

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

**КОМПЛЕКСНА  
КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА  
РОБОТА**

**на тему:**

**«Інформаційно-аналітична система адаптації  
навчального контенту випускової кафедри до  
вимог ринку праці. Інформаційна технологія  
машинного навчання»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Довбиш А.С.**

**Студент групи ІН.м-81н**

**Катькало Р.В**

**СУМИ 2020**

Сумський державний університет

(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук

Спеціальність «Інформатика»

Затверджую:

зав.кафедри \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Катькало Руслан Вікторович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційно-аналітична система адаптації навчального контенту випускної кафедри до вимог ринку праці. Інформаційна технологія машинного навчання

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. №

2. Термін здачі студентом закінченого проекту (роботи) \_\_\_\_\_

3. Вхідні данні до проекту (роботи) \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми та постановки задачі дослідження 2) Опис методу дослідження 3) Інформаційне та програмне забезпечення інформаційно-аналітичної системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

3 схематичних зображення для опису роботи алгоритмів; 14 рисунків виконання програмного коду.

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

| Розділ                       | Консультант        | Підпис, дата   |                  |
|------------------------------|--------------------|----------------|------------------|
|                              |                    | Завдання видав | Завдання прийняв |
| <i>1.3 Постановка задачі</i> | <i>Довбиш А.С.</i> |                |                  |
|                              |                    |                |                  |
|                              |                    |                |                  |
|                              |                    |                |                  |
|                              |                    |                |                  |

7.Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_  
(підпис)

Завдання прийняв до виконання

\_\_\_\_\_  
(підпис)

## КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів дипломного проекту (роботи)                                       | Термін виконання проекту (роботи) | Примітка |
|-------|--|-----------------------------------|----------|
| 1.    | <i>Огляд літератури згідно теми диплома</i>                                    | 02.03.2020-20.03.2020             |          |
| 2.    | <i>Аналіз проблеми та постановки задачі дослідження</i>                        | 21.03.2020-05.04.2020             |          |
| 3.    | <i>Опис методу дослідження</i>   | 06.04.2020-17.04.2020             |          |
| 4.    | <i>Інформаційне та програмне забезпечення інформаційно-аналітичної системи</i> | 18.04.2020-30.04.2020             |          |
| 5.    | <i>Оформлення пояснювальної записки до дипломної роботи</i>                    | 01.05.2020-15.05.2020             |          |

Студент – дипломник

\_\_\_\_\_  
(підпис)

Керівник проекту

\_\_\_\_\_  
(підпис)

## РЕФЕРАТ

**Записка:** 57 стор., 17 рис., 1 табл., 1 додаток, 15 джерел.

**Мета роботи** – підвищення функціональної ефективності інформаційно-аналітичної системи адаптації навчального контенту випускової кафедри до вимог ринку праці.

**Об'єкт дослідження** – формування вхідного математичного опису інформаційно-аналітичної системи адаптації навчального контенту кафедри до вимог ринку праці.

**Предмет дослідження** – вхідна навчальна матриця інформаційно-аналітичної системи адаптації навчального контенту кафедри до вимог ринку праці.

**Методи дослідження** – методи теорії розпізнавання образів, теорії інформації та інформаційно-екстремальної інтелектуальної технології аналізу даних.

**Результати** – Розроблено алгоритм навчання аналітично-інформаційної системи та його програмна реалізація в рамках інформаційно-екстремальної технології для адаптації навчального контенту до вимог ринку праці. Алгоритм було реалізовано за допомогою мови програмування C#.

АНАЛІТИЧНО-ІНФОРМАЦІЙНА СИСТЕМА, МАШИННЕ  
НАВЧАННЯ, БАЗОВИЙ АЛГОРИТМ МАШИННОГО  
НАВЧАННЯ, НАВЧАЛЬНА МАТРИЦЯ, МАТЕМАТИЧНИЙ  
ОПИС, ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА  
ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, ЯКІСТЬ НАВЧАЛЬНОГО  
КОНТЕНТУ

## ABSTRACT

**Note:** 57 pages, 17 figures, 1 table, 1 appendix, 15 sources.

**The purpose of the work** - to increase the functional efficiency of the information-analytical system of adaptation of the educational content of the graduating department to the requirements of the labor market.

**The object of research** - the formation of the input mathematical description of the information-analytical system of adaptation of the educational content of the department to the requirements of the labor market.

**The subject of the research** - the input educational matrix of the information-analytical system of adaptation of the educational content of the department to the requirements of the labor market.

**Research methods** - information theory and information-extreme intellectual technology of data analysis, methods of image recognition theory.

**Results** - The algorithm of training of analytical-information system and its program realization within the limits of information-extreme technology for adaptation of educational content to requirements of the labor market is developed. The algorithm was implemented using the C # programming language.

ANALYTICAL INFORMATION SYSTEMS, MACHINE LEARNING, BASIC  
MACHINE LEARNING ALGORITHMS, TRAINING MATRIX,  
MATHEMATICAL DESCRIPTION, INFORMATION OF EXTREME  
INTELLECTUAL TECHNOLOGIES, HIGH-QUALITY EDUCATIONAL  
CONTENT

## Зміст

|   |           |
|---|-----------|
| <b>ВСТУП.....</b>   | <b>6</b>  |
| <b>1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКИ ЗАДАЧІ ДОСЛІДЖЕННЯ</b>                               | <b>7</b>  |
| 1.1 Сучасний стан і стан розвитку комп'ютеризованих систем освіти .....                 | 7         |
| 1.2 Аналіз методів машинного навчання системи підтримки прийняття рішень в освіті ..... | 10        |
| 1.3 Постановка задачі .....   | 22        |
| <b>2 ОПИС МЕТОДУ ДОСЛІДЖЕНЬ .....</b>   | <b>24</b> |
| 2.1 Основні визначення інформаційно-екстремальної інтелектуальної технології .....      | 24        |
| 2.2 Математична модель машинного навчання.....  | 29        |
| 2.3 Оцінка функціональної ефективності машинного навчання Кульбака                      | 33        |
| <b>3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ.....</b>   | <b>35</b> |
| 3.1 Формування вхідної навчальної матриці .....   | 35        |
| 3.2 Алгоритм інформаційно-екстремального машинного навчання.....                        | 35        |
| 3.3 Короткий опис програмного забезпечення.....   | 42        |
| 3.4 Результати фізичного моделювання.....   | 43        |
| <b>ВИСНОВКИ .....</b>   | <b>51</b> |
| <b>СПИСОК ЛІТЕРАТУРИ .....</b>  | <b>52</b> |
| <b>ДОДАТОК.....</b>   | <b>54</b> |

## ВСТУП

Головною метою функціонування системи управління навчальним процесом у вищих навчальних закладах (ВНЗ), як і раніше, залишається підвищення рівня освіти за рахунок ефективної і скоординованої організації роботи з наданням можливості контролю, аналізу і коригування прийнятих управлінських рішень на базі об'єктивних результатів безперервного моніторингу кількісного і якісного рівня знань студентів. Оскільки одним з ефективних інструментів оцінки рівня знань в даний час є системи автоматизованого тестового контролю, актуальною є наукова проблема інтелектуалізації методологічних і технологічних інструментів тестування шляхом розробки і створення на їх основі експертних систем підтримки прийняття рішень з управління навчальним процесом у ВНЗ

Інститут вищої освіти є основоположним у розвитку сучасної науки і економіки в цілому, в зв'язку з чим підвищення якості освіти є найважливішою соціально-економічною проблемою для системи вищої професійної освіти. Сьогодні вимоги до якості підготовки фахівців в українських вищих школах формуються відносно міжнародних норм якості освіти, що пов'язано з процесами зближення і «гармонізацією» систем освіти різних країн. Крім того, якість освіти сьогодні НЕ відповідає вимогам споживача в обличчя роботодавців, заснованих на компетентності (знання, вміння, навички і здібності) у вирішенні конкретних виробничих завдань.

У магістерській роботі було розроблено комплекс моделей машинного навчання інформаційно-аналітичної системи адаптації навчального контенту до вимог ринку праці. На основі яких було розроблено і програмно реалізовано мовою C# алгоритм машинного навчання з оптимізацією контрольних допусків на ознаки розпізнавання.

# 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКИ ЗАДАЧІ ДОСЛІДЖЕННЯ

## 1.1 Сучасний стан і стан розвитку комп'ютеризованих систем освіти

Швидкий розвиток технологій сприяв до змін майже в усіх сферах суспільства. До них належить сфера освіти, яка нещодавно пережила складний перехідний період. Ми спостерігаємо стрімкий розвиток та практичне застосування дистанційної освіти в Україні. У той же час суспільство розраховує на впровадження нових концепцій у навчанні "протягом усього життя" [6]. Стандартні, досить статичні п'яти-шестирічні університетські курси не можуть повністю задовольнити мінливі потреби ринку праці. Інновації проникли у всі сфери життя, постійно змінюючи попит на різні професії, особливо професії. Тому навчання не може завершитися після п'ятого року університету, воно має тривати і надалі. Технологія дистанційної освіти має великий потенціал для відповіді на подібні соціальні виклики, забезпечуючи зручні механізми підтримки навчання протягом усього життя

За оцінками, середньорічний темп приросту нових знань становить 4-6%. Це означає, що фахівці повинні здобути приблизно 50% своїх професійних знань після закінчення навчання. Час, необхідний для відновлення професійних знань високоосвічених фахівців, становить 28% від загального часу, витраченого працівниками протягом усього робочого періоду. Постійне навчання стало важливим фактором конкурентоспроможності експертів на ринку праці.[4]. Вся історія становлення та розвитку навчання людини як самостійного напрямку його діяльності свідчить про те, що навчання повинно бути безперервним та адаптованим[5].



Сьогодні ми маємо безліч чудових прикладів Web-систем для організації дистанційного навчання, серед них Blackboard [1], WebCt [2], Moodle [3]. та ін.

Система Blackboard забезпечує єдине інтерактивне середовище для студентів, викладачів та персоналу для навчання, взаємодії та обміну інформацією. Система дає змогу керувати віртуальними навчальними середовищами, створювати електронні освітні ресурси, забезпечувати віддалений доступ до освітніх ресурсів навчальних закладів, контролювати навчальні процеси, забезпечувати платформу для дистанційних курсів навчання, накопичувати, організувати, керувати доступом, доповнювати навчальний фонд та забезпечити учасників засобами комунікації та інформації. Система Blackboard дозволяє автоматизувати наступні основні області діяльності вузу в освітньому процесі: підготовка освітніх матеріалів, дистанційне навчання, спільна науково-дослідна діяльність, облік і контроль персональних критеріїв освітнього процесу, ведення нормативно-довідкової інформації, спільна робота віддалених членів освітніх проектів.

До переваг цієї системи належать:

- Здатність працювати в єдиній системі різними мовами;
- Підтримка користувачів технічно та методично;
- Поширення рішень із забезпечення якості;
- Користування реалізацією системи в проектах з більш ніж 100 000 користувачів;
- швидка автоматизована підготовка звітів;
- Використовувати єдину централізовану базу даних;
- Інтеграція з одним каталогом користувачів.

WebCT (Web Course Tools) Кампус-версія - це платформа електронного навчання клієнт/сервера CMS, підтримувана системою управління реляційними базами даних (термінологія-глобальна база даних WebCT CE). База даних була реалізована за допомогою скриптів Perl.

Програмне забезпечення WebCT містить кілька функцій, включаючи:

- UNIX-сервери для різної кількості студентів / слухачів;
- надана можливість викладачам інструменти для високоякісних онлайн-курсів, включаючи тестоване створення шаблонів курсів, словники гіпертексту для термінів курсу, інструменти пошуку інформації в курсах на основі заданих шаблонів та бібліотеки мультимедійних файлів;
- on-line самотестування і тестування студентів / слухачів, їх доступ до результатів тестування (з візуалізацією помилок), моніторинг поточної академічної успішності, виставлення робіт на «електронну дошку оголошень», можливість зміни паролю;

Недоліки.

- Відсутність можливості створення особистих навчальних програм (для користувачів, групи користувачів).
- До недоліків можна віднести неможливість врахувати результати навчання електронних висловлювань у особистих файлах.

Moodle - це повністю відкритий і вільно розповсюджуваний проект.

Переваги Moodle:

- система спроектована з урахуванням досягнень сучасної педагогіки з акцентом на взаємодію та обговорення між учнями;
- можливо використовувати для дистанційного та денного навчання;
- має простий та ефективний мережевий інтерфейс;
- має великий набір мовних пакетів (43 мови) дозволяє досягти повного комфорту;
- зміни курсу з часу останнього входу користувача в систему, можуть відобразитися на першій сторінці курсу;
- всі оцінки завдань можна зібрати на одній сторінці (або як файл);

– має забезпечення повних звітів про вхід та роботу користувачів, а також графіки та детальну інформацію про роботу різних модулів.

Поки кожна з розглянутих систем дистанційного навчання має свої переваги та недоліки, досягаючи важливих функцій електронного навчання, але далеко не всі вони забезпечують складні процеси дистанційного навчання. Не всі системи підтримують такі важливі функції, як навчання, як підтримка звітів, оцінка якості вмісту освіти, розробка та модифікація розкладу та синхронізація календаря. Управління бюджетом, контроль за зарплатою та навчанням, інтеграція з платіжними системами, внутрішній та зовнішній облік транзакцій, інтеграція із системами виставлення рахунків (або вбудованими функціями виставлення рахунків та рахунків) також є дуже важливими для автоматизованих навчальних систем.

## **1.2 Аналіз методів машинного навчання системи підтримки прийняття рішень в освіті**

Машинне навчання або Machine learning – один з розділів AI, алгоритми, що дозволяють комп'ютеру робити висновки на підставі даних, не слідуючи жорстко заданим правилами.[15] Іншими словами, машина може знайти закономірності у складних і багатопараметричних завданнях (які людський мозок не може вирішити) та знайти більш точні відповіді. Результат - правильний прогноз.

Ціль машинного навчання - частково або й повністю автоматизувати рішення різних складних аналітичних задач.

Тому насамперед машинне навчання має на меті дати найбільш точні прогнози на основі даних про вхід, щоб власники бізнесу, маркетологи та працівники могли приймати правильні рішення на роботі.

В результаті тренування машина може передбачити результат, запам'ятати його, відтворити його при необхідності та вибрати найкраще з декількох варіантів.

В даний час машинне навчання охоплює широкий спектр застосувань: від банків, ресторанів, АЗС до роботів на виробництві. Нові завдання, які з'являються майже щодня, призводять до нових напрямків машинного навчання.

Машинне навчання базується на трьох опорах та найефективнішій взаємодії з клієнтами:

- дані – базова інформація, яку ми зазвичай вимагаємо від клієнтів. Сюди входять усі вибірки даних, для яких потрібна система навчання;
- ознаки – ця частина роботи виконується у тісній співпраці із замовниками. Ми визначаємо ключові вимоги бізнесу та спільно визначаємо, які особливості та ознаки система повинна відслідковувати через навчання;
- алгоритм - вибір методу вирішення бізнес-задач.

Дане завдання вирішується без участі замовника, силами наших співробітників.

За ознакою наявності вчителя, навчання ділиться на навчання з учителем (Supervised Learning), без вчителя (Unsupervised Learning) та з підкріпленням (Reinforcement Learning).

- Навчання з учителем застосовують коли потрібно навчити машину розпізнавати об'єкти або сигнали. Загальний принцип навчання з учителем це "дивись, ось це двері і це теж двері, і ось це теж двері"

- Навчання без вчителя використовує принцип "ця річ така ж як інші". Алгоритми вивчають подібності і можуть виявити відмінність і виконати виявлення аномалій, розпізнаючи, що є незвичайним або несхожим.

- Навчання з підкріпленням використовують там, де перед машиною стоїть завдання - правильно виконати поставлені завдання у зовнішньому середовищі маючи безліч можливих варіантів дії Наприклад в комп'ютерних іграх, трейдингових операціях, для безпілотної техніки.

За типом застосовуваних алгоритмів можна виділити два види:

– класичне навчання - відомі і добре вивчені алгоритми навчання, розроблені в основному більше 50-ти років тому для статистичних бюро. Підходить, насамперед, під завдання роботи з даними: класифікація, кластеризація, регресія і т.п. Застосовують для прогнозування, сегментації клієнтів і так далі.

– нейронні мережі і глибоке навчання - найбільш сучасний підхід до машинного навчання. Нейронні мережі застосовуються там, де потрібні розпізнавання або генерація зображень і відео, складні алгоритми управління або прийняття рішень, машинний переклад і подібні складні завдання.

При вмілому підході, комбінуючи різні види машинного навчання, можна домогтися автоматизації всіх рутинних бізнес-процесів. Іншими словами, роботи, підготовлені за допомогою машинного навчання, можуть виконувати всю рутинну роботу.

Оскільки завданням роботи є аналіз методів класифікації, то надалі розглядатимемо лише методи навчання з учителем [7].

Класифікатори можна розділити на параметричні і непараметричні. Параметричні методи включають, наприклад, метод максимальної правдоподібності, оскільки він працює в припущенні, що функція щільності ймовірності для кожного з класів забезпечується гаусовим розподілом. Непараметричні класифікатори, в свою чергу, не засновані на будь-яких припущеннях про розподіл вхідних даних. З огляду на той факт, що в більшості випадків функція розподілу невідома, непараметричні класифікатори поширені набагато ширше.

Важливим атрибутом класифікатора є не тільки здатність вводити дані (вихід класифікатора), що належать до певного класу, але й можливість визначати ймовірність приналежності до кожного класу, завдяки чому найбільш надійний клас можна легко вибрати. Такі особливості мають, наприклад, логістична регресія (логістична регресія). Отримані ймовірності в кожній категорії можуть бути використані для опрацювання результатів

класифікації, наприклад, комбінування результатів різних моделей у наборі, фільтрація шуму тощо.

Найпоширенішими методами машинного навчання для класифікаційних проблем є штучні нейронні мережі, логістична регресія, k-методи найближчого сусіда, методи підтримки векторної машини (SVM) та випадкові ліси.

Конфігурація штучної нейронної мережі сильно відрізняється. Тим не менш, мережева парадигма має багато спільного.

Відповідно до структури зв'язку між нейронами мережі та типом нейронів, що застосовуються, різниця між нейронними мережами - це тип топології.

Штучні нейронні мережі можна розглядати як спрямовані графіки із зваженими зв'язками, де штучні нейрони є вузлами. Відповідно до архітектури з'єднання, ШНМ можна розділити на два типи мереж прямого розподілу, де графи не мають циклів, мережі, що повторюються, або мережі з зворотним зв'язком.

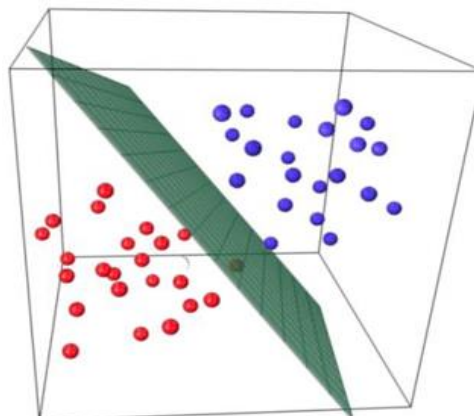
У випадку навчання нейронної мережі з вчителем на вхід подаються дані з деякого навчального набору. Потім сигнали підсумовуються і активують приховані нейрони [11]. Цей процес повторюється шар за шаром, поки не буде досягнутий вихідний шар. Сигнали нейронів можна трактувати як відповідь на якесь питання, наприклад, про приналежність зображення якогось класу. Якщо це відповідь правильна, то переходять до наступного зразка, в іншому випадку відбувається процес зворотного поширення помилки. Це можна трактувати так: учитель поставив свою оцінку за цю відповідь, і потрібно вивчити нове правило [11]. Якщо параметри нейронної мережі підібрані вірно, то, обробивши достатню кількість навчальних зразків на вході, нейронна мережа стає здатна класифікувати незнайомі об'єкти.

Логістична регресія (англ. logistic regression) - статистичний регресійний метод. Логістична регресія - це метод статистичної регресії. У

випадку, коли залежна змінна представлена різними категоріями, дослідники використовували логістичну регресію в проблемі класифікації, тобто вона може отримати лише обмежений набір значень (таких як 0 і 1). [12]

Порівняно із звичайною регресійною моделлю, метод логістичної регресії не передбачає значення числової змінної на основі вибірки вихідного значення незалежної змінної. Натомість значення функції - це ймовірність того, що вихідне значення належить до однієї з категорій. Для спрощення проблеми, припустимо, у нас є лише дві категорії, і ймовірність, яку слід визначити, - це ймовірність того, що значення належить до категорії "+". Тому результати та ймовірності логістичної регресії завжди знаходяться в межах  $[0, 1]$ .

Основна суть логістичної регресії полягає в тому, що простір початкового значення незалежної змінної можна розділити на дві категорії лінійною площиною (тобто прямою лінією). Точніше, лінійна площина з двома параметрами (двовимірною) являє собою пряму лінію, яка розділяє категорії (не вигнуті). За трьома вимірами це площини тощо. Це обмеження залежить від доступних вихідних даних та алгоритмів навчання. Для того, щоб все працювало належним чином, точки вихідних даних необхідно розділити на дві вищевказані області через лінійну площину. Якщо точки джерела відповідають цій вимозі, їх можна назвати лінійним поділом. Візуально це показано на рис. 1.1.



### Рисунок 1.1- Візуалізація класифікатора

як показано на малюнку. Площина 1.1 називається лінійною дискримінантною. За своєю функцією та моделлю вона лінійна, а також розділяє або розрізняє точки на різні класи. Якщо лінійний розподіл точок у вихідному просторі неможливий або вони нероздільні лінійно, вам слід спробувати перетворити векторний елемент у більший простір, тим самим додавши додаткові ефекти взаємодії, більш високий ступінь належності тощо. Цей метод називається "фокус ядра". У цьому випадку використання лінійного алгоритму дає деякі переваги для вивчення нелінійних функцій, оскільки межа стає нелінійною при поверненні до вихідного простору.

Метод  $k$  найближчого сусіда - це алгоритм машинного навчання, який використовується для автоматичної класифікації об'єктів на основі показників. Суть методу найближчого сусіда полягає у віднесенні об'єкта до найпоширенішого класу серед сусідніх елементів елемента. Знаходження сусідів на основі сукупності об'єктів, клас яких відомий, і використовує гіперпараметр  $k$ . Цей гіперпараметр обчислює, який клас має найбільше серед сусідів. Кожен об'єкт має обмежену кількість атрибутів. Це алгоритм навчання з учителем, тому для роботи алгоритму потрібна розмічена вибірка даних.[13]

Метод  $k$  найближчого сусіда - непараметричний метод класифікації даних. Для вирішення задачі класифікації об'єктів у атрибутивному просторі для всіх інших об'єктів використовуються різні відстані (евклідова відстань, відстань Мінковського). Вибираються об'єкти з найменшою відстанню та об'єднуються в окремих класів. Приклад роботи класифікатора наведено на рис. 1.2.



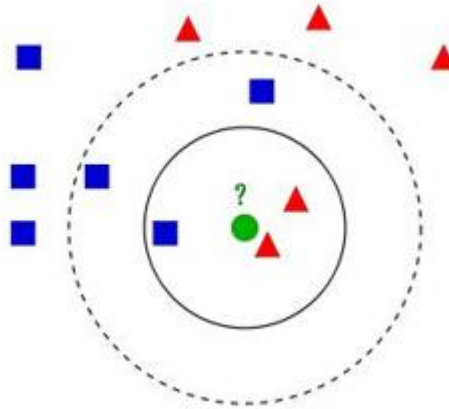


Рисунок 1.2- Візуалізація класифікатора

Приклад включає дві категорії: сині квадрати (категорія 1) та червоні трикутники (категорія 2). Потрібно класифікувати зелене коло в одну з наступних категорій. Якщо гіперпараметр  $k = 3$ , зелене коло буде віднесено до другої категорії, оскільки менший круг має два трикутники і лише один квадрат. Якщо гіперпараметр  $k = 5$ , коло буде віднесено до другої категорії, тому що у більшому колі навпроти двох трикутників є три квадрати.

Алгоритм можна застосовувати до зразків з великою кількістю атрибутів (багатовимірних). З цієї причини перед використанням необхідно визначити функцію відстані. Класичний варіант визначення відстані - це відстань в евклідовому просторі.

Однак різні атрибути можуть мати різні діапазони значень, представлені у вибірці (наприклад, атрибут А вказує діапазон від 0,1 до 0,5, а атрибут В позначає діапазон від 1000 до 5000), а значення відстані значною мірою залежить від широкого спектру ознак. Тому, як правило, дані нормалізуються. У кластерному аналізі є два основні методи стандартизації даних: Міні-макс нормалізація:

$$x' = (x - \text{MIN}[X]) / (\text{MAX}[X] - \text{MIN}[X])$$

В цьому випадку всі значення будуть лежати в діапазоні від 0 до 1. Дискретні бінарні значення визначаються як 0 і 1.

Z-нормалізація:

$$x' = (x - M[X]) / \sigma[X]$$

### Виділення значущих атрибутів

Деякі важливі атрибути можуть бути важливішими, ніж інші, тому кожному атрибуту може бути присвоєно певне політичне значення (наприклад, обчислене за допомогою вибіркового тесту та оптимізації помилок упередженості). Тому кожному атрибуту  $k$  буде присвоєна вага, так що значення атрибута потрапить у  $[0; \max ()]$  (значення, нормалізоване методом  $\text{mini-max}$ ). Наприклад, якщо вага, присвоєна атрибуту, становить 2,7, нормоване зважене значення буде  $[0; 2,7]$ .

Цей метод враховує не тільки кількість певних категорій у даній місцевості, але і їх віддаленість від нового значення. Для кожного класу  $j$  визначте близьку оцінку:

$$Q_j = \sum_{i=1}^n \frac{1}{d(x, a_i)^2}$$

Тут  $d(x, a)$  - дистанція від нового значення  $x$  до об'єкта  $a$ .

У якого класу вище значення близькості, той клас і присвоюється новому об'єкту [7].

Метод векторної машини підтримки (SVM) - це аналогічний набір алгоритмів навчання, заснованих на вчителях, для класифікаційних та регресійних аналізів [14], він належить до сімейства лінійних класифікаторів, а також може розглядатися як особливий випадок регуляризації Тихонова.

Однією з характеристик методу опорного вектора є постійне зменшення помилки емпіричної класифікації та збільшення відстані (запасу), тому цей метод також називають методом класифікатора з найбільшою відстані.

Принцип роботи такий:

Враховуючи набір прикладів навчання, кожному прикладу навчання присвоюється належність до тієї чи іншої з двох категорій, алгоритм навчання SVM створює модель, яка призначає новий приклад тієї чи іншої категорії, що

робить неможливим бінарний лінійний класифікатор (хоча існують такі методи, як Плат-маштаб, які використовують SVM для імовірнісної класифікації). Модель SVM представляє приклади як точки в просторі, і її фіксованим способом є поділ прикладів кожної категорії на максимально можливий чіткий поділ. Потім поміщають нові дані в той же простір і припускають, що вони належать до тієї категорії, на якій засновані навчальні зразки.

Окрім виконання лінійної класифікації, SVM також може ефективно виконувати нелінійну класифікацію, використовуючи так звані «трюки ядра», які неявно відображатимуть її вхід у більшому просторі.

Якщо дані не марковані, система не може вчитися з викладачем, і вам потрібно скористатися спонтанним навчанням, яке намагається знайти природні кластери цих груп, а потім класифікувати нові дані в ці сформовані групи. Тобто, фактично, проблема кластеризації спочатку вирішується методом (метод k-означає, метод ієрархічної кластеризації), а потім безпосередньо вирішується проблемою класифікації. Алгоритм векторного кластеризації, розроблений Гавою Сігельманом та Володимиром Вапником, використовує векторну статистику підтримки, розроблену в алгоритмі машини підтримки векторів, для класифікації немечених даних і є одним з найбільш широко використовуваних алгоритмів кластеризації в промислових цілях.

Приклад роботи класифікатора наведено на рис. 1.3.

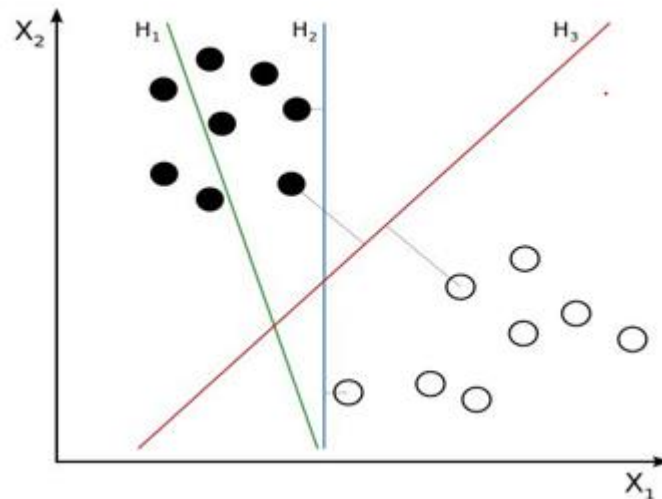


Рисунок 1.3- Візуалізація класифікатора

Гіперплощина  $H_1$  не є роздільною. Гіперплощина  $H_2$  є роздільною, але не з максимальним розділенням.  $H_3$  є роздільною гіперплощиною із максимальним розділенням.

Випадкові ліси (Random Forest) або ліси випадкових рішень - це метод вивчення класифікації, регресії та інших завдань шляхом побудови декількох дерев рішень під час навчання та виведення класу як одного з класів (класифікації) або середнього прогнозування (регресія) Працювати. ) Єдиного дерева. Рішення дерев випадкового лісу вирішують проблему перенавчання окремих дерев рішень на навчальній вибірці. Перший алгоритм Random Forest був створений Тін Кам Хо з використанням методу випадкового підпростору, який, за формулюванням Хо, є способом реалізації підходу "стохастичної дискримінації" до класифікації, запропонованої Євгеном Клейнбергом.

Розширення цього алгоритму було розроблено Лео Брайманом та Адель Катлер, "Випадковий ліс" є їх торговельною маркою. Розширення поєднує в собі концепцію "сумки" Бреймена та випадково вибрані функції, які спочатку запропонували Хо, а потім самі Аміт та Джеман, щоб створити колекцію дерев рішень з контрольованою дисперсією.

Дерево рішення - популярний метод для виконання різних завдань машинного навчання. Вузьке дослідження дерев, "близьке до виконання вимог як незавершена процедура виявлення даних", - стверджує Хасти та співавтори, - "оскільки вона є інваріантною при масштабуванні та інших різноманітних перетвореннях значень ознак, є надійною до включення невідповідних функцій, а також створює моделі для перевірки, однак вони рідко точні."

Зокрема, дерева з більшою глибиною, як правило, вивчають дуже нерегулярні структури: вони накладають навчальний набір, тобто рівень зсуву низький, але мінливість дуже висока. Випадковий ліс - це метод усереднення декількох дерев рішень, що навчаються в різних регіонах однієї і тієї ж навчальної комбінації для зменшення відмінностей. Це пов'язано з незначним збільшенням упередженості та деякими втратами в інтерпретації, але зазвичай значно покращить ефективність кінцевої моделі.

Випадкові ліси можна використовувати для класифікації значень змінних регресії або для їх класифікації природним шляхом. Наступні методи описані в оригінальній документації Braiman та реалізовані в програмному пакеті R randomForest.

Перший крок вимірювання змінних значень у наборі даних - це узгодження випадкового лісу з даними. Під час створення випадкових лісів помилка кожної точки даних записується та усереднюється в лісі (якщо під час тренування не використовується упаковка, помилку на незалежному тестовому наборі можна замінити). Для того, щоб виміряти важливість  $j$ -ї функції після тренування, значення  $j$ -ї функції змінюється між навчальними даними, і помилка знову перераховується на цьому наборі даних .

Шляхом усереднення різниці помилок до та після розташування всіх дерев можна оцінити важливість  $j$ -ої функції.

Стандартизуйте оцінки за стандартним відхиленням цих відмінностей.

З цієї причини важливіше оцінювати функції, які забезпечують більшу цінність, ніж функції, які надають меншу цінність.

Русін Чжу дав статистичне визначення змінної та проаналізував її. Цей спосіб визначення змінних значень має деякі недоліки. Для даних (включаючи категоріальні змінні з різними рівнями чисел), для цих атрибутів з більш високими рівнями використовують випадкові перекошені ліси. Для вирішення цієї проблеми можна використовувати такі методи, як часткова заміна та вирощування дерев злиття. Якщо дані містять комбіновану функціональну групу, схожу на результат, менша група вибере більшу групу.

#### Переваги і недоліки

Серед інших методів пошуку даних дерева рішень мають різні переваги:

- просто зрозуміти та пояснення.
- здатний обробляти як цифрові, так і категоричні дані.
- не вимагає підготовки даних.
- використовує модель білої коробки.
- є змога використовувати статистичні тести для тестування моделі.
- підходить для великих наборів даних.
- більш відображає рішення людей, ніж інші методи.
- дерево рішення може бути наближене будь-якою булевою функцією рівняння.

До недоліків можна віднести:

- Дерева можуть бути дуже нестійкими. Незначні зміни в даних про навчання можуть спричинити значні зміни в дереві, що може призвести до остаточного прогнозу.
- Створити надскладне дерево, придатне для навчальних даних у загальному вигляді, легко, але важко передбачити дані тестів [13].

### 1.3 Постановка задачі

Структура інтелектуальної аналітичної системи складається з системи збору інформації та формування векторів розпізнавання і здатної навчатися системи підтримки прийняття рішень (СППР).

Розглянемо в рамках ІЕІ-технології формалізовану задачу синтезу здатної навчатися СППР для оцінки навчального контенту освітньої програми напрямку «Інформатика».

Дано алфавіт класів розпізнавання  $\{X_m \mid m = \overline{1, M}\}$ , які характеризують функціональні стани навчального процесу, і сформована навчальна матриця  $\|y_{(m,i)}^{(j)} \mid i = \overline{1, N}, j = \overline{1, n}\|$ , де  $N$  – кількість ознак розпізнавання, якими є експертні оцінки змістовних модулів навчальних дисциплін нормативної та варіативної частин навчального плану підготовки фахівців напрямку «Інформатика» рівня бакалавр, що виставлялися респондентами,  $n$  – кількість респондентів. Крім того, дано структурований вектор просторово-часових параметрів функціонування

$$g = \langle g_1, \dots, g_\xi, \dots, g_\Xi \rangle,$$

які впливають на функціональну ефективність здатної навчатися СППР, з відповідними на них обмеженнями  $R \cap (g_1, \dots, g_\Xi) \leq 0$ .

В процесі навчання було визначено оптимальні значення координат вектора параметрів функціонування  $\{g_\xi^*\}$ , які забезпечують максимум інформаційного ентропійного критерію оптимізації в області визначення його функції:

$$E_m^* = \max_G E_m,$$

де  $E_m$  – інформаційний критерій оптимізації параметрів машинного

навчання інформаційно-аналітичної системи розпізнавати реалізації класу  $X_t^o$ ;

$G$  - область допустимих значень параметрів функціонування СППР.

На етапі екзамену, тобто в режимі безпосереднього оцінювання, необхідно прийняти рішення про належність реалізації, що розпізнається, до одного із класів розпізнавання із заданого алфавіту, що і є основною задачею даної роботи.

Щоб вирішити проблему на основі даних, отриманих після запуску системи в режимі навчання, необхідно виконати такі завдання:

- сформуванати навчальну матрицю аналітично-інформаційної системи;
- обрати математичну модель для функціонування аналітично-інформаційної системи в режимі навчання;
- оцінити функціональну ефективність інтелектуальної аналітично-інформаційної системи.
- розробити та програмно реалізувати алгоритми викладання аналізу та знань частини інформаційної систем шляхом оптимізації основних функціональних параметрів;



## 2 ОПИС МЕТОДУ ДОСЛІДЖЕНЬ

### 2.1 Основні визначення інформаційно-екстремальної інтелектуальної технології

Основна ідея машинного навчання у рамках ІЕІ-технології полягає в трансформації апріорного у загальному випадку нечіткого розбиття простору ознак у чітке розбиття класів еквівалентності шляхом ітераційної оптимізації параметрів функціонування ІС. При цьому здійснюється цілеспрямовано пошук глобального максимуму багатоекстремальної функції статистичного інформаційного критерію в робочій (допустимій) області її визначення і одночасного відновлення оптимальних роздільних гіперповерхонь, що будуються в радіальному базисі бінарного простору ознак розпізнавання.

Відмінністю методів ІЕІ-технології є те, що трансформація вхідного нечіткого розподілу реалізацій образів в чітке здійснюється в процесі оптимізації системи контрольних допусків, що приводить до цілеспрямованої зміни значень ознак розпізнавання і дозволяє побудувати безпомилкові за багатовимірною навчальною матрицею вирішальні правила.[8]

Тому в рамках ІЕІ-технології можна поєднати стандартизацію зображень, яка включає корекцію апріорної деформації зображення відносно еталонного зображення та етап навчання безпосередньо встановлення визначальних правил.

Нехай відомий алфавіт класів розпізнавання  $\{X_m^o \mid m = \overline{1, M}\}$ . У загальному випадку при прийнятті гіпотези нечіткої компактності реалізацій образу розбиття простору ознак на класи розпізнавання є нечітким розбиттям  $\tilde{\mathfrak{R}}^{|M|}$ , яке відповідає умовам:

$$\begin{aligned} & (\forall X_m^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq \emptyset]; \\ & (\exists X_k^o \in \tilde{\mathfrak{R}}^{|M|}) (\exists X_l^o \in \tilde{\mathfrak{R}}^{|M|}) [X_k^o \neq X_l^o \rightarrow X_k^o \cap X_l^o \neq \emptyset]; \\ & (\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|}) (\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|}) [X_k^o \neq X_l^o \rightarrow \text{Ker} X_k^o \cap \text{Ker} X_l^o = \emptyset]; \end{aligned}$$

$$\bigcup_{x_m^o \in \tilde{\mathfrak{R}}} X_m^o \subseteq \Omega_\varepsilon; \quad k \neq l; \quad k, l, m = \overline{1, M} \quad (2.1)$$

При цьому елементи розбиття  $\tilde{\mathfrak{R}}^{(M)}$  є нечіткими класами розпізнавання.[8]

У просторі двійкових ознак формою найкращого контейнера для ідентифікації класу є суперпаралельний шестигранник. З метою узагальнення та сприяння побудові такого контейнера ми припускаємо, що існує "псевдосуперсфера", яка описує суперпаралельний шестигранник, тобто, включаючи всі його вершини.

Це дозволяє додатково розглянути наступні параметри оптимізації контейнера на радіальній основі простору функції  $\Omega$ , наприклад, опорний вектор, наприклад,  $x_m \in X_m^o$ , його вершина визначає геометричний центр контейнера  $K_m^o$ , та радіус псевдосферичного контейнера, визначений формулою в просторі Хеммінга:

$$d_m = \sum_{i=1}^N (x_{m,i} \oplus \lambda_i), \quad (2.2)$$

де  $x_{m,i}$  –  $i$ -та координата еталонного вектора  $x_m$ ;  $\lambda_i$  –  $i$ -та координата деякого вектора  $\lambda$ , вершина якого належить контейнеру  $K_m^o \in X_m^o$ .

Надалі, з метою спрощення, кодова відстань (2.2), наприклад, між векторами  $x_m$  і  $\lambda$  буде позначатися у вигляді  $d_m = d(x_m \oplus \lambda)$ .

За ІЕІ-технологією відновлення оптимального контейнера в радіальному базисі, наприклад,  $K_m^o$  здійснюється шляхом його цілеспрямованої послідовної трансформації в гіперсферичний габарит, радіус  $d_m$  якого збільшується на кожному кроці навчання за рекурентною процедурою:

$$d_m(k) = [d_m(k-1) + h] \quad d_m(k) \in G_m^d,$$

де  $k$  – змінна числа збільшень радіуса контейнера  $K_m^o$ ;  $h$  – крок збільшення радіуса;  $G_m^d$  – область допустимих значень радіуса  $d_m$ .

Нехай класи  $X_k^o$  і  $X_l^o$  є «найближчими сусідами», тобто мають серед усіх класів найменшу міжцентрову відстань  $d(x_k \oplus x_l)$ , де  $x_k, x_l$  – еталонні вектори відповідних класів. Тоді за ІЕІ-технологією з метою запобігання «поглинання» одним класом ядра іншого класу умови (2.1) доповнюються таким предикатним виразом:

$$\left( \forall X_k^o \in \tilde{\mathfrak{R}}^{(M)} \right) \left( \forall X_l^o \in \tilde{\mathfrak{R}}^{(M)} \right) [X_k^o \neq X_l^o \rightarrow (d_k^* < d(x_k \oplus x_l)) \& (d_l^* < d(x_k \oplus x_l))],$$

де  $d_k^*, d_l^*$  – оптимальні радіуси контейнерів  $K_k^o$  і  $K_l^o$  відповідно.

Алгоритм навчання за ІЕІ-технологією полягає в реалізації багатоциклічної ітераційної процедури оптимізації структурованих просторово-часових параметрів функціонування ІС шляхом пошуку глобального максимуму усередненого за алфавітом  $\{X_m^o\}$  значення КФЕ навчання.

Нехай вектор параметрів функціонування ІС у загальному випадку має таку структуру:

$$g = \langle g_1, \dots, g_{\xi_1}, \dots, g_{\Xi_1}, f_1, \dots, f_{\xi_2}, \dots, f_{\Xi_2} \rangle, \Xi_1 + \Xi_2 = \Xi \quad (2.3)$$

де  $\langle g_1, \dots, g_{\xi_1}, \dots, g_{\Xi_1} \rangle$  – генотипні параметри функціонування ІС, які впливають на параметри розподілу реалізацій образу;  $\langle f_1, \dots, f_{\xi_2}, \dots, f_{\Xi_2} \rangle$  – фенотипні параметри функціонування ІС, які прямо впливають на геометрію контейнера класу розпізнавання.

При цьому відомі обмеження на відповідні параметри функціонування:

$$R_{\xi_1}(g_1, \dots, g_{\xi_1}, \dots, g_{\Xi_1}) \leq 0; \quad R_{\xi_2}(f_1, \dots, f_{\xi_2}, \dots, f_{\Xi_2}) \leq 0$$

У рамках методології об'єктно-орієнтованого проектування подамо тестовий алгоритм навчання за ІЕІ-технологією для загального випадку ( $M > 2$ )

як ієрархічну ітераційну процедуру оптимізації структурованих просторово-часових параметрів (2.3) функціонування ІС:

$$\begin{aligned}
 & (\forall g_{\xi_1} \in G) (\forall f_{\xi_2} \in \mathcal{G}) (\exists g_{\xi_1} \in G_{\xi_1}) \{ \text{if } E^* = \max_{\xi_2} \bar{E} \text{ then} \\
 & g_{\xi_1}^* = \arg \langle \max_{\xi_1} \dots [ \max_{\xi_2} [ \max_{F_{\xi_2}} [ \dots [ \max_{F_1} \bar{E} ] \dots ] ] \dots \rangle \text{ else} \\
 & (\text{if } \xi_1 \leq \bar{\xi}_1 \text{ then } \xi_1 = \xi_1 + 1 \text{ else STOP} \}, \xi_1 = 1, \bar{\xi}_1 = \bar{\xi}_1, \xi_2 = 1, \bar{\xi}_2 = \bar{\xi}_2,
 \end{aligned}$$

де  $G_{\xi_1}, \dots, G_1$  – області допустимих значень відповідних генотипних

параметрів навчання;

$$\bar{E} = \frac{1}{M} \sum_{m=1}^M E_m -$$

усереднене значення КФЕ навчання ІС;  $G_E$  – область значень функції інформаційного КФЕ навчання;  $g_{\xi_1}^*$  – оптимальне значення параметра навчання, яке визначається у зовнішньому циклі ітераційної процедури оптимізації;

$F_{\xi_2}, \dots, F_1$  – області допустимих значень відповідних фенотипних параметрів навчання. Тут  $E_m$  – інформаційний КФЕ навчання ІС розпізнавати реалізації класу  $X_m^o$ .

Алгоритм навчання технології ІЕІ полягає в реалізації багатоциклового ітераційного процесу оптимізації структурованих просторово-часових параметрів ІС шляхом пошуку глобального максимального значення середнього букви середнього навчального КФЕ. Такими параметрами, наприклад, для гіперсферичних контейнерів класів є їх радіуси. Генотипи включають параметри навчання, які безпосередньо впливають на розподіл методів реалізації класу (наприклад, контрольні допуски для виявлення ознак, узгодження рівнів вибору з посиленням на двійкові вектори, параметри оптимізації функцій словникового запасу, навчальні плани, параметри навколишнього середовища тощо). Послідовна оптимізація кожного з цих параметрів може збільшити значення максимальної підготовки КФЕ, тим самим підвищивши загальну ймовірність прийняття правильного рішення на іспиті. Обов'язковим процесом алгоритму навчання ІЕІ є оптимізація допуску

керування, і його контрольне значення безпосередньо впливає на значення відповідної функції розпізнавання та параметр розподілу зображення. [8]

При порівняльному розпізнаванні ( $M = 2$ ) воно відбувається шляхом порівняння розпізнаного зображення з еталонним зображенням і виникає в таких завданнях, як розпізнавання персоналу, керована авіація та автоматична настройка класифікації. В ньому використовується ітеративний алгоритм навчання за ІЕІ-технологією має такий структурований вигляд:

$$\begin{aligned}
 & (\forall g_{z_1} \in g) (\forall f_{z_1} \in g) (\exists g_{z_1} \in G_{z_1}) \{ \text{if } E_1^* = \max_{G_2} E_1 \text{ then} \\
 & g_{z_1}^* = \arg < [\max_{G_1} [\dots [\max_{G_1} [\max_{F_2} [\dots [\max_{F_1} E_1] \dots]]]]] > \\
 & \text{else (if } \xi_1 \leq \bar{\Xi}_1 \text{ then } \xi_1 = \xi_1 + 1 \text{ else STOP)} \}, \\
 & \xi_1 = \overline{1, \bar{\Xi}_1}, \xi_2 = \overline{1, \bar{\Xi}_2},
 \end{aligned}$$

де  $E_1$  – інформаційний КФЕ навчання ІС розпізнавати реалізації еталонного класу  $X_1^o$ .

За умови обґрунтування гіпотези компактності (чіткої або нечіткої) основна ідея навчання за технологією ІЕІ - це розповсюдження апріорного виміру зображення шляхом цілеспрямованого перетворення зображення, щоб нормалізувати вхідний математичний опис ІС, щоб максимально побудувати радіальну основу в процесі навчання. Захоплення категорії контейнерів. Найкращий контейнер для технології ІЕІ забезпечує найбільшу різноманітність між суміжними класами, а його показник дорівнює максимальному значенню інформаційного навчання КФЕ в робочій зоні, де визначається його функція. Оптимальні геометричні параметри контейнера, отримані в процесі навчання технології ІЕІ, дозволяють інспекції приймати рішення на основі відносно простих детермінованих правил, що важливо для реалізації алгоритмів прийняття рішень в режимі реального часу. Повна надійність класифікатора близька до максимальної асимптотики, що залежить від ефективності навчального процесу. Поки статистична стабільність та статистична однорідність матриць освіти та екзамену однакові, асимптотична надійність може бути досягнута при обстеженні. У процесі функціонального

та статистичного тестування ця умова безпосередньо досягається при навчанні ІС.

Визначивши тенденцію асимптотично точних особливостей навчального процесу, можна досягти мети оптимізації просторово-часових параметрів ІС на основі технології ІЕІ [10-11].

## 2.2 Математична модель машинного навчання

В рамках ІЕІ-технології розглянемо математичну модель, здатної навчатися СППР у вигляді категорійної моделі, яка відображає за допомогою відповідних операторів множини, задіяні при функціонуванні системи, одна на одну. [10] Вхідний математичний опис СППР, що навчається за інформаційно-екстремальним алгоритмом, може бути виражений у вигляді структури теорії множин.

$$\Delta_B = \langle G, T, \Omega, Z, Y; \Pi, \Phi_1, \Phi_2 \rangle \quad (2.5)$$

де  $G$  – простір вхідних факторів, які впливають на об'єкт керування;

$T$  – множина моментів зняття інформації;

$\Omega$  – простір ознак розпізнавання;

$Z$  – простір можливих функціональних станів;

$Y$  – вибіркова множина, на основі якої формується навчальна матриця;

$P : G \times T \times \Omega \rightarrow Z$  – оператор зміни станів керованого процесу під впливом внутрішніх та зовнішніх факторів;

$\Phi : G \times T \times Z \times \Omega \rightarrow Y$  – оператор формування вибіркової множини.

Розглянемо категорійну модель навчання ІАС за базовим алгоритмом у вигляді діаграми відображення множин. При обґрунтуванні гіпотези нечіткої компактності має місце нечітке розбиття  $\tilde{\mathfrak{R}}^{[M]} \subset \Omega$ .

Введемо оператор  $\theta$  нечіткої факторизації простору ознак:  $\theta: Y \rightarrow \tilde{\mathfrak{R}}^{l|l}$  і оператор класифікації  $\psi: \tilde{\mathfrak{R}}^{l|l} \rightarrow I^{l|l}$ , який перевіряє основну статистичну гіпотезу про належність реалізацій  $\{x_m^{(j)} \mid j=\overline{1, n}\}$  нечіткому класу  $X_m^o$ . Тут  $l$  – кількість статистичних гіпотез. Оператор  $\gamma: I^{l|l} \rightarrow \mathfrak{S}^{l|l}$  шляхом оцінки статистичних гіпотез формує множину точнісних характеристик  $\mathfrak{S}^{l|l}$ , де  $q = l^2$  – кількість точнісних характеристик. Оператор  $\Phi: \mathfrak{S}^{l|l} \rightarrow E$  обчислює множину значень інформаційного ЕКО, який є функціоналом точнісних характеристик. Контур оптимізації геометричних параметрів нечіткого розбиття  $\tilde{\mathfrak{R}}^{l|l}$  шляхом пошуку максимуму ЕКО навчання розпізнаванню реалізацій класу  $X_m^o$  замикається оператором  $\Gamma: E \rightarrow \tilde{\mathfrak{R}}^{l|l}$  [11].

Структурна діаграма відображень множин у процесі навчання за базовим інформаційно-екстремальним алгоритмом зображена на рис. 2.1.

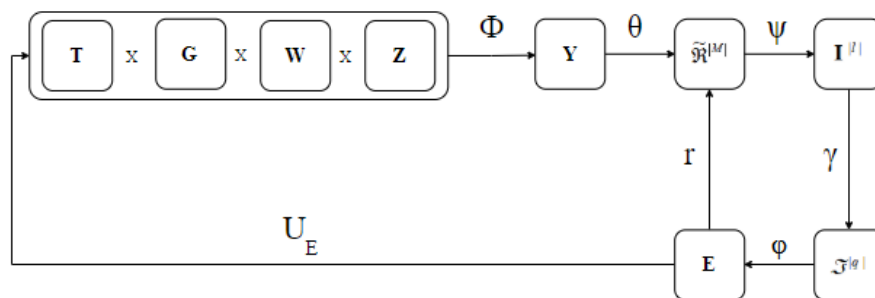


Рисунок 2.1 – Структурна діаграма відображень множин у процесі машинного навчання

Контур операторів, який зображено на рис. 2.2, оптимізує геометричні параметри розбиття  $\tilde{\mathfrak{R}}^{l|l}$ .

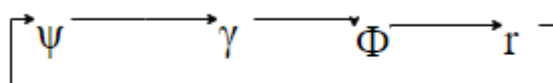


Рисунок 2.2 – Оператори, що оптимізують геометричні параметри системи

Оператор  $U_E : E \rightarrow G \times T \times \Omega \times Z$  регламентує процес навчання і дозволяє оптимізувати параметри його плану, які визначають, наприклад, обсяг і структуру випробувань, черговість розгляду класів розпізнавання та інше.

На рис. 2.3 показано категорійну модель оптимізації параметрів функціонування в процесі навчання СППР з оптимізацією всіх можливих параметрів її функціонування (системи контрольних допусків, рівня селекції, словника ознак, вибір форми контейнерів класів розпізнавання, просторовочасових характеристик).

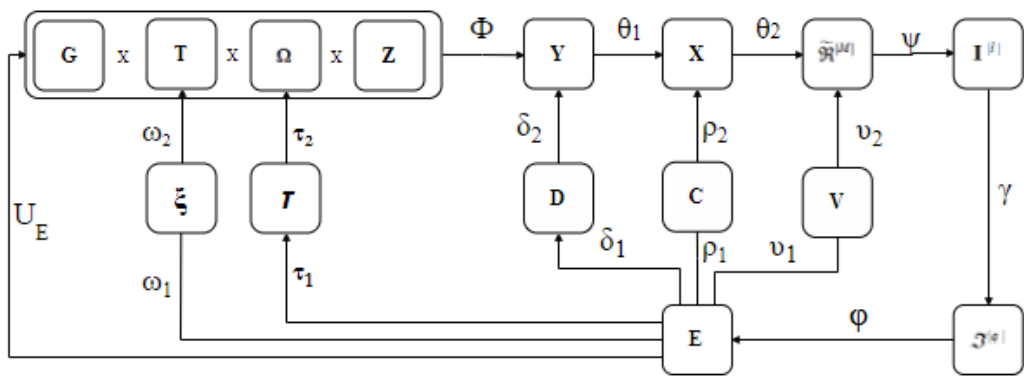


Рисунок 2.3 – Структурна діаграма відображень множин у процесі машинного навчання з оптимізацією всіх параметрів

Формування бінарної матриці  $X$  з вибіркової множини значень параметрів  $Y$  за допомогою оператора  $\theta_1 : Y \rightarrow X$ . відновлення на кожному кроці навчання оптимального в інформаційному розумінні розбиття простору ознак на  $M$  класів розпізнавання за допомогою оператора  $\theta_2 : X \rightarrow \tilde{\mathfrak{R}}^{|M|}$ . Основна статична гіпотеза про належність векторареалізації  $y_m^{(j)}$  до класу розпізнавання  $X_m^0$  за допомогою оператора класифікації  $\psi : \tilde{\mathfrak{R}}^{|M|} \rightarrow I^{|l|}$ , де  $I^{|l|}$  – множина  $l$  статистичних гіпотез. Множини точнісних характеристик  $\mathfrak{Z}^{|q|}$  формуються за допомогою оператора  $\gamma : I^{|l|} \rightarrow \mathfrak{Z}^{|q|}$ ; Оператор  $\Phi : \mathfrak{Z}^{|q|} \rightarrow E$  обчислення та формування терм-множини значень інформаційного ентропійного критерію оптимізації  $E$ .



В діаграмі відображення множин (рис. 2.3) показано контури оптимізації основних параметрів структурованого вектору  $g$ :

контур оптимізації геометричних габаритів  $V$  контейнерів класів розпізнавання, які безпосередньо формують розбиття  $\tilde{\mathfrak{R}}^{|M|}$  за допомогою оператора  $v : E \rightarrow \tilde{\mathfrak{R}}^{|M|}$ , який є композицією операторів  $v = v_1 \circ v_2$ , де оператор  $v_1$  здійснює вибір типу вирішальних правил, а оператор  $v_2$  – змінює геометричні параметри розбиття  $\tilde{\mathfrak{R}}^{|M|}$  залежно з вибраним типом вирішальних правил:

- контур оптимізації системи контрольних допусків  $D$  на ознаки розпізнавання за допомогою оператора  $\delta : E \rightarrow D$ , який є композицією операторів  $\delta = \delta_1 \circ \delta_2$ , де оператор  $\delta_1$  здійснює ініціалізацію масиву СКД, а оператор  $\delta_2$  – виконує оптимізацію залежно з вибраним типом;

- контур оптимізації часових параметрів аналізу вхідних даних  $T$  на ознаки розпізнавання за допомогою оператора  $\tau : E \rightarrow T$ , який є композицією операторів  $\tau = \tau_1 \circ \tau_2$ , де оператор  $\tau_1$  здійснює оптимальне розбиття інтервалів росту, а оператор  $\tau_2$  – визначає період дискретизації датчиків;

контур оптимізації рівня селекції двійкових векторів  $\rho$  за

допомогою оператора  $\rho : E \rightarrow C$  який є композицією операторів  $\rho = \rho_1 \circ \rho_2$ , де оператор  $\rho_1$  здійснює ініціалізацію масиву значень рівня селекції, а оператор  $\rho_2$  – виконує оптимізацію залежно з вибраним типом;

- контур оптимізації словника ознак за допомогою оператора  $\omega : E \rightarrow \Sigma$ , який є композицією операторів  $\omega = \omega_1 \circ \omega_2$ , де оператор  $\omega_1$  здійснює вибір правила перебору ознак, а оператор  $\omega_2$  – визначає тип ознаки, що тестується.

## 2.3 Оцінка функціональної ефективності машинного навчання Кульбака

Розглянемо модифікацію диференціальної інформаційної міри Кульбака, яка виражається як добуток відношення ймовірності  $\Lambda$ , тобто відхилення відповідного розподілу ймовірностей.

Згідно із принципом Лапласа-Бернуллі, що  $p(\mu_m) = p(\mu_c) = 0,5$ , і після переозначення апіорних умовних імовірностей відповідними точнісними характеристиками загальна міра Кульбака остаточно набирає вигляду:

$$E_{K_m}^{(k)} = \log_2 \frac{P_{i,m}^{(k)}}{P_{f,m}^{(k)}} * [P_{i,m}^{(k)} - P_{f,m}^{(k)}] = \log_2 \left( \frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * [1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))] \quad (2.4)$$

Оптимізуючи параметри функціонування СПР під час навчання навчання за ІЕІ-технологією нормування критеріїв оптимізації не є обов'язковим, оскільки вона вирішує екстремальне значення параметрів навчання, що відповідає глобальному максимальному значенню КФЄ, у визначеній робочій області. Але нормування критеріїв оптимізації є доцільним при порівняльному аналізі результатів досліджень і при оцінці ступеня близькості реальної СПР до потенційної.

Розглянемо процедуру обчислення модифікації диференціальної інформаційної міри Кульбака. Оскільки інформаційний критерій є функціоналом від точнісних характеристик, то при репрезентативному обсязі навчальної вибірки необхідно користуватися їх оцінками:

$$D_{1,m}^{(k)}(d) = \frac{K_{1,m}^{(k)}}{n_{\min}}; \alpha_m^{(k)}(d) = \frac{K_{2,m}^{(k)}}{n_{\min}}; \beta_m^{(k)}(d) = \frac{K_{3,m}^{(k)}}{n_{\min}}; D_{2,m}^{(k)}(d) = \frac{K_{4,m}^{(k)}}{n_{\min}}, \quad (2.5)$$

де  $K_{1,m}^{(k)}$  – кількість подій, які означають належність реалізацій образу контейнеру  $K_{1,mk}^o$ , якщо дійсно  $\{x_1^{(j)}\} \in X_1^o$ ;  $K_{2,m}^{(k)}$  – кількість подій, які

означають неналежність реалізацій контейнеру  $K_{1,m}^o$ , якщо дійсно  $\{x_1^{(j)}\} \in X_1^o$ ;  $K_{3,m}^{(k)}$  – кількість подій, які означають належність реалізацій контейнеру  $K_{1,m}^o$ , якщо вони насправді належать класу  $X_2^o$ ;  $K_{4,m}^{(k)}$  – кількість подій, які означають неналежність реалізацій контейнеру  $K_{1,m}^o$ , якщо вони насправді належать класу  $X_2^o$ ;  $n_{\min}$  – мінімальний обсяг репрезентативної навчальної вибірки.

Робоча модифікація критерію Кульбака після відповідної підстановки оцінок (2.5) у вираз (2.4) набирає вигляду

$$E = \frac{1}{n} \log_2 \left\{ \frac{2n + 10^{-r} - [K_2^{(k)} + K_3^{(k)}]}{[K_2^{(k)} + K_3^{(k)}] + 10^{-r}} \right\} [n - (K_2^{(k)} + K_3^{(k)})], \quad (2.6)$$

де  $r$  – число цифр у мантисі значення критерію  $E_m^{(k)}$ .

Розглянемо схему обчислення коефіцієнтів  $K_2^{(k)}$  і  $K_3^{(k)}$  у формулі (2.6).

$$\begin{aligned} & (\forall X_1^o \in \mathfrak{R}^{[N]}) (\forall X_2^o \in \mathfrak{R}^{[N]}) [\text{if } x_1^{(j)} \in X_1 \text{ then} \\ & \quad K_1(j) := K_1(j-1) + 1 \text{ else } K_2(j-1) + 1]; \\ & (\forall X_1^o \in \mathfrak{R}^{[N]}) (\forall X_2^o \in \mathfrak{R}^{[N]}) [\text{if } x_2^{(j)} \in X_1 \text{ then} \\ & \quad K_3(j) := K_3(j-1) + 1 \text{ else } K_4(j) := K_4(j-1) + 1]. \end{aligned} \quad (2.7)$$

Тому інформаційні стандарти (2.4) та (2.6) можуть базуватися як на точних характеристиках рішення, так і на віддалених критеріях, тобто вони можуть розглядатися як узагальнення відомих статистичних даних та віддалені критерії для оптимізації параметрів СПР.

## **3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ**

### **3.1 Формування вхідної навчальної матриці**

Вхідна матриця навчання - це матриця типу "атрибути об'єкта", реалізована у рядках та стовпцях матриці. У створеній системі знань ми маємо 4 класи розпізнання.

Ця матриця представлена буквою  $Y$ .

Наступні кроки стосуються формування навчальної матриці:

– ми вибираємо функціональний стандарт системи підтримки прийняття рішень. Іншими словами, ідеально вивчити або контролювати стан об'єкта та з'ясувати всі можливі стани об'єкта управління, які можуть негативно вплинути на функціональний стандарт.

– набір ознак ідентифікації визначається відповідно до критерію функції. пошук кількості класів розпізнавання, які відповідатимуть функціональному стану об'єкта.

– пошук кількості реалізацій, що описують функціональний стан об'єкта.

– сформуванню навчальної матриці.

В результаті цього пункту буде з'ясовано, які функції можуть ідентифікувати об'єкти в різних станах, і скільки функціональних об'єктів можна виділити за критеріями функції, щоб сформуванню типовий набір векторів реалізації у конкретному стані.

### **3.2 Алгоритм інформаційно-екстремального машинного навчання**

Алгоритм інформаційного екстремального машинного навчання - це ітераційний процес, який знаходить критерій оптимізації ентропії та знаходить

своє максимальне значення на кожному кроці, оптимізуючі геометричних параметрів контейнера категорії розпізнавання, результат полягає в тому, що з геометричних правил слід побудувати надійне вирішальне правило, яке буде використано на етапі перевірки для визначення віднесення до одного з розпізнаних класів у визначеному алфавіті.

Розглянемо алгоритм базового інформаційно-екстремального алгоритму машинного навчання більш детально[15] Вхідною інформацією є:

- дійсний в загальному випадку масив реалізацій образу  $\{y_m^{(j)} | m = \overline{1, M}; j = \overline{1, n}\}$ ;
- система полів контрольних допусків  $\{\delta_{k,i}\}$  на ознаки розпізнавання;
- рівні селекції  $\{\rho_m\}$  координат еталонних векторів-реалізацій, які за замовчуванням дорівнюють 0,5 для всіх класів розпізнавання.

Розглянемо етапи реалізації алгоритму:

1. Формування бінарної навчальної матриці  $\|x_{m,i}^{(j)}\|$ , яке здійснюється за правилом

$$x_1^{(j)} = \begin{cases} 1, & \text{if } y_{1,i} - \delta \leq y_{1,i}^{(j)} \leq y_{1,i} + \delta \\ 0, & \text{if else.} \end{cases}$$

Тобто ознака розпізнавання дорівнює одиниці якщо вона попадає в межі контрольних допусків, і дорівнює нулю в протилежному випадку.

2. Формування масиву еталонних двійкових векторів-реалізацій  $\{x_{m,i} | m = \overline{1, M}, i = \overline{1, N}\}$ , елементи яких визначаються за правилом

$$x_{m,i} = \begin{cases} 1, & \text{if } \frac{1}{n} \sum_{j=1}^n x_{m,i}^{(j)} > \rho_m \\ 0, & \text{if else,} \end{cases}$$

де  $\rho_m$  – рівень селекції координат вектора  $x_m \in X_m^o$ ,

$x_{m,i}$  –  $i$ -я координата еталонного вектору класа  $X_m^o$ ,

$n$  – кількість реалізацій ( $n > 2$ ), які належать контейнеру класа  $X_m^o$

1. Розбиття множини еталонних векторів на пари найближчих "сусідів":

$$\mathfrak{R}_m^{[2]} = \langle x_m, x_l \rangle,$$

де  $x_l$  – еталонний вектор сусіднього класу  $X_l^o$ , за такою схемою алгоритму:

а) структурується множина еталонних векторів, починаючи з вектора  $x_1$  базового класу  $X_1^o$ , який характеризує найбільшу функціональну ефективність інформаційної системи;

б) будується матриця кодових відстаней між еталонними векторами розмірності  $M \times M$ ;

в) для кожного рядка матриці кодових відстаней знаходиться мінімальний елемент, який належить стовпчику вектора, найближчого до вектора, що визначає рядок. За наявності більше двох однакових мінімальних елементів обирається з них будь-який, оскільки вони є рівноправними;

г) формується структурована множина елементів попарного розбиття  $\{\mathfrak{R}_m^{[2]} | m = \overline{1, M}\}$ , яка задає хід навчання.

4. Оптимізація кодової відстані  $d_m$  відбувається за рекурентною процедурою

$$d_m(k) = [d_m(k-1) + h | d_m(k) \in G_m^d],$$

де  $k$  – змінна числа збільшень радіуса контейнера  $K_m^o$ ;

$h$  – крок збільшення радіуса;

$G_m^d$  – область допустимих значень радіуса  $d_m$ . При цьому береться  $E_m(0) = 0$ .

5. Процедура закінчується при знаходженні максимуму ентропійного критерію оптимізації в робочій області його визначення:

$$E_m^* = \max_{\{d\}} E_m,$$

де  $\{d\} = \{d_1, \dots, d_k, \dots, d_{m_{l_{max}}}\}$  – множина радіусів концентрованих гіперсфер, центр яких визначається вершиною еталонного вектора  $x_m \in X_m^o$ . При цьому множина  $\{d\}$  є так само множиною кроків навчання інтелектуальної системи.

Вихідні параметри базового алгоритму такі як – оптимальні кодові відстані  $\{d_m^*\}$ , оптимальні еталонні вектори-реалізації  $\{x_m^*\}$  для заданого алфавіту  $\{X_m^o\}$  є обов'язковими вхідними даними для функціонування інтелектуальної системи в режимі екзамену, тобто в режимі моніторингу.

На ефективність навчання СППР значно впливає використання контуру оптимізації СКД. Алгоритм оптимізації контрольних допусків за МФСВ полягає у наближенні глобального максимуму інформаційного критерію оптимізації до найбільшого його значення в області значень функції. Введемо такі позначення:

$x = \langle x_1, \dots, x_i, \dots, x_N \rangle$  – структурований вектор ознак розпізнавання;

$\delta_K = \langle \delta_{K,1}, \dots, \delta_{K,i}, \dots, \delta_{K,N} \rangle$  – структурований вектор стартових параметрів контрольних допусків на ознаки розпізнавання;

$l$  – кількість прогонів ітераційної процедури послідовної оптимізації контрольних допусків;

$E_{1\max}^{(l)}(x, \delta)$  – максимальне значення КФЕ в робочій області його визначення при  $l$ -му прогоні ітераційної процедури;

$E_1^*$  – найбільший глобальний максимум функції КФЕ в області її значень;

$\delta_{k,i}^{(l)}$  – значення параметра поля контрольних допусків для  $i$ -ої ознаки, яке отримано при  $l$ -му прогоні ітераційної процедури та дорівнює половині інтервалу  $[y_{1,i}^{(j)} \pm \delta_{K,i}^{(l)}]$ ;

$\delta_{K,i}^{(l)}$  – екстремальне значення параметра поля контрольних допусків для  $i$ -ї ознаки;

$\delta_{K,i}^*$  – оптимальне значення поля контрольних допусків для  $i$ -ї ознаки:

$$\delta_{K,i}^* = \arg \max_{G_i} E_1^* .$$

Існує декілька можливих стратегій зміни поля допусків  $\delta_{K,i}$ , серед яких відзначимо дві основні: симетрична стратегія  $S_1$ , яка є виправданою, наприклад, за умови співпадання номінального значення  $A_0$  з центром розсіювання реалізацій образу і асиметрична стратегія  $S_2$ , яка має місце при неспівпаданні значення  $A_0$  з емпіричним центром розсіювання реалізацій. Оптимізація СКД за МФСВ принципово може здійснюватися за трьома алгоритмами:

- послідовний алгоритм LEARNING - 1, при якому контрольні допуски оптимізуються послідовно для кожної ознаки при фіксованих значеннях інших ознак;
- паралельний алгоритм LEARNING - 2, при якому контрольні допуски оптимізуються для всіх ознак одночасно;
- алгоритм оптимізації за зведеним полем допусків.

Загалом, коли функція є багатофакторною групою, рекомендується використовувати послідовний алгоритм. Рекомендується використовувати паралельний алгоритм, якщо характеристики відображають вплив фактору. Алгоритм оптимізації СКД за зведеним полем допусків може розглядатися як узагальнення послідовного та паралельного алгоритмів за наявності різних



шкал виміру ознак. В нашому випадку було вирішено використовувати паралельний алгоритм оптимізації СКД.

Паралельний алгоритм LEARNING-2 оптимізує параметри контейнерів класів розпізнавання за умови ітераційної процедури визначення для базового класу  $X_1^o$  оптимальних контрольних допусків на всі ознаки одночасно. Вхідні дані такі самі як і для алгоритму LEARNING-1, але за область визначення параметра  $\delta$  приймається інтервал  $[1; \delta_H / 2]$ , де  $\delta_H$  – ширина нормованого поля допусків. Розглянемо кроки реалізації цього алгоритму:

- 1 Обнулюється лічильник кроків зміни параметра  $\delta$ :  $l:=0$ .
- 2 Запускається лічильник:  $i:=i+1$  та обчислюються нижні та верхні контрольні допуски для всіх ознак:  $\{A_{HK,i}[l] := y_{m,i} - \delta[l]\}$  і  $\{A_{BK,i}[l] := y_{m,i} + \delta[l]\}$ ,  $i = \overline{1, N}$ , відповідно.
- 3 Реалізується базовий алгоритм навчання.
- 4 Якщо  $E_1^*[l] \geq E_1^*[l-1]$ , то виконується пункт д), інакше пункт е).
- 5 Якщо  $\delta \leq \delta_H / 2$ , то виконується пункт б), інакше пункт е).
- 6  $\{A_{HK,i}^* := A_{HK,i}[l-1]\}; \{A_{BK,i}^* := A_{BK,i}[l-1]\}, i = \overline{1, N}; E_1^* := E_1^*[l-1]$  і “ЗУПИН”.

Алгоритм послідовної оптимізації LEARNING-1, при якому контрольні допуски оптимізуються послідовно для кожної ознаки розпізнавання при фіксованих (стартових) значеннях допусків для інших ознак, дозволяє підвищити функціональну ефективність навчання системи розпізнавання у порівнянні з алгоритмом паралельної оптимізації.

Алгоритм оптимізації контрольних допусків у рамках ІЕІТ полягає у наближенні глобального максимуму інформаційного критерію оптимізації до граничного його значення в області значень функції критерію. У праці [5] наведено для гіперсферичного класифікатора структуру ітераційної

процедури оптимізації параметра поля контрольних допусків  $\sigma_{K,i}$  для  $i$ -ї ознаки розпізнавання у вигляді:

$$\{\delta_{K,i}^*\} = \langle \arg\{\max_{\sigma_{\delta_i}} \{\max_{\sigma_{\varepsilon}} \left[ \bigotimes_{l=1}^L \max_{\sigma_{d_i}} E_1^{(l)} \right]\} \} \rangle, \quad i = \overline{1, N}, \quad (3.3)$$

де  $G_{\delta_i}$ ,  $G_{\varepsilon}$ ,  $G_{d_i}$  – області допустимих значень поля контрольних допусків, критерію оптимізації і кодової відстані  $d_1$  радіуса контейнера класу  $X_1^o$  відповідно;  $\bigotimes$  – символ операції повторення;  $L$  – кількість прогонів ітераційної процедури послідовної оптимізації контрольних допусків;  $E_1^{(l)}$  – значення інформаційного критерію оптимізації контрольного поля допусків для  $i$ -ї ознаки розпізнавання на  $l$ -му прогоні процедури оптимізації.

1 обнулення лічильника прогонів процедури оптимізації параметра поля контрольних допусків:  $l:=0$ .

2 для стартової системи допусків обчислюється за базовим алгоритмом навчання LEARNING [5] значення функції  $E_{\max,1}^{(l)}$ .

3 формування лічильника прогонів:  $l:=l+1$ .

4 обнулення лічильника ознак розпізнавання:  $i:=0$ .

5 формування лічильника ознак розпізнавання:  $i:=i+1$ .

6 визначення екстремального значення параметра поля допусків  $\hat{\delta}_{K,i}^{(l)}$  за процедурою (3.1), в якій внутрішній цикл оптимізації реалізує базовий алгоритм LEARNING [5].

$$7 \quad \hat{\delta}_{K,i}^{(l)} := \max_{\delta_{K,i} \in \delta_{H,i}} \delta_{K,i}^{(l)}.$$

8 якщо  $i \leq N$ , то виконується пункт д), інакше - пункт и).

9 якщо  $\left| E_{\max,1}^{(l-1)} - E_{\max,1}^{(l)} \right| \leq \varepsilon$ , де  $\varepsilon$  – будь-яке мале позитивне число,

то виконується пункт к), інакше – пункт в).

10  $\{\delta_{K,i}^*\} := \{\hat{\delta}_{K,i}^{(l)}\}$  і “Зупин”.

### 3.3 Короткий опис програмного забезпечення

Програмна реалізація була створена на мові програмування C#. C# це об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET Синтаксис C# близький до C++ і Java Мова має сувору статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції класу, атрибути, події, атрибути, винятки та коментарі у форматі XML. На основі своїх попередників C ++, Delphi, Module та Smalltalk, -C # базується на практиці використання, виключаючи деякі моделі, які виявилися проблематичними в розробці програмного забезпечення: Так, C # не підтримує багаторазове успадкування класів C ++ відносно) або тип виводу (відрізняється від Haskell).

Основні методи інформаційно-аналітичної системи наведено в таблиці 3.1

Таблиця 3.1 – Основні методи програми

| Назва методу                              | Короткий опис   |
|---|---|
| double[]searchAverage                     | Метод пошуку середнього значення  |
| double[]searchLimit                       | Метод пошуку верхнього нижнього допуску   |
| double[,]convertBinMatrix                 | Метод конвертації в бінарну матрицю   |
| double[]SearchEtalVecBin                  | Метод пошуку еталонних векторів   |
| double SearchMaxCount                     | Метод знаходження максимуму   |
| double SearchMinCount                     | Метод знаходження мінімуму  |
| double SearchCountXOR                     | Метод розрахунків оптимальних значень радіусів контейнерів класів розпізнавання |
| double[] SearchCountXORforEachLinesMatrix | Метод розрахунків кодових відстаней між центрами класів та реалізаціями.        |
| double[] SearchKFE                        | Метод пошуку значення КФЕ   |

### 3.4 Результати фізичного моделювання

Для перевірки ефективності системи було створено навчальну матрицю з чотирьох класів. Вхідна навчальна матриця для кожного класу розпізнавання складалася із 40 структурованих векторів, а кожний вектор – із 120 діагностичних ознак.

Критерієм оптимізації параметрів функціонування використовувався інформаційна міра Кульбака.

На рис. 3.1-3.4 голубим коліром позначено робочу область зміна КФЕ.

Аналіз рисунку вказує на те, що оптимальним в інформаційному розумінні значення параметра  $\delta = 46$ . Оскільки йому відповідає максимальне усередненого значення КФЕ  $\bar{E}^* = 1,57$ .

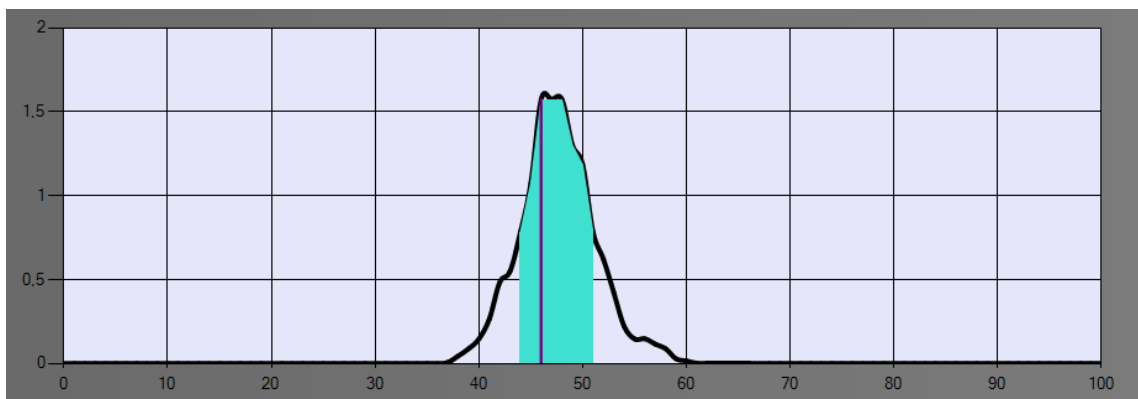


Рисунок 3.1 – Залежність міри Кульбака від радіусу контейнера класу  $X_1$

Аналіз рис. 3.2 вказує на те, що оптимальним в інформаційному розумінні значення параметра  $\delta = 29$  Оскільки йому відповідає максимальне усередненого значення КФЕ  $\bar{E}^* = 0.14$ .

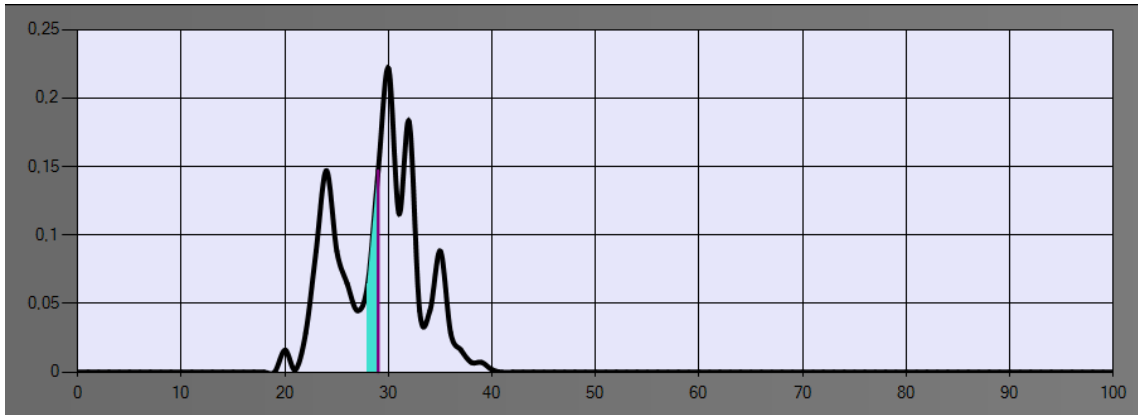


Рисунок 3.2 – Залежність міри Кульбака від радіусу контейнера класу  $X_2$

Аналіз рис. 3.3 вказує на те, що оптимальним в інформаційному розумінні значення параметра  $\delta = 29$  Оскільки йому відповідає максимальне усередненого значення КФЕ  $\bar{E}^* = 0.36$ .

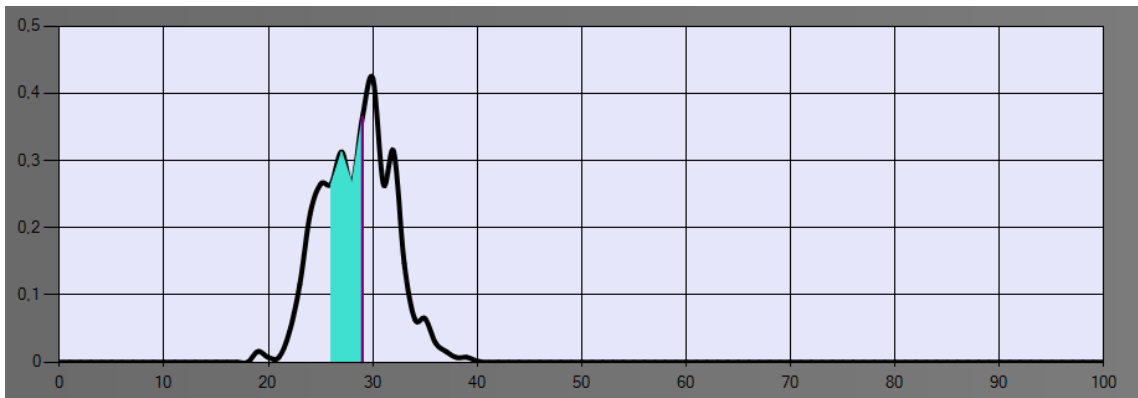


Рисунок 3.3 – Залежність міри Кульбака від радіусу контейнера класу  $X_3$

Аналіз рис 3.4 вказує на те, що оптимальним в інформаційному розумінні значення параметра  $\delta = 7$  Оскільки йому відповідає максимальне усередненого значення КФЕ  $\bar{E}^* = 7,65$ .

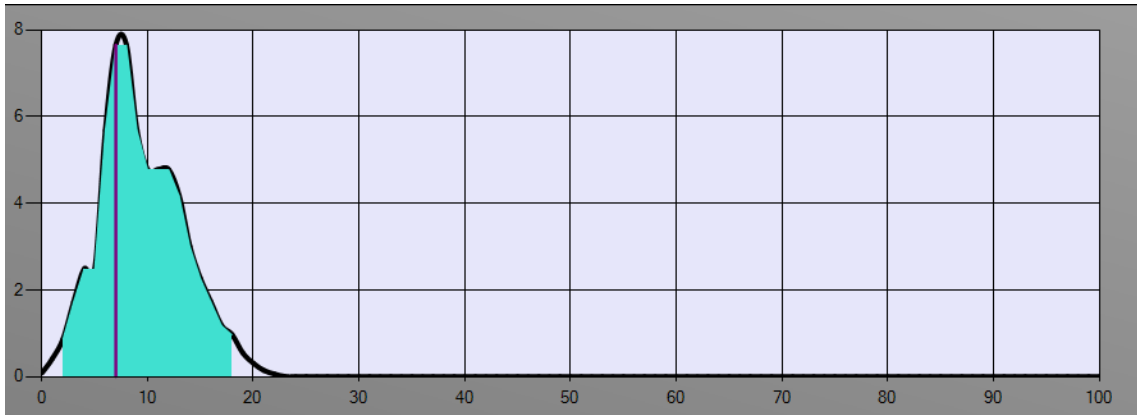


Рисунок 3.4 – Залежність міри Кульбака від радіусу контейнера класу  $X_4$

Для підвищення ефективності навчання було застосовано паралельний алгоритм оптимізації системи контрольних допусків на ознаки розпізнавання.

На рис. 3.5 подано динаміки усереднено КФЕ в процесі оптимізації СКД.

Аналіз рис. 3.5 показує про досягнення максимуму узагальненого КФЕ  $\bar{E} = 1,6$  при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta = 27$ .

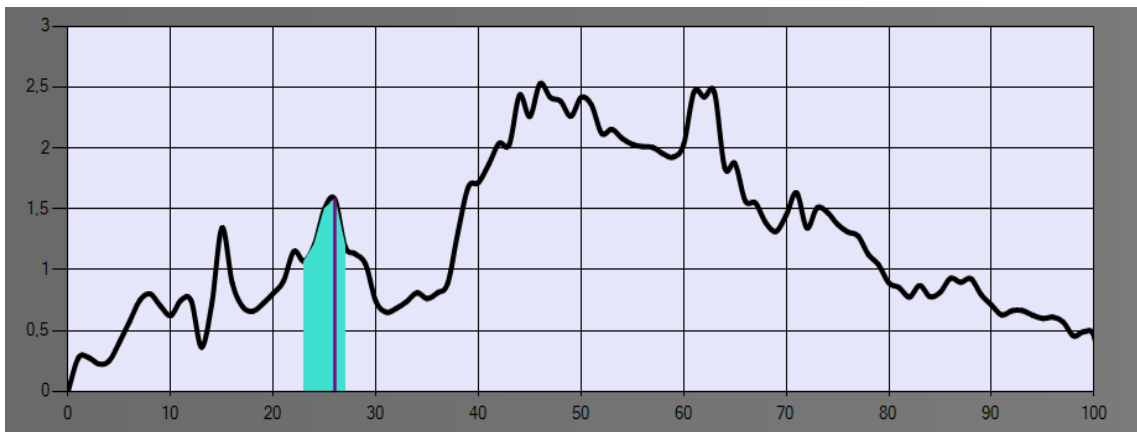


Рисунок 3.5 – Графік зміни усередненого критерію функціональної ефективності при паралельній оптимізації системи контрольних допусків на ознаки розпізнавання

На рис. 3.6-3.9 показано процес оптимізації радіусів контейнерів класів розпізнавання при оптимальній системі контрольних допусків.

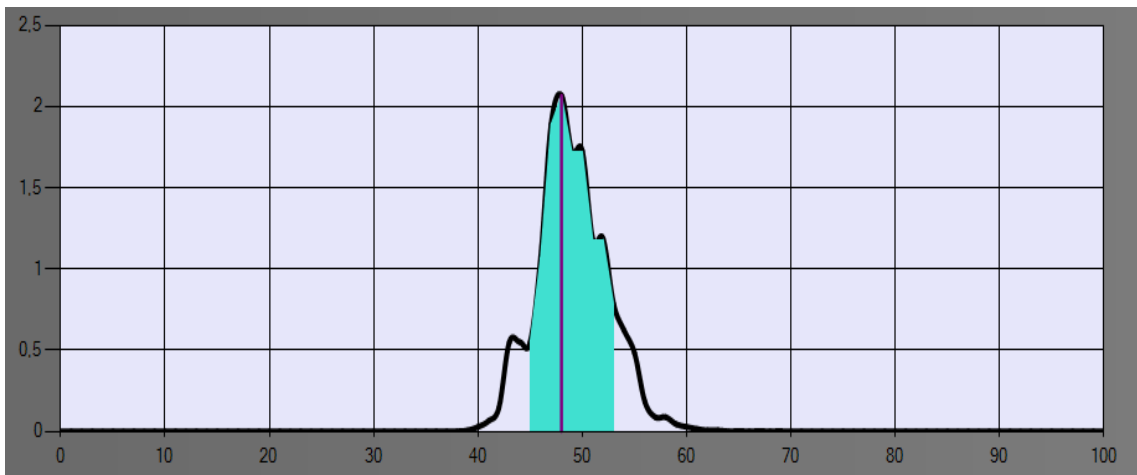


Рисунок 3.6 – Графік оптимізації радіусів контейнера класу розпізнавання  $X_1$

Рис. 3.6 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 2,11$  при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 48$ .

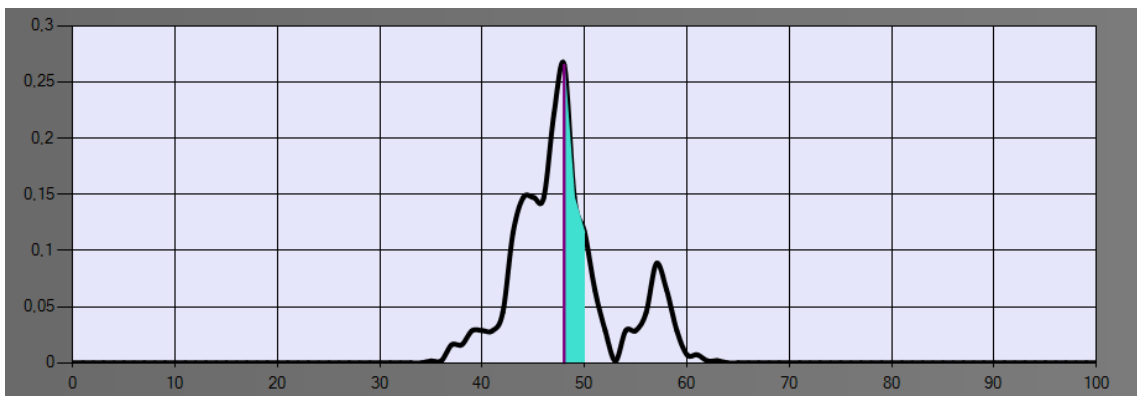


Рисунок 3.7 – Графік оптимізації радіусів контейнера класу розпізнавання  $X_2$

Аналіз рис. 3.7 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 0,271$  при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 48$ .

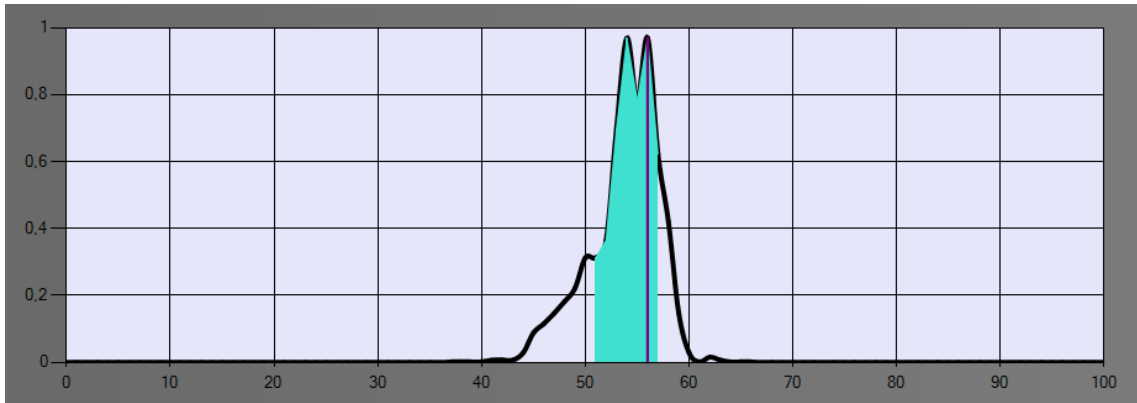


Рисунок 3.8 – Графік оптимізації радіусів контейнера класу розпізнавання  $X_3$

Аналіз рис. 3.8 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 0,98$  при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 56$ .

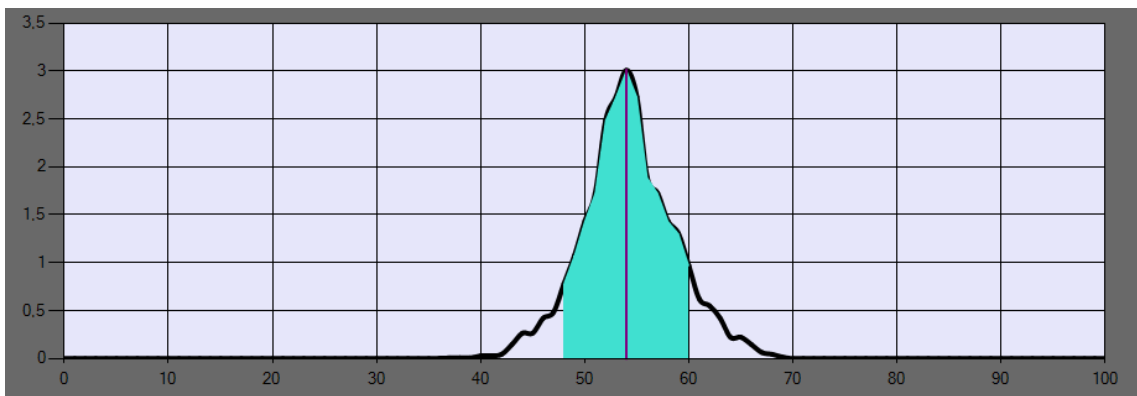


Рисунок 3.9 – Графік оптимізації радіусів контейнера класу розпізнавання  $X_4$

Аналіз рис. 3.9 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 3$  при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 54$ .

Для підвищення ефективності навчання було застосовано послідовний алгоритм оптимізації системи контрольних допусків на ознаки розпізнавання. На рис. 3.10 подано динаміки усереднено КФЕ в процесі оптимізації СКД.



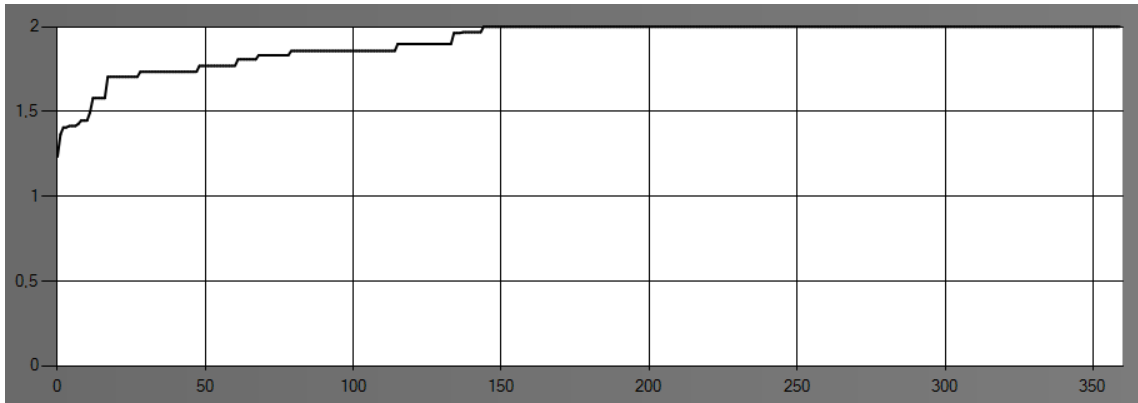


Рисунок 3.10 – Графік зміни усередненого КФЕ при послідовній оптимізації СКД

При значенні параметра СКД, якому відповідає максимальне значення усередненого КФЕ  $E=4,77$  в робочій області визначення даного параметра, послідовна оптимізація завершилася і проводилася відповідна оптимізація параметрів контейнерів класів розпізнавання результати яких можна побачити на рис. 3.11-3.14.

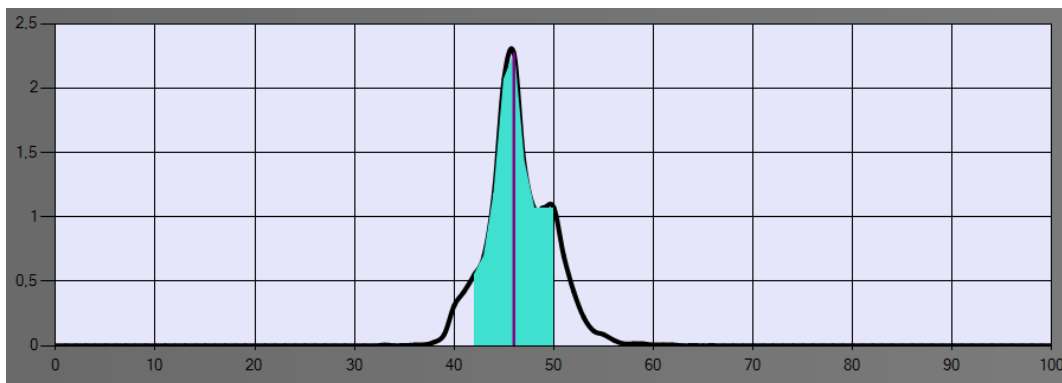


Рисунок 3.11 – Оптимізації геометричних параметрів контейнерів класів розпізнавання після послідовної оптимізації СКД  $X_1$  класу розпізнавання

Аналіз рис. 3.11 показує про досягнення максимуму узагальненого КФЕ  $E_1= 2,26$ , при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 46$ .

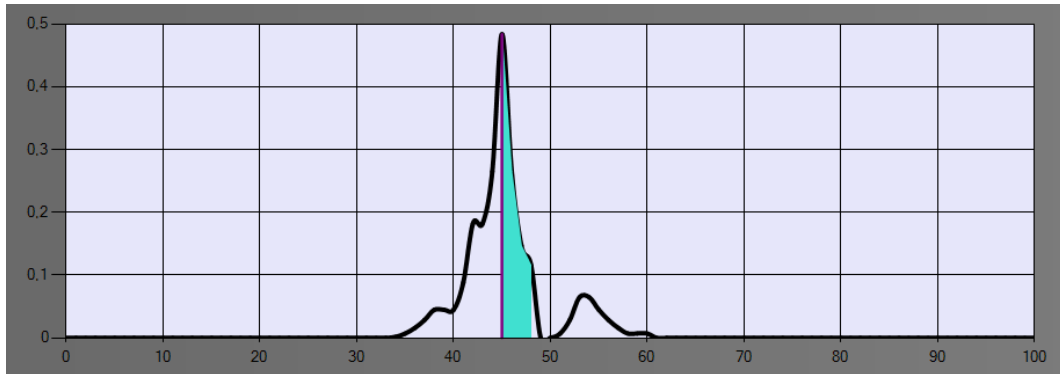


Рисунок 3.12 – Оптимізації геометричних параметрів контейнерів класів розпізнавання після послідовної оптимізації СКД  $X_2$  класу розпізнавання

Аналіз рис. 3.12 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 0,48$ , при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 45$ .

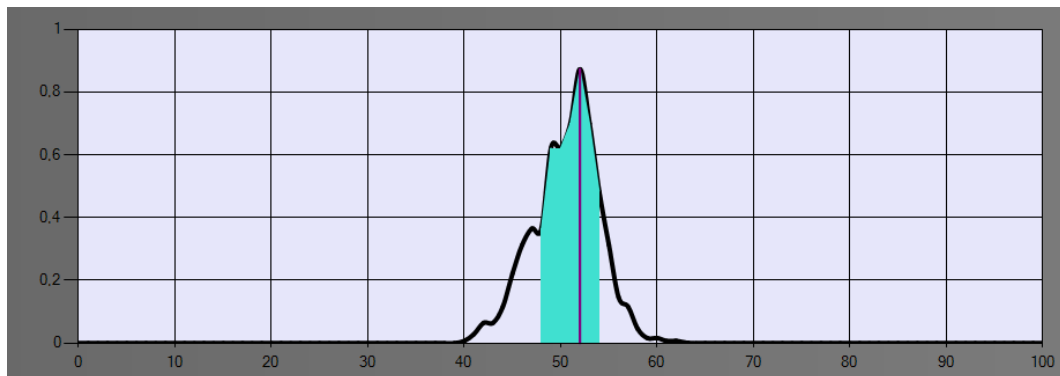


Рисунок 3.13 – Оптимізації геометричних параметрів контейнерів класів розпізнавання після послідовної оптимізації СКД  $X_3$  класу розпізнавання

Аналіз рис. 3.13 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 0,87$ , при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 52$ .

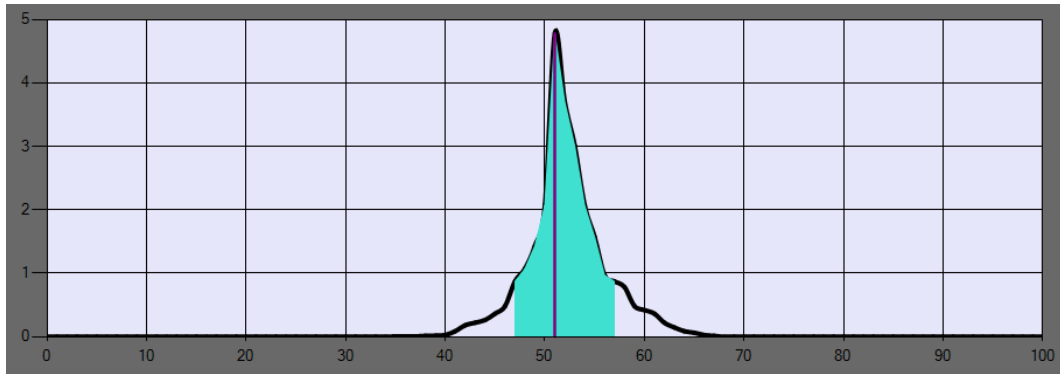


Рисунок 3.14 – Оптимізації геометричних параметрів контейнерів класів розпізнавання після послідовної оптимізації СКД  $X_4$  класу розпізнавання

Аналіз рис. 3.14 показує про досягнення максимуму узагальненого КФЕ  $E_1 = 4,77$ , при оптимальному значенню параметра поля контрольних допусків дорівнює  $\delta_1 = 51$ .

Отримані результати, які були отримані під час фізичного моделювання, свідчать про те, що дана система підтверджує високу достовірність розпізнавання та надійність запропонованого алгоритму.

## ВИСНОВКИ

У даній роботі було створено аналітично-інформаційну систему оцінки якості навчального контенту з використанням базових алгоритмів навчання інформаційно-екстремальної інтелектуальної технології . При цьому були виконані такі дії:

- Було сформовано навчальну матрицю аналітично-інформаційної системи;
- Обрано математичну модель для функціонування аналітично-інформаційної системи в режимі навчання;
- Було оцінено функціональну ефективність інтелектуальної аналітично-інформаційної системи.
- Розроблено та програмно реалізувано алгоритми викладання аналізу та знань частини інформаційної систем шляхом оптимізації основних функціональних параметрів;

## СПИСОК ЛІТЕРАТУРИ

1. Blackboard: Teaching and Learning [Електронний ресурс]. – Режим доступу : <http://www.blackboard.com/International/EMEA.aspx?lang=en-us>.
2. Exploring WebCT Campus Edition 6.0. Designer and Instructor Tutorials [Електронний ресурс]. – Режим доступу : <http://tutorials.webct.com/exploring/>.
3. The official web-site of Moodle LMS [Електронний ресурс]. – Режим доступу : <http://moodle.org/>
4. Астахова В. И. Непрерывное образование как приоритетное направление образовательной политики украинского государства / В. И. Астахова // Методологія, теорія та практика соціологічного аналізу сучасного суспільства : зб. наук. пр. / М-во освіти і науки України; Харк. нац. ун-т ім. В. Н. Каразіна. – Х., 2002. – С. 497–499. 89
5. Астахова В. И. Образование непрерывное / В. И. Астахова // Глоссарий современного образования / Нар. укр. акад. ; под общ. ред. Е. Ю. Усик ; [сост.: Астахова В. И. и др.]. – Харьков : Изд-во НУА, 2007. – С. 262–263
6. Астахова В. І. З досвіду становлення інноваційного модуля безперервної освіти (перші підсумки експерименту) / В. І. Астахова // Вчені зап. Харк. гуманіт. ун-ту «Нар. укр. акад.». – Т. 9. – 2003. – С. 7–12.
7. Довбиш А. С. Основи проектування інтелектуальних систем: навчальний посібник.
8. Довбыш А. С. Основи проектування інтелектуальних систем [Електронний ресурс]. URL: <https://essuir.sumdu.edu.ua/handle/123456789/1407> .
9. Довбыш А. С.; Шелехов І. В. ОСНОВИ ТЕОРІЇ РОЗПІЗНАВАННЯ ОБРАЗІВ Навчальний посібник.

10. Інформаційно-аналітична система адаптації навчального контенту випускової кафедри до вимог ринку праці / А. В. Васильєв, А. С. Довбиш, Є. С. Кулік, З. В. Козлов // Бионика интеллекта: информация, язык, интеллект. 15
11. Камынина К. Как устроены нейронные сети [Електронний ресурс] / Ксения Камынина // 2016 – Режим доступу до ресурсу: <http://downtown.ru/voronezh/technology/8800>.
12. Логістична регресія. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Логістична\\_регресія](https://uk.wikipedia.org/wiki/Логістична_регресія) 10
13. Метод k-найближчих сусідів. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Метод\\_kнайближчих\\_сусідів](https://uk.wikipedia.org/wiki/Метод_kнайближчих_сусідів).
14. Метод опорних векторів. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Метод\\_опорних\\_векторів](https://uk.wikipedia.org/wiki/Метод_опорних_векторів).
15. Штучний інтелект, машинне навчання та нейронні мережі: в чому різниця і для чого їх використовують – [Електронний ресурс]. – Режим доступу: <https://evergreens.com.ua/ua/articles/machine-learning-overview.html>

## ДОДАТОК

```
public double[] searchLimit(double[] array, string which_one, double delta)
{
    double[] retArray = new double[array.Length];
    if (which_one == "Up")
    {
        for (int i = 0; i < array.Length; i++)
        {
            retArray[i] = array[i] + delta;
        }
    }
    if (which_one == "Down")
    {
        for (int i = 0; i < array.Length; i++)
        {
            retArray[i] = array[i] - delta;
        }
    }
    return retArray;
}

public double[] searchAverage(double[,] Matrix)
{
    double z = 0;
    double[] meanMas = new double[Matrix.GetLength(0)];

    for (int i = 0; i < Matrix.GetLength(0); i++)
    {
        for (int j = 0; j < Matrix.GetLength(1); j++)
        {
            z = z + Matrix[i, j];
        }
        meanMas[i] = z / 40d;
        z = 0;
    }
    return meanMas;
}
```

```
public double[,] convertBinMatrix(double[,] matrix, double[] dopuskDownn, double[] dopuskUpn)
{
    double[,] Bin = new double[matrix.GetLength(0), matrix.GetLength(1)];

    for (int j = 0; j < matrix.GetLength(1); j++)
    {
        for (int i = 0; i < matrix.GetLength(0); i++)
        {
            if (matrix[i, j] < dopuskUpn[i] && matrix[i, j] > dopuskDownn[i])
            {
                Bin[i, j] = 1d;
            }
            else
            {
                Bin[i, j] = 0d;
            }
        }
    }
    return Bin;
}

public double[] SearchEtalVecBin(double[] array)
{
    double[] newMeanBin = new double[array.Length];
    for (int i = 0; i < array.Length; i++)
    {
        if (array[i] > 0.5d)
        {
            newMeanBin[i] = 1;
        }
        else
        {
            newMeanBin[i] = 0;
        }
    }
    return newMeanBin;
}
```



```

public double SearchMaxCount(double[] arr)
{
    double max = arr[0];
    for (int i = 1; i < arr.Length; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }
    return max;
}

public double[] SearchCountXORforEachLinesMatrix(double[] mainEtal, double[,] bin)
{
    double[] array = new double[bin.GetLength(1)];
    int sum;
    for (int j = 0; j < bin.GetLength(1); j++)
    {
        sum = 0;
        for (int i = 0; i < bin.GetLength(0); i++)
        {
            if (mainEtal[i] != bin[i, j])
            {
                sum++;
            }
        }
        array[j] = sum;
    }
    return array;
}

public double[] SearchCountXORforEachLinesMatrix(double[] mainEtal, double[,] bin)
{
    double[] array = new double[bin.GetLength(1)];
    int sum;
    for (int j = 0; j < bin.GetLength(1); j++)
    {
        sum = 0;
        for (int i = 0; i < bin.GetLength(0); i++)
        {
            if (mainEtal[i] != bin[i, j])
            {
                sum++;
            }
        }
        array[j] = sum;
    }
    return array;
}

```