Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Програмні засоби виявлення аномалій у вихідних даних

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІз-21-1
Софія ВОЛОЩУК
(власне ім'я, прізвище)

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

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

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник: ст. викл. В'ячеслав РАДЧЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"_____"_____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Волощук Софії Володимирівни
(прізвище, ім'я, по батькові)

1. Тема роботи Програмні засоби виявлення аномалій у вихідних даних

затверджена наказом по університету від "05" травня 2025 р. № 72 СТз

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи
виявлення аномалій
штучні імунні мережі
розробка комп'ютерної системи
Microsoft Visual Studio

4. Перелік питань, що потрібно опрацювати у роботі
Теоретичні основи виявлення аномалій у даних
Огляд існуючих програмних засобів
Розробка системи виявлення аномалій
Розробка програмних модулів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 11 слайдів

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	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	05.05.2025–10.05.2025	
2	Огляд існуючих аналогів	11.05.2025–03.06.2025	
3	Вибір алгоритмів	04.06.2025–06.06.2025	
4	Вибір програмних засобів	07.06.2025–08.06.2025	
5	Програмна реалізація	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання "05" травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. В'ячеслав РАДЧЕНКО
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 55 с., 10 рис., 1 дод., 10 джерел.

НЕГАТИВНИЙ ВІДБІР, ШТУЧНА ІМУННА СИСТЕМА, МОДЕЛЬ, ПРОГРАМНІ ЗАСОБИ, АНОМАЛІЇ, ВИХІДНІ ДАНІ.

Метою кваліфікаційної роботи є розробка програмних засобів виявлення аномалій в вихідних даних на основі штучних імунних систем.

У ході виконання кваліфікаційної роботи проведено аналіз існуючих засобів виявлення аномалій у вихідних даних, а також аналіз моделей штучних імунних систем. Розроблено модель негативного відбору для вирішення практичної задачі. Визначені основні характеристики моделі негативного відбору. Проведено тестування розроблених програмних засобів.

ABSTRACT

Bachelor’s thesis: 55 pages, 10 figures, 1 appendices, 10 sources.

NEGATIVE SELECTION, ARTIFICIAL IMMUNE SYSTEM, MODEL, SOFTWARE TOOLS, ANOMALIES, OUTPUT DATA.

The major goal of this thesis is the development of software tools for anomaly detection in output data based on artificial immune system principles.

In order to an in-depth analysis of existing anomaly detection methods as well as models inspired by artificial immune systems. A negative selection model was designed to address a practical task, with its key characteristics clearly defined. The developed software tools were subjected to testing to evaluate their performance and reliability.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 7
1 ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ АНОМАЛІЙ У ДАНИХ 10
1.1 Основні поняття та визначення 10
1.2 Класифікація методів виявлення аномалій 11
1.3 Статистичні методи виявлення аномалій 13
1.4 Методи машинного навчання для виявлення аномалій 15
1.5 Методи глибокого навчання 17
1.6 Спеціалізовані методи для різних типів даних 18
2 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ 20
2.1 Бібліотеки Python для виявлення аномалій 20
2.2 Інструменти R для детекції аномалій 22
2.3 Комерційні рішення 23
2.4 Open-source інструменти 24
2.5 Порівняльний аналіз інструментів 25
3 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ 27
3.1 Аналіз вимог до системи 27
3.2 Архітектура системи 29
3.3 Детальний опис модулів 32
3.4 Технології та інструменти розробки 34
3.5 Реалізація ключових алгоритмів 36
4 РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ 41
4.1 Об'єктно орієнтована модель штучної імунної системи 43
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 48
ДОДАТОК А Графічний матеріал кваліфікаційної роботи 49

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС – операційна система

ПЗ – програмні засоби

ПІС – природна імунна система

ШІС – штучна імунна система

ШНМ – штучна нейронна мережа

ВСТУП

Сучасний світ характеризується експоненційним зростанням обсягів даних, що генеруються різноманітними системами, процесами та пристроями. За оцінками провідних аналітичних компаній, щоденно у світі створюється понад 2,5 квінтільйона байтів інформації, причому цей показник продовжує зростати з кожним роком. В умовах такого масштабного накопичення інформації критично важливою стає здатність ефективно аналізувати дані та виявляти в них аномальні патерни, які можуть свідчити про несправності систем, шахрайську діяльність, кібератаки, медичні відхилення або інші критичні ситуації.

Виявлення аномалій у даних представляє собою фундаментальну задачу, яка знаходить широке застосування в численних галузях людської діяльності. У фінансовій сфері автоматичне виявлення підозрілих транзакцій дозволяє мінімізувати втрати від шахрайських операцій, які, за даними міжнародних організацій, щорічно сягають мільярдів доларів. У галузі інформаційної безпеки системи виявлення аномалій відіграють ключову роль у захисті від кіберзагроз, дозволяючи своєчасно ідентифікувати спроби несанкціонованого доступу або атаки на інфраструктуру. Медична сфера також активно використовує методи детекції аномалій для ранньої діагностики захворювань та моніторингу стану пацієнтів.

Традиційні підходи до виявлення аномалій, що базуються на експертній оцінці та мануальному аналізі, виявляються неефективними в умовах сучасних масштабів даних. Людський фактор обмежує як швидкість обробки інформації, так і здатність виявляти складні, багатовимірні патерни аномальної поведінки. Це обумовлює гостру необхідність розробки автоматизованих програмних засобів, здатних в режимі реального часу аналізувати великі обсяги даних та ідентифікувати потенційно аномальні ситуації.

Сучасний розвиток методів машинного навчання та штучного інтелекту відкриває нові можливості для створення досконаліших систем виявлення аномалій. Алгоритми глибокого навчання демонструють вражаючі результати в задачах розпізнавання складних патернів, однак їх адаптація для специфічних потреб детекції аномалій потребує глибокого розуміння як теоретичних основ, так і практичних аспектів їх реалізації.

1 ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ АНОМАЛІЙ У ДАНИХ

1.1 Основні поняття та визначення

Концепція аномалії в контексті аналізу даних не має універсального, загальноприйнятого визначення, що пояснюється багатогранністю цього явища та різноманітністю контекстів його застосування. В найзагальнішому розумінні аномалія представляє собою спостереження або групу спостережень, які суттєво відрізняються від очікуваної норми або домінуючого патерну в даних. Однак таке визначення є надто широким і потребує уточнення залежно від конкретної предметної області та типу аналізованих даних.

Класичне статистичне розуміння аномалії ґрунтується на концепції викиду або відхилення від статистичної норми. В цьому контексті аномальним вважається спостереження, що знаходиться на значній відстані від центральної тенденції розподілу даних, зазвичай визначеної через середнє арифметичне або медіану. Проте такий підхід має обмеження, оскільки не враховує контекстуальну інформацію та складні взаємозв'язки між різними атрибутами даних.

Сучасне розуміння аномалії розширює традиційне статистичне визначення, включаючи поняття контекстуальної та колективної аномальності. Контекстуальна аномалія характеризується тим, що спостереження може бути нормальним в одному контексті, але аномальним в іншому. Наприклад, температура 30 градусів Цельсія є нормальною влітку, але аномальною взимку. Колективна аномалія виникає тоді, коли окремі спостереження можуть бути нормальними, але їх сукупність або послідовність виявляється аномальною.

Важливо розрізняти аномалії, шум та помилки в даних, оскільки ці поняття часто плутають, хоча вони мають різну природу та вимагають різних

підходів до обробки. Шум представляє собою випадкові відхилення в даних, що не несуть корисної інформації та зазвичай є наслідком недосконалості процесу вимірювання або передачі інформації. Помилки в даних виникають через технічні несправності, людський фактор або збої в системах збору інформації. На відміну від шуму та помилок, аномалії можуть нести важливу інформацію про досліджуваний процес або систему.

Типологія аномалій включає кілька основних категорій. Точкові аномалії представляють собою окремі спостереження, що відрізняються від норми. Цей тип аномалій є найпростішим для розуміння та виявлення, оскільки кожне спостереження аналізується незалежно від інших. Контекстуальні аномалії, як уже зазначалося, визначаються залежно від контексту, що робить їх виявлення більш складним завданням. Колективні аномалії виявляються на рівні груп або послідовностей спостережень і часто зустрічаються в аналізі часових рядів або мережевих даних.

1.2 Класифікація методів виявлення аномалій

Різноманітність підходів до виявлення аномалій обумовила необхідність їх систематизації та класифікації за різними критеріями. Найбільш фундаментальною є класифікація за типом навчання, яка відображає спосіб отримання знань алгоритмом про нормальні та аномальні патерни в даних.

Методи з учителем характеризуються наявністю попередньо розміченого навчального набору, що містить приклади як нормальних спостережень, так і різних типів аномалій. Цей підхід дозволяє алгоритму навчитися розрізняти нормальні та аномальні патерни на основі історичних даних. Основною перевагою наглядних методів є їх висока точність в задачах, схожих на ті, на яких проводилося навчання. Однак їх застосування обмежується доступністю якісних розмічених даних, що в практиці аналізу аномалій часто є проблематичним.

Методи без учителя не потребують попередньої розмітки даних і базуються на припущенні, що аномальні спостереження суттєво рідше зустрічаються в даних порівняно з нормальними. Ці методи аналізують статистичні властивості даних, виявляючи спостереження, що не відповідають домінуючим патернам. Незважаючи на свою універсальність, ненаглядні методи можуть давати більшу кількість помилкових спрацьювань, особливо в даних зі складною структурою.

Напівнаглядні методи займають проміжне положення, використовуючи переважно дані про нормальну поведінку системи з мінімальною кількістю прикладів аномалій або взагалі без них. Такий підхід відображає реальну ситуацію, коли нормальні дані легко доступні, а приклади аномалій є рідкісними або дорогими у отриманні.

Альтернативна класифікація методів виявлення аномалій базується на підходах до аналізу даних. Статистичні методи ґрунтуються на математичному моделюванні розподілу даних та використанні статистичних тестів для виявлення відхилень. Ці методи мають міцну теоретичну основу та добре підходять для аналізу одновимірних або простих багатовимірних даних з відомими статистичними властивостями.

Методи машинного навчання використовують алгоритми автоматичного виявлення закономірностей в даних без явного програмування правил детекції. Вони краще справляються зі складними, високорозмірними даними та здатні виявляти нелінійні залежності між змінними. Проте ці методи часто менш інтерпретовані порівняно зі статистичними підходами.

Методи глибокого навчання представляють найсучасніший напрям розвитку технологій детекції аномалій. Вони використовують багатошарові нейронні мережі для автоматичного вивчення складних представлень даних. Ці методи показують найкращі результати на високорозмірних даних, таких як зображення, тексти або аудіо, але потребують значних обчислювальних ресурсів та великих обсягів даних для навчання.

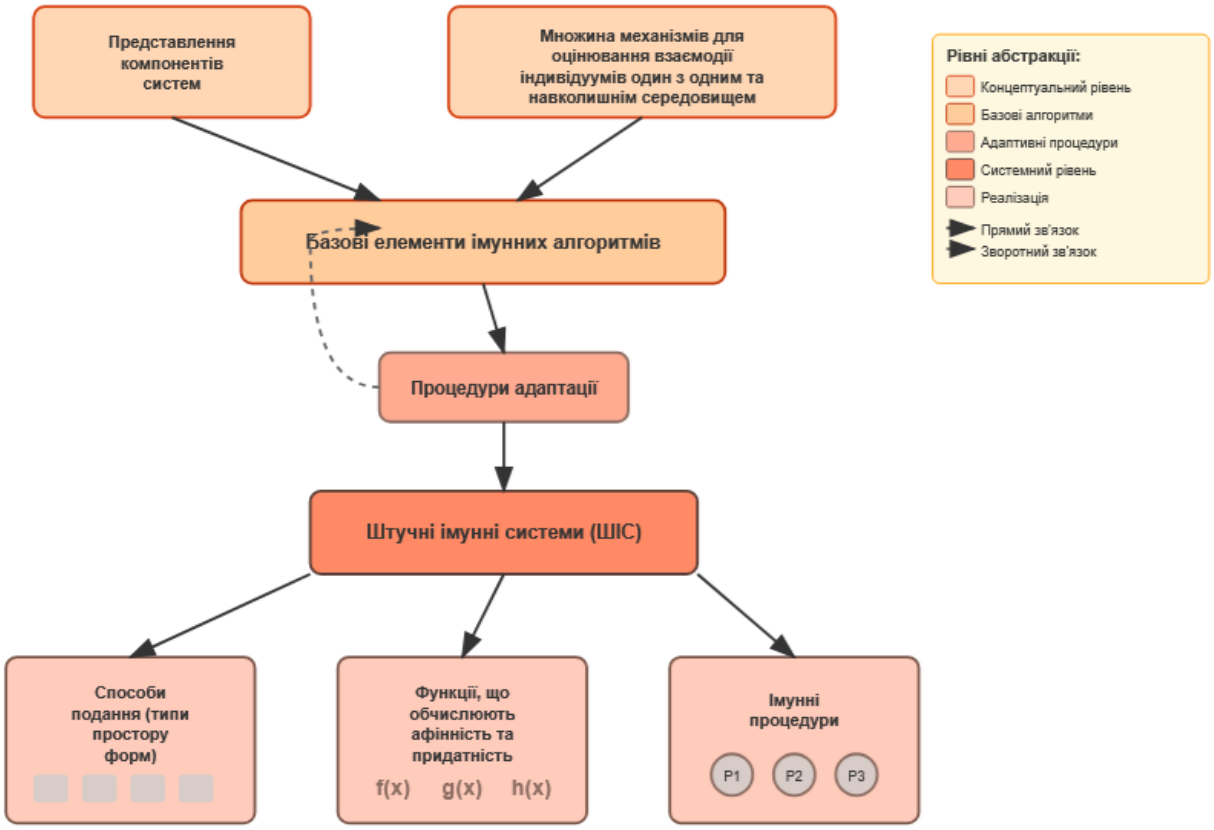

Рисунок 1.1 – Компоненти системи для виявлення аномалій

Ієрархічна структура з п'яти рівнів абстракції від концептуального до реалізації. Концептуальний рівень (верх):

- представлення компонентів систем;
- механізми оцінювання взаємодії з середовищем;
- базові елементи імунних алгоритмів як фундамент системи;
- процедури адаптації - ключовий механізм навчання та пристосування;
- штучні імунні системи - центральний системний рівень.

1.3 Статистичні методи виявлення аномалій

Статистичні методи виявлення аномалій становлять фундаментальну основу всіх підходів до детекції відхилень у даних. Ці методи базуються на припущенні, що дані можна описати певною статистичною моделлю, а

аномалії представляють собою спостереження, що не відповідають цій моделі або мають низьку ймовірність згідно з нею.

Параметричні статистичні методи припускають, що дані підпорядковуються певному відомому розподілу, параметри якого можна оцінити з вибірки. Найпростішим та найпоширенішим є припущення про нормальний розподіл даних. В рамках цього підходу аномаліями вважаються спостереження, що знаходяться на відстані більше певного порогу від середнього значення, виміряної в одиницях стандартного відхилення.

Z-score або стандартизований залишок є класичним інструментом параметричної детекції аномалій. Цей показник обчислюється як відношення відхилення спостереження від середнього арифметичного до стандартного відхилення. Спостереження з абсолютним значенням Z-score більше певного порогу, зазвичай 2 або 3, класифікуються як аномальні. Метод ефективний для одновимірних даних з нормальним або близьким до нормального розподілом, але його застосування обмежується чутливістю до викидів при обчисленні параметрів розподілу.

Модифікований Z-score використовує робастні оцінки центральної тенденції та розсіювання, замінюючи середнє арифметичне медіаною, а стандартне відхилення - медіанним абсолютним відхиленням. Це робить метод менш чутливим до наявності викидів у даних, що є важливою перевагою в практичних застосуваннях.

Критерій Граббса представляє собою формальний статистичний тест для виявлення викидів в одновимірних нормально розподілених даних. Тест базується на відношенні максимального відхилення від середнього до стандартного відхилення і дозволяє з певним рівнем значущості визначити, чи є найбільш відхиляюче спостереження статистично значущим викидом.

Непараметричні статистичні методи не роблять припущень про конкретний вид розподілу даних і базуються на емпіричних характеристиках вибірки. Метод міжквартильного розмаху є одним з найпростіших та найробастніших непараметричних підходів. Він визначає аномалії як

спостереження, що знаходяться за межами діапазону, обмеженого першим квартилем мінус 1.5 міжквартильних розмахи та третім квартилем плюс 1.5 міжквартільних розмахи.

Гістограмний аналіз передбачає побудову емпіричного розподілу даних у вигляді гістограми та виявлення спостережень у бінах з низькою частотою. Цей підхід особливо корисний для візуального аналізу та інтерпретації результатів, але його ефективність залежить від правильного вибору кількості та ширини бінів.

Багатовимірні статистичні методи розширюють одновимірні підходи на випадок кількох змінних, враховуючи кореляційні зв'язки між ними. Відстань Махаланобіса є багатовимірним узагальненням Z-score, що враховує коваріаційну структуру даних. Вона обчислюється як квадратний корінь з квадратичної форми, що включає зворотну коваріаційну матрицю, і дозволяє ідентифікувати спостереження, що є аномальними з точки зору багатовимірного розподілу.

1.4 Методи машинного навчання для виявлення аномалій

Розвиток машинного навчання відкрив нові можливості для детекції аномалій, особливо в складних, високорозмірних даних, де традиційні статистичні методи виявляються недостатньо ефективними. Методи машинного навчання здатні автоматично виявляти складні нелінійні закономірності та адаптуватися до специфічних характеристик конкретних наборів даних.

Методи на основі відстані базуються на припущенні, що аномальні спостереження знаходяться на значній відстані від своїх найближчих сусідів у просторі ознак. Алгоритм k-найближчих сусідів в контексті детекції аномалій обчислює відстань від кожного спостереження до його k найближчих сусідів. Спостереження з найбільшими середніми відстанями до сусідів класифікуються як потенційні аномалії.

Локальний фактор викиду розширює концепцію k-найближчих сусідів, враховуючи локальну щільність точок у просторі ознак. Алгоритм обчислює відношення локальної щільності навколо даного спостереження до локальних щільностей його сусідів. Високі значення цього відношення вказують на те, що спостереження знаходиться в області з нижчою щільністю порівняно з його околицею, що є ознакою аномальності.

Методи кластеризації підходять до детекції аномалій з позиції групування схожих спостережень. Алгоритм k-means може використовуватися для виявлення аномалій шляхом ідентифікації спостережень, що знаходяться на великій відстані від центроїдів найближчих кластерів. Альтернативно, можна вважати аномальними спостереження, що формують дуже малі або слабо пов'язані кластери.

DBSCAN представляє density-based підхід до кластеризації, який природним чином ідентифікує викиди як точки, що не можуть бути віднесені до жодного кластера. Алгоритм визначає кластери як області високої щільності, розділені областями низької щільності. Спостереження в областях низької щільності автоматично класифікуються як аномальні.

Методи на основі щільності оцінюють ймовірнісну щільність розподілу даних та ідентифікують аномалії як спостереження з низькою щільністю ймовірності. Kernel Density Estimation будує непараметричну оцінку щільності розподілу, використовуючи ядрові функції навколо кожного спостереження. Спостереження в областях з низькою оціненою щільністю вважаються аномальними.

Gaussian Mixture Models моделюють розподіл даних як суміш кількох багатовимірних нормальних розподілів. Після навчання моделі спостереження з низькою ймовірністю згідно з навченою сумішшю класифікуються як аномальні. Цей підхід особливо ефективний для даних з кластерною структурою.

Методи зменшення розмірності використовуються для проекції високорозмірних даних на простір меншої розмірності з подальшим аналізом

помилок реконструкції. Principal Component Analysis проектує дані на підпростір головних компонент, що пояснюють максимальну варіацію. Аномалії часто характеризуються великими помилками реконструкції при поверненні з проекційного простору до оригінального.

1.5 Методи глибокого навчання

Глибоке навчання революціонізувало багато областей машинного навчання, включаючи детекцію аномалій. Здатність глибоких нейронних мереж автоматично вивчати складні ієрархічні представлення даних робить їх особливо привабливими для аналізу високорозмірних та структурованих даних, таких як зображення, тексти або часові ряди.

Автоенкодери є однією з найпопулярніших архітектур глибокого навчання для детекції аномалій. Ці мережі складаються з двох частин: енкодера, що стискає вхідні дані до компактного представлення, та декодера, що відновлює оригінальні дані з цього представлення. Навчання відбувається на нормальних даних з метою мінімізації помилки реконструкції. Аномалії виявляються як спостереження з високою помилкою реконструкції, оскільки мережа не навчалася їх відтворювати.

Варіаційні автоенкодери розширюють базову архітектуру автоенкодера, додаючи ймовірнісну інтерпретацію латентного простору. Енкодер виводить не детерміністичне представлення, а параметри розподілу в латентному просторі. Це дозволяє не лише реконструювати дані, але й оцінювати їх ймовірність, що робить детекцію аномалій більш принциповою з ймовірнісної точки зору.

Denoising автоенкодери навчаються відновлювати чисті дані з зашумлених версій. Цей підхід робить мережу більш робастною та здатною краще узагальнювати на нові дані. В контексті детекції аномалій такі автоенкодери часто показують кращу здатність розрізняти нормальні та аномальні патерни.

Рекурентні нейронні мережі, зокрема LSTM та GRU, особливо підходять для аналізу послідовних даних, таких як часові ряди або тексти. Ці архітектури здатні моделювати складні часові залежності та виявляти аномалії в контексті історичної поведінки системи. Навчання зазвичай проводиться на задачі прогнозування наступного елемента послідовності, а аномалії виявляються як моменти з великими помилками прогнозування.

Attention механізми дозволяють мережі фокусуватися на найбільш релевантних частинах вхідних даних при прийнятті рішень. В контексті детекції аномалій це може допомогти ідентифікувати конкретні ознаки або моменти часу, що відповідають за аномальну поведінку, роблячи систему більш інтерпретованою.

Генеративно-змагальні мережі представляють альтернативний підхід до використання генеративних моделей для детекції аномалій. AnoGAN навчає генератор на нормальних даних, а потім використовує процедуру оптимізації для знаходження найкращого латентного представлення нових спостережень. Аномалії характеризуються неможливістю знайти адекватне латентне представлення.

BiGAN розширює стандартну архітектуру GAN, додаючи енкодер, що відображає дані в латентний простір. Це дозволяє безпосередньо обчислювати латентні представлення без складної процедури оптимізації, роблячи детекцію аномалій більш ефективною.

1.6 Спеціалізовані методи для різних типів даних

Ефективність методів виявлення аномалій суттєво залежить від типу та структури аналізованих даних. Різні типи даних мають специфічні характеристики, що вимагають адаптації загальних підходів або розробки спеціалізованих методів.

Часові ряди представляють особливий клас даних, де порядок спостережень має критичне значення. Аномалії в часових рядах можуть

проявлятися як різкі стрибки значень, зміни в трендах, порушення сезонності або появи нетипових патернів. Seasonal Hybrid ESD алгоритм, розроблений Twitter, використовує сезонну декомпозицію часового ряду та статистичні тести для виявлення аномалій з урахуванням сезонних коливань.

Прогнозні моделі, такі як ARIMA або експоненційне згладжування, можуть використовуватися для виявлення аномалій шляхом порівняння фактичних значень з прогнозованими. Спостереження, що значно відхиляються від прогнозу, класифікуються як потенційні аномалії. Цей підхід особливо ефективний для даних з чіткими трендами та сезонними патернами.

Потокові дані вимагають online методів обробки, здатних працювати в режимі реального часу з обмеженою пам'яттю. Incremental learning алгоритми адаптують свої моделі в міру надходження нових даних, не перенавчаючись на всьому історичному наборі. Drift detection методи виявляють зміни в розподілі даних, що можуть вказувати на появу нових типів аномалій або зміну характеристик нормальної поведінки.

Високорозмірні дані створюють особливі виклики через прояв curse of dimensionality, коли традиційні метрики відстані втрачають дискримінативну здатність. Feature selection та dimensionality reduction методи стають критично важливими для ефективної детекції аномалій в таких даних. Sparse методи використовують припущення про те, що аномалії можуть бути представлені як розріджені відхилення від нормальних патернів.

2 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ

2.1 Бібліотеки Python для виявлення аномалій

Екосистема Python пропонує широкий спектр інструментів для виявлення аномалій, що варіюються від загальних бібліотек машинного навчання до спеціалізованих пакетів, розроблених виключно для детекції відхилень. Ці інструменти відрізняються за функціональністю, простотою використання, продуктивністю та підтримуваними алгоритмами.

Scikit-learn займає особливе місце серед бібліотек машинного навчання завдяки своїй універсальності та добре продуманому API. Бібліотека включає кілька алгоритмів виявлення аномалій в модулі sklearn.ensemble та sklearn.svm. Isolation Forest реалізує алгоритм ізоляційного лісу, що базується на припущенні про те, що аномалії легше ізолювати в просторі ознак порівняно з нормальними спостереженнями. Алгоритм будує ансамбль випадкових дерев розбиття та використовує середню глибину ізоляції як міру аномальності спостереження.

One-Class SVM представляє адаптацію метода опорних векторів для задачі виявлення новизни. Алгоритм навчається на даних одного класу, зазвичай нормальних спостереженнях, та будує гіперплощину, що відділяє цей клас від походження координат у просторі ознак. Нові спостереження класифікуються як аномальні, якщо вони потрапляють по протилежний бік гіперплощини від навчальних даних.

Local Outlier Factor реалізує алгоритм локального фактора викиду, що оцінює локальну щільність кожного спостереження відносно щільності його найближчих сусідів. Високі значення LOF вказують на те, що спостереження знаходиться в області з нижчою щільністю порівняно з його околицею, що є ознакою аномальності.

Elliptic Envelope базується на припущенні про багатовимірний

нормальний розподіл даних та будує еліптичну оболонку, що охоплює більшість нормальних спостережень. Алгоритм використовує робастні оцінки середнього та коваріації для зменшення впливу викидів на параметри моделі.

PyOD являє собою спеціалізовану бібліотеку, розроблену виключно для задач виявлення аномалій. Бібліотека пропонує уніфікований інтерфейс для більш ніж 40 алгоритмів детекції аномалій, включаючи як класичні статистичні методи, так і сучасні підходи на основі глибокого навчання. Архітектура PyOD базується на принципах модульності та розширюваності, дозволяючи легко додавати нові алгоритми та комбінувати існуючі методи.

Бібліотека включає реалізації методів на основі відстані, таких як k-найближчих сусідів та локальний фактор викиду, методи кластеризації, включаючи k-means та BIRCH, лінійні моделі, такі як PCA та MCD, а також нейронні мережі, включаючи автоенкодери різних архітектур. Особливістю PyOD є можливість комбінування різних детекторів для підвищення надійності виявлення аномалій.

TensorFlow та Keras забезпечують потужну платформу для розробки методів глибокого навчання для детекції аномалій. Ці бібліотеки дозволяють реалізовувати складні архітектури нейронних мереж, включаючи автоенкодери, варіаційні автоенкодери, рекурентні мережі та генеративно-змагальні мережі. Гнучкість TensorFlow дозволяє експериментувати з нестандартними архітектурами та функціями втрат, спеціально адаптованими для задач виявлення аномалій.

Реалізація автоенкодерів в Keras є відносно простою завдяки високорівневому API бібліотеки. Можна легко створювати архітектури різної складності, від простих повнозв'язних мереж до складних конволюційних або рекурентних архітектур. LSTM мережі в TensorFlow особливо корисні для аналізу часових рядів та послідовних даних.

ADTK представляє спеціалізований інструмент для виявлення аномалій в часових рядах. Бібліотека пропонує модульний підхід до обробки часових

рядів, включаючи попередню обробку, детекцію аномалій та пост-обробку результатів. ADTK підтримує різні типи аномалій в часових рядах, включаючи точкові викиди, зміни рівня, зміни тренду та сезонні аномалії.

Luminol, розроблений LinkedIn, спеціально призначений для виявлення аномалій в часових рядах метрик моніторингу. Бібліотека використовує статистичні методи та машинне навчання для автоматичного виявлення різних типів аномалій, включаючи короткострокові спайки та довгострокові зміни трендів.

2.2 Інструменти R для детекції аномалій

Статистичне середовище R пропонує потужні інструменти для аналізу часових рядів та виявлення аномалій, що особливо корисно для статистичного моделювання та дослідницького аналізу даних. Екосистема R включає як загальні пакети для аналізу даних, так і спеціалізовані рішення для детекції аномалій.

Пакет forecast є одним з найпопулярніших інструментів для аналізу та прогнозування часових рядів в R. Хоча основною метою пакету є прогнозування, він також включає функції для виявлення аномалій через аналіз залишків прогнозних моделей. Пакет підтримує широкий спектр моделей часових рядів, включаючи ARIMA, експоненційне згладжування та сезонну декомпозицію.

Anomalize представляє сучасний підхід до виявлення аномалій в часових рядах, спеціально розроблений для роботи з великими обсягами даних та інтеграції з екосистемою tidyverse. Пакет використовує декомпозицію часових рядів для виділення тренду, сезонності та залишків, після чого застосовує статистичні методи для виявлення аномалій в кожній компоненті.

AnomalyDetection, розроблений Twitter, реалізує Seasonal Hybrid Extreme Studentized Deviate алгоритм для виявлення аномалій в часових

рядах з сезонними компонентами. Пакет автоматично визначає сезонність та виявляє як глобальні, так і локальні аномалії з врахуванням часових трендів.

2.3 Комерційні рішення

Ринок комерційних рішень для виявлення аномалій характеризується різноманітністю підходів та цільових аудиторій. Хмарні платформи великих технологічних компаній інтегрують можливості детекції аномалій в свої екосистеми сервісів, пропонуючи масштабовані рішення для корпоративних клієнтів.

AWS GuardDuty представляє сервіс виявлення загроз, що використовує машинне навчання для аналізу подій безпеки та виявлення підозрілої активності в AWS середовищі. Сервіс аналізує VPC Flow Logs, DNS запити та CloudTrail події для виявлення аномальної поведінки, що може вказувати на компроментацію системи або несанкціоновану активність.

Azure Security Center інтегрує можливості виявлення аномалій в свою платформу безпеки, використовуючи алгоритми машинного навчання для аналізу логів та метрик з різних джерел. Система здатна виявляти складні атаки, що розгортаються на кількох етапах, аналізуючи кореляції між різними типами подій.

Google Cloud Security Command Center пропонує централізовану платформу для моніторингу безпеки та виявлення аномалій в Google Cloud середовищі. Система використовує передові алгоритми машинного навчання для аналізу активності користувачів, мережевого трафіку та конфігурацій ресурсів.

Корпоративні системи моніторингу та аналітики також активно інтегрують можливості виявлення аномалій. Splunk пропонує широкий спектр інструментів для аналізу машинних даних, включаючи спеціалізовані додатки для виявлення аномалій в різних предметних областях. Платформа дозволяє створювати власні моделі детекції аномалій та інтегрувати їх з

існуючими процесами моніторингу.

Elastic Stack забезпечує потужні можливості для збору, зберігання та аналізу логів з можливостями машинного навчання для виявлення аномалій. Elasticsearch включає X-Pack Machine Learning, що дозволяє автоматично виявляти аномалії в часових рядах метрик та логах без необхідності попереднього налаштування моделей.

DataDog спеціалізується на моніторингу продуктивності додатків та інфраструктури, пропонуючи алгоритми виявлення аномалій для метрик системного моніторингу. Платформа використовує сезонні алгоритми та машинне навчання для автоматичного виявлення відхилень в продуктивності систем.

2.4 Open-source інструменти

Відкриті програмні рішення відіграють важливу роль в екосистемі інструментів для виявлення аномалій, забезпечуючи доступність передових технологій для широкого кола користувачів та сприяючи розвитку нових підходів через колективні зусилля спільноти розробників.

Apache Spark MLlib інтегрує можливості виявлення аномалій в розподілену платформу обробки великих даних. Бібліотека включає реалізації кількох алгоритмів машинного навчання, адаптованих для роботи з великими обсягами даних в розподіленому середовищі. Особливо корисними є реалізації кластеризаційних алгоритмів та методів зменшення розмірності, що можуть використовуватися для виявлення аномалій.

H2O.ai пропонує відкриту платформу машинного навчання з потужними можливостями автоматизації. Платформа включає автоматичні алгоритми виявлення аномалій та дозволяє легко експериментувати з різними підходами. Особливістю H2O є можливість автоматичного налаштування гіперпараметрів та вибору оптимальних алгоритмів для конкретних наборів даних.

Weka представляє класичну платформу машинного навчання з графічним інтерфейсом користувача, що робить алгоритми виявлення аномалій доступними для дослідників без глибоких знань програмування. Платформа включає широкий спектр алгоритмів та інструментів для візуалізації результатів.

2.5 Порівняльний аналіз інструментів

Вибір оптимального інструменту для виявлення аномалій потребує комплексного аналізу різних факторів, включаючи функціональні можливості, продуктивність, простоту використання та підтримку спільноти. Кожен інструмент має свої переваги та обмеження, що роблять його більш або менш придатним для конкретних застосувань.

Функціональність інструментів суттєво варіюється від простих статистичних методів до складних ансамблів алгоритмів машинного навчання. Scikit-learn пропонує добре протестовані реалізації основних алгоритмів з акцентом на стабільність та надійність. PyOD забезпечує найширший спектр алгоритмів детекції аномалій в єдиному інтерфейсі, що робить його привабливим для експериментального дослідження. TensorFlow надає максимальну гнучкість для розробки нестандартних рішень, але потребує більших зусиль для реалізації.

Продуктивність інструментів критично важлива для застосувань з великими обсягами даних або жорсткими вимогами до часу відгуку. Реалізації в scikit-learn оптимізовані для швидкості та ефективного використання пам'яті. Apache Spark MLlib забезпечує горизонтальну масштабованість для обробки великих даних, але може мати накладні витрати для малих наборів даних.

Простота використання впливає на швидкість розробки та підтримки рішень. Інструменти з високорівневими API, такі як scikit-learn та PyOD, дозволяють швидко прототипувати рішення. Комерційні платформи часто

пропонують графічні інтерфейси та автоматизовані workflow, що знижують бар'єр входу для нетехнічних користувачів.

Документація та підтримка спільноти відіграють важливу роль в довгостроковому успіху проекту. Scikit-learn та TensorFlow мають обширну документацію та активні спільноти розробників. Комерційні рішення зазвичай пропонують професійну технічну підтримку, але можуть обмежувати гнучкість налаштувань.

Масштабованість стає критичним фактором для корпоративних застосувань. Хмарні рішення автоматично масштабуються відповідно до навантаження, але можуть мати обмеження щодо конфіденційності даних. Відкриті рішення дозволяють повний контроль над інфраструктурою, але потребують власних ресурсів для масштабування.

Інтеграційні можливості визначають, наскільки легко інструмент може бути включений в існуючі системи та процеси. Python-бібліотеки легко інтегруються з наявними data science workflow. Корпоративні платформи часто пропонують готові інтеграції з популярними системами моніторингу та бізнес-аналітики.

3 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

3.1 Аналіз вимог до системи

Проектування ефективної системи виявлення аномалій потребує ретельного аналізу функціональних та нефункціональних вимог, що відображають потреби кінцевих користувачів та технічні обмеження середовища функціонування. Визначення чітких вимог є критично важливим етапом, що визначає архітектурні рішення та впливає на всі подальші етапи розробки.

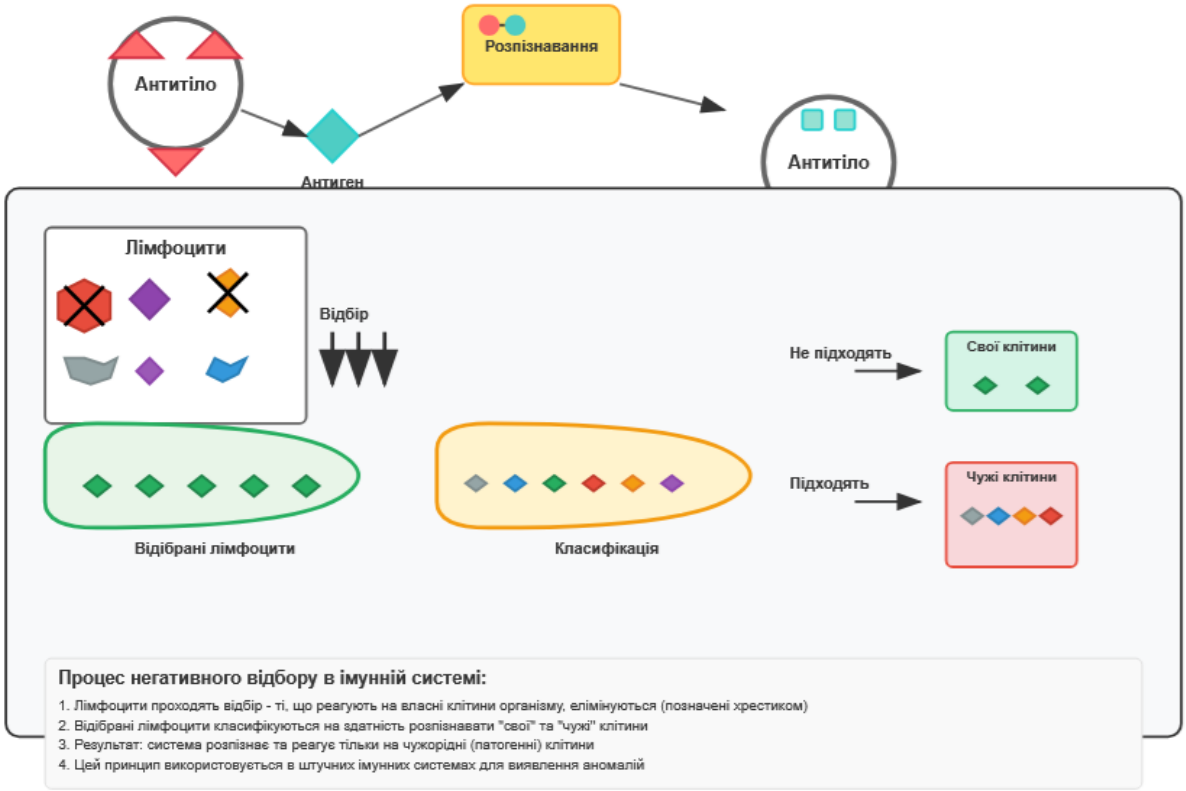

Рисунок 3.1 – Схема імунної системи та негативного відбору

Функціональні вимоги визначають основні можливості системи та описують, що саме система повинна робити для задоволення потреб користувачів. Система повинна забезпечувати підтримку широкого спектру

форматів вхідних даних, включаючи структуровані дані у форматах CSV, JSON, XML, а також можливість підключення до різних типів баз даних. Це забезпечить універсальність рішення та можливість його використання в різних організаційних контекстах без необхідності додаткової конвертації даних.

Критично важливою функціональною вимогою є підтримка множинних алгоритмів детекції аномалій, що дозволить користувачам експериментувати з різними підходами та обирати найбільш підходящий для їх специфічних даних та завдань. Система повинна включати як класичні статистичні методи, так і сучасні алгоритми машинного навчання та глибокого навчання.

Візуалізація результатів представляє ще один ключовий функціональний аспект системи. Користувачі повинні мати можливість не лише отримувати числові результати детекції, але й візуально аналізувати виявлені аномалії в контексті всього набору даних. Це включає можливості створення різноманітних типів графіків, heat map'ів, діаграм розсіювання та інших візуальних представлень.

Функція експорту звітів забезпечить можливість документування результатів аналізу та їх подальшого використання в презентаціях, звітах або інших аналітичних системах. Система повинна підтримувати експорт у популярні формати, включаючи PDF, HTML, Excel та CSV.

Нефункціональні вимоги визначають якісні характеристики системи та обмеження, в рамках яких вона повинна функціонувати. Продуктивність є одним з найважливіших нефункціональних аспектів, особливо враховуючи потенційно великі обсяги даних, що можуть обробляти користувачі. Система повинна забезпечувати прийнятний час відгуку для інтерактивної роботи користувача та ефективно використовувати доступні обчислювальні ресурси.

Масштабованість визначає здатність системи справлятися з зростаючими обсягами даних та кількістю користувачів. Архітектура системи повинна дозволяти горизонтальне та вертикальне масштабування без значних змін в основному коді. Це може включати можливість розподіленої

обробки даних або інтеграцію з хмарними обчислювальними ресурсами.

Надійність системи забезпечується через обробку помилкових ситуацій, валідацію вхідних даних та стабільну роботу в різних умовах експлуатації. Система повинна gracefully обробляти неочікувані формати даних, пропущені значення та інші аномальні ситуації, що можуть виникнути при роботі з реальними даними.

Безпека даних є критично важливою вимогою, особливо при роботі з конфіденційною інформацією. Система повинна забезпечувати належний рівень захисту даних користувачів, включаючи можливості контролю доступу, шифрування даних та аудиту використання системи.

3.2 Архітектура системи

Архітектурний дизайн системи виявлення аномалій базується на принципах модульності, розширюваності та слабкого зв'язування компонентів. Такий підхід забезпечує гнучкість в розробці, тестуванні та підтримці системи, а також дозволяє легко додавати нові функції та алгоритми без впливу на існуючий код.

Загальна архітектура системи побудована на основі багатошарової моделі, що включає рівень представлення, бізнес-логіки, доступу до даних та зберігання даних. Рівень представлення відповідає за взаємодію з користувачем та включає веб-інтерфейс, API endpoints та компоненти візуалізації. Рівень бізнес-логіки містить основну функціональність системи, включаючи алгоритми детекції аномалій, обробку даних та управління workflow.

Модульна структура системи передбачає виділення окремих компонентів для кожної основної функції. Це дозволяє розробляти та тестувати кожен модуль незалежно, а також забезпечує можливість заміни або оновлення окремих компонентів без впливу на решту системи. Кожен модуль має чітко визначений інтерфейс взаємодії з іншими компонентами.

Паттерни проектування відіграють важливу роль в забезпеченні якості архітектури. Паттерн Strategy використовується для інкапсуляції різних алгоритмів детекції аномалій, дозволяючи динамічно обирати та змінювати алгоритми в залежності від типу даних та вимог користувача. Паттерн Observer забезпечує можливість сповіщення різних компонентів про зміни стану обробки даних.

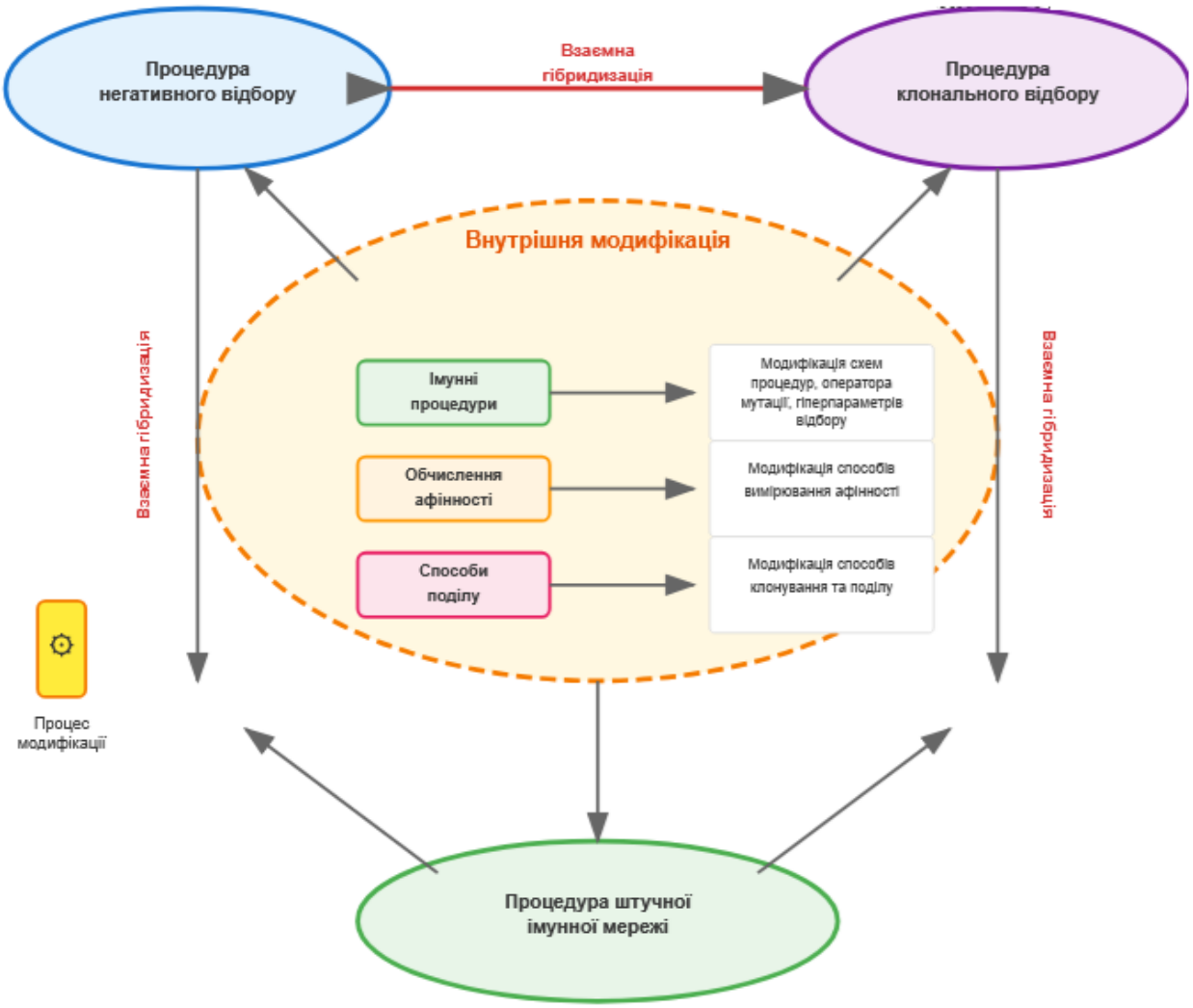

Рисунок 3.2 – Гібридизація ШІС

Паттерн Factory Method використовується для створення екземплярів різних типів детекторів аномалій, забезпечуючи абстракцію процесу створення об'єктів та спрощуючи додавання нових алгоритмів. Паттерн Decorator дозволяє динамічно додавати нову функціональність до існуючих

детекторів, наприклад, попередню обробку даних або пост-обробку результатів.

Інтерфейси взаємодії між компонентами системи стандартизовані та добре документовані. Це забезпечує можливість незалежної розробки різних модулів різними командами розробників та спрощує інтеграцію з зовнішніми системами. REST API забезпечує програмний доступ до функціональності системи для інтеграції з іншими додатками.

В роботі будуть використовуватися моделі клонального відбору та штучні імунні системи (алгоритм роботи представлений на рисунку 3.3).

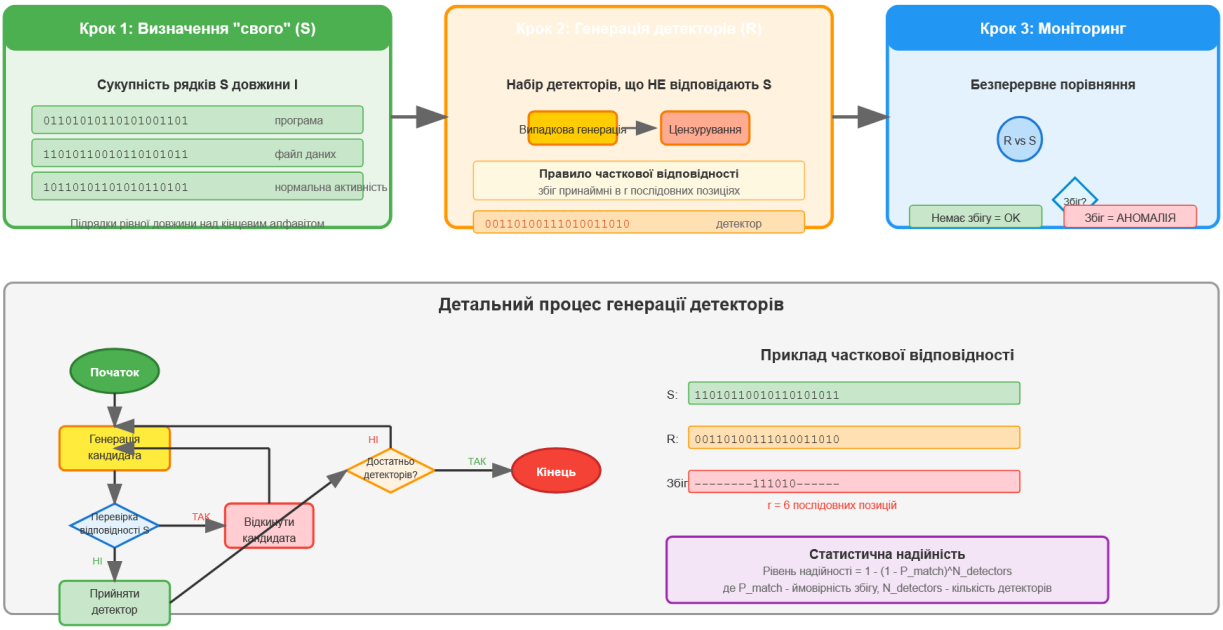

Рисунок 3.3 – Алгоритм

Крок 1: визначимо «своє» як сукупність рядків S довжини l над кінцевим алфавітом, яку необхідно захищати або контролювати. Наприклад, як S можуть виступати програма, файл даних (будь-яке програмне забезпечення) або нормальна форма активності, що підрозділяються на підрядки рівної довжини).

Крок 2: утворюємо набір детекторів R, кожен з яких не повинен відповідати будь-якому рядку в S. Замість точної, або ідеальної, відповідності використовуємо правило часткової відповідності, при якому

два рядки відповідають один одному, якщо і тільки якщо вони збігаються принаймні в r наступних один за одним позиціях, де r – деякий цілісний параметр.

Крок 3: перевіримо S на предмет змін шляхом безперервного порівняння детекторів R з елементами S. Якщо хоча б один з детекторів виявиться відповідним, значить відбулася зміна, оскільки детектори за визначенням відібрані так, щоб не відповідати будь-якому рядку S. У вихідному описі алгоритму кандидати в детектори генерувалися випадково і потім перевірялися (зазнавали цензурування) на відповідність будь-якому рядку свому. Якщо відповідність мала місце, такий кандидат відкидався. Процедура повторювалася до того часу, доки створювалися необхідна кількість детекторів, необхідне забезпечення певного рівня надійності, яке оцінювалося з використанням статистичних методів.

3.3 Детальний опис модулів

Модуль завантаження даних відповідає за отримання вхідної інформації з різноманітних джерел та її первинну валідацію. Цей модуль реалізує підтримку широкого спектру форматів даних, включаючи текстові файли з роздільниками, JSON документи, XML структури та прямі підключення до баз даних. Кожен тип джерела даних обробляється спеціалізованим адаптером, що забезпечує уніфікований інтерфейс для подальшої обробки.

Валідація даних включає перевірку структури файлів, типів даних, наявності обов'язкових полів та відповідності очікуваним форматам. Модуль забезпечує детальне логування помилок валідації та надає користувачу зрозумілі повідомлення про виявлені проблеми. Імплементована система автоматичного визначення типів даних та структури для спрощення процесу завантаження.

Обробка помилок в модулі завантаження реалізована через систему

винятків з можливістю відновлення після помилок, коли це можливо. Модуль підтримує часткове завантаження даних у випадках, коли частина інформації може бути оброблена незважаючи на помилки в інших частинах файлу.

Модуль попередньої обробки виконує критично важливі операції підготовки даних для подальшого аналізу. Очищення даних включає виявлення та обробку пропущених значень, дублікатів записів, некоректних форматів та інших проблем якості даних. Різні стратегії обробки пропущених значень дозволяють користувачу обирати найбільш підходящий підхід для їх конкретного випадку.

Нормалізація даних забезпечує приведення різних змінних до спільного масштабу, що є критично важливим для багатьох алгоритмів машинного навчання. Модуль підтримує різні методи нормалізації, включаючи min-max scaling, z-score normalization та robust scaling. Автоматичне визначення оптимального методу нормалізації базується на статистичних характеристиках даних.

Feature engineering включає створення нових ознак на основі існуючих даних, що може суттєво покращити якість детекції аномалій. Модуль підтримує автоматичне генерування поліноміальних ознак, взаємодій між змінними, часових ознак для даних з часовими мітками та інших корисних трансформацій.

Модуль виявлення аномалій є серцем системи та містить реалізації основних алгоритмів детекції. Архітектура модуля базується на абстрактному базовому класі детектора, що визначає загальний інтерфейс для всіх алгоритмів. Це забезпечує уніфікований підхід до використання різних методів та спрощує додавання нових алгоритмів.

Реалізація основних алгоритмів включає як статистичні методи, такі як Z-score та IQR, так і алгоритми машинного навчання, включаючи Isolation Forest, Local Outlier Factor та One-Class SVM. Кожен алгоритм оптимізований для специфічних типів даних та завдань, з можливістю налаштування параметрів відповідно до потреб користувача.

Система налаштування параметрів включає як ручне налаштування досвідченими користувачами, так і автоматичний підбір оптимальних параметрів на основі характеристик даних. Реалізовані механізми cross-validation та grid search для систематичного пошуку найкращих комбінацій параметрів.

Комбінування методів дозволяє використовувати ансамблі детекторів для підвищення надійності та точності виявлення аномалій. Різні стратегії комбінування, включаючи voting, averaging та stacking, дають користувачам можливість експериментувати з різними підходами до побудови композитних моделей.

Модуль візуалізації забезпечує графічне представлення даних та результатів детекції аномалій. Реалізовані різноманітні типи візуалізацій, від простих графіків розподілу до складних багатовимірних представлень. Інтерактивні елементи дозволяють користувачам детально досліджувати виявлені аномалії та розуміти їх контекст.

Heat maps забезпечують ефективне відображення кореляційних зв'язків між змінними та допомагають виявляти патерни в багатовимірних даних. Інтерактивні charts дозволяють користувачам динамічно фільтрувати та досліджувати дані, змінювати параметри відображення та експортувати результати у різних форматах.

3D візуалізації особливо корисні для аналізу високорозмірних даних після їх проекції на тривимірний простір за допомогою методів зменшення розмірності. Такі візуалізації допомагають інтуїтивно зрозуміти структуру даних та розташування аномалій в просторі ознак.

3.4 Технології та інструменти розробки

Вибір технологічного стеку для розробки системи виявлення аномалій базується на аналізі вимог до функціональності, продуктивності, масштабованості та підтримки. Python обрано як основну мову

програмування завдяки її широкій екосистемі бібліотек для машинного навчання, аналізу даних та наукових обчислень, а також активній спільноті розробників та обширній документації.

Екосистема Python для data science включає потужні інструменти, що значно спрощують розробку складних аналітичних систем. NumPy забезпечує ефективні операції з багатовимірними масивами та математичні функції, що є основою для всіх числових обчислень в системі. Pandas надає зручні структури даних та інструменти для маніпуляції з табличними даними, включаючи завантаження, очищення та трансформацію інформації.

Scikit-learn служить основною бібліотекою для реалізації алгоритмів машинного навчання завдяки своєму консистентному API, широкому спектру алгоритмів та відмінній документації. Бібліотека включає не лише алгоритми детекції аномалій, але й інструменти для попередньої обробки даних, оцінки якості моделей та налаштування гіперпараметрів.

Matplotlib та Plotly використовуються для створення візуалізацій різного рівня складності. Matplotlib забезпечує базові можливості створення статичних графіків з високою якістю, тоді як Plotly додає інтерактивність та можливості створення веб-сумісних візуалізацій. Комбінація цих бібліотек дозволяє створювати як прості діаграми для швидкого аналізу, так і складні інтерактивні дашборди.

Streamlit обрано для створення веб-інтерфейсу завдяки його простоті використання та можливості швидкого прототипування. Фреймворк дозволяє створювати інтерактивні веб-додатки з мінімальним кодом, автоматично обробляє взаємодію з користувачем та забезпечує реактивне оновлення інтерфейсу при зміні даних.

Інструменти розробки включають сучасні IDE та системи підтримки життєвого циклу розробки. PyCharm Professional забезпечує потужні можливості для розробки Python додатків, включаючи інтелектуальне автодоповнення, рефакторинг, дебагінг та інтеграцію з системами контролю версій. Visual Studio Code служить альтернативним редактором для швидкого

редагування та експериментів.

Git використовується для контролю версій коду та забезпечення можливості колективної розробки. Репозиторій структурований відповідно до best practices, включаючи чітке розділення на модулі, документацію та тести. Branching strategy базується на Git Flow для забезпечення стабільності основної гілки розробки.

Conda обрано як пакетний менеджер та менеджер середовищ завдяки його можливостям управління як Python пакетами, так і системними залежностями. Це особливо важливо для наукових обчислень, де часто потрібні оптимізовані версії бібліотек з нативним кодом.

3.5 Реалізація ключових алгоритмів

Реалізація алгоритмів виявлення аномалій в розробленій системі базується на принципах модульності, ефективності та розширюваності. Кожен алгоритм інкапсульований в окремий клас, що наслідує загальний інтерфейс детектора, забезпечуючи уніфікований підхід до їх використання та можливість легкого додавання нових методів.

Z-score метод реалізований як один з базових статистичних підходів до виявлення аномалій. Алгоритм обчислює стандартизовані залишки для кожного спостереження та класифікує як аномальні ті, що перевищують заданий поріг. Реалізація включає опції для використання робастних статистик, таких як медіана та медіанне абсолютне відхилення, що робить метод менш чутливим до наявності викидів у даних.

IQR метод базується на розподілі квартилів та виявляє спостереження, що знаходяться за межами діапазону, визначеного першим та третім квартилями з урахуванням коефіцієнта масштабування. Реалізація забезпечує можливість налаштування цього коефіцієнта залежно від специфіки даних та вимог до чутливості детекції.

Isolation Forest реалізований як один з найефективніших алгоритмів для

виявлення аномалій у високорозмірних даних. Алгоритм будує ансамбль ізоляційних дерев, кожне з яких створюється шляхом випадкового вибору ознаки та порогового значення для розбиття даних. Аномалії характеризуються меншою глибиною ізоляції в деревах, оскільки їх легше відділити від основної маси даних.

Реалізація включає оптимізації для роботи з великими обсягами даних, включаючи sampling стратегії та паралельну обробку. Параметри алгоритму, такі як кількість дерев, розмір підвибірки та поріг аномальності, можуть налаштовуватися автоматично на основі характеристик вхідних даних.

Local Outlier Factor алгоритм реалізований для виявлення локальних аномалій, що можуть не виявлятися глобальними методами. Алгоритм обчислює локальну щільність кожного спостереження відносно його k найближчих сусідів та ідентифікує точки з значно нижчою локальною щільністю як аномалії.

Ефективна реалізація LOF використовує оптимізовані структури даних для пошуку найближчих сусідів, такі як KD-дерева або Ball trees, що значно прискорює обчислення для великих наборів даних. Адаптивний вибір параметра k базується на аналізі локальної структури даних.

Автоенкодер для виявлення аномалій реалізований на основі TensorFlow/Keras та представляє нейронну мережу, що навчається стискати та відновлювати вхідні дані. Архітектура мережі автоматично адаптується до розмірності вхідних даних, використовуючи принципи проектування deep learning моделей.

Навчання автоенкодера проводиться виключно на нормальних даних з метою мінімізації помилки реконструкції. Аномалії виявляються як спостереження з високою помилкою відновлення, оскільки мережа не навчалася їх ефективно кодувати та декодувати. Реалізація включає техніки регуляризації та early stopping для запобігання перенавчанню.

3.6 Користувацький інтерфейс

Користувацький інтерфейс системи розроблений з урахуванням принципів user experience design та забезпечує інтуїтивну взаємодію користувачів різного рівня технічної підготовки з функціональністю системи. Веб-інтерфейс побудований на основі Streamlit фреймворку, що забезпечує реактивну поведінку та автоматичне оновлення результатів при зміні параметрів.

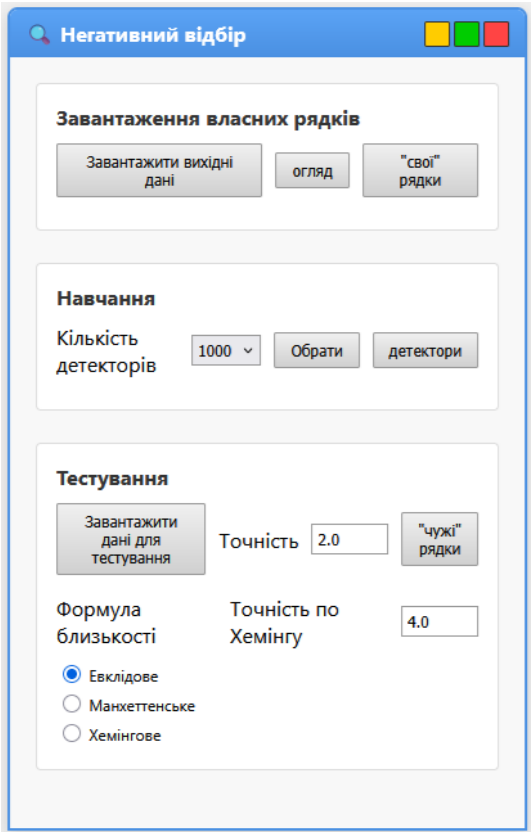

Рисунок 3.4 – Інтерфейс

Головна сторінка інтерфейсу надає огляд можливостей системи та швидкий доступ до основних функцій. Навігація організована у вигляді табів або сайдбару, що дозволяє легко переключатися між різними етапами аналізу. Кожна секція має контекстну довідку та приклади використання.

Workflow обробки даних організований як послідовність логічно пов'язаних кроків, починаючи з завантаження даних та закінчуючи аналізом

результатів. Кожен крок має чітко визначені вхідні та вихідні дані, а також можливості валідації та попереднього перегляду результатів. Користувач може повертатися до попередніх кроків для зміни параметрів без втрати попередньо виконаної роботи.

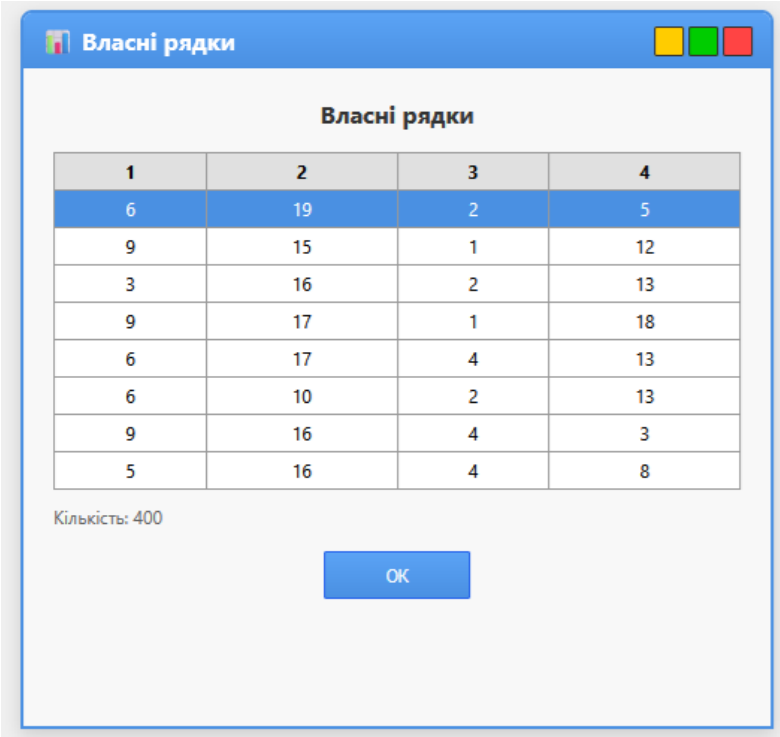

Рисунок 3.5 – Інтерфейс

Інтерфейс завантаження даних підтримує drag-and-drop функціональність та автоматичне визначення формату файлів. Попередній перегляд даних дозволяє користувачу переконатися в правильності завантаження та виявити потенційні проблеми на ранньому етапі. Статистичний огляд завантажених даних включає інформацію про розмір набору, типи змінних та базові статистики.

Налаштування параметрів алгоритмів організоване у вигляді інтерактивних елементів управління з миттєвим відображенням їх впливу на результати. Експертний режим дозволяє досвідченим користувачам налаштовувати детальні параметри, тоді як автоматичний режим забезпечує

розумні значення за замовчуванням для початківців.

Перегляд результатів включає як табличне представлення виявлених аномалій, так і різноманітні візуалізації. Інтерактивні графіки дозволяють користувачам детально досліджувати виявлені аномалії, фільтрувати результати за різними критеріями та експортувати дані для подальшого аналізу. Система надає можливості створення звітів у різних форматах для документування результатів аналізу.

4 РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ

Створення програмних реалізацій алгоритмів штучної імунної системи вимагає застосування об'єктно-орієнтованої парадигми програмування та формування відповідної класової архітектури незалежно від обраної платформи розробки.

Аналіз доступних засобів програмування виявив специфічні характеристики кожного з розглянутих варіантів. C++ демонструє значні можливості завдяки комплексній підтримці об'єктно-орієнтованих принципів, розширеному спектру вбудованих типів даних та ефективним механізмам управління пам'яттю. C# характеризується універсальною класовою архітектурою для всіх елементів мови, що забезпечує високий рівень структурованості коду, проте ця особливість супроводжується зниженням продуктивності та збільшенням складності синтаксичних конструкцій.

Pascal демонструє обмежені можливості щодо адаптивності програмних рішень та неефективні механізми роботи з оперативною пам'яттю. Basic характеризується надмірною спрощеністю реалізації та недостатньою варіативністю програмних рішень. Java виділяється серед розглянутих альтернатiv завдяки поєднанню адаптивності та нативної підтримки об'єктно-орієнтованих принципів, однак її основне призначення орієнтоване на розробку міжплатформенних рішень та веб-орієнтованих технологій.

Комплексна оцінка програмних засобів розробки відповідно до специфіки поставленого завдання обґрунтувала доцільність використання інтегрованого середовища Microsoft Visual Studio 2010 з мовою програмування C#.

Була розроблена модель програмної системи (рисунок 4.2). Розроблений імунний алгоритм архівації використовує дійсне кодування

антитіл, де Ab_{ij} – координати центру антитіла в евклідовому просторі, r – радіус антитіла, що виконує функцію крос-реактивного порога, який адаптивно підбирається в процесі роботи алгоритму.

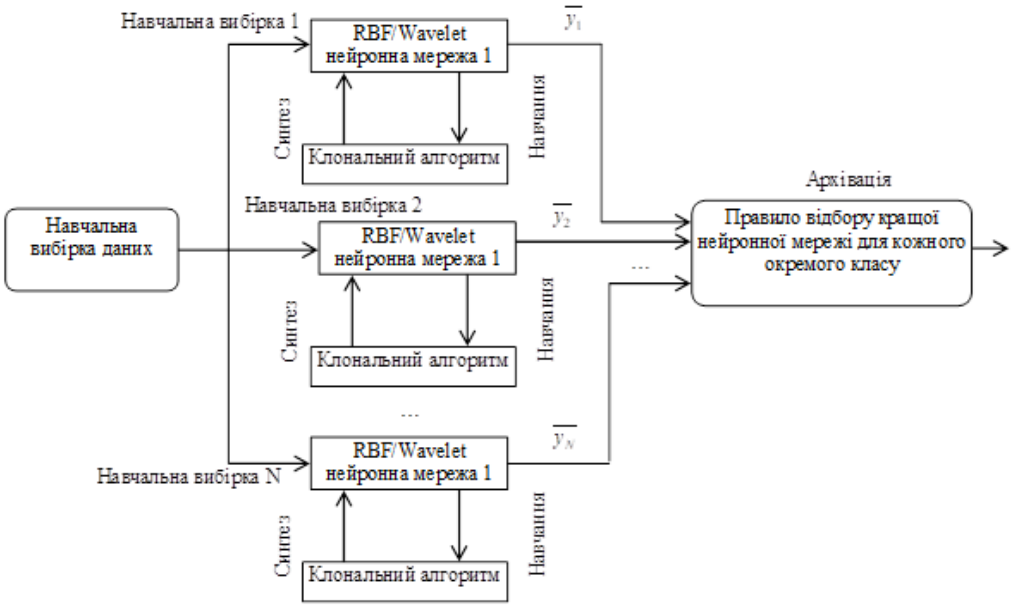

Рисунок 4.1 – Синтез колективу бінарних класифікаторів вейвлет-нейронних мереж

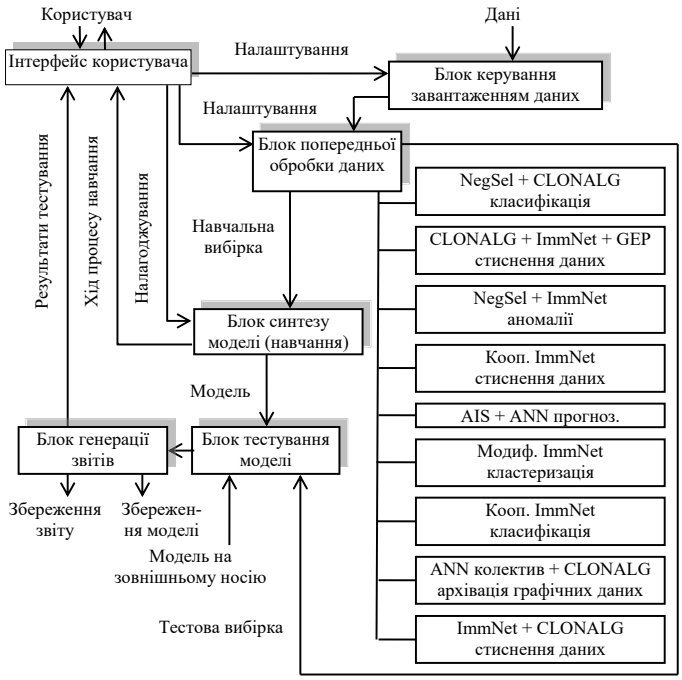

Рисунок 4.2 – Модель комп’ютерної системи при рішенні прикладного завдання

Формальне подання запропонованої процедури імунної мережі представлено на рисунку.

4.1 Об'єктно орієнтована модель штучної імунної системи

Основними поняттями об'єктно-орієнтованого підходу є об'єкт і клас. Об'єкт визначається як відчутна реальність - предмет або явище, що мають чітко визначається поведінка. Об'єкт має статки, поведінкою і індивідуальністю; структура і поведінка схожих об'єктів визначають загальний для них клас.

Клас - це безліч об'єктів, пов'язаних спільністю структури і поведінки. Будь-який об'єкт є екземпляром класу. Розглянемо розроблені класи:

- клас DataVector (рисунок 3.3) зберігає і надає доступ до списку довільної довжини data, який зберігає елементи типу double;
- Count - кількість елементів у списку;
- Data - надання доступу до самих даних; Equals - перевантажений метод, що виконує порівняння двох об'єктів типу DataVector;
- EuclideAffinity - обчислення афінності за формулою Еквкліда;
- HemmingAffinity - обчислення афінності за формулою Хемминга;
- ManhattanAffinity - обчислення афінності по Манхеттенського віддалі;
- LoadVectors - завантаження векторів з файлу;
- ToString - перетворення даних, що зберігаються в рядок.
- Клас DoubleRange (рисунок 4.4) зберігає діапазон чисел, обмежений мінімальним і максимальним значенням.

Клас NegativeSelectionAlgoritm (рисунок 4.3) Цей клас виконує реалізацію алгоритму негативного відбору:

- alienElements - чужорідні елементи;
- detectors - детектори;
- vectors - «свої» елементи;

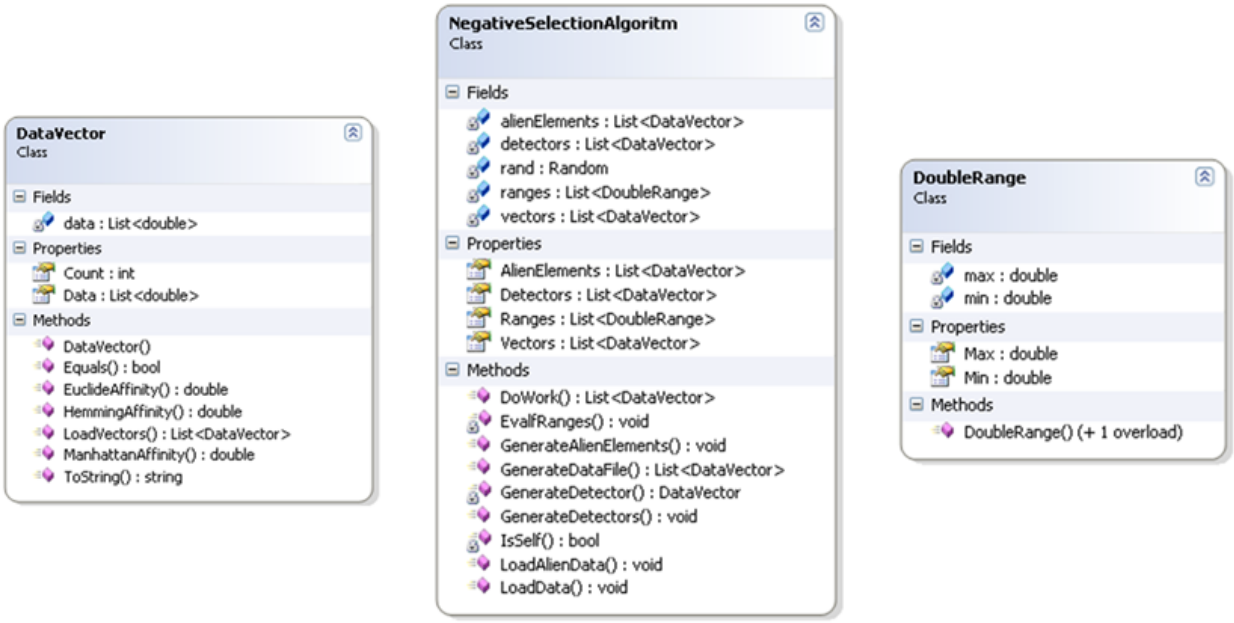

Рисунок 4.3 – Класи ШІС

- rand - об'єкт класу, що відповідає за генерацію випадкових чисел; ranges - мінімальне і максимальне значення кожного поля в «своїх» даних;
- LoadData - виконує завантаження «своїх» даних;
- LoadAlienData - виконує завантаження «чужих» даних;
- GenerateDetector - виконує генерацію одного детектора; GenerateDetectors - виконує генерацію count детекторів;
- GenerateDataFile - генерує базу значень, які визначаються як «свої» і виписує їх в файл;
- GenerateAlienElements - генерує базу значень, які визначаються як «чужі» і виписує їх в файл;
- EvalfRanges - пошук мінімальних і максимальних значень полів в «своїх» даних;
- IsSelf - виконує перевірку входження елемента v в список «своїх»
- DoWork - виконує пошук «чужих» рядків з урахуванням алгоритму обчислення афінності і точності обчислень.

Валідація функціональних можливостей створеного програмного забезпечення здійснювалася з використанням відомого набору даних Iris Dataset, запропонованого Рональдом Фішером. Цей класичний датасет

містить інформацію про 150 ботанічних зразків, рівномірно розподілених між трьома видами: по п'ятдесят екземплярів Iris setosa та Iris versicolor кожного.

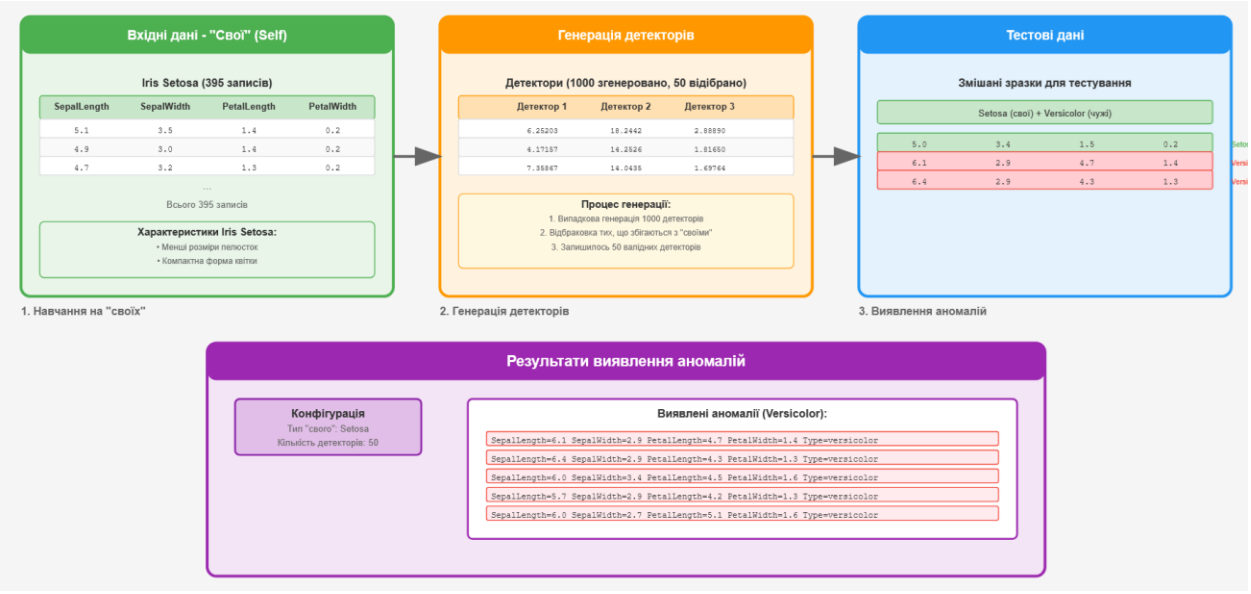

Рисунок 4.4 – Результати роботи

Кожен зразок у колекції характеризується набором з чотирьох морфометричних параметрів, зафіксованих у сантиметрах:

- параметр довжини чашолистка;
- параметр ширини чашолистка;
- параметр довжини пелюстки;
- параметр ширини пелюстки.

На рисунку 4.4 представлено демонстрацію функціонування розробленої програми. У якості еталонних даних було обрано зразки сорту Iris setosa. Етап навчання системи полягав у генерації множини детекторів згідно з принципами алгоритму негативного відбору, при цьому забезпечувалася умова відсутності збігів між будь-яким детектором та записами обраного "власного" класу ірисів.

На стадії тестування в систему завантажувалися дані, що містили як нормальні зразки сорту Iris setosa, так і аномальні записи представлені зразками сорту Iris versicolor. Сформована на попередньому етапі імунна система використовувала згенеровані детектори для ідентифікації відхилень

у вхідному потоці даних.

Вихідні дані програми зберігаються у текстовому форматі, з якого видно поетапну роботу системи. Спочатку користувачеві було запропоновано визначити еталонний клас серед доступних сортів ірисів - у даному випадку було обрано "setosa" як репрезентант нормальної поведінки. На наступному кроці система запитала параметри конфігурації, зокрема кількість детекторів для захисту еталонних записів, де користувач вказав значення 50. Фінальний етап роботи програми продемонстрував результати детекції - система успішно ідентифікувала 7 аномальних записів, що відповідали зразкам сорту "versicolor".

ВИСНОВКИ

Штучні імунні системи належать до методів, основу яких лежать біологічні процеси, що у природі. ШІС являють собою напрямок, близький до ідей та методів штучного інтелекту. Застосування ШІС дозволяє вирішувати багато завдань, таких як класифікація, кластеризація, розпізнавання, забезпечення комп'ютерної безпеки та багато інших. У зв'язку з тим, що зараз немає розроблених пакетів програм для роботи з ШІС, актуальним завданням є створення програмних засобів реалізації обчислювальних моделей ШІС. У роботі було проведено огляд штучних імунних систем та обчислювальних моделей, заснованих на принципах функціонування імунної системи

Розглянуто імунний алгоритм, що ґрунтується на принципах негативного відбору. Розроблено блок-схему даного алгоритму; розроблено програмні засоби реалізації обчислювальної моделі негативного відбору для виявлення аномалій у вихідних даних; розроблено інтерфейс користувача, проведена верифікація розробленого програмного забезпечення на основі тестового прикладу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. D. Dasgupta (Ed.) Artificial Immune Systems and Their Applications. // Springer, 1999.

2. L.N. de Castro and J. Timmis Artificial Immune Systems: A New Computational Intelligence Approach // Springer, 2002.

3. A.O.Tarakanov, V.A. Skormin and S.P. Sokolova Immunocomputing: Principles and Applications //Springer, 2003.

4. The Online Home of Artificial Immune Systems www.artificial-immune-systems.org

5.L.N. de Castro and J. Timmis Artificial Immune Systems: A New Computational Intelligence Approach // Springer, 2002.

6.L. N. de Castro and F. J. Von Zuben Learning and Optimization Using the Clonal Selection Principle // IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, vol. 6, n. 3, pp. 239-251, 2002.

7.L. N. de Castro and F. J. Von Zuben The Clonal Selection Algorithm with Engineering Applications // Proceedings of GECCO'00, Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA, July, pp. 36-37, 2000.

8.E. Clark, A. Hone and J. Timmis A Markov Chain Model of the B-Cell Algorithm // ICARIS 2005, Springer-Verlag, LNCS 3627, pp. 318 – 330, 2005.

9. L.N. de Castro and J. Timmis .Artificial Immune Systems: A New Computational Intelligence Approach // Springer, 2002.

10. D.Dasgupta A comparison of negative and positive selection algorithms in novel pattern detection.