КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультету радіофізики, електроніки та комп'ютерних систем
Кафедра комп'ютерної інженерії

Алгоритми визначення аномалій в IoT системах

Кваліфікаційна робота бакалавра
студента 4 року навчання
Спеціальність: 123 «Комп'ютерна інженерія»
Богдана ЮРЧЕНКА

Науковий керівник,
асистент Юрій ЮРЧИК

Рецензент
канд. фіз.-мат. наук Олександр СУДАКОВ,
доцент кафедри медичної фізики

До захисту допускаю

Завідувач кафедрою
канд. фіз.-мат. наук Юрій БОЙКО,
доцент кафедри комп'ютерної інженерії

Ухвалено на засіданні кафедри "____"_____2022 р., протокол № ___

КИЇВ 2022

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра містить 44 сторінки, 23 рисунки, 5 формул, 9 використаних джерел.

Метою роботи є практичний аналіз та порівняння алгоритмів пошуку аномальних даних в Інтернеті речей. Для покращення алгоритмів обробки даних в IoT мережах, оскільки на даному етапі розвитку Інформаційних технологій, пристрої що підпадають під визначення Інтернету речей, не є абсолютно захищеними.

Безпека пристрою IoT не може бути визначена стандартними критеріями, оскільки якщо узагальнити IoT-безпеку як спектр вразливості пристрою коливається від абсолютно незахищених пристроїв, що не мають захисних функцій, протоколів або механізмів, до високо захищених систем із декількома рівнями захисту.

Головною метою роботи є застереження від помилкового визнання аномальних обчислювальних даних, що будуть отримані від сенсорів в Інтернеті речей за правдиві.

Оскільки Інтернет речей є перспективною та новітньою технологією і помилки на окремому сенсорі можуть призвести до повної непрацездатності системи.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ВИЗНАЧЕНЬ

Гетерогенна мережа - комп'ютерна мережа, що з'єднує персональні комп'ютери та інші пристрої з різними операційними системами або протоколами передавання даних.

IoT – концепція мережі передачі даних між фізичними об’єктами, обладнана вбудованими засобами та технологіями для взаємодії друг з другом або з зовнішнім середовищем в реальному часі.

Кібератака – спроба реалізації кіберзагрози, тобто будь-яких обставин або подій, що можуть бути причиною порушення політики безпеки інформації або завдання збитків автоматизованій системі.

Скомпрометованість – можливість використати недоліки проти особи або системи в її шкоду.

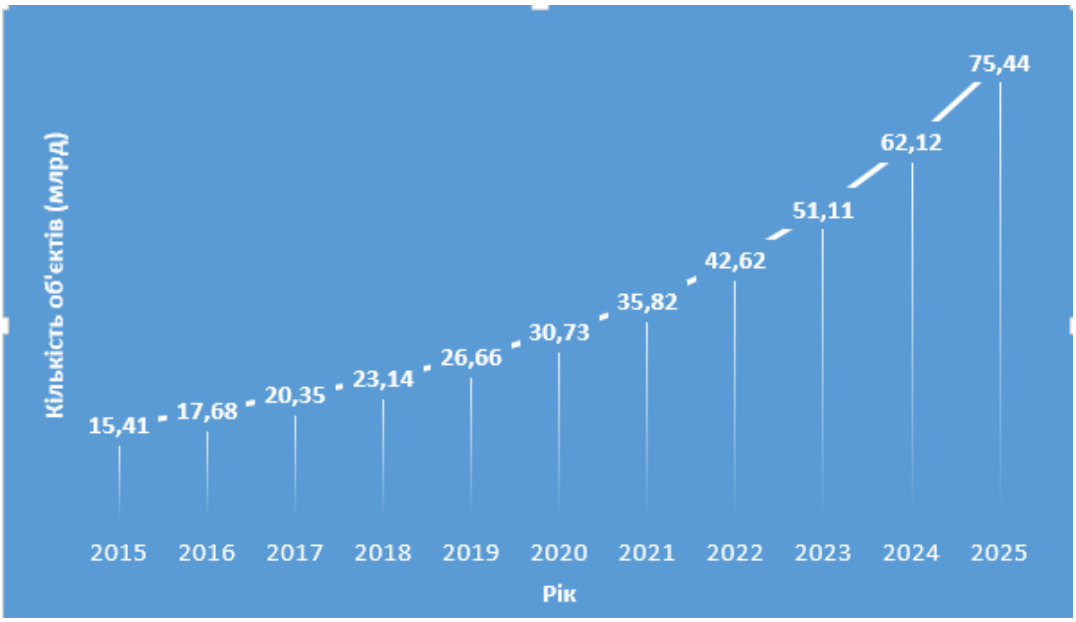
РОЗДІЛ 1. ІНТЕРНЕТ РЕЧЕЙ

1.1 Перспективи розвитку IoT

Інтернет речей за одним з визначень являє собою мережу для передачі даних між фізичними об’єктами обладнану вбудованими засобами та технологіями для взаємодії один з одним або навколишнім середовищем у реальному часі. Інтернет речей це одна з найперспективніших та еволюційних технологій яка з’явилась відносно нещодавно.

В Інтернеті речей сполучені між собою телефони, автомобілі, комп’ютери, датчики які збирають і обмінюються інформацією один з одними. Використовуючи дану технологію, більшість компаній можуть поліпшити свою продуктивність завдяки використанню Інтернету речей. В першу чергу завдяки збільшенню ефективності та покращенню рівня безпеки.

Хоча Інтернет речей все ще на ранній стадії розвитку, але цей сектор швидко розвивається. Компанія BI Intelligence прогнозує, що до 2020 року кількість підключених об’єктів досягне 30 мільярдів у всьому світі, і майже 6 трильйонів доларів буде витрачено на рішення IoT протягом наступних п’яти років (рис 1.1).



“ Рисунок 1.1. – Прогнозований розвиток технології IoT”

Згідно зі звітом іншої компанії Research and Markets, до 2021 року заробіток від програмного забезпечення IoT досягне 379 мільярдів доларів, а прогнозований обсяг світового ринку Інтернету Речей складе 9,1 трлн доларів в тому ж році. Ці чинники стимулюють провідні світові компанії до того, щоб розвивати цей напрямок бізнесу.

1.2 Архітектура Інтернету речей

Розглянемо ширше поняття Інтернету речей та його архітектуру.

Існує багаторівнева архітектура Iot. У цій архітектурі є чотири рівні : рівень пристрою, мережевий рівень, підтримки сервісу та додатків, а також рівень додатків.[3] Крім цих рівнів в [3] запропонували два додаткових рівні: управління та безпеки.

“Рисунок 1.2. – Еталонна модель IoT “[1]

На (рис. 1.2) показані всі рівні в запропонованій архітектурі IoT. Основні рівні: пристрою, мережі, підтримки додатків, додатків, – це функціональні рівні. Іншими словами, ці рівні використовуються для збору, передачі та обробки даних у мережах IoT. Інші рівні беруть участь у функціональних можливостях кожного рівня з метою управління та безпеки.

У переліку функціональних рівнів є рівень пристрою. На цьому рівні існують пристрої IoT, такі як розумні лічильники електроенергії та датчики,

носії даних та інші пристрої. Цей рівень також містить шлюзи, які є дуже важливим компонентом у системах IoT.

Шлюзи IoT – це пристрої, які отримують та збирають дані з датчиків у межах приймання їх діапазону та передають протоколи датчиків, дані датчиків обробки та дані маршрутів, як всередину, так і зовні.

[4]Рекомендація Y.2067 закріплює вимоги до шлюзів IoT, які зазвичай розподіляються на три категорії:

- Шлюз потрібен для підтримки комунікаційного моста між пристроями та комунікаційні мережі.

- Шлюз повинен підтримувати зв’язок принаймні з однією програмою.

- Шлюз рекомендовано підтримує кілька комунікаційних технологій для взаємодії з мережами та пристроями. Повинен мати можливість покращувати можливості комунікаційних інтерфейсів.

Мережевий рівень в архітектурі IoT забезпечує можливість передачі даних від об'єктів спостереження і шлюзів до хмари за допомогою Інтернету. Шлюзи, а також деякі об'єкти спостереження мають можливість підтримувати Інтернет-протоколи, такі як Ipv4, Ipv6.[7]

Також існує рівень підтримки сервісу та додатків, який як і раніше, подібний до транспортного та сеансового рівня у поєднанні між системами взаємодії OSI. Він контролює тип передачі, якості та порт призначення на іншій стороні з’єднань. Іншими словами, це допомагає переносити дані спостереження до потрібної програми на рівні додатків, для їх подальшої обробки.[6]

Рівень додатка – це місце, де розташовані програми, завдяки яким ми керуємо пристроями і Інтернеті речей. Список деяких програм IoT містить в себе: розумний транспорт, розумні будівлі, розумне житло та інші.

1.3 Характеристики IoT

В даному розділі розглянемо саме ті чинники що роблять Інтернет речей перспективною технологією.

Існують три характеристики IoT, що робить IoT майбутнім Інтернету.[8]

1. Повсюдне сприйняття. Інтернет речей має здатність охоплення фізичного та віртуального світів та сприйняття їх за рамками людського сприйняття (оптичного, тактильного, звукового). Різноманітність та чутливість датчиків можуть подолати межі, що існують для людини та призвести до об’єднання віртуального та фізичного світів з метою поліпшення якості життя. Guardian Angels for a Smarter life – це проект, який фінансується ЄС та спрямований на створення автономних і розумних особистих супутників, які можуть допомагати і захищати людей протягом усього життя, від дитинства до старості. Вивчення об’єктів з нульовим енергоспоживанням може використовуватися для реалізації та моніторингу фізичного стану людей упродовж цілого дня та обміну даними з лікарями. Об'єкти екологічного спостереження будуть використовуватися для вивчення навколишнього середовища та попередження людей у разі небезпечних умов. Об'єкти емоційного спостереження будуть використовуватися для вивчення людських емоцій та реагування на них.

2. Мережа мереж. IoT передбачає кілька різних мереж. Іншими словами, це гетерогенна мережа, що охоплює GSM, CDMA, WCDMA та IP мережі.

3. Інтелектуальна обробка. Деякі об’єкти в Інтернеті речей розробляються як інтелектуальні, подібні до людини. Вони можуть виконувати ті самі задачі, що й людина, але основна перевага перед людьми – це час та швидкість обробки даних. Об’єкти IoT можуть обробляти величезну

кількість отриманих даних, набагато швидше та якісніше ніж це можуть зробити люди.

1.4 Класифікація Інтернету речей

Відповідно до стандартів Міжнародного союзу електрозв'язку (Interntional Telecommunication Union) існують певні визначення, що описують об'єкти, які належать до IoT [3]:

- Інтернет речей (IoT): На основі вже чинних та розвинутих інформаційно-комунікаційних технологій створена глобальна інфраструктура, що забезпечує можливість надання послуг шляхом з'єднання між собою як фізичних, так і віртуальних речей.
- Річ: Відповідно до Інтернету речей уособлює предмет фізичного чи інформаційного розділу, який можна ідентифікувати в мережі.
- Пристрій: Відповідно до Інтернету речей уособлює предмет або елемент обладнання, що має певні властивості такі як: можливість вимірювання, передачі, обробки, зберігання, введення даних.

У [2] описують Інтернет речей як сукупність фізичних об'єктів, контролерів, виконавчих механізмів та мережі. Це трактування описує саму суть Інтернету речей. Примірник IoT складається з набору фізичних об'єктів, кожен з яких:

- містить мікроконтролер, що забезпечує інтелектуальність;
- містить датчик, що вимірює будь-який фізичний параметр, і/або виконавчий механізм, що спрацьовує від будь-якого фізичного параметра;
- має можливість комунікації через Інтернет або будь-яку іншу мережу.

Єдине що не входить до цієї формули це ідентифікація кожної речі.

Унікальною особливістю Інтернету речей, в порівнянні з іншими мережами є велика кількість пристроїв та фізичних предметів, що відсутні або відмінні від інших пристроїв в інших мережевих системах.

“Рисунок 1.4.1 – Типи пристроїв і їх взаємозв’язок з фізичними речами”[3]

На рис. 1.4.1, адаптованому з рекомендації Y.2060[3], зображені типи пристроїв в моделі MCE-T. Модель розглядає IoT як мережу пристроїв, тісно пов'язаних з речами. Сенсорні й виконавчі пристрої взаємодіють з фізичними речами в навколишньому середовищі. Пристрої збору даних отримують показники від фізичних речей або зберігають дані на фізичних речах, за допомогою пристроїв перенесення даних чи носіїв, що приєднані або пов’язані з фізичним об’єктом.

В Рекомендації Y.2060[3] відзначається, що в Інтернеті речей використовують перелік технологій, завдяки яким здійснюється передача та взаємодія між пристроями в мережі. Це пристрої збору, збереження, передачі та носії даних.

Приклади технологій і пристроїв з якими вони використовуються:

- Радіочастотна ідентифікація (RFID)-бірки, або радіопозначки. Здійснюється за допомогою закріплення за об’єктом спеціальних міток, що несуть ідентифікаційну та іншу інформацію.
- Інфрачервоні мітки, використовуються в військових цілях, медичних та інших середовищах, де потрібно відстежувати розташування і

переміщення. Прикладом таких міток є: інфрачервоні нашивки на формі, ідентифікаційні бейджі, з яких можливо зчитати інформацію, нанесені позначки на предмети за допомогою яких ідентифікується предмет. Бейджі можуть використовуватись для ідентифікації особи при проходженні через портал чи спеціальну рамку, що використовують на підприємствах також для забезпечення обмеженого доступу до будівель. Принцип дії заснований на принципі приймання відбитого від предмета або мітки інфрачервоного випромінювання спрямованого випромінювачем датчика.

- Штрих-коди і QR-коди належать до датчиків що сприймаються оптично. Принцип дії полягає в закодуванні інформації під виглядом послідовних чорних і білих смужок або у вигляді візерунка (QR-коду), що потім зчитується певними технічними методами.

- Медичні імпланти, які використовують електропровідні властивості людського тіла. В ході комунікації між імплантом і поверхнею, гальванічна пара передає сигнали з імпланта на електроди, виведені на шкіру. Ця схема використовує дуже мало енергії, що дозволяє знизити розмір і складність імплантованого пристрою.

1.5 Датчики

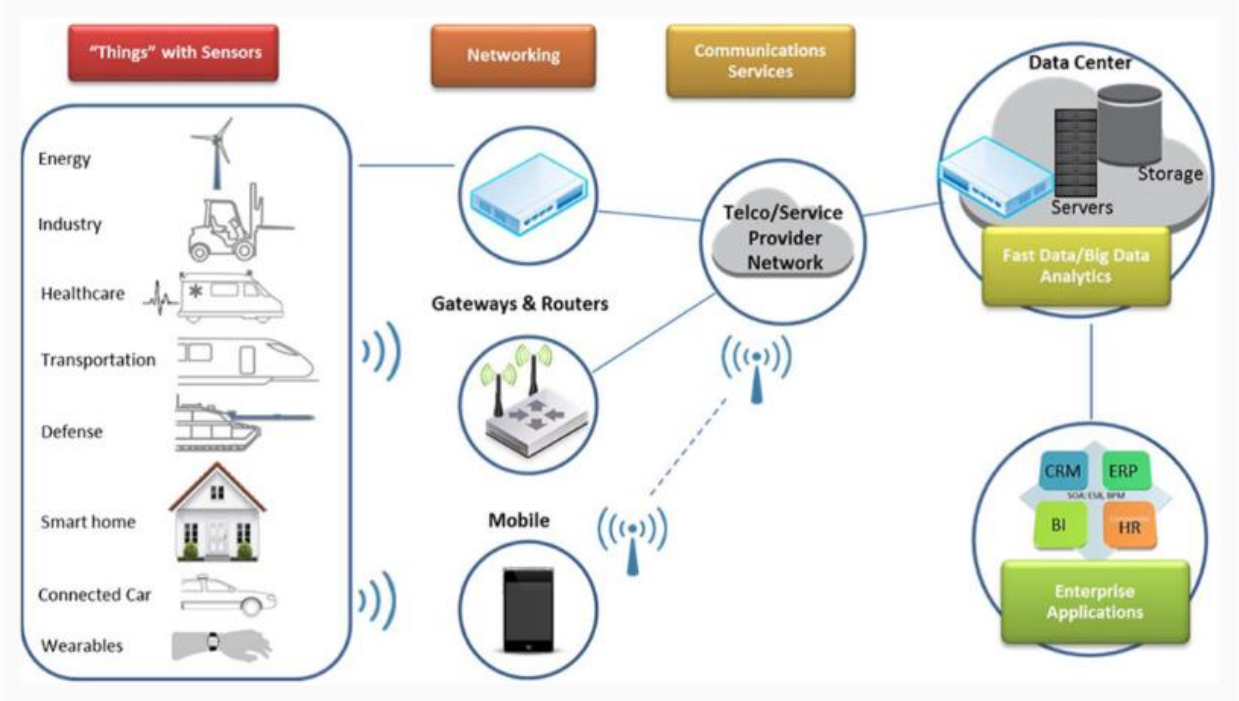
Датчики або пристрої збору даних є основою Інтернету речей. Саме датчики збирають інформацію про роботу різних пристроїв в мережі та передають її далі для обробки.

Датчики можуть вимірювати зовсім різні показники від температури повітря чи його рівня забрудненості до інфрачервоного випромінювання.

Для цього їх оснащують різними чутливими елементами, наприклад світлочутливими діодами чи металевими пластинами, які змінюють свої властивості в залежності від навколишнього середовища.

Після збору інформації її потрібно відправити для цього в датчики вбудовують передаваний пристрій або модуль. В IoT це зазвичай модуль бездротового зв’язку, наприклад: Wi-Fi, Bluetooth, NFC.

Іноді для зменшення затрат декілька датчиків приєднують до одного модуля чи пристрою для передачі.



“Рисунок 1.4 – Місце датчиків в Інтернеті речей”[5]

Датчики, що збирають інформацію збирають її в аналоговому вигляді. Такі дані не можна передавати мережею, тому їх перетворюють в цифрові.

Для цього дані ділять на певні ділянки. На кожній ділянці потрібно призначити конкретні значення.

Наприклад, візьмемо пристрій виміру температури. Коли температура змінюється, то металева пластина, що в ньому вбудована, змінює свою провідність залежно від температури. Для людини не зрозуміло на скільки змінилась температура залежно від провідності, для розуміння людиною придумали певні шкали вимірювання температури. В нашому випадку це градуси за Цельсієм (С°). Дана інформація є аналоговою. А от для того, щоб передати дані саме температури, а не зміни провідності в пристрій вбудовують перетворювач зі шкалою який розуміє яка відбулася зміна температури відносно зміни провідності. Даний перетворювач має назву (АЦП), що і є аналогово-цифровим перетворювачем, через який у вимірах присутній так названий шум перетворювача, що потрібно враховувати при перевірці отриманих даних. Потім вже отриману та перетворену на цифрову інформацію датчик може передавати.

1.6 Проблеми IoT мереж

В теперішній час, перед Інтернетом речей стоїть ряд проблем, які суттєво стримують його потенціал та сповільнюють його розвиток, серед них: несанкціонований доступ до підключених пристроїв, небезпека щодо конфіденційності інформації, не відповідна робота датчиків.

Ці та інші технічні питання продовжують залишатися невирішеними, а також виникають нові складності в області політики, законодавства та подальшого розвитку технології.

Наразі існує декілька ключових проблем пов’язаних з технологією Інтернету речей, це: безпека; конфіденційність; інтероперабельність і стандарти; соціальні та економічні проблеми.

Головна проблема Інтернету речей - це питання безпеки, до 80 відсотків пристроїв не захищені. Це стосується як інформаційного впливу, тобто заміна при передачі даних чи їх зберігані, так і відсутності чітких алгоритмів пошуку аномальностей даних при їх вимірюванні сенсорами. У сегменті промислового Інтернету речей проблема вирішується радикальним чином: жорсткі правила і нормативи, а також спеціальні протоколи безпеки. Для критичних пристроїв буде необхідна абсолютна надійність мережі, адже найменший збій може призвести до ризику для людей та конфіденційності даних.

Рішення безпеки, такі як виявлення вторгнень, використовуються для мінімізації ризику того, що будь-який компонент або дані в інформаційній системі будуть скомпрометовані. Це означає, що, всупереч великій кількості досліджень та рішень з питань безпеки, для захисту систем IoT, все ще існує ймовірність того, що об'єкт IoT буде скомпрометований, а передані дані будуть неправдиві.

Питання безпеки пристроїв та аналізу даних об’єктів Інтернету речей займає ключові позиції. На разі існує недостатня кількість досліджень, які направлені на протидію наслідкам невірно отриманих даних.

Таким чином, дане дослідження являє собою спробу заповнити цей пробіл, додавши ще один рівень захисту до вже наявних, які розробляються для забезпечення ефективності виявлення аномальних даних та оптимального рівня безпеки систем IoT.

Проведено аналіз літератури за темою дослідження та чинних систем виявлення подій пов’язаних зі спотворенням даних. На основі проведеного аналізу виявлено відсутні ланки в загальних підходах аналізу аномальних даних в Інтернеті речей. Визначено завдання, що ставить за мету заповнити відсутній підхід до ідентифікації аномальних даних в IoT, а саме розробки алгоритму виявлення аномальних даних в Інтернеті речей. Ці аномальності напряму пов’язані з некоректною роботою датчиків.

РОЗДІЛ 2. АЛГОРИТМИ ПОШУКУ АНОМАЛІЙ

2.1 Аномальність даних

На даному етапі розвитку Інформаційних технологій, пристрої що підпадають під визначення Інтернету речей, не є абсолютно захищеними.

Безпека пристрою IoT не може бути визначена стандартними критеріями оскільки якщо узагальнити IoT-безпеку як спектр вразливості пристрою коливається від абсолютно незахищених пристроїв, що не мають захисних функцій, протоколів або механізмів, до високо захищених систем із декількома рівнями захисту.

З вдосконаленням методів нападів та підходів отримання несанкціонованого доступу, нові загрози безпеки мають бути попереджені в якомога коротший термін та з максимальною ефективністю, хоча виробники пристроїв та оператори мережі постійно реагують на вирішення нових загроз, далеко не всі підходи направлені на першопричину вразливості пристроїв IoT.

Існує також багато різних типів аномалій та багатьох різних проблемних подій пов’язаних зі спотворенням даних.

Основі типи аномалій та подій:

- Аномалія явної та неявної надлишковості – дана аномалія виникає через деякі чинники : похибки вимірювань датчика, невірного округлення отриманих даних. Також неможливо уникнути такого чинника як шум тобто вплив інших приладів на датчик чи помилках при перетворенні даних від аналогових до цифрових. Не можна і виключати несправність самого датчика.
- Аномалія вставки – дана аномалія вже проявляється під час запису даних які були отримані датчиком, або під час пересилання даних в

мережі. Вона являє собою додавання хибних даних до вже наявних, саме додавання нових даних, а не заміну наявних тим самим певним чином впливаючи на подальші розрахунки та алгоритми що будуть працювати з отриманими даними.

- Аномалія оновлення – дана аномалія спостерігається під час пересилання даних в мережі внаслідок несанкціонованого доступу до них. Її наслідком буде заміна присутніх даних на нові. В результаті чого дані будуть сфальсифіковані.
- Аномалія видалення – ця аномальність може бути проявом як навмисного, так і технічного впливу на отримані дані. Тобто в результаті неправильного запису даних чи ж прогалинам при передачі даних так можливо і при вимірі даних. Внаслідок неї отримані дані втрачаються через що можлива втрата повної картини за якою велось спостереження.

На разі в парадигмі Інтернету речей найпоширенішою атакою є використання аномальних даних, що призводить до спотворення даних. Даний тип аномальностей можна віднести як до Аномалій явної та неявної надлишковості якщо джерелом аномальностей буде відправник так і до Аномальностей оновлення якщо дані були надіслані вірні і в результаті пересилання змінились на неправдоподібні.

2.2 Алгоритми пошуку аномалій

В підході до розробки схем виявлення аномалій, слід враховувати унікальні проблеми, які має IoT. Наприклад, інтрузивні схеми виявлення, які потребують значно вищої обчислювальної потужності або споживання ресурсів – не підходять для використання в IoT.

Крім того, пристрої IoT є гетерогенними і генерують неоднорідні дані, які містять масиви великих даних. Таким чином, підходи до виявлення аномалій в IoT не повинні обмежуватися лише виявленням мережевих атак, замість цього вони повинні поглиблюватися та виявляти аномалії в отриманих даних або інформації що надходить від сенсорів.

Для виявлення аномалій в даних існує цілий ряд алгоритмів таких як:

- Статистичні тести – які в розумінні IoT застосовуються до кожної ознаки окремо й забезпечують вірність кожної змінної вимірювання.
- Ітераційні методи – на кожній ітерації ми прибираємо особливо змінені показники, хоча дані методи можуть бути досить затратними.
- Метричні методи – зважаючи на кількість публікацій на цю тему вони є найпопулярнішими. Інтуїтивно зрозуміло, якщо дані не правдиві, то числова різниця між аномальними даними буде набагато більша ніж між правдивими.
- Методи підміни задачі – коли одну задачу або в нашому випадку результат виміру пристрою розбивають на під задачі або іншими словами досліджують кожен аспект виміру.
- Метод машинного навчання – один з найефективніших так і найзатратніший метод, коли в результаті отриманих даних пристрій розуміє можливі дані та відразу відкидає ті які не входять в правдивий діапазон.
- Сукупність алгоритмів – використання одного алгоритму добре, проте можливе й використання декількох одразу або ще краще взяти найефективніші аспекти кожного з них та намагатися створити новий.

2.3 Опис досліджуваного алгоритму на основі Sliding Window

На основі вже наявного алгоритму Sliding Window та використання статистичних тестів було отримано певний алгоритм пошуку аномальностей даних.

Принцип його роботи полягає в розбиті всього проміжку (об’єму даних) на менші проміжки які перекривають один одного тобто якщо в нас є проміжок з і елементів. Його проміжки виглядатимуть так (n[0:i];n[1:i+1];n[k:i+k]), де n це номер проміжку, k – теперішній елемент, і – кількість елементів в проміжку.

Було досліджено проміжки по 3, 10, 30 та 60 елементів зроблено висновки, з урахування точності та завадостійкості обрано проміжки по 10 елементів. Розбивши весь діапазон даних на проміжки по 10 елементів.. Тобто по 10 хв і аналізуючи дані дійшли висновку що зміна за 10 хв не може бути більшою за 0.2 градуси по модулю в нашому випадку оскільки температура вимірювалась в приміщені. Далі було отримано формулу для пошуку можливого значення відхилення:

$$A = m + k \tag{2.3.1}$$

де A – максимальна амплітуда зміни температури на проміжку;

m – мінімально можлива амплітуда вимірювання температури;

k – шум сенсора АЦП.

Було враховано мінімально можливу амплітуду вимірювання, що дорівнює 0.1 градус до нього додано можливий шум АЦП який рівний 0.1 градуса ми отримуємо максимальну амплітуду зміни температури рівну 0.2 градуси. Відповідно до цього зміна більша за мінус 0.2 чи плюс 0.2 градуси

буде сприйматися як помилка. Умову правдивості даних обчислюють за формулою:

$$(Max(X(i-n, i)) - Min(X(i-n, i))) - A < 0 \qquad (2.3.2)$$

Де n - кількість елементів в масиві даних ;

i – теперішній номер елемента ;

A - максимальна амплітуда зміни температури на проміжку.

Як продемонстровано в формулі (2.3.2) на кожному проміжку визначалось максимальне та мінімальне значення. Було знайдено їх різницю і порівняно її з можливою зміною температури яка дорівнює 0.2 градуси. І з даних міркувань робився висновок присутності аномального значення на даному проміжку чи ні. Якщо різниця максимальної та мінімальної температури перевищувала 0.2 градуси то робився висновок, що на даному проміжку присутня похибка, а оскільки проміжки перекривають один одного існує можливість визначити в який саме момент часу було передано дані з похибкою.

Найголовнішою перевагою алгоритму на основі рухомого вікна є простота виконання самого алгоритму та мінімальність потрібних обчислювальних можливостей. Тому, що дослідження стосувалися саме даних датчика виміру температури, певні чинники для інших типів датчиків можуть змінюватись.

Але алгоритм має певні недоліки такі як неможливість працювання без попереднього аналізу. Мається на увазі статистичного дослідження датчика: області його вимірювання, точності, розмірності.

Для його коректної роботи потрібно також опрацювання досліджуваного середовища. Якщо це датчик тиску чи температури, то зміна за короткий проміжок часу не може бути надто великою в той самий час, якщо датчик буде відстежувати рівень забруднення повітря, то його

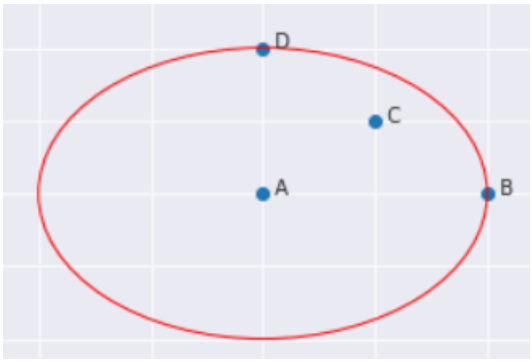
показники можуть змінюватись в десятки разів за відносно короткий проміжок, що для нашої версії алгоритму, яка налаштована на датчик виміру температури в приміщені буде підходити під визначення аномальності і відкидатися.

2.4 Опис досліджуваного алгоритму Локальний фактор викиду (Local Outlier Factor)

Локальний фактор викиду – це алгоритм знаходження аномалій в даних спостереження. В даному алгоритмі точка ідентифікується як викид на основі щільності сусідів. Він добре працює коли щільність даних не однакова на проміжку. [9]

В розумінні алгоритму Локального фактора викиду існують певні поняття:

K- відстань – це відстані між точкою та її найближчим сусідом. K-сусідів, включає набір точок, що лежать всередині кола чи на ньому. K-сусідів може бути більше або рівним кількості K.



“Рисунок 2.4.1 - Приклад сусідів”[9]

Наприклад в нас є 4 точки рис 2.4.1. Сусідами A будуть B, C, D. Значення K=2, але кількість точок в колі або на ньому = 3.

Відстань досяжності (RD) визначається як максимальна відстань K та відстань до найвіддаленішої точки, що являється сусідом. Також нею позначають відстань між сусідами. Визначається за формулою:

$$RD = \max[\text{відстань}(k1), \text{відстань}(k2), \text{відстань}(k),\ldots] \quad (2.4.1)$$

Щільність локальної досяжності (LRD) визначається як обернена рівність від середньої відстані досяжності до кожного з сусідів. Визначається за формулою:

$$\text{LRD} = \frac{1}{\frac{1}{k}(RD(k1)+RD(k2)+RD(k3)+..)} \quad (2.4.2)$$

де k - кількість сусідів;

RD- відстань до конкретного сусіда.

Фактор локальної досяжності (LOF) визначається за допомогою ділення середньої щільності сусідів на щільність нашої точки. Визначається за формулою:

$$\text{LOF} = \frac{\frac{1}{k}(LRD(k1)+LRD(k2)+LRD(k3)..)}{LRD(x)} \quad (2.4.3)$$

де k- кількість сусідів ;

x- досліджувана точка;

LRD – щільність локальної досяжності.

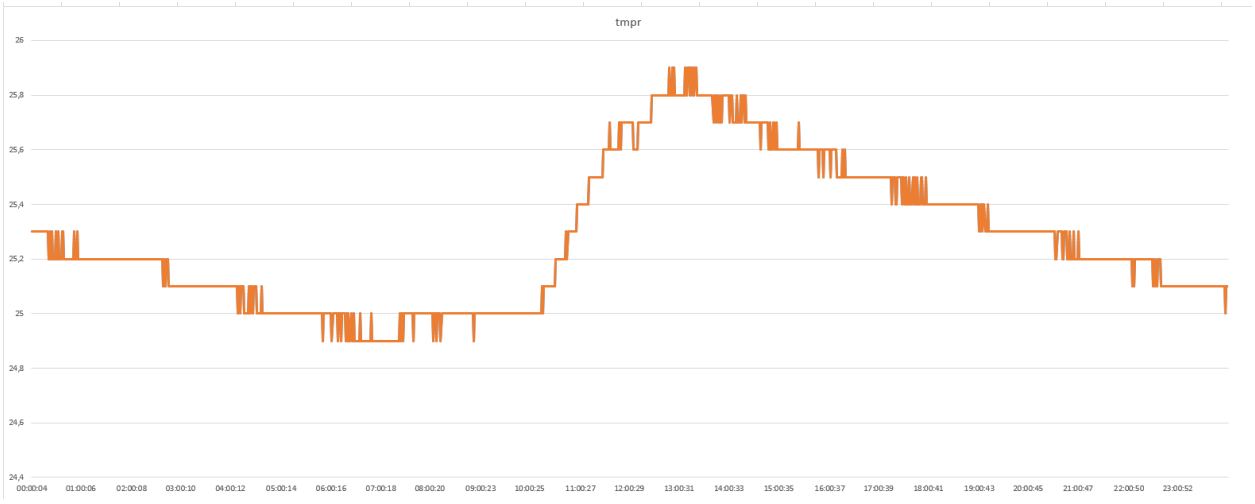
На основі отриманих даних LOF і робиться висновок являється досліджувана точка аномальною чи ні. Оскільки якщо наше значення сильно менше від середнього це означає, що досліджувана точка знаходиться далеко від сусідів. Тому її вважають аномальною.

В залежності від поставленої задачі перед алгоритмом, значення LOF більше якого точки вважаються аномальними змінюється. За замовченням значення більше 1.1 вважається як аномальне і точка визначається як помилкова.

РОЗДІЛ 3. ЗАСТОСУВАННЯ АЛГОРИТМІВ

3.1 Досліджуваний алгоритм на основі Sliding Window

Для проведення аналізу наявних алгоритмів пошуку аномальності даних було обрано дані виміру кімнатної температури протягом 24 годин з кроком 1 хвилина. Тобто наші виміри проходили в реальному часі з датчика виміру температури в приміщені.



“Рисунок 3.1.1 - Початкові дані”

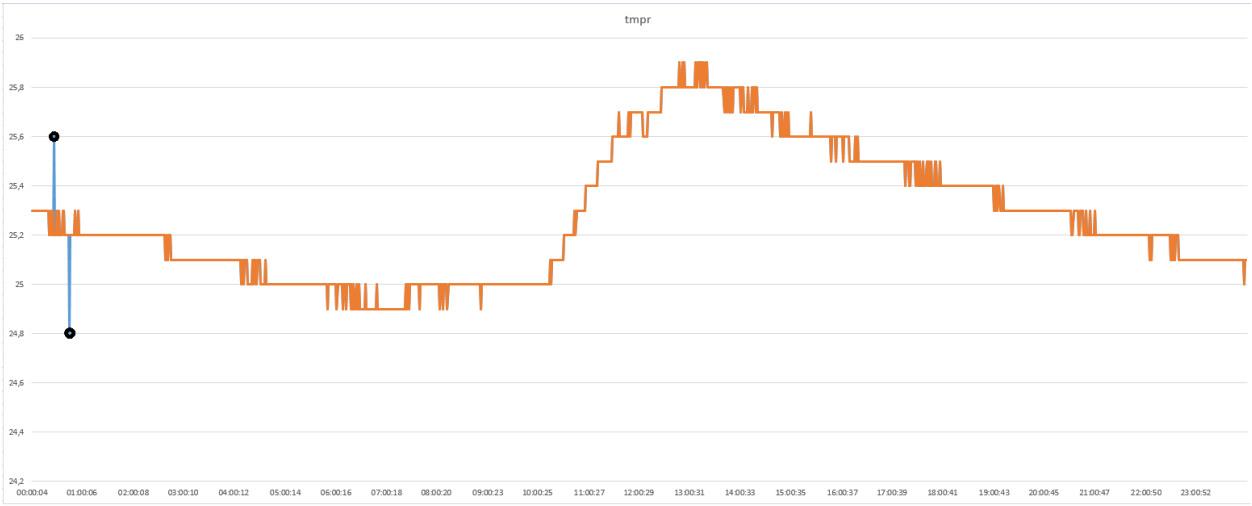
На рисунку 3.1.1 зображено початкові дані в графічному вигляді для кращого сприйняття.

Короткочасні стрибкоподібні викиди які можливо спостерігати на графіку є шумом АЦП, що в нашому випадку дорівнює 0.1 градуса за Цельсієм. Точність самого пристрою дорівнює 0.5 градусів за Цельсієм.

Початок проміжку	Кінець проміжку	Макс	Мінімум	Різниця	Еталон	Перевірка
0	9	25,3	25,2	0,1	0,2	Все вірно
1	10	25,3	25,2	0,1	0,2	Все вірно
2	11	25,3	25,2	0,1	0,2	Все вірно
3	12	25,3	25,2	0,1	0,2	Все вірно
4	13	25,3	25,2	0,1	0,2	Все вірно
5	14	25,3	25,2	0,1	0,2	Все вірно
6	15	25,3	25,2	0,1	0,2	Все вірно
7	16	25,3	25,2	0,1	0,2	Все вірно
8	17	25,3	25,2	0,1	0,2	Все вірно
9	18	25,3	25,2	0,1	0,2	Все вірно
10	19	25,3	25,2	0,1	0,2	Все вірно
11	20	25,3	25,2	0,1	0,2	Все вірно
12	21	25,3	25,3	0	0,2	Все вірно
13	22	25,3	25,3	0	0,2	Все вірно
14	23	25,3	25,2	0,1	0,2	Все вірно
15	24	25,3	25,2	0,1	0,2	Все вірно
16	25	25,3	25,2	0,1	0,2	Все вірно
17	26	25,3	25,2	0,1	0,2	Все вірно
18	27	25,3	25,2	0,1	0,2	Все вірно
19	28	25,3	25,2	0,1	0,2	Все вірно
20	29	25,3	25,2	0,1	0,2	Все вірно
21	30	25,3	25,2	0,1	0,2	Все вірно
22	31	25,3	25,2	0,1	0,2	Все вірно
23	32	25,3	25,2	0,1	0,2	Все вірно
24	33	25,3	25,2	0,1	0,2	Все вірно
25	34	25,3	25,2	0,1	0,2	Все вірно
26	35	25,3	25,2	0,1	0,2	Все вірно
27	36	25,3	25,2	0,1	0,2	Все вірно
28	37	25,3	25,2	0,1	0,2	Все вірно
29	38	25,3	25,2	0,1	0,2	Все вірно
30	39	25,3	25,2	0,1	0,2	Все вірно
31	40	25,3	25,2	0,1	0,2	Все вірно
32	41	25,3	25,2	0,1	0,2	Все вірно
33	42	25,3	25,2	0,1	0,2	Все вірно
34	43	25,3	25,2	0,1	0,2	Все вірно
35	44	25,3	25,2	0,1	0,2	Все вірно
36	45	25,3	25,2	0,1	0,2	Все вірно
37	46	25,3	25,2	0,1	0,2	Все вірно

“Рисунок 3.1.2 – Результат роботи алгоритму з проміжками по 10 хв ”

На рисунку 3.1.2 зображено аналіз початкових даних. Аномальних даних виявлено не було тому для перевірки алгоритму було внесено похибки в дані. Та перевірено як алгоритм визначить їх присутність.



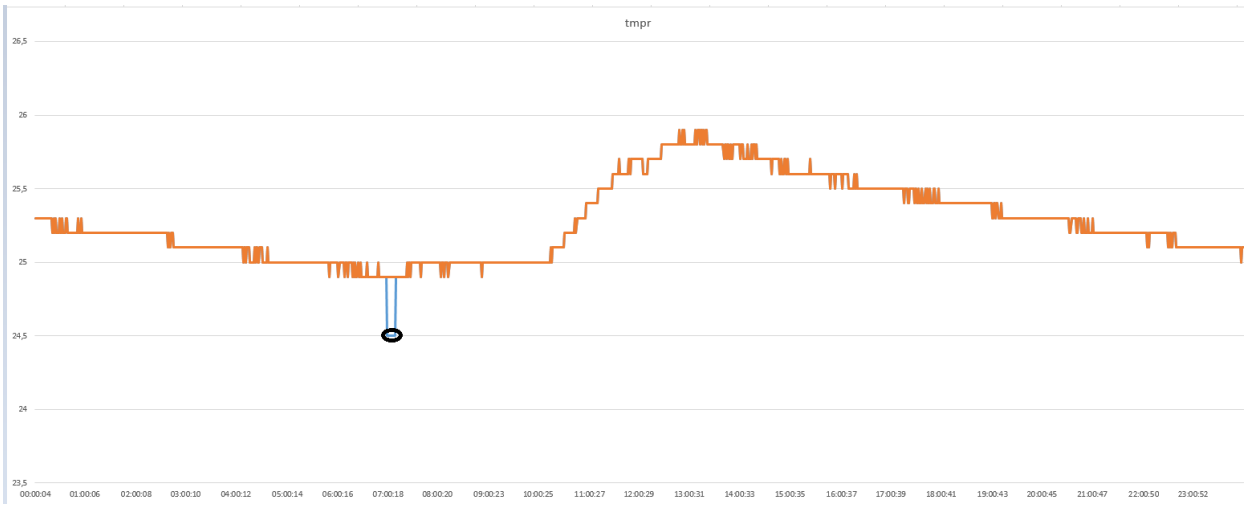
“Рисунок 3.1.3 - Внесено похибку в дані”

На рисунку 3.1.3 показано як було внесено декілька не правдивих даних.

13	22	25,3	25,2	0,1	0,2	Все вірно
14	23	25,3	25,2	0,1	0,2	Все вірно
15	24	25,3	25,2	0,1	0,2	Все вірно
16	25	25,3	25,2	0,1	0,2	Все вірно
17	26	25,3	25,2	0,1	0,2	Все вірно
18	27	25,6	25,2	0,4	0,2	Знайдена похибка
19	28	25,6	25,2	0,4	0,2	Знайдена похибка
20	29	25,6	25,2	0,4	0,2	Знайдена похибка
21	30	25,6	25,2	0,4	0,2	Знайдена похибка
22	31	25,6	25,2	0,4	0,2	Знайдена похибка
23	32	25,6	25,2	0,4	0,2	Знайдена похибка
24	33	25,6	25,2	0,4	0,2	Знайдена похибка
25	34	25,6	25,2	0,4	0,2	Знайдена похибка
26	35	25,6	25,2	0,4	0,2	Знайдена похибка
27	36	25,6	25,2	0,4	0,2	Знайдена похибка
28	37	25,3	25,2	0,1	0,2	Все вірно
29	38	25,3	25,2	0,1	0,2	Все вірно
30	39	25,3	25,2	0,1	0,2	Все вірно
31	40	25,3	25,2	0,1	0,2	Все вірно
32	41	25,3	25,2	0,1	0,2	Все вірно
33	42	25,3	25,2	0,1	0,2	Все вірно
34	43	25,3	25,2	0,1	0,2	Все вірно
35	44	25,3	25,2	0,1	0,2	Все вірно
36	45	25,3	24,8	0,5	0,2	Знайдена похибка
37	46	25,3	24,8	0,5	0,2	Знайдена похибка
38	47	25,2	24,8	0,4	0,2	Знайдена похибка
39	48	25,2	24,8	0,4	0,2	Знайдена похибка
40	49	25,2	24,8	0,4	0,2	Знайдена похибка
41	50	25,2	24,8	0,4	0,2	Знайдена похибка
42	51	25,3	24,8	0,5	0,2	Знайдена похибка
43	52	25,3	24,8	0,5	0,2	Знайдена похибка
44	53	25,3	24,8	0,5	0,2	Знайдена похибка
45	54	25,3	24,8	0,5	0,2	Знайдена похибка
46	55	25,3	25,2	0,1	0,2	Все вірно
47	56	25,3	25,2	0,1	0,2	Все вірно
48	57	25,3	25,2	0,1	0,2	Все вірно
49	58	25,3	25,2	0,1	0,2	Все вірно

“Рисунок 3.1.4 - Результат перевірки”

На рисунку 3.1.4 подано результат перевірки та далі аналітично ми можемо визначити що похибка перша відбулася на 27-й хвилині оскільки алгоритм визначив що з 28-ї хвилини похибка відсутня. Таким самим чином визначено наступну похибку на 45-й хвилині.

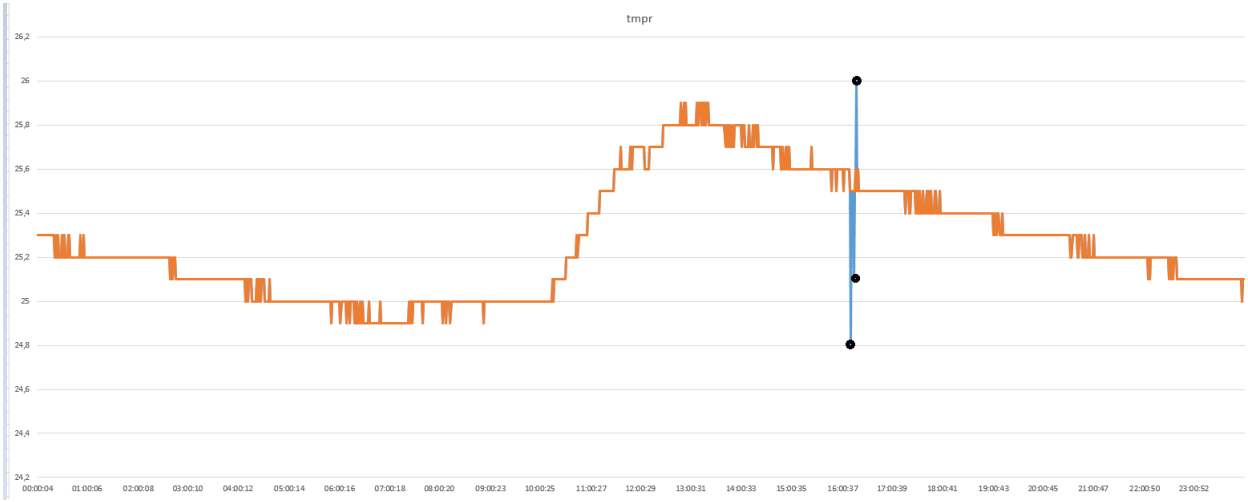


“Рисунок 3.1.5 - Графічне зображення даних з похибкою ”

F	G	H	I	J	K	L
404	413	25	24,9	0,1	0,2	Все вірно
405	414	25	24,9	0,1	0,2	Все вірно
406	415	25	24,9	0,1	0,2	Все вірно
407	416	25	24,9	0,1	0,2	Все вірно
408	417	25	24,9	0,1	0,2	Все вірно
409	418	24,9	24,9	0	0,2	Все вірно
410	419	24,9	24,9	0	0,2	Все вірно
411	420	24,9	24,9	0	0,2	Все вірно
412	421	24,9	24,5	0,4	0,2	Знайдена похибка
413	422	24,9	24,5	0,4	0,2	Знайдена похибка
414	423	24,9	24,5	0,4	0,2	Знайдена похибка
415	424	24,9	24,5	0,4	0,2	Знайдена похибка
416	425	24,9	24,5	0,4	0,2	Знайдена похибка
417	426	24,9	24,5	0,4	0,2	Знайдена похибка
418	427	24,9	24,5	0,4	0,2	Знайдена похибка
419	428	24,9	24,5	0,4	0,2	Знайдена похибка
420	429	24,9	24,5	0,4	0,2	Знайдена похибка
421	430	24,9	24,5	0,4	0,2	Знайдена похибка
422	431	24,9	24,5	0,4	0,2	Знайдена похибка
423	432	24,9	24,5	0,4	0,2	Знайдена похибка
424	433	24,9	24,5	0,4	0,2	Знайдена похибка
425	434	24,9	24,5	0,4	0,2	Знайдена похибка
426	435	24,9	24,5	0,4	0,2	Знайдена похибка
427	436	24,9	24,5	0,4	0,2	Знайдена похибка
428	437	24,9	24,5	0,4	0,2	Знайдена похибка
429	438	24,9	24,5	0,4	0,2	Знайдена похибка
430	439	24,9	24,9	0	0,2	Все вірно
431	440	24,9	24,9	0	0,2	Все вірно
432	441	24,9	24,9	0	0,2	Все вірно
433	442	24,9	24,9	0	0,2	Все вірно
434	443	25	24,9	0,1	0,2	Все вірно
435	444	25	24,9	0,1	0,2	Все вірно
436	445	25	24,9	0,1	0,2	Все вірно
437	446	25	24,9	0,1	0,2	Все вірно
438	447	25	24,9	0,1	0,2	Все вірно
439	448	25	24,9	0,1	0,2	Все вірно
440	449	25	24,9	0,1	0,2	Все вірно
441	450	25	24,9	0,1	0,2	Все вірно

“Рисунок 3.1.6 - Внесено похибку в дані”

На рисунку 3.1.5 зображено наші початкові дані та вказано на дані які є аномальними. З рисунка під номером 3.1.6 можливо помітити, що алгоритм зайшов всі аномальні показники.



“Рисунок 3.1.7 - Графічне зображення даних з похибкою”

Останньою перевіркою для алгоритму було внесення різних похибок на один проміжок часу, що зображено на рисунку під номером 3.1.7.

947	956	25,6	25,5	0,1	0,2	Все вірно
948	957	25,6	25,5	0,1	0,2	Все вірно
949	958	25,6	25,5	0,1	0,2	Все вірно
950	959	25,6	25,5	0,1	0,2	Все вірно
951	960	25,6	25,5	0,1	0,2	Все вірно
952	961	25,6	25,5	0,1	0,2	Все вірно
953	962	25,6	25,5	0,1	0,2	Все вірно
954	963	25,6	25,5	0,1	0,2	Все вірно
955	964	25,6	25,5	0,1	0,2	Все вірно
956	965	25,6	25,5	0,1	0,2	Все вірно
957	966	25,6	25,5	0,1	0,2	Все вірно
958	967	25,6	25,5	0,1	0,2	Все вірно
959	968	25,6	25,5	0,1	0,2	Все вірно
960	969	25,6	25,5	0,1	0,2	Все вірно
961	970	25,6	25,5	0,1	0,2	Все вірно
962	971	25,6	24,8	0,8	0,2	Знайдена похибка
963	972	25,6	24,8	0,8	0,2	Знайдена похибка
964	973	25,6	24,8	0,8	0,2	Знайдена похибка
965	974	25,6	24,8	0,8	0,2	Знайдена похибка
966	975	25,6	24,8	0,8	0,2	Знайдена похибка
967	976	25,6	24,8	0,8	0,2	Знайдена похибка
968	977	25,6	24,8	0,8	0,2	Знайдена похибка
969	978	26	24,8	1,2	0,2	Знайдена похибка
970	979	26	24,8	1,2	0,2	Знайдена похибка
971	980	26	24,8	1,2	0,2	Знайдена похибка
972	981	26	25,3	0,7	0,2	Знайдена похибка
973	982	26	25,3	0,7	0,2	Знайдена похибка
974	983	26	25,3	0,7	0,2	Знайдена похибка
975	984	26	25,5	0,5	0,2	Знайдена похибка
976	985	26	25,5	0,5	0,2	Знайдена похибка
977	986	26	25,5	0,5	0,2	Знайдена похибка
978	987	26	25,5	0,5	0,2	Знайдена похибка
979	988	25,6	25,5	0,1	0,2	Все вірно
980	989	25,5	25,5	0	0,2	Все вірно
981	990	25,5	25,5	0	0,2	Все вірно
982	991	25,5	25,5	0	0,2	Все вірно
983	992	25,5	25,5	0	0,2	Все вірно
984	993	25,5	25,5	0	0,2	Все вірно
985	994	25,5	25,5	0	0,2	Все вірно

“Рисунок 3.1.8 - Знайдено похибки в даних ”

На рисунку 3.1.8 показано як алгоритмом було знайдено похибку її початок на 970-й хвилині та кінець на 988-й. Але було знайдено певний недолік що при даних розмірах проміжків точніше по 10 елементів виявлення точного часу 3-ї похибки унеможливлюється.

Цього можливо уникнути зменшивши розмір проміжку до 3 хв, що в результаті призведе до збільшення часу виконання і можливим неполадкам в роботі алгоритму при довготривалих збоях роботи датчика, але можливо досягнути максимальної точності при виявлених аномалій даних. Тому саме це і було здійснено.

954	2021-07-24 15:52:37	25,5	952	954	25,6	25,5	0,1	0,2	Все вірно
955	2021-07-24 15:53:37	25,6	953	955	25,6	25,6	0	0,2	Все вірно
956	2021-07-24 15:54:37	25,6	954	956	25,6	25,6	0	0,2	Все вірно
957	2021-07-24 15:55:37	25,6	955	957	25,6	25,6	0	0,2	Все вірно
958	2021-07-24 15:56:37	25,6	956	958	25,6	25,6	0	0,2	Все вірно
959	2021-07-24 15:57:37	25,6	957	959	25,6	25,6	0	0,2	Все вірно
960	2021-07-24 15:58:37	25,6	958	960	25,6	25,6	0	0,2	Все вірно
961	2021-07-24 15:59:37	25,6	959	961	25,6	25,5	0,1	0,2	Все вірно
962	16:00:37	25,6	960	962	25,6	25,5	0,1	0,2	Все вірно
963	2021-07-24 16:01:37	25,5	961	963	25,6	25,5	0,1	0,2	Все вірно
964	2021-07-24 16:02:37	25,6	962	964	25,6	25,6	0	0,2	Все вірно
965	2021-07-24 16:03:37	25,6	963	965	25,6	25,6	0	0,2	Все вірно
966	2021-07-24 16:04:37	25,6	964	966	25,6	25,6	0	0,2	Все вірно
967	2021-07-24 16:05:37	25,6	965	967	25,6	25,6	0	0,2	Все вірно
968	2021-07-24 16:06:37	25,6	966	968	25,6	25,6	0	0,2	Все вірно
969	2021-07-24 16:07:37	25,6	967	969	25,6	25,5	0,1	0,2	Все вірно
970	2021-07-24 16:08:37	25,6	968	970	25,6	24,8	0,8	0,2	Знайдена похибка
971	2021-07-24 16:09:37	25,5	969	971	25,5	24,8	0,7	0,2	Знайдена похибка
972	2021-07-24 16:10:37	24,8	970	972	25,5	24,8	0,7	0,2	Знайдена похибка
973	2021-07-24 16:11:37	25,5	971	973	25,5	25,1	0,4	0,2	Знайдена похибка
974	2021-07-24 16:12:37	25,5	972	974	25,5	25,1	0,4	0,2	Знайдена похибка
975	2021-07-24 16:13:38	25,1	973	975	25,6	25,1	0,5	0,2	Знайдена похибка
976	2021-07-24 16:14:38	25,5	974	976	26	25,5	0,5	0,2	Знайдена похибка
977	2021-07-24 16:15:38	25,6	975	977	26	25,5	0,5	0,2	Знайдена похибка
978	2021-07-24 16:16:38	26	976	978	26	25,5	0,5	0,2	Знайдена похибка
979	2021-07-24 16:17:38	25,5	977	979	25,6	25,5	0,1	0,2	Все вірно
980	2021-07-24 16:18:38	25,6	978	980	25,6	25,5	0,1	0,2	Все вірно
981	2021-07-24 16:19:38	25,5	979	981	25,5	25,5	0	0,2	Все вірно
982	2021-07-24 16:20:38	25,5	980	982	25,5	25,5	0	0,2	Все вірно
983	2021-07-24 16:21:38	25,5	981	983	25,5	25,5	0	0,2	Все вірно
984	2021-07-24 16:22:38	25,5	982	984	25,5	25,5	0	0,2	Все вірно
985	2021-07-24 16:23:38	25,5	983	985	25,5	25,5	0	0,2	Все вірно
986	2021-07-24 16:24:38	25,5	984	986	25,5	25,5	0	0,2	Все вірно
987	2021-07-24 16:25:38	25,5	985	987	25,5	25,5	0	0,2	Все вірно
988	2021-07-24 16:26:38	25,5	986	988	25,5	25,5	0	0,2	Все вірно
989	2021-07-24 16:27:38	25,5	987	989	25,5	25,5	0	0,2	Все вірно
990	2021-07-24 16:28:38	25,5	988	990	25,5	25,5	0	0,2	Все вірно
991	2021-07-24 16:29:38	25,5	989	991	25,5	25,5	0	0,2	Все вірно
992	2021-07-24 16:30:38	25,5	990	992	25,5	25,5	0	0,2	Все вірно
993	2021-07-24 16:31:38	25,5	991	993	25,5	25,5	0	0,2	Все вірно
994	2021-07-24 16:32:38	25,5	992	994	25,5	25,5	0	0,2	Все вірно
995	2021-07-24 16:33:38	25,5	993	995	25,5	25,5	0	0,2	Все вірно
996	2021-07-24 16:34:38	25,5	994	996	25,5	25,5	0	0,2	Все вірно

“Рисунок 3.1.9 - Результат роботи алгоритму з проміжками по 3 хв ”

На рисунку 3.1.9 зображено результат пошуку аномалій з тими самими даними що зображені на рисунку 3.1.7. Вони були знайдені. Можливо помітити що зі зменшенням проміжку до мінімального, точність виявлення поодиноких викидів стає максимальною. Це допомагає виявити короткочасні зміни в даних. З максимальною точністю.

Але в даному алгоритмі є певний недолік, якщо проміжок аномальних даних буде більший за наш обраний проміжок і ці дані будуть однаковими алгоритм зможе виявити початок та кінець проміжку, але середня частина проміжку не буде ідентифікована як не вірна. Саме через це краще обирати проміжок в 10 хв.

Удосконаленням алгоритму може слугувати використання багаторівневої перевірки наших даних. А саме першочерговий відбір відбувається на мінімально можливому проміжку.

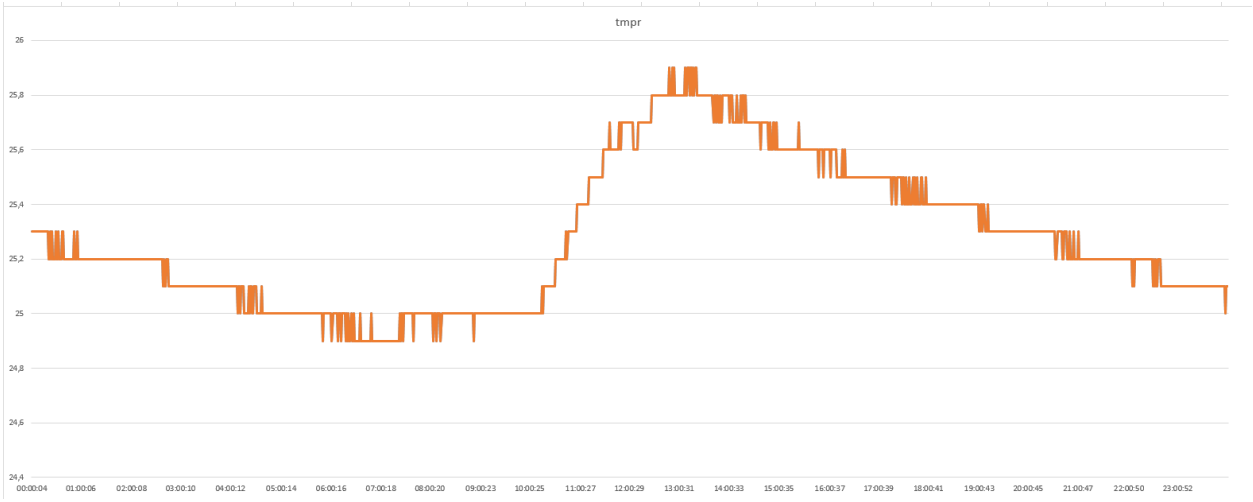
В нашому випадку це 3 хв оскільки виміри у нас за кожну хвилину. На цьому етапі будуть відсіюватись видимі відхилення зрозуміло що температура не може змінитись більше ніж на декілька градусів за хвилину в приміщені.

На яку саме кількість градусів буде можливе відхилення показує статистичний аналіз середовища дослідження. В моєму випадку він склав 0.2 градуси зважаючи на можливий шум АЦП в 0.1 градуса та мінімальну амплітуду вимірювання 0.1 градус.

Іншим рівнем перевірки може бути проміжок кожен елемент якого 10 хвилин, 30 хв чи 60 в залежності від можливих відхилень та потрібної точності. Дана перевірка вбереже алгоритм від довготривалого збою датчика і виключить можливість сприйняття таких даних за правдиві.

3.2 Досліджуваний алгоритм Локального фактора викиду (Local outlier factor)

Дані для перевірки ми будемо використовувати такі самі як і в першому алгоритмі, а саме результат виміру температури протягом 24 годин з кроком 1 хвилина в приміщені.



“Рисунок 3.2.1 - Початкові дані”

На рисунку 3.2.1 зображено початкові дані у вигляді графіка.

				x 3-1	y	range	x3-2	y	range	x3-4	y	range	x3-5	y	range	LDR	LOF		ВІРНІСТЬ
2	25,3																		
3	25,3		точка3	4	0,01	2,002498	1	0	1	1	0	1	4	0	2	0,666389	0,728719		ВІРНО
4	25,3		точка4	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,966545		ВІРНО
5	25,3		точка5	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1,041545		ВІРНО
6	25,3		точка6	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
7	25,3		точка7	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
8	25,3		точка8	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
9	25,3		точка9	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
10	25,3		точка10	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
11	25,3		точка11	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
12	25,3		точка12	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
13	25,3		точка13	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
14	25,3		точка14	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
15	25,3		точка15	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
16	25,3		точка16	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
17	25,3		точка17	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,999911		ВІРНО
18	25,3		точка18	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,999644		ВІРНО
19	25,3		точка19	4	0	2	4	0	2	1	0	1	4	0,01	2,002498	0,571225	0,999822		ВІРНО
20	25,3		точка20	4	0	2	4	0	2	1	0,01	1,004988	4	0,01	2,002498	0,570818	1,000445		ВІРНО
21	25,2		точка21	4	0,01	2,002498	4	0,01	2,002498	1	0	1	4	0,01	2,002498	0,570817	1,00009		ВІРНО
22	25,2		точка22	4	0,01	2,002498	4	0	2	1	0,01	1,004988	4	0	2	0,570818	0,999911		ВІРНО
23	25,3		точка23	4	0,01	2,002498	4	0,01	2,002498	1	0,01	1,004988	4	0	2	0,570615	1,000267		ВІРНО
24	25,2		точка24	4	0	2	4	0,01	2,002498	1	0,01	1,004988	4	0	2	0,570818	1		ВІРНО
25	25,3		точка25	4	0	2	4	0,01	2,002498	1	0,01	1,004988	4	0,01	2,002498	0,570615	1,000624		ВІРНО
26	25,2		точка26	4	0	2	4	0,01	2,002498	1	0	1	4	0	2	0,571225	0,999555		ВІРНО

“Рисунок 3.2.2 – Результат роботи алгоритму”

На рисунку 3.2.2 зображено вигляд алгоритму в програмі excel.

Кількість досліджуваних сусідів було обрано 4 тобто k = 4. Відповідно до не надто складної роботи та лінійності наших вимірів зменшення кількості сусідів забезпечує більшу швидкодію та меншу затратність на виконання алгоритму, що являється одним з головних чинників в Інтернеті речей.

У колонці range визначається відстань між точками за допомогою теореми Піфагора:

$$a^2+b^2=c^2 \quad (3.2.1)$$

Де a, b – катети;

c – гіпотенуза.

По графіку отримується відстань катетів як різницю між координатами точок та за допомогою Формули (3.2.1) знаходиться відстань між точками.

Далі за допомогою Формул (2.4.2) та (2.4.3) визначається LDR та LOF відповідно. Як критерій аномальності було обрано стандартний тобто LOF більше 1.1 вважається аномальним і дана точка визначається помилковою.

Якщо ж при визначених змінах температури за 1 хв відбулася зміна більша за 0.2 градуси то обирається наступний найближчий сусід, 0.2 градуси обрані через те, що сума мінімально можливої амплітуди зміни температури рівна 0.1 градуса та шум АЦП рівний 0.1 градуса, що в сумі й дорівнює 0.2.

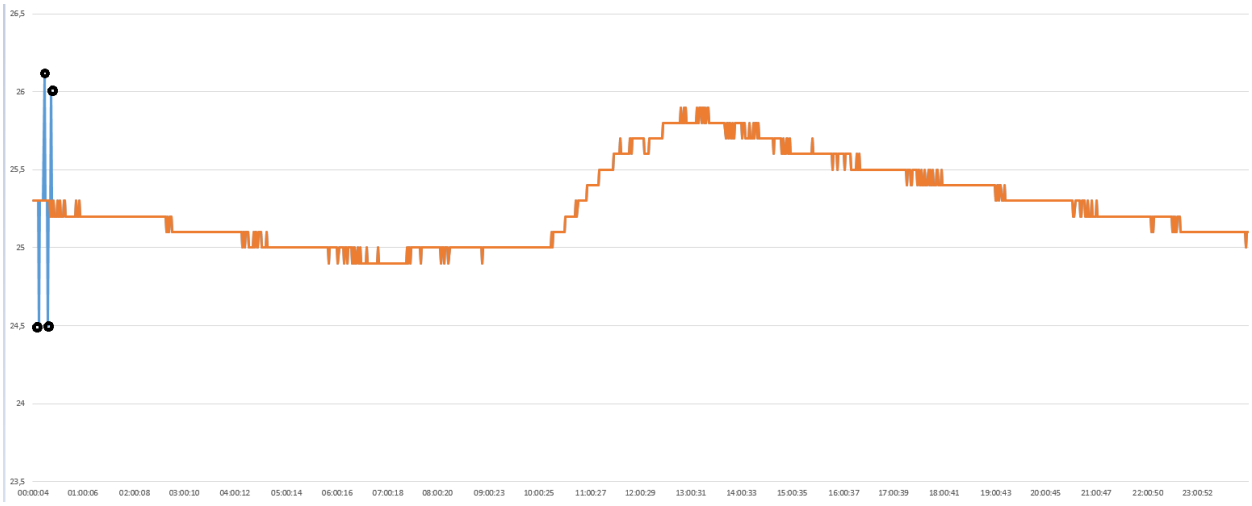
При обраному стандартному критерію фактору локальної досяжності (LOF) а саме 1.1 алгоритм визначає викиди лише коли амплітуда відхилення більше за 0.8 градуси за Цельсієм

3	25,3	точка3	4	0,01	2,002498	1	0	1	1	0	1	4	0	2	0,666389	0,728719	ВІРНО
4	25,3	точка4	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,966545	ВІРНО
5	25,3	точка5	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1,041545	ВІРНО
6	25,3	точка6	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
7	25,3	точка7	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
8	24,5	точка8	4	0,64	2,154066	4	0,64	2,154066	1	0,64	1,280625	4	0,64	2,154066	0,516607	1,106118	ПОХИБКА
9	25,3	точка9	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
10	25,3	точка10	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
11	25,3	точка11	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1	ВІРНО
12	25,3	точка12	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
13	25,3	точка13	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
14	26,1	точка14	4	0,64	2,154066	4	0,64	2,154066	1	0,64	1,280625	4	0,64	2,154066	0,516607	1,106118	ПОХИБКА
15	25,3	точка15	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016	ВІРНО
16	25,3	точка16	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,954841	ВІРНО
17	25,3	точка17	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,978736	ВІРНО
18	24,5	точка18	4	0,64	2,154066	4	0,64	2,154066	1	0,49	1,220656	4	0,49	2,118962	0,52303	1,092244	ВІРНО
19	25,3	точка19	4	0	2	4	0	2	1	0	1	4	0,01	2,002498	0,571225	0,978729	ВІРНО
20	25,3	точка20	4	0	2	4	0	2	1	0,01	1,004988	4	0	2	0,571022	0,960098	ВІРНО
21	25,2	точка21	4	0,01	2,002498	4	0,01	2,002498	1	0	1	4	0,01	2,002498	0,570817	0,981461	ВІРНО
22	26	точка22	4	0,49	2,118962	4	0,49	2,118962	1	0,49	1,220656	4	0,49	2,118962	0,527876	1,081446	ВІРНО
23	25,3	точка23	4	0,01	2,002498	4	0	2	1	0,01	1,004988	4	0	2	0,570818	0,981103	ВІРНО
24	25,2	точка24	4	0	2	4	0,01	2,002498	1	0,01	1,004988	4	0	2	0,570818	0,981282	ВІРНО
25	25,3	точка25	4	0	2	4	0,01	2,002498	1	0,01	1,004988	4	0,01	2,002498	0,570615	1,000713	ВІРНО
26	25,2	точка26	4	0	2	4	0,01	2,002498	1	0	1	4	0	2	0,571225	0,999555	ВІРНО
27	25,2	точка27	4	0,01	2,002498	4	0	2	1	0	1	4	0	2	0,571225	0,999644	ВІРНО
28	25,2	точка28	4	0	2	4	0	2	1	0	1	4	0,01	2,002498	0,571225	0,999555	ВІРНО
29	25,2	точка29	4	0	2	4	0	2	1	0,01	1,004988	4	0	2	0,571022	0,99991	ВІРНО

“Рисунок 3.2.3 - Приклад знаходження помилки при умові LOF>1.1”

На рисунку 3.2.3 в точках 8, 14,18, 22 внесено помилки.

Алгоритм визначив аномальними похибки лише ті в яких відмінність від сусідів була більшою за 0.8 градуси за Цельсієм.



“Рисунок 3.2.4 - Відображення внесених помилок”

Як можливо помітити на рисунку відображено внесені помилки і їх 4, але алгоритм визначив лише 2 з них в яких різниця від сусідів більша за 0.8 градуси.

Однак, якщо змінити порогове значення LOF відносно якого вважається точка хибною на менше, а саме 1.05 тоді всі 4 помилки будуть знайдені, і лише за умови віддаленості від сусідів на 0.6 градуси

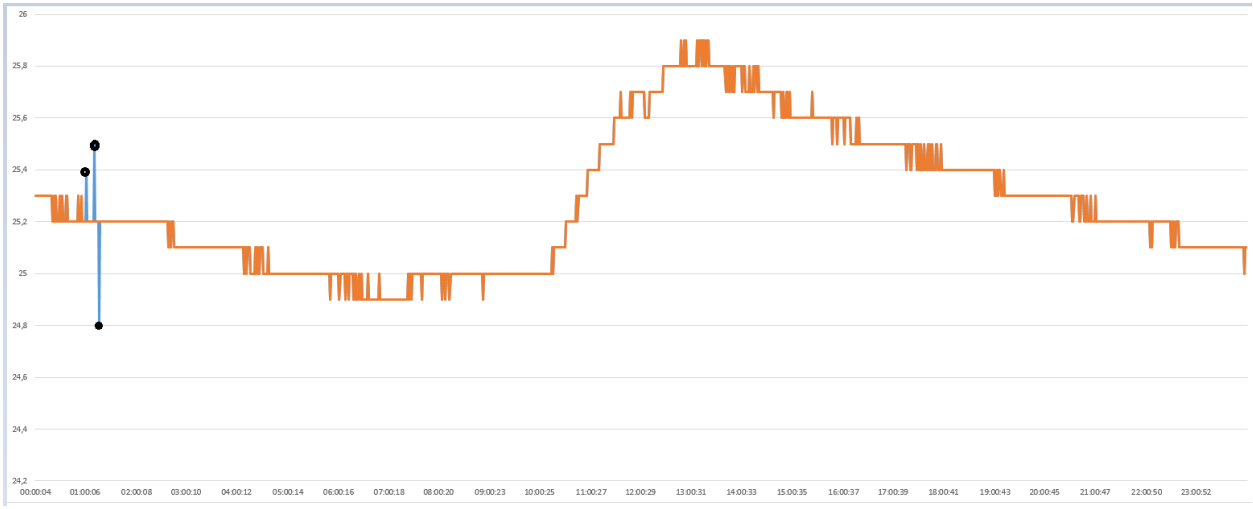
3	25,3		точка3	4	0,01	2,002498	1	0	1	1	0	1	4	0	2	0,666389	0,728719		ВІРНО
4	25,3		точка4	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,966545		ВІРНО
5	25,3		точка5	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1,041545		ВІРНО
6	25,3		точка6	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
7	25,3		точка7	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
8	24,5		точка8	4	0,64	2,154066	4	0,64	2,154066	1	0,64	1,280625	4	0,64	2,154066	0,516607	1,106118		ПОХИБКА
9	25,3		точка9	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
10	25,3		точка10	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
11	25,3		точка11	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
12	25,3		точка12	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
13	25,3		точка13	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
14	26,1		точка14	4	0,64	2,154066	4	0,64	2,154066	1	0,64	1,280625	4	0,64	2,154066	0,516607	1,106118		ПОХИБКА
15	25,3		точка15	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,976016		ВІРНО
16	25,3		точка16	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,954841		ВІРНО
17	25,3		точка17	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,978736		ВІРНО
18	24,5		точка18	4	0,64	2,154066	4	0,64	2,154066	1	0,49	1,220656	4	0,49	2,118962	0,52303	1,092244		ПОХИБКА
19	25,3		точка19	4	0	2	4	0	2	1	0	1	4	0,01	2,002498	0,571225	0,978729		ВІРНО
20	25,3		точка20	4	0	2	4	0	2	1	0,01	1,004988	4	0	2	0,571022	0,960098		ВІРНО
21	25,2		точка21	4	0,01	2,002498	4	0,01	2,002498	1	0	1	4	0,01	2,002498	0,570817	0,981461		ВІРНО
22	26		точка22	4	0,49	2,118962	4	0,49	2,118962	1	0,49	1,220656	4	0,49	2,118962	0,527876	1,081446		ПОХИБКА
23	25,3		точка23	4	0,01	2,002498	4	0	2	1	0,01	1,004988	4	0	2	0,570818	0,981103		ВІРНО
24	25,2		точка24	4	0	2	4	0,01	2,002498	1	0,01	1,004988	4	0	2	0,570818	0,981282		ВІРНО
25	25,3		точка25	4	0	2	4	0,01	2,002498	1	0,01	1,004988	4	0,01	2,002498	0,570615	1,000713		ВІРНО
26	25,2		точка26	4	0	2	4	0,01	2,002498	1	0	1	4	0	2	0,571225	0,999555		ВІРНО
27	25,2		точка27	4	0,01	2,002498	4	0	2	1	0	1	4	0	2	0,571225	0,999644		ВІРНО
28	25,2		точка28	4	0	2	4	0	2	1	0	1	4	0,01	2,002498	0,571225	0,999555		ВІРНО
29	25,2		точка29	4	0	2	4	0	2	1	0,01	1,004988	4	0	2	0,571022	0,99991		ВІРНО

“Рисунок 3.2.5 - Приклад знаходження помилки при умові LOF>1.05”

З рисунка видно, що алгоритм знайшов помилки, отже зменшення порогу аномальності збільшило якість та точність нашого алгоритму відповідно до даних вимірів.

Також потрібно розуміти така точність можлива лише тому, що дані отримані в приміщені. Якщо ж отримувати дані зовні такий малий поріг фактору локальної досяжності може спричините неполадки в роботі алгоритму, а саме визначення звичайної зміни температури як аномальності та відповідно відкидання значення надалі.

Якщо ж оцінювати точність відносно саме наших даних, то можливо застосувати умову LOF>1.01. З такою умовою точність буде максимальною, а саме кожна зміна більша чи рівна 0.3 градуси від сусідів буде помічена алгоритмом як похибка.



“Рисунок 3.2.4 - Відображення внесених помилок з меншою різницею, а саме 0.3 градуси”

57	25,2		точка57	4	0	2	4	0,01	2,002498	1	0	1	4	0	2	0,571225	0,999644		ВІРНО
58	25,2		точка58	4	0,01	2,002498	4	0	2	1	0	1	4	0	2	0,571225	0,999467		ВІРНО
59	25,2		точка59	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,998761		ВІРНО
60	25,2		точка60	4	0	2	4	0	2	1	0	1	4	0,04	2,009975	0,570615	0,998863		ВІРНО
61	25,2		точка61	4	0	2	4	0	2	1	0,04	1,019804	4	0	2	0,569816	1,000347		ВІРНО
62	25,4		точка62	4	0,04	2,009975	4	0,04	2,009975	1	0,04	1,019804	4	0,04	2,009975	0,567398	1,005319		ВІРНО
63	25,2		точка63	4	0	2	4	0,04	2,009975	1	0	1	4	0	2	0,570615	0,998596		ВІРНО
64	25,2		точка64	4	0,04	2,009975	4	0	2	1	0	1	4	0	2	0,570615	0,999303		ВІРНО
65	25,2		точка65	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,999289		ВІРНО
66	25,2		точка66	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,999644		ВІРНО
67	25,2		точка67	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
68	25,2		точка68	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
69	25,2		точка69	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
70	25,2		точка70	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,996092		ВІРНО
71	25,2		точка71	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,996092		ВІРНО
72	25,5		точка72	4	0,09	2,022375	4	0,09	2,022375	1	0,09	1,044031	4	0,09	2,022375	0,562497	1,015879		ПОХИБКА
73	25,2		точка73	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,996092		ВІРНО
74	25,2		точка74	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,996092		ВІРНО
75	25,2		точка75	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
76	25,2		точка76	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,993196		ВІРНО
77	25,2		точка77	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,993196		ВІРНО
78	24,8		точка78	4	0,16	2,039608	4	0,16	2,039608	1	0,16	1,077033	4	0,16	2,039608	0,555875	1,027979		ПОХИБКА
79	25,2		точка79	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,993196		ВІРНО
80	25,2		точка80	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	0,993196		ВІРНО
81	25,2		точка81	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
82	25,2		точка82	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
83	25,2		точка83	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
84	25,2		точка84	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
85	25,2		точка85	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО
86	25,2		точка86	4	0	2	4	0	2	1	0	1	4	0	2	0,571429	1		ВІРНО

“Рисунок 3.2.5 - Приклад знаходження помилки при умові LOF>1.01”

На рисунку 3.2.5 видно, що похибки внесені о 62, 72, 78 хвилинах. Алгоритм визначає дані аномальними якщо різниця температур з сусідами більше або рівна 0.3 градуси, що відображено на 72 та 78 хвилині, а зміна в 0.2 градуси не відображається і може вважатися як вірні дані, хоча й зміна на таку малу величину не вплине на подальшу роботу інших алгоритмів, що будуть опрацьовувати дані.

3.3 Порівняння алгоритмів

Дослідивши та відтворивши роботу алгоритмів на основі Sliding Window та Локального фактора відхилення можливо дійти певних висновків.

Найголовнішою умовою роботи любого алгоритму в Інтернеті речей є попередній аналіз досліджуваного середовища, відносно якого змінюються точність виявлення аномалій та значення після якого дані вважаються як аномальні.

Перевагою досліджуваних алгоритмів є простота використання, а саме можливість розбиття алгоритму на прості математичні обрахунки в результаті яких робляться висновки про правдивість даних.

Також важливим чинником в роботі з IoT мережею є швидкодія. В цьому аспекті алгоритм на основі рухомого вікна показав себе краще оскільки йому не потрібно спочатку отримувати масив даних, а лише потім його обробляти. Він може використовувати попередньо отриману інформацію від датчика, щоб прогнозувати правдивість отриманих даних в реальному часі та відразу відкидати аномальні показники. А для фактора локального викиду потрібно отримати всі результати вимірювань.

Потрібно зазначити, що отримані нами дані були лінійні що краще підходить для використання алгоритму на основі статистики, якщо ж дані мали б нерівномірний характер тоді працездатність алгоритму на основі рухомого вікна унеможливлювалась, а Локальний фактор викиду визначив би аномальності в даних оскільки він краще працює з нерівномірними даними.

Головним недоліком обох алгоритмів є неможливість визначення аномальностей довготривалих, а саме:

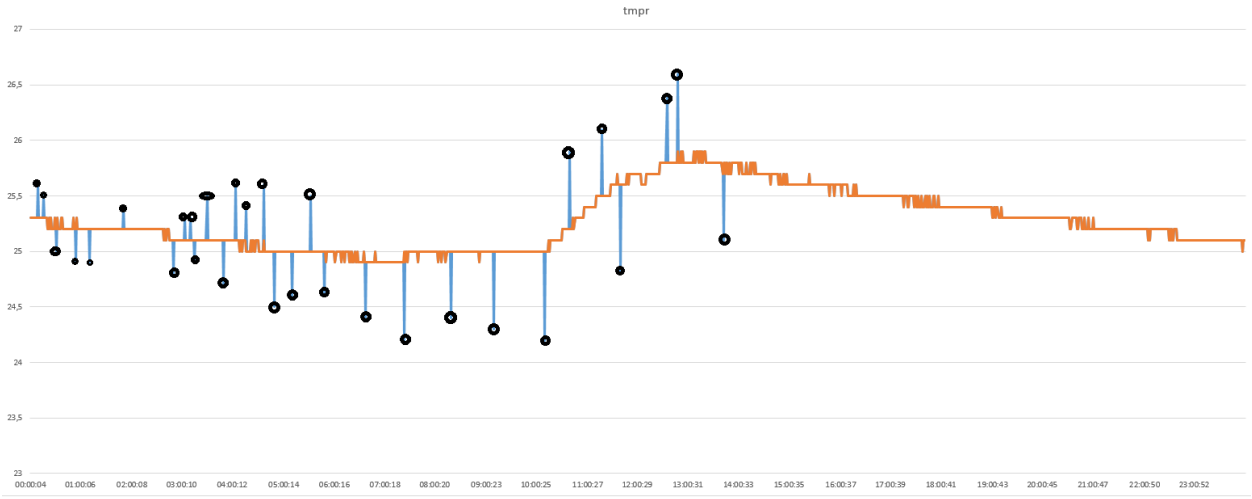
- Для алгоритму на основі рухомого вікна такий інтервал складає довжину самого досліджуваного проміжку. Тобто якщо відбудеться заміна

даних на лінійні, тривалість яких буде визначена на проміжку більшою за досліджувану аномальність буде виявлена некоректно.

- Те ж саме можна сказати й про алгоритм на основі локального фактора викиду, якщо відбудеться збій сенсора і отримані дані будуть отримані з досить великою тривалістю, а саме більшою за кількість досліджуваних сусідів (k) тоді алгоритм вважатиме цю область новою зоною і відповідно щільність її елементів буде відносно рівномірною. Тоді алгоритм не визначить їх як неправдиві.

Вважаючи найголовнішою проблемою саме поодинокі викиди обидва алгоритми виконали свою роботу та визначили помилково отримані дані з великою точністю, а саме 0.2 градуси за Цельсієм якщо це алгоритм рухомого вікна та 0.3 градуси для фактора локального викиду.

Для дослідження було внесено вручну похибки в 3 діапазонах: 0.2-0.3, 0.4-0.5, 0.6-0.8.



“Рисунок 3.3.1 – Внесено помилки в різних діапазонах”

Відповідно до алгоритмів на проміжку зміна на якому була по модулю рівною:

0.2-0.3 градуси за Цельсієм

Алгоритм на основі Sliding window знайшов 60 відсотків

Алгоритм локального фактора викиду зайшов 40 відсотків

0.4-0.5 градуси за Цельсієм

Алгоритм на основі Sliding window знайшов 100 відсотків

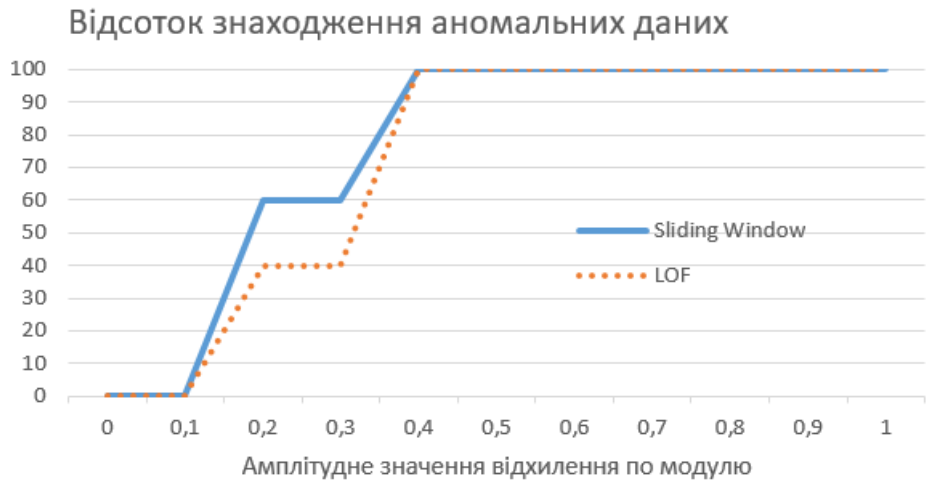
Алгоритм локального фактора викиду знайшов 100 відсотків

0.6-0.8 градуси за Цельсієм

Алгоритм на основі Sliding window знайшов 100 відсотків

Алгоритм локального фактора викиду знайшов 100 відсотків

Оцінюючи результат можна дійти висновку, що алгоритм на основі Sliding Window краще підходить для оцінювання аномальностей даних вимірювання температури повітря.



“Рисунок 3.3.2 – Відсоток знаходження аномальностей даних”

З рисунка видно, що алгоритм на основі рухомого вікна краще знаходить аномальність при малих амплітудах.

ВИСНОВКИ

В ході виконання дипломної роботи, було досліджено питання сутності IoT мережі. Недостатній рівень захисту пристроїв є причиною сповільнення розвитку Інтернету речей.

Аналіз датчиків показав, що доцільно використовувати алгоритми для визначення правдивості даних. Зовнішні чинники можуть спотворити загальну картину або взагалі змінити виміри, також можливі несправності в роботі самих датчиків. Саме через це доцільно використовувати алгоритми пошуку аномальних даних.

В результаті дослідження алгоритму Sliding Window під час моделювання та обробки реальних даних було виявлено аномальність наших даних. До переваг алгоритму слід віднести його простоту реалізації та можливість працювати в реальному часі, а отже його можливо застосувати на пристроях з обмеженими обчислювальними потужностями. Для його коректної роботи проміжки з якими працює алгоритм слід обирати в залежності від даних з якими працюєте.

В результаті дослідження алгоритму Локального фактора викиду під час моделювання даних аномальність отриманих даних була виявлена. До переваг алгоритму слід віднести можливість працювання з різними видами даних, а саме: рівномірними та нерівномірними. Але при використанні його в межах Інтернету речей слід зазначити необхідність отримання об’єму даних та неможливість роботи в реальному часі.

Результатом дослідження даних алгоритмів можливо зробити висновок та зазначити, що доцільність використання того чи іншого алгоритму залежить від характеру досліджуваного об’єкта. Якщо дані будуть лінійними, такими як виміри температури тоді доцільніше використовувати алгоритм на основі рухомого вікна, його ймовірність знаходження аномальностей на 20 відсотків вище при малих амплітудах зміни даних та мінімальні аномальні

відхилення, що визначаються алгоритмом менші на 0.1 градус, якщо ж дані будуть нерівномірними, наприклад рівень забруднення повітря тоді потрібно використовувати алгоритм локального фактора викиду. Попередньо провівши дослідження та відкоригувавши критерій оцінювання даних як аномальні.

ПЕРЕЛІК ДЖЕРЕЛ

1. https://www.bizmaster.xyz/2020/12/internet-rechei-merezheva-arkhitektura-ta-arkhitektura-bezpeky.html (Інтернет-ресурс)

2. Adrian McEwen, Hakim Cassimally. Designing the Internet of Things

3. Recommendation ITU-T Y.2060 - 2012

4. Recommendation ITU-T Y.2067- 2014

5. https://datafloq.com/read/internet-of-things-more-than-smart-things/ (Інтернет-ресурс)

6. Attaway, S. Matlab: A Practical Introduction to Programming and Problem Solving № 3 / Attaway, S. // Butterworth-Heinemann – 2013.

7. Gartner A., Forecast: The Internet of Things / Gartner A. // Worldwide – 2014.

8. Holler J., Tsiatsis V., Mulligan C., Karnouskos S., Avensand S., Boyle D. From Machine-to-Machine to the Internet of Things Introduction to a New Age of Intelligence / Holler J., Tsiatsis V., Mulligan C., Karnouskos S., Avensand S., Boyle D. // Elsevier, pp. 13-25 – 2014.

9. Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J. (2000). LOF: identifying density-based local outliers. In *ACM sigmod record* (Vol. 29, №2, pp. 93–104). ACM.