Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання”

Виконав здобувач вищої освіти
ІІ курсу, групи КН-25М
ОПП «Комп’ютерні науки»
спеціальності F3 «Комп’ютерні науки»
_____ Щербак О.Ю.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
«____» _____ 2026 р.

Нормоконтроль проекту
_____ Коваленко А.С.
«____» _____ 2026 р.

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Щербак Олександр Юрійович
Сертифікат: 514B5C86A1E5DA1104000000C2DF62002F797705
Дійсний з 08.03.2026 20:52:44 по 08.03.2027 20:52:43

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань _F_____*"*Інформаційні технології*"*_____

Спеціальність F3 *"*Комп'ютерні науки*"*

Освітньо-професійна (освітньо-наукова) програма _*"*Комп'ютерні науки*"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 1 » вересня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Щербаку Олександру Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання*

2. Керівник роботи *Смірнов Сергій Анатолійович, кандидат технічних наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 від 17.08.2026 року

3. Строк подання студентом роботи до захисту *2.12.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*5. Впровадження системи в промислову експлуатацію*
2. Перегляд аналогічних існуючих систем.	*6. Наукова новизна*.
3. Опис і обґрунтування проектних рішень.	*7. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*8. Висновки.*

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

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2026	16.11.2026

8. Дата видачі завдання « 1 » вересня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2026 р.	
3.	Розробка моделі компонента	20.10.2026 р.	
4.	Розробка структур даних	25.10.2026 р.	
5.	Розробка алгоритмів системи	30.10.2026 р.	
6.	Програмування алгоритмів	10.11.2026 р.	
7.	Розрахунки з охорони праці та техніки безпеки	15.11.2026 р.	
8.	Оформлення ПЗ	17.11.2026 р.	
9.	Попередній захист роботи	2.12.2026 р.	

Дата видачі завдання
« 1 » вересня 2026 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 1 » вересня 2026 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Щербак О.Ю. Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання. ФЗ Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Об'єктом дослідження є процес інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Предметом дослідження є методи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Методи дослідження базуються на методах хмарних обчислень, методах машинного навчання, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Visual C#.

Ключові слова: комп’ютерні науки, інтелектуальна оптимізація, продуктивність, хмарні сховища даних, машинне навчання

ABSTRACT

Shcherbak O.Yu. Research and Software Implementation of an Intelligent Performance Optimization System for Cloud Data Storage Using Machine Learning Algorithms. F3 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification thesis for the second (master's) level of higher education, software has been developed for an intelligent performance optimization system for cloud data storage, utilizing machine learning algorithms.

The objective of this development is the research and software implementation of an intelligent performance optimization system for cloud data storage using machine learning algorithms.

The object of the research is the process of intelligently optimizing the performance of cloud data storage using machine learning algorithms.

The subject of the research consists of methods for the intelligent optimization of cloud data storage performance using machine learning algorithms.

The research methodology is based on approaches from cloud computing, machine learning, mathematical statistics, and software engineering.

The outcome of the work is a software implementation of an intelligent performance optimization system for cloud data storage using machine learning algorithms.

During the development of the software model, an analysis of existing hardware and software tools was conducted. All components of the developed software are described in detail.

A user-friendly interface has been developed. Instructions for operating the software tools are provided. The program can be used on PCs running Windows 10/11.

The program was developed in the Visual C# environment.

Keywords: computer science, intelligent optimization, performance, cloud data storage, machine learning

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	83	–	
2	А4	Робоча програма	56	–	Додаток А
		Графічні документи			
3	А4	Наукова новизна	1	–	
4	А4	Структурна схема	1	–	
5	А4	Функціональна схема роботи системи	1	–	
6	А4	Діаграма взаємодії процесів	1	–	
7	А4	Блок-схема основної програми	1	–	
8	А4	Блок-схема роботи підпрограми	1	–	

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 10
ВСТУП..... 11
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 14
1.1 Призначення системи..... 14
1.2 Область застосування 14
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 16
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 16
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 20
2.3 Розгорнута постановка завдання 21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 23
3.1 Опис функціонування системи 23
3.2 Розробка структурної схеми..... 28
3.3 Розробка функціональної схеми 36
3.4 Розробка діаграми процесів 38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 40
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 60
6 НАУКОВА НОВИЗНА 64

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 65
7.1 Вступ 65
7.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ... 67
7.3 Розробка заходів з умов поліпшення охорони праці 70
7.4 Розрахункова частина 71
7.5 Висновки до розділу 73
8 ОСНОВНІ ВИСНОВКИ 74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АВІ – адміністратора віртуальної інфраструктури

АІБ – адміністратор інформаційної безпеки

ВІ – віртуальна інфраструктура

ЗЗІ – засоби захисту інформації

ПД – персональні дані

ПЗ – програмне забезпечення

ЦЗОД – центр зберігання й обробки даних

ЦОД – центр обробки даних

DLP – захист від витоків даних

IPS – системи запобігання вторгнень

CIRC – Cross Interleaved Reed Solomon Code

EAB – Embedded Array Block, блок зосередженої пам'яті

ECC – error-correcting code, код корекції помилок

FEC – метод прямої корекції помилок

LDPC – коди Галлагера

NAK – негативне підтвердження

ВСТУП

Актуальність теми. Хмарні обчислення складають основу сучасної цифрової інфраструктури, забезпечуючи масштабованість, еластичність та економічну ефективність, що перевищують показники традиційних локальних архітектур. Інтеграція штучного інтелекту (ШІ), машинного навчання (МН) та глибокого навчання (ГН) ще більше трансформує хмарні середовища, забезпечуючи прогнозну аналітику, інтелектуальну автоматизацію, оптимізацію робочого навантаження та прийняття рішень у режимі реального часу на розподілених платформах. Незважаючи на швидке впровадження, організації продовжують стикатися зі структурними проблемами, які обмежують продуктивність та сталий розвиток. Зокрема, динамічні моделі ціноутворення «оплата за використання» ускладнюють контроль витрат, коливання робочих навантажень навантажують механізми розподілу ресурсів, а дедалі складніші кіберзагрози виявляють вразливості, особливо в секторах, чутливих до даних, таких як охорона здоров'я та фінанси. Ці постійні обмеження вимагають адаптивних стратегій оптимізації на основі даних, здатних вирішувати багатовимірні та взаємозалежні операційні цілі.

Існуючі дослідження оптимізації хмарних технологій на основі штучного інтелекту залишаються фрагментарними, оскільки попередні дослідження зазвичай розглядають окремі аспекти, такі як підвищення продуктивності, виявлення аномалій або зниження витрат, без інтеграції цих перспектив в єдину аналітичну основу. Така компартменталізація обмежує систематичне розуміння колективного внеску ШІ в ефективність, безпеку та стійкість у хмарних екосистемах. Більше того, попередні огляди рідко надають порівняльний аналіз різних парадигм ШІ, включаючи машинне навчання, навчання з підкріпленням та навчання з підкріпленням, у кількох областях оптимізації. Сучасні дослідження також демонструють методологічні та контекстуальні обмеження, включаючи відсутність стандартизованих наборів даних для відтворюваної

оцінки, обмежену валідацію в гібридних та багатохмарних середовищах, недостатнє тестування масштабного розгортання та неадекватне вивчення етичних, регуляторних та конфіденційних наслідків, пов'язаних з хмарними системами на базі. Усунення цих недоліків вимагає ретельного синтезу, який об'єднує емпіричні дані, одночасно критично оцінюючи методологічну надійність та практичну застосовність.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

– Дослідження системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

– Програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Об’єктом дослідження є процес інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Предметом дослідження є методи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Методи дослідження базуються на методах хмарних обчислень, методах машинного навчання, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

– Розроблено вітчизняний продукт інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVIII науково-технічній конференції здобувачів вищої освіти «Наука-виробництву», основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Рішення для зберігання даних на основі штучного інтелекту спеціально розроблені для обробки величезних обсягів даних, що генеруються, обробляються та зберігаються робочими навантаженнями штучного інтелекту та машинного навчання (ШІ/МН). Рішення для зберігання даних на основі ШІ забезпечує швидке, ефективне та надійне зберігання та пошук даних, забезпечуючи оптимальну продуктивність програм ШІ/МН.

Традиційні рішення для зберігання даних часто не справляються з вимогами до обсягів даних, характерними для операцій штучного інтелекту/машинного навчання. Їм бракує швидкості, ємності та масштабованості для обробки швидкого потоку даних, що призводить до вузьких місць у потоці даних та знижує продуктивність програм штучного інтелекту. Такі рішення, як корпоративне об'єктне сховище Cloudian, пропонують ідеальну альтернативу з масштабованою, високопродуктивною інфраструктурою для підтримки потреб у неструктурованих даних конвеєрів штучного інтелекту.

1.2 Область застосування

Ідеальна система зберігання даних на основі штучного інтелекту використовує передові технології та архітектури для забезпечення надійних, високошвидкісних та масштабованих рішень для зберігання даних, які можуть відповідати вимогам робочих навантажень штучного інтелекту/машинного навчання. Вони забезпечують такі функції, як еластична масштабованість, швидкий доступ до даних та оптимізована обробка.

Хмарні обчислення зараз є основою, через яку працюють сучасні цифрові сервіси, забезпечуючи еластичний доступ до обчислювальних ресурсів та сховищ на вимогу. Тим не менш, складність, неоднорідність та динамічні моделі робочого навантаження хмарних середовищ призвели до значних проблем в управлінні продуктивністю. Традиційні статичні або засновані на правилах методи оптимізації, які часто використовуються, зазвичай не здатні враховувати зміни попиту в режимі реального часу. Деякі проблеми з продуктивністю, що виникають внаслідок цього, включають неефективне використання ресурсів, погіршення якості послуг та збільшення експлуатаційних витрат.

Метою цього дослідження є об'єднання алгоритмів машинного навчання (ML) для покращення загальної продуктивності та ефективності систем хмарних обчислень. Деякі з ключових показників продуктивності, що використовуються в цьому дослідженні, включають використання процесора та пам'яті, затримку, пропускну здатність, мережевий трафік та споживання енергії.

Порівняльний аналіз було проведено з використанням різних парадигм ML, включаючи навчання з учителем (наприклад, регресія, дерева рішень), навчання без учителя (наприклад, кластеризація, виявлення аномалій) та накладні витрати на підкріплення, і обговорюється, що це може означати для практичного впровадження ML у хмарних системах виробничого рівня. Загалом, це дослідження підтверджує, що інтелектуальні, автономні хмарні операції можуть бути досягнуті за допомогою стратегій оптимізації на основі машинного навчання, які значно перевершують традиційні евристики. Отже, воно відкриває шлях до створення самооптимізуючих хмарних платформ, що задовольняють вимоги дедалі складніших застосунків реального часу.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Cloudian

Cloudian пропонує S3-сумісне об'єктне сховище, розроблене для підтримки вимог до обсягів даних, пов'язаних з робочими навантаженнями штучного інтелекту в великих масштабах. Замість того, щоб змушувати організації вибирати між локальним керуванням та гнучкістю хмари, Cloudian забезпечує єдину інфраструктуру даних, яка охоплює периферійні, основні та хмарні середовища, зберігаючи при цьому суверенітет даних.

Основні характеристики включають:

– Архітектура, що відповідає стандартам S3: забезпечує універсальний доступ до даних через стандартні S3 API, що забезпечує безперешкодну інтеграцію з фреймворками та інструментами штучного інтелекту.

– Партнерство з NVIDIA: Платформа даних штучного інтелекту HyperScale® поєднує хмарне сховище з прискореними обчисленнями NVIDIA для оптимізованого навчання та логічного висновку штучного інтелекту.

– Суверенітет даних: Зберігає конфіденційні дані локально або в контрольованому середовищі, враховуючи нормативні вимоги та вимоги безпеки для ініціатив у сфері штучного інтелекту.

– Безмежна масштабованість: масштабується від терабайтів до ексабайтів зі стабільною продуктивністю, підтримуючи зростання озер даних штучного інтелекту та репозиторіїв моделей.

– Гібридна та мультихмарна гнучкість: забезпечує мобільність даних між середовищами без прив'язки до постачальника, підтримуючи розподілені робочі процеси зі штучним інтелектом.

IBM

IBM надає єдину платформу зберігання даних, призначену для вирішення завдань, пов'язаних зі штучним інтелектом, машинним навчанням та аналітикою. Її рішення, IBM Storage Scale, – це програмно-визначена система зберігання даних, яка об'єднує файлові, блочні та об'єктні служби, зберігаючи при цьому високу пропускну здатність, низьку затримку та стабільну доступність даних.



Рисунок 2.1 – IBM Storage Scale

Основні характеристики включають:

– Уніфікована платформа зберігання: підтримує зберігання файлів, блоків та об'єктів в одній системі, що зменшує складність та ізоляцію даних.

– Висока продуктивність: Забезпечує низьку затримку та високу пропускну здатність доступу для підтримки вимогливих робочих навантажень штучного інтелекту/машинного навчання у великих масштабах.

– Зберігання з урахуванням контенту: витягує семантичне значення з неструктурованих даних, допомагаючи формувати більш релевантні відповіді моделі ШІ.

– Хмарна архітектура: масштабується в локальних, хмарних та периферійних середовищах

– Співпраця та обмін даними: Забезпечує безпечний та ефективний обмін даними для підвищення продуктивності та використання ресурсів.

VAST Data

VAST Data пропонує сховище для підтримки вимог до продуктивності, масштабованості та доступності робочих навантажень штучного інтелекту. Замість того, щоб покладатися на застарілі багаторівневі архітектури, VAST впроваджує однорівневу систему на основі флеш-пам'яті, яка усуває традиційні вузькі місця у сховищі. Її дезагрегована архітектура відокремлює обчислення від сховища, що дозволяє незалежне масштабування кожного ресурсу.

Основні характеристики включають:

– Однорівнева архітектура: консолідує робочі навантаження на одному рівні флеш-пам'яті, усуваючи складність традиційного багаторівневого сховища.

– Розділені обчислення та сховище: Забезпечує незалежне масштабування обчислень та сховища, уникаючи оновлень усієї системи та підвищуючи гнучкість.

– Продуктивність на основі флеш-пам'яті: пришвидшує роботу штучного інтелекту завдяки швидкому доступу до даних з низькою затримкою.

– Лінійна масштабованість: підтримує стабільну продуктивність у міру зростання систем, підтримуючи масштабне навчання та логічний висновок ШІ.

– Висока доступність: Розроблено для безперервної роботи із захистом даних та відмовостійкістю

Pure Storage

Pure Storage пропонує єдину платформу зберігання даних, призначену для задоволення потреб у швидкості та масштабуванні робочих навантажень штучного інтелекту. Завдяки системі FlashBlade//EXA та архітектурам, сертифікованим NVIDIA, вона дозволяє організаціям пришвидшувати навчання, логічні висновки та підготовку даних штучного інтелекту на єдиній платформі.

Основні характеристики включають:

– Уніфікована платформа даних ШІ: підтримує конвеєр ШІ, включаючи підготовку, навчання та логічний висновок, в одній системі.

– Висока пропускна здатність у великих масштабах: забезпечує пропускну здатність понад 10 ТБ/с в межах одного простору імен масштабу

екзабайт для швидкого навчання та точного налаштування моделі.

– Постійна продуктивність: Забезпечує 99,9999% безвідмовної роботи, оновлення без переривання роботи та постійну доступність для робочих навантажень штучного інтелекту.

– Архітектура, сертифікована NVIDIA: забезпечує оптимізовану продуктивність та спрощене розгортання завдяки перевіреним конфігураціям, готовим до роботи з графічними процесорами.

– Дезагрегована система FlashBlade//EXA: забезпечує масштабовану, паралелізовану архітектуру з продуктивністю метаданих, що підходить для випадків використання штучного інтелекту та високопродуктивних обчислень.

Huawei Technologies

Платформа зберігання даних OceanStor AI від Huawei спрямована на підтримку комплексних робочих процесів штучного інтелекту, задовольняючи вимоги до продуктивності та масштабованості великих мультимодальних моделей. Її розподілені системи зберігання файлів, такі як OceanStor A800 та A600, розроблені для подолання ізольованих середовищ даних, пришвидшення завантаження навчальних наборів та підвищення ефективності логічного висновку в різних галузевих застосуваннях.

Основні характеристики включають:

– Надзвичайна продуктивність: Забезпечує пропускну здатність у сотні ТБ/с та понад 100 мільйонів IOPS, прискорюючи генеративні процеси штучного інтелекту.

– Підтримка нових типів даних ШІ: оптимізовано для тензорів, векторів та кешу KV для зменшення затримки логічного висновку та галюцинацій у великих моделях.

– Ефективне управління метаданими: Забезпечує глобальну видимість та переміщення даних із десятикратним покращенням мобільності даних.

– Масштабована архітектура: підтримує кластери екзабайтного масштабу та обчислення майже на рівні сховища з інтеграцією графічних процесорів, процесорів цифрових процесорів та нейронних процесорів.

– Висока надійність: Забезпечує безвідмовну роботу системи на рівні 99,9999% та включає вбудовану функцію виявлення програм-вимагачів з точністю 99,99%.

Висновок

Зі прискоренням впровадження штучного інтелекту роль спеціалізованих постачальників сховищ стає критично важливою для підтримки потреб сучасних робочих навантажень у продуктивності, масштабованості та надійності. Ці постачальники пропонують системи, розроблені з урахуванням стрімкого зростання обсягу даних, вимог паралельної обробки та складності гібридних і розподілених середовищ. Зосереджуючись на переміщенні даних, швидкості доступу, управлінні життєвим циклом та інтеграції з інструментальними ланцюжками штучного інтелекту, вони дозволяють організаціям підтримувати ефективні та стійкі канали обробки даних від експериментів до виробництва.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації програмних модулів системи інтелектуальної оптимізації продуктивності хмарних сховищ даних було використано мову програмування Visual C#. Вибір цього інструментарію зумовлений високими вимогами до надійності, безпеки, продуктивності та гнучкості проєктування складних розподілених сервісів, що виконуються в середовищі .NET. Мова Visual C# надає потужну систему типізації, розвинені об'єктно-орієнтовані механізми та багаті вбудовані бібліотеки, що суттєво спрощує створення багатопотокових рішень для аналізу телеметрії сховищ, обробки метрик продуктивності та виконання алгоритмів машинного навчання.

Синтаксис Visual C# є суворим, виразним та зручним для розробки високонавантажених компонентів. Водночас він пропонує низку інноваційних конструкцій, які є критично важливими для реалізації логіки інтелектуального управління сховищами:

– LINQ (Language-Integrated Query): забезпечує зручні засоби суворо типізованих запитів до логів, метрик продуктивності хмарних сховищ та внутрішніх структур даних моделей оптимізації.

– Делегати та події: дозволяють організувати асинхронну обробку подій у реальному часі (наприклад, під час динамічного реагування на зміну навантаження або появи вузьких місць у хмарному середовищі).

– Об’єктно-орієнтована парадигма: підтримка інкапсуляції, спадкування й поліморфізму гарантує чітку модульність програмного забезпечення (розподіл на модулі збору метрик, аналізу за допомогою машинного навчання, прийняття рішень та оптимізації).

– Механізми Interop та безпека: за необхідності глибокої оптимізації критичних ділянок коду або інтеграції з нативними бібліотеками прогнозування Visual C# дозволяє застосовувати взаємодію з зовнішніми DLL-компонентами та використовувати блоки «небезпечного» коду (unsafe) із прямим керуванням пам'яттю там, де це вимагається для максимальної швидкодії.

Застосування мови Visual C# та платформи .NET забезпечує створення надійного, продуктивного, гнучкого та легко масштабованого програмного ядра для системи інтелектуальної оптимізації продуктивності хмарних сховищ даних.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних

і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Компоненти архітектури зберігання даних штучного інтелекту

ШІ-сховище часто охоплює набір технологій та компонентів, які працюють разом, щоб забезпечити ефективне та результативне рішення для зберігання даних на основі штучного інтелекту та машинного навчання. Давайте розглянемо деякі компоненти, які дозволяють рішенням для зберігання підтримувати дані на основі штучного інтелекту.

Флеш-накопичувач NVMe

NVMe (незалежна пам'ять Express) – це тип пристрою зберігання даних, який використовує флеш-пам'ять. Він розроблений для використання низької затримки та високошвидкісних можливостей твердотільних накопичувачів (SSD), що робить його придатним для робочих навантажень штучного інтелекту та машинного навчання. Флеш-накопичувачі NVMe пропонують кілька переваг порівняно з традиційними пристроями зберігання даних. Вони забезпечують швидший доступ до даних, зменшену затримку та вищу пропускну здатність. Вони також підтримують паралельну обробку даних, що дозволяє одночасно обробляти кілька запитів на дані. Це робить їх відповідним вибором для сховищ зі штучним інтелектом, де швидкий, ефективний та надійний доступ до даних є критично важливим.

Твердотільні накопичувачі

Твердотільні накопичувачі (SSD) – це тип пристроїв зберігання даних, які використовують флеш-пам’ять. На відміну від традиційних жорстких дисків (HDD), SSD не мають рухомих частин, що робить їх швидшими, надійнішими та менш схильними до фізичних пошкоджень. Твердотільні накопичувачі забезпечують швидкість і надійність, необхідні для обробки великих обсягів даних та вимог до інтенсивного доступу до даних у застосунках штучного

інтелекту та машинного навчання. Їх можна легко масштабувати для розміщення зростаючих обсягів даних, що робить їх практичним і гнучким рішенням для зберігання даних, пов'язаних зі штучним інтелектом.

Розподілені файлові системи та сховище об'єктів

Розподілені файлові системи є вирішальними для зберігання даних штучного інтелекту, оскільки вони дозволяють зберігати дані на кількох машинах, забезпечуючи високу доступність та відмовостійкість. Вони розроблені для горизонтального масштабування, що дозволяє їм керувати величезними обсягами даних, що генеруються робочими навантаженнями штучного інтелекту та машинного навчання. Ці системи підтримують високопродуктивні операції та можуть обробляти одночасний доступ до даних кількома програмами, що робить їх ідеальними для розподілених обчислювальних середовищ. Об'єктне сховище обробляє дані як окремі одиниці або об'єкти, а не як файли в каталозі. Кожен об'єкт містить самі дані, змінну кількість метаданих та глобально унікальний ідентифікатор. Об'єктне сховище є високомасштабованим і особливо добре підходить для зберігання неструктурованих даних, таких як документи, зображення та відео, які є поширеними в застосунках штучного інтелекту. Воно надає можливість зберігати величезні обсяги інформації у високоорганізований та доступний спосіб.

Озера даних

Озера даних – це централізовані сховища, призначені для зберігання, обробки та захисту великих обсягів структурованих та неструктурованих даних. Вони можуть обробляти дані з різних джерел та форматів, що робить їх ідеальними для застосувань штучного інтелекту та машинного навчання, які потребують широкого спектру типів та структур даних. Озера даних підтримують зберігання необроблених даних, забезпечуючи гнучку обробку, аналітику та дослідження даних. Ця можливість дозволяє спеціалістам з обробки даних та алгоритмам штучного інтелекту отримувати доступ до даних та аналізувати їх без обмежень традиційних систем управління даними.

Переваги систем зберігання даних, розроблених для робочих навантажень зі штучним інтелектом

Масштабованість

Рішення для зберігання даних на основі штучного інтелекту повинні швидко адаптуватися до вимог масштабування даних робочих навантажень штучного інтелекту/машинного навчання. Масштабованість сховищ на основі штучного інтелекту дозволяє розширювати ємність та продуктивність сховища без порушення поточних операцій. Це означає, що зі зростанням обсягів даних або кількості проектів штучного інтелекту система зберігання може ефективно враховувати це зростання. Масштабовані системи зберігання забезпечують оптимальне використання ресурсів, запобігаючи їх надмірному виділенню та недостатньому використанню.

Продуктивність

Продуктивність є ключовою перевагою систем зберігання даних, розроблених на основі штучного інтелекту, оскільки вони пропонують високу пропускну здатність і низьку затримку, необхідні для потреб робочих навантажень штучного інтелекту та машинного навчання. Ці системи оптимізовані для обробки великих наборів даних і підтримки ресурсомістких обчислювальних завдань, необхідних для навчання та запуску моделей штучного інтелекту. Висока продуктивність систем зберігання даних зі штучним інтелектом означає швидший доступ до даних і їх обробку, що призводить до швидшого отримання аналітичних даних і прийняття рішень. Це особливо важливо в додатках, що потребують аналітики в режимі реального часу, або тих, що обробляють великі обсяги даних.

Паралелізм

Рішення для зберігання даних на основі штучного інтелекту часто підтримують паралелізм, що дозволяє одночасно виконувати кілька процесів. Це життєво важливо для обробки одночасних запитів на дані, характерних для робочих навантажень штучного інтелекту та машинного навчання. Паралелізм підвищує ефективність обробки та аналізу даних, дозволяючи швидше навчати

моделі та виводити дані. Підтримуючи кілька потоків даних паралельно, системи зберігання даних на основі штучного інтелекту можуть значно скоротити час, необхідний для виконання завдань з інтенсивною обробкою даних.

Сумісність з фреймворками штучного інтелекту

Системи зберігання даних на основі штучного інтелекту розроблені для сумісності з широким спектром фреймворків та інструментів штучного інтелекту. Ця сумісність забезпечує безперешкодну інтеграцію з популярними середовищами розробки, бібліотеками та фреймворками штучного інтелекту та машинного навчання, що сприяє простішій та ефективнішій розробці моделей штучного інтелекту. Рішення для зберігання даних, сумісні з фреймворками штучного інтелекту, допомагають зменшити складність управління даними для проектів штучного інтелекту, дозволяючи спеціалістам з обробки даних та розробникам зосередитися на розробці та інноваціях моделей, а не на проблемах управління даними.

Ключові міркування щодо вибору рішень для зберігання даних на основі штучного інтелекту

Ось деякі основні питання, які слід враховувати під час порівняння рішень для зберігання даних на основі штучного інтелекту.

Неструктуровані дані

Застосунки штучного інтелекту часто потребують обробки величезних обсягів неструктурованих даних, таких як зображення, відео та текст. Відповідне рішення для зберігання даних на основі штучного інтелекту повинно ефективно зберігати, керувати та отримувати неструктуровані дані. Воно повинно пропонувати надійні механізми для обробки змінності та складності даних, забезпечуючи швидкий доступ до неструктурованих даних та їх обробку для завдань штучного інтелекту.

RESTful API

RESTful API мають вирішальне значення для інтеграції рішень для зберігання даних на основі штучного інтелекту з іншими системами та

програмами. Вони забезпечують легкий доступ та маніпулювання збереженими даними через Інтернет, сприяючи безперебійному потоку даних між різними компонентами робочих процесів штучного інтелекту та машинного навчання. Вибираючи рішення для зберігання даних на основі штучного інтелекту, шукайте таке, яке надає комплексні та добре документовані RESTful API для забезпечення сумісності та гнучкості інтеграції.

Шифрування

Безпека даних є першочерговим завданням, особливо для чутливих застосувань штучного інтелекту. Шифрування гарантує, що дані, що зберігаються в рішеннях для зберігання даних штучного інтелекту, захищені від несанкціонованого доступу та порушень. Розгляньте рішення для зберігання, які пропонують надійні можливості шифрування як у стані спокою, так і під час передачі, захищаючи цілісність та конфіденційність даних.

Хмарні рішення

Хмарні рішення для зберігання даних на основі штучного інтелекту пропонують гнучкість, масштабованість та економічну ефективність, що робить їх ідеальними для організацій, які використовують хмарні обчислення для робочих навантажень на основі штучного інтелекту. Ці рішення розроблені для процвітання в хмарному середовищі, забезпечуючи безперешкодну інтеграцію з хмарними сервісами та інфраструктурою. Вибираючи рішення для зберігання даних на основі штучного інтелекту, враховуйте його сумісність з хмарними архітектурами, щоб переконатися, що воно може використовувати весь потенціал хмарних обчислень.

Незмінність

Незмінність гарантує, що після збереження дані не можуть бути змінені або видалені, захищаючи від підробки та втрати даних. Це особливо важливо для дотримання нормативних вимог та для підтримки цілісності даних, що використовуються в застосунках штучного інтелекту. Шукайте рішення для зберігання даних зі штучним інтелектом, які підтримують незмінність, щоб забезпечити надійність та безпеку даних.

Кодування стиранням та RAID

Стійкість та доступність даних є критично важливими для застосувань штучного інтелекту. Кодування зі стиранням та RAID (надмірний масив незалежних дисків) – це технології, що використовуються для захисту даних від збоїв обладнання. Вибираючи рішення для зберігання даних на основі штучного інтелекту, враховуйте компроміси між кодуванням зі стиранням та RAID з точки зору продуктивності, масштабованості та можливостей захисту даних, щоб забезпечити високу доступність та надійність даних.

Можливість безперешкодного переміщення даних між різними середовищами (наприклад, локальними, хмарними, периферійними) без значної переналаштування є критично важливою для сучасних застосувань штучного інтелекту. Така портативність дозволяє організаціям використовувати кілька обчислювальних середовищ, оптимізуючи продуктивність і витрати. Виберіть рішення для зберігання даних на основі штучного інтелекту, яке пропонує високу портативність, щоб забезпечити гнучкість у тому, де і як дані зберігаються та обробляються.

3.2 Розробка структурної схеми

Робочі навантаження штучного інтелекту та машинного навчання генерують, обробляють та аналізують величезні обсяги даних з високою швидкістю, виводячи звичайні системи зберігання даних за межі їхніх можливостей. Постачальники сховищ на базі штучного інтелекту реагують на це продуктами, оптимізованими для характеристик, унікальних для штучного інтелекту, таких як висока паралельність, нелінійні шаблони доступу та потреба у швидкому отриманні та виклику даних.

Серед постачальників систем штучного інтелекту для зберігання даних є Cloudian, VAST Data, IBM та Pure Storage. Зазвичай ці постачальники інтегрують апаратні та програмні функції, адаптовані для середовищ штучного інтелекту. Вони часто пропонують паралельні файлові системи, високопродуктивне

сховище об'єктів, багаторівневе розподілення даних та пряму інтеграцію з популярними фреймворками машинного навчання.

Мета полягає в мінімізації затримки та вузьких місць, задоволенні вимог до обробки кластерів графічних процесорів та спрощенні робочих процесів, від підготовки даних до розгортання моделі. Зосереджуючись на конкретних проблемах зберігання даних, пов'язаних зі штучним інтелектом, ці постачальники допомагають організаціям уникнути уповільнень навчання, неефективних конвеєрів даних або неочікуваних проблем із масштабуванням.

Варіанти використання та застосування штучного інтелекту для зберігання даних

Великомасштабні навчальні конвеєри

Навчання великих моделей штучного інтелекту, таких як ті, що використовуються для комп'ютерного зору або обробки мови, вимагає швидкого доступу до величезних наборів даних. Протягом навчального конвеєра дані завантажуються, перетворюються та багаторазово звертаються до них високопродуктивними обчислювальними блоками, такими як графічні процесори (GPU) або процесори TPU. Традиційні архітектури сховищ даних мають труднощі з забезпеченням необхідної пропускної здатності та кількості операцій вводу/виводу за секунду (IOPS), що призводить до недостатнього використання обладнання та збільшення часу навчання.

Рішення для зберігання даних на основі штучного інтелекту вирішують ці проблеми з продуктивністю, використовуючи паралельний доступ до даних, кешування та оптимізоване сховище файлів або об'єктів, забезпечуючи повне використання навчального обладнання. У розподілених умовах, де навчання залежить від кількох вузлів, що працюють одночасно, стабільна доступність даних та продуктивність є критично важливими. Постачальники сховищ даних на основі штучного інтелекту часто розробляють архітектури сховищ, які забезпечують надійну пропускну здатність у великих масштабах, підтримуючи одночасний доступ тисяч працівників без зниження продуктивності.

Висновок у реальному часі

Такі програми штучного інтелекту, як автономні транспортні засоби, системи виявлення шахрайства та механізми рекомендацій, покладаються на логічний висновок у реальному часі. Ці сценарії вимагають від системи зберігання даних надавати вхідні дані з наднизькою затримкою та підтримувати високу доступність, навіть коли моделі доступу до даних швидко змінюються. Постачальники сховищ на основі штучного інтелекту пропонують високопродуктивні рішення, здатні підтримувати ці суворі вимоги, часто використовуючи NVMe, сховище в пам'яті або архітектури на основі периферії для мінімального часу відгуку.

Інфраструктура також має бути достатньо гнучкою для масштабування на вимогу. Під час піків трафіку або раптового збільшення попиту система зберігання даних повинна обробляти велику кількість невеликих, швидких транзакцій без перевантажень. Такі функції, як попередня вибірка даних, інтелектуальне кешування та балансування навантаження, зазвичай впроваджуються для оптимізації робочих процесів виведення.

Озеро корпоративних даних та аналітика для штучного інтелекту

Корпоративні середовища накопичують величезні обсяги структурованих і неструктурованих даних, які часто зберігаються в централізованих озерах даних. Для ініціатив у сфері штучного інтелекту ця інформація повинна бути ефективно отримана, каталогізована та доступна для аналітики, навчання моделей та управління. Постачальники сховищ на основі штучного інтелекту підтримують ці вимоги, надаючи масштабовані, багаті метаданими платформи зберігання, розроблені для швидкого індексування та пошуку даних.

Ці рішення зазвичай включають надійні функції керування даними: інтегрований пошук, багаторівневе розподілення на основі політик та автоматизоване керування життєвим циклом даних. Завдяки автоматичному індексуванню та класифікації даних вони допомагають командам швидко знаходити відповідні набори даних, зменшувати ручні зусилля на обробку даних та забезпечувати відстеження для цілей відповідності вимогам та аудиту.

Архівування, контрольні точки та повторне використання наборів даних штучного інтелекту

Тривалі проекти штучного інтелекту генерують цінні проміжні набори даних, контрольні точки моделей та необроблені архіви, які необхідно зберігати для регуляторних, відтворюваних або майбутніх дослідницьких цілей. Постачальники сховищ на базі штучного інтелекту пропонують багаторівневі, економічно ефективні рішення для зберігання даних, які переносять застарілі або рідко використовувані дані в холодне сховище без шкоди для доступності. Політики архівування, керування версіями та контрольних точок інтегровані, підтримуючи робочі процеси резервного копіювання та відновлення.

Можливість швидкого відновлення та повторного використання архівованих наборів даних дозволяє командам ефективно перенавчати або налаштовувати моделі за допомогою історичних даних. Системи зберігання даних часто надають механізми контролю версій, дедуплікації даних та управління метаданими, що забезпечує співпрацю між командами та проектами.

Ключові вимоги до постачальників сховищ на базі штучного інтелекту

Продуктивність

Постачальники штучного інтелекту (ШІ) сховищ повинні забезпечувати стабільно високу пропускну здатність і низьку затримку для підтримки як навчання, так і логічного висновку в масштабах. Ці характеристики продуктивності є критично важливими для запобігання простою графічних процесорів або спеціалізованих прискорювачів через брак даних. Паралельні шаблони доступу, оптимізоване кешування, інтеграція NVMe та локальність даних – це деякі з технічних стратегій, що використовуються для скорочення часу очікування та підвищення ефективного використання обчислювальних ресурсів. Окрім чистої швидкості, життєво важливою є також передбачувана продуктивність у періоди високої конкуренції або масштабування. Навчальні завдання та запити на виведення повинні обслуговуватися з мінімальним зниженням часу відгуку або без нього, незалежно від розміру набору даних чи паралельності доступу.

Масштабованість

Масштабованість є фундаментальною, оскільки робочі навантаження на основі штучного інтелекту можуть швидко зростати від терабайтів до петабайтів даних на різних етапах проекту. Постачальники сховищ даних на основі штучного інтелекту розробляють свої рішення для горизонтального масштабування, що дозволяє організаціям розширювати потужність і продуктивність, просто додаючи більше вузлів зберігання. Це гарантує, що зростаючі обчислювальні кластери та набори даних можуть бути підтримані без архітектурних переробок або руйнівних міграцій.

Більше того, масштабованість поширюється на обробку мільйонів або мільярдів окремих файлів та об'єктів без уповільнення індексації чи падіння продуктивності. Рішення часто забезпечують безперешкодне розширення простору імен, розподілені метадані та об'єднані можливості управління.

Управління даними та багаторівневе розміщення

Ефективне управління даними є ключовим фактором для максимізації рентабельності інвестицій у сховище даних у середовищах штучного інтелекту. Постачальники сховищ даних на основі штучного інтелекту впроваджують детальну автоматизацію на основі політик для міграції даних між рівнями, починаючи від високошвидкісного основного сховища для поточних робочих навантажень і закінчуючи економічним, довгостроковим холодним сховищем для архівів. Таке розподілення на рівні забезпечує оптимальне використання ресурсів, балансуючи продуктивність із витратами на зберігання без ручного втручання.

Управління метаданими відіграє вирішальну роль у цьому контексті. Рішення для зберігання даних на основі штучного інтелекту все частіше оснащені надійними сервісами метаданих з можливістю пошуку, що дозволяє командам знаходити, категоризувати та відстежувати набори даних на різних рівнях. Автоматизовані політики життєвого циклу та інтелектуальне розміщення даних допомагають спростити дотримання вимог, зберігання та гігієну даних.

Підтримка гібридних/хмарних/периферних мереж

Робочі навантаження штучного інтелекту часто охоплюють локальні центри обробки даних, публічні хмари та периферійні пристрої, що вимагає інфраструктур зберігання даних, здатних об'єднувати ці середовища. Провідні постачальники сховищ даних на основі штучного інтелекту пропонують гібридні архітектури з безперешкодною мобільністю даних, що дозволяє отримувати, зберігати та обробляти дані там, де це найефективніше або економічно вигідно. Ця гнучкість є життєво важливою для таких випадків використання, як федеративне навчання, аналітика Інтернету речей та глобальні програми штучного інтелекту.

Підтримка периферійних мереж стає дедалі важливішою для чутливих до затримки або розподілених розгортань штучного інтелекту, де дані мають оброблятися поблизу місця їх генерації. Рішення для зберігання даних на периферії, які безпосередньо підтримують розгортання на периферії, з такими можливостями, як автономна робота та синхронізація з хмарою, є критично важливими факторами для периферійного штучного інтелекту.

Інтеграція штучного інтелекту/машинного навчання та спеціалізовані функції

Постачальники сховищ на основі штучного інтелекту все частіше інтегрують функції, що враховують машинне навчання, безпосередньо у свої платформи. Це може включати вбудовану підтримку популярних фреймворків штучного інтелекту/машинного навчання, автоматизовану оркестрацію конвеєрів та такі функції, як попереднє вибірка даних або керування версіями наборів даних, адаптовані для ітеративного навчання та експериментування моделей. Ці інтеграції зменшують труднощі у побудові комплексних робочих процесів зі штучним інтелектом та допомагають організаціям ефективніше переміщувати моделі з дослідницької у виробничу фазу.

Спеціалізовані функції можуть також включати інструменти для маркування, анотування або перетворення даних, ключових дій перед початком навчання моделі. Деякі постачальники надають API для прямого зворотного

зв'язку з моделлю, детального контролю доступу або перехоплювачів, які запускають дії подальшої обробки.

Надійність, безпека та управління даними

Дані є критично важливим активом у розробках штучного інтелекту; втрата або пошкодження можуть відкласти проекти на місяці. Постачальники сховищ даних на основі штучного інтелекту надають пріоритет надійності завдяки таким функціям, як кодування зі стиранням, реплікація, наскрізне підсумовування та послідовне резервне копіювання, гарантуючи, що дані залишатимуться доступними та неушкодженими навіть у разі збоїв обладнання або перебоїв на сайті. Автоматизовані плани аварійного відновлення та відновлення після збоїв зазвичай включаються для максимізації часу безвідмовної роботи та стійкості.

Безпека та управління даними однаково важливі, оскільки можуть бути задіяні конфіденційні або регульовані дані. Постачальники підтримують детальний контроль доступу, шифрування під час зберігання та передачі, ведення журналу аудиту та функції відповідності для дотримання правових та нормативних стандартів. Комплексне управління гарантує, що лише авторизовані користувачі та програми можуть взаємодіяти з даними.

Розроблене програмне забезпечення виділяється як рішення для зберігання даних, спеціально розроблене для систем штучного інтелекту, пропонуючи масштабоване, економічно ефективне та стійке сховище об'єктів, яке відповідає унікальним вимогам робочих навантажень штучного інтелекту та машинного навчання. Воно забезпечує міцну основу як для потокових, так і для пакетних конвеєрів штучного інтелекту, забезпечуючи ефективне управління та обробку великих обсягів неструктурованих даних. Завдяки опціям георозподілу організації можуть розгортати системи розробленого програмного забезпечення за потреби, вибираючи між конфігураціями на основі флеш-пам'яті та жорстких дисків, щоб відповідати вимогам до продуктивності їхнього конкретного робочого навантаження.

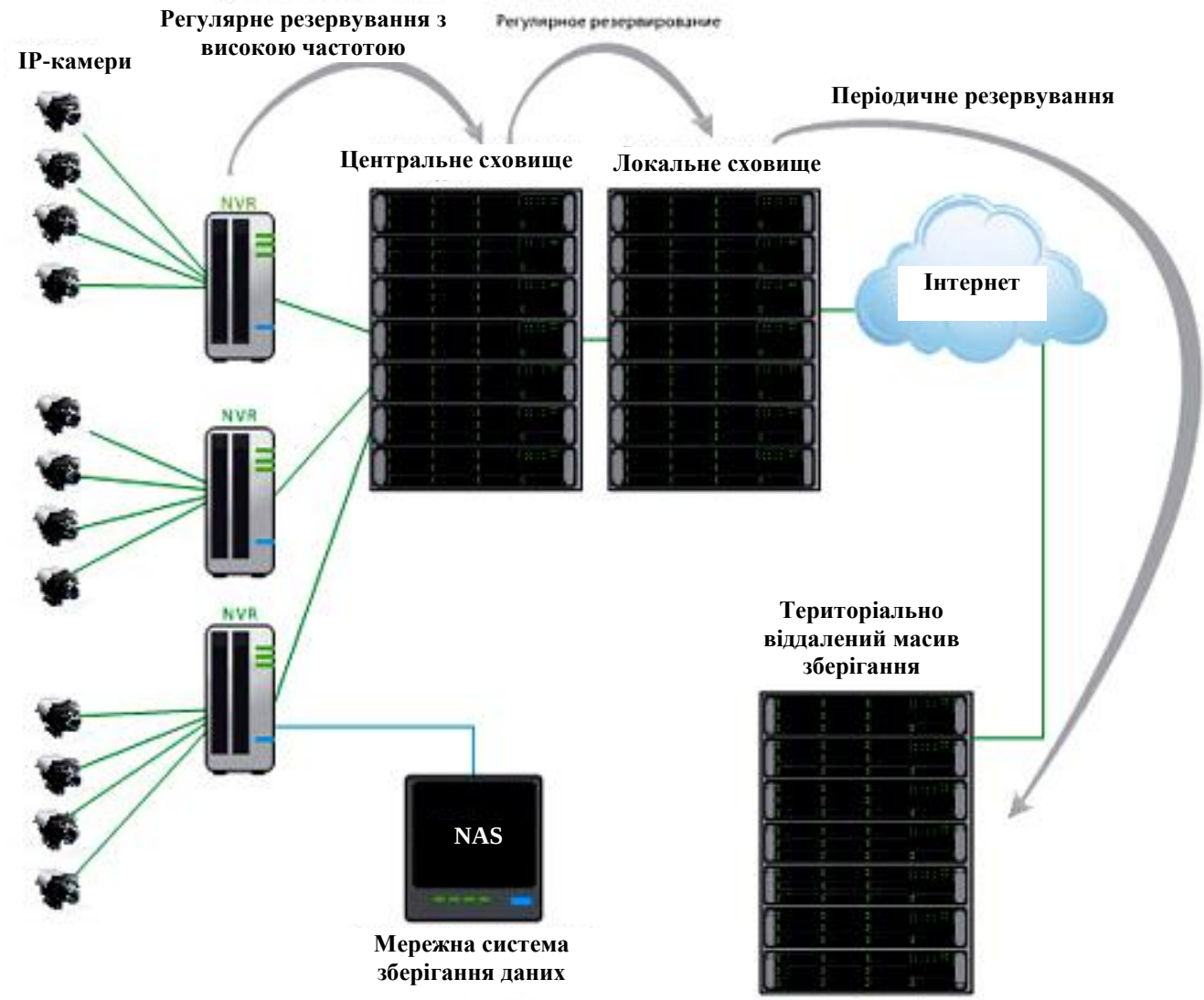

Рисунок 3.1 – Структурна схема системи

Сумісність платформи з популярними фреймворками машинного навчання, такими як TensorFlow, PyTorch та Spark ML, спрощує процес оптимізації робочого навантаження штучного інтелекту. Ці фреймворки оптимізовані для паралельного навчання безпосередньо зі сховища об'єктів, що підвищує продуктивність та сумісність.

Розроблене програмне забезпечення спрощує керування даними завдяки таким функціям, як багаті метадані об'єктів, керування версіями та теги, а також сприяє співпраці завдяки можливостям багатокористувацького користування та HyperSearch, пришвидшуючи робочі процеси зі штучним інтелектом. Крім того,

підтримка NVIDIA Triton та сумісність з кількома типами сховищ – сховищами ознак, векторними базами даних та сховищами моделей – дозволяє організаціям ефективно керувати, шукати та використовувати свої дані, забезпечуючи надійну та ефективну інфраструктуру даних штучного інтелекту.

3.3 Розробка функціональної схеми

Забезпечення надійності та відказостійкості хмарних сховищ на основі кодів Ріда-Соломона

Функціональну схему системи підвищення надійності зберігання даних у хмарному середовищі наведено на рисунку 3.2. Розроблений програмний комплекс включає такі базові компоненти:

– Сервери мережних систем зберігання: фізичні або віртуальні вузли, що виступають носіями інформаційних масивів.

– Кодер Ріда-Соломона: модуль, який здійснює процедуру перешкодостійкого кодування під час запису даних на сервери сховища. Ураховуючи природу кодів із надмірністю, загальний обсяг даних разом із перевірочними символами узгоджується з місткістю дискового простору.

– Декодер Ріда-Соломона: модуль, відповідальний за верифікацію та відновлення даних під час їх зчитування з носіїв.

У складі кодера виділяються такі функціональні елементи: вхідні дані для збереження у хмарному сховищі, породжувальний поліном та відмовостійки коди.

Коди Ріда-Соломона (Reed-Solomon codes) є недвійковими циклічними кодами, розробленими І. Рідом і Г. Соломоном у 1960 році, елементами кодового вектора яких є не окремі біти, а цілі блоки (октети / байти). Вони є важливим частковим випадком БЧХ-кодів і широко застосовуються в сучасних розподілених дата-центрах та хмарних інфраструктурах для створення архівів відновлення, завадостійкого кодування й захисту від масових пакетних збоїв.



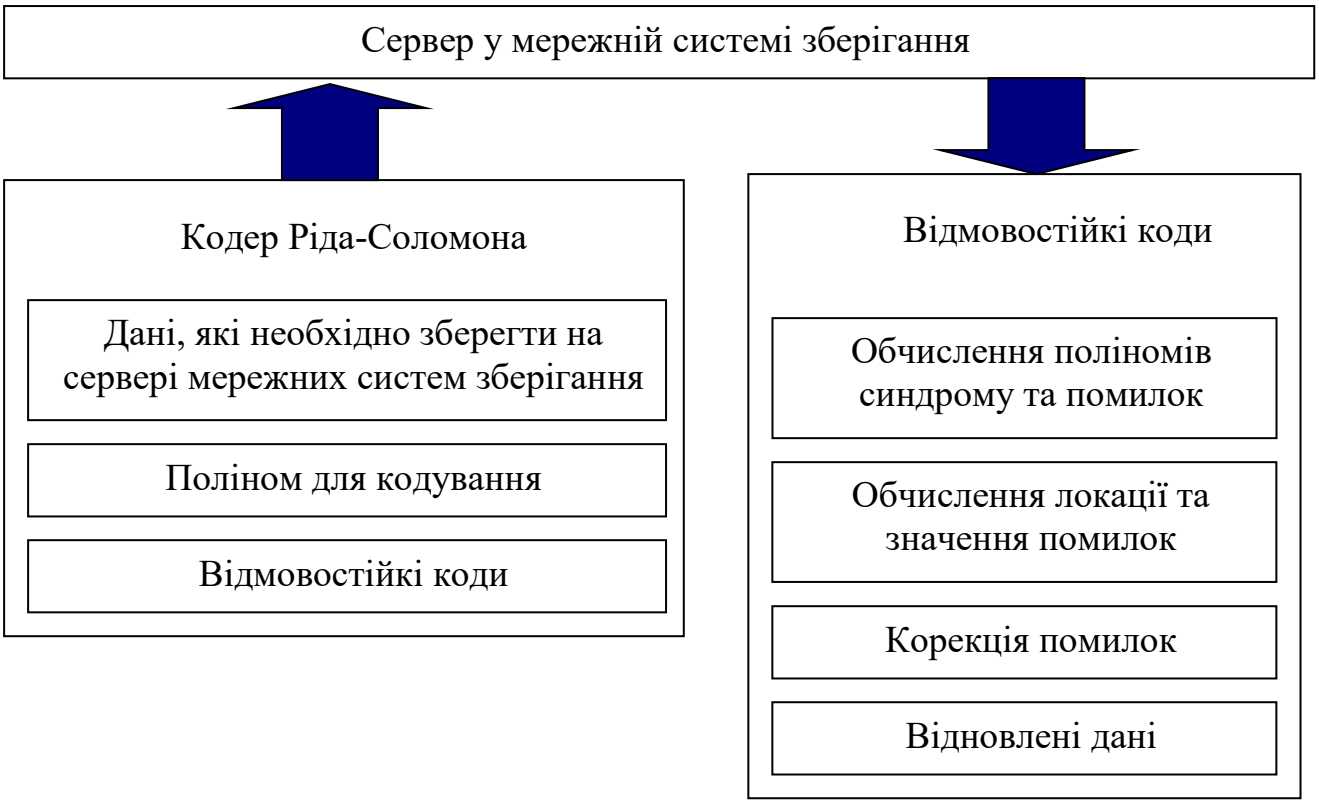

Рисунок 3.2 – Функціональна схема системи

Алгоритми прямої корекції помилок (Forward Error Correction, FEC) на базі кодів Ріда-Соломона дозволяють ефективно відновлювати втрачену інформацію навіть у разі пошкодження кількох секторів носія чи виникнення

Практична реалізація кодування може виконуватися двома методами:

– Несистематичне кодування: інформаційне слово множиться на незвідний поліном у полі Галуа; отриманий результат повністю відрізняється від вихідних даних, що потребує обов'язкового декодування для доступу до інформації.

– Систематичне кодування: до інформаційного блоку довжиною k

дописуються 2t перевірочних символів. У разі відсутності помилок вихідні дані зчитуються без додаткових обчислювальних витрат, що є оптимальним для архітектури хмарних сховищ із високою інтенсивністю операцій читання.

Процес декодування за авторегресивним спектральним методом послідовно виконує такі етапи:

1. Обчислення синдрому помилки: ділення кодового слова на породжувальний поліном; наявність остачі свідчить про наявність помилок.

2. Побудова полінома помилок: формування відповідності між помилкою та її позицією з використанням алгоритму Берлекемпа-Месси або алгоритму Евкліда.

3. Знаходження коренів полінома: визначення локації перекручених символів за допомогою процедури Ченя (Chien search).

4. Корекція помилок: визначення характеру пошкоджень та їх виправлення за алгоритмом Форни (Forney), накладення маски за допомогою операції XOR і остаточне відновлення інформаційного слова.

Застосування оптимізованих апаратно-програмних рішень на базі кодів Ріда-Соломона у складі хмарних інфраструктур забезпечує високий рівень відказостійкості та мінімізує ризики втрати критично важливих даних за умов інтенсивного навантаження.

3.4 Розробка діаграми процесів

Діаграму взаємодії процесів розробленої у межах магістерської роботи системи наведено на рисунку 3.3. Її детальний аналіз дозволяє простежити логіку взаємодії внутрішніх компонентів та маршрутизацію інформаційних потоків у межах платформи інтелектуальної оптимізації хмарних сховищ. У процесі проєктування застосовується модель графічного представлення потоків даних (Data Flow Diagram, DFD), яка відображає рух, трансформацію та збереження інформації (зокрема, метрик продуктивності хмарного середовища, логів навантаження та прогнозних моделей машинного навчання).

Діаграми взаємодії процесів є базовим інструментом структурного проєктування для візуалізації обробки даних. Традиційно розробка розпочинається з побудови контекстної діаграми потоків даних, яка ілюструє загальну взаємодію системи оптимізації із зовнішнім середовищем (хмарною інфраструктурою, джерелами метрик та адміністратором сховища). У подальшому ця модель деталізується шляхом поетапної декомпозиції процесів і потоків даних з метою повної та всебічної демонстрації архітектури розроблюваного програмного комплексу.

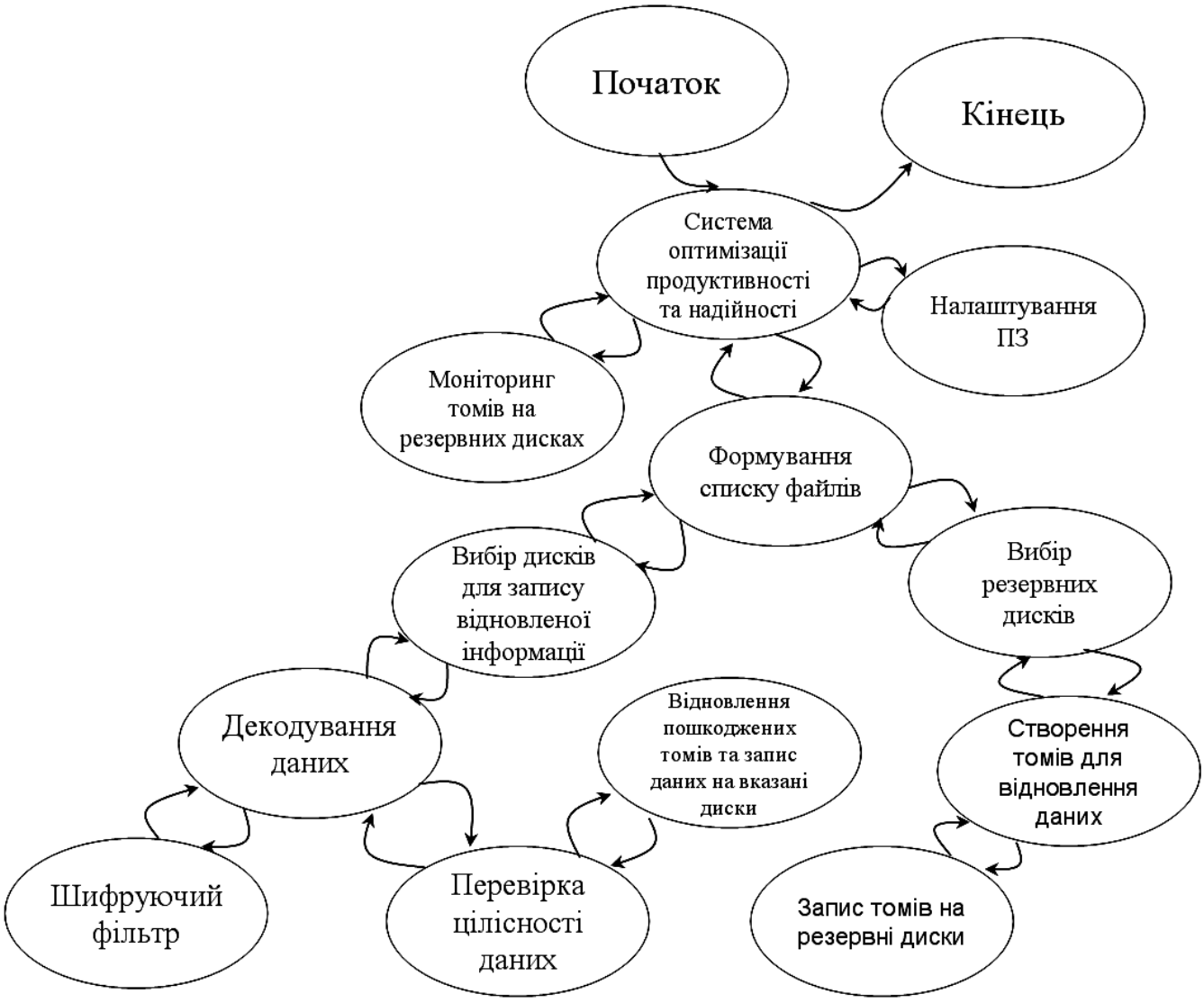

Рисунок 3.3 – Діаграма взаємодії процесів

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Алгоритмічне забезпечення системи адаптивного регулювання складності мультимедійного аркадного комплексу призначене для автоматизованого визначення рівня підготовленості гравця, аналізу його поточної результативності та динамічної зміни параметрів ігрового процесу. Основним завданням є підтримання такого рівня складності, який відповідає можливостям користувача та забезпечує поступове ускладнення ігрових завдань без надмірного збільшення кількості невдалих спроб. Для цього передбачається використання алгоритмів машинного навчання, які встановлюють залежності між характеристиками поведінки гравця, параметрами ігрового середовища та результатами проходження окремих етапів. На основі отриманих прогнозів система визначає необхідність зміни інтенсивності перешкод, швидкості руху об'єктів, часових обмежень та інших характеристик складності.

Загальну послідовність функціонування програмного комплексу відображено на рисунку 4.1. Основний алгоритм починається з ініціалізації ігрового середовища та створення користувацького сеансу. Під час проходження рівня система реєструє результати дій гравця, визначає показники його результативності та формує поведінковий профіль. Після накопичення необхідної кількості спостережень виконується аналіз поточного стану за допомогою моделі машинного навчання. На основі прогнозованої успішності проходження наступного етапу визначається доцільний рівень складності. Обрана конфігурація передається ігровому модулю та застосовується в дозволений момент. Подальші результати використовуються для перевірки правильності прийнятого рішення.

Для організації роботи системи доцільно передбачити архітектуру з окремими компонентами керування ігровим процесом, реєстрації подій, формування поведінкових характеристик і машинного навчання. Ігровий модуль відповідає за відображення графічних об'єктів, оброблення керувальних дій та виконання правил аркадної гри. Компонент телеметрії накопичує інформацію про результати проходження. Аналітична підсистема формує вектори характеристик і визначає поточний стан гравця. Модуль адаптації використовує прогноз моделі для вибору наступної конфігурації складності. Окремий компонент зберігання забезпечує доступ до параметрів моделей, профілів користувачів та історії проходження ігрових етапів.

Особливістю мультимедійного аркадного комплексу є взаємодія декількох складових ігрового середовища. Складність може залежати від кількості об'єктів на екрані, швидкості їхнього переміщення, тривалості доступного часу, частоти появи перешкод і правил нарахування результатів. Додатково використовуються графічні ефекти, звукові повідомлення та інші мультимедійні засоби зворотного зв'язку. При цьому параметри візуального й звукового оформлення необхідно відокремлювати від безпосередніх характеристик складності. Наприклад, зміна швидкості появи цілей може впливати на вимоги до реакції гравця, тоді як зміна музичного супроводу не обов'язково повинна змінювати умови виконання завдання.

Алгоритм роботи підпрограми інтелектуального вибору рівня складності представлено на рисунку 4.2. Підпрограма отримує поведінковий вектор гравця, поточну конфігурацію ігрового середовища та перелік допустимих варіантів наступного етапу. Для кожного кандидата модель машинного навчання визначає прогнозовану ймовірність успішного проходження. Після цього результати порівнюються з установленим цільовим діапазоном. Якщо прогноз свідчить про надмірну простоту завдання, може бути обрано складнішу конфігурацію. Якщо очікувана результативність недостатня, перевага надається доступнішому варіанту. Остаточний вибір обмежується правилами поступової адаптації, які запобігають різким змінам ігрового процесу.

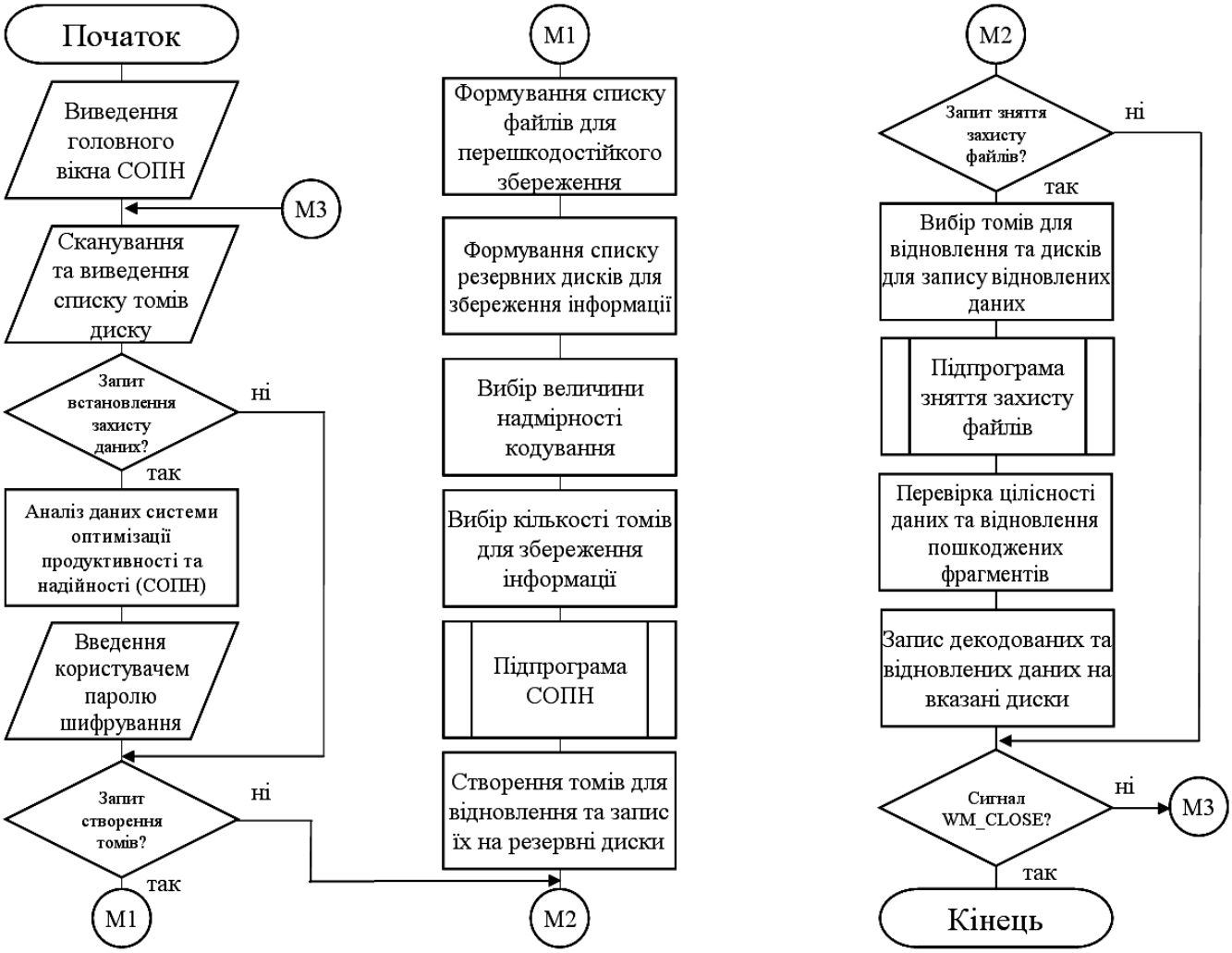

Рисунок 4.1 – Блок-схема основної програми

Під час реалізації адаптивного регулювання необхідно розрізняти фактичну результативність і рівень ігрових навичок. Висока кількість набраних балів не завжди свідчить про достатню підготовленість. Вона може бути наслідком проходження простого рівня або використання допоміжних механізмів. Аналогічно невдала спроба не обов'язково означає низький рівень навичок. На результат можуть впливати випадкові події, тимчасове відволікання та особливості керування. Тому оцінювання необхідно виконувати за сукупністю показників, які враховують умови проходження та історію попередньої активності.

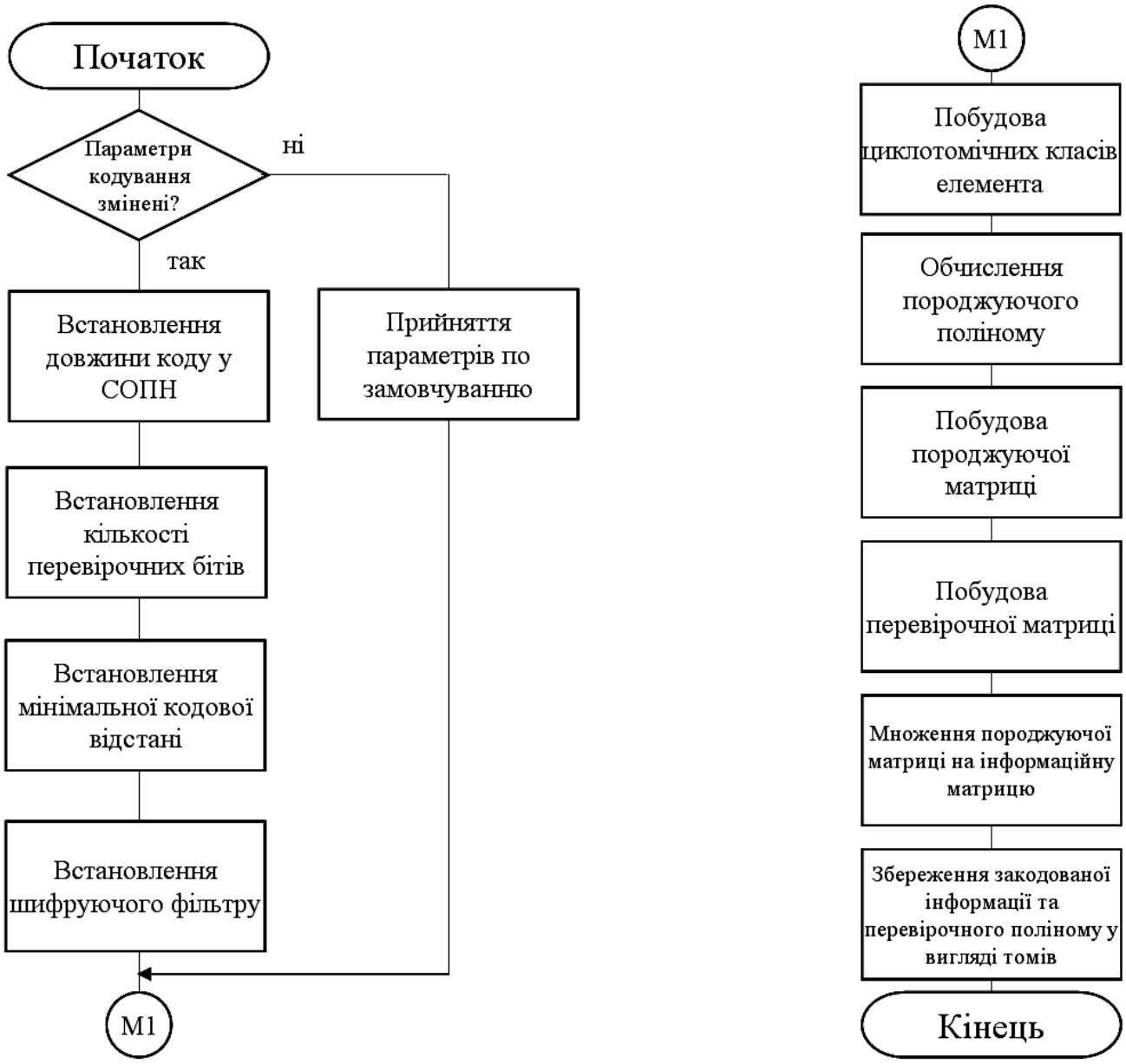

Рисунок 4.2 – Блок-схема роботи підпрограми

Алгоритм ініціалізації ігрового середовища

Початковим етапом функціонування системи є підготовка мультимедійного комплексу до нового сеансу. Вхідними даними алгоритму є вибраний ігровий режим, доступні параметри керування та відомості про користувацький профіль, якщо він існує.

На першому етапі перевіряється доступність необхідних програмних ресурсів. До них можуть належати графічні об'єкти, звукові файли, конфігурації рівнів і параметри ігрових механік. Якщо необхідний ресурс недоступний,

система повинна визначити можливість використання резервної конфігурації або завершити ініціалізацію з відповідним повідомленням.

Далі створюється ідентифікатор ігрового сеансу. Для нього визначаються початковий рівень складності, обраний режим та поточний стан проходження. Якщо користувач має збережений профіль, його характеристики можуть використовуватися для вибору стартової конфігурації.

Для нового користувача доцільно встановлювати початковий рівень, який дозволяє виконати базові дії та отримати перші спостереження. Завдання не повинно вимагати попереднього знання складних ігрових механік, якщо вони ще не були представлені користувачу.

Після підготовки конфігурації активується ігровий цикл. Він забезпечує оновлення стану об'єктів, оброблення дій користувача, визначення зіткнень і відображення мультимедійних елементів.

Результатом алгоритму є підготовлений ігровий сеанс із визначеними початковими параметрами.

Алгоритм керування станами ігрового процесу

Для узгодження ігрової логіки з механізмами машинного навчання доцільно використовувати модель станів. Вона дозволяє визначати моменти, у які дозволяється оновлювати параметри складності.

Початковим станом є підготовка сеансу. Після завершення ініціалізації виконується перехід до активного проходження. У цьому стані ігровий модуль обробляє керувальні дії та змінює положення об'єктів відповідно до поточної конфігурації.

Після завершення етапу система переходить до стану оцінювання. У ньому обчислюються підсумкові характеристики результативності та формується запит до модуля адаптації. Після отримання рішення виконується підготовка наступного етапу.

Окремо доцільно передбачити стани паузи, завершення сеансу та відновлення після технічної помилки. Перебування в стані паузи не повинно збільшувати показник часу реакції або враховуватися як активне проходження.

Зміна складності під час незавершеної ігрової дії може призвести до некоректної оцінки результату. Наприклад, раптове прискорення об'єктів безпосередньо перед завершенням завдання порушує однаковість умов проходження. Тому основним моментом застосування адаптації доцільно обрати перехід між рівнями, хвилями або іншими завершеними етапами.

Результатом алгоритму є керований життєвий цикл сеансу, у якому ігрова логіка та аналітичні операції виконуються без взаємних порушень.

Алгоритм реєстрації користувацької активності

Для побудови поведінкової моделі необхідно отримувати інформацію про дії гравця в процесі виконання завдань. Такими діями можуть бути натискання клавіш, переміщення керувального об'єкта, вибір цілі, виконання успішної дії або допущення помилки.

Вхідними даними алгоритму є внутрішні повідомлення ігрового модуля. На першому етапі визначається тип події. Далі до неї додаються ідентифікатор сеансу, номер етапу, час виникнення та параметри поточної складності.

Для реєстрації реакції може використовуватися інтервал між появою відповідного стимулу та правильною дією користувача. Однак такий показник необхідно застосовувати лише для завдань, у яких визначено момент появи стимулу та очікувану реакцію.

Окремо реєструються результати виконання. До них можуть належати кількість успішних дій, кількість помилок, тривалість проходження, кількість використаних спроб та підсумковий ігровий результат.

Важливо не зберігати необмежений потік усіх подій без практичної необхідності. Для подальшого аналізу часто достатньо агрегованих характеристик завершеного етапу. Детальні записи доцільно використовувати переважно під час тестування алгоритмів або діагностики.

Результатом є структурований набір повідомлень про активність гравця з відомостями про умови її виникнення.

Алгоритм попереднього оброблення ігрової телеметрії

Отримані події можуть містити повтори, пропущені значення та нерівномірні часові інтервали. Перед застосуванням машинного навчання необхідно виконати їхню перевірку.

На початковому етапі визначається належність кожного повідомлення до відповідного ігрового сеансу. Події, які не мають коректного ідентифікатора або виникли після завершення сеансу, обробляються окремо.

Далі виконується перевірка часових характеристик. Якщо час реакції має від'ємне значення або перевищує допустимі межі через технічну помилку, він не використовується як звичайне спостереження.

Наступним етапом є групування подій за завершеними етапами. Для кожного етапу формуються агреговані показники. Це дозволяє зменшити обсяг інформації та забезпечити однакове представлення результатів різних проходжень.

Окремо контролюються події, що виникли під час паузи або втрати фокуса ігрового вікна. Такі інтервали можуть виключатися з обчислення активного часу. Інакше модель може помилково інтерпретувати тимчасову відсутність дій як погіршення навичок. Результатом алгоритму є очищена ігрова телеметрія, придатна для формування характеристик поведінки.

Алгоритм формування вектора результативності гравця

Для використання алгоритмів машинного навчання результати проходження необхідно перетворити на числовий вектор. Він повинен відображати не лише підсумкові бали, а й характер виконання завдань.

Вхідними даними є агреговані показники завершених етапів та інформація про їхню складність. На першому етапі визначається частка успішних дій. Для цього враховується кількість правильно виконаних операцій відносно загальної кількості спроб. Далі аналізується частота помилок. Для аркадних режимів, які потребують швидкого реагування, може використовуватися кількість невдалих дій за одиницю активного часу. Для інших режимів доцільніше враховувати помилки відносно кількості завдань.

Окремо визначається швидкість виконання. Наприклад, час реакції може характеризувати здатність користувача своєчасно взаємодіяти з рухомими об'єктами. Для зменшення впливу випадкових затримок доцільно використовувати медіану або інші стійкі статистичні показники.

Важливим є врахування поточної складності. Однакова частка успішних дій на різних рівнях не обов'язково означає однакову підготовленість. Тому до вектора додаються параметри середовища, за яких отримано відповідний результат.

Після формування числових характеристик виконується їхнє масштабування. Для цього можуть застосовуватися стандартизація або нормалізація до визначених діапазонів. Параметри перетворення необхідно визначати на навчальній вибірці та зберігати разом із моделлю.

Результатом алгоритму є узгоджений поведінковий вектор, який використовується для оцінювання навичок і прогнозування проходження.

Алгоритм формування індивідуального профілю гравця

Для персоналізації складності необхідно накопичувати інформацію про характерні результати конкретного користувача. Такий профіль дозволяє враховувати не лише останній етап, а й попередню динаміку проходження.

Вхідними даними є поведінкові вектори завершених етапів. На першому етапі перевіряється достатність спостережень. Якщо користувач лише розпочав гру, профіль може бути недостатньо інформативним.

Для накопичених даних обчислюються узагальнені характеристики. До них можуть належати середня успішність, типова частота помилок, характерний час реакції та показники стабільності результатів.

Для врахування змін навичок може використовуватися експоненційне згладжування. Нові результати впливають на поточний профіль із визначеним коефіцієнтом. Це дозволяє поступово враховувати навчання користувача.

Водночас надмірна залежність від останнього результату є небажаною. Одна невдала спроба не повинна різко змінювати оцінку підготовленості. Тому доцільно використовувати декілька попередніх спостережень і контролювати

розкид показників. Результатом є динамічний профіль результативності, який зберігається в межах сеансу або між сеансами за наявності відповідного користувацького механізму.

Алгоритм початкового оцінювання навичок

Для нового гравця модель не має достатньої історії поведінки. У такому випадку необхідно використовувати процедуру початкового оцінювання, яка не потребує тривалого проходження.

Вхідними даними є результати перших ігрових етапів. На початку встановлюється базова складність із передбачуваними умовами виконання завдань.

Після завершення першого етапу система визначає основні характеристики користувача. Якщо він демонструє стабільну результативність, наступний етап може отримати дещо вищу складність. Якщо кількість помилок значна, поточні параметри зберігаються або коригуються в бік спрощення.

Для початкового оцінювання доцільно обмежувати максимальний розмір зміни. Це запобігає ситуації, коли випадково успішний результат призводить до різкого ускладнення гри.

Після накопичення достатньої кількості даних система може перейти від початкової евристичної процедури до прогнозування на основі машинного навчання.

Результатом алгоритму є початковий рівень підготовленості та набір спостережень для подальшої персоналізації.

Алгоритм групування гравців за поведінковими характеристиками

Для підвищення якості початкового налаштування може застосовуватися кластеризація історичних поведінкових профілів. Вона дозволяє визначати групи користувачів із подібними характеристиками проходження.

Вхідними даними є нормалізовані вектори результативності, сформовані за порівнянних ігрових умов. На першому етапі визначається набір ознак, які характеризують навички та стиль проходження.

Для групування може використовуватися K-means. Алгоритм формує визначену кількість кластерів у просторі поведінкових характеристик. Кожен кластер описується відповідним центром.

У результаті можуть бути сформовані групи, які відрізняються швидкістю реакції, точністю виконання дій або стабільністю результатів. Однак такі групи не слід без додаткової перевірки ототожнювати з фіксованими категоріями майстерності.

Для нового користувача визначається найближча група за доступними початковими спостереженнями. Її характеристики можуть використовуватися для формування попереднього прогнозу.

Зі збільшенням кількості індивідуальних результатів вплив групового профілю може поступово зменшуватися. Таким чином система переходить від узагальненої моделі до персоналізованого оцінювання.

Результатом алгоритму є груповий контекст, який допомагає формувати стартові параметри складності за недостатньої користувацької історії.

Алгоритм навчання моделі прогнозування успішності

Центральним елементом інтелектуального регулювання є модель, яка прогнозує результат проходження наступного ігрового етапу залежно від характеристик гравця та складності завдання.

Вхідними даними алгоритму є історія завершених етапів. Кожен навчальний запис повинен містити характеристики гравця, доступні до початку етапу, параметри його складності та фактичний результат.

Як цільова ознака може використовуватися успішне завершення етапу. У такому випадку завдання машинного навчання формулюється як бінарна класифікація з оцінюванням імовірності успіху.

Для початкової реалізації можуть використовуватися Random Forest, Gradient Boosting або інші алгоритми класифікації. Вони дозволяють аналізувати нелінійні залежності між поведінковими характеристиками та параметрами складності.

Важливо, щоб навчальний набір містив приклади проходження за різних рівнів складності. Якщо користувачі з високою підготовленістю завжди отримували лише складні завдання, а нові гравці лише прості, модель може засвоїти залежність між рівнем складності та характеристиками користувача без достатніх даних для оцінювання альтернативних умов.

Для зменшення такого зміщення доцільно використовувати контрольовані експериментальні проходження з обмеженим варіюванням складності. Отримані результати формують навчальні спостереження для оцінювання впливу різних конфігурацій.

Під час формування вибірок необхідно уникати витоку інформації. До вхідного вектора не можна включати підсумкові характеристики того етапу, успішність якого прогнозується. Для оцінювання якості на нових користувачах доцільно розділяти навчальні та тестові дані за гравцями.

Після навчання модель перевіряється на окремій вибірці. Крім точності класифікації, необхідно оцінювати відповідність прогнозованих імовірностей фактичній частоті успіху. Для цього може застосовуватися калібрування ймовірностей.

Результатом алгоритму є модель прогнозування, яка отримує поведінкові характеристики гравця та складність кандидата і повертає оцінку ймовірності успішного проходження.

Алгоритм інтелектуального вибору рівня складності

Після отримання результатів машинного навчання необхідно визначити конкретну конфігурацію наступного етапу. Для цього формується перелік допустимих кандидатів.

Вхідними даними алгоритму є індивідуальний профіль, прогнозна модель, поточний рівень та встановлені обмеження адаптації.

На першому етапі визначається допустимий діапазон складності. Він залежить від ігрового режиму та попередньої конфігурації. Для запобігання різким змінам максимальний перехід між сусідніми етапами доцільно обмежувати.

Після формування кандидатів модель прогнозує ймовірність успішного проходження кожного варіанта. Для цього використовується один і той самий поведінковий профіль із різними значеннями параметрів складності.

Наступним етапом є порівняння результатів із цільовою зоною успішності. Її значення визначається завданнями ігрового дизайну та уточнюється експериментально.

Якщо прогнозована ймовірність успіху істотно перевищує бажану, система може обрати складніший варіант. Якщо вона нижча за допустиму межу, доцільно спростити завдання.

За наявності декількох прийнятних кандидатів перевага може надаватися конфігурації, яка найменше відрізняється від поточної. Це забезпечує більш передбачувану адаптацію.

Якщо прогноз моделі недостатньо достовірний, система повинна зберігати попередню конфігурацію або використовувати заздалегідь визначене евристичне правило.

Результатом алгоритму є рішення про рівень складності наступного етапу з поясненням підстав його вибору.

Алгоритм обмеження швидкості зміни складності

Безпосереднє застосування прогнозованого рівня не завжди забезпечує комфортне проходження. Якщо після одного успішного етапу складність різко зростає, користувач може отримати непропорційно складне завдання.

Для запобігання цьому передбачається алгоритм згладжування керувальних рішень.

Вхідними даними є рекомендована складність, поточна конфігурація та історія попередніх змін. На першому етапі визначається різниця між поточним і рекомендованим значеннями.

Якщо різниця перевищує встановлене обмеження, зміна скорочується до допустимого кроку. Таким чином навіть значне відхилення прогнозованої успішності не призводить до неконтрольованого переходу.

Окремо використовується механізм гістерезису. Він дозволяє уникати багаторазового переключення між двома сусідніми рівнями за незначних коливань результативності.

Для цього можуть застосовуватися різні пороги підвищення та зниження складності. Додатково визначається мінімальна кількість етапів між послідовними змінами.

За повторюваних невдач може передбачатися швидше зменшення складності, ніж її підвищення. Однак така асиметрія повинна перевірятися з урахуванням особливостей ігрового режиму.

Результатом алгоритму є згладжена конфігурація, яка відповідає обмеженням поступової адаптації.

Алгоритм перетворення рівня складності на ігрові параметри

Рекомендований рівень складності є узагальненим показником. Для його практичного використання необхідно визначити конкретні параметри ігрового середовища.

Вхідними даними є вибраний рівень складності та характеристики активного аркадного режиму.

Для гри, орієнтованої на швидкість реакції, можуть налаштовуватися швидкість руху цілей, тривалість їхньої доступності та часовий інтервал між появою об'єктів.

Для режиму ухилення від перешкод доцільно змінювати щільність перешкод, швидкість переміщення або характеристики безпечних зон. При цьому необхідно зберігати можливість виконання завдання за передбаченими правилами.

Для аркадного режиму з нарахуванням очок можуть використовуватися параметри частоти появи цілей, кількості одночасних об'єктів і допустимої кількості помилок.

На початковому етапі визначається відповідна конфігурація. Після цього виконується перевірка її допустимості. Наприклад, швидкість руху об'єктів не

повинна перевищувати значення, за якого завдання стає технічно невиконуваним у межах установлених правил.

Окремо необхідно контролювати взаємний вплив параметрів. Одночасне збільшення кількості об'єктів і швидкості їхнього переміщення може створювати значно більший ефект, ніж окреме налаштування кожного показника.

Результатом алгоритму є конкретний набір ігрових параметрів, який передається модулю виконання наступного етапу.

Алгоритм керування мультимедійним зворотним зв'язком

Мультимедійні засоби використовуються для відображення результатів дій і повідомлення про зміни ігрового середовища. До них можуть належати графічні індикатори, анімації, звукові ефекти та текстові підказки.

Вхідними даними алгоритму є стан ігрового сеансу, результати проходження та обрана конфігурація складності.

На першому етапі визначається тип повідомлення. Успішне виконання завдання може супроводжуватися відповідним графічним або звуковим сигналом. Невдала спроба може відображатися через окрему анімацію.

Якщо система змінює складність між етапами, доцільно забезпечити зрозуміле повідомлення про нові умови. Це дозволяє уникати ситуації, коли користувач сприймає зміну швидкості об'єктів як технічну помилку.

Окремо необхідно враховувати налаштування доступності. Звукова гучність, контрастність і можливість відключення окремих ефектів не повинні довільно використовуватися як засоби підвищення складності.

За потреби може застосовуватися система підказок. Її активація повинна враховувати правила конкретного режиму та фіксуватися в телеметрії. Це необхідно для правильного інтерпретування результативності проходження з допомогою.

Результатом алгоритму є узгоджене мультимедійне представлення ігрового процесу без порушення логіки адаптації.

Алгоритм застосування рішення до ігрового модуля

Після вибору конфігурації необхідно передати її компоненту, який безпосередньо керує виконанням гри.

Вхідними даними є рекомендований рівень, його конкретні параметри та поточний стан ігрового сеансу.

На першому етапі перевіряється можливість застосування зміни. Якщо гравець ще виконує поточне завдання, рішення зберігається до завершення етапу.

Після переходу до стану підготовки наступного рівня ігровий модуль отримує нову конфігурацію. Далі перевіряється допустимість усіх параметрів.

Для забезпечення узгодженості доцільно застосовувати набір параметрів як єдину конфігурацію. Якщо частина значень некоректна, система повинна відхилити зміну повністю або використати заздалегідь перевірений резервний варіант.

Після успішного застосування в історії сеансу фіксується фактична конфігурація. Це необхідно для подальшого навчання, оскільки результат проходження повинен зіставлятися саме з реально використаними параметрами, а не з попередньою рекомендацією.

Результатом алгоритму є підготовлений наступний етап з узгодженими характеристиками складності.

Алгоритм оновлення поведінкової моделі

Під час експлуатації користувач може поступово вдосконалювати навички. Крім того, до комплексу можуть додаватися нові ігрові режими. Тому попередньо навчена модель не завжди зберігатиме однакову якість прогнозування.

Для врахування таких змін передбачається процедура періодичного оновлення моделі.

Вхідними даними є нові записи завершених етапів, попередня навчальна вибірка та результати перевірки поточної моделі. На першому етапі виконується контроль повноти нових даних.

Далі формуються ознаки з урахуванням фактичної складності проходження. Необхідно виключити записи, у яких результат був суттєво

спотворений технічною помилкою, втратою фокуса або іншими неврахованими чинниками.

Після накопичення достатнього обсягу даних може виконуватися повторне навчання. Для моделей, які підтримують інкрементальне оновлення, допускається використання відповідного механізму. Для інших алгоритмів необхідне повторне навчання на підготовленому наборі.

Нова модель перевіряється на контрольних спостереженнях. Якщо якість прогнозування не відповідає встановленим вимогам, попередня версія залишається активною.

Результатом алгоритму є оновлена модель, яка може використовуватися для подальшого регулювання після підтвердження її придатності.

Алгоритм виявлення нестабільної результативності

Окремим завданням є визначення ситуацій, у яких короткочасні результати не відповідають звичайній поведінці користувача. Причиною може бути випадкова невдача, зміна умов керування або тимчасове зниження концентрації.

Вхідними даними алгоритму є послідовність результатів декількох завершених етапів. На першому етапі аналізується розкид показників результативності. Далі визначається наявність різких змін точності, часу реакції та кількості помилок. Якщо спостерігається значна нестабільність, доцільно зменшити швидкість адаптації. Система може зберігати поточну конфігурацію протягом додаткових етапів для накопичення більш достовірних даних. Для розрізнення короткочасного відхилення та стійкого погіршення може використовуватися ковзне вікно спостережень. Після накопичення даних перевіряється, чи зберігається відповідна тенденція. Результатом алгоритму є оцінка стабільності користувацької результативності, яка впливає на допустиму інтенсивність зміни складності.

Алгоритм оброблення помилок і відмов компонентів

Під час виконання ігрового комплексу можливі помилки отримання телеметрії, недоступність моделі машинного навчання, некоректні параметри адаптації та порушення збереження результатів.

Для запобігання впливу таких ситуацій на основний ігровий процес необхідно відокремити роботу модуля адаптації від механізмів відображення графіки та оброблення керувальних дій.

На першому етапі кожного запиту перевіряється коректність сформованого поведінкового вектора. Якщо частина необхідних характеристик відсутня, модель не повинна отримувати довільно підставлені значення без передбаченої процедури їхньої обробки.

Якщо прогнозування завершилося помилкою, система може зберегти попередню складність або використати базове евристичне правило. Відмова аналітичного компонента не повинна призводити до аварійного завершення ігрового сеансу.

Окремо контролюється час виконання прогнозу. Якщо відповідь не надходить у межах допустимого інтервалу, ігровий модуль повинен продовжити підготовку наступного етапу з безпечною конфігурацією.

Результатом алгоритму є кероване функціонування комплексу за умов часткової недоступності аналітичних компонентів.

Особливості програмної реалізації системи

Програмну реалізацію доцільно організувати таким чином, щоб ігрова логіка не залежала від конкретного алгоритму машинного навчання. Для цього між ігровим модулем та аналітичним компонентом використовується визначений програмний інтерфейс.

Ігровий модуль передає характеристики завершеного етапу та отримує результат вибору наступної складності. Для нього не повинні мати значення внутрішня структура класифікатора або спосіб навчання моделі.

Для безпосередньої реалізації мультимедійного середовища може використовуватися ігровий рушій із підтримкою двовимірної графіки, звукових ефектів та оброблення подій. Конкретний рушій і мова програмування визначаються фактичною реалізацією комплексу.

Важливо забезпечити незалежність частоти оновлення графічного середовища від швидкості виконання аналітичної моделі. Прогнозування доцільно

виконувати асинхронно або в окремому компоненті. Це дозволяє уникати блокування ігрового циклу під час оброблення даних.

Для зберігання результатів проходження може застосовуватися локальна база даних або відповідне серверне сховище. Необхідно забезпечити зв'язок між користувацьким сеансом, параметрами складності та отриманим результатом.

Забезпечення конфіденційності та коректності користувацьких даних

Система адаптивного регулювання накопичує відомості про дії користувачів і характер їхнього проходження. Для виконання основних алгоритмів зазвичай достатньо технічних характеристик ігрової активності без збирання додаткової особистої інформації.

На початковому етапі визначається перелік необхідних показників. До нього можуть належати частота помилок, тривалість проходження та результати виконання завдань.

Для збереження історії може використовуватися псевдонімізований ідентифікатор користувача. Якщо персоналізація між сеансами не потрібна, профіль може існувати лише протягом одного запуску гри.

Окремо необхідно контролювати доступ до збережених даних і параметрів моделей. Некоректна зміна навчальної вибірки або підміна моделі може призводити до неправильного вибору складності.

Для забезпечення відтворюваності результатів доцільно зберігати версію моделі та конфігурацію адаптації, які використовувалися під час конкретного ігрового сеансу.

Результатом відповідних механізмів є контрольована обробка користувацької телеметрії з обмеженням зайвого накопичення інформації.

Перевірка коректності алгоритмів та оцінювання ефективності

Для дослідження роботи запропонованого алгоритмічного забезпечення необхідно передбачити функціональні та експериментальні випробування. Основним завданням є визначення правильності оброблення користувацької активності, якості прогнозування результативності та впливу адаптивного регулювання на проходження гри.

Початковим етапом є перевірка алгоритмів реєстрації подій. Для цього використовуються контрольовані сценарії, у яких відомі час появи стимулу, момент дії користувача та очікуваний результат. Необхідно перевірити правильність визначення часових характеристик і формування підсумкових показників.

Далі перевіряється побудова поведінкових векторів. Для контрольованих наборів подій обчислюються очікувані значення точності, частоти помилок і часу реакції. Отримані результати порівнюються з даними програмного компонента.

Наступним етапом є перевірка моделі машинного навчання. Для цього формується набір історичних проходжень, розділений на навчальну та тестову частини. Доцільно забезпечити, щоб тестові користувачі не були представлені в навчальному наборі під час оцінювання здатності моделі працювати з новими гравцями.

Для оцінювання прогнозованої ймовірності успіху можуть використовуватися Brier Score та Log Loss. Додатково перевіряється калібрування прогнозів, тобто відповідність передбачених імовірностей фактичній частоті успішного проходження.

Окремо необхідно оцінити алгоритм вибору складності. Для цього формуються контрольовані поведінкові профілі з різною результативністю. Система повинна коректно визначати допустимі конфігурації та дотримуватися обмежень на величину зміни.

Для перевірки стабільності можуть використовуватися послідовності чергування успішних і невдалих проходжень. Під час аналізу необхідно визначити, чи не призводять невеликі коливання результатів до постійного переключення між рівнями.

Експериментальне порівняння може передбачати три режими роботи комплексу. Перший використовує фіксовану складність, другий реалізує евристичне регулювання, а третій застосовує запропонований алгоритм машинного навчання. Порівняння необхідно виконувати за узгодженими правилами проходження.

Основними показниками можуть бути частка успішно завершених етапів, кількість повторних спроб, тривалість проходження та частота зміни складності. Для оцінювання стабільності адаптації доцільно враховувати величину й напрямок відповідних змін.

Додатково може досліджуватися суб'єктивне сприйняття складності користувачами. Для цього застосовуються опитування після проходження або інші дозволені способи отримання зворотного зв'язку. Необхідно враховувати, що висока успішність сама по собі не підтверджує зацікавленості чи задоволеності грою.

Для визначення впливу машинного навчання доцільно виконати окреме порівняння прогнозної моделі з простими правилами зміни складності. Такий підхід дозволяє встановити, чи забезпечує використання складнішого алгоритму практично значущу відмінність.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання. Головне вікно програми показане на рисунку 5.1. Впровадження програмного комплексу передбачає підготовку середовища виконання, встановлення програмних залежностей, налаштування параметрів та підключення модуля машинного навчання. Основними функціональними складовими є мультимедійний ігровий модуль, компонент реєстрації користувацької активності, підсистема формування поведінкових характеристик і модуль адаптивного керування складністю. Для аналітичного компонента як можливе технологічне рішення розглядається C#. Комп'ютер повинен мати достатню продуктивність процесора, оперативну пам'ять та графічні засоби для стабільного відображення.

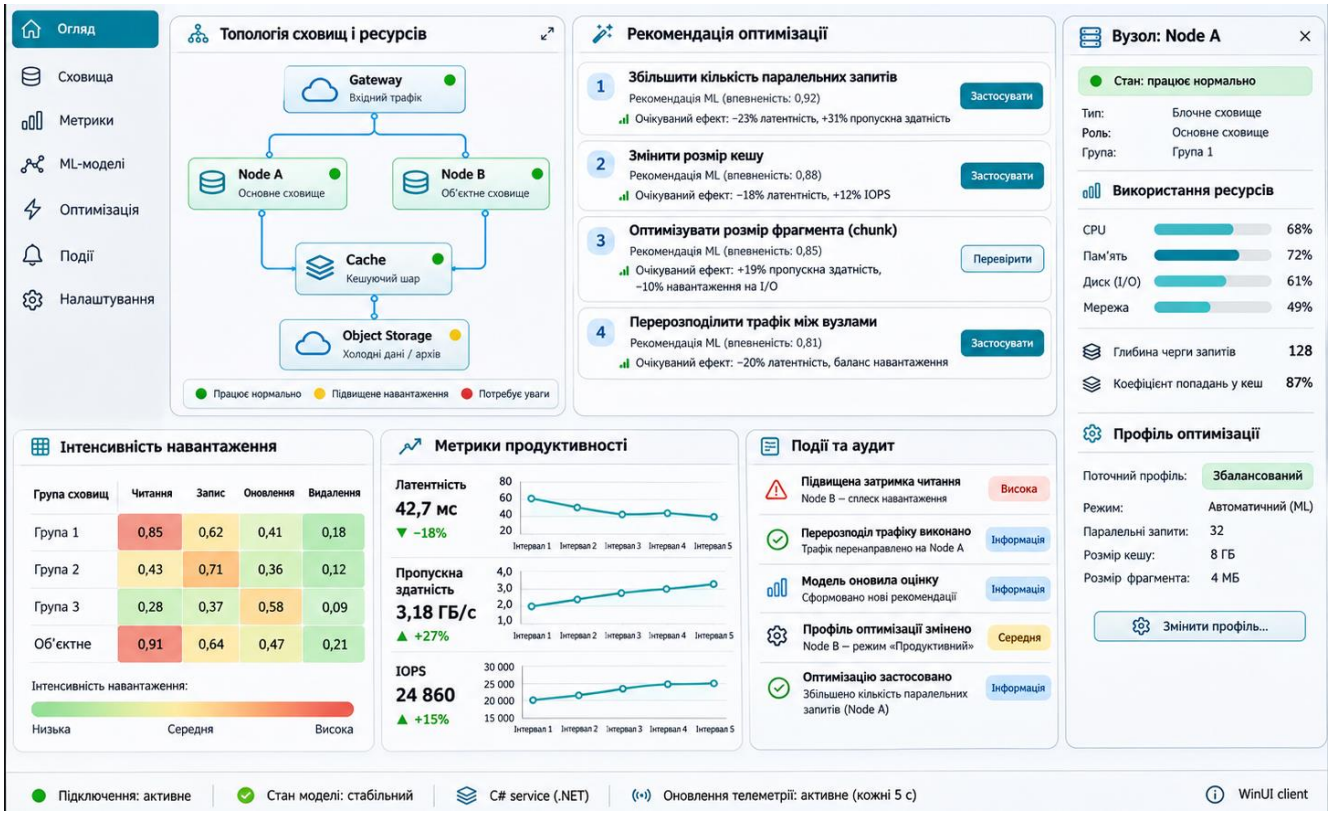

Рисунок 5.1 – Головне вікно розробленого ПЗ

За використання окремого аналітичного сервісу необхідно забезпечити його доступність. Водночас для компактних моделей класифікації можливе локальне виконання без виділеного сервера або графічного прискорювача.

Підготовка програмного середовища починається зі встановлення необхідних компонентів і перевірки їхньої сумісності.

Важливо забезпечити, щоб усі можливі конфігурації залишалися технічно виконуваними та відповідали встановленим правилам гри.

Для збереження результатів може використовуватися локальна база даних або серверне сховище, якщо комплекс передбачає роботу з декількома користувачами. Необхідно забезпечити зв'язок між сеансом, поведінковими характеристиками та застосованою конфігурацією складності. За потреби зберігаються попередні результати користувача для персоналізації наступних запусків. Доступ до цих відомостей повинен відповідати встановленим правилам оброблення користувацьких даних.

Окремої підготовки потребує модель машинного навчання. Для її створення використовуються результати завершених ігрових етапів, що містять поведінкові характеристики, параметри складності та фактичну успішність проходження. Як можливі алгоритми розглядаються Random Forest і Gradient Boosting. Навчання доцільно виконувати до розгортання основного програмного комплексу. Підготовлені параметри моделі зберігаються разом із відомостями про структуру та масштабування вхідних ознак.

Під час підключення навченої моделі необхідно перевірити відповідність її вхідних даних характеристикам, які формує ігровий модуль. Зокрема, однаковими повинні залишатися порядок ознак, одиниці вимірювання та правила їхнього перетворення. Невідповідність цих параметрів може спричиняти некоректне прогнозування навіть за успішного завантаження моделі. Перед використанням необхідно перевірити, що прогнозовані ймовірності мають допустимі значення.

Наступним етапом є налаштування правил адаптації. Для цього визначаються мінімальний і максимальний рівні складності, допустима величина переходу між етапами та цільовий діапазон успішності. Окремо встановлюється

мінімальна кількість спостережень для персоналізованого прогнозування. До накопичення необхідної історії доцільно використовувати початкову конфігурацію або обмежені евристичні правила. Конкретні значення параметрів повинні уточнюватися за результатами експериментальної перевірки.

Застосування нових параметрів складності необхідно узгодити зі станами процесу. Рекомендовану конфігурацію доцільно передавати після завершення рівня або хвилі. Це запобігає непередбачуваній зміні швидкості об'єктів чи кількості перешкод безпосередньо під час виконання завдання. Після прийняття рішення ігровий модуль перевіряє допустимість параметрів і використовує їх під час підготовки наступного етапу.

Рекомендована послідовність запуску починається з перевірки мультимедійних ресурсів і доступності сховища користувацьких результатів. Далі ініціалізується аналітичний компонент і завантажуються параметри навченої моделі. Після успішної перевірки прогнозування активується модуль. Під час створення сеансу визначається початкова складність, запускається реєстрація подій та забезпечується взаємодія між компонентами й адаптації.

Для перевірки адаптивного регулювання можуть використовуватися заздалегідь підготовлені профілі з різною результативністю. Необхідно перевірити правильність отримання прогнозу, вибору наступної складності та дотримання обмежень на її зміну. Окремо контролюються випадки, коли модель не повертає коректного результату. За таких умов комплекс повинен продовжувати роботу з попередньою або резервною конфігурацією.

Додатково перевіряється стабільність мультимедійного середовища під час виконання аналітичних операцій. Прогнозування не повинно блокувати оновлення графіків або оброблення керувальних дій. Для цього за потреби використовуються асинхронні виклики чи окремий процес аналізу. Під час випробувань контролюються затримка формування прогнозу, стабільність частоти кадрів і правильність переходів між ігровими станами.

У процесі експлуатації можливі помилки завантаження моделі, відсутність мультимедійного ресурсу, некоректні поведінкові дані та порушення записування

результатів. Для діагностики доцільно використовувати службові журнали, які містять відомості про стан компонентів і причини помилок. Перед регулярним використанням необхідно перевірити відновлення роботи після перезапуску, збереження користувацьких налаштувань і можливість продовження гри за недоступності аналітичного компонента.

Для ідентифікації програмного продукту передбачається представлення відомостей про його розробника. Така інформація використовується під час документування системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання та організації подальшого супроводження програмного забезпечення. Авторські дані системи адаптивного регулювання складності із застосуванням алгоритмів машинного навчання представлені на рисунку 5.2.

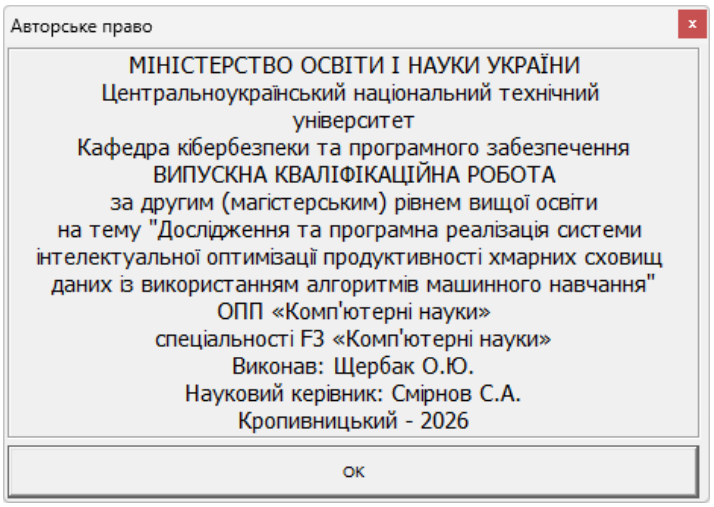

Рисунок 5.2 – Вікно розробника ПЗ

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Об'єктом дослідження є процес інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Предметом дослідження є методи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Методи дослідження базуються на методах хмарних обчислень, методах машинного навчання, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

– Розроблено вітчизняний продукт інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

В охорону праці включають санітарно-гігієнічні, лікувально-профілактичні та організаційно-технічні системи правових і соціально-економічних заходів.

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

– Інструкцій з охорони праці по кожній професії і загальні.

– Положення про охорону праці.

– Накази з охорони праці.

– Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий

акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-97.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-17.

Умови праці програміста включають наступні фактори:

– параметри повітряного середовища в приміщенні;

– вентиляція приміщення;

– освітлення приміщення;

– параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп’ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

7.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 7.1.

У зазначеному приміщенні працюють 9 людей. За даними, які наведено у табл. 7.1, та табл. 7.2, можна зробити висновок, що площа та об’єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Таблиця 7.1 – Розміри приміщення

Найменування	Значення, м
Ширина	7
Довжина	8
Висота	3,3

Таблиця 7.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	$м^2$	не менше 6.0	6,2
Об'єм, V	$м^3$	не менше 20.0	20,5

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 7.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 7.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Воло-гість,%	Швидкість повітря, м/с	Температура, °С	Воло-гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-24	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,9

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року. В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер Xerox WorkCentre 3025BI (3025VBI), електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України "Природне і штучне освітлення" – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у тому числі світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинно бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

7.3 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином, можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

7.4 Розрахункова частина

Система освітлення робочого місця користувача ПК має відповідати наступним вимогам (рис. 7.1).

Проведемо розрахунок штучного освітлення за методом коефіцієнту використання світлового потоку для приміщення ширина якого складає 7 м, довжина – 8 м, висота – 3,3 м.

У зазначеному приміщенні працює 9 людей.



Рисунок 7.1 – Вимоги до системи освітлення робочого місця користувача ПК

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = E \cdot S \cdot K \cdot Z/n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; *E = 300 Лк;*

S – площа освітлюваного приміщення.

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку *K = 1,5);*

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку *Z = 1,1*);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін.}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін}$ = 50% і $\rho_{стелі}$ = 50%.

Обчислимо індекс приміщення за формулою:

$$i = S/(h \cdot (A+B)),$$

де:

S – площа приміщення, *S* = 56 $м^2$*;*

h – розрахункова висота підвісу, *h* = 3,2 *м. (співпадає з в*исотою стелі, т.я. лампи освітлення закріплюються на стелі)*;*

A – ширина приміщення, *A = 7 м;*

B – довжина приміщення, *B* = 8 *м.*

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$i=1,1.$$

Знаючи індекс приміщення, за знаходимо *n = 0,46 (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним*

типом лампам) [8]. Підставимо всі значення у формулу, визначимо світловий потік: F=60260 Лм.

Для розрахунку будемо використовувати стельові *світлодіодні панелі* Призма-72 6400К, світловий потік яких $F_л$ = 7200 *Лм.*

Число ламп визначається по формулі:

$$N=F/F_л$$

де:

F – світловий потік,

$F_л$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та кількість світильників:

N= 60260 / 7200=8,3 шт.

Приймаємо необхідну кількість *світлодіодних світильників* 9 шт.

7.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров’я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров’я програміста. Розроблено заходи з охорони праці.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

– Досліджена система інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Visual C#. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Щербак О.Ю. Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності хмарних сховищ даних із використанням алгоритмів машинного навчання // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.
2. Brink H., Richards J., Fetherolf M. Real-World Machine Learning. – Manning, 2016. – 474 p.
3. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
4. Fenner M. Machine Learning with Python for Everyone (Addison-Wesley Data & Analytics Series) 1st Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 586 p.
5. Foreman J.W. Data Smart: Using Data Science to Transform Information into Insight 1st Edition. – Wiley, 2013. – 432 p.
6. Hurbans R. Grokking Artificial Intelligence Algorithms. – Manning, 2020. – 631 p.
7. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology 1st Edition. – Cambridge University Press, 2008. – 556 p.
8. Kotu V., Deshpande B. Data Science: Concepts and Practice. – Elsevier Science, 2018. – 953 p.
9. Knowledge Base A Complete Guide – 2021 Edition // The Art of Service – Knowledge Base Publishing, 2020. – 306 p.
10. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
11. Mattmann C. Machine Learning with TensorFlow, Second Edition. – Manning, 2020. – 1124 p.

12. Mueller J.P., Massaron L. Machine Learning For Dummies. – Wiley, 2016. – 714 p.

13. Teofili T. Deep Learning for Search. – Manning, 2019. – 695 p.

14. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. – Publishdrive, 2019. – 587 p.

15. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. – O'Reilly. 2019. – 252 p.

16. Rajasekaran S., Vijayalakshmi Pai G.A. Neural networks, fuzzy logic, and genetic algorithms: synthesis and applications (with cd-rom) Kindle Edition. – PHI, 2013. – 628 p.

17. Vintenko, B., Myronets, I., Smirnov, O., Smirnova, T. «Enhancing the Reliability of Input Data for Nuclear Power Plant Operator Support Systems». *International Conference on Sensors, Devices and Systems. SDaS 2025. Lecture Notes in Electrical Engineering,* 2026. vol 1617. pp 3-12 Springer, Cham.

18. Богаченко, С., Смірнов, О., Буравченко, К., Смірнова, Т., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С. «Обмеження ресурсів мобільних та edge-платформ при розробці програмного забезпечення для реалізації та оптимізації моделей машинного навчання». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2026, т. 2 № 34, С. 677-686.

19. Смірнов, О., Ткачук, Р., Козірова, Н., Константинова, Л., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С.. «Дослідження застосування векторних баз даних у генеративному штучному інтелекті». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 1 № 33, 667-682.

20. Вінтенко Б.Ю., Смірнова Т.В., Миронець І.В., Смірнов О.А., Буравченко К.О. «Метод оцінки функціональної стійкості комп'ютерно-орієнтованих процедур системи підтримки оперативного персоналу АЕС». *Системи управління, навігації та зв'язку,* 2026, вип. 2(84), С. 62-69.

21. Смірнов О., Заріцкий В., Буравченко К., Конопліцька-Слободенюк

О., Константинова Л., Якименко Н., Смірнов С. «Оптимізація розпізнавання облич за допомогою CUDA прискореної бібліотеки dlib». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 4, № 32, С. 573-582.

22. Лисенко, І., Минайленко, Р., Смірнов, С., Буравченко, К., Якименко, Н., Смірнов, О. «Дослідження інструментів штучного інтелекту для інтелектуального аналізу даних». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2025, № 3(31), С. 227–241.

23. Kuznetsov O., Frontoni E., Arnesano M., Smirnov O., Khruskov B. «Biometric Authentication in TinyML: Opportunities and Challenges». *Tiny Machine Learning: Design Principles and Applications*, 2026, pp. 587-633.

24. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

25. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427, 2025.

26. Смірнова, Т.В. «Дослідження методів, моделей та сучасних ІТ-рішень для підтримки технологічних процесів у критичній інфраструктурі держави». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С.195-208, 2025.

27. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

28. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends*

Technologies and Applications, 2025, pp. 193–224.

29. Kuznetsov, O., Smirnov, O., Akhmetov, B., Alimseitova, Z., Imoize, A.L. «Deep Learning Frontiers in Copy-Move Forgery Detection: Advances, Challenges, and Future Directions». *Advancements in Cybersecurity Next Generation Systems and Applications*, 2025. 202-229.

30. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

31. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. Вип. 11(42), ч. II. С.52-62.

32. Смірнов, О.А., Константинова, Л.В., Конопліцька-Слободенюк, О.К., Козірова, Н.В, Якименко, Н.М., Доренський, О.П., Буравченко, К.О. «Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних». *Кібербезпека: освіта, наука, техніка.* 2025. №3(27), С. 429–448.

33. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

34. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings* Volume 3664, 2024, Pages 11-23.

35. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

36. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

37. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

38. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

39. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.

40. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

41. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12.

42. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

43. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol

131. 2023. Springer, Singapore. pp. 21-34.

44. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.

45. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w.

46. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.

47. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

48. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

49. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

50. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

51. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

52. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

53. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

54. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

55. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

56. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019.

57. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

58. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

59. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S.,

Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

60. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

61. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

62. Smirnov, S., Bulekbaeva, G., Kikvidze, O.G., Lakhno, V., Brzhanov, R., Tabylov, A. «Computer simulation in the MathCAD package of plastic deformation of the deposited layer on the flat surface of the part». Journal of Theoretical and Applied Information Technology Volume 97, Issue 20, 2019, Pages 2467-2484.

63. Smirnov A.A., Kuzne tsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

64. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95

65. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв’язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.