МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ГРИЩУК ЄВГЕНІЙ ВІКТОРОВИЧ

Допускається до захисту:
В.о. завідувача кафедри
прикладної математики та
кібербезпеки,
_____Луценко А. В.
«__»_____20__р.

Розробка технології аналізу поведінки для виявлення аномальних активностей на IoT систему «Smart City»

Спеціальність 105 Прикладна фізика та наноматеріали

Кваліфікаційна (магістерська) робота

Науковий керівник:
Загоруйко Л. В.,
к.т.н., доцент,
доцент кафедри прикладної
математики та кібербезпеки
_____
(підпис)

Оцінка : ___/ ___ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця 2024

АНОТАЦІЯ

Грищук Є.В Розробка технології аналізу поведінки для виявлення аномальних активностей на loT систему "Smart City. Спеціальність 105 «Прикладна фізика та наноматеріали», Освітня програма «Технології інтернету речей». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (магістерській) роботі досліджено технології аналізу поведінки для виявлення аномальних активностей в комп'ютерних системах. Проаналізовано методи виявлення аномальних активностей на технічні системи Розроблено програму яка виявляє аномальні активності в енергоспоживанні районів міста

Ключові слова: аномальні активності, технічні системи, система «Smart City», методи виявлення аномалій.

51 с., 2 табл., 14 рис., 2 дод., 20 джерел.

Hryshchuk Ev. Development of a methodology for monitoring the circulation of digital financial assets based on the analysis of cryptocurrency transactions. Specialty 105 «Applied physics and nanomaterials», Educational program «Internet of Things technologies». Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

In the qualification (master's) thesis, behavior analysis technologies for detecting abnormal activities in computer systems were investigated. The methods of detecting anomalous activities on technical systems were analyzed. A program was developed that detects anomalous activities in the energy consumption of city districts

Keywords: anomalous activities, technical systems, "Smart City" system, anomaly detection methods.

51 pp., 2 tables, 14 figures, 20 sources.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 5
ВСТУП 6
РОЗДІЛ 1. АНОМАЛІЇ ТА ЇХНІ ТИПИ У ТЕХНІЧНИХ СИСТЕМАХ 8
1.1 Класифікація аномалій 8
1.2 Визначення аномальних активностей 11
1.3 Характеристика мережевих аномалій від джерела виникнення 13
1.4 Виявлення мережевих та немережевих аномалій 16
1.5 Методи виявлення аномалій в технічних системах 19
1.5.1 Метод на основі прикладів поведінки 20
1.5.2 Метод на базі нейромережевої моделі 21
1.5.3 Статистичний метод 24
1.5.4 Сигнатурні методи 24
1.5.5 Метод на основі кластерного аналізу 25
Висновок до розділу 1 26
РОЗДІЛ 2 МЕТОД КЛАСТЕРНОГО АНАЛІЗУ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ 27
2.1 Кластеризація даних 27
2.2 Основні підходи, що використовуються в кластерному аналізі 29
2.3 Метод кластеризації K-середніх 31
2.4 Перевірка кількості кластерів 33
2.5 Компоненти технології виявлення аномалій в електромережі за допомогою кластеризації 35
Висновок до розділу 2 36
РОЗДІЛ 3. ВИЗНАЧЕННЯ АНОМАЛІЙ В ЕНЕРГОСПОЖИВАННІ МІСТА 37
3.1 Характеристика аномального енергоспоживання 37
3.2 Розробка програмного забезпечення для виявлення аномалій в енергосистемі 37
3.3 Визначення нормальних і аномальних значень 39
3.4 Кластеризація методом k-means: 41

Висновок до розділу 3 46
ВИСНОВКИ 47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 48
ДОДАТОК А 51
ДОДАТОК Б 53
ДОДАТОК В 57
ДОДАТОК Ґ 59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Advanced Persistent Threats (APT) — цілеспрямовані кібератаки

Sum of squared errors (SSE) — сумарна квадратична помилка

Internet of Things (IoT) – концепція мережевої взаємодії фізичних пристроїв

ВСТУП

Підключення до Інтернету та цифрові технології широко використовуються у всьому світі. Системи виявлення аномалій вважаються важливим інструментом виявлення широкого спектру ворожої діяльності в кіберпросторі. IT-системам загрожує постійно зростаюча кількість кібератак. З появою Advanced Persistent Threats (APT) акцент перейшов з готових шкідливих програм на багатосторонні атаки, адаптовані до конкретних організацій чи систем. Ці загрози викликані різними мотивами, наприклад шпигунством або саботажем, і часто завдають значно більшої шкоди [1].

Хоча APT використовують шкідливе ПЗ, як і більшість звичайних атак, рівень складності шкідливих програм зазвичай вищий. Це проблематично, оскільки захисні заходи, які пропонують постачальники засобів безпеки, часто використовують у першу чергу системи на основі сигнатур, які є ефективними для захисту від відомих носіїв вірусів , непередачуванних атак, або дій користувача, але безсилі з раніше невідомим шкідливим ПЗ [2]: Звичайне виявлення неправомірних дій значною мірою залежить від баз даних сигнатур, які необхідно оновлювати з появою нової техніки або зразків атак. Що стосується нових загроз, то такі бінарні шаблони задіяного шкідливого програмного забезпечення навряд чи існуватимуть під час атаки. Полі- і метаморфічні методи [3] додатково заплутують шкідливе програмне забезпечення, створюючи варіанти шкідливого ПЗ, що змінюється, і має різний статичний зовнішній вигляд для шифрувальника.

Системи "Smart City" стають ключовим компонентом сучасної урбаністики, інтегруючи інформаційно-комунікаційні технології (ІКТ) та Інтернет речей (IoT) для забезпечення ефективного управління міськими ресурсами, зменшення впливу на довкілля та покращення якості життя громадян.

Ці системи використовуються у багатьох аспектах міської інфраструктури, таких як управління транспортом, енергетикою, водопостачанням, сміттям, а також у забезпеченні безпеки.

Однак стрімке зростання використання IoT-пристроїв у розумних містах супроводжується новими викликами, зокрема зростанням кількості кіберзагроз. Основними проблемами є вразливість IoT-пристроїв до атак через слабкий захист, обмежені обчислювальні ресурси для забезпечення традиційних засобів кібербезпеки, а також різноманітність протоколів та платформ. Однією з найбільш небезпечних форм загроз є аномальна активність, яка може включати як цілеспрямовані кібератаки (наприклад, DoS-атаки), так і несподівані збої, що порушують нормальну роботу системи.

Актуальність роботи. Розробка технології аналізу поведінки для виявлення аномальних активностей у IoT-системах "Smart City" є критично важливою для забезпечення безперервності роботи міської інфраструктури, захисту конфіденційних даних та підвищення загальної стійкості міста до кіберзагроз. Ця робота сприятиме підвищенню рівня довіри до IoT-технологій, їх широкому впровадженню та покращенню якості життя громадян.

Мета роботи – Розробка технології аналізу поведінки для виявлення аномальних активностей на loT систему «Smart City»

Об'єктом дослідження – є IoT-система «Smart City»,саме енергосистема міста

Предметом дослідження – є методи виявлення аномалій та алгоритми кластеризації для аналізу поведінки в енергосистемах IoT-системи «Smart City».

Наукова новизна полягає у створенні програмного забезпечення для виявлення аномалій, яке може бути інтегроване в існуючі енергетичні системи. Програма допоможе підвищити ефективність моніторингу та швидкість реагування на аномальні події.

Для досягнення поставленої мети потрібно виконати наступні ***завдання***:

1. Аналіз відомих аномалій та їх типи у технічних системах.
2. Порівняння методів виявлення аномалій та вибір оптимальної технології
3. Розробка технології аналізу поведінки для виявлення аномальних активностей на loT систему «Smart City»

РОЗДІЛ 1. АНОМАЛІЇ ТА ЇХНІ ТИПИ У ТЕХНІЧНИХ СИСТЕМАХ

1.1 Класифікація аномалій

Аномалії зустрічаються у багатьох сферах і є об’єктом активного дослідження у широкому спектрі застосувань, включаючи мережеву безпеку, Інтернет речей та виробничі системи [1, 3, 4, 5]. Загальною характеристикою для всіх цих галузей є розуміння аномалії як відхилення від звичайних правил або порушення, яке не відповідає нормальній поведінці системи [2]. Такі відхилення зазвичай мають випадковий характер, спричиняють нестабільність системи та стають причиною неефективності або помилок у її роботі. Рисунок 1.1 ілюструє класифікацію аномалій в технічних системах

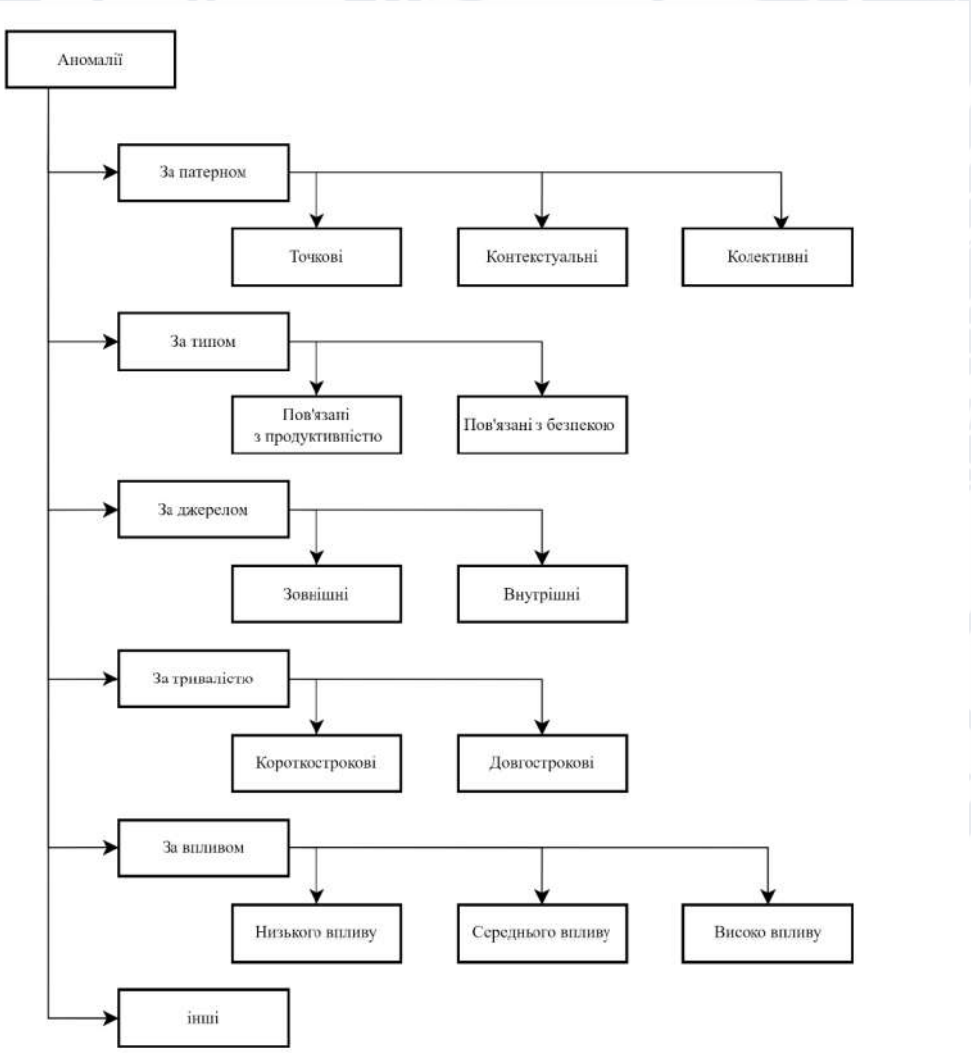

Рисунок 1.1– Класифікація аномалій

Класифікація аномалій поширюється на всі досліджувані сфери і може бути визначена за різними характеристиками, такими як об’єкт фокусування вимірність та тимчасова поведінка. Залежно від конкретної сфери застосування, фокус може бути спрямований на вплив аномалії на загальну динаміку системи або на локальні порушення, пов’язані, наприклад, з датчиками чи виконавчими механізмами. Аномалії можуть бути виявлені безпосередньо або через непрямі методи оцінки стану системи. Також вони можуть мати як лінійні, так і нелінійні характеристики.

У рамках дослідженої літератури зазначено, що динаміка аномальних систем, рідко буває інваріантною у часі. Щодо часових змін, їх поділяють на стаціонарну або нестаціонарну, а також на короткострокову чи довготривалу поведінку [5].

Для виявлення таких аномалій спочатку основну увагу приділяли стохастичним методам ідентифікації аномалій, зокрема точковим аномаліям. У цих підходах визначають ймовірнісні щільності для ключових параметрів, а параметри, що виходять за межі заданих процентилів, класифікували як викиди, які розглядалися як аномалії певного ступеня [6].

Огляд стохастичних методів виявлення аномалій [3], представляє інші типи аномалій, які зустрічаються в даних часових рядів технічних систем. Як наслідок, колективні та контекстуальні порушення визначаються як додаткові типи аномалій на додаток до статистично описаних викидів. Рис. 1.2 ілюструє три типи аномалій, які включені в одновимірний часовий ряд [4].

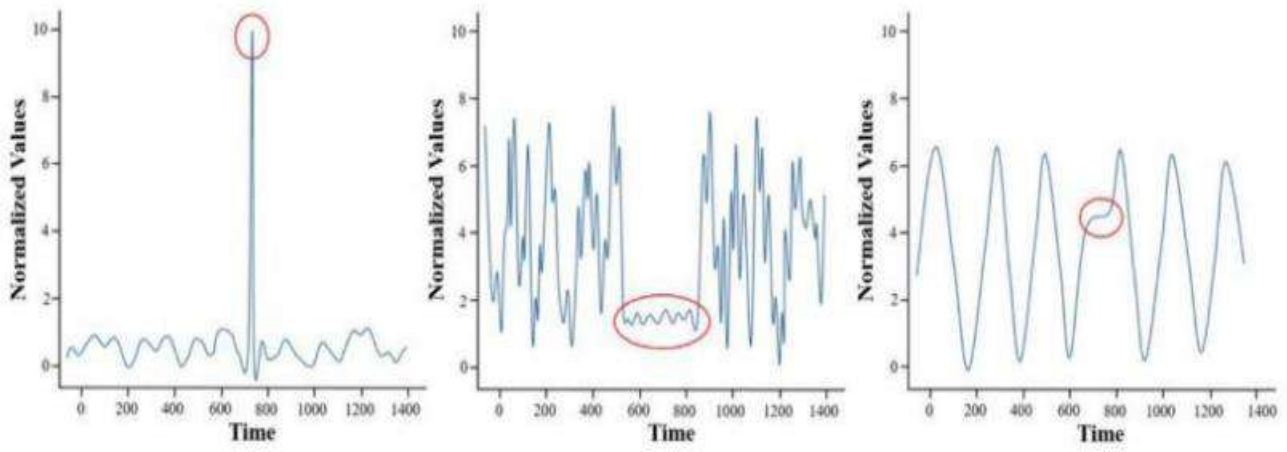

Рисунок 1.2 – Точкова аномалія (ліворуч), колективна аномалія(середина),контекстна аномалія (праворуч)[3]

Колективні аномалії характеризуються групою векторів даних, де кожен окремий вектор виглядає нормальним, але їхня сукупність демонструє відхилення. Таке відхилення визначається внутрішньою структурою послідовності. Для багатовимірних часових рядів запропоновано використовувати метрику, що оцінює рівень відхилення кожного вектора в групі, з урахуванням експоненціального зменшення впливу в часі. Якщо інтегральна метрика перевищує змінний поріг, система визначає наявність колективної аномалії [6].

Контекстуальні аномалії описують вектори даних або їхні групи, які в межах конкретного контексту вказують на нерегулярність, навіть якщо самі по собі виглядають нормальними. Контекст визначається сукупністю сусідніх векторів даних або структур, які утворюють оточення даних. Це дозволяє точніше виявляти порушення, знижувати кількість хибнопозитивних результатів і ідентифікувати першопричину аномалій за допомогою аналізу метаданих (атрибуція аномалій).

На відміну від колективних аномалій, виявлення контекстуальних порушень залежить від короткострокових і довгострокових особливостей

навколишніх структур даних. У багатовимірних часових рядах контекст формується сукупністю сусідніх векторів у межах певного часово-просторового горизонту. Часто для характеристики контекстуальних аномалій застосовуються метрики відстані. Реалізація цього підходу можлива через техніку ковзного вікна, коли для кожної нової вибірки даних розраховується відстань до попереднього контексту. Якщо метрика перевищує динамічний поріг, фіксується контекстуальна аномалія [7]. У разі відсутності аномалії база контекстів оновлюється.

Можливий такий випадок, коли окремо той, чи інший екземпляри не будуть мати аномальну поведінку, а разом – поводитись аномально. Розглянемо наступну послідовність подій на хості: smtp-mail, http-web, http-web, ftp, smtpmail, http-web, ssh, bufferoverflow, ftp, http-web. Виокремлені дії відносяться до відомої атаки віддаленою машиною з копіюванням даних з хосту на віддалений пункт призначення через ftp. Потрібно зазначити, що point anomaly та collective anomalies можуть бути перетворені на contextual anomaly шляхом включення контекстної інформації.

Залежно від сфери застосування контекст може мати різні масштаби й виміри — від часових до просторових або просторово-часових характеристик середовища. У виробничих системах він включає дані про процеси й відповідні технологічні параметри[4].

1.2 Визначення аномальних активностей

Аномальні активності в кібербезпеці визначаються як відхилення від звичайних патернів поведінки, що можуть свідчити про потенційні загрози, зокрема несанкціонований доступ, спроби злому системи чи деструктивну діяльність у мережах. Ідентифікація таких аномалій є критичним аспектом сучасного захисту інформаційних систем, оскільки своєчасне виявлення й реагування можуть значно зменшити шкоду, яку здатна завдати кібератака.

Аномалії в кібербезпеці трактуються через різні підходи:

1. **Традиційне визначення.** Аномалії є даними або активностями, що суттєво відрізняються від нормального функціонування системи. Це визначення підтримується класичними роботами [8],[9] які акцентують увагу на аномаліях як порушеннях, здатних свідчити про атаки.
2. **Класифікація.** Згідно з [10], аномалії поділяються на точкові, колективні та контекстуальні. У сфері кібербезпеки це відображається на прикладах, таких як:

Сучасні методи ідентифікації аномалій

Для аналізу аномальних активностей використовуються різноманітні інструменти та підходи. Найбільш визнані методи детально описані в літературі:

- Машинне навчання і великі дані

Застосування алгоритмів машинного навчання (ML) у поєднанні з аналітикою великих даних є основою сучасних підходів до виявлення аномалій[11]. Відомі методи включають:

Кластеризацію для виявлення схожих патернів, які виходять за межі норми.

Глибокі нейронні мережі такі як LSTM (довготривала короткочасна пам'ять), які дозволяють виявляти відхилення в часі завдяки аналізу послідовних даних.

- Поведінковий аналіз

Згідно з [12] поведінковий аналіз дозволяє ідентифікувати аномалії шляхом визначення нормальної активності користувачів, пристроїв або мереж. Прикладом є виявлення нетипового доступу до баз даних у незвичайний час.

- Системи виявлення вторгнень (IDS)

Системи IDS, як зазначено в роботі [13] використовують сигнатурний підхід або аналіз поведінки для виявлення аномальної активності, наприклад, фішингових атак або спроб експлуатації вразливостей.

- Інтелектуальна аналітика безпеки

Інтеграція штучного інтелекту (AI) та аналітики великих даних дозволяє ідентифікувати складні атаки, які залишаються непомітними для традиційних

систем. [14] підкреслюють, що автоенкодери глибокого навчання є ефективними для моделювання нормальної поведінки та виявлення відхилень.

- Системи управління подіями та інформацією про безпеку (SIEM)

SIEM забезпечують централізоване збирання й аналіз даних із різних джерел. У роботі [14] описано, як використання SIEM дозволяє об'єднати події з мереж, серверів і додатків, щоб визначати аномалії та автоматично генерувати відповіді.

- Застосування в практичних сценаріях

DDoS-атаки– [8] пропонують використовувати багатовимірний аналіз трафіку для виявлення колективних аномалій.

Фішинг – [9] наголошує на важливості аналізу контекстуальних ознак для ідентифікації підозрілих повідомлень.

***Інсайдерські загрози*.** Підходи до поведінкового аналізу, описані [11], дозволяють виявляти нетипову активність користувачів усередині організації.

1.3 Характеристика мережевих аномалій від джерела виникнення

Можна виділити п'ять основних причин ненормальної поведінки систем в мережі .

1) Network – targeted anomalies

З групи мережевих атак можна виділити ті, які мають на меті викликати аномальну поведінку мережевого трафіку в корпоративній мережі, виводячи всю мережу з ладу. Аномалії мережі також можна класифікувати, розділивши їх на дві великі групи: програмно- апаратні відхилення та проблеми безпеки.

Відхилення програмного та апаратного забезпечення можна класифікувати як: Несправність обладнання, помилка програмного забезпечення, помилка конфігурації, порушення продуктивності пристрою.Проблеми безпеки можна розділити на сканування, вірусну активність, використання вразливостей, атаки на відмову в обслуговуванні, мережеві модифікатори та аналізатори трафіку [5].

Типові приклади апаратних і програмних аномалій включають збої файлового сервера, широкомовні шторми, чати вузлів і тимчасові перевантаження .

Наприклад, збільшення кількості FTP-запитів до файлового сервера може призвести до збою файлового сервера , такого як збій веб-сервера. Помилки мережевої підкачки виникають, коли програма перевищує ліміт пам'яті робочої станції та починає використовувати пам'ять на мережевому файловому сервері.

Хоча ця аномалія може не вплинути на одного користувача , вона впливає на інших користувачів мережі, що призводить до браку пропускної здатності в мережі . Широкомовний шторм — це ситуація, за якої широкомовні пакети використовуються, поки мережа не вийде з ладу.Babbling node виникають, коли вузол надсилає невеликі пакети в нескінченному циклі , шукаючи інформацію, наприклад звіти про стан . Якщо в мережі є «гаряча точка», збій відповідного каналу в цій точці або надмірне навантаження в мережі призведе до перевантаження на короткий період часу[9].

У деяких випадках проблеми з програмним забезпеченням можуть проявлятися як аномалії мережі, наприклад помилки в реалізації протоколу. Наприклад, якщо на супер сервері (inetd) виникає помилка припущення протоколу , доступ до мережі обмежується, що впливає на навантаження мережевого трафіку. Помилки в програмному забезпеченні, яке забезпечує компоненти інформаційної системи, можуть порушити надання послуг і знизити продуктивність як окремих компонентів, так і системи в цілому.

Друга основна категорія мережевих аномалій – проблеми безпеки. Прикладами таких аномалій є атаки на відмову в обслуговуванні та злам мережі. Атака на відмову в обслуговуванні відбувається, коли послугу, що надається мережею, захоплює зловмисник. Зловмисник може вимкнути критичні служби, такі як сервер доменних імен (DNS), і спричинити збій віртуальної мережі. У цьому випадку аномалія може характеризуватися дуже низькою пропускною здатністю. У разі зламу мережі зловмисник може зменшити пропускну здатність мережі та заповнити мережу непотрібним трафіком.[6]

Мережеве сканування виконує сканування, щоб проаналізувати топологію мережі та визначити служби, які можуть бути використані для атак .У більшості випадків скануватиметься вся підмережа. Це виражається в тому, що в мережі на яку здійснена атака, є багато пакетів з однієї IP-адреси сканера на багато IP-адрес досліджуваної підмережі.

Аналізатор трафіку (сніффер) призначений для перехоплення та аналізу мережевого трафіку. У найпростішому випадку мережевий адаптер в апаратному комплексі перемикається в режим прослуховування, роблячи потік даних у підключеному сегменті доступним для подальшого дослідження.

Вірусна мережева активність є результатом спроб розповсюдження комп'ютерних вірусів і хробаків через ресурси мережі. Встановлений вірус може використовувати одну або кілька вразливостей у мережевих додаткових службах. Найпоширенішими атаками, пов'язаними з мережевими аномаліями, є DoS/DDoS, широкомовні шторми, мережевий пейджинг, помилки файлового сервера, вузли, тимчасові перевантаження, сканування/перерахування та підробка пакетів. [11].

2) Host – related anomalies (Аномалії, пов'язані з хостом)

Атаки, пов'язані з хостом, націлені на певний хост або систему шляхом запуску шкідливого програмного забезпечення для компрометації або пошкодження функціональності системи.

Більшість атак Host належать до категорії шкідливих вірусів, рекламного , шпигунського програмного забезпечення та троянських програм .

3) Software – related anomalies (Аномалії, пов'язані з програмним забезпеченням)

Приклади аномалій, пов'язаних із програмним забезпеченням, можуть включати дії, пов'язані з поведінкою атак, таких як впровадження коду та виконання міжсайтових сценаріїв (XSS).

4) Physical-related anomalies (Фізична аномалія)

Фізична атака є результатом спроби посилити мережевий пристрій (вузол або інший пристрій) або його конфігурацію. Це може включати зміну конфігурацій і введення backdoors (тобто The Evil Maid).

5) Human-related anomalies(Рукотворні аномалії)

Остання категорія мережевих атак базується на діях людини.

Фішинг — це форма атаки "людина посередині", під час якої зловмисник використовує електронну пошту або електронні повідомлення чи служби для отримання облікових даних або конфіденційної інформації. Будь-яка спроба користувача отримати вищі привілеї вважається R2L-атакою людини від користувача до root.

Людські атаки також можуть включати викрадення сесії. Ці атаки покладаються на те, що зловмисник отримує данні через активний сеанс і отримує доступ до файлів cookie та маркерів.[9]

1.4 Виявлення мережевих та немережевих аномалій

Основним підходом до виявлення аномалій є створення профілю організацій на основі певних параметрів.

Алгоритму виявлення аномалії показано на рисунку 1.3

Дані, які аналізуються, щоб визначити, чи існує аномальний екземпляр для конкретного випадку, є трафіком даних. Ви можете підключатися чи ні, залежно від вашої системи та цілей моніторингу. Для мереж трафік складається з серії пакетів, зазвичай фрагментованих на рівні IP. Дані збираються протягом певного періоду часу та далі нормалізуються для визначення основних характеристик профілю діяльності. Зазначений набір символів порівнюється з попередньо визначеними символами під час звичайної діяльності системи. Якщо є значна різниця в параметрах порівняння, трафік вважається ненормальним, інакше звичайна модель поведінки змінюється і ця діяльність переноситься в звичайний режим мережі [8].

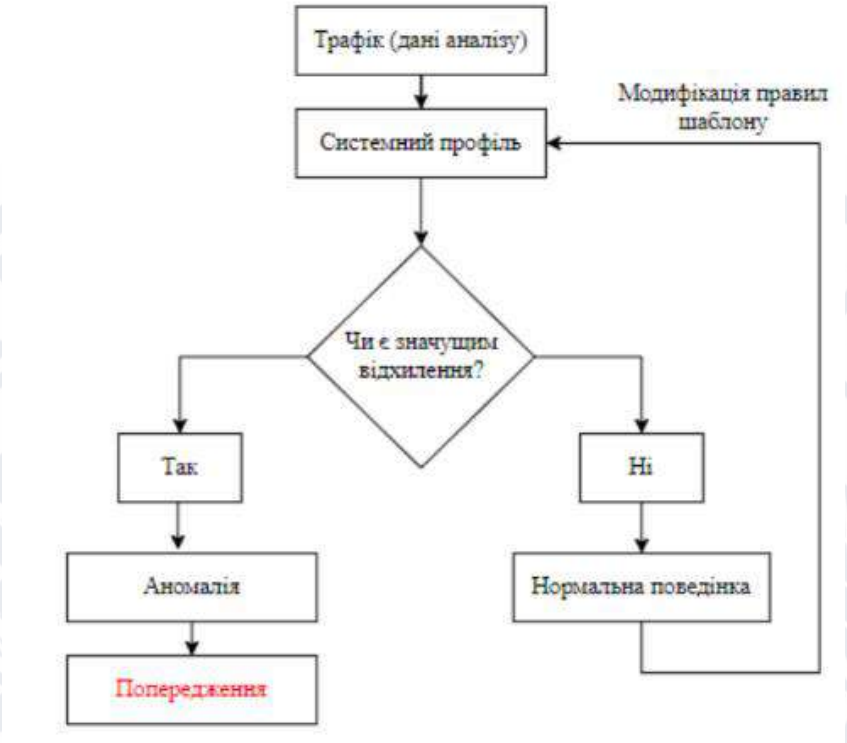

Рисунок 1.3 - Алгоритм дії системного профілю

Принципи моніторингу аномалій, не пов'язаних із мережею, такі ж, як і для аномалій мережі. Залежно від конкретних характеристик компанії, використовуваних програм, типу операційної системи тощо, здійснюється звичайна класифікація видів діяльності та створення шаблонів.

Таблиця 1.1 показує групу окремих випадків, які вимагають уваги (приклад тут стосується операційної системи Windows версій (7 - 11).

Таблиця 1.1 – Характеристика немережевих аномалій

Windows: підозрілі мережеві комунікації	Для того, щоб виявити спілкування процесів з зовнішніми IP адресами, що зазвичай відрізняється (профіль мережевої активності)

Windows: елементи запуску	Для того щоб виявити невідомі ознаки (елементи) запуску (профіль елементів запуску)
Windows: заплановані завдання	Для того щоб виявити невідомі заплановані завдання (профіль запланованих завдань)
Windows: запущені процеси	Для того, щоб виявити: - Діяльність невідомих дивних процесів; - Процес із невідомими шляхами; - Процеси із нестандартними хешами;
Windows: прослуховування портів	Для того, щоб виявляти невідомі процеси, що відкривають та прослуховують порти
Windows: etc / hosts	Для того, щоб виявляти підозрілі зміни etc / hosts файлів
Windows: cmd.exe та powershell.exe процеси	Для того, щоб виявляти аномальні процеси, що запускають cmd.exe та powershell.exe процес
Windows: завантаження драйверів	Для того, щоб моніторити всі драйвери, що завантажені із підозрілих сайтів
Windows: загрузка модулів	Для того, щоб моніторити всі модулі, що завантажені із підозрілих сайтів
Windows: сервіси	Для того, щоб виявляти нові сервіси або зміни в існуючих
Windows: батьківські та дочірні процеси	Для того, щоб виявляти батьківські та дочірні процеси

Наприклад, певний процес може ініціювати підозріле підключення до Інтернету. Таблиця процесів збирається щоразу, коли встановлюється нове підключення до Інтернету, і порівнюється з профілем звичайної діяльності. Якщо ви хочете стежити за процесом відкриття порту, схема працює наступним чином: Існує список портів, які часто відкриваються, і список портів, які відкриваються рідко. Якщо відкрито порт, який належить до списку «рідких» портів , буде спрацьовувати сповіщення (щоб ви могли визначити, чому незаплановано відкрито порт). Перевірте, який процес запустив модуль, і знайдіть хеш модуля.

Тобто на основі різної службової інформації, що дає нам ОС будуються профілі організацій з певними виключеннями для випадків.

1.5 Методи виявлення аномалій в технічних системах

Виявлення аномалій ґрунтується на застосуванні різних підходів та технік, які охоплюють широкий спектр методів. Основні з них включають такі атрибути:

— *Порогові значення* :Об'єкти аналізу оцінюються шляхом встановлення числових інтервалів. Якщо спостереження виходить за межі цих інтервалів, така поведінка вважається аномальною. Наприклад, параметрами можуть бути кількість файлів, які були відкриті користувачем за певний час, число невдалих спроб авторизації, рівень завантаження центрального процесора тощо. Порогові значення можуть бути:

- ***Статичними*** — заздалегідь визначеними для всієї системи.
- ***Динамічними*** — такими, що адаптуються до специфіки конкретної системи або змінюються у реальному часі.

— *Статистичні методи*: Рішення про можливу атаку приймається на основі аналізу великого обсягу даних, які проходять статистичну обробку. Зазвичай ці методи дозволяють ідентифікувати відхилення, використовуючи моделі нормального розподілу, середніх значень або відсоткових відхилень.

— *Параметричні підходи* : Ці методи передбачають створення профілю нормальної поведінки системи, заснованого на певних шаблонах. Профіль задає "еталонні" параметри для нормальної роботи, які порівнюються із поточними спостереженнями.

— *Непараметричні підходи:* У цьому випадку профіль нормальної поведінки системи створюється під час фази навчання на основі спостережень за об'єктом. Це дозволяє адаптуватися до змін у поведінці системи і більш точно виявляти відхилення.

— *Методи, засновані на правилах (або сигнатурах):* Ці методи схожі з непараметричними, але додатково використовують "правила" або сигнатури, що описують нормальну поведінку. У період навчання формується опис очікуваної поведінки, який надалі використовується для порівняння із новими даними.

— *Інші методи :* До цієї категорії належать сучасні підходи, такі як:

- ***Нейронні мережі,*** які дозволяють аналізувати складні залежності між вхідними параметрами.
- ***Генетичні алгоритми,*** що ефективно класифікують ознаки, зібрані сенсорами.

1.5.1 Метод на основі прикладів поведінки

Найпростіший підхід полягає в тому, щоб безпосередньо запам'ятовувати приклади дій, послідовності команд користувача або взагалі всі параметри, які можна використовувати для реєстрації навчання на основі екземплярів (instance-based learning). Хоча цей підхід неможливо застосувати до інших випадків моделювання поведінки людини, він дуже ефективний для завдань виявлення вторгнень. Це пов’язано з обмеженою кількістю дій, які може виконувати суб’єкт комп’ютерної системи, значною детермінованістю завдань, які можуть виконуватися, і надзвичайно важливими функціями структури операційної системи. Реакція комп’ютерної системи на ненормальну поведінку процесу

полягає в тому, щоб змусити його сповільнитися. Таким чином, більшість процесів, які демонструють активну ненормальну поведінку, зупиняються та автоматично припиняються як такі, що не відповідають системі. Раніше записані підпослідовності системних викликів, що входять до навчального набору, зберігаються, а їх присутність у поточному сеансі перевіряється під час роботи. Оскільки системні виклики програми контролюються через їх високу регулярність (порядок і тип викликів в основному визначаються вихідним кодом програми), розмір бази підпослідовностей невеликий [7].

Часткові послідовності поточного сеансу, які не були частиною тренувального сеансу, вважаються ненормальними. Цей підхід вимагає додавання ядра операційної системи, що непросто і не завжди можливо. Крім того, постійна присутність таких компонентів моніторингу знижує загальну швидкість роботи всієї системи на 4-50% [9]. Іншим прикладом системи на основі екземплярів є [8]. Вводиться спеціальна метрика для подібності рядків.

Оскільки для кожного користувача необхідно зберігати великий обсяг даних, для зменшення ємності бази даних послідовностей необхідно застосовувати спеціальні методи. Автори розглядають два таких методи, а саме: вибіркової селекції прикладів, коли зберігаються не всі приклади послідовностей, а тільки останні n , або всі, крім n з найменшими ймовірностями, та перетворення бази на базу представників класів елементів. Через високі вимоги instance-based системи можна використовувати лише для завдань виявлення аномалій, коли кількість можливих сигналів невелика, а спостережувані об’єкти мають здебільшого статичну поведінку.

1.5.2 Метод на базі нейромережевої моделі

Використання нейронних мереж базується на найбільш неформальному формулюванні проблеми: виявленні аномальної поведінки. Ідея полягає в тому, щоб отримати «навчальний» набір із вхідних і вихідних параметрів, які

характеризують поведінку системи, щоб мережа могла «звикнути» до системи. Результатом може бути "фактор нормальності" поведінки або один із системних параметрів. Якщо у вхідних даних є закономірності, передбачається, що мережа може «навчатися» на цих закономірностях. У процесі роботи, якщо наданий нейронною мережею вихід відповідає коефіцієнту, та він потрапляє в небезпечний діапазон або відхиляється від значення реальної системи, та не є одним із параметрів системи ми можемо зробити висновок, що в системі виникла аномалія.

Для створення шаблону поведінки користувача використовуються наступні параметри: кількість часу, протягом якого зазвичай працює користувач, набір вузлів, з яких користувач починає робочий сеанс, характеристики використання системних ресурсів [10]. Ці параметри оцифровуються та вводяться в нейронну мережу зворотного поширення помилки (BPNN). Результатом буде коефіцієнт нуля для користувачів із нормальною поведінкою та 1 для аномальної поведінки. Тобто мережа навчається з парою типів (параметр «нормальний», 0) і (параметр «ненормальний», 1. Отримання «ненормальної» поведінки вимагає змусити користувача поводитися не так, як він звик, тому аномальні дані генеруються випадковим чином і ускладнюють порівняння результатів відносно роботи з реальними данними. У дослідженні [11] запропонована система ідентифікує користувачів на основі обмежених команд - 100.

Враховується кількість запусків кожної команди, а не їхня послідовність. Ці числа, закодовані певним чином, стають вхідними даними для BPNN, якими вона навчається зі своїм учителем. Результатом є ідентифікатор користувача (UNIX має числове значення, але параметр, який штучно зближує користувачів із подібними ідентифікаторами). Якщо користувача буде ідентифіковано неправильно, про це буде повідомлено адміністратору. Але з лише 10 користувачами результати були хорошими. Цю ситуацію називають «тепличною», оскільки в реальній системі кількість користувачів може досягати

тисяч, і більшість із них виконують однакові дії, що ускладнює розрізнення між користувачами таким чином.

Підхід на основі BPNN також запропоновано в [12]. На відміну від [10, 11], тут не обмежувалися вузьким діапазоном користувачів, кількість команд була більшою (до 512 команд), і не використовувалися штучні дані. Мережа використовувала наступну, команду сеансу користувача, використовуючи попередні n-команд. Через невпорядкований характер набору команд було запропоновано компромісне кодування. Кожній команді було призначено вектор (двійковий запис номерів команд після будь-якої перенумерації всіх команд).

Таким чином, автори частково усунули проблеми штучного впорядкування при призначенні скалярних значень командам і отримали досить економний метод кодування. Після навчання з входами та виходами ,які відповідають контексту кількох попередніх команд, які вміщуються у вікно шириною n, мережа повинна була використовувати доступний контекст, щоб обчислити наступну. Якщо відносна кількість правильно передбачених команд у сеансі не перевищує певного значення, це вказує на аномалію. Щоб забезпечити адаптивність, нейронна мережа дотреновувалась після кожного сеансу. Хоча цей результат демонструє доцільність застосування цього підходу та ефективність обраного методу кодування команд, нерегулярність поведінки користувача значно підвищує рівень хибних спрацьовувань.

Очікується, що використання даних із регулярніших джерел, таких як системні процеси, дасть кращі результати. Недоліком багатьох нейронних мереж є те, що вони погано підходять для роботи з невпорядкованими значеннями. Оскільки нейронна мережа враховує числову близькість числових величин, введення штучного порядку до набора значень елементів лише спотворить картину.

1.5.3 Статистичний метод

Дана група методів заснована на побудові статистичного профілю поведінки системи протягом деякого періоду «навчання», при якому поведінка системи вважається нормальним [8, 9]. Для кожного параметра функціонування системи будується інтервал допустимих значень, з використанням деякого відомого закону розподілу. Далі, в режимі виявлення, система оцінює відхилення спостережуваних значень від значень, отриманих під час навчання. Якщо відхилення перевищують деякі задані значення, то фіксується факт аномалії (атаки).

Прикладом таких методів може бути інтервальний метод. З цією метою розбиваємо область можливих значень величини X (потік подій) на B частин де, для вибіркового середнього ознакою аномалії будемо вважати перевищення заданого порогу при відхиленні величини від її середнього значення. Докладно приклад розглядається в статті [10]. У цьому прикладі використовується статистика характеристики подій X. Величина X підкоряється відомому розподілу X – розподілу з (B1) ступенями свободи, де B – число подій в потоці.

У такому випадки ознакою появи аномалії будемо вважати перевищення величиною X встановленого порогового значення. З усього цього випливає, що для статистичного аналізу характерний високий рівень помилкових спрацьовувань при використанні в локальних мережах, де поведінка об'єктів не має гладкого, усередненого характеру. Крім того, даний метод стійкий тільки в межах конкретної системи, тобто побудовані статистичні профілі можна використовувати на інших аналогічних системах.

1.5.4 Сигнатурні методи

Найбільш часто використовувана група методів, суть яких полягає в складанні деякого алфавіту з спостережуваних в системі подій і описі безлічі сигнатур атак у вигляді регулярних виразів (у загальному випадку) в побудованому алфавіті. Як правило, сигнатурні методи працюють на

найнижчому рівні абстракції і аналізують дані, які безпосередньо передаються по мережі, параметри системних викликів і запису файлів журналів. Принцип роботи даного методу розглянутий в статті. Сигнатурні методи примітні тим, що для них добре застосовні апаратні прискорювачі, але при цьому метод не є адаптивним.

1.5.5 Метод на основі кластерного аналізу

Суть даної групи методів полягає в розбитті безлічі спостережуваних векторів-властивостей системи на кластери, серед яких виділяють кластери нормального поведінки [11]. У кожному конкретному методі кластерного аналізу використовується своя метрика, яка дозволяє оцінювати приналежність спостережуваного вектора властивостей системи одному з кластерів або вихід за межі відомих кластерів. Згідно [12], методи кластерного аналізу можна розділити на 3 класи: – ієрархічні методи, – методи розбиття, – комбіновані методи.

Результатом роботи ієрархічного методу є ієрархія кластерів (таксономія). Самі кластери можуть бути отримані шляхом розбиття дерева на основі певного критерію. Результатом роботи алгоритму, заснованого на методі розділення, є безліч кластерів, не пов'язаних між собою певною ієрархією.

Комбіновані методи використовують ієрархічні методи, і методи розбиття. Найбільш часто використовується послідовно спочатку метод розбиття, а потім будується ієрархія на підставі отриманих кластерів (наприклад, [10, 9]). Даний алгоритм є найкращім з даної групи за рахунок двоступеневої кластеризації великих обсягів даних, що працює на обмеженому обсязі пам'яті, є локальним алгоритмом. Перевагою даної групи методів є те, що він є адаптивним. Недоліком є сильна залежність результату від вибору кількості кластерів і початкового розташування кластерів.

Висновок до розділу 1

У даному розділі були розглянуті основні класифікації аномалій, які охоплюють різні типи виявлення аномальних подій: від заснованих на правилах та статистичних методів до використання методів машинного та глибинного навчання. Також були розглянуті відомі методи виявлення аномалій

РОЗДІЛ 2 МЕТОД КЛАСТЕРНОГО АНАЛІЗУ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1 Кластеризація даних

Кластеризація, або кластерний аналіз, — це процес поділу сукупності об’єктів на групи, які називаються кластерами. У межах кожного кластера об’єкти повинні бути максимально схожими, тоді як між об’єктами з різних груп повинні бути суттєві відмінності. Основна відмінність кластеризації від класифікації полягає в тому, що при кластеризації перелік груп заздалегідь не визначається, а формується в ході роботи алгоритму.

Використання кластерного аналізу в загальному вигляді зводиться до наступних етапів:

1. Відбір вибірки об’єктів для кластеризації.

2. Визначення множини змінних, за якими будуть оцінюватися об’єкти в вибірці. При необхідності – нормалізація значень змінних.

3. Підрахунок значень міри схожості між об’єктами.

4. Використання методу кластерного аналізу для створення схожих об’єктів (кластерів).

5. Представлення та інтерпретація результатів аналізу.

Після отримання і аналізу результатів можливий вибір необхідної метрики та методу кластеризації для отримання оптимального результату.

Як же визначити схожість об’єктів? Для початку необхідно створити вектор характеристик для кожного об’єкту – як правило, це набір числових даних, зокрема таких, що отримані методами статистичного аналізу. Також існують алгоритми, які працюють з якісними характеристиками.

Після того, як визначено вектор характеристик, можна провести нормалізацію, щоб всі компоненти давали однаковий внесок при розрахунку відстаней. В процесі нормалізації всі значення приводяться до деякого діапазону,

наприклад [-1,1] або [0,1]. Для кожної пари об’єктів визначається —відстань між ними – ступінь схожості.

Основні метрики для визначення ступеня схожості: 1. Евклідова відстань. Найпопулярніша функція відстані. Представляє собою геометричну відстань в багатовимірному просторі

$$p(x, x') = \sqrt{\sum_i^n (x_i - x'_i)^2} \qquad (2.1)$$

2. Квадрат евклідової відстані. Використовується для додання більшої ваги більш віддаленим один від одного об’єктам

$$(x, x') = \sum_i^n (x_i - x'_i)^2 \qquad (2.2)$$

3. Taxicab metric (відстань кварталів) - це метрика, яка використовується для вимірювання відстані між двома точками на сітці (наприклад, у двовимірному просторі), де рух дозволений лише по вертикалі та горизонталі, а не по діагоналі. В більшості випадків ця міра відстані приводить до результатів, які подібні до результатів отриманих при використанні відстані Евкліда. Але для цієї міри вплив окремих великих викидів зменшується.

$$(x, x') = \sum_i^n |x_i - x'_i| \qquad (2.3)$$

4. Відстань Чебишева. Ця відстань може бути корисною, коли необхідно визначити два об’єкта як —різні‖, якщо вони відрізняються по якійсь з однієї координат.

$$p(x, x') = \max (|x_i - x'_i|) \qquad (2.4)$$

Вибір метрики повністю залежить від користувача, оскільки результати кластеризації можуть суттєво відрізнятися при використанні різних мір[15].

2.2 Основні підходи, що використовуються в кластерному аналізі

На сьогодні існує кілька підходів до розв'язання задач кластерного аналізу, які ґрунтуються на різних способах трактування проблеми, застосуванні додаткових даних, характерних для конкретних предметних галузей, та інших аспектах. Наведемо короткий огляд найбільш поширених підходів. Важливо зазначити, що подана далі класифікація є умовною: окремі методи можуть поєднувати елементи різних підходів.

1) Імовірнісний підхід базується на припущенні, що кожен об'єкт у популяції належить до одного з K класів, але номери цих класів заздалегідь невідомі. Об'єкти обираються випадково та незалежно один від одного, тому змінні, які їх описують, розглядаються як випадкові. Для кожного класу визначається ймовірнісний розподіл, який належить до певної сімейства розподілів, причому параметри цього розподілу невідомі. Наявна вибірка спостережень є реалізацією суміші таких розподілів.

2) Підхід з використанням аналогії з центром ваги. Для кожної групи визначається вектор середніх значень показників, інтерпретується як «центр ваги» групи. Використовується критерій внутрішньо групового розсіювання: де координата «центру ваги» k-го кластера в змінній Xj, j = 1,2, ..., n, k = 1,2, ..., K. Оптимальне групування для даного K відповідає мінімальному значенню критерію;

3) Підхід, що базується на теорії графів, передбачає використання методів графового аналізу. Найвідомішим алгоритмом цього типу є алгоритм найкоротшого відкритого шляху. Спочатку будується мінімальне прольотове дерево графа, де вершини представляють об'єкти, а ребра характеризуються довжиною, що відповідає відстані між об'єктами. Для формування кластерів із дерева видаляються ребра з найбільшою довжиною

4) Ієрархічний підхід також тісно пов'язаний із графотеоретичними методами. Результати кластеризації в цьому випадку представляються у вигляді дерева кластерів, або дендрограми. Алгоритми цього напряму поділяються на агломеративні, які поступово об'єднують найближчі об'єкти чи групи, та

подільні, що починають із загальної групи і послідовно ділять її на підгрупи, найбільш віддалені одна від одної, з подальшим поділом підгруп. Таким чином, формується вкладена ієрархія підгруп.

5) Підхід, заснований на концепції найближчого сусіда**.** У цьому методі кластеризація здійснюється шляхом поступового додавання об'єктів до кластерів. Об'єкт приєднується до кластера, в якому знаходиться найближчий елемент, за умови, що відстань між ними не перевищує заданий поріг. Визначення відстані може варіюватися, і при оцінюванні близькості іноді враховується положення інших сусідніх точок.

6) Алгоритми нечіткого кластерного аналізу**.**У рамках цього підходу кожен кластер розглядається як нечітка множина об'єктів, що дозволяє одному об'єкту належати до кількох кластерів із різним ступенем належності.

7) Підхід із використанням штучних нейронних мереж**.** Цей метод базується на аналогії з біологічними нейронними системами. У багатьох алгоритмах такого типу використовується одношарова мережа, де кожен нейрон відповідає конкретному кластеру. У процесі навчання мережі ваги зв'язків між вхідними і вихідними вузлами ітеративно змінюються, оптимізуючи критерій кластеризації. Штучні нейронні мережі також підтримують ефективну реалізацію паралельних обчислень.

8) Еволюційний (генетичний) підхід**.** Алгоритми цього типу імітують принципи природної еволюції. Вони працюють із популяцією, яка складається з різних варіантів кластеризації, або хромосом. Еволюційні оператори, такі як відбір, рекомбінація та мутація, використовуються для створення нових рішень (нащадків) із наявних. Генетичні алгоритми спрямовані на знаходження глобального мінімуму функції якості кластеризації.

Серед додаткових методів можна виокремити моделювання відпалу, яке базується на ідеї імітації природних процесів, і може бути ефективним у задачах глобальної оптимізації, пов'язаних із кластерним аналізом[16].

2.3 Метод кластеризації K-середніх

Алгоритм k-means — метод кластерного аналізу, метою якого є розподіл m спостережень на k кластерів, при цьому кожне спостереження відноситься до того кластеру, до центру (центроду) якого воно ближче всього. Алгоритм к-середніх потребує заздалегідь визначено кількості кластерів.

Таким чином, метод кластеризації об'єднює об'єкти в кластери відповідно до характеристик точок даних, при цьому кожна точка даних в кластері ідентична тим, які знаходяться в тому ж кластері, але відрізняється від інших кластерів. З цієї причини кластеризація є однією з найкращих концепцій в області безконтрольованого навчання, оскільки виявлення аномалій в цілому є безконтрольованим виявленням.

Метою метода к-середніх є знаходження прототипу для кожного кластера, і потім всі об'єкти даних назначаються найближчому прототипу, який формує кластер. Прототип називається центроїдом, центром кластера. Центроїд — це середнє значення всіх об'єктів даних в кластері. Щоб розділити дані, необхідно визначити міру близкості. Найчастішою мірою для числових атрибутів є евклідова відстань.

Процес кластеризації методом k-means складається з таких етапів:

1) Визначення кількості кластерів.

2) Ініціалізація центроду. Першим кроком в алгоритмі k-середніх після визначення кількості кластерів є визначення центродів. В дані роботі пропонується використати формулу середнього як описання середнього значення кластерів, які показані в формулі:

$$\mu_i = \frac{x_1 + x_2 + x_3 + \cdots + x_j}{j} \quad (2.5)$$

Де x_j - значення об'єкта в кластері та j - кількість об'єктів в кластері, а формула для визначення медіани є наступною:

$$ME = \frac{x_n + 1}{2} \qquad (2.6)$$

3) Розподіл даних. Всі дані або об'єкти будуть розташовані у наближчому кластері. Відстань між двума об'єктами визначає близкість об'єкта. Формула Евклідово відстані:

$$d = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \ldots + (x_n - c_n)^2} \qquad (2.7)$$

де d - відстань точок x і c, x_n- дані критерію, c_n- центроїд n-го кластера.

4) Оновлення центроїдів — це ключовий етап методу k-means, де обчислюються нові координати центроїдів на основі точок, які були призначені до кожного кластеру. Математично це крок можна виразити як мінімізацію суми квадратних помилок від усіх точок даних кластера до цетродного кластера. Загальною метою цього кроку є мінімізація суми квадратних помилок кожно групи. Нова формула визначення цетроїду:

$$\mu_i = \frac{1}{ji} \sum_{x \in C_i} X \qquad (2.8)$$

Де μ_i- центроїд, X - об'єкт у кластері, ji- кількість об'єктів у клатері.

Етап обчислення нового центрода та крок присвоєння точок даних новому центроду повторюється до тих пір, поки не перестануть відбуватися зміни в призначенні точок даних. Останній центрод використовується як прототип кластера і використовується для опису всіє моделі групування.Етапи оригінального алгоритму к-середніх зображені на рис. 2.1(а), а алгоритм k-means

який рикористовується в даній роботі з використанням формул медіани і середнього проілюстровані на рис. 2.1(б)[15].

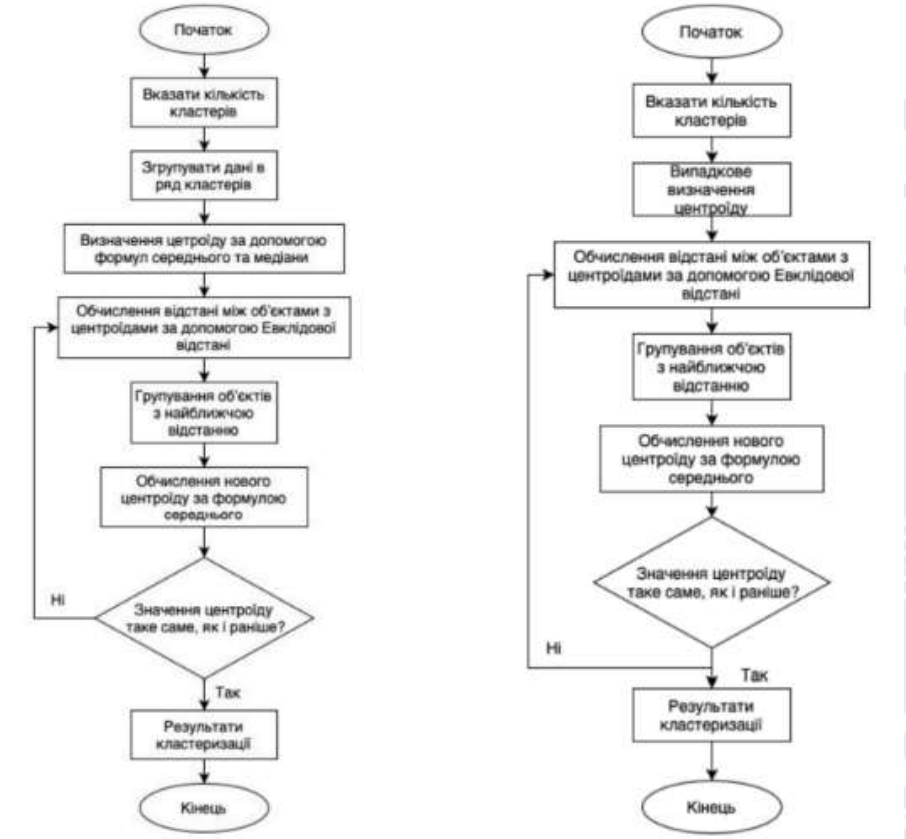

Рисунок 2.1 — а) Оригінальний алгоритм к-середніх; б) алгоритм k-means з використанням формул медіани і середнього

2.4 Перевірка кількості кластерів

Метод кластеризації k-means застосовується для визначення оптимальної кількості кластерів у наборі даних. Цей підхід є візуальним способом оцінки кількості кластерів. Суть методу полягає у тому, щоб:

1. Задати початкову кількість кластерів.
2. Поступово збільшувати кількість кластерів.
3. Для кожного розподілу розраховувати суму квадратів відхилень (SSE).

Сумарна квадратична помилка (SSE) визначає, наскільки близькими є точки всередині кластерів до їх центроїдів.

Результати зображаються графічно: на осі x – кількість кластерів, на осі y – значення SSE. Зменшення SSE зазвичай стає менш значним із кожним новим кластером. «Метод ліктя» визначає точку на графіку, де зниження SSE сповільнюється найвідчутніше, утворюючи кут («лікоть»). Ця точка вважається оптимальною кількістю кластерів.

Формула для обчислення SSE виглядає так:

$$SSE = \sum_{i=1}^{k}, \sum_{x_j \in C_i}^{1} \left\| x_j - \mu_i \right\|^2 \quad (2.9)$$

Де C_i— точки, що належать кластеру, i, u_i — центроїд цього кластеру, а $\left\| x_j - \mu_i \right\|^2$квадрат відстані від точки до центроїда[4].

Алгоритм для визначення оптимальної кількості кластерів (k):

1. Встановити початкове значення k=1k = 1k=1.
2. Запустити алгоритм кластеризації kkk-means.
3. Збільшити значення kkk.
4. Обчислити суму квадратів відхилень (SSE).
5. Визначити значення kkk, при якому спостерігається різке зменшення SSE — це і є оптимальна кількість кластерів.
6. Завершити процес.

Аналіз результатів :

Як показано на Рис. 2.2, оптимальна кількість кластерів для цього набору даних дорівнює 2.

Якщо б графік показав доволі гладку криву, доцільним було б вибрати інший метод визначення кластерів, такий як silhouette method (метод визначення силуетів), або переглянути чи правильним рішенням є кластеризація для даного набору данних.

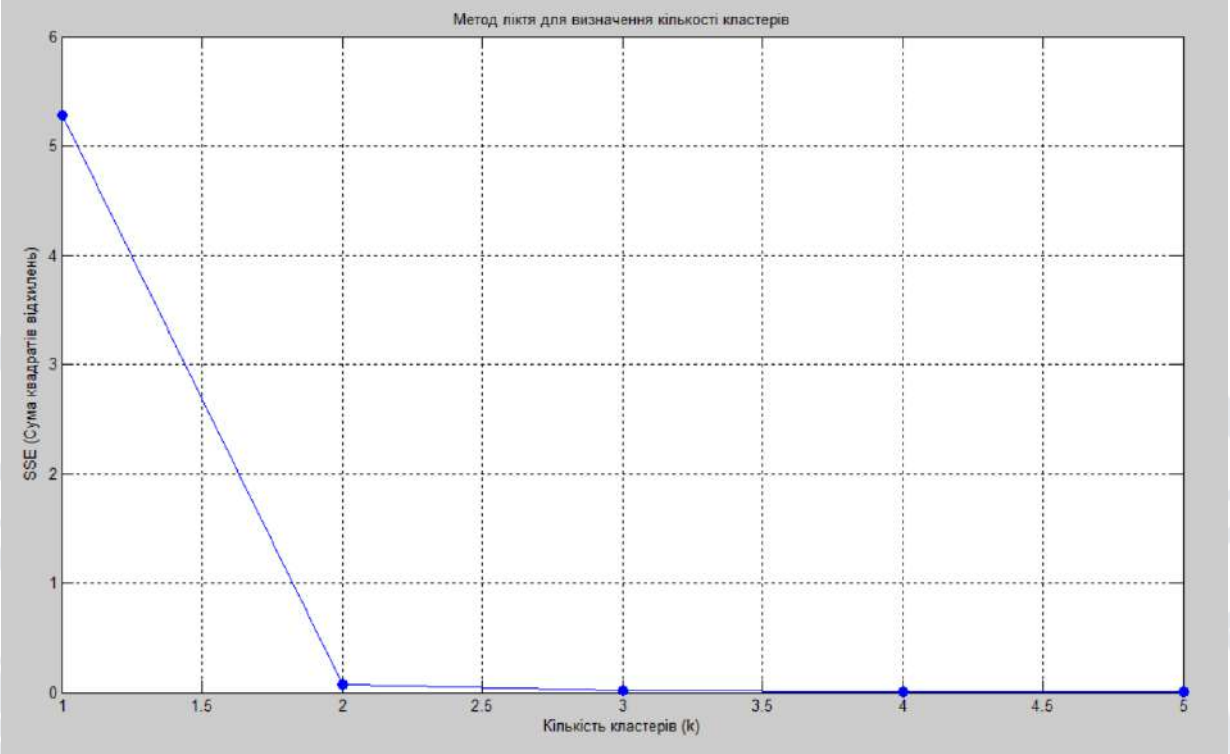

Рисунок 2.2 — SSE метод для визначення кількості кластерів.

2.5 Компоненти технології виявлення аномалій в електромережі за допомогою кластеризації

Для маштабування технології виявлення аномалій в електромережі міста за допомогою методу кластеризації можуть використовуватись наступні компоненти:

IoT-сенсори:

- Сенсори встановлюються в ключових точках енергомережі (лінії електропередач, трансформаторні підстанції, кінцеві пристрої споживачів).
- Вони збирають дані про:
  - Напругу та частоту в мережі.
  - Температуру та стан обладнання.
  - Споживання електроенергії в реальному часі.
- Дані передаються через мережу до централізованої платформи.

***Аналіз великих даних (Big Data Analytics)**:*

- Величезні обсяги даних, зібраних сенсорами, аналізуються за допомогою методу кластеризації
- Ці алгоритми виявляють:
    - Патерни нормального енергоспоживання.
    - Аномалії (раптові піки, провали або нестабільності).

***Автоматизоване управління**:*

- У разі виявлення аномалії система може автоматично реагувати:
    - Перенаправляти потоки енергії.
    - Знижувати навантаження на окремі ділянки мережі.
    - Вмикати резервні джерела живлення.

***Комунікаційна інфраструктура**:*

- Платформи для передачі даних через мобільний зв'язок, Wi-Fi, оптоволоконні лінії чи спеціалізовані енергомережі.
- Ця інфраструктура забезпечує обмін даними між сенсорами, аналітичними центрами та операторами[17].

Висновок до розділу 2

У цьому розділі було проведено загальний огляд методу кластеризації k-means та його переваги для виявлення кібератак. Запропоновано оптимізувати стандартний метод к-середніх методом з використанням формул медіани та середнього. Визначенно кількість кластерів візуальним методом — суми квадратів помилки , який спостерігає найнижче значення SSE, формуючи «згиб». Також проведенно перелік компонентів масштабування технології виявлення аномалій у енергосистемі міста

РОЗДІЛ 3. ВИЗНАЧЕННЯ АНОМАЛІЙ В ЕНЕРГОСПОЖИВАННІ МІСТА

3.1 Характеристика аномального енергоспоживання

Нормальне енергоспоживання — це споживання електроенергії, яке не виходить за межі певного діапазону, що базується на середньому значенні та стандартному відхиленні даних за певний період часу.

Аномальне енергоспоживання — це значення, яке суттєво відхиляється від середнього значення і може вказувати на несправності в мережі або незвичні обставини. Для визначення таких відхилень ми використовуємо метод кластеризації, який визначає, наскільки сильно значення відхиляється від стандартоного діапазону значень.

3.2 Розробка програмного забезпечення для виявлення аномалій в енергосистемі

Для розробки даного алгоритму ми будемо притримуватись плану
Опис алгоритму:

Збір даних:

- Дані енергоспоживання збираються протягом 24 годин для трьох районів міста — Тяжилів, Вишенька та Київська. Вони вимірюються щогодини.

Дані представленні у таблиці 3.1 ,були взяті на офіційному сайті

https://data.gov.ua/dataset/2-58-_19_-dahi-npo-cnoxnbahhr-komyhajibhnx-pecypcib1 - дані про споживання електроенергії

Також данну інформацію можна напряму збирати з енергостанцій міста за допомогою відповідних приладів

Дані енергоспоживання представлені у вигляді матриці, де кожний рядок відповідає конкретному району, а кожен стовпець — годині доби.

Завдання полягало у визначенні, які години містять аномальні значення споживання для кожного району.

Особливості задачі:

1. Дані мають часову природу, однак у межах роботи аналізується їх розподіл значень.
2. Дані можуть мати різні масштаби, тому була потрібна нормалізація.

Для розробки програми використовуєм середовище програмування MATLAB версії 7.12.0.635 (R2011a)[18].

Таблиця 3.1 – Споживання енергії в районах міста

Години	Споживання енергії кВт/год Тяжилів	Споживання енергії кВт/год Вишенька	Споживання енергії кВт/год Київська
01	10	15	25
02	12	17	24
03	11	16	26
04	9	18	23
05	10	19	27
06	11	17	26
07	10	16	24
08	12	200	300

09	50	17	25
10	11	15	26
11	13	16	24
12	12	18	23
13	14	19	26
14	12	15	25
15	11	17	23
16	13	16	300
17	12	18	26
18	50	19	25
19	11	200	24
20	10	17	23
21	11	16	27
22	13	18	26
23	12	19	24
24	14	17	25

3.3 Визначення нормальних і аномальних значень

Нормалізація даних:

Щоб підготувати дані до кластеризації, було реалізовано автоматичну нормалізацію. Це дозволило звести всі значення в діапазон [0,1][0, 1][0,1], незалежно від абсолютного масштабу показників.

Формула нормалізації:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)'} \qquad (3.1)$$

де x' — поточне значення, min(X) і max(X) — мінімум і максимум в ряді відповідного району[19].

- Рис.3.1 Для кожного району дані енергоспоживання нормалізуються у діапазон [0, 1] для покращення результатів кластеризації. Це допомагає враховувати різницю в діапазоні енергоспоживання між районами.

```
% Автоматична нормалізація даних для покращення кластеризації
norm_energy_consumption = zeros(size(energy_consumption));
for i = 1:num_districts
    norm_energy_consumption(i, :) = (energy_consumption(i, :) - min(energy_consumption(i, :))) / (max(energy_consumption(i, :)
end
```

Рисунок 3.1- Нормалізація данних для кластеризації

Визначення кількості кластерів

Метод кластеризації k-means потребує визначення кількості кластерів перед виконанням алгоритму. Для вибору оптимального k було реалізовано метод, який аналізує залежність **SSE** (суми квадратів відхилень) від кількості кластерів.

Процедура:

1. Для k від 1 до 5 було виконано кластеризацію.
2. Обчислено SSE для кожного k.
3. Побудовано графік SSE(k) для візуального визначення.

Результати: Графік показав, що оптимальна кількість кластерів дорівнює 2, що відповідає логіці задачі (нормальні значення та аномалії).

- За допомогою методу визначаємо оптимальну кількість кластерів для нашого набору данних. На рисунку 3.2 показано на графіку що оптимальна кількість кластерів для нашого набору k=2

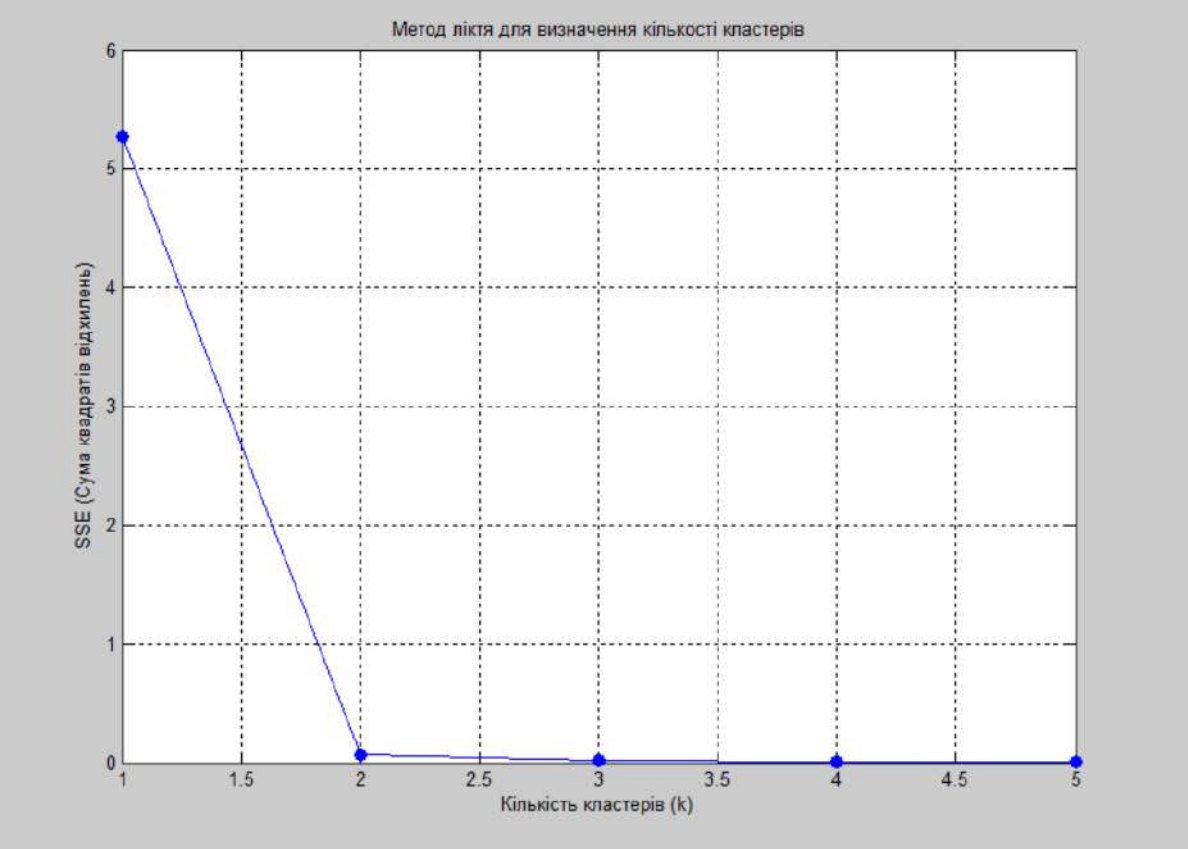

Рисунок 3.2 – Визначення кількості кластерів

3.4 Кластеризація методом k-means:

Алгоритм k-means був використаний для розподілу даних на два кластери: нормальні значення та аномалії.

Етапи алгоритму:

1. Ініціалізація кластерних центрів випадковими значеннями.
2. Розподіл точок до найближчого кластера за метрикою Евкліда:

$$d(x,c) = \sqrt{\sum_{i=1}^{n}(x_i - c_i)^2} \qquad (3.2)$$

де x— точка, c — центр кластера.

3. Перерахунок центрів як середнє точок у кожному кластері.
4. Повторення кроків 2–3, доки центри не стабілізуються[20].

Особливість реалізації:

- Аномальним кластером визначено той, центр якого мав більше значення.
- Алгоритм k-means ділить дані на 2 кластери Рис.3.3 один відповідає нормальному споживанню, інший — аномальному.

```
% Використовуємо метод кластеризації k-means для виявлення аномалій
% Об'єднуємо всі дані в одну матрицю для кластеризації
reshaped_data = norm_energy_consumption(:);

% Використовуємо k-means з двома кластерами (норма та аномалії)
[cluster_idx, cluster_centers] = kmeans(reshaped_data, 2);
```

Рисунок 3.3 - Ділимо вхідні дані на кластери

- Після кластеризації виявляється кластер з більшими значеннями Рис3.4 (він вважається аномальним).

```
% Визначення аномального кластера (кластер з більшими значеннями - аномалії)
[~, anomaly_cluster] = max(cluster_centers);
```

Рисунок 3.4 – Визначення аномальних значень енергоспоживання

- Результати кластеризації перетворюються назад у матрицю, що відповідає годинам і районам рис.3.5.

```
% Перетворюємо результати кластеризації назад у структуру районів
cluster_matrix = reshape(cluster_idx, num_districts, num_hours);

% Ініціалізуємо змінні для збереження аномальних годин
anomalies = cell(num_districts, 1);

% Визначаємо години від 00 до 23
hours = 0:23;

figure;
hold on;
```

Рисунок 3.5- Перетворення даних у матрицю

- Визначаєм індекс аномальних значень рис.3.6. визначається, який кластер є аномальним, на основі порівняння центрів кластерів. Центр з більшим значенням вважається аномальним

```
for i = 1:num_districts
    % Визначаємо індекси аномальних значень для району
    anomaly_indices = find(cluster_matrix(i, :) == anomaly_cluster);

    % Зберігаємо інформацію про аномальні години
    anomalies{i} = anomaly_indices;
```

Рисунок 3.6- Визначення індексів аномальних значень

Виведення результатів:

Для кожного району було побудовано графіки енергоспоживання:

- Сині точки представляють нормальні значення.
- Червоні точки позначають аномалії.

Крім графіків, програма генерує текстовий звіт, що вказує години з аномальними значеннями та їх величини.

- Програма виводить години і райони, де було виявлено аномальні відхилення рис.3.7, рис.3.8.

```
fprintf('Звіт про аномалії енергоспоживання:\n');
for i = 1:num_districts
    if ~isempty(anomalies{i})
        fprintf('Район: %s\n', districts{i});
        fprintf('Аномальні години: %s\n', num2str(hours(anomalies{i})));
        for j = 1:length(anomalies{i})
            fprintf('Час: %02d:00, Енергоспоживання: %.2f\n', hours(anomalies{i}(j)), energy_consumption(i, anomalies{i}(j)));
        end
        fprintf('\n');
    else
        fprintf('Аномалій не виявлено в районі %s.\n\n', districts{i});
    end
end
```

Рисунок 3.7- Звіт про аномальні споживання енергії

```
Звіт про аномалії енергоспоживання:
Район: Тяжилів
Аномальні години: 8  17
Час: 08:00, Енергоспоживання: 50.00
Час: 17:00, Енергоспоживання: 50.00

Район: Вишенька
Аномальні години: 7  18
Час: 07:00, Енергоспоживання: 200.00
Час: 18:00, Енергоспоживання: 200.00

Район: Київська
Аномальні години: 7  15
Час: 07:00, Енергоспоживання: 300.00
Час: 15:00, Енергоспоживання: 300.00
```

Рисунок 3.8 – Вивід інформації про аномальні відхилення

- Візуалізація даних з підсвічуванням аномалій рис.3.9.

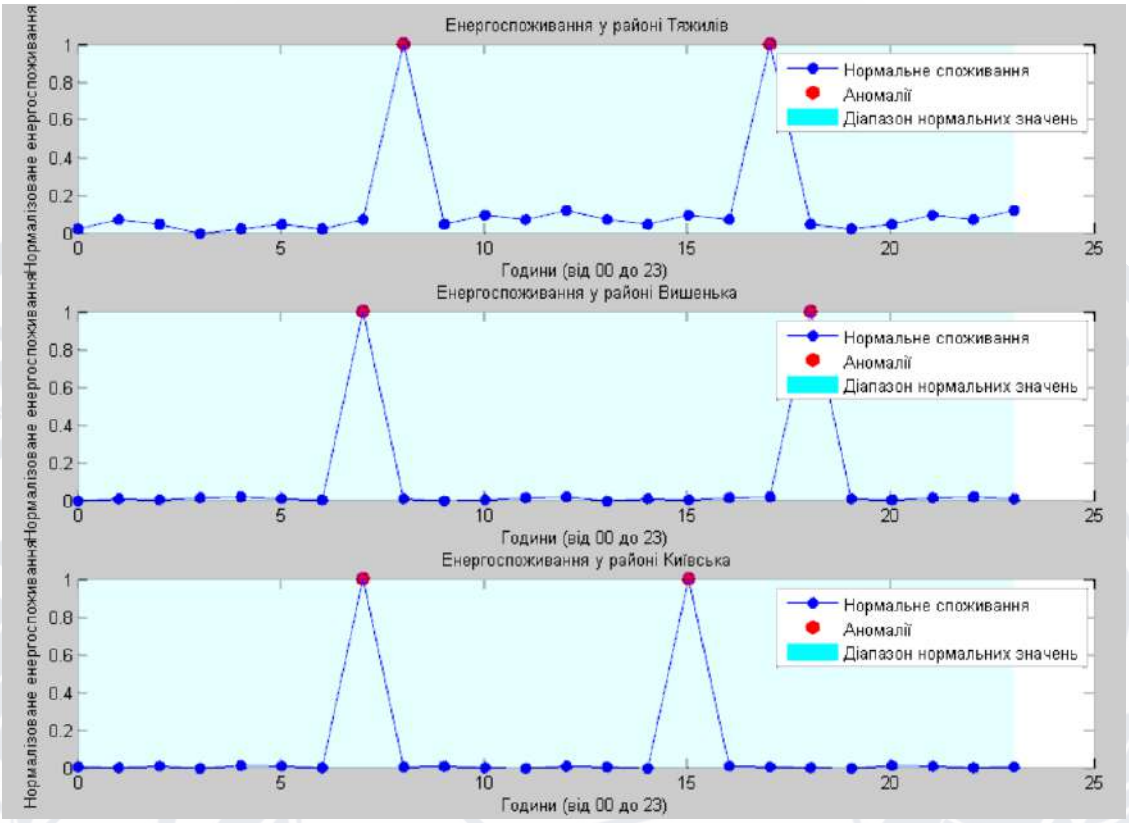

Рисунок 3.9- Графіки споживання енергії за аномальними відхиленнями

Програма успішно виявила аномалії в даних енергоспоживання:

- У районі **Тяжилів** аномалії спостерігалися о 8:00 і 17:00.
- У районі **Вишенька** аномалії зафіксовано о 7:00 і 18:00.
- У районі **Київська** аномалії спостерігались о 7:00 і 15:00.

Переваги підходу:

1. **Точність**: k-means допомагає чіткіше розподіляти дані на класи, навіть якщо аномалії не є простими відхиленнями від середнього.
2. **Гнучкість**: алгоритм k-means дозволяє автоматично адаптуватися до змін у даних і знаходити патерни навіть у складних ситуаціях.

3. **Масштабованість**: нормалізація даних забезпечує ефективну обробку даних різних масштабів, що є корисним для великих проєктів IoT, таких як "Smart City".

Це робить систему для виявлення аномалій більш серйозною та ефективною для реального використання.

Висновок до розділу 3

Розроблена програма демонструє ефективність у виявленні аномалій енергоспоживання. Вона поєднує потужний метод кластеризації k-means із принципами нормалізації даних та автоматичного вибору кількості кластерів. Графічна візуалізація та текстові звіти роблять програму зручною для використання в реальних умовах.

У майбутньому може бути проведена інтеграція з реальними датчиками енергоспоживання та розширення алгоритму для врахування тимчасових залежностей.

ВИСНОВКИ

У данній роботі проведено аналіз класифікації аномалій в технічних системах та методів їх виявлення в технічних системах

Для розробки програми для виявлення аномалій було підібрано да проаналізовано оптимальний метод виявлення аномалій за допомогою кластеризації методом «k-середніх» та приведено компоненти для масштабування технології виявлення аномалій у енергосистемі за допомогою методу кластеризації

Розроблено програму яка визначає аномальні активності в системі енергоспоживання міста за допомогою методу кластеризації та може інтергруватись в реальні системи датчиків збору інформації в енергетичних системах

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Sub biah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakan tan, Pranav Shyam, Girish Sastry, Amanda Askell, Sand hini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Advances in Neural Information Pro cessing Systems, pages 1877-1901, 2020. 1, 8;

2. Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to end object detection with transformers. In European Confer ence on Computer Vision, pages 213–229. Springer, 2020. 3;

3. Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Pi otr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882, 2020. 1, 2, 4, 5, 6, 11;

4. Mathilde Caron, Hugo Touvron, Ishan Misra, Herve J´ egou, ´ Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerg ing properties in self- supervised vision transformers. arXiv preprint arXiv:2104.14294, 2021. 1, 2, 3, 4, 5, 6, 8, 11, 12, 13;

5. Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tian heng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155, 2019. 11;

6. Ting Chen, Simon Kornblith, Mohammad Norouzi, and Ge offrey Hinton. A simple framework for contrastive learning of visual representations. In ICML, 2020. 1, 4, 12;81

7. Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020. 1, 2, 4, 5, 6;

8. Chandola V., Banerjee A., Kumar V. *Anomaly Detection: A Survey.* ACM Computing Surveys, 2009.

9. Aggarwal C.C. *Outlier Analysis.* Springer, 2016.

10. Breunig M.M., Kriegel H.P., et al. *LOF: Identifying Density-Based Local Outliers.* ACM SIGMOD, 2000.

11. Ahmed M., Mahmood A.N., Hu J. *A Survey of Network Anomaly Detection Techniques.* Elsevier, 2016.

12. Zhang Z., Liu Y., et al. *Deep Learning-Based Anomaly Detection for IoT Systems.* IEEE IoT Journal, 2020.

13. Kim D., Park K. *Contextual Anomaly Detection in Cyber-Physical Systems.* Sensors, 2017.

14. Sommer R., Paxson V. *Outside the Closed World: On Using Machine Learning for Network Intrusion Detection.* IEEE Symposium, 2010.

15. Edy Umargono, Jadmiko Endro Suseno and Vincensius Gunawan S. K. K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median. —2019.

16. Junyuan Xie, Ross Girshick, Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. — 2016.

17. Інтелектуальні комунікації . URL: https://www.siemens.com/ua/uk/produkty/enerhetyka/avtomatizatsiya-intelektualni-merezhi/intelektualni-enerhomerezhi.html (дата звернення 03.09.2024)

18. MATLAB. URL: https://www.mathworks.com/products/matlab.html (дата звернення 05.09.2024)

19. Hodge, Victoria J., and Jim Austin. "A survey of outlier detection methodologies." *Artificial Intelligence Review* 22, no. 2 (2004): 85-126.

20. Towards Data Science. (2020, August 10). Understanding Data Normalization in Machine Learning. *Medium*. URL: https://towardsdatascience.com

ДОДАТОК А

Перевірка к-сті кластерів за допомогою «методу ліктя» у MATLAB

```
% Підбір кількості кластерів методом ліктя
max_clusters = 10; % Максимальна кількість кластерів для перевірки
WCSS = zeros(1, max_clusters);
% Побудова графіка методу ліктя
figure;
valid_k = ~isnan(WCSS); % Виключаємо значення з помилками
plot(1:max_clusters, WCSS, '-o', 'LineWidth', 2, 'MarkerSize', 8);
xlabel('Кількість кластерів (k)');
ylabel('WCSS (сума внутрішньокластерних відстаней)');
title('Метод ліктя для вибору кількості кластерів');
grid on;
% Нормалізація WCSS для точності аналізу
x = 1:max_clusters; % Кількість кластерів
y = WCSS;
% Обчислення динамічного нахилу (градієнта)
gradient_diff = diff(y); % Перша похідна
gradient_diff2 = diff(gradient_diff); % Друга похідна (прискорення)
```

```
% Знаходимо точку з найбільшим відхиленням
[~, elbow_point] = max(abs(gradient_diff2)); % Індекс точки коліна
optimal_k = x(elbow_point + 1); % Враховуємо зсув на +1 через похідну
% Відображення на графіку
hold on;
plot(optimal_k, WCSS(optimal_k), 'ro', 'MarkerSize', 10, 'LineWidth', 2);
text(optimal_k, WCSS(optimal_k), ['\leftarrow k = ', num2str(optimal_k)], 'FontSize', 12);
hold off;
fprintf('Оптимальна кількість кластерів (k): %d\n', optimal_k);
```

ДОДАТОК Б

Розробка програми виявлення аномалій у середовищі MATLAB

```
% Дані енергоспоживання за 24 години для трьох районів з аномаліями
energy_consumption = [
   10, 12, 11, 9, 10, 11, 10, 12, 50, 11, 13, 12, 14, 12, 11, 13, 12, 50, 11, 10, 11, 13, 12, 14; % Тяжилів
   15, 17, 16, 18, 19, 17, 16, 200, 17, 15, 16, 18, 19, 15, 17, 16, 18, 19, 200, 17, 16, 18, 19, 17; % Вишенька
   25, 24, 26, 23, 27, 26, 24, 300, 25, 26, 24, 23, 26, 25, 23, 300, 26, 25, 24, 23, 27, 26, 24, 25  % Київська
];
% Визначення районів
districts = {'Тяжилів', 'Вишенька', 'Київська'};
% Кількість районів і годин
num_districts = size(energy_consumption, 1);
num_hours = size(energy_consumption, 2);
% Автоматична нормалізація даних для покращення кластеризації
norm_energy_consumption = zeros(size(energy_consumption));
for i = 1:num_districts
   norm_energy_consumption(i, :) = (energy_consumption(i, :) - min(energy_consumption(i, :))) / (max(energy_consumption(i, :)) - min(energy_consumption(i, :)));
end
% Використовуємо метод кластеризації k-means для виявлення аномалій
% Об'єднуємо всі дані в одну матрицю для кластеризації
reshaped_data = norm_energy_consumption(:);
% Визначення оптимальної кількості кластерів методом ліктя
max_k = 5; % Максимальна кількість кластерів для аналізу
```

```
sse = zeros(max_k, 1); % Масив для збереження SSE
for k = 1:max_k
    [~, ~, sumd] = kmeans(reshaped_data, k, 'Replicates', 10); % Виконуємо k-means
    sse(k) = sum(sumd); % Зберігаємо SSE
end
% Побудова графіка методу ліктя
figure;
plot(1:max_k, sse, '-o', 'MarkerFaceColor', 'b');
title('Метод ліктя для визначення кількості кластерів');
xlabel('Кількість кластерів (k)');
ylabel('SSE (Сума квадратів відхилень)');
grid on;
% Для аналізу аномалій використовуємо 2 кластери
optimal_k = 2; % Обрано за методом ліктя
[cluster_idx, cluster_centers] = kmeans(reshaped_data, optimal_k, 'Replicates', 10);
% Визначення аномального кластера (кластер з більшими значеннями -
аномалії)
[~, anomaly_cluster] = max(cluster_centers);
% Перетворюємо результати кластеризації назад у структуру районів
cluster_matrix = reshape(cluster_idx, num_districts, num_hours);
% Ініціалізуємо змінні для збереження аномальних годин
anomalies = cell(num_districts, 1);
% Визначаємо години від 00 до 23
hours = 0:23;

figure;
hold on;

for i = 1:num_districts
    % Визначаємо індекси аномальних значень для району
```

```
    anomaly_indices = find(cluster_matrix(i, :) == anomaly_cluster);

    % Зберігаємо інформацію про аномальні години
    anomalies{i} = anomaly_indices;

    % Побудова графіку
    subplot(num_districts, 1, i);

    % Відображення нормалізованих даних
    plot(hours, norm_energy_consumption(i, :), '-o', 'Color', 'b', 'MarkerFaceColor',
'b');
    hold on;

    % Позначаємо аномальні значення червоним кольором
    if ~isempty(anomaly_indices)
        plot(hours(anomaly_indices), norm_energy_consumption(i, anomaly_indices),
'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 8);
    end

    % Відображення тіні, що позначає діапазон нормальних значень
    lower_bound = min(norm_energy_consumption(i, :));
    upper_bound = max(norm_energy_consumption(i, :));
    fill([hours, fliplr(hours)], [repmat(lower_bound, 1, num_hours),
fliplr(repmat(upper_bound, 1, num_hours))], 'cyan', 'FaceAlpha', 0.1, 'EdgeColor',
'none');
    title(['Енергоспоживання у районі ', districts{i}]);
    xlabel('Години (від 00 до 23)');
    ylabel('Нормалізоване енергоспоживання');
    legend('Нормальне споживання', 'Аномалії', 'Діапазон нормальних значень');
end
```

```
% Виводимо детальний звіт про аномалії
fprintf('Звіт про аномалії енергоспоживання:\n');
for i = 1:num_districts
    if ~isempty(anomalies{i})
        fprintf('Район: %s\n', districts{i});
        fprintf('Аномальні години: %s\n', num2str(hours(anomalies{i})));
        for j = 1:length(anomalies{i})
            fprintf('Час: %02d:00, Енергоспоживання: %.2f\n', hours(anomalies{i}(j)),
energy_consumption(i, anomalies{i}(j)));
        end
        fprintf('\n');
    else
        fprintf('Аномалій не виявлено в районі %s.\n\n', districts{i});
    end
end

hold off;
```

ДОДАТОК В

Характеристика немережевих аномалій у системі Windows (7-11)

Windows: підозрілі мережеві комунікації	Для того, щоб виявити спілкування процесів з зовнішніми IP адресами, що зазвичай відрізняється (профіль мережевої активності)
Windows: елементи запуску	Для того щоб виявити невідомі ознаки (елементи) запуску (профіль елементів запуску)
Windows: заплановані завдання	Для того щоб виявити невідомі заплановані завдання (профіль запланованих завдань)
Windows: запущені процеси	Для того, щоб виявити: - Діяльність невідомих дивних процесів; - Процес із невідомими шляхами; - Процеси із нестандартними хешами;
Windows: прослуховування портів	Для того, щоб виявляти невідомі процеси, що відкривають та прослуховують порти
Windows: etc / hosts	Для того, щоб виявляти підозрілі зміни etc / hosts файлів
Windows: cmd.exe та powershell.exe процеси	Для того, щоб виявляти аномальні процеси, що запускають cmd.exe та powershell.exe процес
Windows: завантаження драйверів	Для того, щоб моніторити всі драйвери, що завантажені із підозрілих сайтів

Windows: загрузка модулів	Для того, щоб моніторити всі модулі, що завантажені із підозрілих сайтів
Windows: сервіси	Для того, щоб виявляти нові сервіси або зміни в існуючих
Windows: батьківські та дочірні процеси	Для того, щоб виявляти батьківські та дочірні процеси

ДОДАТОК Ґ

Данні споживання електроенергії в районах міста

Години	Споживання енергії кВт/год Тяжилів	Споживання енергії кВт/год Вишенька	Споживання енергії кВт/год Київська
01	10	15	25
02	12	17	24
03	11	16	26
04	9	18	23
05	10	19	27
06	11	17	26
07	10	16	24
08	12	200	300
09	50	17	25
10	11	15	26
11	13	16	24
12	12	18	23
13	14	19	26
14	12	15	25
15	11	17	23
16	13	16	300
17	12	18	26
18	50	19	25

19	11	200	24
20	10	17	23
21	11	16	27
22	13	18	26
23	12	19	24
24	14	17	25