

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**«Інтелектуальна система ідентифікації аудіо-
реклами»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Шелехов І.В.

Студента групи ІІ.м – 81н

Кондратенко А.С.

СУМИ 2020

Сумський державний університет

(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук

Спеціальність Інформатика

Затверджую:

зав.кафедрою _____

“ _____ ” _____ 20__ р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Кондратенку Артему Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інтелектуальна система ідентифікації аудіо реклами

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) Аналітичний огляд рішення проблеми та постановка задачі дослідження.

2) Інформаційно-екстремальна інтелектуальна технологія. 3) Розробка

Інформаційного та програмного забезпечення інтелектуальної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) 1 картинка

приклад розрахунку звукових відбитків, 5 схематичних зображень для опису роботи

алгоритмів, 1 скриншот реалізації навчальної матриці за зображенням осцилограмми 1

скриншот із зображенням використаної програми у роботі, 1 скриншот зовнішнього

вигляду розробленої програми, 4 схематичні зображення із результатами виконання

програмного коду.

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1.3 Постановка задачі інформаційного синтезу інтелектуальної системи	Шелехов І.В.		

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>		
2.	<i>Написання першого розділу: Аналітичний огляд рішення проблеми та постановка задачі дослідження</i>		
3.	<i>Написання другого розділу: Інформаційно-екстремальна інтелектуальна технологія</i>		
4.	<i>Вибір та налаштування середовища розробки. Підбір інструментів та бібліотек для розробки програмного забезпечення.</i>		
5.	<i>Завантаження вхідних даних, проведення дослідження та обробка отриманих даних.</i>		
6.	<i>Написання третього розділу: Розробка Інформаційного та програмного забезпечення інтелектуальної системи</i>		
7.	<i>Написання розділів: Вступ, Висновки, Додатки</i>		

Студент – дипломник

_____ (підпис)

Керівник проекту

_____ (підпис)

РЕФЕРАТ

Записка: 72 стор., 13 рис., 2 табл., 1 додаток, 20 джерел.

Об'єкт дослідження — записаний аудіо потік, сформований реальним звучанням.

Мета роботи - розробка інтелектуальної системи аналізу аудіо потоку інтернет-радіо з метою ідентифікації та видалення, або заміни рекламних повідомлень.

Методи дослідження — метод функціонально-статистичних випробувань, теорії інформації та інформаційно-екстремальної інтелектуальної технології аналізу даних.

Результати — у рамках інформаційно-екстремальної інтелектуальної технології розроблено інформаційне та програмне забезпечення інтелектуальної системи ідентифікації рекламних аудіо роликів. Сформовано математичну модель оптимізації у функціональному параметрі інтелектуальних систем з використанням інформаційних критеріїв функціональної ефективності. Розроблено та програмно реалізовано алгоритм функціонування інтелектуальної системи за допомогою інструментального програмного середовища Visual Studio 2019.

СИСТЕМА ІДЕНТИФІКАЦІЇ АУДІО РЕКЛАМИ,
ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, НАВЧАЛЬНА МАТРИЦЯ,
МЕТОД ФУНКЦІОНАЛЬНО-СТАТИСТИЧНИХ ВИПРОБУВАНЬ,
ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА
ТЕХНОЛОГІЯ, АУДІО РЕКЛАМА, ІНФОРМАЦІЙНИЙ СИНТЕЗ
ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.

ЗМІСТ

ВСТУП.....	6
1. АНАЛІТИЧНИЙ ОГЛЯД РІШЕННЯ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	8
1.1. АНАЛІЗ ПРОБЛЕМИ РЕКЛАМНОГО ВПЛИВУ НА ЛЮДИНУ	8
1.2. СУЧАСНІ МЕТОДИ ІДЕНТИФІКАЦІЇ АУДІО РЕКЛАМИ	10
1.3. ПОСТАНОВКА ЗАДАЧІ ІНФОРМАЦІЙНОГО СИНТЕЗУ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	16
2. ІНФОРМАЦІЙНО–ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ.....	18
2.1. ОСНОВНІ ПОНЯТТЯ ТА ВИЗНАЧЕННЯ.....	18
2.2. МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ НАВЧАННЯ.....	21
2.3. ОЦІНКА ФУНКЦІОНАЛЬНОЇ ЕФЕКТИВНОСТІ	25
2.4. БАЗОВИЙ АЛГОРИТМ НАВЧАННЯ	27
3. РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	32
3.1. ФОРМУВАННЯ НАВЧАЛЬНОЇ МАТРИЦІ.....	32
3.2. МАТЕМАТИЧНА МОДЕЛЬ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ НАВЧАННЯ.....	33
3.3. АЛГОРИТМ ПАРАЛЕЛЬНОЇ ОПТИМІЗАЦІЇ СИСТЕМИ КОНТРОЛЬНИХ ДОПУСКІВ НА ОЗНАКИ РОЗПІЗНАВАННЯ	35
3.4. КОРОТКИЙ ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ.....	35
3.5. АНАЛІЗ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ	39
ВИСНОВКИ.....	47
СПИСОК ЛІТЕРАТУРИ.....	48
ДОДАТОК А.....	51

ВСТУП

Станом на кінець 2019 року 4.1 мільярда людей у світі є користувачами Інтернету, що складає більше половини населення нашої планети і збільшується.[1] З цієї статистики маємо, що всі користувачі, коли заходять в всесвітню мережу переглядає рекламу, і можливо не один раз. Наприклад, переглянути рекламу перед відео із Youtube чи інших джерел або фільму, банери та схожі. Більшість такої реклами може бути заблоковано, що дає аудиторії Інтернету вільно користуватись ресурсами без набридливої оголошень.

Усе б було добре, кількість реклами зменшено до мінімуму, з часом реклама стає такою, яка обходить блокувальники, що дає стимул удосконалювати їх. Винахідливість людей, які створюють рекламу вишла на новий рівень, і тому вони «запахнули» рекламу в аудіо і поширюють її в радіо, музичних композиціях, між переходами різних пісень і так далі.

Блокувальники реклами давно стали головним болем для онлайн-видавців. Проте, популярність таких розширень до браузера зростає шаленими темпами.

Написано безліч програмного забезпечення із використанням інформаційно-інтелектуальні технології. Теоретичне дослідження алгоритмів ведеться уже досить тривалий час, як і мають практичне використання.

Конкурентоспроможністю цього способу обробки інформації є очевидною. Тому в порівнянні з існуючими на способами особливої уваги потребує проблема визначення завдань, яких було б кращим та більш продуктивним застосування нейромережевих алгоритмів. Розпізнавання образів - є одним із завдань, успішно вирішувальним нейромережами. Однією з теорій розпізнавання образів є також розпізнавання аудіо сигналів. Проблема розпізнавання давно приваблювала дослідників, і хоч досягнуті успіхи в даній складовій штучного інтелекту, але вона залишається відкритою.

Спробувати застосувати нейромережеві алгоритми на практиці, описати і вирішити виниклі проблеми, а також розробити теорію самообучаючихся систем і реалізувати її на конкретному прикладі - ось які завдання ставилися в рамках цієї роботи.

1. АНАЛІТИЧНИЙ ОГЛЯД РІШЕННЯ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1. Аналіз проблеми рекламного впливу на людину

Людина, будучи відкритою системою, не може не реагувати на дуже велику кількість інформації, тому важливо зрозуміти, до чого призведе такий потік інформації, чи не виявиться він згубним для людини.

Реклама впливає на формування моральних, етичних, суспільно-політичних і естетичних принципів людини.

Через швидкий розвиток реклами у сучасної людини з'являється постійна потреба в нових відчуттях, нових почуттях і переживаннях, переданих рекламою. Якщо раніше покупка нових меблів, одягу та посуду проводилася досить рідко, то зараз довга відсутність обновок призводить людей до поганого самопочуття. Рекламщики диктують людству, без чого людина не може жити, яку новинку йому варто спробувати, як ставитися до тієї чи іншої події в житті. [2]

Друга проблема реклами - це створення в мозку людини нав'язливої ідеї вічної молодості і краси. Сучасна реклама пропонує людині неможливе: змінити природні фізіологічні процеси організму, це робить рекламований товар своєрідною формою чарівної палички з казок, але ж кожному відомо, що чудес в реальному житті не буває.

Також подібна реклама може бути побудована на контрастах: спочатку потенційним покупцям показують всі непривабливі сторони життя, пов'язані зі сферою застосування товару, а потім чудове позбавлення від всіх проблем - барвисте життя з цим товаром. Однак при першому ж провалі такої рекламної акції споживачі перестають довіряти компанії, намагаються обійти всі фірми зі схожими послугами. [3]

Ще одна риса, яка характерна багатьом рекламним роликам - залучення до їх створення відомих артистів, зірок естради і кіно, популярних спортсменів і політиків. Таким чином, реклама закликає бути модним і схожим на людину

відому, яскраву, успішну і багату. Якщо ж це не всім вдається, то можна хоча б на психологічному рівні компенсувати цю нестачу щасливого і райдужного життя, купуючи товари, якими користується кумир мільйонів.

Ретельно продумана і спрямована на певного глядача реклама може показати, що товар необхідний споживачеві як повітря. Адже для того, щоб він це зрозумів, необхідно ціле мистецтво психоаналізу і знання споживчих потреб, що дається дуже не легко.[4]

- Реклама, багаторазово повторюючись, створює свої посилання;
- реклама роз'яснює споживачеві, що даний товар дуже для нього потрібен, впливаючи ситуаційно і емоційно;
- реклама просуває скоріше не товар, а спосіб життя, пов'язуючи свої посилання з побутом;
- реклама доносить свої посилання саме в той момент, коли свідомість людини діє;
- реклама, впливаючи на своїх споживачів, потихеньку міняє моральний клімат у суспільстві. Але для того, щоб занадто сильно не маніпулювати людьми, тому як вони з одного боку інертні, а з іншого занадто емоційні, був створений маркетинг;

Багато споживачів сміється над рекламою, не помічаючи що самі попадаються в її пастку, тому як реклама переконує не в перевазі свого товару, а в можливості з його допомогою вести «солодкий» спосіб життя. Виходить що реклама, таким чином, продає душевний стан.

Творці реклами дуже добре розуміють, що всередині кожного споживача сидить невпевнена і боязка істота, і користуються цим знанням у побудові внутрішнього і зовнішнього виду реклами.

Тому дана проблема дуже актуальна. Моя робота спрямована на ідентифікацію рекламних роликів з аудіо потоку, з можливістю заміни її на інформацію що не несе реклами.

1.2. Сучасні методи ідентифікації аудіо реклами

Розпізнавання аудіо реклами можна поділити залежно від критерію. Так, наприклад поділяють на такі типи[5]:

- гучність;
- блокування по годинам;
- метадані;
- маркування вручну;
- краудсорсинг рекламної бази, знаходження звуковим відбиткам;
- розпізнавання мови і аналіз лексичного поля;
- класифікація реклами, розмов та музики на машинном;
- розпізнавання мови і аналіз лексичного поля.

1.2.1. Гучність

Перше, що прийде на розум – перевірка гучності, адже саму аудіо-рекламу роблять гучнішою ніж звуки. Для реклами нерідко використовується акустичне стиснення. Це досить цікавий критерій, але його використання не завжди можливо. Найчастіше для того, щоб слухачі відволіклись від своїх думок та звернули свою увагу на рекламу саме тому і підвищують гучність, але не скрізь. Використання такої самої гучність або менше також можливо, само собою увага буде в рази менше, але це компенсується тим, що частіше запускають рекламу. Деяким особам головне сама наявність реклами.

1.2.2. Блокування по годинам

Ще один варіант, який працює тільки на онлайн радіо на якому реклама транслюється тільки по розкладу. Блокування оголошень таким способом є досить складним, адже на радіостанції можливо затримали випуск реклами до декількох хвилин.

1.2.3. Метадані

Один із варіантів, можливо використовувати метадані, за якими онлайн програвачі, як наприклад, VLC або AIMP відображають інформацію про то, що зараз грає. Але це також не завжди буде правдиво, оскільки дані можуть бути зламани, або коли буде озвучена реклама то передаватиметься інформація про пісню яка буде грати наступною.

1.2.4. Маркування вручну

Взагалі, виявлення реклами на онлайн радіо можна без алгоритму. Попросити слухачів виявляти рекламу натиснувши кнопку, коли розпочинається або закінчується реклама. Всі слухачі, навіть різних радіостанцій виграють від цього. Така стратегія працює на телеприставках TiVo Volt. Вона дає можливість видаляти рекламу на певних встановлених каналах у відповідний час. Результати такої стратегії прекрасні, але не дуже поширюється на тисячі радіостанцій.

Недолік такого полягає в тому, що складно запустити описану систему з нуля. На новоствореній радіостанції може бути мала кількість аудиторії для правильної роботи.

Ще одна трудність полягає в тому, що радіостанціям для того, щоб саботувати систему будуть відправляти підроблені сигнали. Для цього потрібно механізм модерації або використання системи консенсусу.

1.2.5. Краудсорсинг рекламної бази, знаходження звуковим відбиткам

Техніка пошуку яка базується на звукових відбитках схожа на алгоритм, що робить програма Shazam для розпізнавання пісень на своїх серверах. Даний алгоритм працює по принципу аудіо фінденпринтинга на основі лендмарка [6,7].

«Фінгерпрінтінг ідентифікує користувача не за спеціальними мітками, збереженим на його системі, а по унікальним особливостям його браузера, системи і пристроя»[8].

Фінгерпрінтинг добре підходить для ідентифікації аудіо реклами, тому що вони транслюються в однаковому вигляді. Так само він розпізнає і музичні композиції. Але це не буде працювати на мовлення людини, оскільки складно вимовляти одні й ті самі слова однаково. Це можливо тільки тоді, коли трансляція передачі повторюється, що нас зовсім не цікавить. Значить, що в базу відбитків необхідно вносити і рекламу, і музику (як тип, що не відноситься до реклами).

«Звукові відбитки - це перетворення деяких звукових характеристик в ряд чисел, які називаються відбитком»[9]. Якщо в потоці радіостанції багато збігів відбитків з базою то, скоріш за все, передається реклама. Для найбільш ефективного вирішення, тимчасового і частотного діапазону потрібна деякі налаштування. Різні приклади повинні досить непогано відрізнитись. Однак система повинна справно працювати працювати навіть якщо були внесені зміни алгоритмів стиснення звуку або на радіостанції відбулась зміна налаштування еквайзера. Зрештою, повинно бути обмежити на кількість відбитків для того, щоб не завантажувати обчислювальні ресурси.

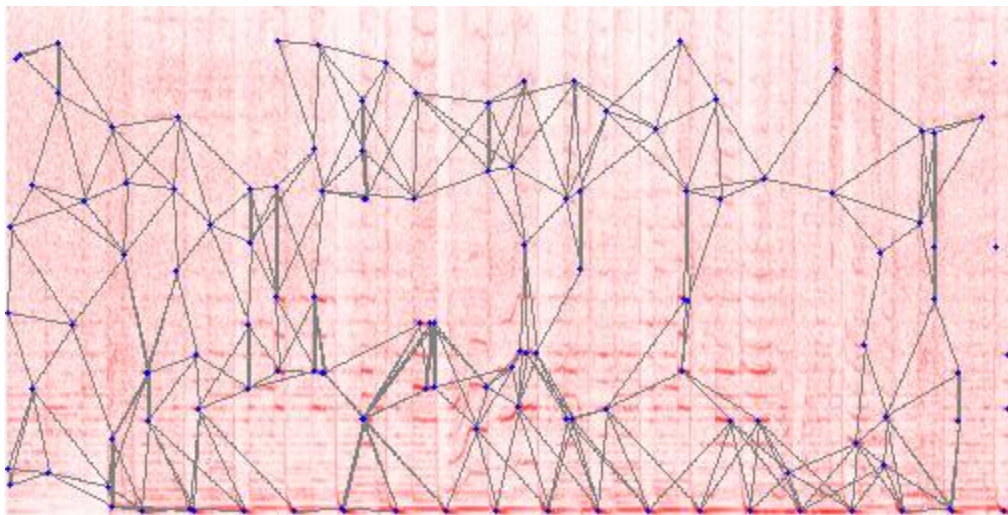


Рис.1.1 - Приклад розрахунку звукових відбитків[5].

На Рис.1.1 показано, що фон - це спектрограма. Вона показує як змінюється інтенсивність звуку по частоті (внизу - низькі частоти). На Рис.1.1 спектральні піки ідентифікуються (сині точки) і з'єднуються (сірі лінії).

Розташування, довжина та орієнтація кожної лінії перетворюється в унікальне число, яке і є відбиток.

Бінарна систематизація дає підсумок: є приклад рекламою або ні. У випадку Якщо досліджувати випадки, де сталися похибки, в такому випадку концепція практично постійно давала неправильно негативні підсумки, в такому випадку ігнорувала рекламу, також вельми вкрай рідко фіксувала непоганий медіа-контент так само як рекламу. Користувачі можуть інформувати про невиявлених оголошеннях один клацанням миші, що гарантує прекрасний користувацький інтерфейс. Відповідне звучання відразу додається у БД.

Досить складно зберігати та оновлювати базу під актуальному стані, так як рекламна частина змінюються, але оголошення передаються з досить маленькими варіаціями. Вони крім того часто оновлюються, у певних випадках раз на декілька діб. Певні радіостанції із невеликою кількістю слухачів вельми слабо розпізнавалися.

Оголошення в однаковій мірі передаються багато разів кожен день. Це можливо застосовувати з метою їх ідентифікації. У записах виконувався відбір максимально повторюваних послідовностей (MRS). Інший медіа-контент також повторюється, такі як пісні також заставки. Відсортував все без винятку черговості згідно із протяжності та також забрав зразки із довжиною приблизно 30 секунд, що характерно для рекламних роликів. Подібним способом вельми часто виходило виявити рекламу. Однак в деяких випадках бували приспіві пісень або в тому числі і записані моніторинги погоди.

Є метод відфільтрувати велику частину музичних повторів: провести аналіз плейлистів станцій, завантажити пісні й також інтегрувати їх у базу з відміткою «не реклама». За цією причини становиться більше претендентів у MRS виявлялися дійсно рекламними роликами. Однак ніяк не всі, з цієї причини підтримка користувачів залишилася необхідною.

Уже потреба в ручній роботі зменшено, однак навантаження на сервери стала проблемою. Озираючись назад, не кращим опинився вибір SQLite для цих ресурсоемних, критичних згідно часу операцій із БД.

Але, на щастя, близько декілька секунд існувало, для того щоб встановити, чи є звучання рекламою або ні. Це відбувається внаслідок того, що мережа інтернет-радіомовлення застосовують аудіобуфер, як правило 4-30 секунд, який ніяк не відображається на екрані пристрою кінцевого користувача. [10] Це може допомогти уникнути обриви передач у разі тимчасової втрати зв'язку.

Застосовувавши цю затримку з метою постпроцесингу, для того щоб зробити моніторинги методу найбільш стабільними і також контекстно-залежними. Перед самим відтворенням звуку в кінцевого користувача метод розглядає підсумки моніторингів, які всі без винятку ще перебувають у буфері, але крім того найбільш старі, які раніше відтворені. Він відрізає підозрілі точки даних з деякими збігами відбитків, показуючи гистерезис. Крім того передбачає середньозважений період, для того щоб згладити ймовірні перебої. [9]

1.2.6. Розпізнавання мови і аналіз лексичного поля

Реклама – це також і лексичне поле: придбання продуктів, отримання накопичувальних балів на картку та схожі приклад. [4] У випадку якщо розпізнати мову, в такому випадку можливо використовувати типові прилади з метою боротьби зі спамом. Застосування характеристик згідно замовчуванням: типові словники (перелік імовірний текстів та також відповідних фонем), мовна модель (ймовірності послідовностей текстів) також звукова модель (взаємозв'язок фонем формою звукових хвиль) .Налаштування словника і також мовну модель в незначні набори даних, що ділять звучання приладом діарізації.

З 2017 роки досягнуто значного зростання у розпізнаванні мови. Опубліковані новітні інструменти з відкритим вихідним кодом, такі як Mozilla Deep Speech [11].

1.2.7. Класифікація реклами, розмов та музики на машинном

Подальший варіант методу досліджує акустику: з низьких аж до високих звучань, а також їх зміна в часі. Новітні незнайомі рекламні оголошення детектуються практично таким чином добре, так само як і старі, у яких, відбувалося навчання, тільки лише відповідно до показників шумності, а також і настирливості. Це найбільш непростий спосіб розгляду гучності звуку.

Застосовували інструменти машинного навчання, безпосередньо бібліотеку Keras, підключену до Tensorflow. Отримано досить хороші результати зі з невеликим застосуванні CPU. Зараз дійсно розрізняти розмови і музику, з цієї причини систематизація стала більш чіткою: замість «реклама / не реклама» - «реклама / діалог / музика».[12]

Досліджуємо деталі. Звучання перетворюється на 2D-карту, в місці інтенсивності звуку показана так само як функція від частоти і часу (приблизно 4 секунди). Дана схема схожа в червону картку у області про відбитки. Головна відмінність в тому, що замість традиційного діапазону Фур'є застосовується Мел-кепстральні коефіцієнти, важливі у контексті визначення мови.[13]

Послідовні карти з різними позначками періоду потім аналізувалися так само як ілюстрації у рекуррентної нейронної мережі виду LSTM (long short-term memory). Будь-яка карта аналізувалася незалежно (RNN без збереження стану), проте карти перекривалися з іншими. Карти протяжністю 4 секунди, також будь-яку одну секунду виникала нова. Остаточним підсумком з метою будь-якої карти робився вектор softmax, наприклад, ad: 81%, talk: 7%, music 12%. Дані моніторинги потім оброблялися цим способом, що описаний в області про відбитки.[14]

Незважаючи на відмінну роботу, достовірність систематизації все без винятку-дійсно стала трохи менше надій користувачів. При навчанні достовірність моніторингу групи зібрала 93-95%, проте збереглися помилкові класифікації, які залишали користувачів трохи роздратованими.

1.3. Постановка задачі інформаційного синтезу інтелектуальної системи

Розглянемо загальну постановку задачі інформаційного синтезу інтелектуальної системи. Нехай відома навчальна матриця «об'єкт-властивість», яка характеризує m -й функціональний стан інтелектуальної системи X_m^o :

$$\|y_{m,i}^{(j)}\| = \begin{vmatrix} y_{m,1}^{(1)} & y_{m,2}^{(1)} & \dots & y_{m,1}^{(1)} & \dots & y_{m,N}^{(1)} \\ y_{m,1}^{(2)} & y_{m,2}^{(2)} & \dots & y_{m,1}^{(2)} & \dots & y_{m,N}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y_{m,1}^{(j)} & y_{m,2}^{(j)} & \dots & y_{m,1}^{(j)} & \dots & y_{m,N}^{(j)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y_{m,1}^{(n)} & y_{m,2}^{(n)} & \dots & y_{m,1}^{(n)} & \dots & y_{m,N}^{(n)} \end{vmatrix} \quad (1.1)$$

У матриці (1.1) рядок є реалізацією образу $\{y_{m,i}^{(j)} | i = \overline{1, N}\}$, де N – кількість ознак розпізнавання, а стовпець матриці – випадкова навчальна вибірка $\{y_{m,i}^{(j)} | j = \overline{1, n}\}$, де n – обсяг вибірки, Дано структурований вектор параметрів функціонування $g = \langle g_1, \dots, g_\xi, \dots, g_\Xi \rangle$, які прямо або непрямо впливають на функціональну ефективність інтелектуальної системи, з відповідними на них обмеженнями

$R_\xi(g_1, \dots, g_\Xi) \leq 0$. Необхідно визначити оптимальні значення параметрів функціонування $\{g_\xi^*\}$, які забезпечують максимум КФЕ:

$$E_m^* = \max_G E_m \quad (1.2)$$

де E_m – КФЕ процесу навчання інтелектуальної системи розпізнавати реалізації класу X_m^o ; G – область допустимих значень параметрів функціонування.

У роботі необхідно розробити інтелектуальну систему аналізу аудіо потоку інтернет-радіо з метою ідентифікації та видалення, або заміни рекламних повідомлень. При цьому особливими завданнями роботи є:

- 1) формування вхідного математичного опису системи;
- 2) розробка алгоритмів оптимізації функціональних параметрів системи розпізнавання сигналів;
- 3) програмна реалізація розробленого алгоритму;
- 4) перевірка працездатності системи.

2. ІНФОРМАЦІЙНО–ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ

2.1. Основні поняття та визначення

Метод функціонально-статистичних випробувань (МФСВ) – непараметричний інформаційно-екстремальний метод аналізу та синтезу інтелектуальної системи, який ґрунтується на прямій оцінці інформаційної здатності системи за умов нечіткої компактності реалізацій образу, та обмеження навчальної вибірки, яка є прийнятною для задач контролю і управління. Метод призначено для розв’язання практичних задач контролю та управління слабо формалізованими системами і процесами шляхом автоматичної класифікації їх функціональних станів за умови невизначеності.[15]

МФСВ окрім системних та специфічних принципів ґрунтується також на 2-х дистанційних принципах[16]:

- максимально-дистанційному, який вимагає максимальної міжцентрової відстані між класами;
- мінімально-дистанційному, вимагає мінімальної середньої відстані реалізацій від центру свого класу.

Класом розпізнавання (образом) X_m^0 називається відбиття властивостей m -го функціонального стану СР і відношень між елементами системи. Клас розпізнавання – топологічна категорія, яка задається в просторі ОР областю $X_m^0 \subset \Omega_B$.

Детерміновано-статистичний підхід до моделювання систем вимагає завдання систем нормованих (експлуатаційних) і контрольних допусків на ОР. Нехай X_m^0 – базовий клас, який характеризує максимальну функціональну ефективність, тобто є найбільш бажаним для розробника інформаційного забезпечення системи. Нормованим називається поле допусків $\{\delta_{H,i} \mid i = \overline{1, N}\}$, в якому значення i -ї ОР знаходиться з імовірністю $p_i = 1$ або $p_i = 0$, за умови, що

функціональний стан відноситься до класу X_1^0 . Контрольним називається поле допусків $\{\delta_{K,i} \mid i = \overline{1, N}\}$, в якому значення i -ї ОР знаходиться з імовірністю $0 < p_i < 1$ за умови, що функціональний стан відноситься до класу X_1^0 .

В МФСВ система контрольних допусків (СКД) вводиться з метою рандомізації процесу прийняття рішень, оскільки для повного дослідження об'єкту контролю та управління (ОКУ) необхідно використовувати як детерміновані, так і статистичні характеристики. Зрозуміло, що $|\delta_{K,i}| \leq |\delta_{H,i}|$ і базова (відносно класу X_1^0) СКД є сталою для всієї абетки класів розпізнавання.

Реалізацією образу $x_m^{(j)} \in X_m^0$ називається випадковий структурований бінарний вектор $x_m^{(j)} = \langle x_{m,1}^{(j)}, \dots, x_{m,i}^{(j)}, \dots, x_{m,N}^{(j)} \rangle$, $j = \overline{1, n_{\min}}$, де $x_{m,i}^{(j)}$ – i -та координата вектора, яка приймає одиничне значення, якщо значення i -ї ОР знаходиться в полі допусків $\delta_{K,i}$, і нульове значення, якщо не знаходиться; n_{\min} – мінімальна кількість випробувань, яка забезпечує репрезентативність реалізацій образу.

При обґрунтуванні гіпотези компактності (чіткої, або нечіткої) реалізацій образу за геометричний центр класу X_m^0 приймається вершина бінарного еталонного вектору x_m .

Еталонний вектор (ЕВ) x_m – це математичне сподівання реалізацій класу X_m^0 .

Він подається у вигляді детермінованого структурованого бінарного вектора $x_m = \langle x_{m,1}, \dots, x_{m,i}, \dots, x_{m,N} \rangle$, $m = \overline{1, M}$, де $x_{m,i}$ – i -та координата вектора, яка приймає одиничне значення, якщо значення i -ї ОР знаходиться в нормованому полі допусків $\delta_{H,i}$, і нульове значення, якщо не знаходиться.

Основною задачею етапу навчання за МФСВ є розбиття простору ОР за поданою навчальною матрицею на області класів розпізнавання деяким оптимальним в інформаційному сенсі способом, який забезпечує на етапі

екзамену прийняття рішень з достовірністю, наближеною до максимальної асимптотичної достовірності.[17]

Параметром функціонування називається характеристика інформаційного забезпечення, яка прямо або непрямо впливає на функціональну ефективність системи. Такими параметрами можуть бути параметри навчання, перетворення образу, впливу середовища та інші, які безпосередньо впливають на асимптотичну достовірність.

Як критерій оптимізації процесу навчання системи прийняттю рішень в рамках МФСВ застосовується статистичний інформаційний КФЕ, який є природною мірою різноманітності (або схожості) класів розпізнавання і одночасно функціоналом асимптотичних точнісних характеристик СПР. При цьому важливо, щоб параметри навчання були оптимальними в інформаційному розумінні, тобто забезпечували максимальну функціональну ефективність СПР, яка визначається достовірністю прийняття рішень на екзамені.[15]

Достовірність класифікатора залежить від геометричних параметрів роздільних гіперповерхонь класів розпізнавання.

У загальному випадку, коли класи розпізнавання перетинаються, розглянемо відносний коефіцієнт нечіткої компактності реалізації образу для класу

$$X_m^0 : l_{m,d} = d_m^* / d(x_m \oplus x^{(j)}) \quad (2.1)$$

Процес навчання полягає в мінімізації цього виразу.

В МФСВ, який ґрунтується на допущенні гіпотези компактності (чіткої або нечіткої) реалізації образу, як наближення "точної" роздільної гіперповерхні для класу X_m^0 розглядається гіперсфера, центром якої є еталонний вектор $x_m \in X_m^0$, а радіусом d_m – кодова відстань, яка у просторі Хеммінга визначається як

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

де $x_{m,i}$ – i -та координата вектора $x_m \in X_m^0$; $\lambda_{m,i}$ – i -та координата деякого вектора λ_m , вершина якого знаходиться на роздільній гіперповерхні класу X_m^0 ; \oplus – операція складання за модулем два.

Оптимальною кодовою відстанню (радіусом) між вектором x_m і контейнером K_m^0 називається екстремальне значення d_m^* , яке визначає максимум інформаційного КФЕ $E_m^* = \max_{\{d\}} E_m$, де $\{d\}$ – послідовність збільшень радіуса контейнера K_m^0 .

Побудова оптимальної в інформаційному сенсі РГП у вигляді гіперсфери за МФСВ зводиться до оптимізації радіуса роздільної гіперсфери d_m , яка відбувається за ітераційним алгоритмом

$$d_m(k) = d_m(k-1) + \begin{cases} h, & \text{if } E_m[d(k)] \geq E_m[d(k-1)]; \\ 0, & \text{if } \textit{else}, \end{cases} \quad (2.3)$$

де k – змінна числа збільшень радіуса РГП; h – крок збільшення.

Процедура закінчується при знаходженні екстремального значення критерію $E_m^* = \max_{\{d\}} E_m$, де $\{d\} = \{0, 1, \dots, d(x_m \oplus x_{m+1})\}$ – множина радіусів концентрованих гіперсфер, центр яких визначається вершиною еталонного вектора $x_m \in X_m^0$, x_{m+1} – еталонний вектор найближчого (до X_m^0) класу X_{m+1}^0 .

2.2. Математична модель процесу навчання

Розглянемо математичну модель інтелектуальної системи, яка реалізує класичну задачу розпізнавання образів за МФСВ. Математична модель повинна включати як обов'язкову складову частину вхідний математичний опис, який подамо на рівні системного аналізу у вигляді теоретико-множинної структури $\Delta_B = \langle G, T, \Omega, Z, \Psi, \Pi, \Phi \rangle$, де G – простір вхідних сигналів (факторів); T – множина моментів часу зняття інформації; Ω – простір ознак розпізнавання; Z – простір можливих станів; Y – множина сигналів після первинної обробки інформації;

$\Pi: G \times T \times \Omega \rightarrow Z$ – оператор переходів, що відбиває механізм зміни станів під дією внутрішніх і зовнішніх збурень; $\Phi: G \times T \times \Omega \times Z \rightarrow Y$ – оператор оброблення зображення (формування вибіркової множини Y на вході СПР), який є реакцією на внутрішні і зовнішні збурення. Як універсум випробувань W розглядається декартовий добуток: $W = G \times T \times \Omega \times Z$.

Для чіткого детермінованого розбиття $\mathfrak{R}^{[M]}$ потужності M , оператор побудови розбиття η задає відображення $\eta: Y \rightarrow \mathfrak{R}^{[M]}$. Априорно оператором $\nu: Y \rightarrow L^{[M]}$ будується покриття $L^{[M]} = \{X_m^o\}$, яке визначає абетку класів розпізнавання і на якому на відміну від розбиття відношення еквівалентності відсутнє. Оператор $\alpha: \mathfrak{R}^{[M]} \rightarrow L^{[M]}$ переводить апостеріорно розбиття $\mathfrak{R}^{[M]}$ в покриття $L^{[M]}$ і замикає комутативне кільце. Перевірка гіпотези про належність реалізації образу здійснюється оператором класифікації $\Psi: \mathfrak{R}^{[M]} \rightarrow I^{[M+1]}$, де $I^{[M+1]} = \{\gamma_1, \gamma_2, \dots, \gamma_M, \gamma_{M+1}\}$ – множина допустимих гіпотез (рішень). При цьому гіпотеза γ_{M+1} означає відмову від класифікації. У загальному випадку математичну модель будь-якої інтелектуальної системи, яка розв'язує задачу класифікаційного аналізу, подамо у вигляді діаграми відображень відповідних множин:

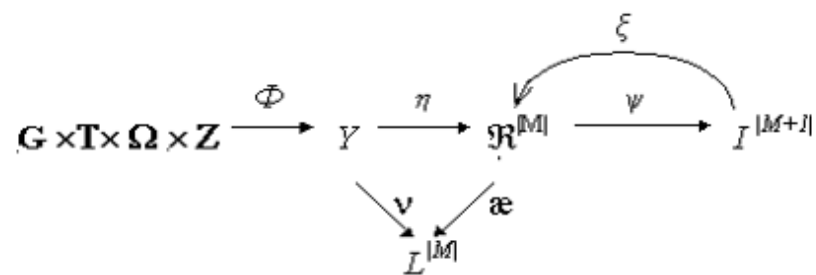


Рисунок 2.1 – Математична модель інтелектуальної системи

Оператор зворотного зв'язку ξ корегує геометричні параметри розбиття $\mathfrak{R}^{[M]}$ з метою поліпшення точнісних характеристик СПР. Саме спосіб реалізації оператора ξ породжує різні методи та підходи до розв'язання проблеми адаптивного навчання інтелектуальної системи.

При обґрунтуванні гіпотези нечіткої компактності має місце нечітке розбиття $\tilde{\mathfrak{R}}^{[M]} \subset \Omega$.

Тоді замість оператора η застосуємо оператор θ нечіткої факторизації простору ознак: $\theta: Y \rightarrow \tilde{\mathfrak{R}}^{[M]}$. Нехай оператор класифікації $\psi: \tilde{\mathfrak{R}}^{[M]} \rightarrow I[|]$ визначає перевірку основної статистичної гіпотези про належність реалізацій $\{x_m^{(j)} \mid j=\overline{1, n}\}$ класу X_m° , де l – кількість статистичних гіпотез. Принциповою відмінністю МФСВ від відомих методів автоматичної класифікації є ітераційна оптимізація процесу навчання за інформаційним КФЕ.

Для обчислення КФЕ оператор $\gamma: I[|] \rightarrow \mathfrak{Z}[q]$ шляхом оцінки прийнятих гіпотез γ_l формує множину ТХ $\mathfrak{Z}[q]$, де $q=l2$ – кількість ТХ, а оператор $\varphi: \mathfrak{Z}[q] \rightarrow E$ обчислює множину значень інформаційного КФЕ, який є функціоналом від ТХ. Ітераційний процес оптимізації геометричних параметрів розбиття $\tilde{\mathfrak{R}}^{[M]}$ реалізується оператором $r: E \rightarrow \tilde{\mathfrak{R}}^{[M]}$ шляхом пошуку максимуму КФЕ $E_m^* = \max_{\{d\}} E_m(d)$, де $\{d\}$ – множина кроків навчання розпізнаванню реалізацій класу X_m° . Структурна діаграма процесу навчання за МФСВ для випадку нечіткого розбиття має вигляд:

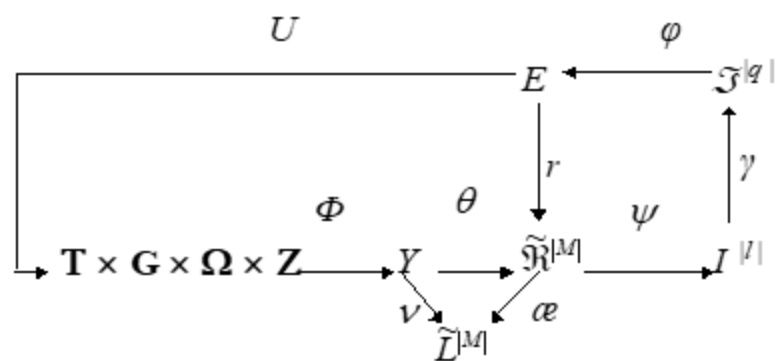


Рисунок 2.2 – Процес навчання за МФСВ у випадку нечіткого розбиття

Таким чином, контур операторів $\boxed{\psi \rightarrow \gamma \rightarrow \varphi \rightarrow r}$ безпосередньо оптимізує геометричні параметри розбиття $\tilde{\mathfrak{R}}^{[M]}$. Оператор U :

$E \rightarrow G \times T \times \Omega \times Z$ регламентує процес навчання. Оскільки в цьому контурі також застосовано процедуру обчислення інформаційного критерію, то він дозволяє розв'язувати задачу оптимізації параметрів плану навчання, які визначають, наприклад, обсяг і тривалість випробовувань, потужність словника ОР, черговість подання для навчання класів розпізнавання та інше.

2.3. Оцінка функціональної ефективності

Задача вибору та обчислення КФЕ є центральною проблемою оцінки функціональної ефективності інтелектуальної системи, що навчається, для вирішення якої застосовуються два основні підходи: один з них базується на економічних показниках, а інший – на інформаційних. Подолання існуючої суперечності між цими підходами полягає у реалізації принципу підпорядкованості економічних показників функціонування системи інформаційним. Інформаційний підхід природно базується на використанні для оцінки функціональної ефективності інформаційного критерію. У рамках ІЕІ-технології знайшли широке використання дві інформаційні міри: ентропійна міра Шеннона, яка є інтегральною мірою,

$$E_m^{(k)} = 1 + 0,5 \left(\frac{\alpha^{(k)}}{\alpha^{(k)} + D_2^{(k)}} \log_2 \frac{\alpha^{(k)}}{\alpha^{(k)} + D_2^{(k)}} + \frac{D_1^{(k)}}{\beta^{(k)} + D_1^{(k)}} \log_2 \frac{D_1^{(k)}}{\beta^{(k)} + D_1^{(k)}} + \frac{\beta^{(k)}}{\beta^{(k)} + D_1^{(k)}} \log_2 \frac{\beta^{(k)}}{\beta^{(k)} + D_1^{(k)}} + \frac{D_2^{(k)}}{\alpha^{(k)} + D_2^{(k)}} \log_2 \frac{D_2^{(k)}}{\alpha^{(k)} + D_2^{(k)}} \right) \quad (2.4)$$

та міра Кульбака, яка дозволяє оцінювати диференційну інформативність ознак:

$$J_m^{(k)} = 0,5 \log_2 \left(\frac{D_1^{(k)} + D_2^{(k)}}{\alpha^{(k)} + \beta^{(k)}} \right) * \left[(D_1^{(k)} + D_2^{(k)}) - (\alpha^{(k)} + \beta^{(k)}) \right] \quad (2.5)$$

де $D_1^{(k)}, D_2^{(k)}, \alpha^{(k)}, \beta^{(k)}$ – точнісні характеристики процесу навчання: перша та друга достовірності, помилки першого та другого роду відповідно, отримані на k -му кроці навчання.[15]

У загальному випадку побудований за (2.4) графік функції $E = f(D_1, D_2)$ є поверхнею в тривимірному просторі (рис. 2.3).

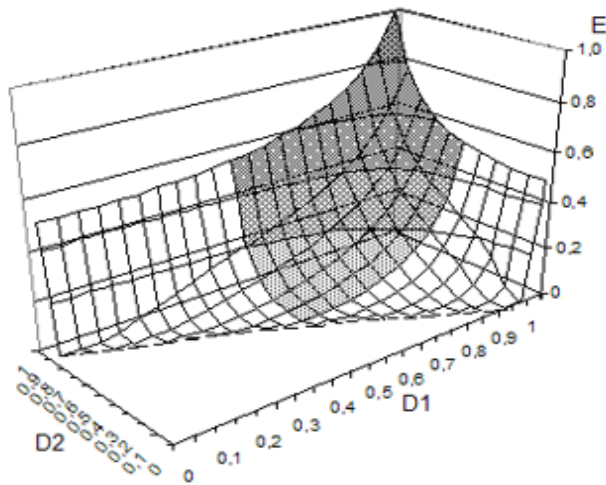


Рисунок 2.3 – Графік залежності критерію (2.4) від точнісних характеристик при двохальтернативному рішенні

Графік на рис. 2.3 є симетричним відносно площини $D_1 = D_2$. (друга частина на графіку не відображена).

Тривимірну поверхню модифікованого критерію $J = f(D_1, D_2)$, побудовану за формулою (2.5), показано на рис. 2.4.

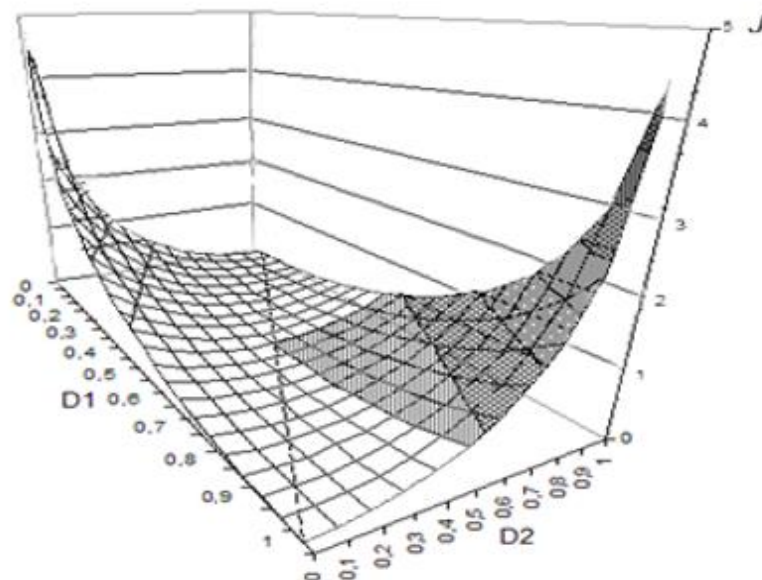


Рисунок 2.4. Графік залежності критерію (2.5) від точнісних характеристик: першої та другої достовірностей

Як видно з рис. 2.3 та рис. 2.4, функції (2.4) та (2.5) є взаємно-неоднозначні. На практиці цей недолік усувається шляхом введення робочої (допустимої) області визначення функцій, в якій значеннями точнісних характеристик – першої та другої достовірностей, повинні бути більшими відповідних помилок першого та другого роду, тобто $D1 \geq 0,5$ та $D2 \geq 0,5$. [18] Робочі області на рис. 2.3 і рис. 2.4 показано на відповідних графіках сірим кольором. Аналіз цих графіків показує, що при збільшенні у робочій області як першої, так і другої достовірностей кількість інформації теж збільшується, що знаходиться у відповідності з другим принципом адитивності інформації.

2.4. Базовий алгоритм навчання

Вхідною інформацією для навчання за базовим алгоритмом є дійсний, в загальному випадку, масив реалізацій образу $\{y_m^{(j)} \mid m = \overline{1, M}; j = \overline{1, n}\}$; система полів контрольних допусків $\{\delta_{k,i}\}$ і рівні селекції $\{\rho_m\}$, які за умовчанням дорівнюють 0,5 для всіх класів розпізнавання.

Розглянемо етапи реалізації алгоритму:

1. Формування бінарної навчальної матриці $\|x_{m,i}^{(j)}\|$, елементи якої дорівнюють

$$x_{m,i}^{(j)} = \begin{cases} 1, & \text{if } y_{m,i}^{(j)} \in \delta_{k,i}; \\ 0, & \text{if } y_{m,i}^{(j)} \notin \delta_{k,i}. \end{cases} \quad (2.6)$$

2. Формування масиву еталонних двійкових векторів $\{x_{m,i} \mid m = \overline{1, M}, i = \overline{1, N}\}$, елементи якого визначаються за правилом:

$$x_{m,i} = \begin{cases} 1, & \text{if } \frac{1}{n} \sum_{j=1}^n x_{m,i}^{(j)} > \rho_m; \\ 0, & \text{if } \text{else}, \end{cases} \quad (2.7)$$

де ρ_m – рівень селекції координат вектору $x_m \in X_m^o$.

1. Розбиття множини еталонних векторів на пари найближчих "сусідів": $\mathcal{R}_m^{(2)}$

$= \langle x_m, x_l \rangle$, де x_l – еталонний вектор сусіднього класу X_l^o , за таким алгоритмом:

- а) структурується множина еталонних векторів, починаючи з вектора x_l базового класу X_l^o , який характеризує найбільшу функціональну ефективність інтелектуальної системи;
- б) будується матриця кодових відстаней між еталонними векторами розмірності $M \times M$;
- в) для кожної строки матриці кодових відстаней знаходиться мінімальний елемент, який належить стовпчику вектора – найближчого до вектора, що визначає строку. При наявності декількох однакових мінімальних елементів вибирається з них будь-який, оскільки вони є рівноправними;
- г) формується структурована множина елементів попарного розбиття $\{\mathcal{R}_m^{(2)} \mid m = \overline{1, M}\}$, яка задає план навчання.

Оптимізація кодової відстані d_m відбувається за рекурентною процедурою. При цьому приймається $E_m(0) = 0$.

2. Процедура закінчується при знаходженні максимуму КФЕ в робочій області його визначення: $E_m^* = \max_{\{d\}} E_m$, де $\{d\} = \{0, 1, \dots, d < d(x_m \oplus x_l)\}$ – множина радіусів

концентрованих гіперсфер, центр яких визначається вершиною $x_m \in X_m^o$.

Таким чином, базовий алгоритм навчання :

$$d_m^* = \arg \max_{\{d\}} E_m^* \quad (2.8)$$

На рис. 2.5 наведено структурну схему базового алгоритму навчання.

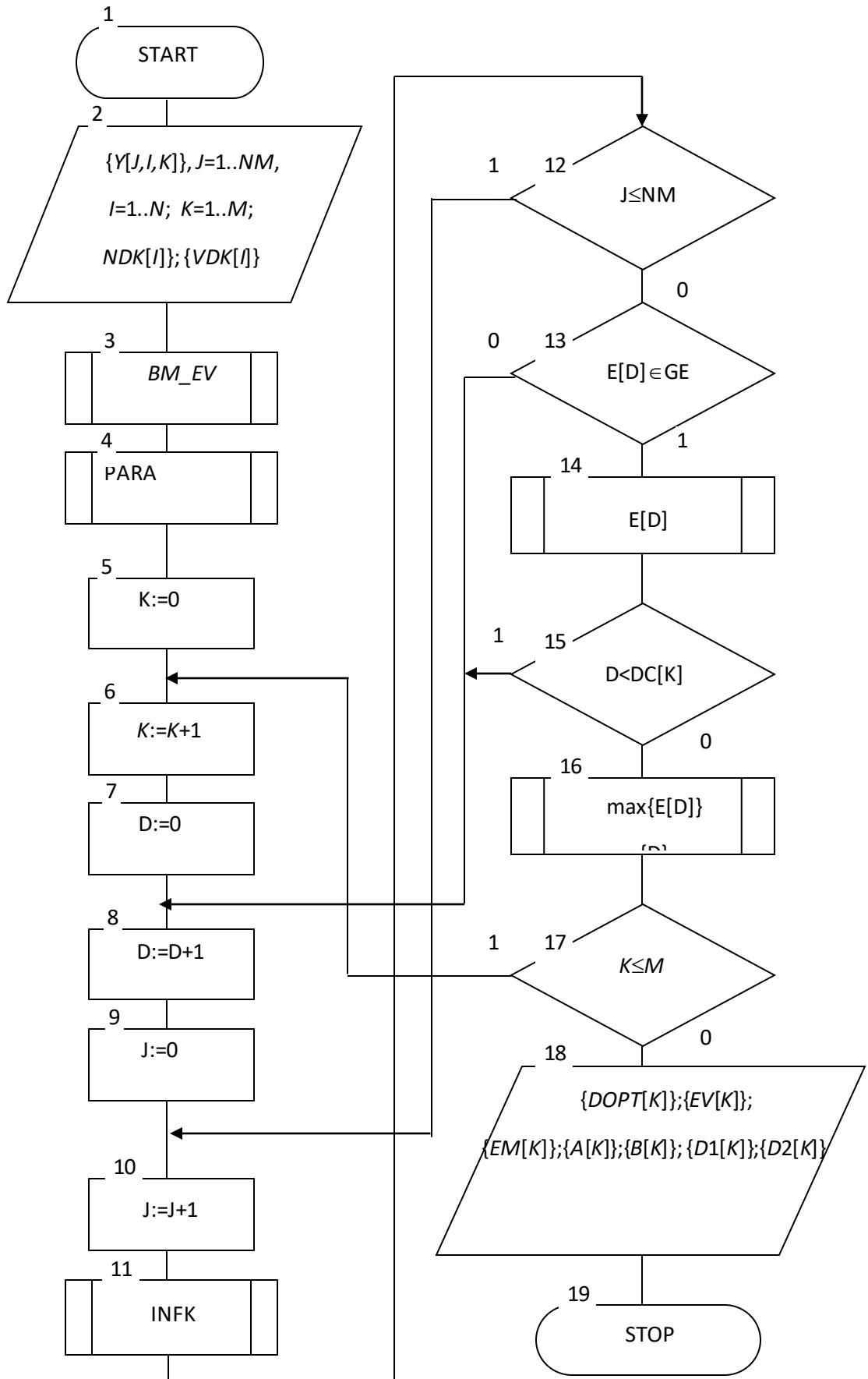


Рисунок 2.5– Структурна схема базового алгоритму навчання [19]

Тут показано такі вхідні дані: $\{Y[J,I,K]\}$ – масив навчальних вибірок, $J=1..NM$ – змінна кількості випробувань, де NM – мінімальний обсяг репрезентативної навчальної вибірки, $I=1..N$ – змінна кількості ознак розпізнавання, $K=1..M$ – змінна кількості класів розпізнавання; $\{NDK[I]\}$, $\{VDK[I]\}$ – масиви нижніх і верхніх контрольних допусків на ознаки відповідно. Результатом реалізації алгоритму є: $\{DOPT[K]\}$ – цілий масив оптимальних значень радіусів контейнерів класів розпізнавання у кодовій відстані Хеммінга; $\{EV[K]\}$ – масив еталонних двійкових векторів класів розпізнавання; $\{EM[K]\}$ – дійсний масив максимальних значень інформаційного КФЕ процесу навчання; $\{D1[K]\}$, $\{A[K]\}$, $\{B[K]\}$, $\{D2[K]\}$ – дійсні масиви оцінок екстремальних значень точнісних характеристик процесу навчання для відповідних класів розпізнавання: перша вірогідність, помилки першого та другого роду і друга вірогідність відповідно. Змінна D є робочою змінною кроків навчання, на яких послідовно збільшується значення радіуса контейнера. У структурній схемі алгоритму (рис. 1) блок 3 формує масив навчальних двійкових вибірок $\{X[J,I,K]\}$ шляхом порівняння значень елементів масиву $\{Y[J,I,K]\}$ з відповідними контрольними допусками за правилом (1) і формує масив еталонних двійкових векторів $\{EV[K]\}$ шляхом статистичного усереднення стовпців масиву $\{X[J,I,K]\}$ за правилом (2) при відповідному рівні селекції, який за умовчанням дорівнює $\rho_m = 0,5$. Блок 4 здійснює розбиття множини еталонних векторів на пари “найближчих сусідів”. Блок 11 обчислює на кожному кроці навчання значення інформаційного КФЕ і оцінки точнісних характеристик процесу навчання. При невиконанні умови блоку порівняння 12 блок 13 оцінює належність поточного значення критерію $E[D]$ робочій області G_E визначення його функції і при позитивному рішенні блоку 13 це значення запам’ятовується блоком 14. При негативному рішенні блоку порівняння 15, в якому величина $DC[K]$ дорівнює кодовій відстані між парою сусідніх еталонних векторів, блок 16 здійснює у робочій області G_E пошук глобального максимуму КФЕ – $EM[K]$ і

визначає для нього екстремальне значення радіуса гіперсфери – $DOPT[K]$. Аналогічно будуються оптимальні контейнери для інших класів. Якщо параметри навчання $\{DOPT[K]\}$ і $\{EV[K]\}$ є вхідними даними для екзамену, то значення КФЕ та екстремальних оцінок точнісних характеристик використовуються для аналізу ефективності процесу навчання. Таким чином, основною процедурою базового алгоритму навчання за МФСВ є обчислення на кожному кроці навчання статистичного інформаційного КФЕ і організація пошуку його глобального максимуму в робочій області визначення функції критерію.[20]

3. РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1. Формування навчальної матриці

Оптимізацію параметрів навчання інтелектуальної системи будемо проводити загального випадку, коли алфавіт класів складається з трьох і більше класів. Кожна реалізація в навчальній матриці представляє собою геометричний опис піків на осцилограммі у кожний момент часу (рис. 3.1). Таким чином, в навчальній матриці фіксуються послідовність частот звуку.

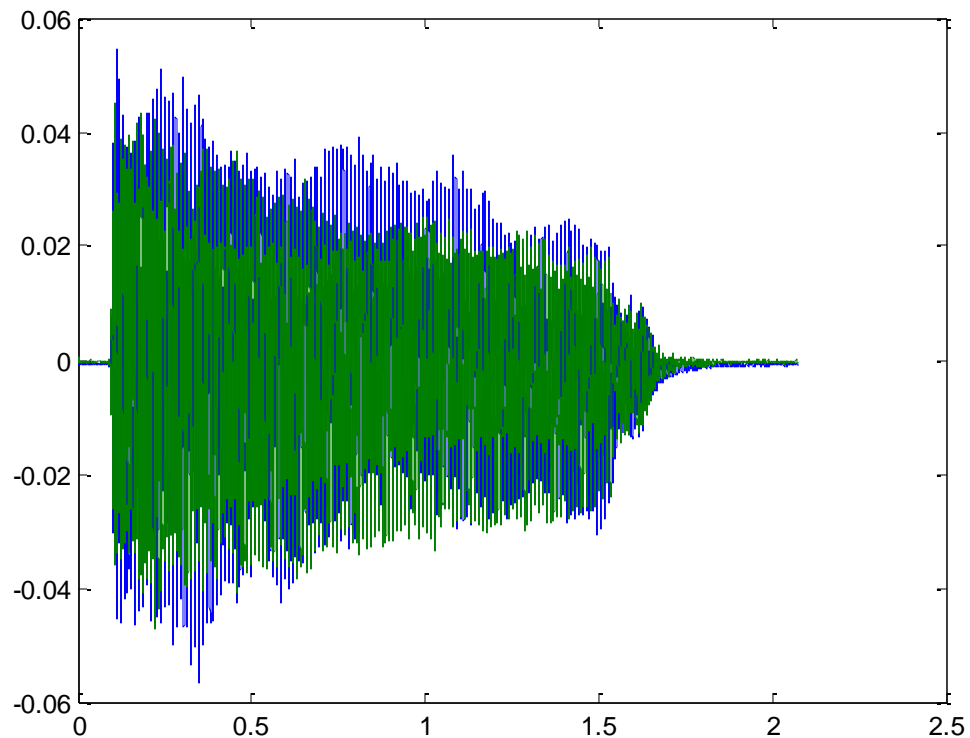


Рисунок 3.1 – Формування реалізації навчальної матриці за зображенням осцилограмми

На рис. 3.1 зелені та сині лінії відображають відповідні канали – лівий та правий. Оскільки навчальні матриці в інформаційно-екстремальній інтелектуальній технології повинні містити однакову кількість ознак розпізнавання для всіх класів, тому було вирізано перші 5000 піків а це приблизно 1/100 с. початку реклами. Таким чином, кількість ознак розпізнавання в

навчальній матриці складає 5000. Слід зауважити, що в залежності від складності звуку, яка в більшості випадків пов'язана з частотною різницею в момент часу, буде змінюватися щільність таких точок, а їх кількість залишиться не змінною. Іншим параметром навчальної вибірки є кількість реалізацій, тобто прикладів реклам. При цьому мінімальний обсяг вибірки за інформаційно-екстремальною інтелектуальною технологією дорівнює 30 реалізацій. Для прискорення процесу формування навчальної матриці кожний аудіо запис вводиться в одному екземплярі. Інші реалізації формуються автоматично шляхом часткової незначної зміни піків, сформованих за першою реалізацією.

3.2. Математична модель оптимізації параметрів навчання

На ефективність навчання інтелектуальної системи значно впливає використання контуру оптимізації системи контрольних допусків (СКД) на ознаки розпізнавання. 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;

$E_{\max}^{(l)}(x, \delta)$ – 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 $[y_{1,i}^{(j)} \pm \delta_{K,i}^{(l)}]$;

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

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

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

З урахуванням введених позначень алгоритм послідовної оптимізації поля контрольних допусків на ознаки розпізнавання приймає вигляд

$$\{\delta_{K,i}^*\} = \left\langle \arg \left\{ \max_{G_{\delta_i}} \left[\max_{G_E} \left[\bigotimes_{l=1}^L \max_{G_{d_1}} E_1^{(l)} \right] \right] \right\} \right\rangle, i = \overline{1, N}, \quad (3.1)$$

де $G_{\delta_i}, G_E, G_{d_1}$ – області допустимих значень поля контрольних допусків для i -ої ознаки, критерію оптимізації і кодової відстані d_1 відповідно; \otimes – символ операції повторення.

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

Використання послідовного алгоритму доцільно в загальному випадку, коли ОР складають різнофакторні групи. Паралельний алгоритм доцільно використовувати за умови, що ОР відбивають вплив одного фактора. Алгоритм

оптимізації СКД за зведеним полем допусків може розглядатися як узагальнення послідовного та паралельного алгоритмів за наявності різних шкал виміру ОР.

3.3. Алгоритм паралельної оптимізації системи контрольних допусків на ознаки розпізнавання

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 δ приймається інтервал $[1; \delta_H / 2]$, де δ_H – ширина нормованого поля допусків. Розглянемо кроки реалізації цього алгоритму:

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

3.4. Короткий опис програмної реалізації

Програмна реалізація запропонованої системи виконувалась за допомогою двох середовища розробки Visual Studio 2019.

Оскільки у Matlab існує математичний пакет для зчитування та перетворення аудіо сигналу, тому за допомогою додаткового пакету MLApp у

середовищі С# виконувалась генерація навчальної матриці, функція екзамену, аналіз приналежності до реклами чи не реклами, та вивід даних у графіках.

Запис аудіо проводився за допомогою програми AIMP v.6.60 build 2167(30.12.2019) з радіо ANISON.FM.

На рис. 3.2 наведено зовнішній вигляд програми.



Рисунок 3.2 – Зовнішній вигляд програми для захвату аудіо потоку з інтернет-радіо

Для формування навчальної матриці використовувалась функція `audioread`, програмний код наведено в додатку. Вихідні результати заносилися у текстовий файл.

Алгоритми оптимізації в інформаційному розумінні функціональних параметрів інтелектуальної системи верифікації підписів за інформаційно-екстремальною інтелектуальною технологією були реалізовані у вигляді користувацького інтерфейсу як показано на рис. 3.3.

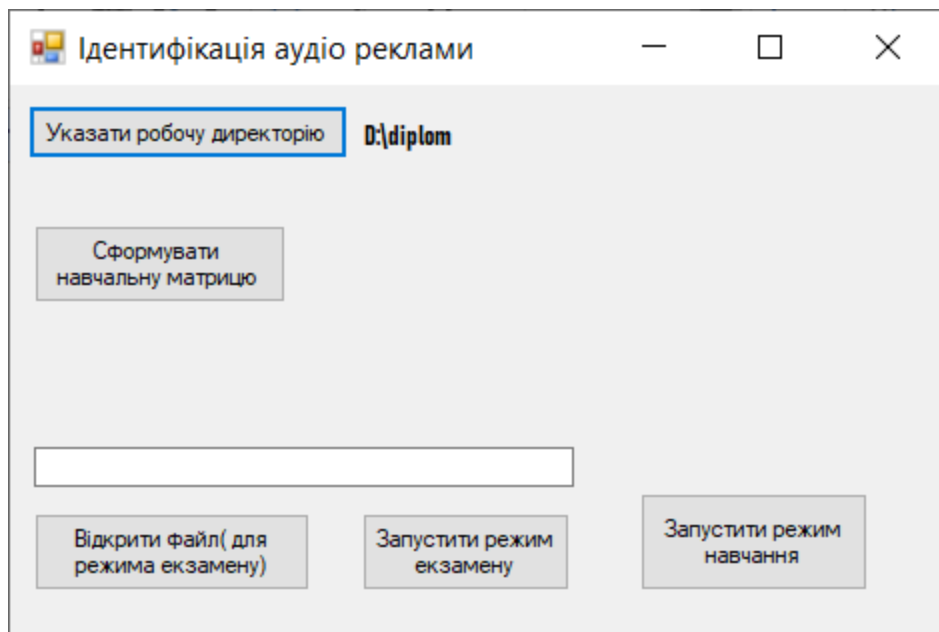


Рисунок 3.3 – Зовнішній вигляд програми

. Були використані наступні константи та змінні (Табл. 3.1):

Таблиця 3.1 Основні константи та змінні програми

Ім'я константи або змінної	Короткий опис
<code>klas = 10</code>	Кількість класів розпізнавання
<code>ozn = 5000</code>	Кількість ознак розпізнавання
<code>realiz = 30;</code>	Кількість реалізацій

Продовження таблиці 3.1

Ім'я константи або змінної	Короткий опис
<pre>int[,] NavM = new int[klas, ozn, realiz]; int[,] BinM = new int[klas, ozn, realiz];</pre>	Навчальна матриця та бінарна навчальна матриця
<pre>int[,] EtalonV = new int[klas, ozn];</pre>	Масив еталонних векторів
<pre>int[,] KodVids = new int[2, realiz];;</pre>	Масив кодових відстаней від еталонного вектора до реалізацій
<pre>double[] A = new double[klas]; double[] B = new double[klas]; double[] D1 = new double[klas]; double[] D2 = new double[klas]; double[] EM = new double[klas];</pre>	Значення точносних характеристик та КФЕ при оптимальних параметрах навчання
<pre>int[] D = new int[klas];</pre>	Оптимальні радіуси контейнерів класів навчання
<pre>int[] PARA = new int[klas];</pre>	Кодова відстань між центрами класів навчання
<pre>int[] VerhD = new int[ozn]; int[] NiznD = new int[ozn];</pre>	Система допусків на ознаки розпізнавання
<pre>String TXT, TXT_Exam;</pre>	Дескриптори файлу вхідних даних
<pre>int[] delta = new int[ozn];</pre>	Масив параметрів СКД, що використовуються при її формуванні і оптимізації.

Крім того було розроблено наступні процедури та функції:

Таблиця 3.2 Основні процедури та функції програми

Ім'я функції або процедури	Короткий опис
<code>public double INFK(double t_, double t_D1, double t_beta);</code>	Функція для розрахунку поточного значення КФЕ
<code>public void InputFiles(int num, string path);</code>	Процедура для завантаження навчальної матриці
<code>public void P_BM();</code>	Розрахунок бінарної навчальної матриці
<code>public void P_DO();</code>	Розрахунок оптимальних значень радіусів контейнерів класів розпізнавання
<code>public void P_EV();</code>	Розрахунок значень центрів контейнерів класів розпізнавання
<code>public void P_PARA();</code>	Розрахунок матриці кодових відстаней між центрами класів та знаходження найближчого.
<code>public void P_SK(int num);</code>	Розрахунок кодових відстаней між центрами класів та реалізаціями.
<code>public void SClas(int num);</code>	Збереження отриманих результатів.
<code>public void PARALEL(int l, int delta_max);</code>	Оптимізація СКД за паралельним алгоритмом

Програмний код наведено в Додатку А.

3.5. Аналіз результатів моделювання

Для перевірки працездатності інтелектуальної системи ідентифікації аудіореклами розв'язувалася задача навчання такої системи розпізнаванню 10 аудіофайлів з початком реклами. Згідно інформаційно-екстремальній технології для кожної підпису було відтворено контейнер гіперсферичної форми в

бінарному просторі Хеммінга. При оптимізації параметрів функціонування інтелектуальної системи застосовувалася алгоритми оптимізації геометричних параметрів таких контейнерів і паралельна оптимізація системи контрольних допусків на ознаки розпізнавання. На нульовій ітерації було проведено оптимізацію СКД паралельним алгоритмом (рис. 3.4). Як критерій оптимізації параметрів функціонування використовувався інформаційний міра Кульбака.

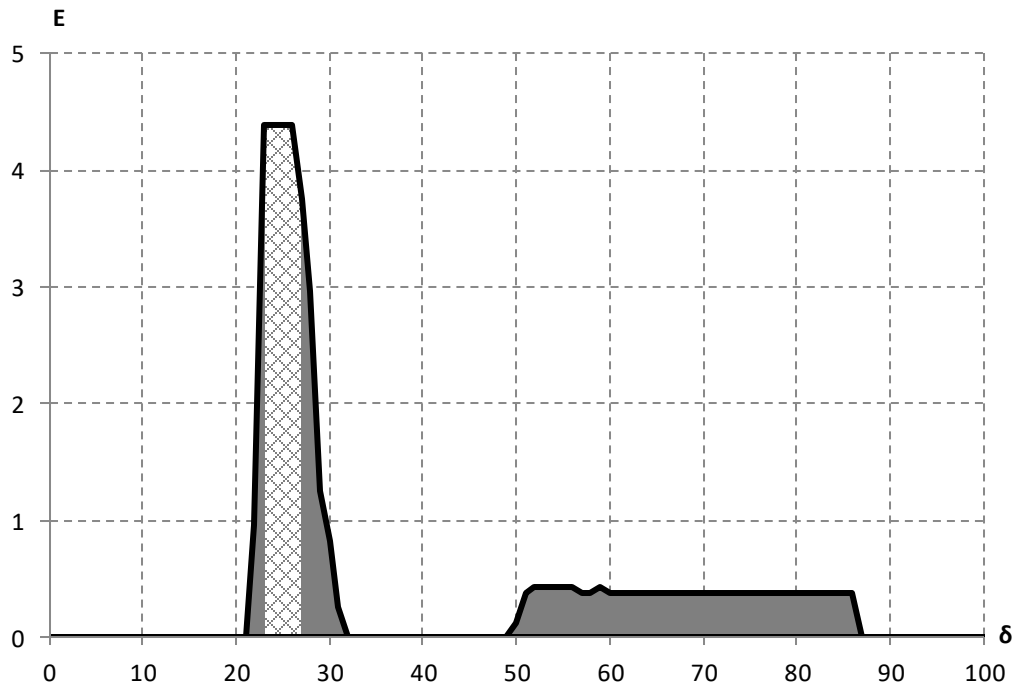
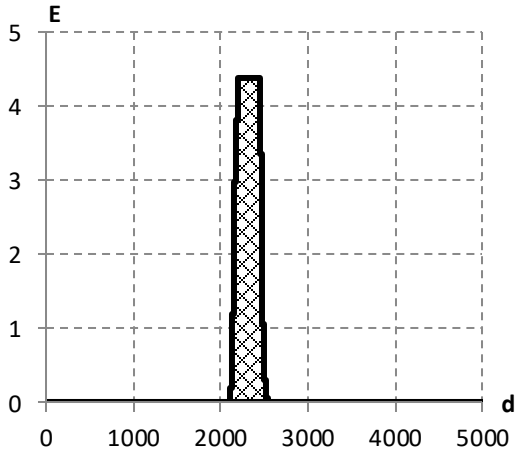
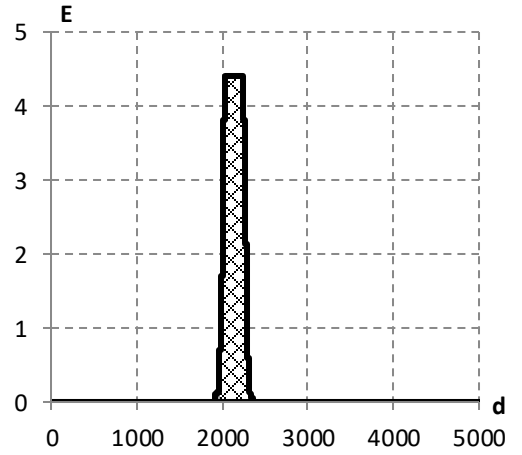


Рисунок 3.4 – Динаміка зміни усередненого КФЕ при паралельній оптимізації СКД

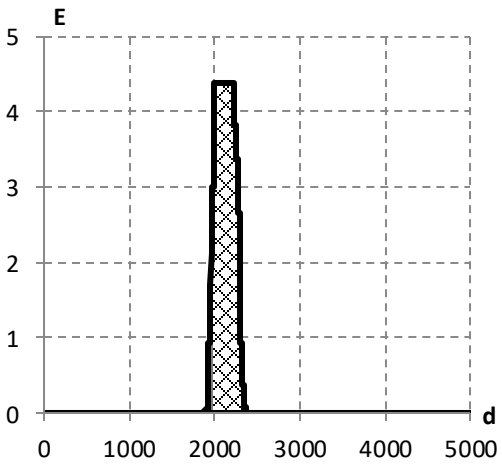
При оптимальному значенні параметра СКД $\delta^* = \pm 22$, якому відповідає максимальне значення усередненого КФЕ $\bar{E}_{\max} = 4,39$ в робочій області визначення даного параметра, проводилася оптимізація геометричних параметрів контейнерів класів розпізнавання (рис. 3.5).



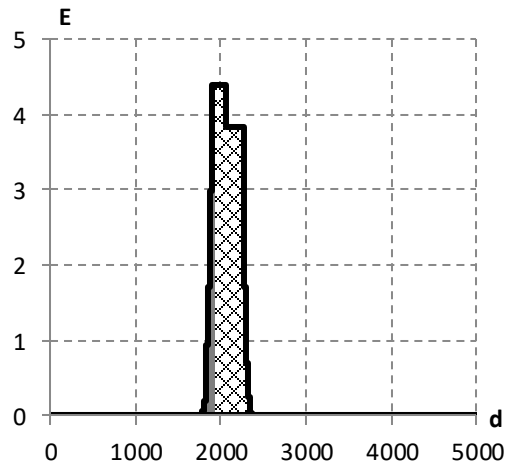
a)



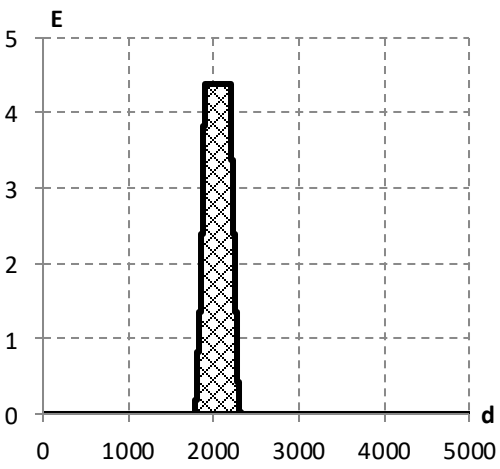
б)



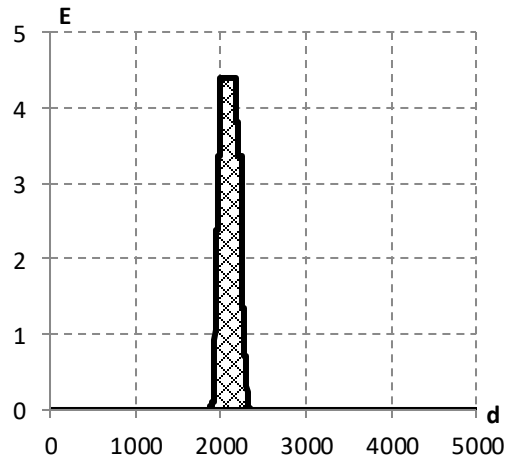
в)



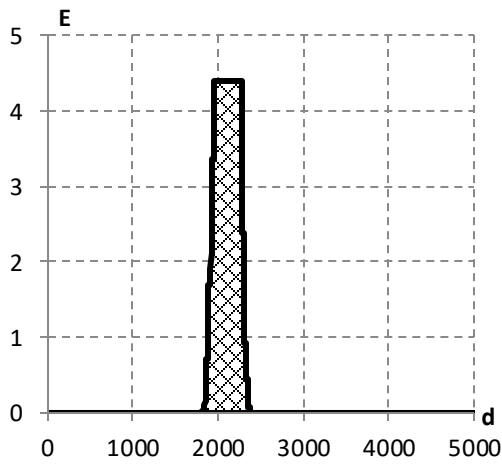
г)



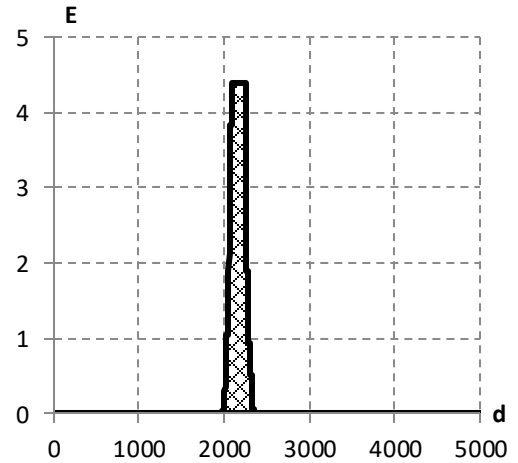
д)



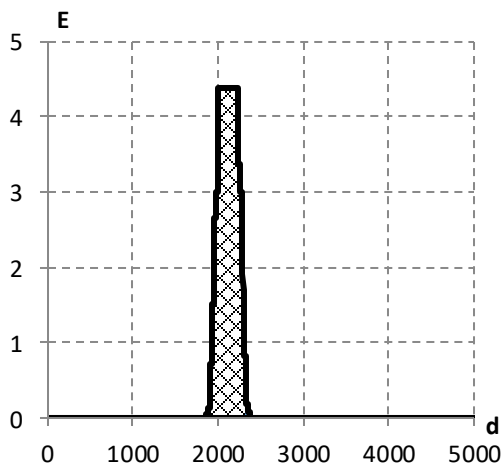
е)



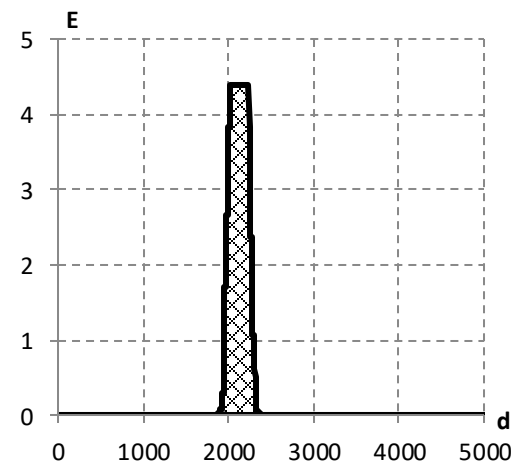
ж)



з)



и)



к)

Рисунок 3.5 – Процес оптимізації геометричних параметрів контейнерів класів розпізнавання після паралельної оптимізації СКД: а) класу X_1^o ; б) класу X_2^o ; в) класу X_3^o ; г) класу X_4^o ; д) класу X_5^o ; е) класу X_6^o ; ж) класу X_7^o ; з) класу X_8^o

Аналіз результатів оптимізації показує, що оптимальними параметрами контейнера класу X_1^o є радіус $d_1=2195$, класу X_2^o є радіус $d_2=2023$, класу X_3^o є радіус $d_3=1989$, класу X_4^o є радіус $d_4=1899$, класу X_5^o є радіус $d_5=1894$, класу X_6^o є радіус $d_6=1982$, класу X_7^o є радіус $d_7=1945$, класу X_8^o є радіус $d_8=2094$, класу

X_9^o є радіус $d_9=2002$, класу X_{10}^o є радіус $d_{10}=2025$. Цим параметрам контейнерів відповідають такі значення КФЕ і точносних характеристик системи керування: для класу $X_1^o \in E_{\max}^{(1)}=4.39232$ ($D1 = 1; \beta = 0$), для класу $X_2^o \in E_{\max}^{(2)}=4.39232$ ($D1 = 1; \beta = 0$), для класу $X_3^o \in E_{\max}^{(3)}=4.39232$ ($D1 = 0,5; \beta = 0$), для класу $X_4^o \in E_{\max}^{(4)}=4.39232$ ($D1 = 1; \beta = 0$), для класу $X_5^o \in E_{\max}^{(5)}=4.39232$ ($D1 = 0,82; \beta = 0$), для класу $X_6^o \in E_{\max}^{(6)}=4.39232$ ($D1 = 0,9; \beta = 0$), для класу $X_7^o \in E_{\max}^{(7)}=4.39232$ ($D1 = 0,85; \beta = 0$), для класу $X_8^o \in E_{\max}^{(8)}=4.39232$ ($D1 = 1; \beta = 0$), для класу $X_9^o \in E_{\max}^{(9)}=4.39232$ ($D1 = 1; \beta = 0$), для класу $X_{10}^o \in E_{\max}^{(10)}=4.39232$ ($D1 = 1; \beta = 0$). Таким чином, в процесі оптимізації параметрів функціонування інтелектуальної системи було побудовано безпомилковий класифікатор.

Для перевірки працездатності системи була сформована екзаменаційна вибірка з аудіо даними, отриманими під час запису аудіо потоку інтернет-радіо ANISON.FM.

Вибірка подана на рис. 3.6 сформована шляхом перетворень аудіо сигналу, аналогічних, що застосовувалися до навчальної вибірки, та відповідає 8 хвилинам реального звучання.

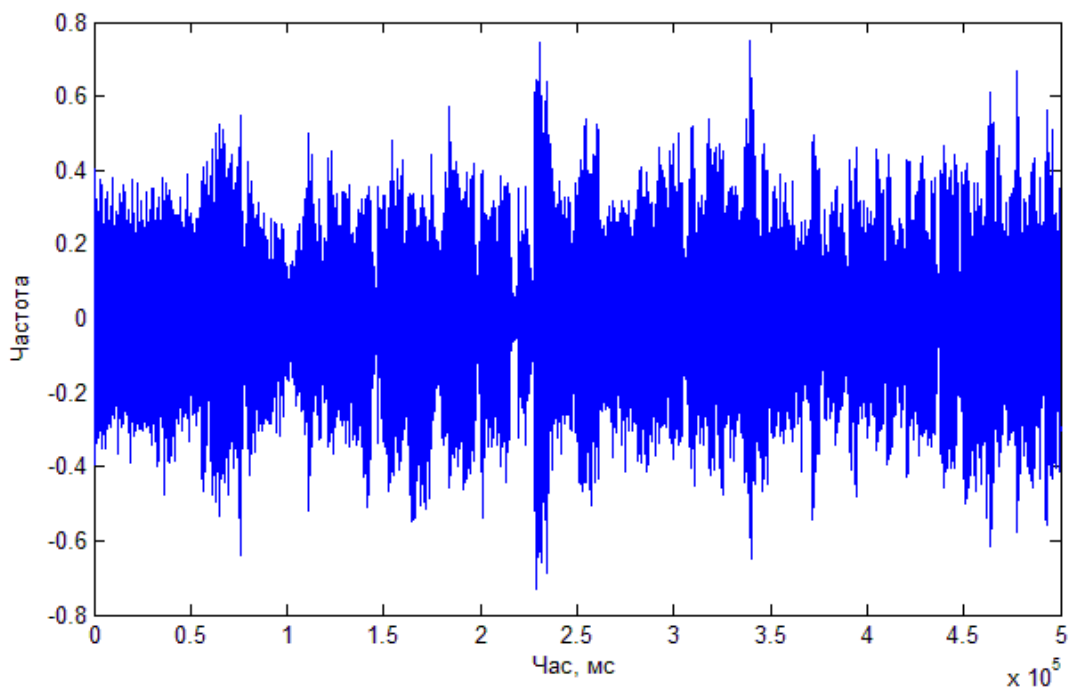


Рисунок 3.6 – Екзаменаційна аудіо вибірка

На рис 3.7 подані значення функції належності послідовності підмножин экзаменаційної вибірки до кожного з 10 класів розпізнавання.

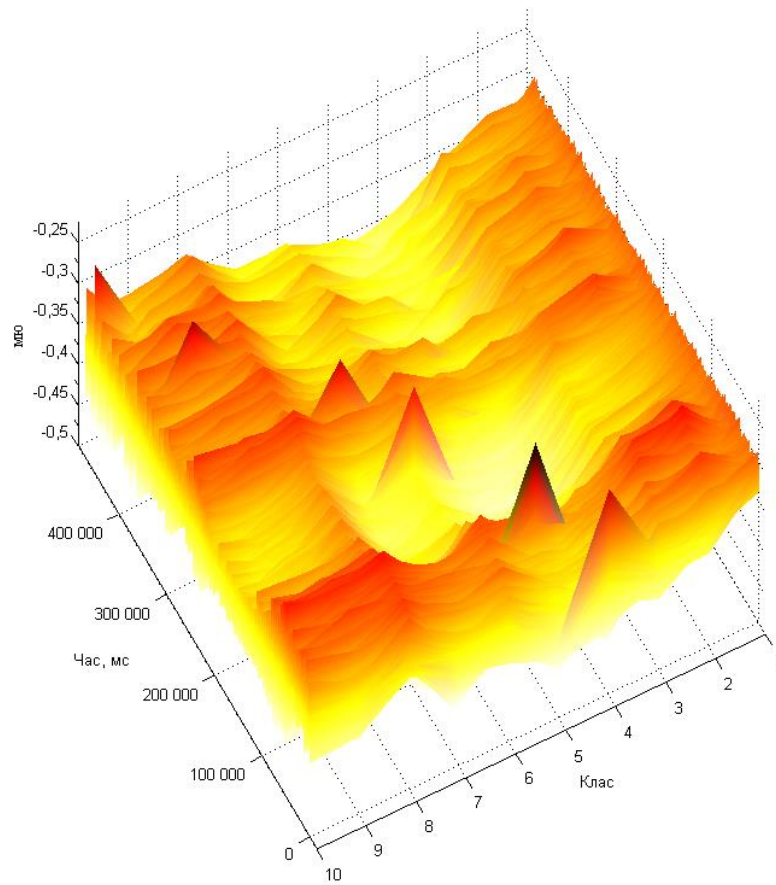
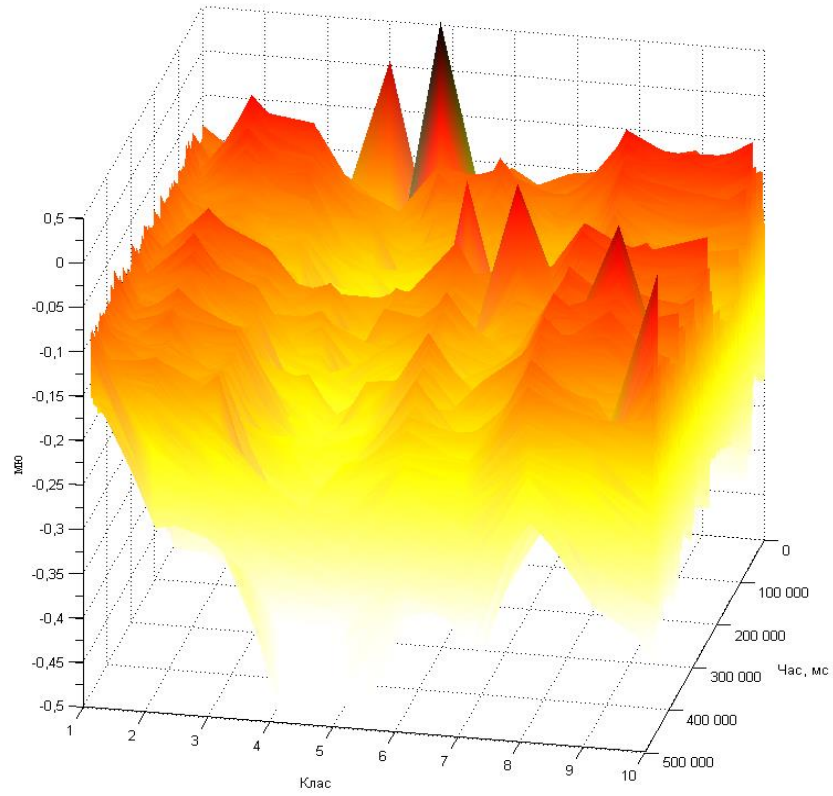


Рисунок 3.7 – Екзаменаційна аудіо вибірка

Аналіз рис. 3.7 показує що екзаменаційна вибірка містить 6 рекламних блоків. Це рекламні блоки, що відносяться до класів 3, 4, 5, 6, 9, 10. При цьому значення функції належності до відповідного контейнеру класу є додатнім, що відображається на графіку піками червоного та темно-червоного кольору.

Таким чином розроблена інтелектуальна система здатна ефективно функціонувати в режимі навчання та режимі ідентифікації

ВИСНОВКИ

Дана робота посвячена знаходженню аудіо реклами в аудіо записаних із радіо або розташованих на просторах Інтернету. Написано безліч працюючих в яких використовуються інформаційно-інтелектуальні технології.

Проблема розпізнавання як одна з складових штучного інтелекту давно приваблювала дослідників, і на сьогоднішній день хоч і досягнуті певні успіхи, вона залишається відкритою.

У роботі представлено вирішення проблеми у рамках інформаційно-екстремальної інтелектуальної технології та розробити інформаційне та програмне забезпечення інтелектуальної системи ідентифікації рекламних аудіо роликів. Сформувати математичну модель оптимізації у функціональному параметрі інтелектуальних систем з використанням інформаційних критеріїв функціональної ефективності.

СПИСОК ЛІТЕРАТУРИ

1. ООН: 4,1 мільярда людей у світі підключені до інтернету [Electronic resource]. URL: <https://www.radiosvoboda.org/a/news-oon-internet/30254946.html> .
2. Евгений Викторович Ромат. Реклама [Electronic resource]. URL: https://books.google.com.ua/books?hl=uk&lr=&id=D7O6jELubHUC&oi=fnd&pg=PA6&dq=реклама+&ots=NjM4tG803y&sig=HCuEkSLyvo0Y3cPSCl0W43woLLg&redir_esc=y#v=onepage&q=реклама&f=false.
3. Аудиореклама в цифровом формате [Electronic resource]. URL: <https://www.elibrary.ru/item.asp?id=38226725>.
4. Мудров А. Н. Основы рекламы: учебник/ А. Н. Мудров. -2-е изд., перераб. и доп.-М. : Магистр, 2008.-397 с. : ил.
5. Разработка Adblock Radio / Хабр [Electronic resource]. URL: <https://habr.com/ru/post/435720/> .
6. Andrew A.M. Automatic Fingerprint Recognition Systems [Electronic resource] // Kybernetes. 2005. Vol. 34, № 6. P. 902. URL: <https://books.google.com.ua/books?id=7y34BwAAQBAJ&pg=PA233&lpg=PA233&dq=landmark+Фінгерпрінтинг&source=bl&ots=ybDeX0Hulk&sig=ACfU3U2AV1N2CvA9enNql6M4aQnv0ell0g&hl=ru&sa=X&ved=2ahUKEwjXq9Wu7bDpAhUqyKYKHauzALUQ6AEwD3oECBgQAQ#v=onepage&q=landmark+Фінгерпрінт>.
7. Professor G. et al. A Landmark Algorithm for the Time-Dependent Shortest Path Problem. 2008.
8. Фінгерпрінтинг: что это такое? Фінгерпрінтинг браузера и не только [Electronic resource]. URL: <http://www.spy-soft.net/fingerprinting/>.
9. Звуковые отпечатки: распознавание рекламы на радио / Хабр [Electronic resource]. URL: <https://habr.com/ru/post/252937/>.
10. Liu Q. et al. Source separation of convolutive and noisy mixtures using audio-visual dictionary learning and probabilistic time-frequency masking // IEEE

- Trans. Signal Process. 2013. Vol. 61, № 22. P. 5520–5535.
11. Announcing the Initial Release of Mozilla’s Open Source Speech Recognition Model and Voice Dataset - The Mozilla Blog [Electronic resource]. URL: <https://blog.mozilla.org/blog/2017/11/29/announcing-the-initial-release-of-mozillas-open-source-speech-recognition-model-and-voice-dataset/>.
 12. Glybovets M., Zhyrkova A. Using machine learning in sound classification tasks // Nauk. Res. Pap. Comput. Sci. National University of Kyiv - Mohyla Academy, 2019. Vol. 2, № 0. P. 22–31.
 13. Аналіз впливу параметрів обробки звукового сигналу на якість розпізнавання голосових команд [Electronic resource]. URL: <https://cyberleninka.ru/article/n/analiz-vplivu-parametriv-obrobki-zvukovogo-signalu-na-yakist-rozpiznavannya-golosovih-komand/viewer>.
 14. Wang A.L., Street K.H. An Industrial-Strength Audio Search Algorithm // International Conference on Music Information Retrieval ISMIR. 2000. Vol. 2, № 2. 7–13 p.
 15. Довбиш А.С., Шелехов І.В. ОСНОВИ ТЕОРІЇ РОЗПІЗНАВАННЯ ОБРАЗІВ Навчальний посібник. Вид-во СумДУ, 2015.
 16. Довбиш А.С. et al. Оцінка інформативності ознак розпізнавання за методом функціонально-статистичних випробувань. Видавництво СумДУ, 2006.
 17. Довбиш А.С., Довбыш А.С., Dovbysh A.S. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ В ПРОМИСЛОВОСТІ І ОСВІТІ. Видавництво СумДУ, 2007.
 18. Довбиш А.С. et al. Розроблення науково-методичних основ та інформаційних засобів проектування здатних самонавчатися адаптивних систем керування технологічними процесами. Видавництво СумДУ, 2010.
 19. Заговора О.В., Концевич В.Г. Алгоритм прогностического обучения информационно-экстремальной интеллектуальной технологии для анализа результатов исследований машин динамического действия [Electronic resource]. URL: <http://web.snauka.ru/issues/2014/09/38334>.

20. Бібик М.В., Довбиш А.С. Оптимізація контрольних допусків на ознаки розпізнавання під час машинного навчання // Journal of Engineering Sciences. Сумський державний університет, 2017. Vol. 4, № 1.

ДОДАТОК А

```

using Microsoft.WindowsAPICodePack.Dialogs;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        String fileContent = string.Empty;
        String filePathLearning, filePathExam;
        String pathDirectory = "C://";

        const int klas = 10; // класи
        const int ozn = 5000; // ознаки
        const int realiz = 30; // реалізації

        // навчальна та бінарна навчальні матриці
        int[, ,] NavM = new int[klas, ozn, realiz];
        int[, ,] BinM = new int[klas, ozn, realiz];
        //система контрольних допусків
        int[] VerhD = new int[ozn];
        int[] NiznD = new int[ozn];
        // еталонні вектори (центри контейнерів класів)
        int[, ] EtalonV = new int[klas, ozn];
        // кодові відстані від еталонного вектору до реалізацій [1] своїх [2] сусідніх
        int[, ] KodVids = new int[2, realiz];
        //точнісні характеристики та критерій функціональної ефективності
        double[] A = new double[klas];
    }
}

```

```

double[] B = new double[klas];
double[] D1 = new double[klas];
double[] D2 = new double[klas];
double[] EM = new double[klas]; // of realiz;
                                // змінні коли відсутня робоча область

double[] A_NRO = new double[klas];
double[] B_NRO = new double[klas];
double[] D1_NRO = new double[klas];
double[] D2_NRO = new double[klas];
double[] EM_NRO = new double[klas]; // of realiz;
// оптимальні радіуси контейнерів класів
int[] D = new int[klas];
//радіуси контейнерів класів, коли нема робочої області
int[] D_NRO = new int[klas];
//номер сусіднього класу
int[] PARA = new int[klas];
// лічильники
int l;
//міжцентрова відстань
int[] dc = new int[klas];
// змінні файлів
String TXT, TXT_Exam;
String MyFile;
int[] tclass = new int[ozn];
int[] delta = new int[ozn];
//ширина коридору СКД
int tdelta;
// змінні для виводу стану
bool VIVOD;
//змінні для виводу звіту оптимізації СКД
bool VIVOD_SKD;
//лічильник Ітерацій
int LICH;

public void InputFiles(int num, string path)
{
    string fClass;

```

```

Console.WriteLine("MyFile for {0} class:(ENTER={1})", num+1, path);
StreamReader file = new StreamReader(path);
string s = file.ReadToEnd();
file.Close();
string[] line = s.Split('\n');
string[] column = line[0].Split(' ');
int[,] a = new int[line.Length, column.Length];
int t = 0;
int n = 0;
for (int j = 0; j < realiz; j++)
{
    column = line[j].Split(' ');
    for (int i = 0; i < ozn; i++)
    {
        t = Convert.ToInt32(column[j]);
        NavM[num, i, j] = t;
    }
}

public void P_Delta(int td)
{
    for (int delta_i = 0; delta_i < ozn; delta_i++)
    {
        delta[delta_i] = td;
    }
}

public void P_SD()
{
    int sum;
    for (int i = 0; i < ozn; i++)
    {
        sum = 0;
        for (int j = 0; j < realiz; j++)
        {
            sum += NavM[tclass[i], i, j];
        }
    }
}

```

```

        sum = sum / realiz;
        VerhD[i] = sum + delta[i];
        NiznD[i] = sum - delta[i];
    }
}

public void P_SD2(int il)
{
    int sum = 0;
    for (int i = 0; i < realiz; i++)
    {
        sum += NavM[tclass[il], il, i];
        sum = sum / realiz;
        VerhD[il] = sum + delta[il];
        NiznD[il] = sum - delta[il];
    }
}

public void P_EV()
{
    int sum;
    for (int k = 0; k < klas; k++)
    {
        for (int i = 0; i < ozn; i++)
        {
            sum = 0;
            for (int j = 0; j < realiz; j++)
            {
                sum += BinM[k, i, j];
                if ((double)(sum / realiz) >= 0.5)
                {
                    EtalonV[k, i] = 1;
                }
            }
            else

```

```
        {
            EtalonV[k, i] = 0;
        }
    }
}

public void P_BM()
{
    for (int k = 0; k < klas; k++)
    {
        for (int i = 0; i < ozn; i++)
        {
            for (int j = 0; j < realiz; j++)
            {
                if ((NavM[k, i, j] >= NiznD[i]) && (NavM[k, i, j] <= VerhD[i]))
                {
                    BinM[k, i, j] = 1;
                }
                else
                {
                    BinM[k, i, j] = 0;
                }
            }
        }
    }
}

public void P_SK(int num)
{
    for (int j = 0; j < realiz; j++)
    {
        KodVids[0, j] = 0;
        KodVids[1, j] = 0;
        for (int i = 0; i < ozn; i++)
        {
```

```

        KodVids[0, j] = KodVids[0, j] + Math.Abs(EtalonV[num, i] - BinM[num, i,
j]);
        KodVids[1, j] = KodVids[1, j] + Math.Abs(EtalonV[num, i] -
BinM[PARA[num], i, j]);
    }
}

```

```

public void P_PARA()
{
    int sum;
    for (int k = 0; k < klas; k++)
    {
        PARA[k] = 1;
        dc[k] = ozn;
        for (int kc = 0; kc < klas; kc++)
        {
            if (k != kc)
            {
                sum = 0;
                for (int i=0; i < ozn; i++)
                {
                    sum += Math.Abs(EtalonV[k, i] - EtalonV[kc, i]);
                }
                if (sum <= dc[k])
                {
                    PARA[k] = kc;
                    dc[k] = sum;
                }
            }
        }
    }
}

```

```

public double INFK(double t_, double t_D1, double t_beta)
{
    int a1 = 0, a2 = 0, a3 = 0, a4 = 0;
    double d1_b;

```



```

for (int j = 0; j < realiz; j++)
{
    if (KodVids[0, j] <= t_) a1++;
    if (KodVids[1, j] <= t_) a3++;
}
a4 = realiz - a3;
a2 = realiz - a1;
t_D1 = a1 / realiz;
t_beta = a3 / realiz;
d1_b = t_D1 - t_beta;
double res = d1_b * Math.Log((1 + d1_b + 0.1) / (1 - d1_b + 0.1)) / Math.Log(2);
//Полный критерий Кульбака, выраженный через D1 и beta
LICH++;
return res;
}

public void SClas(int num)
{
    try
    {
        using (StreamWriter sw = new StreamWriter(TXT, false,
System.Text.Encoding.Default))
        {
            sw.WriteLine("Klas {0}", num);
            sw.WriteLine("BinM[{0}]", num);
            for (int i = 0; i < ozn; i++)
            {
                for (int j = 0; j < realiz; j++)
                {
                    sw.Write("{0} ", BinM[num, i, j]);
                }
                sw.WriteLine("");
            }
            sw.WriteLine("EtalonV[{0}]", num);
            for (int i = 0; i < ozn; i++)
            {
                sw.Write("{0} ", EtalonV[num, i]);
            }
        }
    }
}

```

```

        sw.WriteLine("");
        sw.WriteLine("PARA[{0}] = {1}", num, PARA[num]);
        sw.WriteLine("dc[{0}] = {1}", num, dc[num]);
        sw.WriteLine("d E D1 Betta Alfa D2 K1 K2 K3 K4");
    }
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
}

public void P_DO()
{
    double te, tD1, tbeta = 0;
    int t, ti;
    TXT_Exam = pathDirectory + "\\rez.txt";
    try
    {
        using (StreamWriter txtExam = new StreamWriter(TXT_Exam, false,
            System.Text.Encoding.Default))
        {
            if (VIVOD) {
                for (ti = 0; ti < ozn; ti++)
                {
                    txtExam.Write("{0} {1}", VerhD[ti], NiznD[ti]);
                }
                txtExam.WriteLine("");
                for (int k = 0; k < klas; k++)
                {
                    if (VIVOD)
                    {
                        TXT = pathDirectory + "\\rez" + (k+1) + ".txt";
                        SClas(k);
                    }
                    EM[k] = 0;
                    EM_NRO[k] = 0;
                }
            }
        }
    }
}

```

```

D[k] = 0;
P_SK(k);
tD1 = 1.0;
t = dc[k];

while ((t >= 0) && (tD1 >= 0.5))
{
    te = INFk(t, tD1, tbeta);
    if (VIVOD)
    {
        try
        {
            using (StreamWriter txt = new StreamWriter(TXT,
false, System.Text.Encoding.Default))
            {
                txt.WriteLine("{0} {1} {2} {3} {4} {5} {6} {7}
", t, te, tD1, tbeta, 1 - tD1, 1 - tbeta, Math.Round(tD1 * realiz), Math.Round((1 - tD1) *
realiz), Math.Round((1 - tbeta) * realiz), Math.Round(tbeta * realiz));
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
        if (te > EM_NRO[k])
        {
            EM_NRO[k] = te;
            D_NRO[k] = t;
            D1_NRO[k] = tD1;
            B_NRO[k] = tbeta;
        }
        if ((tD1 >= 0.5) && (tbeta < 0.5) && (t < dc[k])) {
            if (te > EM[k]) {
                EM[k] = te;
                D[k] = t;
                D1[k] = tD1;
                B[k] = tbeta;
            }
        }
    }
}

```



```

public void P_DO_star()
{
    double te, tD1, tbeta = 0;
    int t, ti;
    TXT_Exam = pathDirectory + "\\rez.txt";
    try
    {
        using (StreamWriter txtExam = new StreamWriter(TXT_Exam, false,
System.Text.Encoding.Default))
        {
            if (VIVOD)
            {
                for (ti = 0; ti < ozn; ti++)
                {
                    txtExam.Write("{0} {1}", VerhD[ti], NiznD[ti]);
                }
                txtExam.WriteLine("");
                for (int k = 0; k < klas; k++)
                {
                    if (VIVOD)
                    {
                        TXT = pathDirectory + "\\rez" + k + ".txt";
                    }
                    EM[k] = 0;
                    EM_NRO[k] = 0;
                    D[k] = 0;
                    P_SK(k);
                    tD1 = 1;
                    for (t = 0; t < ozn; t++)
                    {
                        te = INFk(t, tD1, tbeta);
                        if (VIVOD)
                        {
                            try
                            {
                                using (StreamWriter txt = new StreamWriter(TXT,
false, System.Text.Encoding.Default))
                                {

```

```

        txt.WriteLine("{0} {1} {2} {3} {4} {5} {6} {7}
", t, te, tD1, tbeta, 1 - tD1, 1 - tbeta, Math.Round(tD1 * realiz), Math.Round((1 - tD1) *
realiz), Math.Round((1 - tbeta) * realiz), Math.Round(tbeta * realiz));

    }
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
if (te > EM_NRO[k])
{
    EM_NRO[k] = te;
    D_NRO[k] = t;
    D1_NRO[k] = tD1;
    B_NRO[k] = tbeta;
}
if ((tD1 >= 0.5) && (tbeta < 0.5) && (t < dc[k]))
{
    if (te > EM[k])
    {
        EM[k] = te;
        D[k] = t;
        D1[k] = tD1;
        B[k] = tbeta;
    }
}
// t--;
}
if (VIVOD)
{
    try
    {
        using (StreamWriter txt = new StreamWriter(TXT,
false, System.Text.Encoding.Default))
        {
            txt.WriteLine("Class {0} Em={1} do={2} dc={3}
D1={4} Betta{5}", k, EM[k], D[k], dc[k], D1[k], B[k]);
        }
    }
}

```



```

for (i_ds = 0; i_ds < delta_max; i_ds++)
{
    if (i_ds % 20 == 0)
    {
        Console.WriteLine("{0} - ", i_ds);
    }
    P_Delta(i_ds);
    P_SD();
    P_BM();
    P_EV();
    P_PARA();
    P_DO();
    a = true;
    for (my_i = 0; my_i < klas; my_i++)
    {
        if (D[my_i] == 0)
        {
            a = false;
        }
    }
    em_avg = 0;
    for (my_i = 0; my_i < klas; my_i++)
    {
        em_avg += EM[my_i];
    }
    em_avg = em_avg / klas;
    if((em_avg>e0) && a)
    {
        e0 = em_avg;
        j_ds = i_ds;
    }
    if (VIVOD_SKD)
    {
        try
        {
            using (StreamWriter f01 = new StreamWriter(f1, false,
System.Text.Encoding.Default))

```



```

        {
            f01.WriteLine("{0} {1} ", ((VerhD[0]-NiznD[1])/2.0),em_avg);
            for (my_i = 0; my_i < klas; my_i++)
            {
                f01.Write("{0} {1} {2} {3} {4} {5}", EM[my_i], D[my_i],
D1[my_i], B[my_i], PARA[my_i], dc[my_i]);
            }
            f01.WriteLine("");
        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
}
//i_ds++;
}
Console.WriteLine();
P_Delta(j_ds);
P_SD();
P_BM();
P_EV();
P_PARA();
VIVOD = true;
P_DO_star();
}

public void PBC1()
{
    int l = 1;
    double esr;
    for(int t = 0; t < ozn; t++)
    {
        tclass[t] = l;
    }
    VIVOD = false;
    LICH = 0;
}

```

```

    PARALEL(1, 100);
    Console.WriteLine(LICH);
    VIVOD = true;
    P_SD();
    P_BM();
    Console.WriteLine("BinM..Done");
    P_EV();
    Console.WriteLine("EtalonV..Done");
    Console.WriteLine("PARA");
    P_PARA();
    Console.WriteLine("PARA..Done");
    P_DO();
    esr = 0;
    for(int k = 0; k < klas; k++)
    {
        esr += EM[k];
    }
    Console.WriteLine("E={0}",esr/klas);
    Console.WriteLine("Teach..Done");
}

// Метод случайного поиска базового класса для признака
public void PBC2(int t_max)
{
    int t, tk, l = 0;
    Random random = new Random();
    try
    {
        using (StreamWriter frez2 = new StreamWriter((pathDirectory +
"\unbase2.txt"), false, System.Text.Encoding.Default))
        {
            for (t = 0; t < t_max; t++)
            {
                for (l = 0; l < ozn; l++)
                {
                    tclass[l] = (int)Math.Round(random.NextDouble() * (klas - 1)) +
1;
                }
            }
        }
    }
}

```

```

        PARALEL(1, 100);
        //      SKD02(1, 100);
        //VIVOD = true;
        //P_SD();
        //P_BM();
        //Console.WriteLine("BinM..Done");
        //P_EV();
        //Console.WriteLine("EtalonV..Done");
        //Console.WriteLine("PARA");
        //P_PARA();
        //Console.WriteLine("PARA..Done");
        //P_DO();
        //
    }
    Console.WriteLine("E={0}", (EM[1] + EM[2] + EM[3]) / 3.0);
    for (tk = 0; tk < klas; tk++)
    {
        frez2.Write("{0} {1} {2} {3} {4} {5}", EM[tk], D[tk], D1[tk], B[tk],
        PARA[tk], dc[tk]);
    }
    frez2.WriteLine("");
}
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
}

public void PBC3()
{
    int t, tk, tj, tkmin;
    double disp,sr,dispmin;
    for (t = 0; t < ozn; t++)
    {
        tkmin = 0;

```

```

        dispmin = 0;
        for (tk = 0; tk < klas; tk++)
        {
            disp = 0;
            sr = 0;
            for (tj = 0; tj < realiz; tj++)
            {
                disp += Math.Pow(NavM[tk, t, tj], 2);
                sr += NavM[tk, t, tj];
            }
            disp += NavM[tk, t, tj] * NavM[tk, t, tj];
            if ((tkmin == 0) || (disp < dispmin))
            {
                dispmin = disp;
                tkmin = tk;
            }
        }
        tclass[t] = tkmin;
    }
    VIVOD = false;
    PARALEL(1, 100);
    VIVOD = true;
    P_SD();
    P_BM();
    Console.WriteLine("BinM..Done");
    P_EV();
    Console.WriteLine("EtalonV..Done");
    Console.WriteLine("PARA");
    P_PARA();
    Console.WriteLine("PARA..Done");
    P_DO();
    Console.WriteLine("E={0}", ((EM[1] + EM[2] + EM[3]) / 3.0));
    Console.WriteLine("TEACH..Done");
}

public Form1()
{

```

```
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void button5_Click(object sender, EventArgs e)
    {
        using (OpenFileDialog openFileDialog = new OpenFileDialog())
        {
            openFileDialog.InitialDirectory = pathDirectory;
            openFileDialog.RestoreDirectory = true;

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                filePathExam = openFileDialog.FileName;
                textBox2.Text = filePathExam;
            }
        }
    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {
        Form wait = new Form();
        wait.Show();
        wait.Text = "Wait...";
        MApp.MLApp matlab = new MApp.MLApp();
        matlab.Execute("for t = 1:10" + "\n" +
            "clear;" + "\n" +
```

```

"fid = fopen('" + pathDirectory + "\\rez.txt');" + "\n" +
"a = fscanf(fid, '%g %g',[2, 5000]);" + "\n" +
"for k = 1:10" + "\n" +
"m = fscanf(fid, '%g',[1, 1]);" + "\n" +
"ev(k,:) = fscanf(fid, '%g',[1, 5000]);" + "\n" +
"d_o(k) = fscanf(fid, '%g',[1, 1]);" + "\n" +
"end;" + "\n" +
"[d, sr] = audioread('" + filePathLearning + "');" + "\n" +
"p =[d];" + "\n" +
"i_max =size(p,2);" + "\n" +
"for t=1:round(-0.5+(i_max-5000))" + "\n" +
"j =(t-1)+1;" + "\n" +
"if (j/10000)==round(j/10000) j, end;" + "\n" +
"P =p(j:(5000+j));" + "\n" +
"P1 =(P+1)*127.5;" + "\n" +
"for i=1:5000" + "\n" +
"ekz_bm(i)=(P1(i)<=a(1, i))&(P1(i)>=a(2, i));" + "\n" +
"end;" + "\n" +
"for k=1:10" + "\n" +
"mu(k, t)=1-(sum(xor(ev(k,:),ekz_bm))./d_o(k));" + "\n" +
"end;" + "\n" +
"end;" + "\n" +
"surf(mu)" + "\n");

matlab.Quit();
wait.Close();
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    Form wait = new Form();
    wait.Show();
    wait.Text = "Wait...";
    MLabApp.MLabApp matlab = new MLabApp.MLabApp();
    matlab.Execute("for t = 1:10" + "\n" +
        "file_name = strcat('" + pathDirectory + "\\',num2str(t));" + "\n" +
        "file_name = strcat(file_name, '.mp3');" + "\n" +
        "[d, sr] = audioread(file_name);" + "\n" +

```

```

        "size(d)" + "\n" +
        "P = d(1:5000);" + "\n" +
        "P = (P + 1) * 127.5;" + "\n" +
        "for i = 1:30" + "\n" +
        "pp(i,:) = round(P + rand(1, 5000) * 100 - 50);" + "\n" +
        "end;" + "\n" +
        "file_name = strcat('\" + pathDirectory + "\\pp',num2str(t));" + "\n" +
        "file_name = strcat(file_name, '.txt');" + "\n" +
        "fid = fopen(file_name, 'w');" + "\n" +
        "for i = 1:30" + "\n" +
        "fprintf(fid, '%d ', pp(i,:));" + "\n" +
        "fprintf(fid, '\\n');" + "\n" +
        "end;" + "\n" +
        "fclose(fid);" + "\n" +
        "end;" + "\n");
    matlab.Quit();
    wait.Close();
}

```

```

private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
}

```

```

private void button6_Click(object sender, EventArgs e)
{
    VIVOD = true;
    VIVOD_SKD = true;
    string pathFiles;
    Console.WriteLine("LOAD");
    Random random = new Random();
    for(int k = 0; k < klas; k++)
    {
        pathFiles = pathDirectory + "\\pp" + (k+1) + ".txt";
        InputFiles(k, pathFiles);
    }
    Console.WriteLine("LOAD..Done");
    Console.WriteLine("TEACH");
}

```

```
        PBC1();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        using (OpenFileDialog openFileDialog = new OpenFileDialog())
        {
            openFileDialog.InitialDirectory = pathDirectory;
            openFileDialog.RestoreDirectory = true;

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                filePathLearning = openFileDialog.FileName;
            }
        }
    }

    private void button4_Click(object sender, EventArgs e)
    {
        CommonOpenFileDialog dialog = new CommonOpenFileDialog();
        dialog.InitialDirectory = pathDirectory;
        dialog.IsFolderPicker = true;
        if (dialog.ShowDialog() == CommonFileDialogResult.Ok)
        {
            label1.Text = dialog.FileName;
            pathDirectory = dialog.FileName;
        }
    }
}
}
```