережу інтернету речей можна використовувати хмарні платформи, які дозволяють обійтися без власних серверів і передавати дані прямо у хмару захищеними каналами. Крім того, такі платформи допомагають аналізувати дані, зібрані з датчиків, та оптимізувати бізнес-процеси.

Будь-які датчики збирають аналогові дані. Такі дані безперервні - їх можна подати у вигляді звивистої лінії, безперервного потоку інформації. Передавати такі дані за допомогою кабелю або бездротового зв'язку — спочатку сигнал потрібно перетворити на цифрові дані.

Цифрові дані - це послідовність з нулів та одиниць. Щоб перетворити аналогові дані на цифрові, безперервну аналогову лінію потрібно поділити кілька окремих ділянок, і кожному ділянці присвоїти конкретне значення.

Аналогові дані також можна передавати, наприклад, по радіо. Але комп'ютери працюють тільки з цифровими даними, тому їх все одно доведеться переводити в цифру. І краще зробити це до передачі, щоб використовувати більш сучасні та швидкі канали зв'язку.

Прості аналогові датчики не можуть перетворювати сигнал. Щоб отримати від них інформацію у цифровому вигляді, потрібна шкала, де аналоговим значенням відповідають цифрові.

Один окремий датчик може вимірювати лише конкретну фізичну величину, наприклад, вологість. Але майже завжди для цілей бізнесу чи виробництва потрібно вирішувати складніші завдання, ніж просто виміряти один показник. Всі датчики відрізняються один від одного - одні можуть працювати в екстремальних умовах, інші більш точні, треті служать довше. Ці відмінності впливають на ціну - чим більше датчик вміє і чим більше здатний витримати, тим дорожче він коштує.

Екосистема IoT складається з розумних пристроїв із підтримкою Інтернету, які використовують вбудовані системи, такі як процесори, датчики та комунікаційне обладнання, для збору, надсилання та дії з даними, які вони отримують із свого середовища. Пристрої IoT обмінюються даними датчиків, які вони збирають, підключаючись до шлюзу IoT або іншого периферичного пристрою, де дані надсилаються в хмару для аналізу або локального аналізу. Іноді ці пристрої взаємодіють з іншими спорідненими пристроями та діють на основі інформації, яку вони отримують один від одного. Пристрої виконують більшу частину роботи без участі людини, хоча люди можуть взаємодіяти з пристроями - наприклад, щоб налаштувати їх, давати їм інструкції або отримати доступ до даних.

Інтернет речей допомагає людям жити і працювати, а також отримати повний контроль над своїм життям. На додаток до пропозиції розумних пристроїв для автоматизації будинків, IoT є важливим для бізнесу. Інтернет речей надає підприємствам можливість переглядати в режимі реального часу, як насправді працюють їхні системи, надаючи уявлення про все, від продуктивності машин до ланцюга поставок і логістичних операцій.

IoT дозволяє компаніям автоматизувати процеси та знизити витрати на оплату праці. Це також скорочує кількість відходів і покращує надання послуг, що робить виробництво та доставку товарів менш дорогим, а також забезпечує прозорість операцій клієнтів.

Таким чином, IoT є однією з найважливіших технологій повсякденного життя, і вона продовжуватиме набирати обертів, оскільки все більше компаній усвідомлять потенціал підключених пристроїв, щоб підтримувати їх конкурентоспроможність.

2.3.1 Принцип роботи інтернету речей

По-перше, датчики або пристрої збирають дані зі свого середовища. Ці дані можуть бути такими ж простими, як показання температури, або такими ж складними, як повне відео. Кілька датчиків можуть бути об’єднані разом, або датчики можуть бути частиною пристрою, який робить більше, ніж просто збирає інформацію. Однак, незалежно від того, чи це окремий датчик чи повноцінний пристрій, на цьому першому кроці дані щось збираються з навколишнього середовища.

Щоб пристрій міг надсилати інформацію, необхідно здійснювати трансферну передачу даних між пристроєм і хмарою, інакше ваші дані будуть заблоковані всередині датчика. Це можна здійснювати в режимі реального часу за допомогою радіочастоти або кабелю або асинхронно при підключенні ПК/смартфону до пристрою для завантаження. Для наочності можна провести аналогію з нафтовим трубопроводом, яким тече нафту від родовища до конкретного промислового об'єкта.

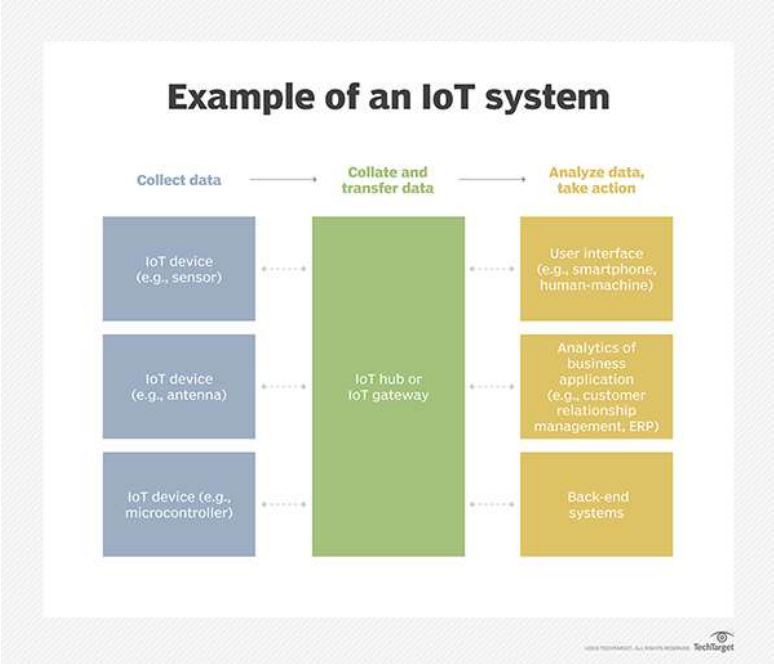


Рисунок 2.1 – Приклад IoT системи

Датчики генерують інформацію про стан активів (місцеположенні, температурі, освітленості, ударах та ін.) постійно, вона передається на платформу для зберігання та аналітики. У зв'язку з цим зростає і вимога до самої мережі: вона повинна витримувати навантаження 24 години на добу 7 днів на тиждень та забезпечувати доступ до даних із пристроїв із ймовірністю 99,99% із мінімальними витратами. На відміну від мережі 5G, що працює з великим обсягом даних від одного джерела, мережа 0G має на увазі надходження інформації від мільйона датчиків, які потім обробляються та аналізуються.

Щоб вся система працювала злагоджено та без втрат, необхідно підібрати пристрої під конкретні завдання. Датчики повинні мати відповідну частоту повідомлень, причому залежно від об'єкта вона буде різною (може змінюватись поріг виявлення, точність та інші параметри). Крім того, важливими є простота установки пристрою та термін його роботи, щоб він збігався з циклом життя об'єкта, на якому він буде розміщений. У такому разі буде знижено витрати на технічне обслуговування.

Кінцеві пристрої є джерелами даних із низькою обчислювальною потужністю, які безперервно передають на шлюз безліч інформації різного формату. Датчик кінцевого пристрою формує аналоговий сигнал, що перетворюється на цифрове (дискретне) значення за допомогою АЦП – аналого-цифрового перетворювача. Це значення маркується міткою часу та класифікується (тегується) локальним процесором кінцевого пристрою. Теги можуть бути простими, наприклад, виявлено рух, або складними з декількох параметрів (рух + швидкість, рух + швидкість + автомобіль тощо). Чим складніший тег, тим потужнішим має бути периферійний процесор та енергоспоживання кінцевого пристрою. Однак, більш інформативні теги дозволяють скоротити кількість даних, що передаються в хмару і смугу пропускання інформації, а це, у свою чергу, збільшує швидкість реакції на подію.

Шлюз, у свою чергу, надсилає дані до хмарного кластеру, де розгорнуто програмну IoT-платформу на базі засобів Big Data для обробки та інтелектуального аналізу інформації.

На хмарному сервері дані від різних периферійних пристроїв інтегруються (сумуються за тегами), систематизуються та аналізуються із застосуванням Machine Learning та інших методів штучного інтелекту. Результати інтелектуального аналізу даних візуалізуються у вигляді графіків, діаграм тощо, відображаючись у вітринах (дешбордах) інтерфейсу користувача IoT-платформи.

Вся IoT-система є розподіленою та масштабованою, проте пов'язаною недостатньо надійними каналами передачі даних. Тому застосовуються механізми гарантованої доставки інформації. Зокрема, якщо не вдається передати дані від кінцевого пристрою на хмару або навпаки, здійснюються повторні спроби передачі. Для обміну сигналами між компонентами розподіленої системи використовуються спеціальні рішення – брокери повідомлень, які гарантують доставку потрібних даних одному чи кільком одержувачам через керовану чергу.

Найбільш популярними брокерами повідомлення вважаються RabbitMQ, Apache Qpid, Apache ActiveMQ. Також для цих цілей використовується розподілений реплікований журнал фіксації змін Apache Kafka, який відмінно масштабується, забезпечуючи нарощування пропускної спроможності при зростанні числа та навантаження з боку джерел даних, а також кількості додатків щодо їх обробки.

У загальному випадку протоколи ділять на групи залежно від ділянки мережі, де вони використовуються. Так, у рамках концепції Інтернету речей існують такі ділянки: сенсорний вузол – сенсорний вузол (найпоширеніший протокол DDS), сенсорний вузол – сервер (CoAP, MQTT, XMPP, STOMP), сервер – сервер (AMQP). Існує безліч протоколів передачі даних, як приклад наведені найпопулярніші. DDS (Data Distribution Service) – реалізує шаблон публікації-підписки для надсилання та прийому даних, подій та команд серед кінцевих вузлів. Вузли-видавці створюють інформацію, «topic» (теми, розділи: температура, місцезнаходження, тиск) та публікують шаблони. Вузол, який зацікавився в цих розділах, DDS прозоро доставляє створені шаблони. Як транспорт – UDP. Також DDS дозволяє керувати параметрами QoS (якість обслуговування).

CoAP (Constrained Application Protocol) – з точки зору користувача схожий на протокол HTTP, але має малий розмір заголовків, що підходить для мереж з обмеженими можливостями. Використовує архітектуру клієнт-сервер та підходить для передачі інформації про стан вузла на сервер (повідомлення GET, PUT, HEAD, POST, DELETE, CONNECT). Як транспорт – UDP.

XMPP (Extensible Messaging and Presence Protocol) – давно використовується в Інтернеті для передачі повідомлень у режимі реального часу, завдяки формату XML підходить для використання в мережах IoT. Працює поверх архітектур видавець-передплатник та клієнт-сервер. Також

використовується для адресації пристроїв у невеликих мережах (адресація типу «name@domain.com»).

MQTT (Message Queue Telemetry Transport) - здійснює збір даних від безлічі вузлів та передачу на сервер. Ґрунтується на моделі видавець-передплатник з використанням проміжного сервера – брокера (пріоритезація повідомлень, формування черг та ін.). Як транспорт – TCP. На основі MQTT було сформовано спеціалізований протокол MQTT-SN для сенсорних мереж.

2.3.2 Стандарт LoraWAN

LoRaWAN — це відкритий стандарт, який визначає протокол зв'язку технології LPWAN на основі чіпа LoRa. LoRaWAN визначає управління доступом до середовища (MAC) на рівні лінії передачі даних та підтримується Альянсом LoRa. Термін Lora часто неправильно використовується для опису всієї системи зв'язку LPWAN. Насправді, LoRa є запатентованим форматом модуляції, що належить компанії «Smartech». Чіпи SX1272 і SX1276 LoRa використовують метод модуляції - розширення спектра, який фізично становить основу технології. Таким чином, можна виділити такі основні поняття в технології LoRa:

- LoRa = Фізичний рівень
- LoRaWAN = рівень MAC
- LoRa + LoRaWAN = стек технології LPWAN

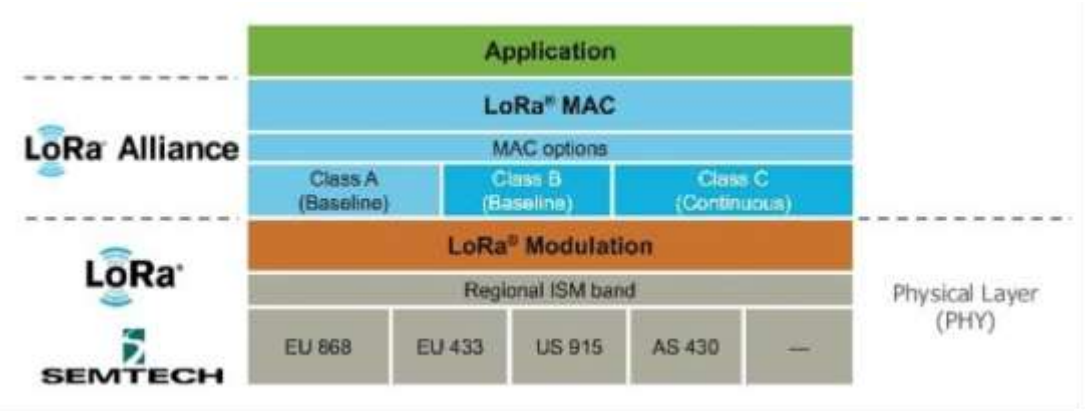


Рисунок 2.2 – Складова технології LoRa

Багато джерел інформації відносять стандарт LoRaWAN до «зоряної» топології і відзначають, що вона краща за стільникову мережу, оскільки дозволяє економити заряд АКБ і має збільшену дальність дії.

Базова станція мережі – шлюз – «слухає» ефір на частотах датчиків. Коли від якогось сенсора вона «чує» сигнал, то відправляє його на сервер. У цьому шлюз не обробляє інформацію. Він служить "перевалочним" пунктом на шляху до сервера.

Сервер необхідний для передачі даних, керування пристроями та приладами, які підключені до датчиків. Якщо датчик спрацював, коли концентрація пилу в повітрі зросла або змінилася вологість у приміщенні, сервер відправить наказ увімкнути витяжку або запустить опалення.

У технології є ще одна ланка – сервери програм. Це комп'ютери, планшети, смартфони та ноутбуки. Вони переводять інформацію від датчиків у зрозумілий вигляд – цифри, графічні схеми та текстові повідомлення. Серверами програм керують співробітники компанії або підприємства.

Зоряна топологія визначає мережеву архітектуру: повідомлення передаються від вузлів на центральний сервер через шлюзи. Кожен кінцевий вузол передає дані на кілька шлюзів. Потім шлюз пересилає дані на мережевий сервер, де виявляється надмірність, а також проводиться перевірка на безпеку та

здійснюється планування повідомлень. Тип зіркової конструкції, що використовується, обумовлений такими властивостями:

1) Просте відстеження активності. Оскільки кінцевий вузол відправляє дані на кілька шлюзів, немає необхідності встановлення зв'язку між шлюзами. Це спрощує логіку програм відстеження активності, де кінцеві вузли є мобільними.
2) Найкраще рішення для загальнодоступних мереж. Наявність центрального сервера, що вирішує проблеми з конфліктами, дозволяє LoRaWAN краще працювати із загальнодоступними мережами порівняно з приватними/клієнтськими. У технології LoRaWAN фактично існує три класи для широкого спектра програм.

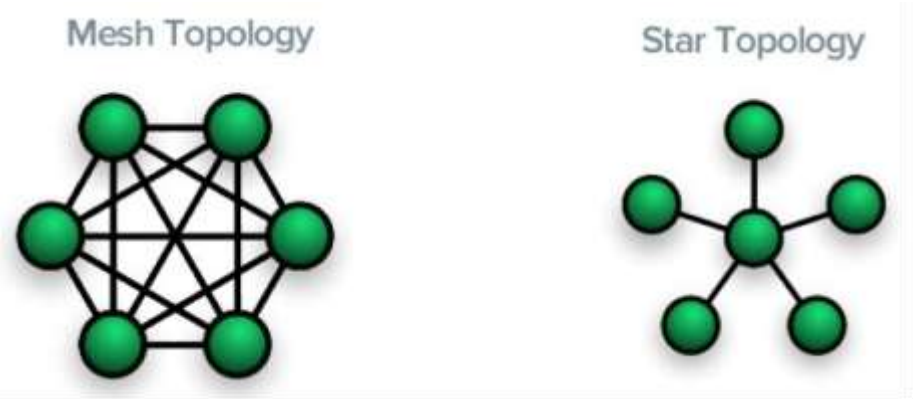


Рисунок 2.3 – Топологія зірка

Передбачуване використання для кожного класу пристроїв описано далі:

- Клас "A" (All - від англ. "все") - приводи або датчики, що працюють від батарейок, що не мають прихованого тимчасового обмеження. Найбільш енергоефективний клас телекомунікаційних систем. Підтримується більшістю пристроїв.
- Клас "B" (Beacon - від англ. "маяк з технологією айбікон") - приводи або датчики, що працюють від батарейок. Енергоефективний клас телекомунікаційних систем, який підходить для передачі пакетних даних із

затримками за часом. Базується на суміщеній з технологією біконів пористої передачі даних.

- Клас "C" (Continuous - від англ. "безперебійні") - магістральні приводи. Пристрої з безперебійним очікуванням сигналу. Не мають затримок у часі.

Фізичний радіоінтерфейс Lo Ra побудований на використанні широкосмугових радіосинхронізованих радіосигналів з великою базою, яка більше одиниці. Цей різновид радіосигналів має дві основні особливості:

- ширина спектра радіосигналу BW значно більша за швидкість передачі даних R b (BW >> Rb);
- кореляційна функція значно кореляційної функції вузькосмугового радіосигналу з базою B ~1.

У зв'язку з цим частотна надмірність широкосмугових радіосигналів обумовлюється їх високою стійкістю до перешкод, а вузька кореляційна функція - високою точністю тимчасової синхронізації.

Широкосмугове радіохвильове випромінювання LoRa є сигналом з модуляції в лінійному частотному діапазоні або CSS (Chirp Spread Spectrum). Частота радіохвиль, які використовуються в CSS, може збільшуватися, так і зменшуватися.

Серед переваг технології можна назвати:

- Надає можливість підключення великої кількості зовнішніх датчиків, так і контрольлерів обладнання. Однак, крім спеціалізованих – жодна інша мережа не витримає сотень сигналізуючих пристроїв на один фінальний сервер та сотні контролерів обладнання додатково. Все це без втрати функціональності останніх.

- Низьке енергоспоживання для зв'язку з базою LoRa, що доходить до того, що одна батарейка на датчику, що посилає свою інформацію на базу LoRa, може служити до декількох років.

- Дистанція передачі. За відсутності зовнішніх перешкод була зареєстрована дистанція в 25 км і це практично рівень горизонту. Все це – не перевищуючи потужності передавачів, за стандартом – 25 мВт. За допомогою цього з'єднання вдалося здійснити передачу пакетів даних, при цьому не було втрачено жодного пакета.

- Малу чутливість сигналу до перешкод, характерними для урбаністичного середовища – зовнішніх передавачів, цегляних та шифельних перекриттів. Слабкий ефект впливу погодних умов на умови прийому.

- Масштабованість для великих відстаней та кількості контрольованого обладнання. Випливає з усіх, вже названих - дальності роботи і безлічі пристроїв, що підключаються. Іншими словами, для розподілу LoRaWAN на великій території не потрібно збільшувати витрати. Тобто, грошова ціна обладнання площі великого масштабу практично дорівнює витратам на таку ж мережу маленького розміру.

- Безпека мережі. Забезпечується трьома факторами – кожний кінцевий пристрій прошивається внутрішнім ідентифікатором та кодом мережі підключення. Зв'язок можливий лише при збігу цих даних з базовою станцією (з боку пристрою – код мережі, з боку бази – приналежність цього ВД поточному приймачеві). Крім того, використовується шифрування даних, що передаються за алгоритмом AES, з обміном ключами між вузлом і підлеглою одиницею. Інформація на рівні програми користувача також кодується.

Крім переваг, є у LoRaWAN певні недоліки. Деякі з них випливають із переваг фізичних або програмних рівнів комунікації:

- Низька швидкість передачі даних. Максимальна швидкість – не більше 5,5 кбіт/секунду. Тут якраз взаємозв'язок між відстанями та відсутністю втрат пакетів інформації. Адже що нижча швидкість обміну, то менше сторонніх перешкод (радіо, іскрові, спотворення сигналів під час проходу через перешкоди) впливають неї у фізичному плані. Це в свою чергу дає велику чіткість сигналів, що передаються, а значить – менша кількість запитів повторного пересилання даних, у разі спотворення вмісту вже відправлених бінарних пакетів. Власне, для датчиків та контролюючих пристроїв цього цілком достатньо, а от аудіо дані або відео інформацію (з камер, припустимо) передати вже не вдасться.

- Стандарт закритий патентами Semetech на рівні чипів. Відповідно, жодних сторонніх виробників, крім тих, хто купив ліцензію на виробництво подібного обладнання, вірніше його компонентів, просто не існує. Великий мінус подібного підходу – практично однакова, завищена ціна на кінцеві датчики, контролери, базові та приймальні станції LoRaWAN.

- Потужність сигналу. Збільшити дальність роботи мережі LoRa простим підвищенням потужності передавачів неможливо, навіть якщо отримати ліцензію контролюючих органів на цю процедуру. Стандарт не має на увазі підвищення понад 25 мВт. Доводиться для збільшення дальності «пробою» зв'язку LoRaWAN розміщувати ретранслююче обладнання на височинах рельєфу або високих об'єктах інфраструктури, що не завжди зручно чи безпечно.

- Затримки. Подібна топологія, яка використовується в мережах типу «зірка», призводить до уповільнення відгуку кінцевих пристроїв, приєднаних до кожного з вузлів, між сервером і ними. Вона не критична, але досить суттєва. Не дозволяє розгорнути подібну інфраструктуру на підприємствах чи організаціях, у яких потрібен високий час відгуку, для систем контролю у часі (таких, наприклад, як АЕС).

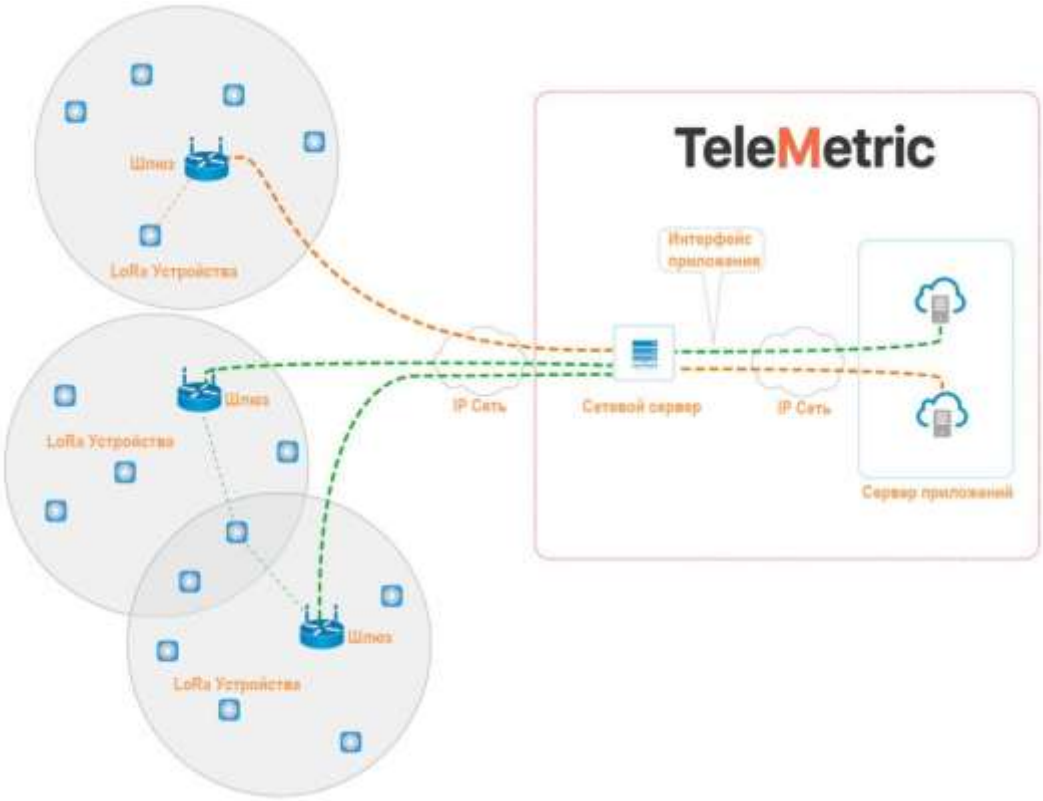


Рисунок 2.4 – Схема мережі LoRa

Обмежена швидкість передачі не дає розгорнути такі мережі багатьох сфер діяльності. Проте, підвищена дальність роботи, за низьких витрат енергії – великий плюс. Отже, для чого найбільше підходять мережі стандарту LoRaWAN:

Спостереження за датчиками, показання яких не критичні. Контактори перетину\руйнування огорож, інформатори стану тунелів, мостів, датчики справності або обладнання. Різні лічильники, що застосовуються в приладах

обліку виробництв або ЖКГ (включаючи можливість управління ними – вкл\викл лінії, що перевіряється).

Управління щодо простими пристроями, завдання програм їхньої роботи (сюди відносяться верстати підприємств, світлофори, лампи зовнішнього чи внутрішнього освітлення, системи вентиляції, витяжки).

Контроль за становищем транспорту. Мережа має хорошу дальність проходження, а значить, вона багато в чому ідеальна для визначення координат тієї чи іншої техніки зараз (при використанні у зв'язці з GPS або Глонасс).

На перший погляд, у нас вже є готові та обкатані рішення. Wi-Fi, LTE, чому не використовувати їх?

Причин кілька. Уявімо будинок на 400 квартир, у кожному з яких коштує два водолічильники та електролічильник. Допустимо, це сучасний будинок, і кожен лічильник передає свідчення в Інтернет.

Об`єм. На один житловий будинок із 400 квартир припаде 1200 лічильників-користувачів. У них буде копійчаний трафік, але якщо всі вони висітимуть, наприклад, на базовій станції LTE, місця для людей на цій базовій станції вже не залишиться. І це один будинок. Адже базову станцію, як правило, ставлять на мікрорайон або навіть більше.

Споживання. Якщо електролічильник ще можна забезпечити харчування, то тягнути кабель до водолічильник не дуже зручно. Отже, радіомодуль водолічильника повинен працювати від батарейки. Але навіть гарну батарею Wi-Fi і LTE з'їдять за кілька діб. Ми ж хочемо, щоб міняти елемент живлення не доводилося щонайменше рік.

Інші пріоритети. Нам не потрібен канал зв'язку 5 мбіт/с, щоб раз на добу передати, скільки кубів води набігло по кожній квартирі. Досить лічених біт. Ми обмежені за потужністю передавача, щоб він не їв батарейку. Отже, можна використовувати правило «більше енергії в один біт – вище ймовірність прийому» таким чином, що канал зв'язку на мінімальній швидкості та з

мінімальною потужністю гарантовано пройде потрібну відстань. Навіть якщо сигнал буде нижчим за рівень шуму.

Після ретельного аналізу ринку компанія Інтерзв'язок вирішила будувати свою мережу на базі стандарту LoRa.

Строго кажучи, абревіатурою LoRa (Long Range) позначають лише вид модуляції, тобто рівень l1 моделі OSI. Протокол канального рівня має ім'я LoRaWAN. Але найчастіше «Лорою» називають сукупну систему, що використовує LoRa на фізичному та LoRaWAN на канальному рівні.

Працює це в такий спосіб. Базова станція слухає ефір у заданому діапазоні частот. Коли вона чує запит від якогось із пристроїв, то відповідає йому на частоті звернення. Ширина каналу становить 125 кГц, максимальна швидкість – трохи більше 5 кілобіт/с. Так-так, ви не дочули. Саме 5 і саме кілобіт/с. Цей стандарт Інтернету не створений для перегляду потокового відео. Його завдання максимально швидко та гарантовано передати невелике повідомлення від датчика на базову станцію. Залежно від радіоумов вибирається оптимальний набір параметрів зв'язку. За це відповідає SF (spreading factor) - коефіцієнт, до якого прив'язуються параметри передачі та прийому. SF - це ціле число, у стандарті він передбачений від 12 до 7. Чим вище SF, тим краще перешкода лінії, але тим нижче швидкість і тим більше часу в ефірі займає передача. Наприклад, максимальна помехозащищенность досягається SF=12. При цьому час пакету в ефірі становить 2466 сек, а швидкість - 292 біт/сек.

2.3.3 Стандарт Narrowband-IoT

NarrowBand-Internet of Things (NB-IoT) — це заснована на стандартах технологія широкої зони з низьким енергоспоживанням (LPWA), розроблена для створення широкого спектру нових пристроїв і послуг IoT. NB-IoT значно покращує енергоспоживання користувальницьких пристроїв, ємність системи та

ефективність використання спектру, особливо в умовах глибокого покриття. Термін служби батареї більше 10 років може підтримуватися в широкому діапазоні випадків використання.

Нові сигнали та канали фізичного рівня розроблені, щоб задовольнити вимоги розширеного покриття – у сільській місцевості та глибоко всередині приміщень – та надзвичайно низької складності пристроїв. Очікується, що початкова вартість модулів NB-IoT буде порівнянна з GSM/GPRS. Проте базова технологія набагато простіша, ніж сучасні GSM/GPRS, і очікується, що її вартість швидко знизиться у міру зростання попиту.

Він призначений для підключення до них автономних пристроїв. Зокрема це стосується медичних датчиків, лічильників споживання ресурсу, пристроїв для розумного будинку тощо. У побуті такі системи зв'язку стали називати інтернет-речей (англ. Internet of things (IoT)). Також у стандарті NB-IoT є три стандарти IoT, розроблені компанією 3GPP для стільникових операторів зв'язку, такі як: eMTC (enhanced Machine-Type Communication), NB-IoT и EC-GSM-IoT.

В даний час eMTC має найбільшу пропускну здатність та розгортається на обладнанні LTE. На обладнання стільникових мереж LTE, а також окремо, у тому числі поверх GSMA, можуть бути встановлені NB-IoT мережі. При цьому EC-GSM-IoT має найменшу пропускну здатність і розгортає поверх мереж стандарту GSM.

Серед переваг Nb-IoT виділяють наступні:

- гнучке управління енергоспоживанням пристроїв (аж до 10 років у мережі від батареї ємністю 5 Вт * год);
- величезна ємність мережі (десятки-сотні тисяч підключених пристроїв однією базову станцію);
- низька вартість пристроїв;
- оптимізована для покращення чутливості модуляція сигналу.

Завдяки розробкам консорціуму 3GPP з'явиться можливість розгортання NB-IoT мереж у всіх частотних діапазонах, у яких застосовується стандарт LTE. При цьому стандарт NB-IoT здатний передавати дані зі швидкістю 200 Кб/с. Цього показника вистачає для безпроблемної роботи пристроїв, які з певною періодичністю передають невеликий обсяг інформації одного типу.

Розробники цієї технології вже встигли пообіцяти, що максимальний термін експлуатації акумуляторів NB-IoT може досягати 10 років. Також слід сказати, що орієнтовна вартість терміналу NB-IoT становитиме 5 доларів.

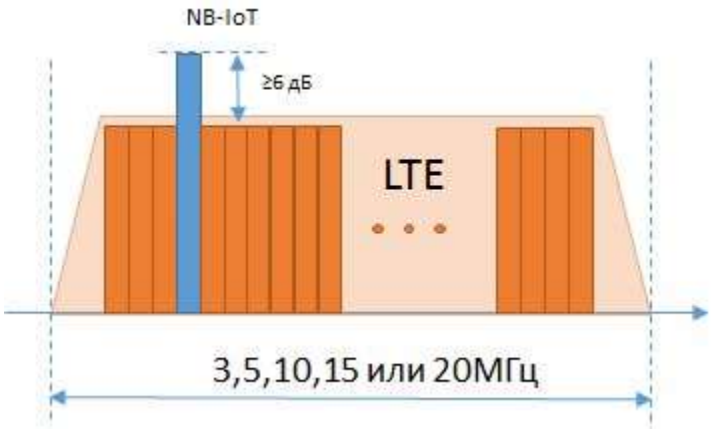


Рисунок 2.5 – Розміщення NB-IoT в режимі In Band

Ще однією надзвичайно важливою перевагою цієї технології вважається можливість підключення близько 100 000 NB-IoT пристроїв до однієї сотні станції. Цей показник на порядок вищий за нинішні стандарти стільникового зв'язку. За рахунок цього забезпечується додаткова комерційна перевага, зумовлена можливістю використання аналізу даних IoT за допомогою методів Big Data. Завдяки цій технології оператори стільникового зв'язку отримають унікальну можливість продавати отримані аналітичні зведення організаціям, що працюють у суміжних сферах діяльності.

Технологія NB-IoT, вперше визначена в 3GPP версії 13, є найбільш підходящою технологією мобільної мережі для IoT пристроїв, для яких потрібні виключно глибоке покриття і наднизьке енергоспоживання. Вони зазвичай

характеризуються низькою швидкістю передачі даних та середнім часом реакції, що становить кілька секунд; як приклад можна навести розумні датчики, підключені контейнери для сміття, велосипеди, бджолині ферми і мишоловки. Розгортання перших мереж знову розпочалося наприкінці 2017 року, а глобальне комерційне розгортання розпочалося 2018 року. NB-IoT може розгортатися на несучій LTE, у захисній смузі частот LTE або як автономне рішення. У майбутньому дуже гнучка нумерологія 5G NR також підтримуватиме внутрішньосмуговий режим роботи 5G для NB-IoT.

У двох словах NB-IoT можна характеризувати як технологію стільникової мережі LPWAN, що працює в низхідному каналі в межах 180 кГц (PRB) і рознесення піднесе 15 кГц, а також у висхідному каналі з передачею, як правило, одного тонального сигналу частотою 3,75 кГц чи 15 кГц. Для виконання додаткових вимог щодо максимального перехідного загасання (MCL) на рівні 20 дБ у технології NB-IoT використовуються методи розширення покриття, такі як форсування потужності в низхідному каналі або повтор підкадрів у висхідному та низхідному каналах. Для продовження терміну служби акумулятора до 10 років і більше потрібні спеціальні функції енергозбереження, такі як PSM, eDRX та індикація відключення (RAI). Крім того, для впровадження NB-IoT і LTE-M була потрібна специфікація оптимальної архітектури ядра, яка, зокрема, забезпечує можливість передачі даних через площину управління і завчасної передачі даних (EDT).

У технології NB-IoT використовується безліч знайомих функцій та методів із LTE. У зв'язку з цим більшість сценаріїв випробувань дуже подібні до випробувань модему LTE. На певні особливості програми LPWAN слід звернути особливу увагу. Наприклад, для продовження терміну служби акумулятора до 10 років і більше рекомендується оптимізувати енергоспоживання для різних конфігурацій та схем роботи. Для роботи в екстремальних умовах покриття потрібні найкращі в класі ВЧ- та антенні пристрої, для перевірки яких ідеально

підходять ефірні випробування. Рекомендується аналізувати вплив різних рівнів CE на енергоспоживання та час затримки, а також на роботу за різних умов завмирання. Також цікавлять наскрізні випробування додатків і, зокрема, перевірка режиму безпечного та конфіденційного зв'язку для пристроїв NB-IoT.

Найбільш поширеним є внутрішньосмуговий. Він широко використовується в Європі, де мережі NB-IoT in-band розгорнуті телекомунікаційними компаніями Vodafone, Deutsche Telekom, Telecom Italia Mobile та іншими.

Внутрішньосмуговий спосіб розгортання дозволяє використовувати NB-IoT всередині існуючих мереж LTE, які мають велике покриття та забезпечують необхідну якість обслуговування інтернету речей. Це оптимальний шлях для підключення IoT, оскільки дозволяє операторам зв'язку використовувати сучасне обладнання без істотних змін апаратного забезпечення. Таким чином, in-band NB-IoT дає можливість задовольнити поточний попит на підключення розумних пристроїв. Незважаючи на поширеність внутрішньосмугового способу використання NB-IoT, він має недоліки. Так, вузькосмуговий сигнал in-band NB-IoT займає частоту 180 кГц і мережах LTE може створювати перешкоди для блоків фізичного ресурсу (PRB). У такому режимі NB-IoT забирає багатофункціональні ресурси стільникових мереж. При внутрішньосмуговому використанні PRB стандарт NB-IoT відокремлюються від інших PRB за допомогою динамічних характеристик потужності. Зокрема, звичайний PRB для LTE потужніший на 6 Дб, ніж PRB для NB-IoT in-band і guard-band. Також специфікації стандарту NB-IoT припускають інші відмінності в структурі сигналу, наприклад, типи модуляції сигналу, що підтримуються, частотна похибка, амплітуда вектора помилок (EVM) і параметри синхронізації та управління каналами.

Показник EVM (Error Vector Magnitude) є важливим критерієм якості модуляції та продуктивності каналу прийому та передачі даних у складних бездротових мережах, включаючи LTE та NB-IoT. Фактично, EVM – це різниця

між «еталонним» переданим сигналом та прийнятим (виміряним) сигналом. EVM — корисний індикатор якості бездротового сигналу, що передається, який допомагає оцінити продуктивність NB-IoT.

При запуску нового сервісу на основі in-band NB-IoT перешкоди між PRB є основною проблемою. Тому перевірка взаємних перешкод PRB при вимірі EVM – найточніша та комерційно ефективніша процедура тестування.

За допомогою NB-IoT можна використовувати практично ті самі діапазони частот як для 2G/3G/4G в «низьких» band. У цьому діапазоні знаходиться B20 (800МГц) та B8(900МГгц), B3(1800мгц) та B3(1800мгч). Це пов'язано з тим, що більш високі частоти сенсу використовувати не варто через більшу згасання сигналу.

Ще одна особливість NB-IoT - функціонал coverage enhancement, який досягається послідовними переповторами сигналу, що передається. Цей механізм не слід плутати з повторною передачею пакета при неуспішному прийомі, у разі coverage enhancement рішення про успішність прийнятого сигналу відбувається після прийому всіх повторених повідомлень (Мал.7). Повторяться можуть усі фізичні канали NPDCCH, NPDSCH, NPRACH та NPUSCH (тут N приставка Narrowband).

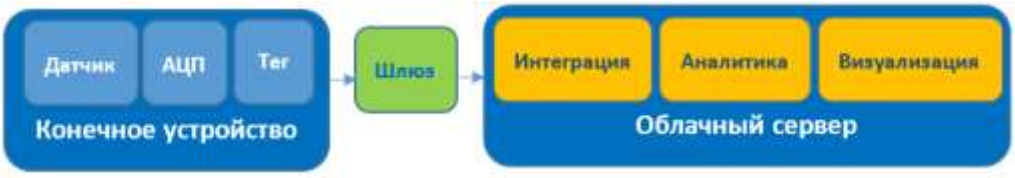


Рисунок 2.6 – Схематична побудова системи використання NB-IoT

Для технології NB-IoT визначили обмежені характеристики для того, щоб можна було використовувати її на базі існуючих стільникових мереж.

NB-IoT обмежений загальною смугою одного RB шириною 180кГц. Радіочастина пристрою, що використовує NB-IoT, має лише одну антену. Передача та прийом рознесені за часом (напівдуплекс).

Є можливість передавати дані за напрямком Uplink на одній піднесеній.
Обмежені типи модуляцій, що використовуються BPSK та QPSK.
Можуть застосовуватися переповтори сигналу, що передається.
Відповідно застосування таких обмежень і було покликане скоротити накладні витрати на споживання енергії, розміри і, звичайно, ціну пристрою.

2.3.4 Технологія ZigBee

Мережі ZigBee, на відміну від інших бездротових мереж передачі даних, повністю задовольняють перелічені вище вимоги, а саме:
а) завдяки пористій (mesh) топології мережі та використанню спеціальних алгоритмів маршрутизації мережа ZigBee забезпечує самовідновлення та гарантовану доставку пакетів у випадках обриву зв'язку між окремими вузлами (появи перешкоди), перевантаження чи відмови якогось елемента;
б) специфікація ZigBee передбачає криптографічний захист даних, що передаються бездротовими каналами, і гнучку політику безпеки;
в) пристрої ZigBee відрізняються низьким електроспоживанням, особливо кінцеві пристрої, для яких передбачено режим «сну», що дозволяє цим пристроям працювати до трьох років від однієї звичайної батареї AA і навіть AAA;
г) мережа ZigBee - самоорганізується, її структура задається параметрами профілю стека конфігуратора і формується автоматично шляхом приєднання (повторного приєднання) до мережі пристроїв, що утворюють її, що забезпечує простоту розгортання і легкість масштабування шляхом простого приєднання додаткових пристроїв;
д) пристрої ZigBee компактні та мають відносно невисоку вартість.

Зв'язок мережі ZigBee здійснюється шляхом послідовної ретрансляції пакетів від вузла джерела до вузла адресата. У мережі ZigBee передбачено кілька альтернативних алгоритмів маршрутизації, вибір яких відбувається автоматично. Стандарт передбачає можливість використання каналів у кількох частотних діапазонах. Найбільша швидкість передачі та найкраща завадостійкість досягається в діапазоні від 2,4 до 2,48 ГГц. У цьому діапазоні передбачено 16 каналів 5 МГц.

Ціна, яку довелося заплатити у мережах ZigBee за мінімізацію енергоспоживання, компактність та дешевизну – відносно низька швидкість передачі даних.

"Брутто" швидкість (включаючи службову інформацію) становить 250 кбіт/с. Середня швидкість передачі корисних даних, залежно від завантаження мережі та кількості ретрансляцій, становить від 5 до 40 кбіт/с.

Відстань між робочими станціями мережі становить десятки метрів усередині приміщень та сотні метрів на відкритому повітрі. За рахунок ретрансляцій зона, що покривається мережею, може бути дуже значною: до декількох тисяч квадратних метрів у приміщенні і до декількох гектарів на відкритому просторі. Більше того, мережа ZigBee у будь-який момент може бути розширена додаванням нових елементів або, навпаки, розбита на кілька зон простим призначенням відповідного числа нових конфігураторів мережі. Це корисно зниження навантаження і підвищення швидкості передачі даних.

Мережі ZigBee будуються з базових станцій трьох основних типів: координаторів, маршрутизаторів та кінцевих пристроїв.

Координатор запускає мережу та керує нею. Він формує мережу, виконує функції центру управління мережею та довірчого центру (trust-центру) – встановлює політику безпеки, задає налаштування у процесі приєднання пристроїв до мережі, розповідає ключами безпеки.

Маршрутизатор транслює пакети, здійснює динамічну маршрутизацію, відновлює маршрути під час навантаження в мережі або відмови будь-якого пристрою. При формуванні мережі маршрутизатори приєднуються до координатора або інших маршрутизаторів і можуть приєднувати дочірні пристрої – маршрутизатори та кінцеві пристрої. Маршрутизатори працюють у безперервному режимі, мають стаціонарне живлення та можуть обслуговувати «сплячі» пристрої. Маршрутизатор може обслуговувати до 32 сплячих пристроїв. Кінцевий пристрій може приймати та відправляти пакети, але не займається їхньою трансляцією та маршрутизацією. Кінцеві пристрої можуть бути підключені до координатора або маршрутизатора, але не можуть мати дочірніх пристроїв.

Кінцеві пристрої можуть переводитися в режим сну для економії заряду акумуляторів. Саме кінцеві пристрої мають справу з датчиками, локальними контролерами та виконавчими механізмами.

Мережа ZigBee - самоорганізується, і її робота починається з формування. Пристрій, призначений при проектуванні координатором персональної мережі (PAN координатор), визначає канал, вільний від перешкод, і очікує на запити на підключення. Пристрої, що намагаються приєднатися до мережі, розсилають широкомовний запит. Поки PAN координатор – єдиний пристрій мережі, відповідає на запит і надає приєднання до мережі тільки він. Надалі приєднання до мережі можуть надавати маршрутизатори, що також приєдналися до мережі.

Пристрій, який отримав відповідь на широкомовний запит, обмінюється з пристроєм, що приєднує повідомленнями, щоб визначити можливість приєднання. Можливість визначається здатністю приєднуючого маршрутизатора обслужити нові пристрої на додаток до раніше підключених. Протоколи, регламентовані стандартами IEEE 802.15.4 та ZigBee 2007 Specification, забезпечують формування та функціонування бездротової сенсорної мережі.

Стандарт IEEE 802.15.4 визначає фізичний та MAC рівні, а специфікація ZigBee визначає мережевий рівень та рівень додатків. На малюнку показано стек протоколів ZigBee.

3 РОЗРОБКА ПРОГРАМИ З РОЗПІЗНАВАННЯ ОБЛИЧ ВИКОРИСТОВУЮЧИ ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ

3.1 Мова програмування Python

Програма буде написана мовою програмування Python версії 3.9, використовуючи вільні для розповсюдження бібліотеки та віртуальне оточення Anaconda.

Коли ми вибираємо, який алгоритм застосувати до завдання розпізнавання об'єкта чи особи, ми маємо засіб порівняння ефективності різних алгоритмів. У цій частині я опишу інструменти, за допомогою яких це робиться.

Оцінка якості роботи системи розпізнавання облич проводиться за допомогою набору метрик, які відповідають типовим сценаріям використання системи для аутентифікації за допомогою біометрії.

Як правило, робота будь-якої нейронної мережі може бути виміряна з точки зору точності: після налаштування параметрів та завершення процесу навчання мережа перевіряється на тестовій множині, для якої ми маємо відгук вчителя, але який відокремлений від навчального набору. Як правило, цей параметр є кількісним заходом: число (часто у відсотках), яке показує, наскільки добре система здатна розпізнавати нові об'єкти. Ще один типовий захід - це помилка (може бути виражена як у відсотках, так і в числовому еквіваленті). Проте для біометрії існують точніші заходи.

У біометрії взагалі та біометрії для розпізнавання осіб, зокрема, існує два типи додатків: верифікація та ідентифікація. Верифікація являє собою процес підтвердження певної особистості шляхом порівняння зображення індивіда (вектор ознак обличчя або іншого вектора ознак, наприклад, сітківки або відбитків пальців) з одним або декількома раніше збереженими шаблонами. Ідентифікація – це процес визначення особистості індивіда. Біометричні зразки збирають та порівнюють з усіма шаблонами в базі даних. Існує ідентифікація у

замкнутому множині ознак, якщо передбачається, що людина існує у базі даних. Таким чином, розпізнавання поєднує один або обидва терміни - верифікацію та ідентифікацію.

Face Detection – перший етап процесу розпізнавання обличчя. На фотографії визначається обличчя людини, після чого вирізується невелика область навколо. Коли обличчя знайдено, починається процес розпізнавання. Нейронна мережа вчиться перетворювати картинку із зображенням обличчя людини у Face Descriptor. Такі фактори, як освітлення, повороти голови, зачіска та макіяж не повинні збентежити II. Нейронна мережа вчиться точно ідентифікувати особистість незалежно від цих змін.

Для здійснення наступного етапу – порівняння дескрипторів, нам потрібна спеціальна база даних. Після видачі нейромережею певного дескриптора по базі запускається пошук, тому Face Descriptors мають зберігатися з прив'язкою до певної особи. У сховищі підбирається максимально схожий виданий нейромережею дескриптор. Таким чином, у разі знаходження правильного вектора в базі, машина дізнається та ідентифікує особу.

Python — це високорівнева мова програмування загального призначення, яка використовується навіть для розробки веб-додатків. Мова орієнтована підвищення продуктивності розробника і читаності коду.

Python підтримує кілька парадигм програмування: структурне, об'єктно-орієнтоване, функціональне, імперативне та аспектно-орієнтоване. У мові присутня динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки винятків, підтримка багатопоточних обчислень та зручні високорівневі структури даних. Програмний код на Python організовується у функції та класи, які можуть об'єднуватись у модулі, а вони у свою чергу можуть бути об'єднані у пакети. Python зазвичай використовується як інтерпретований, але може бути скомпільований в байт-код Java та в MSIL (в рамках платфори .NET).

По продуктивності інтерпретований Python схожий на всі інші подібні мови, але можливість компіляції в байт-код дозволяє досягти більшої продуктивності.

У порівнянні з Ruby та деякими іншими мовами, у Python відсутня можливість модифікувати вбудовані класи, такі як int, str, float, list та інші.

У Python є глобальне блокування інтерпретатора (GIL) - при своїй роботі основний інтерпретатор постійно використовує велику кількість потоково-небезпечних даних. В основному це словники, в яких зберігаються атрибути об'єктів, і звернення до зовнішнього коду, тому, щоб уникнути руйнування цих даних при спільній модифікації з різних потоків, перед початком виконання кількох інструкцій (зазвичай сто) потік інтерпретатора захоплює GIL, а після закінчення звільняє.

Python — одна з основних мов програмування, які застосовують у галузі машинного навчання та штучного інтелекту (Machine Learning та Artificial Intelligence). Наприклад, бібліотека з відкритим вихідним кодом TensorFlow, створена дослідницькою командою Google Brain, написана за допомогою Python. Google використовує цю бібліотеку для програмування та навчання нейронних мереж, які використовуються для вивчення штучного інтелекту.

Ще одна відома бібліотека – scikit-learn. Вона написана на Python із включеннями Cython — статично типізованого компілюваного підмножини Python. Бібліотека scikit-learn застосовується у дослідженнях штучного інтелекту, на навчання інженерів machine learning, на управління промисловими системами.

У Python є кілька потужних та популярних бібліотек, які призначені для роботи з великими даними: аналізу, візуалізації, прогнозування тенденцій. Наприклад, бібліотека з відкритим вихідним кодом SciPy включає модулі для математичних, інженерних та наукових обчислень. Matplotlib – одна з найпопулярніших бібліотек для візуалізації даних. Бібліотека PANDAS використовується для аналізу інформації.

Це не всі сфери застосування Python. Цю мову використовують для створення десктопних програм, розробки ігор та 3D-графіки, програм для обробки аудіо, відео та зображень тощо.

3.2 Бібліотека комп’ютерного зору OpenCV

OpenCV була заснована Intel у 1999 році і в даний час підтримується Willow Garage. OpenCV використовує BSD-подібну ліцензію, тому вона безкоштовна для некомерційних та комерційних програм.

Він може працювати в операційних системах Linux, Windows та Mac OS. Він легкий і ефективний - він складається з ряду функцій C і невеликої кількості класів C++, а також надає інтерфейси для таких мов, як Python, Ruby та MATLAB, і реалізує безліч загальних алгоритмів обробки зображень та комп'ютерного зору.

OpenCV має кросплатформовий API середнього та високого рівня, який включає більше 500 функцій C. Це не залежить від інших зовнішніх бібліотек - хоча деякі зовнішні бібліотеки також можуть бути використані.

OpenCV надає прозорий інтерфейс для Intel® Integrated Performance Primitives (IPP). Це означає, що якщо бібліотеки IPP, оптимізовані для конкретного процесора, OpenCV автоматично завантажить ці бібліотеки під час виконання.

OpenCV (Open Source Computer Vision) — бібліотека комп'ютерного зору з відкритим кодом) — бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізовано на C/C++. Ця бібліотека дуже популярна завдяки своїй відкритості та можливості безкоштовно використовувати як у навчальних, так і комерційних цілях.

Фактично OpenCV – це набір типів даних, функцій та класів для обробки зображень алгоритмами комп'ютерного зору.

Основні модулі бібліотеки:

cxcore - ядро

- базові операції над багатовимірними числовими масивами

- матрична алгебра, математичні ф-ції, генератори випадкових чисел

- запис/відновлення структур даних у/з XML

- базові функції 2D графіки

CV — модуль обробки зображень та комп'ютерного зору

- базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів тощо)

- аналіз зображень (вибір відмітних ознак, морфологія, пошук контурів, гістограми)

- аналіз руху, стеження за об'єктами

- Виявлення об'єктів, зокрема осіб

- калібрування камер, елементи відновлення просторової структури

Highgui — модуль для введення/виводу зображень і відео, створення інтерфейсу користувача

— захоплення відео з камер та відео файлів, читання/запис статичних зображень.

— функції для організації простого UI (всі демо-програми використовують HighGUI)

Cvaux - експериментальні та застарілі функції

- Просторів. зір: стерео калібрація, саме калібрація

- Пошук стерео-відповідності, кліки у графах

- Знаходження та опис рис особи

CvCam - захоплення відео

- дозволяє здійснювати захоплення відео з цифрових відеокамер (підтримка припинена і в останніх версіях цей модуль відсутній).

За допомогою OpenCV з картинкою можна працювати як у графічному редакторі: обрізати, збільшувати чи зменшувати, обертати. В основному програмісти використовують цю можливість для попередньої підготовки картинки перед її розшифровкою, наприклад, обрізають непотрібні частини.

Картинку можна зробити у відтінках сірого або повністю чорно-білого. Це важливо для алгоритмів розпізнавання, які працюють із знебарвленими зображеннями. Можна змінювати колірний тон, розмивати, згладжувати чи геометрично змінювати картинку.

Для розпізнавання елементів OpenCV використовуються обриси об'єктів, сегментація за кольорами, вбудовані методи розпізнавання, які можна настроювати залежно від об'єкта і чутливості алгоритму.

3.3 Бібліотека NumPy

NumPy - це бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом із великою бібліотекою високорівневих (і дуже швидких) математичних функцій для операцій із цими масивами.

NumPy можна як вільну альтернативу MATLAB. Мова програмування MATLAB зовні нагадує NumPy: обидва інтерпретовані, обидва дозволяють виконувати операції над масивами (матрицями), а не над скалярами. Перевага MATLAB є великою кількістю пакетів («тулбоксів»), наприклад, Simulink (англ.). Для NumPy також існують подібні «пакети», наприклад, бібліотека SciPy надає більше MATLAB-подібної функціональності, бібліотека Matplotlib дозволяє створювати графіки в стилі MATLAB. І MATLAB, і NumPy на вирішення основних завдань лінійної алгебри використовують код, заснований на коді бібліотеки LAPACK.

Бібліотека написана мовами C та Fortran. Це компілювані мови (мови програмування, текст яких перетворюється на машинний код — набір інструкцій

для конкретного типу процесора. Перетворення відбувається за допомогою спеціальної програми-компілятора, завдяки ньому обчислення компілюваними мовами відбуваються швидше), на яких обчислення виробляються набагато швидше і ефективніше, ніж мовами, що інтерпретуються (мови програмування, які не заточені під конкретний тип процесора і можуть бути запущені на різних типах пристроїв). До цих мов належить і сам Python.

Звичайно, крім двовимірних масивів, бібліотека NumPy обробляє й інші з різною кількістю осей. Цю варіативність позначають числом N, як будь-яку змінну математичної задачі. Тому зазвичай кажуть, що NumPy працює із N-мірними масивами даних.

Як і сам Python, бібліотека NumPy відрізняється простотою у вивченні та використанні. Спочатку роботи досить освоїти концепцію масивів. Наприклад, в базових арифметичних обчисленнях є спосіб обробки масивів, який називають трансляцією або broadcasting. Якщо в масиві величини вказані милях, а результат потрібно отримати в кілометрах, можна помножити його на просте число 1,6 (скалярну величину). NumPy приймає самостійне рішення помножити на задане число кожен елемент у масиві, і користувачеві не доводиться прописувати окрему команду.

3.4 Пакетний менеджер Anaconda

Віртуальне оточення - це ізольоване середовище (у нашому випадку це оточення Python), яке дозволяє нам використовувати певні версії додатків.

Anaconda – вільний open source дистрибутив для мов програмування Python та R з відкритим кодом для обробки даних великого обсягу, побудови аналітичних прогнозів та наукових обчислень. Розробники дистрибутива мають на меті спростити управління та використання пакетів. Версії пакетів контролюються системою керування пакетами conda.

Основна проблема полягає в тому, що pip, easy_install та virtualenv орієнтовані на Python. Ці інструменти ігнорують бібліотеки залежностей, які реалізовані з використанням інших мов. Наприклад, XSLT, HDF5, MKL та інші, які не мають setup.py у вихідному коді та не встановлюють файли в каталог site-packages.

Conda ж здатна керувати пакетами як Python, так C/ C++, R, Ruby, Lua, Scala та інших. Conda встановлює двійкові файли, тому роботу з компіляції пакета самостійно виконувати не потрібно (порівняно з pip).

3.5 Бібліотека dlib

Dlib – це бібліотека C++ для машинного навчання, яка містить безліч часто використовуваних алгоритмів для машинного навчання. Dlib може допомогти вам створити багато складних програм для машинного навчання, які допоможуть вирішити практичні завдання. В даний час Dlib широко використовується в промислових та академічних областях, включаючи роботів, вбудовані пристрої, мобільні телефони та великі високопродуктивні обчислювльні середовища. Dlib з відкритим вихідним кодом та безкоштовно.

Серед особливостей можна виділити:

- повна документація;
- високоякісний переносний код;
- забезпечує велику кількість алгоритмів машинного навчання / обробки зображень.

Dlib - це кроссплатформове програмне забезпечення загального призначення бібліотека, написана мовою програмування C++. На його дизайн сильно вплинули ідеї з проектування за контрактом та компонентної розробки програмного забезпечення. Таким чином, це насамперед набір незалежних програмних компонентів. Це програмне забезпечення з відкритим вихідним кодом, випущене за ліцензією Boost Software.

З моменту початку розробки в 2002 році Dlib розширився і тепер включає широкий спектр інструментів. Станом на 2016 рік він містить програмні компоненти для роботи з мережею, потоками , графічними інтерфейсами користувача , структурами даних , лінійними алгебра , машинне навчання , обробка зображень , інтелектуальний аналіз даних , XML та аналіз тексту, численна оптимізація. В останні роки більшість розробок була зосереджена на створенні широкого набору інструментів статистичного машинного навчання, і в 2009 році Dlib був опублікований в Journal of Machine Learning Research. З тих пір він використовувався в різних областях.

ВИСНОВОК

Отже, виконавши поставлені завдання, у дипломній роботі розібрано види різних нейронних мереж, їх побудова та технології, які вони використовують

У другому розділі виконано огляд теоретичного і практичного використання концепції «розумного» міста, технології та принципи роботи системи.

Також було розглянуто використання різних топологій, та технологій у імплементації Інтернету Речей, як однієї з основ розумного міста. Була досліджена проблематика та наслідки реалізації розумного міста.

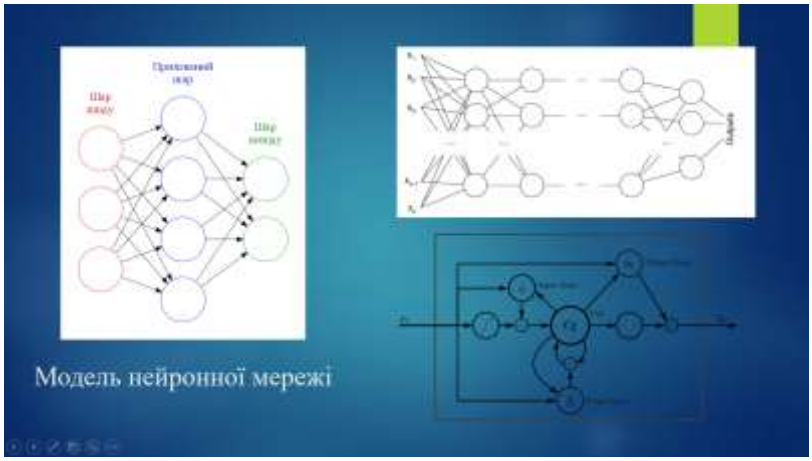
У практичній частині виконано аналіз бібліотек та технологій, які були використані мною. Було проведене порівняння та ґрунтовна оцінка різних бібліотек. В додаток до цього, була написана програма, що використовує нейромережу для розпізнавання обличчя.

ПЕРЕЛІК ПОСИЛАНЬ

1. Курчеева Г. И. Разработка процессной модели «Умный город». [Електронний ресурс] / Г. И. Курчеева, Г. А. Клочков – Режим доступу до ресурсу: https://cyberleninka.ru/article/n/razrabotka-protsessnoy-modeli-umnyy-gorod/viewer.
2. Розумне місто – реальність сучасного світу [Електронний ресурс] – Режим доступу до ресурсу: https://deps.ua/ua/knowegable-base/reference-information/67697.html.
3. IoT in cities market, MarketsandMarkets [Електронний ресурс] – Режим доступу до ресурсу: https://www.marketsandmarkets.com/Market-Reports/iot-smart-cities-market-215714954.html.
4. IEEE, Coverage and Capacity Analysis of Sigfox, LoRa, GPS, and NB-IoT[Електронний ресурс] – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/8108666
5. Технология узкополосного Интернета вещей в мобильной связи [Електронний ресурс] – Режим доступу до ресурсу: http://1234g.ru/novosti/uzkopolosnyj-internet-veshchej-nb-iot
6. IoT in cities market, MarketsandMarkets [Електронний ресурс] – Режим доступу до ресурсу: https://www.marketsandmarkets.com/Market-Reports/iot-smart-cities-market-215714954.html.
7. Корнієвич Л. І. Розумне місто. Розповідаємо, що таке «smart city» та чим воно небезпечне [Електронний ресурс] / Людмила Ігорівна Корнієвич. – 2019. – Режим доступу до ресурсу: https://hromadske.ua/posts/liho-z-rozumu-rozpovidayemo-sho-take-smart-city-ta-chim-vono-nebezpechne.
8. Хабр, Блог компании МТС. NB-IoT как он работает? [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/company/ru_mts/blog/430496/

9. FutureNow, Що таке Інтернет Речей? Все, що треба знати про IoT прямо зараз [Електронний ресурс] – Режим доступу до ресурсу: https://futurenow.com.ua/shho-take-internet-rechej-vse-shho-potribno-znaty-pryamo-zaraz/
10. Макаренко К. В. «Умный город»: стандарты, проблемы, перспективы развития [Електронний ресурс] / К. В. Макаренко, В. О. Логиновская. – 2019. – Режим доступу до ресурсу: https://cyberleninka.ru/article/n/umnyy-gorod-standarty-problemy-perspektivy-razvitiya/viewer.
11. IBM. What are neural networks? [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/cloud/learn/neural-networks
12. Pathmind. A beginners guide to neural networks and deep learning. [Електронний ресурс] – Режим доступу до ресурсу: https://wiki.pathmind.com/neural-network
13. SAS. Neural networks – What are they and why do they matter? [Електронний ресурс] – Режим доступу до ресурсу: https://www.sas.com/ru_ru/insights/analytics/neural-networks.html
14. Elsevier – Neural networks [Електронний ресурс] – Режим доступу до ресурсу: https://www.journals.elsevier.com/neural-networks
15. Apeps – Як працюють нейронні мережі? [Електронний ресурс] – Режим доступу до ресурсу: http://apeps.kpi.ua/neural-networks/en

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



Аналіз використаних бібліотек

- OpenCV - це open source бібліотека комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень. Широко використовується у таких мовах як C, C++, Python та Java.
- Dlib — це сучасний набір інструментів C++, що містить алгоритми машинного навчання та інструменти для створення складного програмного забезпечення на C++ для вирішення реальних проблем.
- Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами.

Приклад роботи програми

Приклад коду програми

Висновок

Отже, виконавши поставлені завдання, у дипломній роботі розібрано види різних нейронних мереж, їх побудова та технології, які вони використовують. У другому розділі виконано огляд теоретичного і практичного використання концепції «розумного» міста, технології та принципи роботи системи. . Також була розглянуто використання різних топологій, та технологій у імплементації Інтернету Речей, як однієї з основ розумного міста. Була досліджена проблематика та наслідки реалізації розумного міста. У практичній частині виконано аналіз бібліотек та технологій, які були використані мною. Було проведене порівняння та грунтовна оцінка різних бібліотек. В додаток до цього, була написана програма, що використовує нейромережу для розпізнавання обличчя.

Апробація результатів

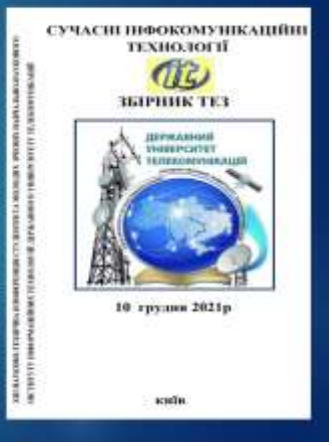
Опубліковано тезу. XIII НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНІ ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ». Ст. 11

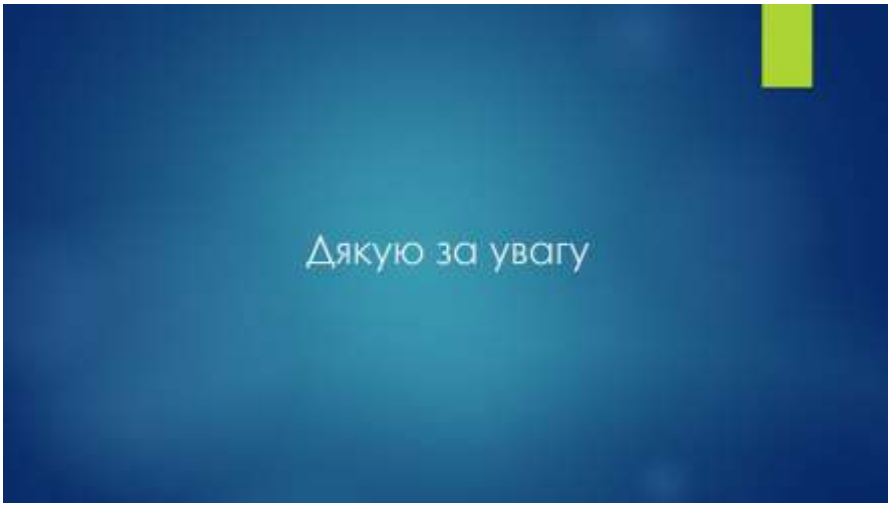
Опубліковано статтю.

Журнал «Зв'язок».

«Методологія проектування промислових бездротових систем Інтернету речей».

Назаренко О.М.
Трамбовецький О.А




Дякую за увагу

ДОДАТОК А

```python
import face_recognition
import cv2
import numpy as np

video_capture = cv2.VideoCapture(0)

# Завантажуємо фотографію щоб почати її розпізнавати
obama_image = face_recognition.load_image_file("files/obama.jpg")
obama_face_encoding = face_recognition.face_encodings(obama_image)[0]

oleksandr_image = face_recognition.load_image_file("files/Oleksandr.jpg")
oleksandr_face_encoding = face_recognition.face_encodings(oleksandr_image)[0]


# Масив с кодуваннями облич
known_face_encodings = [
    obama_face_encoding,
    oleksandr_face_encoding
]
known_face_names = [
    "Barack Obama",
    "Oleksandr T"
]

# Ініціалізуються змінні
```

```python
# Ініціалізуються змінні
face_locations = []
face_encodings = []
face_names = []
process_this_frame = True

while True:
    # Беремо кожний кадр відео
    ret, frame = video_capture.read()

    # Змінюємо розмір до 1/4 для більш швидкого розпізнавання
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

    # Конвертуємо із BRG в RGB для бібліотеки з розпізнавання
    rgb_small_frame = small_frame[:, :, ::-1]

    # Обробляється тільки 1 кадр
    if process_this_frame:
        # Знаходяться усі обличчя у наявному кадрі
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

        face_names = []
```

```
    for face_encoding in face_encodings:
        # Знайти у масиві потрібне обличчя
        matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
        name = "Unknown"

        face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = known_face_names[best_match_index]

        face_names.append(name)

process_this_frame = not process_this_frame


# Показую результат
for (top, right, bottom, left), name in zip(face_locations, face_names):
    # Збільшуємо розмір зображення
    top *= 4
    right *= 4
    bottom *= 4
    left *= 4
```

```
        # Малюю квадрат на обличчі
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

        # Малюю місто для тексту під квадратом
        cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

    # Відображаю результат
    cv2.imshow('Video', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video_capture.release()
cv2.destroyAllWindows()
```