Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи та алгоритми узагальнення знань для систем підтримки прийняття рішень реального часу
(тема)

Виконав:
студент II курсу, групи СПм-19-1
Слюсар О.В.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: доц. Мартовицький В.О
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ ________ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп’ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри ________________
(підпис)

“____”________________ 20__ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Слюсару Олександру Володимировичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи та алгоритми узагальнення знань систем підтримки прийняття рішень реального часу

затверджена наказом по університету від “ 30 ” жовтня 2020 р. № 1486Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи Публікації алгоритми узагальнення знань;
Приклад опису класів ситуацій (об'єктів), що змінюються з часом.

4. Перелік питань, що потрібно опрацювати в роботі
1)Дослідження існуючих методів і алгоритмів уявлення та узагальнення знань в інтелектуальних системах підтримки прийняття рішень;
2)Розробка методів і алгоритмів узагальнення знань, що дозволяють отримувати загальний опис класів ситуацій (об'єктів), що змінюються з часом.
3)Вивчення можливості використання методів і алгоритмів узагальнення знань в інтелектуальних системах підтримки прийняття рішень реального часу.
4)Проектування і розробка програмного комплексу, що реалізує розглянуті в роботі методи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп’ютерних ілюстрацій (слайдів) Слайди презентації 16

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури та аналіз даних в інтелекту-	03.11.20-09.11.20	
	альних системах		
2	Узагальнення для динамічних об'єктів	10.11.20-17.11.20	
3	Дослідження тимчасових рядів	18.11.20-23.11.20	
4	Вивчення методів виявлення аномалій	24.11.20-01.12.20	
5	Опрацювання зашумлених даних	02.12.20-07.12.20	
6	Опрацювання алгоритму «TS-ADEEP-Multi»	08.12.20-09.12.20	
7	Опрацювання алгоритму «CPD»	10.12.20-11.12.20	
8	Впровадження нового алгоритму	12.12.20-13.12.20	

Дата видачі завдання 02 листопада 2020

Студент ____________________
(підпис)

Керівник роботи ____________________ доц. Мартовицький В.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 122 с., 25 рис., 22 табл., 1 дод., 24 джерела.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА, ТИМЧАСОВІ РЯДИ, АНОМАЛІЯ ДЕРЕВО РІШЕНЬ ІСППР, SAX, ID3, TS-ADEEP-MULTI, TS-ADEEP, FTP

Метою атестаційної роботи є є дослідження та розробка методів і алгоритмів узагальнення знань, що дозволяють отримувати узагальнений опис класів ситуацій, що змінюються з часом.

У ході виконання атестаційної роботи було вирішено такі завдання:

- дослідження існуючих методів і алгоритмів уявлення та узагальнення знань в інтелектуальних системах підтримки прийняття рішень;
- розробка методів і алгоритмів узагальнення знань, що дозволяють отримувати загальний опис класів ситуацій (об'єктів), що змінюються з часом;
- вивчення можливості використання методів і алгоритмів узагальнення знань в інтелектуальних системах підтримки прийняття рішень реального часу;
- розширення понятійного апарату: введення понять, які враховують динамічну природу об'єктів узагальнення; формалізація завдання узагальнення для роботи з динамічними даними;
- розробка методів і алгоритмів узагальнення знань для динамічних об'єктів узагальнення;
- проектування і розробка програмного комплексу, що реалізує розглянуті в роботі методи і алгоритми.

ABSTRACT

Bachelor’s thesis: 122 pages, 25 figures, XX tables, appendices, 24 sources.

INTELLECTUAL SYSTEM, TIME SERIES, ANOMALY TREE OF SOLUTIONS ISPR, SAX, ID3, TS-ADEEP-MULTI, TS-ADEEP, FTP.

The major goal of this thesis is the certification work is to study and develop methods and algorithms for generalizing knowledge, which allow to obtain a generalized description of classes of situations that change over time.

In order to during the certification work the following tasks were solved:

- research of existing methods and algorithms of representation and generalization of knowledge in intelligent decision support systems;
- development of methods and algorithms for generalization of knowledge that allow to obtain a general description of classes of situations (objects) that change over time;
- study the possibility of using methods and algorithms for generalizing knowledge in intelligent real-time decision support systems;
- expansion of the conceptual apparatus: the introduction of concepts that take into account the dynamic nature of the objects of generalization; formalization of the generalization task for working with dynamic data;
- development of methods and algorithms for generalization of knowledge for dynamic objects of generalization;
- design and development of a software package that implements the methods and algorithms considered in the work.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІСППР – інтелектуальна система підтримки прийняття рішень

ІС – інтелектуальна система

SAX – перетворення числового ряду в символьне

CBF – набір даних «циліндр-дзвін-воронка»

CPD – алгоритм побудови темпоральних дерев рішень

MLP – багатошаровий персептрон

SVM – метод опорних векторів

ВСТУП

Інтелектуальний аналіз даних на сьогоднішній день розвивається активно, у галузі штучного інтелекту, тісно пов'язаний з проблематикою машинного навчання і завданнями виявлення прихованих закономірностей. Найважливішим класом задач, рішення яких вимагає інтелектуальної підтримки комп'ютерних систем, є завдання управління складними технічними об'єктами. Головною рисою подібних об'єктів управління слід визнати те, що вони є динамічними, мають здатність до розвитку, стану таких об'єктів і систем можуть змінюватися з часом. Поява і розвиток засобів управління об'єктами, які належать до категорії динамічних, тісно пов'язане з розвитком інтелектуальних систем підтримки прийняття рішень (ІСППР), включаючи найбільш складних їх представників – ІСППР реального часу, основним напрямком розвитку яких є розробка динамічних моделей для представлення і маніпулювання знаннями про події, факти, дії, процеси, що відображають динаміку поведінки складного технічного об'єкта. Тому актуальною є задача розробки моделей подання знань, процедур узагальнення накопиченого досвіду та реалізації відповідних базових програмних засобів. Відомий цілий ряд методів і алгоритмів, здатних вирішувати завдання узагальнення: індукція вирішальних дерев, наближені множини, мережі Байеса і багато інших. У розробці таких методів брали участь видатні зарубіжні вчені Quinlan R., Pawlak Z., Mingers J., Utgoff P. Проте характерною особливістю таких методів є те, що результати узагальнення є статичними і не враховують такий важливий при діагностиці станів складної технічної системи фактор як час. Розробкою методів і алгоритмів, що враховують фактор часу, займаються такі зарубіжні вчені, як Console L., Picardi C., Dvorak P., Kuipers B., Sachenbacher M., Malik A., Dupret D, Keogh E., Pazzani M., Olszewski R., Geurts P. За допомогою таких моделей можна представляти не тільки статичні, але й динамічні знання про поведінку складного технічного об'єкту.

Інтелектуальні системи нового покоління орієнтуються на роботу з об'єктами, для яких характерна динамічна зміна станів. Індуктивні моделі, отримані на основі аналізу таких даних, повинні враховувати динаміку поведінки об'єкта, що є вкрай важливим, наприклад, при діагностиці поточного стану і прогнозуванні подальшої поведінки складної технічної системи.

Об'єктом досліджень є інтелектуальні системи підтримки прийняття рішень реального часу (ІСППР РЧ). Предметом досліджень – методи і алгоритми узагальнення знань, що дозволяють враховувати фактор часу, і їх застосування в ІСППР РЧ.

Метою даної роботи є дослідження та розробка методів і алгоритмів узагальнення знань, що дозволяють отримувати узагальнений опис класів ситуацій, що змінюються з часом.

Для досягнення поставленої мети необхідно було вирішити такі завдання:

- дослідження існуючих методів і алгоритмів уявлення та узагальнення знань в інтелектуальних системах підтримки прийняття рішень;
- розробка методів і алгоритмів узагальнення знань, що дозволяють отримувати загальний опис класів ситуацій (об'єктів), що змінюються з часом;
- вивчення можливості використання методів і алгоритмів узагальнення знань в інтелектуальних системах підтримки прийняття рішень реального часу;
- розширення понятійного апарату: введення понять, які враховують динамічну природу об'єктів узагальнення; формалізація завдання узагальнення для роботи з динамічними даними;
- розробка методів і алгоритмів узагальнення знань для динамічних об'єктів узагальнення;
- проектування і розробка програмного комплексу, що реалізує розглянуті в роботі методи і алгоритми.

1 МОДЕЛІ І МЕТОДИ ОБРОБКИ ТА АНАЛІЗУ ДАНИХ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

Розвиток сучасних складних інформаційних систем тісно пов'язаний з розвитком найбільш досконалих їх представників, до яких відносяться інтелектуальні системи. Інтелектуальна система (ІС) [1] може бути розглянута як комп'ютерна система для вирішення класів задач, що традиційно вважаються творчими, що належать конкретній предметній області, знання про яку зберігаються в пам'яті такої системи і які або не можуть бути вирішені людиною в реальному часі, або ж їх рішення вимагає автоматизованої підтримки. Рішення, що надається інтелектуальною системою, має давати результати, зіставні з рішеннями, прийнятими людиною-фахівцем в деякій області. Характеризація комп'ютерної системи як інтелектуальної буде неповною, якщо не будуть уточнені як природа вирішуваних завдань, так і засоби їх вирішення, які реалізуються завдяки певній архітектурі комп'ютерної системи [2].

Найважливішим класом задач, рішення яких вимагає інтелектуальної підтримки комп'ютерних систем [3], є завдання управління складними технічними об'єктами. Головною рисою подібних об'єктів управління слід визнати те, що вони є динамічними, мають здатність до розвитку, стан таких об'єктів і систем може змінюватися з часом.

Поява і розвиток засобів управління об'єктами, які належать до категорії динамічних, тісно пов'язане з розвитком інтелектуальних систем підтримки прийняття рішень (ІСППР). В даний час інтелектуальні системи підтримки прийняття рішень працюють з все більш складними технічними об'єктами і системами. В основі інтелектуальних систем даного типу лежить інтеграція моделей уявлення і маніпулювання знаннями. Моделі повинні бути орієнтовані на специфіку предметної області і мати розвинені засоби представлення знань про події, факти, дії, процеси, що відбуваються на складному технічному об'єкті.

На рисунку 1.1 представлена базова структура ІСППР [4], що включає такі підсистеми, як база даних і база знань, база моделей, блок пошуку рішень, блок аналізу ситуації, засоби інтелектуального інтерфейсу.

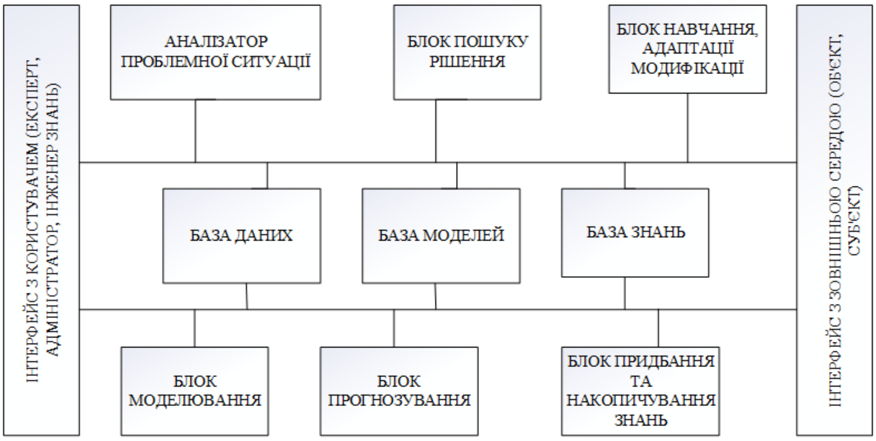


Рисунок 1.1 – Базова структура ІСППР

Завдання індуктивного формування понять повинні відображати такі аспекти природного інтелекту, як здатності узагальнення – впорядкування даних і знань з виділенням істотних параметрів в даних відповідно до поставленої мети.

При вирішенні задачі індуктивного формування понять (завдання узагальнення інформації) в рамках ІСППР необхідно [5]:

- визначити методи представлення знань для вирішення задачі узагальнення;

- вибрати спосіб подачі отриманого узагальненого опису (визначається, перш за все, тим, для яких цілей буде використовуватися отриманий опис);

- вибрати методи і алгоритми узагальнення, універсальні або предметно-орієнтовані, тобто призначені для конкретної предметної області.

Перші два завдання пов'язані з використовуваним в ІСППР методом представлення знань. Розглянемо основні методи і моделі представлення знань в інтелектуальних системах.

1.1 Методи представлення знань в інтелектуальних системах

Існує цілий ряд різних методів представлення знань [6]: до них відносяться продукційні моделі, семантичні мережі, схеми, фрейми, сценарії, штучні нейронні мережі, логічні моделі, дерева рішень та ін.

У системах, заснованих на продукційних правилах, знання представлені у формі множина правил, котрі вказують, які висновки повинні бути зроблені або не зроблені в різних ситуаціях [7].

Інтелектуальна система, яка використовує продукційну модель подання знань, включає в себе базу знань, що зберігає як множина правил виду «Якщо <умова> ТО <висновок>», так і множина фактів, дійсних в даний момент. Інтерпретатор управляє тим, яке правило має бути вибрано для виконання в залежності від наявності справжніх фактів в робочій пам'яті. Система, заснована на правилах, складається з правил IF-THEN, фактів і інтерпретатора, який керує тим, яке правило має бути викликано в залежності від наявності фактів в робочій пам'яті.

В експертних системах часто використовуються такі правила, в яких посилкою є опис ситуації, а висновком – дії, які необхідно зробити в даній ситуації.

Якщо інтелектуальна система призначена для вирішення завдання узагальнення, продукційні правила в якості посилок можуть використовувати умови, яким задовольняє опис даного об'єкту, а висновком повинен стати висновок про належність об'єкта до певного класу.

Широке застосування продукційних моделей при розробці інтелектуальних систем обумовлено наступними причинами [8]:

- модульна організація: окремі продукційні правила можуть бути незалежно додані в базу знань, виключені або змінені, при цьому не потрібно перепрограмування всієї системи. Таким чином, продукційна модель є відкритою моделлю представлення знань. Як наслідок цього, уявлення великих обсягів знань не викликає труднощів;

- наявність засобів пояснення: в продукційній моделі легко вбудовуються засоби пояснення, що дозволяють відстежити, запуск яких правил і в якому порядку було здійснено, таким чином, завжди можна встановити хід міркувань, які привели до певного висновку;
- наявність аналогії з пізнавальним процесом людини: відповідно до гіпотези Ньюелла-Саймона [9] продукційні правила, мабуть, є природний спосіб моделювання процесу вирішення завдань людиною; крім того, продукційні правила легкі для сприйняття людиною;
- за допомогою продукційних правил виражаються як декларативні, так і процедурні знання.

Наступною широко відомою моделлю представлення знань є семантичні мережі. Семантичні мережі [10-11] – класичний спосіб представлення інформації, що використовується в штучному інтелекті. З точки зору математики семантична мережа являє собою позначений орієнтований граф, вершини якого означають деякі сутності (об'єкти, події, процеси, явища, ситуації), а дуги – відносини між сутностями, які вони пов'язують. Відносини мають для семантичних мереж виключно важливе значення, оскільки представляють базову структуру для організації знань: якщо задані відносини, то знання являють собою зв'язну структуру, дослідження якої дозволяє виводити логічним шляхом інші знання. Головною перевагою семантичних мереж є те, що вся інформація, пов'язана з деяким об'єктом, легко може бути отримана зі зв'язків цього об'єкта.

При використанні семантичних мереж в системах узагальнення і вилучення знань виникає проблема пошуку загальних закономірностей в описах об'єктів або ситуацій в разі, коли кожен окремий приклад об'єкта (ситуації) представляється окремою семантичною мережею. Операції над такими прикладами зводяться до операцій над графами, наприклад, до пошуку найбільшого підграфу, загального для всіх прикладів заданого класу. Недоліком даної моделі є складність роботи з неоднорідними графовими структурами і пов'язаний з цим великий перебір.

Однією з різновидів мережевих моделей, яка широко використовується в багатьох системах штучного інтелекту, є модель на основі фреймів [12]. Кожен окремий фрейм є складно організованою структурою і являє собою сценарій, який описує типову ситуацію, пов'язану, наприклад, з будь-яким видом діяльності. Фрейми зазвичай утворюють мережеві структури і ієрархії, пов'язані взаємними посиланнями. Фрейми широко використовуються для вирішення таких завдань, як розуміння зорових образів, аналіз текстів на природних мовах і в ряді інших областей.

Для фрейма характерно уявлення взаємопов'язаних знань по конкретній темі або ситуації у вигляді набору слотів, що характеризують окремі риси, властивості, особливості ситуації, при цьому значення слотів в більшій частині задаються за умовчанням. Типове, або «скелетне» описання ситуації може поповнюватися і змінюватися за рахунок значень, що надходять з інших фреймів, при цьому типові значення слотів уточнюються і конкретизуються. З точки зору узагальнення інформації легко помітити, що фрейми надають зручну структуру для опису об'єктів, типових для якоїсь конкретної ситуації, зокрема, стереотипів об'єктів. Недоліком такої моделі є насамперед її спрямованість на вирішення завдання конкретизації опису випадку, об'єкта, ситуації, а не на отримання нових узагальнених понять шляхом генерації нових фреймів.

Когнітивні карти [13] відносяться до того ж класу систем уявлення знань, що і фрейми. Когнітивні карти можуть бути корисним інструментом для формування і уточнення гіпотези про функціонування досліджуваного об'єкта, що розглядається як складна система [14]. Для того щоб зрозуміти і проаналізувати поведінку складної системи, доцільно побудувати структурну схему причинно-наслідкових зв'язків. Когнітивну карту можна розуміти як схематичне, спрощене описання картини світу індивіда, точніше її фрагмента, що відноситься до даної проблемної ситуації. Психологи останнім часом використовують цей термін у вузькому сенсі, тільки для опису просторових відносин. Звісно ж, що термін «Когнітивна карта» значно тісніше пов'язана з загальноприйнятим розумінням картини світу, ніж введені лінгвістами поняття «фрейм» і «скрипт».

Штучна нейронна мережа – математична модель, побудована за принципом організації та функціонування біологічних нейронних мереж. Мереж нервових клітин живого організму. Основою нейронних мереж є штучний нейрон, який має наступну структуру [14]:

- вхідні сигнали x_i: дані, що надходять з навколишнього середовища або від інших активних нейронів. Діапазон вхідних значень для різних моделей може відрізнятися. Зазвичай вхідні значення бувають дискретними (бінарними) і визначаються множиною {0, 1} або {-1, 1}, або приймають будь-які вагомі значення;
- набір речових вагових коефіцієнтів ω_i: вагові коефіцієнти визначають силу зв'язку між нейронами;
- рівень активації нейрона $\sum \omega_i * \chi_i$, який визначається зваженою сумою його вхідних сигналів x_i;
- порогова функція f, призначена для обчислення вихідного значення нейрона шляхом порівняння рівня активації з деяким порогом.

Порогова функція визначає активний або неактивний стан нейрона.

Першим прикладом нейромереж моделі є нейрон Мак-Каллока-Піттса [15]. В даний час нейронні мережі застосовуються в множині завдань, серед яких найбільш важливими є наступні:

- класифікація;
- розпізнавання образів;
- реалізація пам'яті;
- прогнозування;
- оптимізація;
- фільтрація.

Особливістю нейронних мереж є те, що вони навчаються. Можливість навчання – одне з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних і / або «зашумлених», частково перекручених даних.

Важливою частиною будь-якої інтелектуальної системи є підсистема логічного виведення. Традиційно основою процесу формування міркувань є дедуктивний логічний висновок, заснований на отриманні висновку із засновків. Розвиток логічних моделей і їх реалізація на ЕОМ привели до створення логічного програмування і до розробки таких мов, заснованих на логіці, як PROLOG [16]. Класичні логічні моделі грають важливу роль в експертних системах, оскільки в таких системах необхідні кошти логічного висновку, що дозволяють проводити міркування від фактів до висновків.

Однак завдання, які вирішуються в інтелектуальних системах, часто є некоректними в тому сенсі, що вони вимагають застосування евристик і не припускають повноти знань [17], які є вихідними посилками при класичному логічному висновку. Це означає, що в рамках ІС потрібно відображати такі здібності до міркування [11], як здатності до навчання на основі позитивних і негативних прикладів, і, нарешті, здатності до адаптації у відповідності зі зміною множини фактів і знань. Такі завдання вимагають створення нових алгоритмічних і програмних систем для реалізації нетрадиційного виведення.

Реалізація індуктивних міркувань або міркувань за аналогією дозволяє отримати правдоподібні висновки. Однією з найбільш успішних моделей подання знань для індуктивного виводу є модель дерево рішень. Подання знань за допомогою дерев рішень з успіхом було використано в ряді систем навчання з учителем, наприклад, в алгоритмі ID3 Куінлана [17].

Теорія дерево рішень базується на інформаційних оцінках, дерева рішень використовуються при вирішенні класифікаційних завдань і являють собою процедуру визначення класу для пред'явленого прикладу. Кожен вузол дерева визначає або ім'я класу, або специфічну перевірку, що розділяє простір прикладів, приписаних вузлу, відповідно до можливих результатів перевірки. Кожна підмножина прикладів, що виникла в результаті такого поділу, відповідає класифікації під проблеми для простору прикладів, яке отримано на піддереві. Дерево рішень можна представити як стратегію «дроблення і просування вперед» для об'єкта, що підлягає класифікації. Формально можна визначити дерево рішень як граф, що не містить циклів, в якому кожна вершина – це або кінцевий вузол, зважений ім'ям класу, або проміжний вузол, що містить перевірку значень атрибута з подальшим розщепленням на піддерева для кожного допустимого значення атрибута.

Наприклад, алгоритм ID3 [18] будує дерево рішень на основі множини прикладів, для яких відомий результат класифікації, починаючи з кореневого вузла (вершина дерева) вниз до кінцевих вузлів (листя). На кожному етапі побудови інформаційний зв'язок між класифікаційними і досліджуваними атрибутами використовується для вибору атрибута, на підставі якого відбувається розгалуження в даній точці.

Інформаційний зв'язок між класифікаційними атрибутом і досліджуваним атрибутом називається також приростом інформативності [19], і визначається на основі частоти появи значень ознак атрибута в тестовій множині прикладів.

Дерево рішень можна розглядати як особливу форму тесту, який приписує певні перевірки на кожному кроці аналізу.

Кожен шлях від кореня дерева до кінцевої вершини (листу) відповідає кон'юнкції перевірок умов на значення атрибутів, а дерево в цілому являє диз'юнкцію таких кон'юнкцій. Точніше, вирішальні дерева класифікують приклади шляхом сортування їх за допомогою дерева від кореневого вузла до одного з кінцевих вузлів (листя), в яких виконується класифікація прикладу. Кожен вузол в дереві рішень визначає перевірку значення деякого атрибута прикладу, а кожне розгалуження, що виходить з цього вузла, відповідає одному з можливих значень цього атрибута.

Класифікація прикладу починається з кореня дерева рішень, де виконується перевірка атрибута, приписаного даному вузлу (тест для даного атрибута), потім, вибирається шлях для руху вниз по одній з гілок дерева відповідно до значення атрибута. Процес повторюється в вузлі, яким закінчується обрана гілка, і так далі, до тих пір, поки не буде досягнутий кінцевий вузол (лист). Кінцевому вузлу приписаний один з можливих відповідей (рішення).

З різних видів узагальнення для цілей систем підтримки прийняття рішень [16] реального часу (СППР РЧ) найбільш придатний варіант узагальнення на основі простору ознак опису як самих об'єктів, так і ситуацій, що виникають на складному технічному об'єкті [17].

З розглянутих вище моделей подання знань в подальшому пропонується використовувати такі моделі, як дерево рішень, і продукційні моделі: їх основними рисами є універсальність, простота реалізації і зручність перетворення дерева рішень в продукційні правила.

1.2 Проблема узагальнення понять

У системах, що моделюють мислення, узагальнення розуміють як процес отримання знань, що пояснюють наявні факти, та здатних пояснювати, класифікувати або передбачати нові [18]. У загальному вигляді задача узагальнення була сформульована Михальським [19] наступним чином: за сукупністю спостережень (фактів) F, сукупності вимог і припущень до виду результуючої гіпотези H, і сукупності базових знань і припущень, що включають знання про особливості предметної області, обраному способі представлення знань , допустимих операторів, евристик, сформувати гіпотезу H: H $=> F$ (H «пояснює» F).

Форма подання і загальний вигляд гіпотези H, а також обрані моделі узагальнення залежать від мети узагальнення і обраного способу представлення знань. Згідно Михальським [17], можна виділити моделі узагальнення по вибірках і моделі узагальнення за даними. У першому випадку сукупність фактів F має вигляд навчальної вибірки – множина об'єктів, кожен з яких зіставляється з ім'ям деякого класу. Метою узагальнення в цьому випадку може бути:

- формування понять, тобто побудова за даними навчальної вибірки для кожного класу максимальної сукупності його загальних характеристик;
- класифікація, або побудова за даними навчальної вибірки мінімальної сукупності характеристик, яка відрізняла б елементи одного класу від елементів інших класів;
- визначення закономірності послідовної появи подій.

До моделей узагальнення по вибірках відносяться лінгвістичні моделі, методи автоматичного синтезу алгоритмів і програм за прикладами. У моделях узагальнення за даними апріорний поділ фактів по класах відсутній. Тут можуть ставитися такі цілі:

- отримання гіпотези, узагальнюючої дані факти;
- виділення образів на множину спостережуваних даних, угруповання даних за ознаками;
- встановлення закономірностей, що характеризують сукупність спостережуваних даних.

За способом представлення знань і припущень на загальний вигляд об'єктів, які увійшли в навчальну вибірку, методи узагальнення діляться на методи узагальнення за ознаками і структурно-логічні (концептуальні) методи. У першому випадку об'єкт навчальної вибірки представляються у вигляді сукупності значень непрямих ознак. Методи узагальнення і розпізнавання розрізняються для якісних і кількісних ознак. Правило виводу гіпотези H з фактів F називають індуктивним, якщо з істинності H слід істинність F, а зворотне невірно.

Головною особливістю структурно-логічних методів, на відміну від признакових методів, є використання в навчальних вибірках об'єктів, що мають внутрішню логічну структуру. Такими об'єктами можуть бути послідовності подій, ієрархічно організовані мережі, алгоритмічні та програмні схеми.

1.3 Завдання узагальнення понять за ознаками

Перш за все, з усіх можливих завдань, пов'язаних з побудовою індуктивних залежностей, виділимо коло завдань, які називаються завданнями індуктивного формування понять, це завдання, які моделюють можливість людини давати опис, що охоплюють множину прикладів деякого поняття. В основі процесу індуктивного формування понять лежить вміння людини виділяти деякі найбільш загальні або характерні фрагменти описів серед описів окремих прикладів поняття, позбавляючись від дрібних, незначних характеристик, притаманних конкретних прикладів поняття. Назвемо таку задачу завданням узагальнення.

Під узагальненням, як правило, розуміється перехід від розгляду одиничного об'єкта 0 або деякої множини об'єктів O до розгляду узагальненого поняття D, яке відображає характерні для цієї множини відносини між значеннями ознак і є достатнім для поділу об'єктів, що належать множині, і об'єктів, які не належать йому , за допомогою деякого правила розпізнавання [18].

Процес узагальнення тісно пов'язаний з поняттям машинного навчання. На основі обробки експертної інформації [16] формується база знань СППР, в якій зберігається модель функціонування системи. За допомогою системи узагальнення інформації про характеристики конфліктних ситуацій обробляється спеціальним чином і вводиться в базу знань. Множину ознак, що характеризують факт виникнення різних ситуацій, формується на основі інформації, що циркулює в системі управління. Така інформація часто зберігається у вигляді таблиць в базах даних, причому поля в таких таблицях зберігають поточні значення ознак [13]. Значення ознак зазвичай цілком визначені і достовірні, на підставі аналізу цих значень необхідно автоматично виконати узагальнення з метою розрізняння типових і нестандартних ситуацій, і видавати повідомлення про факт виникнення нестандартної або конфліктної ситуації. Описи різних ситуацій формуються на основі аналізу цілей і завдань функціонування системи і експертної інформації. Для кожної типової ситуації необхідно отримати узагальнений опис [18] у вигляді моделей (гіпотез) з можливістю їх подальшої перевірки [16].

Нехай $O = \{o_1, o_2, \ldots, o_n\}$ – множина об'єктів, яка може бути представлена в інтелектуальній системі S. Кожен об'єкт характеризується q ознаками. Позначимо через $X_1, X_2 .., X_q$ множина допустимих ознак, де $X_k = \{x_{k_1}, x_{k_2}, \ldots, x_{k_m}\}$ $(1 \leq k \leq q)$ і x_{k_i} є значеннями ознак.

Кожен об'єкт $o_i \in O, 1 \leq i \leq n$, представляється як упорядкованість множинаі значень ознак, тобто $o_i = < x_1, x_2, .., x_j, .., x_q >$, де $x_j \in X_j, 1 \leq j \leq q$. Такий опис об'єкта називається ознаковим описом. В якості ознак об'єктів можуть використовуватися кількісні, якісні або шкаліровані ознаки.

В основі процесу узагальнення лежить порівняння описів вихідних об'єктів, заданих сукупністю значень ознак, і виділення найбільш характерних фрагментів цих описів. Залежно від того, входить чи не входить об'єкт в обсяг деякого поняття, назвемо його позитивним або негативним об'єктом для цього поняття.

Нехай O – множина всіх об'єктів, які можуть бути представлені в деякій системі знань, V – множина позитивних об'єктів і W – множина негативних об'єктів. Будемо розглядати випадок, коли позитивні і негативні об'єкти утворюють розбиття множини O, тобто $O = V \cup W, V \cap W = \emptyset$, при цьому множина негативних об'єктів також розбита на підмножини – в ці підмножини входять приклади, що відносяться до різних класів, відмінним від класу об'єктів з V: $W = \cup W_i$ і $W_i \cap W_j = \emptyset, i \neq j$. Нехай K – непорожня множина об'єктів, така, що $K = K^+ \cup K^-$, де $K^+ \subset V$, $K^- \subset W$. Будемо називати K навчальною вибіркою. На підставі навчальної вибірки треба побудувати правило, що розділяє позитивні і негативні об'єкти навчальної вибірки.

Таким чином, поняття сформовано, якщо вдалося побудувати вирішальне правило, яке для кожного прикладу з навчальної вибірки вказує, належить цей елемент поняття чи ні. Алгоритми формують рішення у вигляді набору правил «ЯКЩО <умова> ТО <шукане поняття>». Умова представляється у вигляді логічної функції, в якій булеві змінні, що відображають значення ознак, з'єднані логічними операціями кон'юнкції, диз'юнкції, заперечення. Вирішальне правило вважається коректним, якщо воно в подальшому успішно розпізнає об'єкти, які не ввійшли спочатку в навчальну вибірку.

У приведеній задачі узагальнення важливою проблемою є опис об'єкта $o \in O$. Традиційно у признаковому описі об'єкта розглядається як набір значень ознак $< X_1, X_2, .., X_q >$. Однак сучасні СППР мають справу з об'єктами, які вимагають більш складних засобів опису, ніж така модель, яка не має можливості представити, наприклад, динаміку поведінки складної системи. У зв'язку з цим для опису об'єкта 0 потрібно використовувати більш складний апарат, що дозволяє в описі об'єкта врахувати фактор часу і зміну значень ознак з плином часу.

Так як стан складних технічних об'єктів або систем, з якими доводиться працювати ІСППР (РЧ), змінюється з часом, необхідні використання і розробка методів, неявно або явно враховують фактор часу.

У зв'язку з цим постало завдання інтелектуального аналізу темпоральних даних [17]. У більшості випадків вкрай важко або зовсім неможливо використовувати існуючі методи аналізу даних в таких предметних областях, де необхідно враховувати фактор часу, отже, виникає необхідність модифікації існуючих методів і розробки нових.

Виділяють 4 категорії даних, явним чи неявним чином містять час [15]:

- статичні дані – в таких даних немає і не може бути темпорального контексту, проте фактор часу можна врахувати за рахунок використання журналів реєстрації подій, логів і т. п.

- послідовності даних – впорядковані списки подій. Ця категорія включає впорядковані сукупності обставин, які помічені тимчасовими мітками. Більшість сукупностей зазвичай обмежені відносинами «до» і «після», ця категорія дозволяє ввести більшу кількість відносин, описаних в логіці Аллена [18] та інших;

- дані з тимчасовими мітками: помічені тимчасовими мітками послідовності статичних даних, зафіксовані через більш чи менш регулярні проміжки часу. Приклад цінові та метеорологічні дані, а в деяких випадках – біржові транзакції або мережеву активність;

- безпосередньо темпоральні дані: кожен кортеж в змінюваному в часі відношенні в базі даних може мати одну або кілька тимчасових розмірностей: час транзакції і / або час дії.

1.4 Динамічний об'єкт узагальнення

Розглянемо тепер проблему узагальнення при наявності темпоральних даних. Важливим завданням в таких системах є обробка даних, що залежать від часу. Зазвичай для контролю за станом складного об'єкта використовується набір датчиків, що відображають і, можливо, контролюючих, значення основних параметрів системи. Нехай в системі є q датчиків, показання яких знімаються в деякі дискретні моменти часу: $t = 0, 1, 2, 3, \ldots$.

Тоді показання наявних датчиків в певний момент часу i можна представити у вигляді вектору (формула1.1).

$$s_i =< x_1(t = i), x_2(t = i), \ldots, x_q(t = i), t = i > \qquad (1.1)$$

Очевидно, такий ознаковий опис об'єктів дозволяє лише поглянути на миттєвий «зліпок» стану системи. Для того, щоб простежити динаміку розвитку системи, зміни її стану, тенденції, очевидно, необхідно розглянути впорядковану множину таких векторів, отриманих на кінцевому тимчасовому інтервалі $(t_i, t_i + r - 1), r > 1$. Нехай розглядається q параметрів на часовому інтервалі довжини r. Такі дані зображено в наступному вигляді (таблиця 1.1).

Таблиця 1.1 – Динамічний об'єкт узагальнення

	Параметр$_1$	Параметр$_2$	…	Параметр$_q$	Час (t)
(s_i)	$x_1(t=i)$	$x_2(t=i)$	...	$x_q(t=i)$	i
(s_{i+1})	$x_1(t=i+1)$	$x_2(t=i+1)$	...	$x_q(t=i+1)$ i+1	i+1
(s_{i+2})	$x_1(t=i+2)$	$x_2(t=i+2)$	…	$x_q(t=i+2)$	i+2
...	...	...	...	...	…
(s_{i+r-1})	$x_1(t=i+r-1)$	$x_2(t=i+r-1)$	...	$x_q(t=i+r-1)$	$i+r-1$

Тоді кожна з рядків зазначеної матриці, позначена $(s_i), (s_{i+1}), .., (s_{i+r-1})$, являє собою зліпок стану даної системи на моменти часу відповідно $i, i+1, .., i+r-1$. Кожна клітинка матриці представляє собою значення відповідного параметру в певний момент часу (будемо далі називати ці величини спостереженнями). Кожен стовпець матриці, позначений Параметр$_1$, Параметр$_2$, .., Параметр$_q$, являє собою значення відповідного параметру, що змінюється за інтервал часу $t^* = r$. Назвемо структуру, представлену в таблиці 1.1, динамічним об'єктом узагальнення.

Сам динамічний об'єкт узагальнення можна розглядати, з одного боку, як сукупність статичних зліпків стану системи (рядки матриці), які, тим не менш, тісно пов'язані між собою, так як відображають зміну (динаміку) стану системи за певний інтервал часу, з іншого боку, так як параметри зазвичай є речовими, динамічний об'єкт узагальнення можна розглядати як набір часових рядів (стовпці матриці), які відповідають змінам значень кожного з розглянутих параметрів за інтервал часу $t^* = r$.

Також динамічний об'єкт узагальнення може розглядатися як опис однієї конкретної динамічної ситуації на складному технічному об'єкті, що розвивається за проміжок часу в $N = r$ тактів.

Еквівалентну уявлення для динамічного об'єкта, який буде потрібно надалі, представлено в таблиці 1.2 (транспонована матриця з подання до таблиці 1.1):

Таблиця 1.2 – Динамічний об'єкт узагальнення (еквівалентне уявлення)

	(s_{i})	$(s_{\mathrm{i+1}})$	$(s_{\mathrm{i+2}})$	...	$(s_{\mathrm{i}+r-1})$
Час (t)	i	$i + 1$	$i + 2$	…	$i + r – 1$
Параметр$_1$	$x_1(t = i)$	$x_1(t = i + 1)$	$x_1(t = i + 2)$	...	$x_1(t = i + r – 1)$
Параметр$_2$	$x_2(t = i)$	$x_1(t = i +1)$	$x_2(t = i + 2)$	…	$x_2(t = i + r – 1)$
…	...	...	...	...	...
Параметр$_{\mathrm{q}}$	$x_{\mathrm{q}}(t = i)$	$x_{\mathrm{q}}(t = i + 1)$	$x_{\mathrm{q}}(t = i + 2)$	...	$x_{\mathrm{q}}(t = i + r - 1)$

Наведемо тепер постановку задачі узагальнення для темпорального випадку.

Нехай $O = \{DynO_1, Dyn_{O2}, \dots, DynO_n\}$ – множина динамічних об'єктів узагальнення, які можуть бути представлені в інтелектуальній системі S. Нехай $\widehat{\mathrm{V}}$ – множина позитивних об'єктів і $\widehat{\mathrm{W}}$ – множина негативних об'єктів. Будемо розглядати випадок, коли позитивні і негативні об'єкти утворюють розбиття множини O, тобто $\hat{O} = \widehat{V} \cup \widehat{W}$, $\widehat{V} \cap \widehat{W} = \emptyset$, при цьому множина негативних об'єктів також розбите на підмножини – в ці підмножини входять приклади, що відносяться до різних класів, відмінним від класу об'єктів з $\widehat{V}: \widehat{W} = \cup \widehat{W}_\iota$ і $\widehat{W}_\iota \cap \widehat{W}_J = \emptyset$. Нехай K– непорожня множина об'єктів, таке, що $\widehat{K} = \widehat{K^+} \cup \widehat{K^-}$, де $\widehat{K^+} \subset \hat{V}$, $\widehat{K^-} \subset \widehat{W}$. Будемо називати K навчальною вибіркою. На підставі навчальної вибірки треба побудувати правило, те що розмежовує позитивні і негативні об'єкти навчальної вибірки.

Завдання узагальнення в такій постановці є набагато складнішою і вимагає розробки як нових способів подання вирішальних правил, так і нових алгоритмів отримання таких правил. У зв'язку з цим спочатку буде розглянуто найпростіший випадок: єдиний параметр, що розглядається на деякому часовому інтервалі. В цьому випадку динамічний об'єкт узагальнення вироджується в тимчасовій ряд. У другому розділі конкретизується постановка задачі узагальнення для такого випадку, який фактично може бути зведений до задачі виявлення аномалій в наборах часових рядів, наводиться огляд методів вирішення подібних завдань, пропонуються нові методи вирішення.

Після цього буде розглянуто і загальний випадок: в розділі 3 конкретизована постановка завдання для загального випадку, описаний апарат темпоральних дерев рішень, який спільно з деякими міркуваннями, отриманими в другому розділі, дозволяє вирішити задачу діагностики для динамічних об'єктів узагальнення в інтелектуальних системах підтримки прийняття рішень реального часу.

2 ЗАВДАННЯ УЗАГАЛЬНЕННЯ ДЛЯ ДИНАМІЧНИХ ОБ'ЄКТІВ. ОКРЕМИЙ ВИПАДОК

У цьому розділі розглянуто найбільш простий випадок завдання узагальнення для динамічних об'єктів. Нехай розглядається ситуація на кінцевому тимчасовому інтервалі довжини $t^* = r, r > 1$, динамічний об'єкт описаний єдиним атрибутом і представлений у вигляді таблиці 1.2. Тоді динамічний об'єкт фактично є тимчасовим рядом. Розглянемо тимчасові ряди більш докладно.

2.1 Часові ряди

Опис об'єкта у вигляді набору значень його властивостей використовується, в основному, для подання тих об'єктів, які з часом не змінюються. Для опису ж стану складної технічної системи потрібно спосіб, що дозволяє якимось чином враховувати фактор часу. Зазвичай для контролю за станом складної системи використовується набір датчиків, що відображають і контролюючі значення основних параметрів системи. Зміна значень датчиків з часом дозволяє відслідковувати зміну стану системи в цілому. Послідовність значень кожного з датчиків являє собою тимчасової ряд, який, будучи правильним чином проаналізовано, може багато що сказати про стан і зміну стану складного об'єкта. Саме з цих причин останнім часом приділяється велика увага інтелектуальному аналізу часових рядів [18], які використовуються не тільки в техніці, а й в економіці, медицині, банківській справі.

У цьому розділі розглянуто задачу узагальнення для динамічних об'єктів, поведінка яких характеризується зміною єдиного параметра (або атрибута). В цьому випадку динамічний об'єкт являє собою тимчасовий ряд.

Часовим рядом [19] називають послідовність спостережень, звичайно впорядковану за часом, хоча можливе впорядкування і по якомусь іншому параметру. Основною рисою, що виділяє аналіз часових рядів серед інших видів аналізу є істотність порядку, в якому виробляються спостереження. Якщо в багатьох задачах спостереження статистично незалежні, то у тимчасових рядах вони, як правило, залежні і характер цієї залежності може визначатися положенням спостережень в послідовності. Природа ряду і структура породжує ряд процесу можуть зумовлювати порядок утворення послідовності.

У загальному випадку тимчасовий ряд TS – це кінцева упорядкована послідовність значень $TS = < ts_1, ts_2, .., ts_r >$, описує протікання якого-небудь тривалого процесу, де $ts_i, 1 \le i \le r$ – деяке дійсне число, індекс i відповідає мітці часу. Час, як було введено в розділі, будемо вважати дискретним, які приймають цілочисельні значення 0, 1, 2 .... Значеннями ts_i можуть бути показання датчиків, ціни на який-небудь продукт, курс валюти і т. п. Приклад часового ряду наведено в таблиці 2.1 і на рисунку 2.1.

Таблиця 2.1 – Приклад часового ряду

Час	0	1	2	3	4	5	6	7	8	9
Значення	-1,07	0,13	0,85	0,96	0,81	0,84	-0,07	-1,01	-0,90	-1,14

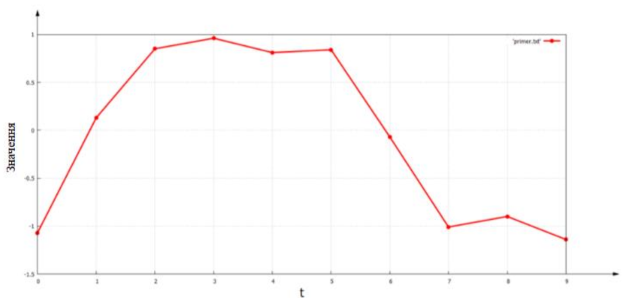


Рисунок 2.1 – Приклад часового ряду

Більшість алгоритмів узагальнення понять працює з дискретними даними, в той час як дані в тимчасових рядах є речові числа. Тому становить інтерес спосіб дискретизації часових рядів таким чином, щоб отримані дані могли використовуватися в алгоритмах узагальнення понять.

2.2 Завдання виявлення аномалій

Для завдання виявлення аномалій зазвичай є описання нормальної роботи системи – наприклад, набір станів системи, при яких неполадки відсутні. Описання ситуацій, відповідних неполадок на об'єкті, часто відсутня або неповна. При навчанні на таких даних потрібно побудувати модель нормальної роботи системи, яка в подальшому могла б передбачати, чи є поточна ситуація на об'єкті «нормальной» або «аномальной», тобто присутні в даний момент будь-які несправності чи ні.

Завдання визначення, чи виявлення, аномалій [20] ставиться як задача пошуку в наборах даних зразків, які не задовольняють деякому передбачуваному типовому поводженню. Можливість знайти аномалії в деякому наборі даних важлива в різних предметних областях – при аналізі роботи складних технічних систем (наприклад, телеметрії супутників), аналізі мережевого трафіку, в медицині (аналіз знімків магнітно-резонансної томографії, електрокардіограм), в банківській справі (аналіз транзакцій, вироблених за допомогою кредитних карт) і ін.

Аномалія, або «викид», визначається як елемент, який явно виділяється з набору даних до якого він належить, і істотно відрізняється від інших елементів вибірки. Неформально завдання визначення аномалій у наборах часових рядів ставиться таким чином. Нехай є колекція часових рядів, що описують деякі процеси. Ця колекція використовується для опису нормального протікання процесів. Потрібно на підставі наявних даних побудувати модель, яка є узагальненим описом нормальних процесів і дозволяє розрізняти нормальні і аномальні процеси.

Завдання ускладнюється, тим, що набір вихідних даних обмежений і не містить прикладів аномальних процесів, також не заданий критерій, за яким можна було б розрізняти «нормальні» і «аномальні» тимчасові ряди. У зв'язку з цим важко точно оцінити якість роботи алгоритму (відсоток правильно виявлених аномалій, число помилкових спрацьовувань і число пропущених аномалій). До того ж багато алгоритмів, добре показали себе на одних наборах даних, абсолютно не підходять для інших предметних областей. Також може відрізнятися і критерій, на підставі якого визначається «нормальність» рядів.

Важливим при вирішенні такої задачі є те, що виявлення аномалій дозволяє отримати інформацію, що вимагає вживання заходів: аномальний шаблон поведінки в комп'ютерній мережі, виявлений при аналізі трафіку, може говорити про те, що до комп'ютера був отриманий несанкціонований доступ і він може розсилати дані стороннім особам, аномалія на знімку, отриманому при проведенні магнітно-резонансной томографії, може свідчити про наявність злоякісної пухлини, аномалії виявлені при аналізі дій з кредитними картами, можуть бути показником того, що карта або персональні дані користувача були скомпрометовані, аномалії в показаннях датчика можуть свідчити про несправності пристрою.

Аномалії в наборах даних можуть бути викликані різними причинами (наприклад, діями зловмисників) але їх об'єднує те, що вони мають інтерес для експерта-аналітика.

Очевидний підхід до вирішення завдання виявлення аномалій наступний: необхідно визначити область, відповідну нормальної поведінки. Тоді будь-які спостереження, що лежить в цій області, будуть вважатися нормальним, а поза області – аномальним. Проте, навіть при такому простому підході виникає багато труднощів:

- визначити область, яка охоплює всі можливі варіанти нормальної поведінки непросто, крім того, межа між нормальними і аномальними спостереженнями дуже часто розмита: аномальне спостереження, що лежить близько до границі області, насправді може бути нормальним, і навпаки;

- в разі, якщо аномалії результат зловмисних дій, зловмисники зазвичай прагнуть замаскувати свої дії таким чином, щоб аномалії виглядали нормальною поведінкою, таким чином, роблячи завдання визначення області нормальної поведінки ще більш складною;
- у багатьох предметних областях область нормальної поведінки змінюється з часом – отже, поточний опис нормальної поведінки може стати недостатньо представницьким в майбутньому;
- точне визначення терміну «аномалія» різниться в залежності від обраної предметній області. Досить часто можна безпосередньо застосовувати методи, розроблені для конкретної предметної області, в інших предметних областях;
- великою проблемою є доступність даних, які використовуються для навчання і перевірки моделей, позначених як нормальні і аномальні;
- часто дані містять шум, за рахунок якого деякі спостереження стають схожими на аномальні, при цьому не будучи такими.

Ключове питання будь-якого методу виявлення аномалій – природа вихідних даних. Вхідними даними зазвичай є колекція примірників даних (що потенційно можуть називатися об'єктами, записами, крапками, шаблонами, зразками, подіями, спостереженнями, сутностями.). Кожен об'єкт може описуватися набором атрибутів (змінних, характеристик, полів). Атрибути можуть бути різних типів – бінарні, категорійні, безперервні. Кожен об'єкт може бути описаний одним атрибутом (одновимірний) або декількома (багатовимірний).

Застосування методів виявлення аномалій визначається характером атрибутів. Наприклад, для статистичних методів при роботі з безперервними і категорійними даними повинні використовуватися різні статистичні моделі. Аналогічно, для методів, заснованих на методі найближчих сусідів, характер атрибутів визначатиме метрику.

За рахунок описаних вище проблем завдання виявлення аномалій в загальній постановці є досить важкою. Насправді всі методи виявлення аномалій вирішують приватні задачі: на постановки задачі впливає природа вихідних даних, доступність / наявність даних, позначених як нормальні і аномальні, тип аномалій, які необхідно виявити, і т. п. Часто ці фактори визначаються предметною областю: використовується поняття з статистики, машинного навчання, інтелектуального аналізу даних, теорії інформації, теорії обробки сигналів – і застосовуються до певних постановок задачі.

У загальному випадку екземпляри даних, що використовуються в задачах виявлення аномалій, можуть бути пов'язані між собою: наприклад, послідовності, просторові дані, графові дані. У послідовності екземпляри даних лінійно впорядковані – це тимчасові ряди, геноми, протеїнові послідовності. В просторових даних кожен екземпляр даних пов'язаний з сусідніми – наприклад, дані про міський трафік, екології. Якщо в просторових даних є тимчасовий компонент, то говорять про просторово-часові данні. У графових даних екземпляри даних являють собою вершини в графі, які пов'язані з іншими вершинами дугами.

Важливим аспектом в методах виявлення аномалій є природа розглянутих аномалій. Аномалії можуть бути розділені на 3 категорії:

- точкові аномалії: якщо окремий об'єкт може вважатися аномалією по відношенню до решти набору даних, то він вважається точковою аномалією;
- контекстні аномалії: якщо об'єкт є аномалією в якомусь контексті (але не інакше), то він вважається контекстний, або умовний характер, аномалією;
- групові аномалії: якщо деякий набір об'єктів є аномалією по відношенню до всього набору даних, то він вважається аномалією, окремі елементи в цьому наборі самі по собі можуть і не бути аномаліями, але разом вони утворюють те, що називається колективною аномалією.

2.2.1 Навчальні вибірки для задачі виявлення аномалій

Зазвичай для даних вказано, чи належать вони до нормальних або аномальних, але при цьому отримати репрезентативну вибірку, яка буде достатньо точною і при цьому описуватиме всі можливі варіанти поведінки, надзвичайно важко. Часто об'єкти, представлені у вибірці, відносять до нормальних або аномальних експерт, при цьому слід зазначити, що отримати вибірку для нормальної поведінки об'єкта або системи простіше, ніж для аномальної, так як в деяких випадках аномальна поведінка зустрічається вкрай рідко і призводить до катастрофічних наслідків (наприклад, безпеку на транспорті). Більш того, аномальна поведінка динамічна за своєю природою, а значить, можуть з'являтися нові типи аномалій, які не були представлені в вихідній вибірці.

Залежно від наявності або відсутності міток даних виділяють три категорії методів виявлення аномалій:

- виявлення аномалій «з учителем» (методи керованого виявлення аномалій): для методів, що відносяться до даної категорії, потрібна наявність в навчальній вибірці об'єктів, що відносяться як до нормальних, так і до аномальних. На підставі таких даних будується модель, яка зможе визначати клас об'єктів, що надходять до неї на вхід;
- виявлення аномалій «без вчителя»: для даної категорії методів передбачається, що дані для навчання не потрібні. Але при цьому робиться припущення про те, що «нормальні» об'єкти зустрічаються набагато частіше, ніж «аномальні»;
- виявлення аномалій при частковому навчанні з «учителем» – щось середнє між першими двома: передбачається, що в навчальній вибірці є тільки приклади «нормальних» об'єктів.

2.2.2 Представлення результатів для методів виявлення аномалій

При використанні методів виявлення аномалій результати зазвичай представляються в наступному вигляді:

- коефіцієнти – метод відносить об'єкт до нормальних або аномальних з деяким ступенем впевненості, в кінцевому підсумку буде отримано список аномалій, ранжируваних за ступенем впевненості, після чого експерт може задати певний поріг, щоб відсікти зі списку об'єкти, швидше за все, не є аномаліями;

- мітки – метод відносить об'єкт до одного з нормальних або аномальних класів.

2.2.3 Області застосування методів виявлення аномалій

Методи виявлення аномалій використовуються в наступних областях:

- системи виявлення вторгнень [13]. Під вторгненням розуміються факти несанкціонованого доступу до комп'ютерної системи чи мережі або несанкціонованого управління ними (в основному, через Інтернет). Системи виявлення вторгнень – це програмні та / або апаратні засоби, які призначені для виявлення подібних фактів і використовуються для виявлення деяких типів шкідливої активності, яка може негативно вплинути на безпечність комп'ютерної системи;
- фрод (від англ. fraud – шахрайство, афера, підробка): вид шахрайства в області інформаційних технологій, зокрема, несанкціоновані дії і неправомірне користування ресурсами і послугами в мережах зв'язку,
- медицина і здоров'я: методи виявлення аномалій в даній області працюють з даними про пацієнтів, які зазвичай включають в себе зріст, вік, вагу, групу крові та інші дані, при цьому є велика кількість даних про нормальний стан пацієнтів, а отже, часто застосовуються методи часткового навчання з учителем;
- виявлення несправностей в складних технічних об'єктах, системах: промислові об'єкти, через їх постійну роботу, мають властивість зношуватися і приходити в непридатність, в зв'язку з цим необхідно вчасно виявляти виникаючі несправності;
- обробка зображень: дана предметна область включає в себе обробку знімків із супутників, розпізнавання образів, спектроскопію, аналіз

мамологічних знімків, відеоспостереження;

- виявлення аномалій в текстових даних: в даній предметній області основні завдання – виявлення нових тем, подій, історій у великих масивах документів або статей;

- сенсорні мережі: останнім часом все більшу увагу приділяють вивченню бездротових сенсорних мереж, це пов'язано з тим що дані, отримані з датчиків, що входять до мережі, мають ряд унікальних характеристик;

- інші області включають в себе більш специфічні завдання, такі як розпізнавання мови, аналіз поведінки робота, аналіз дорожнього трафіку, аналіз астрономічних даних, визначення помилок в веб-додатках..

2.2.4 Огляд і класифікація методів виявлення аномалій

Методи виявлення аномалій поділяються на наступні широкі категорії. Способи виявлення аномалій, засновані на методі найближчого сусіда, використовують наступне припущення: нормальні екземпляри об'єктів розташовані в тісному сусідстві, в той час як аномалії знаходяться на значній відстані від своїх найближчих сусідів. Для використання даної категорії методів необхідно, щоб була задана метрика або функція, яка визначає відстань між об'єктами. Відстань або міра схожості можуть обчислюватися різними способами – наприклад, для безперервних атрибутів зазвичай використовується евклідова відстань [18], для дискретних атрибутів використовується коефіцієнт подібності або інші, більш складніші заходи відстані [19]; для даних з множиною атрибутів обчислюється відстань між кожними з них, а отримані результати якимось чином об'єднують.

Виділяють дві групи методів:

- використовують відстань до k-ого найближчого сусіда;
- використовують відносну щільність для кожного об'єкта.

До переваг таких методів відносять те, що дані не повинні бути спочатку віднесені експертом до будь-яких класів – це методи, керовані даними, і ніяких апріорних припущень про природу і властивості даних не робиться. Проте, навіть невелика участь експерта в навчанні дозволяє підвищити якість визначення аномалій.

До недоліків подібних методів відносять незадовільну їх роботу в разі, коли у нормальних об'єктів занадто мало сусідів або навпаки – у аномальних примірників сусідів занадто багато, що призводить до великої кількості помилок першого і другого роду (відповідно помилкових спрацьовувань і пропусків подій). Також способи, засновані на методі найближчого сусіда, вимагають значної кількості обчислень, так як при віднесенні об'єкта до нормальних або аномальних потрібно обчислити відстань до всіх об'єктів з навчальної вибірки.

Методи пошуку аномалій, засновані на кластеризації. Кластеризацію [20] використовують для розбиття заданої вибірки об'єктів на підмножини, які називаються кластерами, таким чином, щоб кожен кластер складався з схожих об'єктів. При використанні даного класу методів покладаються на одне з наступних припущень:

- нормальні об'єкти належать кластеру, аномальні – ні;
- нормальні об'єкти лежать близько до центру кластера, аномальні – далеко від центру;
- нормальні об'єкти належать великим, щільним кластерам, в той час як аномалії належать невеликим і розрідженим.

Методи, засновані на кластеризації, оцінюють відстань до об'єктів на підставі інформації про кластер, до якого належать об'єкти, в той час як методика використання найближчих сусідів користується локальним оточенням кожного об'єкта.

Перевагами даного класу методів є можливість навчання без учителя, адаптація методів до різних типів даних і невелике число обчислень при віднесенні об'єктів до нормальних або аномальних (так як число кластерів зазвичай незначне). До недоліків слід віднести сильну залежність від обраного алгоритму кластеризації і специфіки його роботи (метод кластеризації не оптимізований для завдання виявлення аномалії – лише побічний результат від роботи алгоритму кластеризації і т. п.).

Основний принцип статистичних методів виявлення аномалій наступний: елементи вибірки розподілені по деякому закону, а аномалія – це спостереження, яке явно виділяється з набору даних, до якого воно належить, істотно відрізняється від інших елементів вибірки так як, швидше за все, було отримано по деякому іншому закону. Відповідно, і припущення, яким оперують статистичні методи виявлення аномалій, полягає в тому, що нормальні спостереження потрапляють в райони стохастичної моделі з високою ймовірністю, а аномалії – в райони з низькою ймовірністю. Статистичні методи застосовують статистичну модель до вихідних даних (зазвичай – визначеним нормальну поведінку) і користуються статистичним висновком для того щоб визначити, чи є спостереження аномалією чи ні. Ті спостереження, які мають низьку ймовірність бути отриманими в даній моделі, вважаються аномаліями. Теоретико-інформаційні методи виявлення аномалій аналізують кількість інформації в даних, використовуючи різні теоретико-інформаційні величини, такі як колмогоровську складність, ентропію. При цьому передбачається, що аномалії в даних призводять до нерівномірності інформаційного змісту набору даних.

Спектральні методи виявлення аномалій намагаються знайти наближення даних з використанням набору атрибутів таким чином, щоб відбити всю різноманітність даних. Спектральні методи використовують в припущенні, що дані можна представити в просторі меншої розмірності, причому в новому вигляді відмінність між нормальними і аномальними даними будуть значні. Отже, основне завдання – визначити такий простір, в якому можна було б легко виявити аномалії [16].

Методи пошуку аномалій, засновані на класифікації. Класифікація використовується для навчання моделі на даних, віднесених до різних класів (етап навчання), і віднесення екземплярів даних до одного з наявних класів з використанням отриманої моделі (етап іспиту). Методи виявлення аномалій, засновані на класифікації, припускають, що якщо класифікатор, може бути навчений в наявному просторі ознак, то він зможе розділити нормальні і аномальні об'єкти.

Серед методів виявлення аномалій, заснованих на класифікації, виділяють методи, які використовують:

- нейронні мережі;
- байєсовські мережі довіри;
- метод опорних векторів;
- продукційні правила.

У цьому завданню прийнято виділяти два випадки [18]: перший випадок – навчальна множина містить приклади єдиного класу, другий випадок – навчальна множина містить приклади декількох класів. У першому випадку важливий сам факт приналежності розглянутих об'єктів до класу з навчальної множини, тут потрібно якимось чином визначити «границю», відповідно до якої тимчасовий ряд належить класу з навчальної множини (не є аномалією) або не належить йому (є аномалією). У другому випадку додатково потрібно визначити приналежність об'єкта до конкретного класу.

До переваг методів виявлення аномалій, заснованих на класифікації, відноситься можливість використовувати множину способів і алгоритмів, розроблених в області машинного навчання – особливо для випадку, коли навчальна множина містить приклади декількох класів. Крім того етап «іспиту» проходить швидко в порівнянні з іншими класами методів, так як використовується попередньо побудована модель (класифікатор).

2.3 Використані в роботі набори даних

Моделювання процесу виявлення аномалій було проведено на даних з репозиторіїв UCR Time Series Data Mining Archive [20], UC Irvine Repository [21]. Також використовувалися дані, зібрані за допомогою спеціальних систем аналізу трафіку при передачі файлів з використанням різних протоколів (набір даних «трафік»).

Набір даних «трафік». «Трафік» – дані, отримані на основі аналізу трафіку при передачі файлів по протоколу ftp в різних умовах (в тому числі при одночасній передачі декількох файлів по декількох протоколах).

Для отримання даних був зібраний спеціальний стенд, на якому здійснювалася передача даних по мережі між двома комп'ютерами по різних протоколах в різних умовах. Фіксувалася лише довжина переданого пакета даних. Зворотня передача, навіть якщо і була, не фіксувалася.

Досліджувалися наступні варіанти передачі даних:

- передача по протоколу FTP (еталон);
- одночасна передача по протоколах FTP і ping (аналізувався FTP-трафік);
- одночасна передача по протоколах FTP і UDP (аналізувався FTP-трафік).

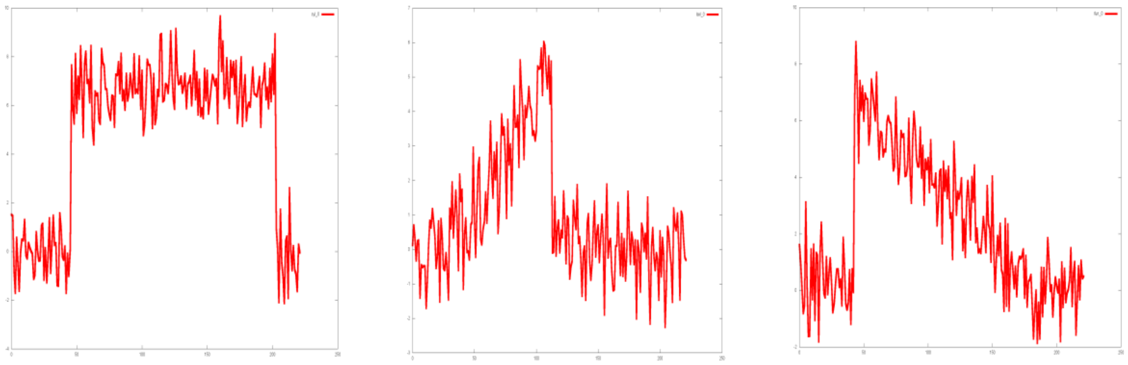
Маючи інформацію подібного виду про передачу даних по мережі, необхідно визначити, чи не є передача даних «підозрілою», що може свідчити про можливі компрометації мережевої інфраструктури, наявності програмних або апаратних закладок.

В якості тестових даних, крім інших, використовувалися спеціальним чином згенеровані тимчасові ряди, що імітують передачу даних.

2.3.1 Набори даних з UCR Time Series Data Mining Archive

Набір даних «циліндр-дзвін-воронка» («cylinder-bell-funnell», «CBF»), як випливає з назви, містить три різних класу часових рядів, умовно названих «циліндр», «дзвін», «воронка». Це відомий набір даних, широко застосовуваний для перевірки алгоритмів, що працюють з тимчасовими рядами [21].

Часові ряди, що відносяться до класу «циліндр», характеризуються наявністю на графіку плато, перед яким спостерігається різке зростання значення параметра, після – різкий спад. Класу «дзвін» відповідає поступове зростання значення від моменту часу, після чого спостерігається різке падіння значення. Для класу «воронка» характерний різкий стрибок значення, після якого спостерігається поступовий спад. Часові ряди – типові представники даних класів (рисунок 2.2).



а) «циліндр» б) «дзвін» в) «воронка»

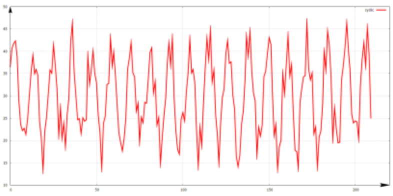
Рисунок 2.2 – Представники тимчасових рядів

Набір даних «контрольні карти »[22] («control chart»,«CC»,«synthetic control») – штучний набір даних, який містить шість різних класів, що описують тренди, які можуть бути присутніми в процесах: циклічність, зменшення значення, різке падіння, збільшення значення, постійна величина, різке зростання. Приклади рядів з даного набору наведені на рисунку 2.3.

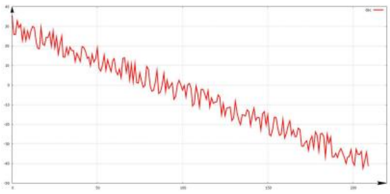
Для наведеного набору даних, як і випадку набору «циліндр-дзвін-воронка», параметр, що визначає хід процесу, є абстрактною величиною. Це дає можливість використовувати такі моделі для дуже широкого кола реальних завдань, де спостерігаються подібні тренди.

У наборі даних «wafer» [22] містяться тимчасові ряди, відповідні показаннями датчиків при виробництві напівпровідникових пластин.

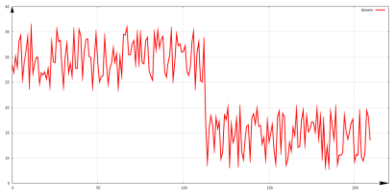
Напівпровідникова пластина (англ. wafer) – напівфабрикат в технологічному процесі виробництва напівпровідникових приладів і мікросхем. Являє собою тонку (250-1000 мкм) пластину з напівпровідникового матеріалу діаметром до 450 мм. Після створення необхідної напівпровідникової структури пластину розрізають на окремі кристали (чіпи).



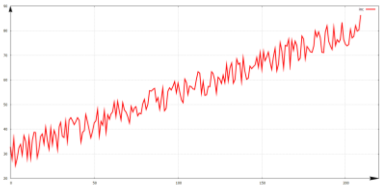
а) «циклічність»



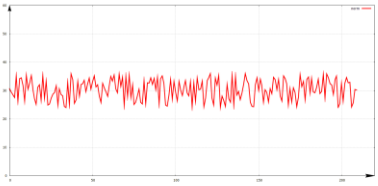
б) «зменшення значення»



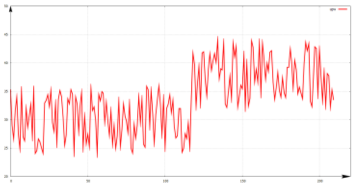
в) «різкий спад»



г) «збільшення значення»



д) «нормальне значення»



е) «різке зростання»

Рисунок 2.3 – Штучний набір даних, який містить шість різних класів, що описують тренди, які можуть бути присутніми в процесах

Виробництво таких пластин (травлення) – складний технологічний процес, що включає в себе більше 250 етапів обробки, на кожному з яких може статися погіршення характеристик або надійності, зменшення виходу продукту або навіть відбраковування, якщо параметри вийшли за необхідні межі. Найбільш критичними є 6 параметрів, серед яких експертами виділено 2, які за результатами експериментів показали найбільш точні результати по визначенню якісних і бракованих виробів: це 405 nanometer (nm) emission, 520 nanometer (nm) emission – інтенсивність випромінювання плазми з довжиною хвилі 405 нм і 520 нм під час виготовлення напівпровідникових пластин. На рисунку 2.4 і 2.5 представлені шість часових рядів, аналіз яких дозволяє розрізняти класи якісних і бракованих пластин.

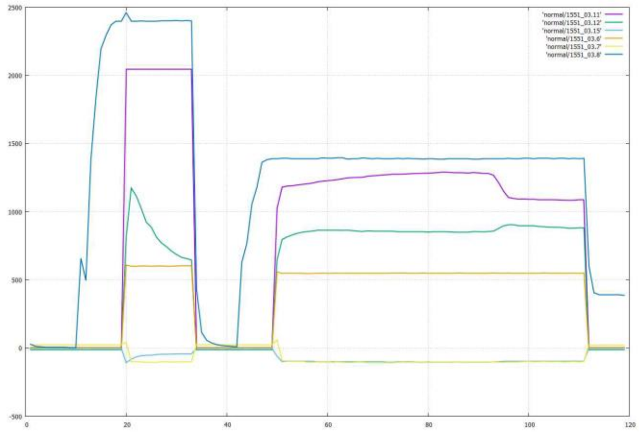


Рисунок 2.4 – «wafer» – нормальне протікання процесу

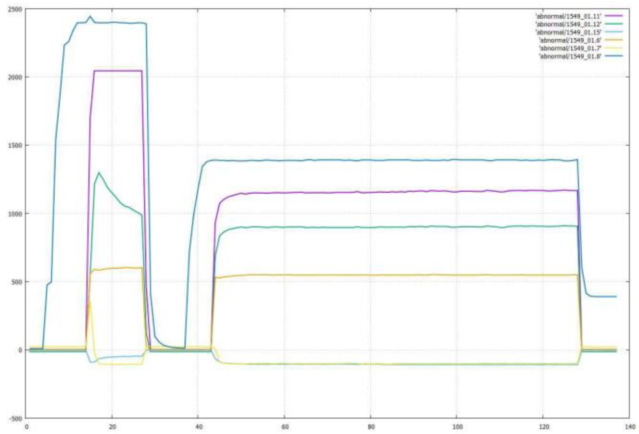


Рисунок 2.5 – «wafer» – ненормальне протікання процесу

У наборі даних «ECG». [86] (ЕКГ – електрокардіографія) містяться свідчення електричних сигналів кардіологічної активності, записаних з електродів, прикріплених в різних місцях. При запису електрокардіограми використовувалися два електроди, при цьому кожен часовий ряд співпадає із записом сигналу з одного електрода протягом одного серцевого скорочення.

Набори даних «Beef», «Coffee», «Olive oil» – спектрограми продуктів [19].

Спектрографи для продуктів використовуються в хемометриці для класифікації типів продуктів – завдання, що має практичне застосування при контролі якості і безпеки продуктів.

Спектрограми для трьох видів продуктів наведені на рисунку 2.6.

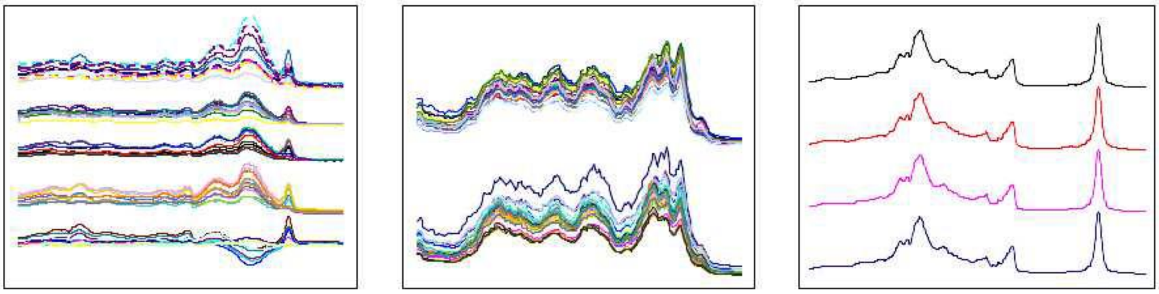


Рисунок 2.6 – Спектрограми: м'ясо – кава – оливкова олія

- Beef (м'ясо): набір даних містить спектрограми, відповідні різного ступеня змісту побічних продуктів в м'ясі.

- Coffee (кава): набір даних містить спектрограми, що відповідають двом класам (двом видам) кави: арабіка і робуста.

- Olive oil (оливкова олія): набір даних містить спектрограми оливкового масла екстракласу фільтрованої (extra virgin olive oil) з різних географічних регіонів.

Набір даних «Lightning 2» [90] аналогічний «Lightning 7», тільки всі блискавки розділені на два класи: наземні – включають в себе класи CG, IR, CP - і внутрішньохмарні I, I2, KM.

2.3.2 Набори даних з UC Irvine Repository

Набір даних «Activities of Daily Living Recognition with Wrist-worn Accelerometer Data Set».

Акселерометр – прилад, що вимірює проекцію удаваного прискорення (різниці між істинним прискоренням об'єкта та гравітаційним прискоренням).

Акселерометри реагують на прискорення або силу, що діє на сенсорний елемент датчика. Прискорення, статичне або динамічне, виникає під дією сили, прискорює датчик, наприклад, внаслідок дії гравітації.

Акселерометри можна використовувати в будь-якому пристрої, робота якого пов'язана з переміщенням, нахилом, вібрацією.

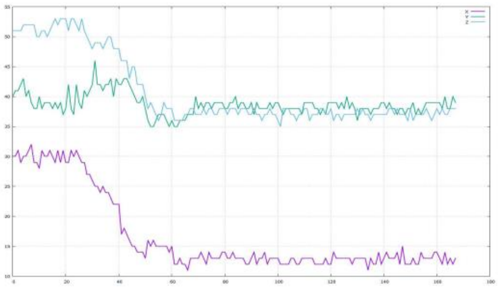
За конструктивним виконанням акселерометри поділяються на однокомпонентні, двокомпонентні, трьохкомпонентні. Відповідно, вони дозволяють вимірювати прискорення вздовж однієї, двох і трьох осей.

У наборі даних «Activities of Daily Living Recognition with Wrist-worn Акселерометр Data Set» (ADL, набір даних «повсякденна активність, записана за допомогою акселерометра») [65] представлені записи за допомогою акселерометрів виконання деяких простих дій, які позначені як «примітиви руху людини »(Human Motion Primitives, HMP), і пераховані нижче:

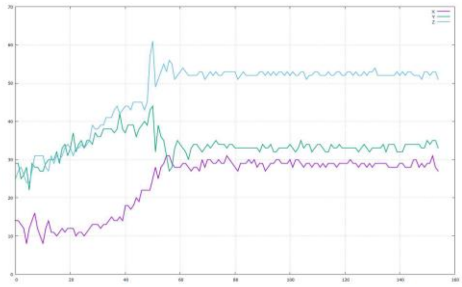
- чистити зуби;
- підніматися по сходинках;
- зачісуватися;
- спускатися по сходах;
- пити воду зі склянки;
- їсти м'ясо (з виделкою і ножем);
- їсти суп (ложкою);
- вставати з ліжка;
- лягати в ліжко;
- наливати воду;
- сідати на стілець;
- вставати зі стільця;
- телефонувати;
- ходити.

Прискорення кодується за такими правилами: [0; +63] = [-1.5g; + 1.5g]. Правило перетворення оцифрованого сигналу в реальне значення прискорення наступне: $real\_val = -1.5g + (coded\_val / 63) * 3g$.

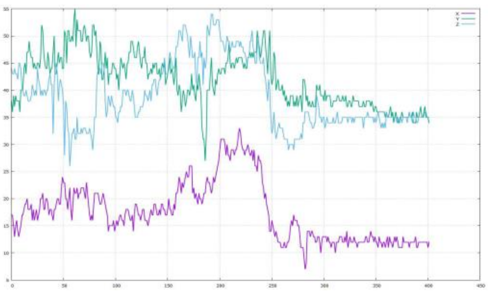
Показання акселерометра, відповідні прикладів деяких дій, наведені на рисунку 2.7.



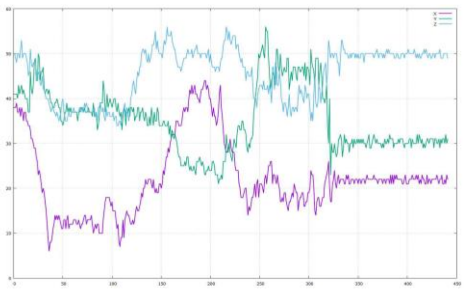
а) показанння акселерометра для дії «вставати зі стільця»



б) показанння акселерометра для дії «сідати на стілець»



в) показанння акселерометра для дії «вставати з ліжка»



г) показанння акселерометра для дії «лягати в ліжко»

Рисунок 2.7 – Показання акселерометра, відповідні прикладів деяких дій

2.4 Модель шуму в даних

При вивченні питань пошуку класифікуючих правил [18] неявно передбачалося, що такі правила існують. Зокрема, передбачалося, що існують детерміновані класифікуючи правила. Хоча таке припущення може бути вірним для штучно створених навчальних множин, використовуваних в машинному навчанні, воно напевно не виконується стосовно реальних баз даних. Використання бази даних в якості навчальної множини викликає такі труднощі. По-перше, інформація в базі даних обмежена, так що не вся інформація, необхідна для визначення класу об'єкта, доступна. По-друге, доступна інформація може бути пошкоджена або частково відсутня. Нарешті, великий розмір баз даних і їх зміна з часом народжує додаткові проблеми. Далі якщо база даних містить всю інформацію, необхідну для коректної класифікації об'єктів, деякі дані можуть не відповідати дійсності.

У роботах [5, 6, 13] проводилося дослідження впливу шуму на роботу алгоритмів узагальнення понять при наявності шуму у вхідних даних. При цьому одним з основних параметрів дослідження був рівень шуму – величина $p_0, 0 < p_0 < 0.5$, яка показує, що з імовірністю p_0 значення ознаки в навчальній або екзаменаційній множині спотворено. Також ця величина показує, що серед усіх N значень ознак в середньому $N * p_0$ значень ознак буде спотворено.

Для випадку ознакового опису об'єктів дана величина була досить інформативною для оцінки ступеня впливу шуму на наявні в розпорядженні дані. Однак коли об'єкти представлені часовими рядами або наборами часових рядів, така оцінка не застосовується.

2.4.1 Набір даних «циліндр-дзвін-воронка»

Крім наборів даних з самого UC Irvine Repository [15], тимчасові ряди для даного набору можна отримати штучно за такими формулами [91]:

1. «циліндр»: $c\ (t) = (6 + \zeta) * \chi_{[a,b]}\ (t) + \epsilon\ (t), 1 \le t \le M,$
2. «дзвін»: $b\ (t) = (6 + \zeta) * \chi_{[a,b]}\ (t) * \frac{(t-a)}{(b-a)} + \epsilon\ (t), 1 \le t \le M,$
3. «воронка»: $f\ (t) = (6 + \zeta) * \chi_{[a,b]}\ (t) * \frac{(b-t)}{(b-a)}\ \epsilon\ (t), 1 \le t \le M,$

де:

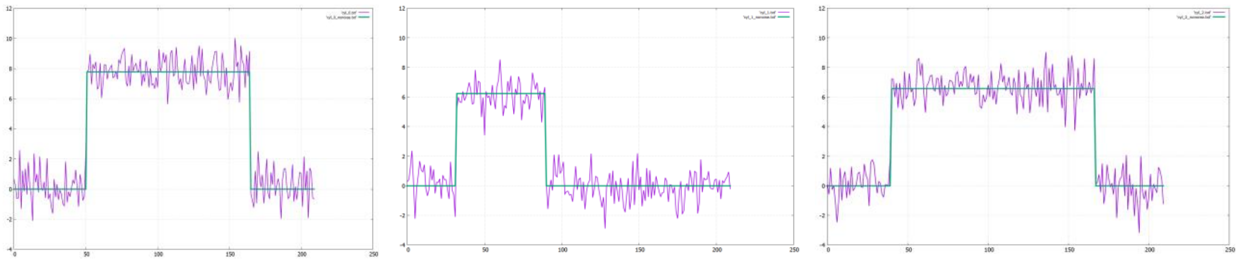
1. M – довжина часового ряду;
2. $\chi_{[a,b]} = \begin{cases} 0, t < a \\ 1, a \le t \le b \\ 0, t > b \end{cases}$
3. ζ – випадкова величина, що підкоряється стандартному нормальному розподілу $N\ (0,1)$;

1. (t) – випадкові величини, що підкоряються стандартному нормальному розподілу N (0,1);

2. a – випадкова величина, що підкоряється рівномірному розподілу на відрізку $[16, 32]$;

3. b – випадкова величина, що підкоряється рівномірному розподілу на відрізку $[32, 96]$.

На рисунку 2.8 наведені приклади штучно побудованих на підставі вищенаведених формул часових рядів класу «циліндр», по 2 на кожному графіку.



а) клас «циліндр» (1) б) клас «циліндр» (2) в) клас «циліндр» (3)

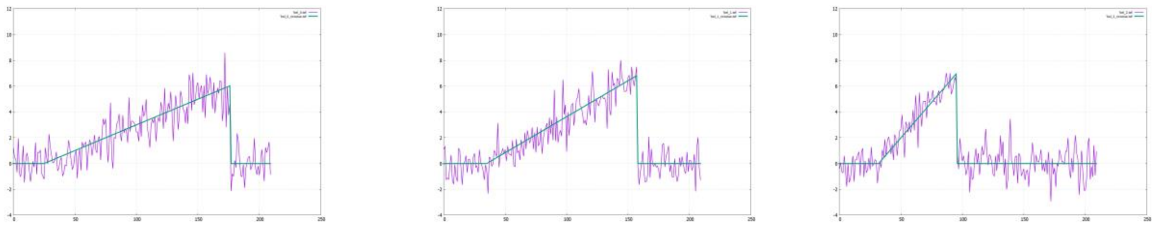
Рисунок 2.8 – Приклади штучно побудованих формул тимчасових рядів класу «циліндр»

На рисунку 2.9 для порівняння на одному графіку наведені окремо не зашумлені і зашумлені тимчасові ряди.

а) приклади тимчасових рядів класу «циліндр» без шуму б) приклади тимчасових рядів класу «циліндр» з шумом

Рисунок 2.9 – Приклади не зашумлених і зашумлених тимчасових рядів

Аналогічно наведені приклади для тимчасових рядів, що відносяться до класів «дзвін» (рисунках 2.10-2.11) і «воронка» (рисунках 2.12-2.13).



а) клас «дзвін» (1) б) клас «дзвін» (2) в)клас «дзвін» (3)

Рисунок 2.10 – Приклади штучно побудованих формул тимчасових рядів класу «дзвін»

а) приклади тимчасових рядів класу «дзвін» без шуму б) приклади тимчасових рядів класу «дзвін» з шумом

Рисунок 2.11 – Приклади штучно побудованих формул тимчасових рядів класу «дзвін»

При цьому кожен з графіків містить вихідний, чи не зашумлений, часовий ряд і відповідний йому тимчасовий ряд з внесеним в нього шумом. Графіки виконані в одному масштабі.Отримані таким чином набори часових рядів використовувалися для вивчення впливу шуму на роботу алгоритмів виявлення аномалій і класифікації. Для оцінки рівня шуму в даних слід звернути увагу на формули, за якими можна генерувати тимчасові ряди кожного класу. У формулах міститься доданок $\epsilon\ (t)$ – випадкова величина, що підкоряється стандартному нормальному розподілу N (0,1), що є адитивним гаусовським шумом.

2.4.2 Набір даних «контрольні карти»

Крім наборів даних з самого UC Irvine Repository [65], тимчасові ряди для даного набору можна отримати за наступними формулами [91]:

1. «нормальне значення»: $norm\ (t)\ =\ m\ +\ s\ *\ \epsilon(t), 1\ \leq\ t\ \leq\ M;$

2. «циклічність»: $cyclic\ (t)\ =\ m\ +\ a\ *\ sin\ (\pi\ *\frac{t}{T}\ ) +\ s\ *\ \epsilon\ (t), 1\ \leq\ t\ \leq\ M;$

3. «зменшення значення»: $dec\ (t)\ =\ m\ -\ g\ *\ t\ +\ s\ *\ \epsilon(t), 1\ \leq\ t\ \leq\ M;$

4. «збільшення значення»: $inc\ (t)\ =\ m\ +\ g\ *\ t\ +\ s\ *\ \epsilon\ (t), 1\ \leq\ t\ \leq\ M;$

5. «різкий спад»: $downw\ (t)\ =\ m\ -\ k\ (t)\ *\ x\ +\ s\ *\ \epsilon(t), 1\ \leq\ t\ \leq\ M;$

6. «різке зростання»: $\mathrm{u}pw\ (t)\ =\ m\ +\ k\ (t)\ *\ x\ +\ s\ *\ \epsilon\ (t), 1\ \leq\ t\ \leq\ M;$

де

1. M – довжина часового ряду;

2. m = 30, s = 2;

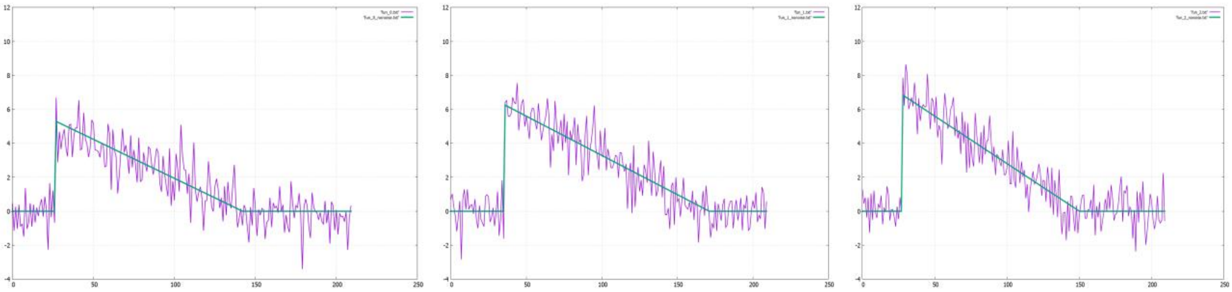
3. ϵ (t) – випадкова величина, що підкоряється рівномірному розподілу на відрізку [-3, 3];

4. a, T – випадкові величини, що підкоряються рівномірному розподілу на відрізку [10, 15];

5. g – випадкова величина, що підкоряється рівномірному розподілу на відрізку [0,2, 0.5];

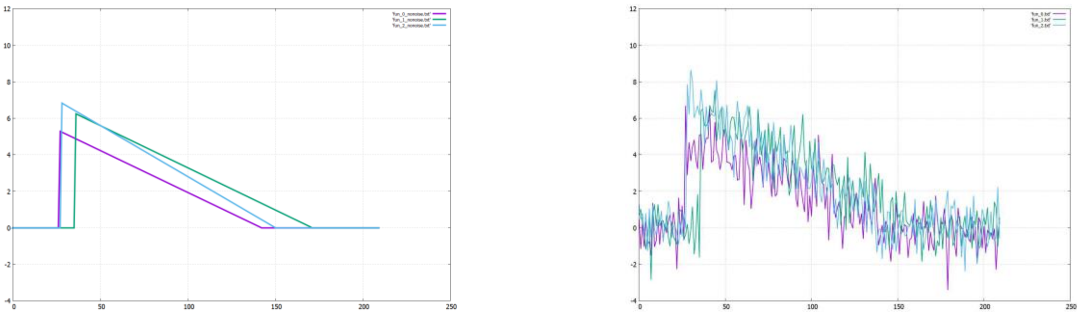
6. x – випадкова величина, що підкоряється рівномірному розподілу на відрізку [7.5, 20];

$k(t)\ = \begin{cases} 0, t\ < t3 \\ 0, t\ > t3 \end{cases}$, де $t3$ – випадкова величина, що підкоряється рівномірному розподілу на відрізку $\left[\frac{M}{3}, \frac{2*M}{3}\right]$.



а) клас « воронка » (1) б) клас « воронка » (2) в)клас « воронка » (3)

Рисунок 2.12 – Приклади штучно побудованих формул тимчасових рядів класу «воронка »



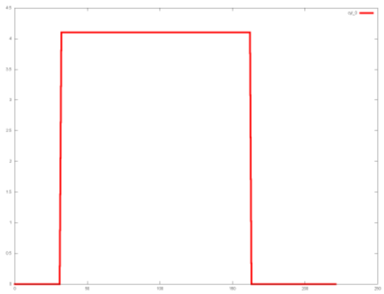
а) приклади тимчасових рядів класу «дзвін» без шуму б) приклади тимчасових рядів класу «дзвін» з шумом

Рисунок 2.13 – Приклади штучно побудованих формул тимчасових рядів класу «дзвін»

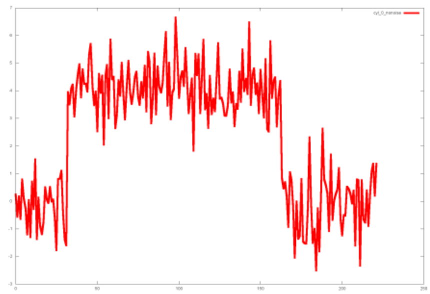
Отримані таким чином набори тимчасових рядів використовувалися для вивчення впливу шуму на роботу алгоритмів виявлення аномалій і класифікації. Для оцінки рівня шуму в даних слід звернути увагу на формули, за якими можна генерувати тимчасові ряди кожного класу. У формулах міститься доданок ε (t) – випадкова величина, що підкоряється стандартному нормальному розподілу N (0,1), що є адитивним гаусовським шумом.

2.5 Методи роботи з зашумленими даними

На рисунку 2.14 наведено приклад тимчасового ряду без шуму, і того ж тимчасового ряду з шумом. Обране уявлення для тимчасових рядів дозволяє успішно працювати з шумом в даних: за рахунок скорочення розмірності можна «згладити» крайні значення для тимчасового ряду, зберігши його форму і основні параметри.



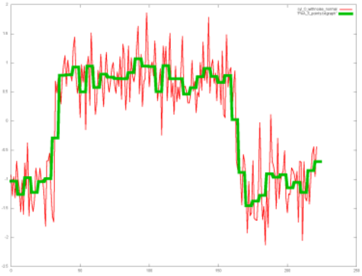
а) тимчасовой ряд без шуму



б) тимчасовой ряд з шумом

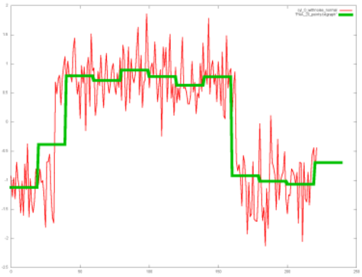
Рисунок 2.14 – Приклади не зашумлених і зашумлених тимчасових рядів

Розглянемо кілька нормалізованих уявлень вихідного часового ряду з різними параметрами. На рисунку 2.15 зображено нормалізовані представлення для тимчасового ряду з шумом. На рисунку 2.15, а) одна точка нормалізованого тимчасового ряду відповідає п'яти точкам вихідного часового ряду, на б) десять точок, на в) двадцять точок і на г) тридцять точок. Як видно, зі збільшенням числа точок вихідного тимчасового ряду, що відповідають одній точці нормалізованого ряду, тимчасовий ряд «згладжується» і нормалізований тимчасовий ряд стає все більше схожим на вихідний тимчасовий ряд без шуму (рисунок 2.14 а)).

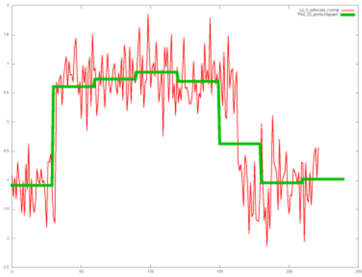


а) «стиснення» в 5 разів

б) «стиснення» в 10 разів



в) «стиснення» в 20 разів



г «стиснення» в 30 разів

Рисунок 2.15 – Приклад нормалізованого представлення для тимчасового ряду з шумом

2.6 Постановка завдання виявлення аномалій

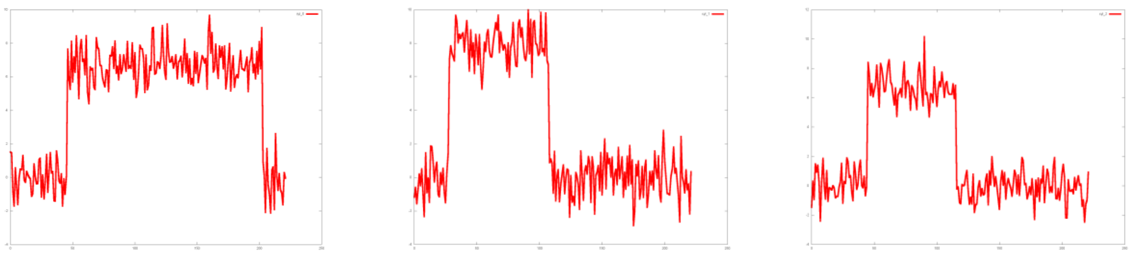
Завдання виявлення аномалій для набору тимчасових рядів ставиться таким чином. Нехай є набір об'єктів, де кожен об'єкт є часовий ряд: $TS\_STUDY = \{ts\_study_1, ts\_study_2, \ldots ts\_study_{m_1}$. Назвемо TS_STUDY навчальної вибіркою. Кожен з тимчасових рядів $ts\_study_i, 1 \leq i \leq m_1$ в навчальній вибірці є прикладом «нормального» протікання деякого процесу.

Множина $TS\_TEST = \{ts\_test_1, ts\_test_2, \ldots, ts\_test_{m2}\}$ назвемо екзаменаційною вибіркою. На підставі аналізу тимчасових рядів з TS_STUDY необхідно побудувати модель, що дозволяє відносити тимчасові ряди з екзаменаційної вибірки TS_TEST до «нормальних рядів» або до «аномалій» на підставі деякого критерію.

Припустимо, що дані ситуації описують нормальний перебіг процесів на складному технічному об'єкті і належать одному класу – «норма». На підставі цих ситуацій необхідно побудувати таку модель, яка описувала б «нормальне» протікання процесів і дозволяла б відносити ситуації, що виникають на об'єкті, до «нормальних» або «аномальних». В даному випадку перед нами завдання виявлення аномалій в наборах тимчасових рядів, коли в навчальній множині містяться приклади єдиного класу («норма»).

У загальному випадку для вирішення завдання визначення аномалій в наборах тимчасових рядів з одним класом поширені підходи, засновані на методі опорних векторів (і його модифікаціях) [22], нейронних мережах [23], використанні дискримінанту Фішера [24], продукційних правилах і ін.

Розглянемо це завдання на простому прикладі. Нехай навчальна вибірка TS_STUDY складається з трьох тимчасових рядів (рисунок 2.16). Екзаменаційна вибірка TS_TEST складається також з трьох тимчасових рядів (рисунок 2.17).



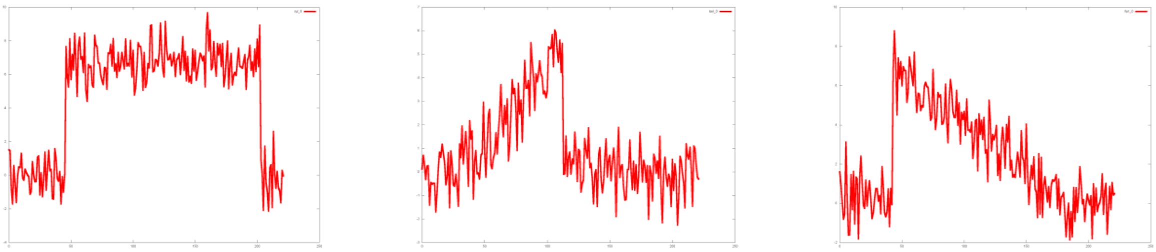
а) ряд 1 навчальна множина

б) ряд 2 навчальна множина

в) ряд 3 навчальна множина

Рисунок 2.16 –Навчальна вибірка TS_STUDY з трьох тимчасових рядів

Виходячи з наведеного вище постановка завдання виявлення аномалій, видно, що часові ряди на рисунку 2.17 б), в) з екзаменаційної множини значно відрізняються (в даному випадку – за формою) від тимчасових рядів з навчальної множини і, отже, будуть аномаліями для даної навчальної множини. При цьому можна припустити, що механізм, або закон, за яким були отримані тимчасові ряди, представлені на цих рисунках, відрізняються від механізму, за допомогою якого були отримані часові ряди з навчальної множини. Навпаки, часовий ряд на рисунку 2.17 а), з екзаменаційної множини не буде аномалією, так як за формою дуже «схожий» на тимчасові ряди з навчальної множини.



а) ряд 1 екзаменаційна множина

б) ряд 1 екзаменаційна множина

в) ряд 1 екзаменаційна множина

Рисунок 2.17 – Екзаменаційна вибірка TS_TEST з трьох тимчасових рядів

Розглянемо тепер набір ситуацій. Кожна ситуація Сит1 – Сит9 відноситься до одного з класів, взятих для прикладу з описаного раніше набору даних «циліндр-дзвін-воронка».

Для стислості позначимо класи CY, BE і FU (cylinder – циліндр, bell – дзвін, funnel – воронка [22]). Необхідно побудувати модель, що описує «нормальне» протікання процесів і дозволяє для кожної ситуації визначити, чи стосується вона до «нормальних» або «аномальних». В даному випадку перед нами завдання виявлення аномалій в наборах тимчасових рядів, коли в навчальній множині містяться приклади декількох класів, оголошених «нормальними ситуаціями». Це можуть бути, наприклад, тимчасові ряди, що відносяться до двох класів: «циліндр» і «дзвін».

У загальному випадку для вирішення завдання визначення аномалій в наборах тимчасових рядів з декількома класами поширений байесівський підхід, підходи, засновані на використанні нейронних мереж, продукційних правил.

Таблиця 2.2 – Опис ситуацій на об'єкті для випадку 1 датчика

t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	-1,07	-0,13	0,85	0,96	0,81	0,84	-0,08	-1,01	-0,90	-1,13	CY
Сит2	-0,72	-0,70	1,25	1,23	1,27	0,03	-0,76	-0,71	-0,71	-0,74	CY
Сит3	-0,94	-0,84	1,06	0,97	1,01	1,04	-0,35	-0,92	-0,83	-0,80	CY
Сит4	-0,56	-0,62	-0,19	0,64	1,45	1,39	-0,69	-0,61	-0,66	-0,62	BE
Сит5	-0,98	-0,91	-0,59	-0,53	0,30	0,80	1,25	1,41	-0,98	-0,99	BE
Сит6	-0,54	-0,44	-0,28	0,75	1,61	0,40	-0,45	0,53	-0,38	-0,61	BE
Сит7	-0,45	1,05	1,25	0,61	-0,35	-0,50	-0,39	-0,27	-0,89	-0,28	FU
Сит8	-0,68	-0,67	1,63	1,07	0,69	0,01	-0,59	-0,70	-0,64	-0,53	FU
Сит9	-1,01	0,50	1,35	0,89	0,33	0,18	-0,34	-0,75	-0,98	-0,65	FU

2.7 Завдання виявлення аномалій в наборах тимчасових рядів з одним класом

2.7.1 Розробка методу виявлення аномалій

У даній роботі пропонується метод виявлення аномалій в наборах тимчасових рядів, який є модифікацією методу, заснованого на «точному описі виключення» [24].

Вихідна постановка задачі, дана в [24], наступна: для заданої кінцевої множини об'єктів I необхідно отримати множину-виняток I_x.

Для цього на множині I вводяться:

1. функція неподібністі (dissimilarity) $D\ (I_j)$, $I_j \in I$, певна на P (I) – множина всіх підмножин I і приймає позитивні значні значення;

2. функція потужності (cardinality) $C\ (I_j)$, $I_j \in I$, певна на P (I) – множина всіх підмножин I і приймає позитивні значні значення, така, що для будь-яких $I_1 \subset I$, $I_2 \subset I$ виконується $I_1 \subset I_2 \Rightarrow C\ (I_1) < C\ (I_2)$;

3. «фактор згладжування» (smoothing factor) $SF\ (I_j) = C\ (I \setminus I_j\ ) * (D\ (I) - D\ (I \setminus I_j\ ))$, який обчислюється для кожного $I_j \subseteq I$.

Тоді $I_x \subset I$ буде вважатися множиним-винятком для I щодо D і C, якщо його фактор згладжування SF (Ix) максимальний [24].

Неформально, множина-виняток – це найменша підмножина з I, яке вносить найбільший вклад в його неподібність. Фактор згладжування показує, наскільки може бути зменшено не подібність множини I, якщо з нього виключити підмножину Ij.

На підставі методу, описаного в [24], автором був розроблений алгоритм $TS - ADEEP$ [3], призначений для виявлення аномалій в наборах тимчасових рядів. Як множина I розглядаються множини $TS\_STUDY \cup \{ts\_test_j\}$ для кожного $ts\_test_j \in TS\_TEST$.

Функція неподібності для тимчасових рядів буде задана в такий спосіб: $D\ (I_j) = \frac{1}{|\ I_j|} * \sum_{i \in I_j} |i - \overline{I_j}\ |^2$, де $\overline{I_j} = \sum_{i \in I_j} \frac{\mathrm{i}}{|\ I_j|}$.

Спочатку обчислюється $\overline{I_j}$ – середнє для тимчасових рядів з I_j. У даному випадку це еквівалентно обчисленню середнього для звичайних векторів: i – часовий ряд з підмножини I_j, | I_j | – число елементів в | I_j |.

Функція неподібністі обчислюється як сума квадратів відстаней (використовується евклідова метрика) між середнім і векторами з I_j, яка потім нормалізується – ділиться на число елементів у множині I_j.

Функція потужності задається формулою: $C\ (I \setminus I_j) = \frac{1}{|\ I_j\ | + 1}$.

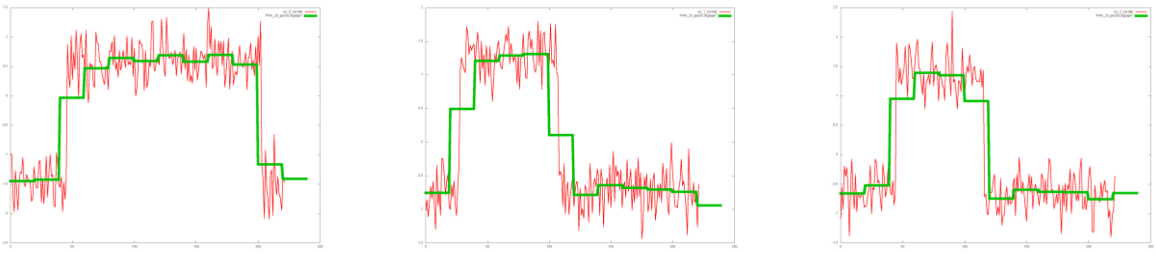
Формула для обчислення фактору згладжування має колишній вигляд $SF\ (I_j) = C\ (I \setminus I_j\ ) * (D\ (I) - D\ (I \setminus I_j\ ))$.

Якщо множина-виняток I_x, отримане для $I = TS\_STUDY \cup \{ts\_test_j\}$ містить $ts\_test_j, 1 \leq j \leq |\ TS\_TEST\ |$, то ts_test_j є аномалією.

2.7.2 Алгоритм «TS-ADEEP»

На підставі описаного вище методу реалізований непараметричний [23] алгоритм $TS - ADEEP$ [3] для визначення аномалій в наборах тимчасових рядів для навчальної множини з одним класом.

Розглянемо роботу алгоритму $TS\text{-}ADEEP$ на прикладі. Нехай в навчальній множині три тимчасових ряду – рисунок 2.18 (позначимо їх для зручності $cyl1$, $cyl2$, $cyl3$).



а) ряд 1 навчальна множина   б) ряд 2 навчальна множина   в) ряд 3 навчальна множина

Рисунок 2.18 – Навчальна вибірка $TS\text{-}ADEEP$ з трьох тимчасових рядів

Потрібно визначити, чи є часовий ряд, представлений на рисунку 2.19 (позначимо його bel), аномалією.

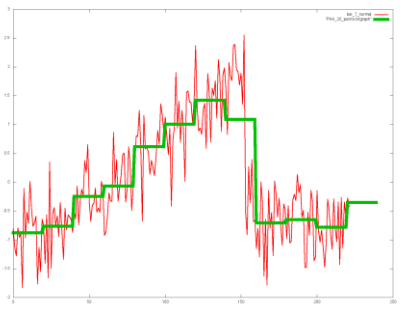


Рисунок 2.19 – bel

Відповідно до алгоритму множина I буде складатися із зазначених чотирьох тимчасових рядів: $I = \{cyl1, cyl2, cyl3, bel\}$. Розглядаються всі можливі підмножини I_j з I (за винятком порожнього множини і самого множини I). Таких підмножин $2^{|I|} - 2 =$ 14: $\{\{cyl1\}, \{cyl2\}, \{cyl3\}, \{bel\}, \{cyl1, cyl2\}, \{cyl1,$ $cyl3\}, \{cyl1, bel\}, \{cyl2, cyl3\}, \ \{cyl2, bel\}, \{cyl3, bel\}, \{cyl1, cyl2, cyl3\}, \{cyl2\ ,$

$cyl3, bel\}, \{cyl1, cyl3, bel\}, \{cyl1, cyl2, bel\}\}$. Для кожного з підмножин за вказаними формулами обчислюється фактор згладжування. Результати обчислень наведені в таблиці 2.9.

Таблиця 2.3 – Результати обчислення фактору згладжування для підмножин I

Підмножини I_j	Фактор згладжування
cyl2, cyl3, bel	0.370713
cyl1, cyl3, bel	0.370713
cyl3, bel	0.0677333
cyl1, cyl2, bel	0.370713
cyl2, bel	0.205667
cyl1, bel	0.45465
bel	0.136392

В даному випадку максимальний фактор згладжування (0.45465) має множина, що складається з часових рядів $I_x = \{cyl1,\ bel\}$, отже, воно і є множинним-винятком. А так як тимчасовий ряд bel потрапив в множину-виключення ($\{bel\} \in I_x$), то він є аномалією.

Обчислювальна складність алгоритму «TS-ADEEP»

Була розрахована обчислювальна складність алгоритму TS $-ADEEP$. Нехай N – число тимчасових рядів в розглянутій множині $TSset$. Для пошуку множини-виключення треба розглянути булеан $TSset$, за винятком порожньої множини і самого $TSset$. Загальна кількість підмножин I_j за винятком порожнього і самого $TSset$ рівно рівно $2^N - 2$ (не перевищує 2^N). Таким чином, складність алгоритму – O (2^N), в зв'язку з чим не рекомендується використовувати в якості навчальних множин велике число тимчасових рядів (більше 20). Однак якщо врахувати той факт, що множина-виняток – це найменша підмножина з I, яка вносить найбільший вклад в його неподібність, то можна обмежитися розглядом підмножин не більше деякого заданого розміру, що дозволяє значно скоротити перебір без зниження точності виявлення аномалій в більшості експериментів, проведених в роботі.

2.8 Завдання виявлення аномалій в наборах тимчасових рядів з декількома класами

2.8.1 Розробка методу виявлення аномалій

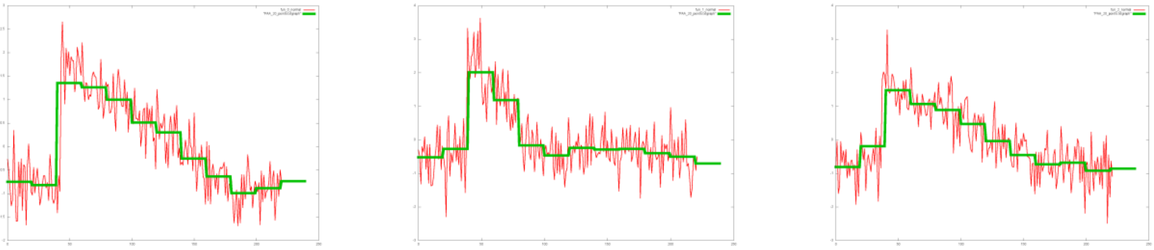
У даній роботі пропонується метод виявлення аномалій в наборах тимчасових рядів з декількома класами, який є узагальненням методу виявлення аномалій для випадку навчальної множини, що містить приклади одного класу.

Узагальнення є досить очевидним: розділивши навчальну множину на підмножини, що містять приклади тільки одного класу і послідовно застосувавши до них і кожному з тимчасових рядів екзаменаційної множини метод виявлення аномалій в наборах тимчасових рядів з одним класом, можна визначити, чи є даний часовий ряд аномалією. Якщо часовий ряд є аномалією для кожної підмножини, то він є аномалією і для всієї навчальної множини.

2.8.2 Алгоритм «TS-ADEEP-Multi»

На підставі описаного вище методу реалізований непараметричний [21] алгоритм TS−ADEEP−Multi, який є узагальненням алгоритму TS−ADEEP для випадку навчальної множини, що містить приклади декількох класів тимчасових рядів.

Розглянемо роботу алгоритму TS−ADEEP−Multi на прикладі. Нехай в навчальній множині шість тимчасових рядів: три тимчасових ряду з попереднього прикладу, рисунок 2.18 (позначимо їх для зручності $cyl1$, $cyl2$, $cyl3$), і три тимчасових ряду, зображених на рисунку 2.20 (позначимо їх для зручності $fun1$, $fun2$, $fun3$).



а) ряд 1 навчальна множина | б) ряд 2 навчальна множина | в) ряд 3 навчальна множина

Рисунок 2.20 – Навчальна вибірка TS–ADEEP–Multi з трьох тимчасових рядів

Потрібно визначити, чи є часовий ряд на рисунку 2.20 (позначимо його bel) аномалією.

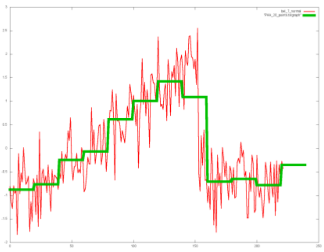


Рисунок 2.20 – bel

Відповідно до алгоритму будуть розглянуті дві множини – $I1$ = {$cyl1$, $cyl2$, $cyl3$, bel}, $I2$ = {$fun1$, $fun2$, $fun3$, bel}. Аналогічно алгоритму $TS–ADEEP$ розглядаються всі можливі підмножини з $I1$ і $I2$ і для кожного з них за вказаними формулами обчислюється фактор згладжування. для $I1$ множинно-винятком стало {$cyl1$, bel} з фактором згладжування 0.45465, для $I2$ – {bel} з фактором згладжування 0.385212. Так як часовий ряд bel є аномалією для обох класів з навчальної множини, цей часовий ряд є аномалією по відношенню і до всієї навчальної множини.

Обчислювальна складність алгоритму «TS-ADEEP-Multi».

Була розрахована обчислювальна складність алгоритму $TS\text{–}ADEEP\text{–}Multi$. Нехай N – число тимчасових рядів в розглянутій множині $TSset$, $N1 < N$ – максимальне число тимчасових рядів, що належать одному і тому ж класу, k – число класів. Для пошуку множини-виключення треба розглянути булеан $TSset$, за винятком порожнього множини і самого $TSset$.

Загальна кількість підмножин I_j за винятком порожнього і самого $TSset$ не перевищує 2^N. Таким чином, складність алгоритму – $O\ (k * 2^N)$, в зв'язку з чим не рекомендується використовувати в якості навчальних множин велике число тимчасових рядів одного і того ж класу (понад 20). Аналогічно алгоритму «TS–ADEEP», можна скоротити перебір, обмежившись розглядом підмножин не більше деякого заданого розміру.

2.8.3 Використання дерева рішень для виявлення аномалій в наборах тимчасових рядів з декількома класами

У першому розділі було розглянуто найбільш успішні для індуктивного формування понять моделі подання знань – дерева рішень. Ця модель використовується в ряді алгоритмів, що відносяться до категорії «навчання з учителем». Відповідно до цієї стратегії на основі навчальної вибірки, що містить приклади і контрприклади об'єктів певного класу, будується дерево рішень, що представляє собою особливу форму тесту, що дозволяє в подальшому успішно класифікувати нові приклади, які не ввійшли спочатку в навчальну вибірку.

Відомий ряд алгоритмів, результатом роботи яких буде побудоване в певній формі дерево рішень: алгоритм ДРЕВ [8], алгоритм, заснований на метриці Хеммінга [18], ID3 [15] і різні модифікації цих алгоритмів – C4.5 [21], ID5R [17] та інші – знайшли широке поширення і зарекомендували себе в широкому спектрі додатків.

Формально дерево рішень – це зважений орієнтований граф $T = (V, E)$. У множині вершин V виділимо вершину $v_0 \in V$ – корінь дерева. Всі вершини розділимо на два класи: $V_\mathrm{i} \subset V$ – множина внутрішніх вершин (вузлів) дерева; V_i включає в себе такі вершини, з яких виходять дуги; $V_\mathrm{l} \subset V$ – множина зовнішніх, кінцевих, вершин дерева (листя); V_l включає в себе такі вершини, з яких дуги не виходять; V_i і V_l утворюють розбиття множини вершин V дерева рішень T: $V_\mathrm{i} \cap V_\mathrm{l} = \emptyset$, $V_\mathrm{i} \cup V_\mathrm{l} = V$.

Внутрішні вершини V_i дерева зважені (позначені) іменами атрибутів, використовуваних при ознаковому описі об'єктів. Вершини-листя V_l зважені (позначені) іменами класів.

Кожна дуга e дерева рішень виважена умовою «атрибут = значення атрибуту» (для якісних значень атрибутів) або «атрибут σ значення атрибуту» (для кількісних значень атрибутів, $\sigma \in \{\geq, >, <, \leq\}$), де «атрибут» – ім'я атрибута в вершині, з якої виходить дуга e, «значення атрибута – одне з можливих значень (кількісне або якісне) ознаки« атрибут ».

Розглянемо можливість застосування алгоритмів побудови дерев рішень для роботи з такими об'єктами, як тимчасові ряди. З перерахованих вище алгоритмів візьмемо алгоритм ID3, оскільки він дозволяє будувати дерева рішень, відмінні від бінарних, і успішно працює з символьними даними. Цей алгоритм також дозволяє класифікувати приклади в разі їх приналежності до окремих класів (2, 3 і більше). Вихідними даними для алгоритму побудови дерева рішень є навчальна множина, представлена у вигляді таблиці. Кожен рядок таблиці містить опис одного з прикладів із зазначенням того, до якого класу належить даний приклад. Опис прикладу є рядком значень атрибутів (ознак), що характеризують властивості даного об'єкта.

Для розглянутого раніше алгоритму «TS–ADEEP–Multi» використовувалися вихідні дані, представлені у вигляді таблиці 2.7. Кожен рядок таблиці 2.7 представляє опис одного з тимчасових рядів, причому явно зазначено, до якого класу належить цей ряд. Щоб мати можливість використовувати ці дані як навчальну вибірку для побудови дерева рішень алгоритмом ID3 застосуємо до числових даних перетворення в символьну форму.

Перетворення виконується за допомогою алгоритму SAX, описаного раніше. Результат перетворення представлений в таблиці 2.4 (використовувався алфавіт з 10 символів). Таблиця 2.4 з формальної точки зору може бути використана як вихідні дані (навчальна вибірка) для алгоритму ID3: кожен рядок являє опис одного об'єкта – тимчасового ряду; відомо, до якого класу належить об'єкт (CY, BE, FU), атрибутами є моменти часу (0, 1, 2,... 9), а їх значеннями – показники датчиків в дискретно символьному поданні до відповідного моменту часу.

Таблиця 2.4 – Опис ситуацій на об'єкті для випадку 1 датчика – символьне уявлення

t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	B	E	I	I	H	I	E	B	B	B	CY
Сит2	C	C	I	I	I	F	C	C	C	C	CY
Сит3	B	B	I	I	I	I	D	B	C	C	CY
Сит4	C	C	E	H	J	J	C	C	C	C	BE
Сит5	B	B	C	E	G	H	I	J	B	B	BE
Сит6	C	D	D	H	J	G	D	C	D	C	BE
Сит7	D	I	I	H	D	D	D	D	B	D	FU
Сит8	C	C	J	I	H	F	C	C	C	C	FU
Сит9	B	G	J	I	G	F	D	C	B	C	FU

На рисунку 2.21 представлено дерево рішень, отримане алгоритмом ID3 за навчальною вибіркою (таблиця. 2.4).

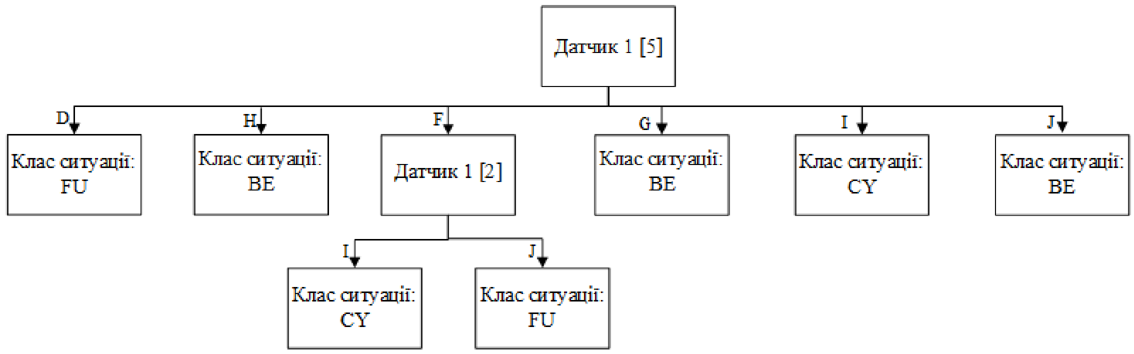


Рисунок 2.21 – Дерево рішень

Отримане дерево рішень можна використовувати для виявлення аномалій в наборах тимчасових рядів: якщо воно відносить певний часовий ряд ts до одного з класів CY, BE або FU, то розглянутий часовий ряд не є аномалією. В іншому випадку тимчасової ряд ts є аномалією.

Незважаючи на можливість використовувати описані вище методи для виявлення аномалій, дані методи мають недолік: вони можуть бути використані для вирішення лише приватного завдання – виявлення аномалій для набору динамічних об'єктів з одним атрибутом. У зв'язку з цим необхідний метод, що дозволяє працювати з динамічними об'єктами загального виду: необхідно враховувати більш однієї ознаки (що відповідає наявності декількох датчиків), а також той факт, що ситуації можуть розвиватися за різні інтервали часу. Крім того, при значній кількості ситуацій, що використовуються як вихідні дані для завдання виявлення аномалій, модель може бути неефективною, в зв'язку з чим необхідно використовувати лише «істотні» дані з набору вихідних ситуацій.

3 ЗАВДАННЯ УЗАГАЛЬНЕННЯ ДЛЯ ДИНАМІЧНИХ ОБ'ЄКТІВ. ЗАГАЛЬНИЙ ВИПАДОК

У другому розділі розглядалася лише приватні завдання і методи їх вирішення – для найбільш простого випадку, коли динамічний об'єкт узагальнення являє собою часовий ряд. При цьому час розглядався лише формально, неявно, тоді як в реальних системах підтримки прийняття рішень потрібно явно враховувати фактор часу.

У цьому розділі описано застосування апарату темпорального дерева рішень, що дозволяє вирішувати задачу узагальнення для динамічних об'єктів з довільним числом атрибутів за умови використання символьного опису такого атрибута (тимчасового ряду).

Нашою метою є дослідження випадку, коли динамічний об'єкт узагальнення характеризується q> 1 ознаками. Припустимо (для прикладу), що q = 3. Задамо деякий $r = t^*$ – максимальний інтервал часу, на якому будемо розглядати ситуацію – такий проміжок часу відповідає максимальній довжині часового ряду. Тут довжина тимчасового інтервалу r = 10, кожен рядок таблиці являє собою значення одного з параметрів на даному інтервалі. Відзначимо, що подання до таблиці 3.1 в точності відповідає уявленню динамічного об'єкта.

Таблиця 3.1 – Приклад динамічного об'єкта узагальнення для випадку отримання спостережень від трьох датчиків

t	0	1	2	3	4	5	6	7	8	9
Датчик1	-0,56	-0,62	-0,19	0,64	1,45	1,39	-0,69	-0,61	-0,66	-0,62
Датчик2	-0,98	-0,91	-0,59	-0,53	0,30	0,80	1,25	1,41	-0,98	-0,99
Датчик3	-0,54	-0,44	-0,28	0,75	1,61	0,40	-0,45	-0,53	-0,38	-0,61

Приклад набору динамічно змінюються ситуацій (Сит1–Сит4) для випадку, коли поведінка складної системи контролюється показаннями декількох датчиків (q = 3), приведено в табл. 3.2., тут довжина тимчасового інтервалу, на якому ведуться спостереження за ситуацією, r = 10, для опису кожної ситуації використовуються показання трьох датчиків на заданому інтервалі, кожна ситуація відноситься до класу $NORM$ (відповідає нормальному стану системи, поведінку якої слід контролювати). Заданий таким чином набір ситуацій пропонується використовувати як вихідні дані для вирішення завдання узагальнення.

Таблиця 3.2 – Набір ситуацій на об'єкті для випадку 3 датчиків

	t	0	1	2	3	4	КС
Сит1	Дат1	-1.07	-0.13	0.85	0.96	0.81	NORM
	Дат2	-0.72	-0.70	1.25	1.23	1.27	
	Дат3	-0.94	-0.84	1.06	0.97	1.01	
Сит2	Дат1	-0.56	-0.62	-0.19	0.64	1.45	NORM
	Дат2	-0.98	-0.91	-0.59	-0.53	0.30	
	Дат3	-0.54	-0.44	-0.28	0.75	1.61	
Сит3	Дат1	-0.45	1.05	1.25	0.61	-0.35	NORM
	Дат2	-0.68	-0.67	1.63	1.07	0.69	
	Дат3	-1.01	0.50	1.35	0.89	0.33	
Сит4	Дат1	-0,72	-0,70	1,25	1,23	1,27	NORM
	Дат2	-0,98	-0,91	-0,59	-0,53	0,30	
	Дат3	-0,68	-0,67	1,63	1,07	0,69	

Набір динамічних об'єктів узагальнення, або динамічних ситуацій, може описувати різні стани складного технічного об'єкта або системи, причому об'єкти можуть описувати як нормальний стан системи, так і ненормальний, або аномальний, тобто відповідний несправності. У такому вигляді набір динамічних об'єктів може використовуватися як вихідні дані для вирішення завдання діагностики – визначення несправності системи і вказівки причин, що викликали несправність. Розглянемо задачу діагностики більш докладно.

3.1 Про технічної діагностики

Технічна діагностика [24] – науково-технічна дисципліна, що вивчає і встановлює ознаки дефектів технічних об'єктів, а також методи і засоби виявлення і пошуку (вказівки місця розташування) дефектів. Основний предмет технічної діагностики – організація ефективної перевірки справності, працездатності, правильності функціонування технічних об'єктів (деталей, елементів, вузлів, блоків, заготовок, пристроїв, виробів, агрегатів, систем, а також процесів передачі, обробки та зберігання матерії, енергії та інформації), тобто організація процесів діагностування технічного стану об'єктів при їх виготовленні та експлуатації, в тому числі під час, до і після застосування за призначенням, при профілактиці, ремонті та зберіганні. Діагностування – одна з важливих заходів забезпечення і підтримки надійності технічних об'єктів.

Діагностування як розділ штучного інтелекту займається розробкою методів і алгоритмів, здатних визначити коректність роботи досліджуваного об'єкта (системи). Якщо система не функціонує належним чином, потрібно якомога точніше визначити, в якій частині системи відбулася відмова і яка помилка сталася. Визначення помилки відбувається на основі спостережень, які дають інформацію про поведінку системи.

Термін «діагностування» також відноситься до визначення несправності системи. Вихідними даними для задачі діагностики зазвичай буває опис некоректних станів об'єкта (або ситуацій, які можуть виникнути на об'єкті) і причини, що до цього призвели. При навчанні на таких даних діагностична система повинна побудувати деяку узагальнену модель, яка в подальшому змогла б розпізнавати подібні (або схожі) ситуації на об'єкті і вказувати причину неполадки.

Математична модель об'єкта діагностування (детермінована або імовірнісна) являє собою опис об'єкта в справному та в несправному його станах у вигляді формальних залежностей між можливими впливами на об'єкт і його реакціями на ці дії. Моделі (навіть справних об'єктів), які використовуються при діагностуванні, можуть відрізнятися від моделей, що використовуються при проектуванні тих же об'єктів. У разі виявлення несправності може бути зроблено деякий коригуючий вплив на об'єкт, що діагностується з метою переведення його в справний стан.

Алгоритм діагностування передбачає виконання деякої умовної або безумовної послідовності певних експериментів з об'єктом. Експеримент характеризується або робочим впливом і складом контрольованих ознак, що визначають реакцію об'єкта на вплив.

Діагностика на основі використання моделі об'єкта

У даній роботі пропонується використовувати методи діагностики на основі використання моделі об'єкта [23]. Для імітації поведінки об'єкта і виявлення несправностей може бути використана модель спеціального виду, яка описує структуру і поведінку складного технічного об'єкта; дана модель являє собою четвірку $<O, E, S, B>$, де:

- O – множина компонент складного технічного об'єкту;
- E – функціональні зв'язки між компонентами;
- S – множина змінних, що описує стан системи (в технічній діагностиці це найчастіше вимірювання, одержувані від датчиків, встановлених в системі, або результати обчислень виконані над отриманими вимірами);
- B – множина керуючих дій, допустимих в системі.
- Для опису окремого компонента $o \in O$ також використовується модель, що представляє із себе трійку $<S', M, R>$, де:

- $S' \subseteq S$ – підмножина змінних, що описують стан даного компонента;

- M – набір режимів роботи, що включають в себе стан «норма» (коректна поведінка) і стану «несправність» (некоректна поведінка);

- R – набір відносин, що пов'язують множину змінних S, що описують стан системи, і набір режимів роботи M.

Введемо поняття C_n – множина станів складного технічного об'єкта, які діагностуються як нормальні, C_f – множина станів, в яких спостерігаються несправності на об'єкті. Для використання методів штучного інтелекту в діагностиці пропонується сформувати опис поняття C_n і C_f в рамках введеної моделі. На основі отриманих узагальнених описів класів C_n і C_f необхідно визначити, яка з несправностей сталася на об'єкті і – в більш складному випадку – виробити рекомендації щодо вибору відновлення дії на складному технічному об'єкті, така дія має переводити систему зі стану «несправне» в стан «норма».

Для оцінки ефективності побудованої моделі, необхідно порівняти поведінку, передбачене моделлю, і спостерігається поведінка об'єкта. Результати спостережень за поведінкою системи представлені у вигляді S – множини показань, що надходять з датчиків, встановлених на складному технічному об'єкті. Щоб побудована модель була корисною і для бортової діагностики, вона повинна включати в себе дії, які повинні проводитися в разі виявлення несправності. У загальному випадку дії (відновлювальні дії) характеризуються деякою вартістю, яка найчастіше виражається в зменшенні функціональності системи. Таким чином, основна мета процедури бортової діагностики полягає у виборі оптимальної дії в аварійному режимі.

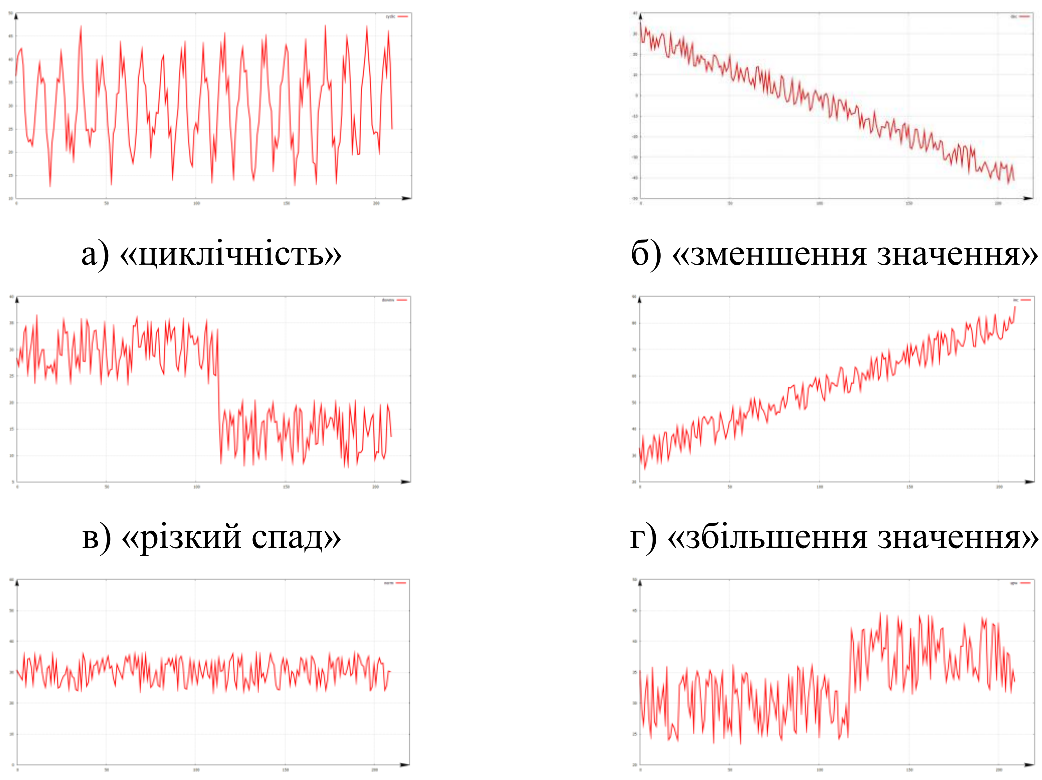
Діагностування на основі моделі являє собою окремий випадок абдуктивного виведення.

Успішність функціонування моделей, що використовуються в діагностиці, залежить від вибору способу описів класів ситуацій C_n і C_f. Для опису класів ситуацій можна використовувати різні методи, такі як продукційні моделі, нечіткі множини, наближені множини, дерева рішень.

У загальному випадку динамічний об'єкт узагальнення має більше одного параметру, тобто являє собою набір з декількох тимчасових рядів.

Використовуючи 2 датчика (2 параметра), можна розрізнити вже всі 3 класу: Клас 2 і Клас 3 – на підставі показань першого датчика (Параметр 1) Клас 1 і Клас 2 – на підставі показань другого датчика (Параметр 2). Таким чином, передбачається, що наявність більшої кількості параметрів може підвищити розрізняти здатність алгоритмів, що використовують такі дані.

Розглянемо більш докладно приклад для набору даних «контрольні карти», що містить 6 класів, що відповідають різним зразкам поведінки параметра (рисунок 3.1).



а) «циклічність» б) «зменшення значення»

в) «різкий спад» г) «збільшення значення»

д) «нормальне значення» е) «різке зростання»

Рисунок 3.1 – Набір даних «контрольні карти»

Сформуємо з вихідного набору даних новий набір наступним чином: динамічні об'єкти узагальнення матимуть п'ять параметрів, тобто являти собою набір з п'яти тимчасових рядів. Кожен новий клас буде містити 5 тимчасових рядів з наступного упорядкованого перерахування:

1. «циклічність»;
2. «зменшення значення»;
3. «різкий спад»;
4. «збільшення значення»;
5. «нормальне значення»;
6. «різке зростання».

Першим класом в новому наборі буде динамічний об'єкт з п'ятьма параметрами, що представляють собою такі тимчасові ряди:

1. «зменшення значення»;
2. «різкий спад»;
3. «збільшення значення»;
4. «нормальне значення»;
5. «різке зростання».

Другим класом в новому наборі буде динамічний об'єкт з п'ятьма параметрами, що представляють собою такі тимчасові ряди:

1. «циклічність»;
2. «різкий спад»;
3. «збільшення значення»;
4. «нормальне значення»;
5. «різке зростання».

Таким чином, жоден з параметрів не є найбільш інформативним і кожен з параметрів вносить свій внесок в процесі поділу об'єктів на класи. Передбачається, що наявність більшої кількості параметрів може підвищити розрізняючу здатність алгоритмів, що використовують такі дані.

Очевидно, що найбільш інформативним є Параметр 5, так як значення цього параметра (форма тимчасового ряду) різні для кожного класу. Передбачається, що значення параметрів Параметр 1 і Параметр 4 повинні кілька поліпшити точність класифікації за рахунок додаткової інформації.

Отже, в нашому випадку вихідні дані для завдання діагностики – це набір динамічних об'єктів (динамічних ситуацій).

Приклад набору ситуацій, що використовуються як вихідні дані для завдання діагностики, наведено в таблиці 3.3. Припустимо, що для контролю за станом складного технічного об'єкта використовуються q = 3 датчика. Задамо деякий r = t^* – ця величина визначає максимальну довжину тимчасових рядів, яку будемо розглядати. Назвемо набір тимчасових рядів, значення яких отримані з кожного з датчиків за період часу t^*, ситуацією на об'єкті. Приклад ситуації для даного випадку наведено в таблиці 3.1. Завдання ускладнюється тим, що час розвитку ситуації на об'єкті може бути різним. У таблиці 3.3 приведено приклад, коли ситуації розглядаються на тимчасовому інтервалі довжиною t^* = 10, тоді як ситуація Сит3 розвивається за час $\hat{t} = 8, \hat{t} < t^*$. У такому випадку час прийняття рішення менше часу, на якому розглядаються ситуації.

Таблиця 3.3 – Опис ситуацій на об'єкті для випадку 3 датчиків. Час для прийняття рішення менше t *

	t	0	1	2	3	4	5	КС
Сит1	Датчик1	-1.07	-0.13	0.85	0.96	0.81	0.84	СУ
	Датчи2	-0.72	-0.70	1.25	1.23	1.27	0.03	
	Датчик3	-0.94	-0.84	1.06	0.97	1.01	1.04	
Сит2	Датчик1	-0.56	-0.62	-0.19	0.64	1.45	1.39	ВЕ
	Датчик2	-0.98	-0.91	-0.59	-0.53	0.30	0.80	

Продовження таблиці 3.3

	Датчик3	-0.54	-0.44	-0.28	0.75	1.61	0.40	
Сит3	Датчик1	-1,03	0,39	0,97	0,82	0,84	-0,63	CY
	Датчик2	-0,73	0,10	1,23	1,27	-0,15	-0,68	
	Датчик3	-0,94	-0,04	0,95	1,04	1,01	-0,86	
Сит4	Датчик1	-0,45	1,05	1,25	0,61	-0,35	-0,50	FU
	Датчик2	-0,68	-0,67	1,63	1,07	0,69	0,01	
	Датчик3	-1,01	0,50	1,35	0,89	0,33	0,18	
Сит5	Датчик1	-0,72	-0,70	1,25	1,23	1,27	0,03	CY
	Датчик2	-0,98	-0,91	-0,59	-0,53	0,30	0,80	
	Датчик3	-0,68	-0,67	1,63	1,07	0,69	0,01	

Розглянемо тепер можливість явно ввести час як один з параметрів в опис стану складного технічного об'єкта. Розширимо признаковий опис об'єктів – введемо поняття «час» як один з атрибутів, використовуваних явно при побудові дерева рішень. Будемо використовувати дискретний час: t = 0,1,2.

Використовуючи спосіб представлення динамічних об'єктів (тимчасових рядів), описаний в другому розділі, можна отримати символьне уявлення для динамічних об'єктів. Після дискретизації тимчасових рядів набір динамічних об'єктів буде виглядати наступним чином. Одним із способів, зручних для роботи з темпоральною або тимчасовою, інформацією при узагальненні понять є темпоральні продукційні правила [21]. Однак часто темпоральні продукційні правила важкі для сприйняття і інтерпретації людиною. Іншим способом, який і буде розглянуто в даній роботі, є темпоральні дерева рішень [22].

3.2 Темпоральні дерева рішень

Введемо тепер поняття темпорального, або тимчасового, дерева рішень T_{temp}.

Неформально темпоральне дерево рішень T_{temp} – це дерево, внутрішні вершини якого позначені іменами атрибутів і тимчасової міткою, а вершини-листя містять назви класів – в задачах діагностики це зазвичай вид несправності і, можливо, пропоноване в даній ситуації відновну дію. Дуги темпорального дерева рішень позначені перевірками значень атрибутів в певний момент часу. Набір ситуацій на об’єкті, символьне представлення (таблиця 3.4).

Таблиця 3.4 – Набір ситуацій на об’єкті – символьне представлення

t	0	1	2	3	4	5	6	7	8	9	КС
Сит1	B	E	I	I	H	I	E	B	B	B	CY
Сит2	C	C	I	I	I	F	C	C	C	C	CY
Сит3	B	B	I	I	I	I	D	B	C	C	CY
Сит4	C	C	E	H	J	J	C	C	C	C	BE
Сит5	B	B	C	E	G	H	I	J	B	B	BE
Сит6	C	D	D	H	J	G	D	C	D	C	BE
Сит7	D	I	I	H	D	D	D	D	B	D	FU
Сит8	C	C	J	I	H	F	C	C	C	C	FU

Дано формальне визначення [12]. Нехай P – процес прийняття рішень, де A – набір можливих рішень, O – набір перевірок, які можуть бути проведені (відповідають параметрам динамічного об'єкта узагальнення в певні моменти часу), $out\ (o_i) = v_1, .., v_{ki}$ – можливі результати перевірки $o_i \in$ O (відповідають значенням параметрів динамічного об'єкта узагальнення в певні моменти часу). Темпоральне дерево рішень для P – це позначена деревоподібна структура $T_{temp} = <v0_{temp}, V_{temp}, E_{temp}, \Lambda_V, \Lambda_E, \tau>$, де:

- $<v0temp, Vtemp, Etemp>$ – деревоподібна структура з коренем $v0temp$, набором вершин $Vtemp$, і набором дуг $Etemp \subset Vtemp \times Vtemp$;
- $Vtemp = Vtemp\ I \cup Vtemp\ L$: $Vtemp$ розділене на множину внутрішніх вершин $Vtemp\ I$ і множину вершин-листя дерева $Vtemp\ L$;
- ΛV – маркована функція, визначена на $Vtemp$;
- ΛE – маркована функція, визначена на $Etemp$;
- якщо $v \in Vtemp\ I$, то $\Lambda V(v) \in O$ – кожна внутрішня вершина дерева позначена назвою перевірки;
- якщо $(v, c) \in Etemp$, то $\Lambda E\ (v, c) \in out\ (\Lambda V(v))$ – дуга з v в c позначена одним з можливих результатів перевірки, пов'язаної з вершиною v;
- більш того, якщо $(v, c1), (v, c2) \in Etemp$ і $\Lambda E((v, c1)) = \Lambda E((v, c2))$, то $c1 = c2$ і для кожного $n \in out\ (\Lambda V(v))$ існує с така, що $(v, c) \in Etemp$ і $\Lambda E\ ((v, c)) = n$ – з вершини v виходить в точності одна дуга, відповідна кожному можливому результату перевірки $\Lambda V(v)$;
- якщо $l \in Vtemp\ L$, то $\Lambda V(l) \in A$ – кожен лист дерева позначений одним з можливих рішень;
- $\tau\ (v)$ – тимчасова мітка;
- * додатково може бути присутнім наступне обмеження: якщо $v\ ' \in Vtemp\ I$ і існує v така, що $(v, v') \in Etemp$, то $\tau\ (v\ ') \geq \tau\ (v)$ – неубування тимчасової мітки при проходженні від кореня дерева до листя.

Таким чином, темпоральні дерево рішень – це зважений орієнтований граф $T_{temp} = (V_{temp}, E_{temp})$. У множині вершин V_{temp} виділена вершина $v0_{temp} \in$

V_{temp} – корінь дерева. Всі вершини розділені на два класи: $V\text{t}_{\text{emp I}} \subset V\text{temp}$ – множина внутрішніх вершини (вузлів) дерева; $V_{\text{temp I}}$ включає в себе такі вершини, з яких виходять дуги; $V_{\text{temp L}} \subset V_{\text{temp}}$ – множина зовнішніх, кінцевих, вершин дерева (листя); $V_{\text{temp L}}$ включає в себе такі вершини, з яких дуги не виходять; $V_{\text{temp I}}$ і $V_{\text{temp L}}$ утворюють розбиття множини вершин V_{temp} темпорального дерева рішень T_{temp}: $V_{\text{temp I}} \cap V_{\text{temp L}} = \emptyset$, $V_{\text{temp I}} \cup V_{\text{temp L}} = V_{\text{temp}}$.

Внутрішні вершини $V_{\text{temp I}}$ дерева зважені (позначені) назвою перевірки і тимчасовою міткою, визначальною, коли треба цю перевірку виробляти.

Вершини-листя $V_{\text{temp L}}$ зважені (позначені) назвою або номером ситуації з $C_{\text{n}} \cup C_{\text{f}}$ і, в більш складному випадку – пропонованими відновлювальною дією, якщо ситуація відноситься до класу C_{f}.

Кожна дуга e темпорального дерева рішень виважена результатом перевірки, проведеної в вершині, з якої вона виходить.

Таким чином, основними відмінностями темпоральних дерев рішень від звичайних дерев рішень є наявність мітки часу в кожному внутрішньому вузлі дерева. Перевірка значення атрибута у внутрішньому вузлі дерева проводиться тільки в тому випадку, якщо момент часу, яким позначений набір значень датчиків, збігається з часовою міткою в цьому вузлі.

3.3 Алгоритми побудови темпоральних дерев рішень

Додаткова мітка часу в кожному вузлі темпорального дерева рішень призводить до суттєвих відмінностей в уявленні даних для алгоритму побудови темпорального дерева рішень у порівнянні з алгоритмом, побудови звичайних дерев рішень, і до істотних відмінностей в самих алгоритмах.

Поява тимчасової змінної і необхідність враховувати крайні терміни ухвалення рішення про віднесення ситуації до певного класу призводять до того, що алгоритм побудови дерева рішень – тепер уже темпорального – буде кілька модифікований. Загальна схема алгоритму побудови темпорального дерева рішень наведена в прикладі 3.1.

```
Алгоритм Побудови_темпорального_дерева_решень
(S: Таблиця з ситуаціями,
O: Спостереження,
M: Модель відновних процесів)
Результат: Темпоральне дерево рішень Ttemp
ПОЧАТОК
Якщо для всіх ситуацій з S відновлювальні дії збігаються,
то повернути Лист (S, M)
Нехай D – мінімальний крайній термін для ситуацій з S.
Якщо ситуації з S невиразні на основі показників датчиків счасовою позначкою t ≤ D, то повернути Лист (S, M).
Вибрати спостереження <s*, t '>, яке буде перевірятися в даному вузлі дерева.
Нехай  s*1,s*2,s*n – розрізняються показання датчика s* в момент
часу t', а S*j, j = 1, 2, .., n – підмножини ситуацій з S, що складаються
з ситуацій з показанням s*jдатчика s * в момент часу t'.
Повернути темпоральні дерево рішень Ttemp з коренем,
позначеним обраним наглядом <s*, t '>,
і дугами, поміченими s*1,s*2...,s*n, що з'єднують корінь відповідно з деревами:
Побудова_темпорального_дерева_решень (S*1,*1, O \ {<s*, t '>}, M)
Побудова_темпорального_дерева_решень (S*2,, O \ {<s*, t '>}, M)
. . .
Побудова_темпорального_дерева_ S*n, O \ {<s*, t '>}, M)
КІНЕЦЬ
```

Приклад 3.1 – Псевдокод алгоритму – побудова темпорального дерева рішень

На вхід алгоритму подаються:

1. таблиця з ситуаціями;
2. спостереження у вигляді множини пар «датчик, тимчасова мітка»;
3. модель відновлювальних процесів (при наявності).

3.3.1 Алгоритм «CPD»

В роботі [22] було дано формальне визначення темпоральних дерев рішень і був запропонований базовий алгоритм побудови темпоральних дерев рішень (назвемо його «CPD» як скорочення від прізвищ авторів L. Console, C. Picardi, D. Dupre).

Алгоритм «CPD» представляє інтерес як один з перших алгоритмів, в якому реалізована процедура побудови темпорального дерева рішень.

Розглянемо далі докладно характерні особливості цього алгоритму. Цей алгоритм пропонувалося використовувати для бортової, або онлайн-діагностики, в зв'язку з чим вихідним обмеженням на темпоральні дерево рішень було спадання тимчасових міток при русі від кореня дерева до листя. Крім цього пропонувалося використовувати певну модель «відновлювальних дій», що дозволяють при виявленні несправності в роботі об'єкта зробити деякий керуючий вплив, яке має по можливості повернути об'єкт або систему в коректний стан. У загальному випадку відновну дію має деяку «вартість», яка виражається в тому, що функціональність об'єкта або системи зменшується, наприклад, зменшення швидкості або повна зупинка автомобіля, відключення яких-небудь модулів системи. Тому розглядається функція очікуваної вартості темпорального дерева рішень, очікувана вартість відновлювального дії, обраного за допомогою темпорального дерева рішень, щодо розподілу ймовірностей несправностей.

У зв'язку з цим побудова кожного вузла темпорального дерева рішень складається з двох етапів – на першому кроці береться до уваги вартість, вибираються тільки ті спостереження, зможуть забезпечити мінімальну очікувану вартість дерева. На другому з них вибирається найбільш інформативне спостереження.

Приклад роботи алгоритму «CPD»

Розглянемо роботу алгоритму «CPD» на прикладі. Нехай задано навчальну множину такого вигляду (таблиця 3.5), де кожному рядку відповідає деяка ситуація Сит$_i$, яка визначається: показаннями датчиків.

Таблиця 3.5 – Ситуації для побудови темпорального дерева рішень

	Датчик1								Датчик2								Tm
t	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
Сит1	n	n	n	n	h	h			l	v	v	v	v				5
Сит2	h	h	h						h	n	n						2
Сит3	n	n	n	n	h	h	h	h	l	l	l	l	v	v	v	v	7
Сит4	n	n	n	h	h	h	h		l	l	l	l	l	v	v		6
Сит5	h	h	h	h					h	n	n	n					3

Продовження таблиці 3.5

Сит6	n	n	n	h	h	h			l	v	v	z	z				5
Сит7	h	h	h	h	h	h			l	l	n	n	l	v			5
Сит8	h	h	h	h	h	h			h	h	n	n	l	l			5

Датчик$_j$, j = 1,2,3 в моменти часу t = 0, 1, .. 7 (тут z, n, l, v, h відповідають якісним показниками датчиків: n – норма, l і h відповідно низьке і високе значення, v – дуже низьке значення, z – нуль); відновлювальною дією Д$_i$ для кожної ситуації, позначених як a, b, c, d і відповідним деяким управляючим діям, які треба зробити в разі виявлення відповідної ситуації; крайнім терміном К$_i$ виконання відповідної дії.

Пояснимо роботу алгоритму «CPD» з побудови темпорального дерева рішень на прикладі даних з таблиці 3.6. У корені дерева необхідно розмістити перевірку: пару виду <s_i, t'>, де s_i – датчик, t' – момент часу. Визначаємо t_{up} – максимальний час спостереження, до настання якого можна не приймати ніяких рішень. Знаходимо мінімальний крайній термін для все ситуацій t_{up} = 2. Аналізуємо поведінку об'єкта в спостережуваних ситуаціях в моменти часу t = 0, 1, 2. Для кожного t виконуємо розбиття ситуацій на класи за принципом збігу значень атрибутів (значеннями атрибутів є показники датчиків h, l, n, v, z). За формулами для отриманих розбиття обчислюються оцінки вартості вибору управляючого впливу; вибирається розбиття з найменшою оцінкою вартості. У прикладі було вибрано розбиття {{Сит1}, {Сит6}, {Сит2, Сит5}, {Сит7}, {Сит8}, {Сит3}, {Сит4}}, отримане в момент часу t = 1. На основі даного розбиття виконується обчислення інформативності спостережень [24] і вибирається спостереження з найкращою оцінкою. У нашому випадку це $s2$ – спостереження, засновані на показаннях Датчик$_2$ в момент часу t = 1. Приклад темпорального дерева рішень, побудованого з використанням алгоритму «CPD», наведено на рисунку 3.2.

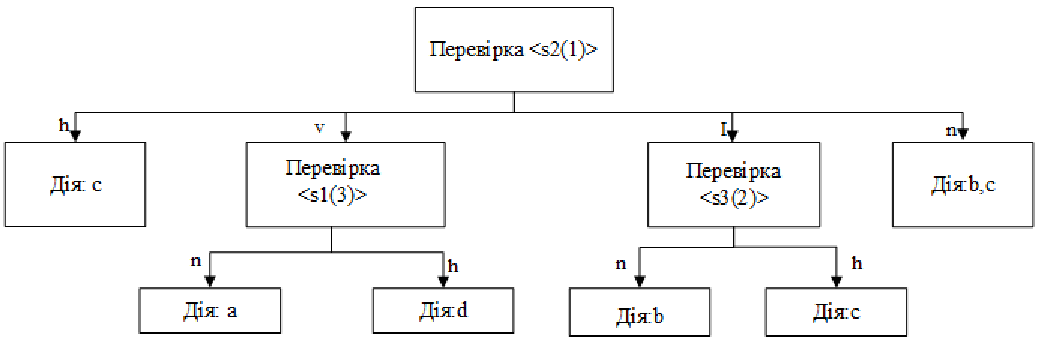


Рисунок 3.2 – Дерево рішень, побудоване з використанням алгоритму «CPD»

Подальші кроки алгоритму дозволяють добудувати і уточнити дерево рішень.

3.3.2 Алгоритм «Темпоральний ID3»

У роботі пропонується оригінальний алгоритм (назвемо його «Темпоральним ID3»), який є розширенням алгоритму ID3 [22], що враховує фактор часу. Псевдокод алгоритму представлений в табл. 3.12. Порівняно з алгоритмом «CPD», також дозволяє враховувати фактор часу, на темпоральні дерева рішень не накладається ніяких обмежень по тимчасовим мітках в вузлах, однак не враховується вартість відновлювальних дій при виборі спостереження на кожному кроці.

При виборі спостереження для розбиття використовується критерій «приріст інформативності» Куінлана [35]. Величина $Gain$ (<s, t>, S) = $Info$ (S) – $Info$ (<s, t>, S) показує кількість інформації, яку ми отримуємо завдяки спостереженню <s, t>. Алгоритм використовує цю величину для оцінки інформативності спостереження при побудові дерева рішень, що дозволяє отримувати дерева мінімальної висоти [23]. Процедура вибору спостереження з використанням даного критерію (приклад 3.2). Так як обмеження на неубування тимчасових міток при русі від кореня дерева до листя відсутня, при побудові темпорального дерева рішень показники датчиків будемо розглядати як звичайні атрибути [18] – наприклад, показання датчика s_1 в момент часу t = 0 буде атрибутом <s1, 0> , в момент часу t = 1 – атрибутом <s1, 1>. Алгоритм будує таке дерево, в якому з кожним вузлом асоційований атрибут, який є найбільш інформативним серед всіх атрибутів, ще не розглянутих на шляху від кореня дерева.

```
Алгоритм Вибір_спостереження_для_розбиття_Темпоральний_ID3
(S: Таблиця з ситуаціями, O: Спостереження)
Результат: o * – найбільш інформативне спостереження
ПОЧАТОК
Для всіх спостережень <s, t> з O обчислюємо кількість інформації, яку отримуємо завдяки цьому спостереженню:
Gain (<s, t>, S) = Info (S) - Info (<s, t>, S), де
Info (S) – ентропія для ситуацій з S (розподіл відновних процесів)
Info (<s, t>, S) – зважене середнє інформації, необхідної для ідентифікації класу ситуації в кожній підмножині, отриманій при розбитті множини ситуацій з S на основі значень <s, t>Повернути <s*, t'> – спостереження з найбільшим значенням Gain (<s, t>, S)
КІНЕЦЬ
```

Приклад 3.2 – Псевдокод алгоритму – вибір спостереження для Темпорального ID3

При виборі спостереження для розбиття слід розглядати тільки ті спостереження, для яких тимчасові мітки не перевищують за крайнього терміну для розглянутої в даний момент часу таблиці ситуацій S.

Обчислювальна складність алгоритму «Темпоральний ID3»

Розрахуємо обчислювальну складність алгоритму «Темпоральний ID3» за аналогією з алгоритмом ID3 [18].

Розмір областей визначення для всіх атрибутів дорівнює розміру використовуваного алфавіту $|A|$. Кількість рекурсивних викликів складе в гіршому випадку $1 + |A| + |A|^2 + ... + |A|^{k-1}$, де k – кількість атрибутів, рівне $q * r$ (q – число параметрів динамічного об'єкта узагальнення, r – довжина розглянутого часового ряду), загальна складність складе $C\,(TID3\,) = \sum_{i=0}^{k} \mathrm{C}(i) * \,|\,A\,|^{i}$, де $C(i)$ – складність одного кроку алгоритму. Трудомісткою операцією є обчислення інформативності атрибутів і визначення вирішального атрибута на кожному кроці, при цьому кількість альтернатив і розміри підмножини навчальних прикладів звужуються зі збільшенням глибини рекурсивної вкладеності. На рівні i вони складають відповідно $k - i$ і $\frac{n}{b^i}$, де n – кількість прикладів. Для обчислення інформативності атрибутів досить одного перегляду множини прикладів, тому загальне число операцій складе $C\,(i) = \frac{n}{b^i} \,*\, (k - i)$. Підставивши цей вираз в суму, що визначає загальну складність алгоритму, отримаємо $C\,(TID3) = O\,(k^2 \,*\, n) = O\,((q \,*\, r)^2 \,*\, n)$.

Приклад роботи алгоритму «Темпоральний ID3»

Розглянемо тепер докладніше процес формування темпорального дерева рішень з використанням алгоритму «Темпоральний ID3». Вихідні дані, будемо розглядати як масив, в якому є 24 інформативних атрибута, вирішальним є атрибут Д – вибір управляючого впливу. Атрибут K є допоміжним, на підставі значень цього параметру – критичного часу прийняття рішень – будемо здійснювати вибірку даних з таблиці.

На першому кроці побудови темпорального дерева рішень треба вибрати ситуації, що вимагають найбільш швидкої реакції. У прикладі такі ситуації визначаються значенням K = 2. Відповідно до цього виберемо з вихідної таблиці атрибути з тимчасовими мітками t = 0, 1, 2. Отримана таблиця буде містити 9 спостережень ($s1$ (0), $s1$ (1), $s1$ (2), $s2$ (0), $s2$ (1), $s2$ (2), $s3$ (0), $s3$ (1), $s3$ (2)); пошук серед них найбільш інформативного для розміщення його в корені дерева рішень проводиться на підставі обчислення оцінок приросту інформативності.

У корені дерева розміщується атрибут $s2(1)$, який був обраний як найбільш інформативний. Чотири дуги, зважені значеннями h, n, v, l, ведуть до вершин наступного рівня. З цих вершин одна є кінцевою (зважена дією c), в інших випадках потрібні додаткові перевірки умов. Для формування гілок в вершині, пов'язаної з коренем дугою v, необхідно провести нову вибірку з таблиці даних. У разі $s2(1) = v$ для ситуацій Сит1 і Сит6 потрібні різні керуючі дії: a і d, при цьому в обох ситуаціях $K = 5$. Відповідно до цього включаємо в вибірку атрибути зі значеннями t від 0 до 5 (за винятком атрибута $s2(1)$), і проводимо пошук найбільш інформативного атрибута в таблиці, що містить 2 рядки (Сит1, Сит6) і 17 атрибутів.

Як показано на рисунку 3.3, для розміщення в вузлі другого рівня був обраний найбільш інформативний атрибут $s1(3)$, розгалуження по решті вершин виконується аналогічно. Вершина стає листом, якщо з нею пов'язана єдина відновлювальна дія, або розглянуті всі інформативні атрибути.

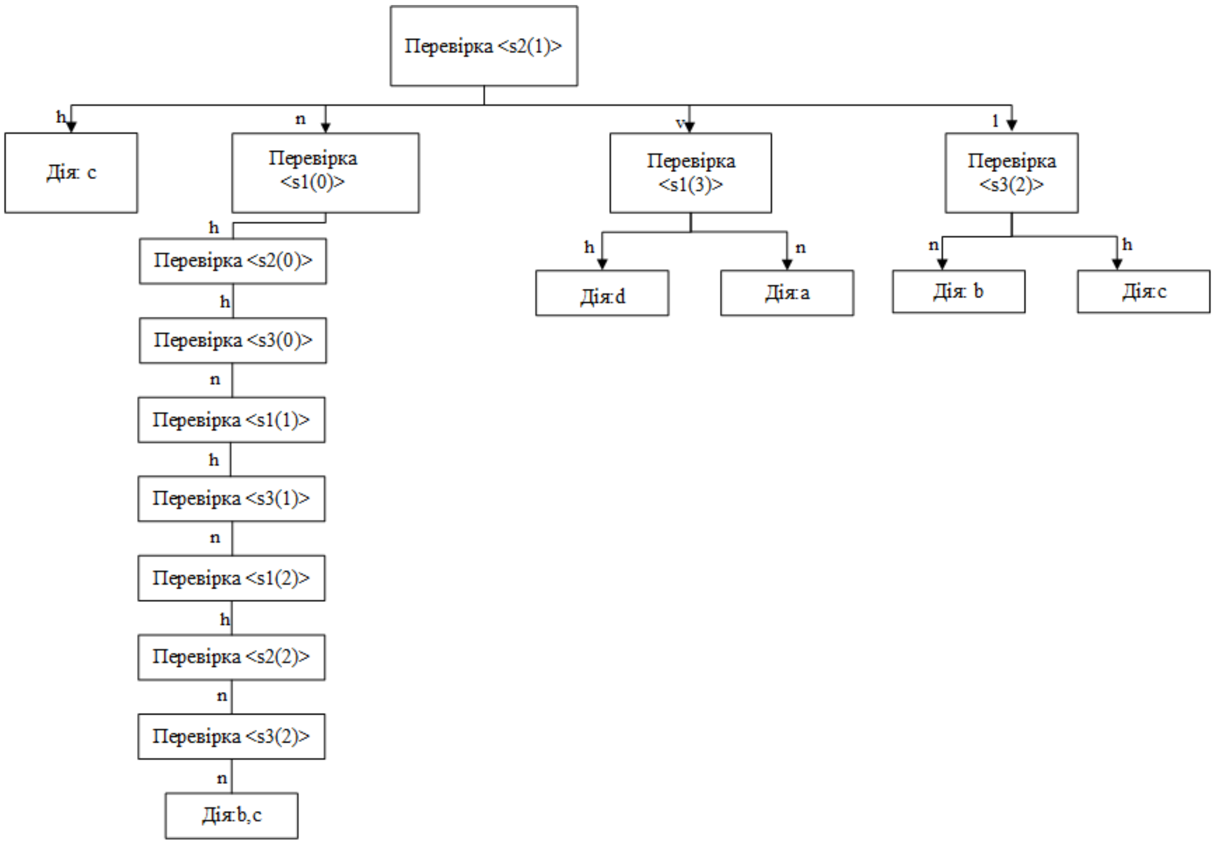


Рисунок 3.3 – Дерево рішень, побудоване з використанням алгоритму «Темпоральний ID3»

Метою побудови темпоральних дерев рішень є їх використання для діагностики складного технічного об'єкта. Далі буде наведено опис процесу діагностики з використанням темпоральних дерев рішень, проведено моделювання процесу діагностики.

3.4 Моделювання процесу діагностики

Експеримент з моделювання процесу діагностики розділено на дві частини. Першу частину названо «апостеріорної діагностикою»: для ситуацій відомі всі значення датчиків на даному часовому інтервалі. ця частина експерименту відповідає тому, що зі складного технічного об'єкта, який працював протягом деякого часу, зняли показання датчиків. Показання можна використовувати, щоб апостеріорно, тобто «після досвіду», перевірити, як побудовані темпоральні дерева рішень зможуть визначити коректність або некоректність ситуацій.

Другу частину експерименту назвемо діагностикою в псевдореальному часі. Ця частина експерименту відповідає тому, що на об'єкті встановлена деяка діагностична система, яка використовує темпоральні дерева рішень. На вхід цієї системи послідовно надходять показники датчиків, встановлених на об'єкті. Діагностична система, яка використовує темпоральні дерева рішень, повинна «на льоту» обробляти показники датчиків, визначати некоректні ситуації і, в разі, якщо вказані дії, що управляють, які дозволять перевести складний технічний об'єкт з стан «несправне» в стан «норма», вказати на необхідність їх виконання.

3.4.1 Апостеріорна діагностика

Для проведення апостеріорної діагностики (тобто такої діагностики, коли вже відомі значення датчиків на всьому розглянутому часовому інтервалі) досить одного темпорального дерева рішень. При апостеріорній діагностиці відома множина ситуацій, темпоральне дерево рішень має визначити некоректні ситуації і видати рекомендації по керуючому впливу, яке могло б перекласти складний технічний об'єкт зі стану «несправне» в стан «норма». Ситуації задаються такою ж таблицею спостережень, як і при побудові темпорального дерева рішень.

У нашій задачі процес діагностики в псевдореальному часі можна розглядати як обробку послідовно надходящих значень датчиків системою діагностики та виявлення некоректної поведінки на основі спочатку заданих ситуацій.

При проведенні діагностики в псевдореальному часі, коли значення датчиків для наступних тимчасових відліків ще невідомі, одного дерева рішень недостатньо.

При побудові темпорального дерева рішень одним з параметрів була величина тимчасового інтервалу для ситуацій r. Тому для проведення діагностики в псевдореальному часі буде потрібно r робочих агентів, кожен з яких буде використовувати темпоральне дерево рішень для визначення некоректних ситуацій. Робота цих агентів організована таким чином: перший агент починає роботу в момент часу $t = 0$, другий – в момент часу $t = 1$, і т. д. Останній, $(r–1)$ агент починає роботу в момент часу $t = r–1$. У кожен момент часу робочий агент володіє такою інформацією:

- поточна локальна тимчасова мітка: момент часу з інтервалу $[0, t^{*}– 1]$;
- поточна вершина дерева.

Отримуючи показання датчиків, агент обробляє їх і збільшує локальну тимчасову мітку. Якщо локальна тимчасова мітка перевищує $t^{*}–1$, то вона скидається в 0, поточним вузлом дерева стає корінь дерева і діагностика для робочого агента починається заново. Крім того, для організації спільної роботи цих агентів потрібен агент-координатор, який буде отримувати інформацію з датчиків, розсилати її робочим агентам і отримувати від них відомості про некоректну роботу об'єкта або системи.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ І РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

4.1 Опис реалізованого програмного комплексу

З метою перевірки ефективності запропонованих в роботі методів був спроектований і розроблений програмний комплекс, що працює в операційній системі Windows. Архітектура програмного комплексу представлено на рисунку 4.1.

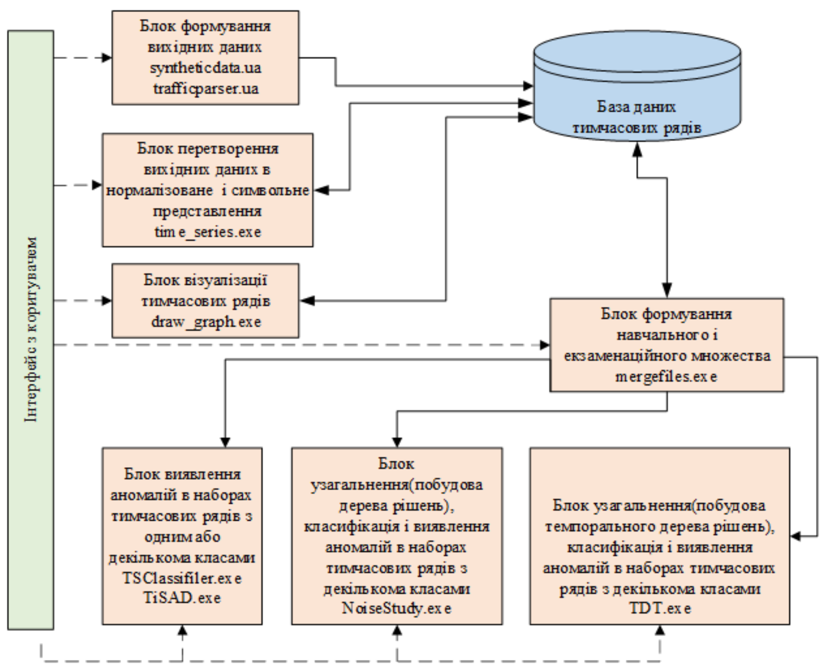


Рисунок 4.1 – Архітектура програмного комплексу

Програмний комплекс складається з:

- набору утиліт [23] для генерації, попередньої обробки та візуального відображення даних;
- додатки Noise Study – Вивчення шуму;
- додатки Time Series Anomaly Detection (TiSAD) – Виявлення аномалій в наборах тимчасових;
- додатки TSClassifier (Time series classifier) – консольної версії TiSAD;

- додатки Temporal Decision Trees (TDT) – Темпоральні дерева рішень.
- Набір утиліт дозволяє:
- згенерувати набори даних «циліндр-дзвін-воронка», «контрольні карти» (syntheticdata.ua) з різними параметрами;
- виділити з наданих для експерименту логів сервера дані для набору «трафік» trafficparser.ua;
- перетворити дані з репозиторіїв UC Irvine Repository і UCR Time Серіес Classification Archive в використовуваний в програмному комплексі формат;
- на підставі заданого розміру часового ряду отримати нормалізоване подання для кожного часового ряду із зазначеного набору даних (time_series.exe);
- на підставі заданого розміру часового ряду і розміру алфавіту отримати символьне уявлення для заданих тимчасових рядів за допомогою алгоритму Symbolic Aggregate approXimation [58] (time_series.exe);
- побудувати графіки вихідних і нормалізованих тимчасових рядів (drawgraph.exe);
- сформувати навчальну та екзаменаційну вибірки для розглянутих в роботі алгоритмів (mergefiles.exe, createStudySets.ua).

Додаток Noise Study–Вивчення шуму дозволяє:

- будувати дерева рішень для наборів тимчасових рядів з декількома класами в символьному поданні;
- проводити класифікацію тимчасових рядів;
- на підставі набору тимчасових рядів – навчальної множини і набору тимчасових рядів – екзаменаційної множини вирішувати завдання виявлення аномалій в наборах тимчасових рядів з декількома класами з використанням дерев рішень.

Додатки $TiSAD$ і $TSClassifier$ дозволяють:

- на підставі набору тимчасових рядів – навчальної множини і набору тимчасових рядів – екзаменаційної множини виявити аномалії серед тимчасових рядів:

- з використанням алгоритму «TS-ADEEP» для навчальної множини з одним класом;
- з використанням алгоритму «TS-ADEEP-Multi» для випадку навчальної множини з декількома класами.

Додаток TDT дозволяє:

- на підставі набору ситуацій, представлених у вигляді табл. 3.8, будувати темпоральні дерева рішень з використанням алгоритму «CPD» і запропонованого в роботі алгоритму «Темпоральний ID3»;
- моделювати апостеріорну діагностику (класифікувати ситуації, що виникли на об'єкті);
- моделювати діагностику в псевдореальному часі.

Приклади роботи програмного комплексу наведені в додатку.

Програми для попередньої обробки даних реалізовані за допомогою мови python, розширення py2exe, bat-файлів. Додаток Noise Study – Вивчення шуму реалізовано на мові програмування C # [24] в середовищі Microsoft Visual Studio Express 2010. Додатки $TiSAD$ і $TSClassifier$ реалізовані на мові програмування C++ [20] в середовищі Microsoft Visual Studio Express 2010 року. Додаток TDT реалізовано на мові C # [24] в середовищі Microsoft Visual Studio Express 2010 року.

4.2 Результати виявлення аномалій для навчальної множини з одним класом

4.2.1 Алгоритм «TS-ADEEP»

Для того щоб визначити, наскільки добре запропонований алгоритм справляється з виявленням аномалій в наборах тимчасових рядів, було проведено його програмне моделювання.

При моделюванні на етапі попередньої обробки даних можна, по-перше, знизити розмірність тимчасових рядів: або вказати «коефіцієнт стиснення» для тимчасових рядів, що дозволяло скоротити розмірність в задане число раз, або задати бажаний новий розмір тимчасового ряду. Далі, при переході до символьного представлення, можна задати бажаний розмір алфавіту. Розмір тимчасового ряду і розмір використовуваного алфавіту впливали на точність виявлення аномалій, що в тому числі було предметом досліджень.

Розглянемо процес моделювання на прикладі набору даних «циліндр-дзвін-воронка». Спочатку в якості навчальної множини TS_STUDY генерується набір тимчасових рядів, що належать до першого з класів, «циліндр». В якості екзаменаційної множини TS_TEST генеруються тимчасові ряди, що належать всім трьом класам – «циліндр», «дзвін», «воронка». Часовий ряд ts_testj є «нормальним», якщо він належить класу «циліндр» і «аномалією», якщо належить класу «дзвін» або «воронка». Відповідно, алгоритм коректно знаходить аномалії, якщо він відносить тимчасові ряди класу «дзвін» і «воронка» з TS_TEST до аномалій, а тимчасові ряди класу «циліндр» аномаліями не вважає. При цьому були розглянуті як чисельне представлення тимчасових рядів, так і символьне з різним розміром алфавіту. Аналогічно моделювання проводилося для класів «дзвін» і «воронка».

У таблиці 4.1, представлені результати виявлення аномалій з використанням алгоритму $TS–ADEEP$. Аномаліями є тимчасові ряди тих класів, які не ввійшли в навчальну вибірку.

Таблиця 4.1 – Точність виявлення аномалій для різних наборів даних. Символьне уявлення. Алгоритм $TS – ADEEP$

Клас рядів в навчальній множині	Точність виявлення аномалій %	
	без шума	з шумом
«трафік»		
normal	—	100.00
«циліндр-дзвін-воронка»		
bell	67.00	70.67

Продовження таблиці 4.1

cylinder	67.00	76.00
funnel	89.00	90.00
Середнє	74.33	78.89
«контрольні карти»		
cyclic	93.33	93.50
dec	83.33	99.67
downw	99.67	98.67
inc	83.33	98.83
norm	83.33	97.67
upw	98.83	99.83
Середнє	90.30	98.03
≪beef≫		
1	—	90.00
2	—	80.00
3	—	80.00
4	—	76.67
5	—	80.00
Середнє	—	81.33
≪coffee≫		
0	—	85,71
1	—	78,57
Середнє	—	82,14
≪Face(four)≫		
1	—	87.50
2	—	85.23
3	—	92.05
4	—	89.77
Середнє	—	88.64
≪Olive oil≫		
1	—	83.33
2	—	70.00
3	—	86.67
4	—	86.67
Середнє	—	81.67

Алгоритм TS – ADEEP справляється із завданням виявлення аномалій в наборах тимчасових рядів з одним класом: показані високі результати (близькі до 100%) на використаних в роботі реальних даних (аналіз трафіку).

Для перевірки роботи алгоритму також використовувалися штучні дані: на наборі часових рядів «циліндр-дзвін-воронка» вибір оптимальних параметрів уявлення тимчасових рядів дозволяє досягти точності до 90% для класу funnel в навчальній множині, 76% для класу cylinder, 70% для класу bell. Для набору контрольні карти вибір оптимальних параметрів уявлення тимчасових рядів дозволяє досягти точності до 99.83% правильно визначених аномалій на деяких наборах даних. У більшості випадків точність виявлення аномалій для «зашумлених» даних вище (в середньому 91.65%), ніж для даних без шуму (в середньому 84.98%).

Щоб оцінити ефективність алгоритму $TS - ADEEP$, можна виходити з такого припущення: виявлення аномалій за допомогою алгоритму є по суті віднесенням розглянутих об'єктів до одного з класів «нормальний» або «аномальний», при цьому, з одного боку, завдання полегшується тим, що не потрібно в точності визначити, до якого з «нормальних» або «аномальних» класів (якщо таких декілька) відноситься об'єкт. З іншого боку, цей же факт ускладнює завдання тим, що при наявності декількох «нормальних» або «аномальних» класів цим неможливо скористатися, так як алгоритм призначений для виявлення аномалій в наборах з єдиним класом. Таким чином, порівняння точності виявлення аномалій з точністю класифікації на таких же наборах даних може в деякому наближенні дозволити оцінити ефективність алгоритму.

Для порівняння будемо використовувати класичні алгоритми:

- метод K найближчих сусідів (Knn);
- алгоритм C4_5;
- байєсовські мережі
- багатошаровий персептрон, логістична регресія (MLP);
- алгоритм Random Forest (RF);
- логістична регресія + дерева рішень (LMT);

Порівняння результатів для запропонованого алгоритму «TS-ADEEP» наведено в таблиці 4.2.

Таблиця 4.2 – Точність класифікації тимчасових рядів класичними алгоритмами [18] і точність виявлення аномалій в наборах тимчасових рядів з одним класом алгоритмом «TS-ADEEP»

	Knn	NB	C4_5	MLP	RF	LMT	SVM	TS-ADEEP середнє
Coffee	75,00	67,86	57,14	96,43	75,00	100,00	96,43	82,14
CBF	85,00	89,67	67,33	85,33	83,56	77,00	87,67	78,89
Olive oil	76,67	76,67	73,33	86,67	86,67	83,33	86,67	81,67
CC	88,00	96,00	81,00	91,33	86,00	92,00	92,33	98,03
Beef	60,00	50,00	56,67	73,33	50,00	80,00	66,67	81,33
Середнє	76,93 (5)	76,04 (7)	67,09 (8)	86,61 (1)	76,25 (6)	86,47 (2)	85,95 (3)	81,41 (4)

Як видно з таблиці, на двох з п'яти розглянутих наборах даних запропонований алгоритм «TS–ADEEP» показав результати краще, ніж інші алгоритми.

В середньому на розглянутих наборах даних точність виявлення аномалій за допомогою алгоритму «TS–ADEEP» вище, ніж у 4 з 7 порівнюваних з ним алгоритмів. У зв'язку з цим можна говорити про ефективність алгоритму «TS–ADEEP» виявлення аномалій в наборах тимчасових рядів з одним класом.

4.3 Результати виявлення аномалій для навчальної множини з декількома класами

4.3.1 Алгоритм «TS-ADEEP-Multi»

Для того щоб визначити, наскільки добре запропонований алгоритм справляється з виявленням аномалій в наборах тимчасових рядів, було проведено його програмне моделювання. При цьому були розглянуті як чисельне представлення тимчасових рядів різної розмірності, так і символьне з різним розміром алфавіту.

Для тимчасових рядів «циліндр-дзвін-воронка» і «трафік» в якості навчальної множини використовувалися різні можливі комбінації з двох класів. Для тимчасових рядів «контрольні карти» розглядалися всі можливі комбінації з двох, трьох, чотирьох і п'яти класів.

У таблиці 4.3 представлені результати виявлення аномалій з використанням алгоритму $TS\text{–}ADEEP\text{–}Multi$ для деяких наборів даних. У дужках для алгоритму $TS\text{–}ADEEP\text{–}Multi$ вказано число класів у навчальній множині. Аномаліями є тимчасові ряди тих класів, які не ввійшли в навчальну вибірку.

Таблиця 4.3 – Точність виявлення аномалій для різних наборів даних. Символьне уявлення. Алгоритм $TS\text{–}ADEEP\text{–}Multi$

Клас рядів в навчальній множині	Точність виявлення аномалій %	
	без шума	з шумом
«циліндр-дзвін-воронка»(2)		
cylinder, funnel	88,33	85,33
bell, funnel	69,00	69,00
Середнє	77,89	77,89
«трафік» (2)		
normal, normal_1	—	100
«контрольні карти» (2)		
cyclic, downw	92,33	91,17
dec, downw	100,00	100,00
Середнє	84,27	92,33

Продовження таблиці 4.3

«контрольні карти» (3)		
downw, norm, upw	80,33	84,67
inc, norm, upw	94,33	92,50
Середнє	83,36	91,48
«контрольні карти» (4)		
dec, inc, norm, upw	77,67	80,83
downw, inc, norm, upw	81,83	82,33
Середнє	83,81	89,97
«контрольні карти» (5)		
cyclic, downw, inc, norm, upw	9,.50	96,17
dec, downw, inc, norm, upw	92,67	91,33
Середнє	90,08	92,19
FaceFour (2)		
1,2	—	72,73
1,3	—	72,33
Середнє	—	72,44
FaceFour (3)		
1,2,3	—	76,14
2,3,4	—	86,36
Середнє	—	82,39

Алгоритм TS–ADEEP–Multi справляється із завданням виявлення аномалій в наборах тимчасових рядів з декількома класами. Показані високі результати (близькі до 100%) на використаних в роботі реальних даних (аналіз трафіку) і в деяких експериментах для наборів даних «контрольні карти». Для інших наборів даних точність виявлення аномалій в середньому становить від

77.89 до 98.50%, що свідчить про ефективність алгоритму TS–ADEEP–Multi.

На зашумлених даних точність виявлення аномалій в середньому вище, максимальна різниця становить 8.12%. Оскільки реальні дані в більшості випадків містять шум, це є перевагою запропонованого в роботі алгоритму.

Для оцінки ефективності алгоритму $TS–ADEEP–Multi$, можна виходити з тих же припущень, що і з алгоритмом $TS–ADEEP$. Виявлення аномалій за допомогою алгоритму є по суті віднесенням розглянутих об'єктів до одного з «нормальних» або «аномальних» класів, при цьому не всі з «нормальних» або «аномальних» класів відомі заздалегідь. Це з одного боку, полегшує завдання тим, що поділ потрібно провести на менше число класів, ніж їх існує в розглянутих наборах даних, з іншого боку, цей же факт ускладнює завдання тим, що неможливо скористатися інформацією, що стосується всіх класів розглянутої предметної області.

Оцінка ефективності алгоритму $TS–ADEEP–Multi$ приведена в таблиці 4.4.

Таблиця 4.4 – Точність класифікації тимчасових рядів класичними алгоритмами [18] і точність виявлення аномалій в наборах тимчасових рядів з декількома класами алгоритмом «TS–ADEEP–Multi»

	Knn	NB	C4_5	MLP	RF	LMT	SVM	TS-ADEEP середнє
CBF	85,00	89,67	67,33	85,33	83,56	77,00	87,67	77,89
CC	88,00	96,00	81,00	91,33	86,00	92,00	92,33	91,49
Face(Four)	87,50	84,09	71,59	87,50	78,41	77,27	88,64	80,40
Середнє	86,83 (4)	89,92 (1)	73,31 (8)	88,05 (3)	82,66 (6)	82,09 (7)	89,55 (2)	83,26 (5)

Алгоритм «TS–ADEEP–Multi» виявлення аномалій в наборах тимчасових рядів з декількома класами показує задовільні результати в порівнянні з класичними алгоритмами класифікації, що свідчить про його ефективність.

4.4 Результати моделювання процесу діагностики з використанням темпоральних дерев рішень

4.4.1 Окремий випадок

Першим етапом експерименту по дослідженню результатів класифікації динамічних об'єктів узагальнення була оцінка точності класифікації тимчасових рядів. Тимчасові ряди як окремий випадок динамічних об'єктів узагальнення були детально розглянуті в другому розділі.

За допомогою розробленого програмного комплексу було проведено ряд експериментів по класифікації тимчасових рядів з використанням апарату темпоральних дерев рішень. Використовувалися два алгоритму побудови темпоральних дерев рішень: «CPD» і запропонований в роботі алгоритм «Темпоральний ID3». Порівняння запропонованого алгоритму проводилося як з класичними алгоритмами класифікації:

- метод K найближчих сусідів (Knn);
- алгоритм C4_5;
- байєсовські мережі (NB);
- багатошаровий персептрон, логістична регресія (MLP);
- алгоритм Random Forest (RF);
- логістична регресія + дерева рішень (LMT);
- метод опорних векторів (SVM).

Так і зі спеціалізованими алгоритмами, створеними для роботи з тимчасовими рядами:

- метод найближчого сусіда (1-NN ED);

- 1–NN Best Warping Window Dynamic Time Warping (r) (1–NN BWW ДТВ (r)) [120];
- 1–NN Dynamic Time Warping, no Warping Window (1–NN DTW no WW) [120].

Метод найближчого сусіда вважаємо спеціалізованим, так як він дозволяє без будь-якої попередньої обробки даних працювати з тимчасовими рядами як з векторами в N–вимірному еквлідовому просторі.

Результати моделювання та їх порівняння зі спеціалізованими алгоритмами класифікації тимчасових рядів [24] наведені в таблиці 4.5.

Таблиця 4.5 – Точність класифікації динамічних об'єктів (%). Приватний випадок, динамічний об'єкт узагальнення – часовий ряд. Порівняння зі спеціалізованими алгоритмами

	1–NN ED	1–NNBWWW DTW (r)	1–NN DTW, no WW	TID3
Wafer	99,50	99,50	98,00	98,64
Coffee	100,00	100,00	100,00	96,43
CBF	85,20	99,60	99,70	95,67
Olive oil	86,60	86,60	83,80	93,30
Trace	76,00	99,00	100,00	88,00
CC	88,00	98,30	99,30	83,33
ECG200	88,00	88,00	77,00	79,00
Lightning2	75,40	86,90	86,90	77,05
Yoga	83,00	84,50	83,60	69,56
Lightning7	57,50	71,20	72,60	65,75
Beef	66,60	66,60	63,30	60,00
Середнє	82,35 (4)	89,11 (1)	87,61 (2)	82,43 (3)

Порівняння показує, що на розглянутих наборах даних точність класифікації з використанням алгоритму Темпоральний ID3 в середньому на 5.18-6.68% нижче, ніж точність класифікації спеціалізованими алгоритмами, створеними для роботи з тимчасовими рядами, але трохи вище (на 0.07%), ніж точність класифікації з використанням методу найближчого сусіда. Проте на одному з наборів даних «Olive oil» – алгоритм «Темпоральний ID3» показав точність класифікації вище, ніж розглянуті спеціалізовані алгоритми.

Результати моделювання та їх порівняння з класичними алгоритмами [18] наведено в таблиці 4.6.

Таблиця 4.6 – Точність класифікації динамічних об'єктів (%). Окремий випадок, динамічний об'єкт узагальнення – часовий ряд. Порівняння з класичними алгоритмами.

	Knn	NB	C4_5	MLP	RF	LMT	SVM	CPD	TID3
Wafer	99,40	70,83	98,20	96,28	99,32	98,09	95,96	97,12	98,64
Coffee	75,00	67,86	57,14	96,43	75,00	100,00	96,43	96,43	96,43
CBF	85,00	89,67	67,33	85,33	83,56	77,00	87,67	92,55	95,67
Olive oil	76,67	76,67	73,33	86,67	86,67	83,33	86,67	56,67	93,30
Trace	82,00	80,00	74,00	77,00	81,00	76,00	73,00	83,00	88,00
CC	88,00	96,00	81,00	91,33	86,00	92,00	92,33	60,67	83,33
ECG200	89,00	77,00	72,00	84,00	81,00	82,00	81,00	73,00	79,00
Lightning2	80,33	67,21	62,30	73,77	78,69	63,93	72,13	75,41	77,05
Yoga	83,30	54,23	69,90	74,50	77,87	71,87	63,07	58,76	69,56
Lightning7	63,01	64,38	54,79	64,38	56,16	64,38	71,23	47,95	65,75
Beef	60,00	50,00	56,67	73,33	50,00	80,00	66,67	46,67	60,00
Середнє	80,16 (5)	72,17 (7)	69,67 (9)	82,09 (2)	77,75 (6)	80,78 (3)	80,56 (4)	71,66 (8)	82,43 (3)

На використаних наборах даних запропонований алгоритм «Темпоральний ID3» показує точність класифікації тимчасових рядів в середньому на 0.34-12.76% відсотків вище, ніж класичні алгоритми класифікації. При цьому на трьох наборах даних CBF, Olive oil, Trace – «Темпоральний ID3» п точності перевершує всі порівнювані алгоритми.

Також запропонований в роботі алгоритм «Темпоральний ID3» показує кращі результати, ніж найбільш близький до нього алгоритм «CPD» [22].

4.4.2 Загальний випадок

У загальному випадку, як описано в третьому розділі, динамічний об'єкт узагальнення являє собою набір тимчасових рядів. За допомогою розробленого програмного комплексу було проведено ряд експериментів за класифікацією таких об'єктів (ситуацій) для перевірки припущення про те, що віднесення ситуацій до різних класів можна провести успішніше, якщо для опису такої ситуації використовується кілька тимчасових рядів. Для порівняння використовувалися два алгоритми, що допускають роботу з динамічними об'єктами узагальнення, що містять кілька параметрів: «CPD» і «Темпоральний ID3». Перевірка проводилася на відповідних наборах даних, описаних раніше «ECG», «wafer», «Activities of daily living», а також на спеціально сформованих навчальних і екзаменаційних вибірках, складених з тимчасових рядів, що відносяться до наборів даних «циліндр-дзвін-воронка», «контрольні карти».

Спочатку були розглянуті випадки, коли всі динамічні об'єкти узагальнення відносяться точно до двох класів. За допомогою алгоритмів «CPD» і «Темпоральний ID3» ми досліджуємо, наскільки успішно можна розрізняти такі об'єкти. У навчальній вибірці «ECG» кожен об'єкт описаний двома ознаками (двома часовими рядами), «Activities of daily living» трьома ознаками, «wafer» – шістьма ознаками.

Результати класифікації для набору даних «Activities of daily living» представлені в таблиці 4.7, для набору даних «ECG» в таблиці 4.8 і на графіку рисунку 4.2, для набору даних «wafer» в таблиці 4.9 і на графіку рисунку 4.3.

Таблиця 4.7 – Точність класифікації (%), набір даних Activities of daily living. Класифікація по одному іількома ознаками. CPD – алгоритм «CPD» [22], TID3 – алгоритм «Темпоральний ID3»

Число ознак	Число класів	Алгоритм	
		CPD	TID3
Класи sitdown_chair, standup_chair навчальна множина - 20% вихідного набору даних			
1 (ось X)	2	99.38	99.38
1 (ось Y)	2	56.17	61.72
1 (ось Z)	2	97.53	97.53
Середнє	2	84.36	86.21
3	2	99.38	99.38
Класи sitdown_chair, standup_chair навчальна множина - 50% вихідного набору даних			
1 (ось X)	2	99,01	99,01
1 (ось Y)	2	58,42	59,41
1 (ось Z)	2	98,02	98,02
Середнє	2	85,15	85,48
3	2	99,01	99,01
Класи getup_bed, liedown_bed			
1 (ось X)	2	97,03	97,03
1 (ось Y)	2	67,33	70,30
1 (ось Z)	2	91,09	93,07
Середнє	2	85,15	86,80
3	2	97,03	97,03

Аналіз результатів дозволяє зробити висновок, що запропонований в роботі алгоритм «Темпоральний ID3» для випадку, коли ситуації описуються кількома тимчасовими рядами, показує результати класифікації краще, ніж «CPD»: на 3% для набору даних «ECG », на 0.49 – 2.01% для набору даних« wafer». На наборі даних «Activities of daily living» результати обох алгоритмів однакові. Також варто відзначити, що класифікація з використанням декількох параметрів в більшості випадків більш точна, ніж класифікація в середньому по одному параметру.

Таблиця 4.8 – Точність класифікації (%) - набір даних ECG. Класифікація по одному і кількома ознаками. CPD – алгоритм «CPD» [22]; TID3 – алгоритм «Темпоральний ID3»

Число ознак	Число класів	Алгоритм	
		CPD	TID3
1 (1-ий)	2	71,00	70,00
1 (2-й)	2	75,00	76,00
Середнє	2	73,00	73,00
2	2	72,00	75,00

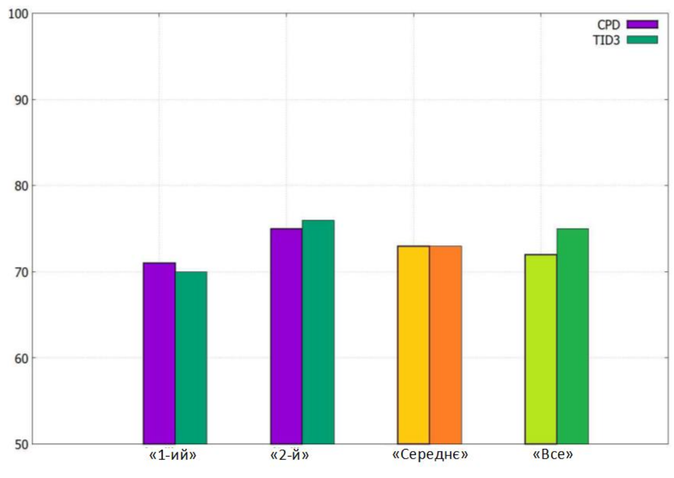


Рисунок 4.2 – Точність класифікації (%) – набір даних ECG. Класифікація по одному і кількома ознаками. CPD – алгоритм «CPD» [22]; TID3 – алгоритм «Темпоральний ID3»

Однак в розглянутих наборах даних зазвичай присутній один з параметрів, який був найбільш інформативним, в зв'язку з чим використання інших параметрів для класифікації було надмірним, а іноді призводило до зменшення точності класифікації. У разі ж, якщо такого параметра немає або він в ході вивчення предметної області ще не виявлений, рекомендується використовувати всі доступні параметри, так як це дозволяє отримати точність класифікації в середньому більшу, ніж використання якого-небудь одного параметра.

Таблиця 4.9 – Точність класифікації (%) – набір даних wafer. Класифікація по одному і кількома ознаками. CPD – алгоритм «CPD» [22]; TID3 – алгоритм «Темпоральний ID3»

Число ознак	Число класів	Алгоритм	
		CPD	TID3
1 (1-ий)	2	81,13	88,17
1 (2-й)	2	84,26	86,90
1 (3-й)	2	77,42	83,19
1 (4-й)	2	94,03	99,02
1 (5-й)	2	87,78	87,98
1 (6-й)	2	91,10	92,77
Середнє	2	85,95	89,67
2	2	69,09	96,58
6	2	92,47	96,48

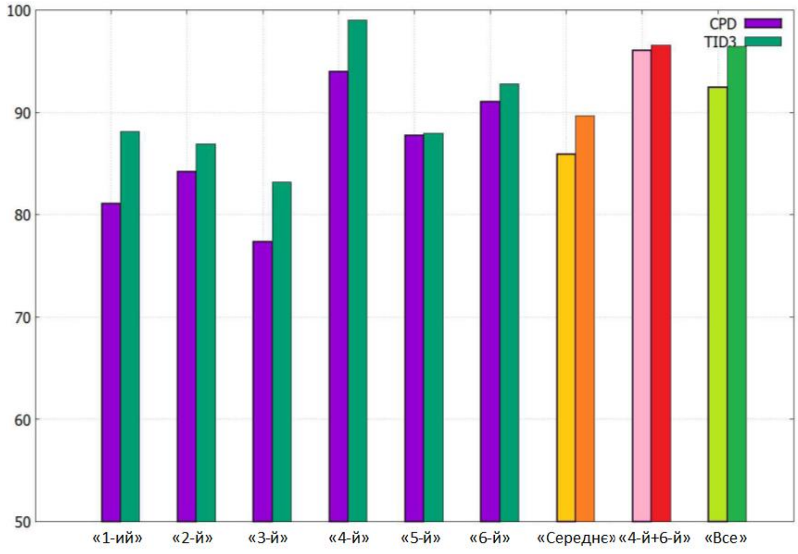


Рисунок 4.3 – Точність класифікації (%) – набір даних wafer. Класифікація по одному і кількома ознаками. CPD – алгоритм «CPD» [22]; TID3 – алгоритм «Темпоральний ID3»

У третьому розділі був описаний принцип, за яким з тимчасових рядів формувалися динамічні об'єкти узагальнення, представлені кількома часовими рядами (таблиці 3.3, 3.4). За таким же принципом були сформовані інші навчальні та екзаменаційні вибірки з наборів даних «циліндр-дзвін-воронка» і «контрольні карти». Результати для цієї частини експерименту представлені в таблиці 4.10.

Таблиця 4.10 – Точність класифікації динамічних об'єктів (%). Набори даних «циліндр-коколол-воронка» (CBF) і «контрольні карти» (CC). Загальний випадок. Кілька ознак. CPD - алгоритм «CPD» [22]; TID3 - алгоритм «Темпоральний ID3»

Число даних	Число ознак	Число класів	Алгоритм	
			CPD	TID3
CBF	2	3	85,00	89,00
CBF	2	9	58,11	71,44

Продовження таблиці 4.10

CC	2	6	97,50	99,00
CC	5	6	98,17	99,00

З таблиці видно, що запропонований в роботі алгоритм «Темпоральний ID3» на розглянутих наборах даних показує більш високу точність класифікації, на 0.83–13.33% вище, ніж алгоритм «CPD». Крім того, використання всіх ознак (тимчасових рядів) з опису ситуацій дійсно дозволяє з високою точністю розділити наявні об'єкти на відповідні класи.

Така ситуація – наявність декількох динамічно змінюющихся параметрів, за допомогою яких необхідно оцінити стан об'єкта управління – найбільш характерна для випадку, коли мова йде про управління складним технічним об'єктом (прикладом такого об'єкта може бути, наприклад, електростанція). При цьому особа, яка приймає рішення (диспетчер), повинен розпізнати ситуацію на складному технічному об'єкті і прийняти рішення про те, що стан об'єкта є нормальним або аномальним. В останньому випадку вкрай корисно віднести реальну ситуацію до певного класу, в залежності від типу несправності. Отже, алгоритми визначення аномалій і класифікації динамічних ситуацій можуть бути дуже корисними при їх використанні в ІСППР РВ.

ВИСНОВКИ

У атестаційній роботі отримані наступні результати:

1. Проведено огляд методів представлення знань в сучасних інтелектуальних системах і розглянута проблема роботи з даними, явно залежать від часу - темпоральними даними. Виділено основні категорії таких даних, які можуть використовуватися в ІСППР реального часу. Введено поняття динамічного об'єкта узагальнення - структури, яка описує динамічний стан складної технічної системи, одним з параметрів якої є час.

2. Розглянуто проблему виявлення аномалій в разі, коли стан складної технічної системи представимо тимчасовим поруч. Дано постановку задачі виявлення аномалій в наборах тимчасових рядів з одним і декількома класами і виконаний огляд існуючих методів вирішення цих завдань.

3. На підставі аналізу підходів до вирішення завдання виявлення аномалій в наборах тимчасових рядів запропоновані методи і розроблені алгоритми «TS-ADEEP», «TS-ADEEP-Multi» виявлення аномалій для наборів тимчасових рядів з одним і декількома класами. Розрахована оцінка обчислювальної складності розроблених алгоритмів.

4. Проведено аналіз різних способів подання знань в інтелектуальних системах. Виділено клас динамічних об'єктів, які представлені кількома часовими рядами. Для даного типу динамічних об'єктів дана постановка завдання узагальнення. Показано, що задача узагальнення для динамічних об'єктів може бути використана для вирішення завдань діагностики станів (ситуацій) в складних динамічних системах. На підставі аналізу підходів до вирішення такого завдання обраний підхід на основі побудови темпоральних дерев рішень і приведено формальний опис темпоральних дерев рішень.

5. Проведено огляд методів і алгоритмів побудови темпоральних дерев рішень. Запропоновано новий алгоритм «Темпоральний ID3» побудови темпоральних дерев рішень, що використовує в якості критерію вибору

спостережень для розбиття величину «приріст інформативності». Отримано оцінку обчислювальної складності алгоритму «Темпоральний ID3» і показано, що вона має поліноміальний характер.

6. Для дослідження можливостей розроблених методів і алгоритмів був спроектований і розроблений програмний комплекс, що дозволяє вирішувати завдання виявлення аномалій для наборів тимчасових рядів з одним і декількома класами; вирішувати завдання узагальнення для динамічних об'єктів, що становлять як тимчасові ряди, так і набори тимчасових рядів.

7. Для алгоритмів TS-ADEEP і TS-ADEEP-Multi показано, що на відомих наборах даних точність виявлення аномалій порівнянна з точністю виявлення аномалій низкою відомих алгоритмів (метод опорних векторів, алгоритм C4_5, байєсовські мережі, алгоритм Random Forest і ін.). Виявлено ряд завдань (наприклад, control chart, beef), для до торих алгоритм TS-ADEEP показує результати, що перевершують результати, показані іншими алгоритмами.

8. Проведено порівняння точності класифікації тимчасових рядів з використанням алгоритму «Темпоральний ID3» з відомими алгоритмами класифікації (метод K найближчих сусідів, C4_5, байєсовські мережі, Random Forest і ін.).

9. Проведено порівняння точності класифікації динамічних об'єктів, представлених наборами тимчасових рядів, з алгоритмом «CPD», найбільш «близьким» до алгоритму «Темпоральний ID3». Показано, що в більшості випадків алгоритм «Темпоральний ID3» перевершує «CPD» (в середньому на 0.83-13.33% для різних наборів даних).

10. Результати експерименту дозволяють зробити висновок про ефективність використання алгоритму «Темпоральний ID3» для роботи з динамічними об'єктами, які представлені наборами тимчасових рядів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Antipov, S.G., Fomina, M.V. (2011). «A method for compiling general concepts with the use of temporal decision trees»:, Scientific and Technical Information Processing, vol. 38, no. 6, pp. 409–419.

2. Vagin, V.N., Fomina, M.V., Antipov, S.G. (2013). «Modeling of algorithms of inductive concept formation in “noisy” databases», Automatic Documentation and Mathematical Linguistics, vol. 47, no. 4. pp. 151–161

3. Вагин В.Н., Фомина М.В., Антипов С.Г. Моделирование алгоритмов индуктивного формирования понятий в «зашумленных» базах данных. Научно-техническая информация, 2013, 7: 20–32.

4. Антипов С. Г. Использование темпоральных деревьев решений для задач диагностики // XII Московская международная телекоммуникационная конференция студентов и молодых ученых "Молодежь и Наука". Тезисы докладов в 2-х частях. –М.: МИФИ, 2009. – Т. 2. С. 138–139.

5. . Антипов С. Г., Фомина М. В. Метод формирования обобщенных понятий с использованием темпоральных деревьев решений // Двенадцатая национальная конференция по искусственному интеллекту с международным участием.– М.: Физматлит, 2010. – Т. 2. – С. 40–46.

6. Антипов С. Г., Вагин В. Н.. Проблема обнаружения аномалий в наборах временных рядов // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием. – К.: 2014,0 – Т. 2. – С. 195–203.

7. Арский Ю.М., Финн В. К. Принципы конструирования интеллектуальных систем // Информационные технологии и вычислительные системы, 2008, (4): 4–37.

8. Башмаков А. И., Башмаков И. А. Интеллектуальные информационные технологии: Учеб. пособие. – М.: Изд. МГТУ им. Н. Э. Баумана, 2005. С. 304.

9. Еремеев А.П., Троицкий В.В. Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // Известия РАН. Теория и системы управления, 2003. – Т. 5. – С. 75–88.

10. Гарратано Д., Райли Г. Экспертные системы: принципы разработки и программирование,. – М.: ООО «И.Д. Вильямс», 2007. С – 1152.

11. Newell A., Herbert A.(1976), «Computer science as empirical inquiry: symbols and search» , Commun. ACM, vol. 19, no. 3, pp. 113–126.

12. Collins A., Quillian, M. (1969) «Retrieval time from semantic memory», Journal of Verbal Learning and Verbal Behavior, vol. 8, pp. 240–248.

13. Minsky M. (1974), «A Framework for Representing Knowledge: Tech». Rep. Cambridge, MA, USA,.

14. Robert M. (1976), «Structure of decision: the cognitive maps of political elites», the University Michigan, Princeton University Press, Princeton, – P. 404.

15. Джордж Ф. Люгер. Искусственный интеллект: стратегии и методы решения сложных проблем. М.: Изда–во «Вильямс», 2003. – 864 с.

16. Colmerauer A., Roussel P. (1996), «History of programming languages», NY, USA: ACM,. – pp. 331–367.

17. . Кандрашина Е. Ю, Литвинцева Л. В., Представление зна–й о времени и пространстве в интеллектуальных системах. М.: Наука, 1989, 328 с.

18. Поспелов Д. А.. Из истории искусственного интеллекта: история искусственного интеллекта до середины 80-х годов. Новости искусственного интеллекта, 1994, 4: 70–90.

19. Roddick F., Spiliopoulou M. (1999), «A bibliography of temporal, spatial and spatio-temporal data mining research», SIGKDD Explor. Newsl, vol. 1, no. 1, pp. 34–38

20. Weiqiang Lin, Mehmet A., Graham Williams J. (2002), «An Overview of Temporal Data Mining», Proceedings of the 1st Australasian Data Mining Workshop.

21. Antunes C. M., Oliveira A. L. (2001), «Temporal data mining: an overview», Eleventh International Workshop on the Principles of Diagnosis.

22. John F. (2002), «A Survey of Temporal Knowledge Discovery Paradigms and Methods», IEEE Transactions on Knowledge and Data Engineering, vol. 14, pp. 750–767.

23. Production system models of learning and development / Ed. by David Klahr, P. Langley, R. Neches. – Cambridge, MA: MIT Press, 1987

24. Брайан Керниган, Роберт Пайк. Unix. Программное окружение. С.: Плюс, 2003, 416 с.