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 першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів”

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Іванченко Д.А.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
«____» _____ 2026 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Освітній ступінь *бакалавр*
Галузь знань ._12 _____*“Інформаційні технології”*_____
Спеціальність 122 *“Комп’ютерні науки”*
Освітньо-професійна (освітньо-наукова) програма _*“Комп’ютерні науки”*_

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

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Іванченку Дмитру Андрійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів*

2. Керівник роботи *Лисенко Ірина Анатоліївна, канд. техн. наук*
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою роботи є розробка програмного забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів*

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

1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію.
6. Висновки

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

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

7. Дата видачі завдання « 06 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 06 » лютого 2026 р.

Підпис керівника
_____ Лисенко І.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 06 » лютого 2026 р.

Підпис здобувача
_____ Іванченко Д.А.
(прізвище та ініціали)

АНОТАЦІЯ

Іванченко Д.А. Програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

Метою розробки є програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

Результат роботи – програмна реалізація системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

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 Python.

Ключові слова: комп'ютерні науки, інтелектуальний моніторинг, телеметрія, ЦОД, прогнозування навантажень, оптимізація ресурсів

ABSTRACT

Ivanchenko D.A. Software for the intelligent monitoring system of data center telemetry with load forecasting and resource optimization. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the intelligent monitoring system of data center telemetry with load forecasting and resource optimization.

The purpose of the development is the software for the intelligent monitoring system of data center telemetry with load forecasting and resource optimization.

The result of the work is the software implementation of the intelligent monitoring system of data center telemetry with load forecasting and resource optimization.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on PCs with Windows 10/11.

The program is developed in the Python environment.

Keywords: computer science, intelligent monitoring, telemetry, data center, load forecasting, resource optimization

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ААУ	–	автономне адаптивне управління
БД	–	блок датчиків
БЗ	–	база знань
БОС	–	блок оцінки стану
БПР	–	блок прийняття рішень
ГІП	–	графічний інтерфейс користувача (GUI)
ВО	–	виконавчий орган
ЕОМ	–	електрона обчислювальна машина
КА	–	кінцевий автомат
НРС	–	недетермінований автомат Рабина-Скотта
НМ	–	нейронна мережа
МНРС	–	модифікований недетермінований автомат Рабина-Скотта
НАНУ	–	Національна академія наук України
ОУ	–	об'єкт управління
ПЧО	–	просторово-часовий образ
ВВ	–	випадкова величина
ПЗ	–	програмне забезпечення
СПДНМ	–	система побудови й дослідження нейронних мереж
УС	–	управляюча система
ФР	–	функція розподілу
ФРО	–	апарат формування й розпізнавання образів
Z^+	–	множина ненегативних цілих чисел
$G(V, N)$	–	граф із множиною вершин V і множиною ребер N
$i \to j$	–	ребро, спрямоване з вершини i у вершину j
$X \leftrightarrow Y$	–	взаємооднозначне відображення множини X на множину Y
$\pi(X)$	–	множина кінцевих підмножин множини X

$R[a,b]$	–	множина речовинних чисел на $[a,b]$, $R \equiv R[-\infty,+\infty]$
B^N	–	простір двійкових векторів розмірності N
Λ	–	порожнє слово із множини вхідних слів КА
0	–	*неправда* у вираженні тризначної логіки
1	–	*істина* у вираженні тризначної логіки
$\lozenge$	–	*невизначеність* у вираженні тризначної логіки
$\vec{X} \subset \vec{Y}$	–	$\vec{X}$ є підвектор (сукупність обраних компонентів) вектора $\vec{Y}$
$X \supset Y$	–	клас Y є нащадком класу X

ВСТУП

Актуальність теми. Традиційний моніторинг, як ця галузь розуміла його десятиліттями, фактично застарів. Статичні інформаційні панелі, сповіщення на основі порогових значень та ізольовані інструменти моніторингу народилися у світі передбачуваної інфраструктури та досить простих архітектур додатків.

Але в сучасному цифровому середовищі їм важко впоратися з масштабом, швидкістю та складністю сучасних систем. Хмарні платформи, розподілені мікросервіси та постійно доступний цифровий досвід значно розширили традиційну модель моніторингу. З настанням 2026 року організації вступають в еру інтелектуального спостереження. Згідно з цією парадигмою, системи не просто повідомляють про збій. Швидше, вони допомагають командам зрозуміти мотив, що стоїть за цим, як це поширюється на різні служби та які наступні кроки для пом'якшення або навіть запобігання впливу.

Спостережуваність більше не стосується того, як людина реагує після невдачі; вона стосується постійного розуміння та превентивних дій. Цей зсув не є поступовим – він є фундаментальним. Історично моніторинг відповідав лише на ті питання, які, як ми знали, ставили: «Чи працює сервер?» Або «Чи перевищує використання процесора певний поріг?» Однак завдяки спостережуваності команди зможуть ставити нові питання, яких вони раніше не очікували. Це надає глибини та контексту, необхідних для осмислення та розуміння розподілених систем, складних з точки зору принципів їхньої роботи. До 2026 року це розмежування не буде академічним. Воно забезпечить чітке визначення того, які організації можуть безпечно працювати у великих масштабах, а які продовжують страждати від збоїв, проблем з продуктивністю та зростання операційних витрат.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

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

– Огляд існуючих систем інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

– Дослідження системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

– Програмна реалізація системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Кінцева мета спостережуваності полягає у визначенні внутрішнього стану системи на основі інформації, яку вона вам надає, в основі її існування. Замість перевірки та пасивного видання статичних сигналів для перевірок та виходів, спостережуваність дозволяє інженерам та операторам бачити на льоту, як система поводиться під час роботи, особливо коли виникають нові режими відмов. Класичне розуміння даних спостережуваності складається з трьох фундаментальних стовпів:

– Журнали: Багатий звіт про інформацію та події, що детально описує діяльність, що відбулася в програмі або системі в певний момент часу.

– Метрики: агреговані числові вимірювання з плином часу для звітування про тенденції, продуктивність та потужність.

– Трасування: Повне уявлення про запити під час їх переміщення через розподілену систему, що виявляє залежності, затримки та вузькі місця.

Хоча ці основи все ще допомагають нам, зараз вони є лише основою фундаменту. В основі сучасних середовищ лежать мікросервіси, хмарно-орієнтовані архітектури, Kubernetes та дедалі генеративніші системи штучного інтелекту та агентні системи. Вони вносять такі крайні динамічність у ці технології. Сервіси динамічно масштабуються самопідтримуваним чином, залежності змінюються постійно, розгортання відбуваються швидкими темпами, а збої поширюються нелінійно по всій області.

Таким чином, спостережуваність перетворилася з діагностичної можливості на стратегічну. Спостережуваність перетворилася на місток між технічною продуктивністю та користувацьким досвідом, бізнес-ефективністю, фінансовою ефективністю та регулюванням сьогодні. Вона перетворилася на важливу основу для цифрової довіри та операційної досконалості.

1.2 Область застосування

Інтелектуальна спостережуваність стосується еволюції платформ спостережуваності, які не лише збирають та співвідносять телеметричні дані, але й застосовують міркування на основі штучного інтелекту для прогнозування збоїв, рекомендацій щодо дій та автоматичного усунення інцидентів. У 2026 році системи спостережуваності активно беруть участь в операціях, а не пасивно повідомляють про проблеми.

До 2026 року штучний інтелект у спостережуваності перейде від другого пілота для підтримки рішень до автономного операційного агента. Ці системи співвідноситимуть журнали, метрики, трасування, топологію та дані про зміни, щоб виявляти першопричини та виконувати дії з усунення несправностей без втручання людини.

Сучасні платформи спостережуваності використовують інтелектуальну фільтрацію телеметрії, адаптивну вибірку та багаторівневе сховище для зменшення непотрібного споживання даних. Це гарантує, що організації платять за практичну аналітику, а не за обсяг необроблених даних, узгоджуючи інвестиції у спостережуваність з фінансовими операціями та бізнес-пріоритетами.

У складних розподілених системах збої неминучі. Стійкість вимірює, наскільки добре система витримує збої, обмежує радіус вибуху, поступово деградує та швидко відновлюється. До 2026 року стійкість, а не лише час безвідмовної роботи, стане справжнім показником операційної досконалості.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Цифрова трансформація за підтримкою HPE

Цифрова трансформація – процес непростої. Для неї необхідна стратегія, перегляд бізнес-моделей і процесів, нова інфраструктура, нове програмне забезпечення, оптимізація набору послуг, ефективні механізми впровадження, програми навчання й надійна поточна підтримка. Все це може надати компанія HPE. Наш портфель пропозицій включає хмарні рішення, засоби забезпечення безпеки, технології Інтернету речей (IoT), технології мобільного доступу й рішення інфраструктури. HPE Pointnext – це досвід, необхідний для гарантованого успіху трансформації.

Усунення проблеми поганої якості планування ресурсів

Іноді важко визначити, які ресурси й у яких обсягах будуть потрібні для успішного впровадження цифрових технологій у майбутньому. На щастя, компанія HPE пропонує гнучкі рішення, здатні в лічені хвилини оптимізувати плани, зв’язані с ресурсами.

Оплата тільки за фактично використовувані послуги

Фахівці HPE допоможуть вам у переході на модель «IT як послуга» і до компонуємої інфраструктури, що зменшує складність операцій IT-відділів і знижує вартість володіння.

Інфраструктура на базі інтелектуальних технологій

Без сумнівів, штучний інтелект є невід'ємною частиною цифрового підприємства майбутнього. Компанія HPE включає штучний інтелект у процеси

аналізу даних і експлуатаційної підтримки, завдяки чому інфраструктура обслуговує сама себе.

Переваги Інтернету речей

Компанія HPE успішно розробляє інтелектуальні технології, інтелектуальні простори й гібридні IT-рішення, які підвищують ефективність, прибутковість і конкурентоспроможність підприємств. Завдяки нашої допомоги ваш бізнес перетворить швидше й простіше, ніж ви могли собі представити.

Хмарна платформа предиктивного аналізу InfoSight

Хмарна платформа предиктивного аналізу InfoSight, що компанія HPE придбала разом з виробником твердотільних систем зберігання Nimble Storage, дозволяє виявляти й прогнозувати проблеми в IT-інфраструктурі.

InfoSight накопичує й аналізує дані, що надходять від датчиків, які встановлені в інформаційних системах більш ніж 10 тис. замовників Nimble Storage, і обробляючи понад 1 млн події в секунду. Система повинна швидко виявити й ідентифікувати причину виниклої проблеми, знайти шляхи її усунення на основі аналізу поточної ситуації з використанням усього масиву накопичених даних.

У базі даних InfoSight утримуються відомості, зібрані протягом декількох років. Прогнозна аналітика реалізується хмарним ПЗ компанії VoltDB, математичний апарат якого базується на авторегресійних моделях прогнозування й методі Монте-Карло. Це дозволяє передбачати настання подій певного роду – приміром, дефіциту ємності накопичувачів масиву зберігання або перевищення пропускної здатності системи вводу-виводу ЦОД.

В InfoSight застосовуються технології машинного навчання й аналізу значних обсягів інформації. З початку 2025 року ємність бази даних, використовуваних для навчання, значно зросла, тому що тепер сфера дії платформи поширюється й на масиви зберігання 3PAR StoreServ, оснащених операційною системою 3PAR OS 3.3.1.

У першій версії HPE InfoSight для масивів HPE 3PAR реалізований ряд цікавих функцій: аналіз стеків віртуальних машин; виявлення причин, що викликають зниження продуктивності віртуальних машин, які використовують ресурси систем зберігання HPE 3PAR; виявлення віртуальних машин, що створюють проблеми для інших робочих навантажень ЦОД.

У ході використання InfoSight удалося встановити, що у випадку бізнес-додатків значна частина проблем із продуктивністю операцій вводу-виводу не зв'язана безпосередньо з масивами зберігання. Зіставлення даних, отриманих InfoSight, з іншою діагностичною інформацією дозволяє виявити вузькі місця на всьому шляху проходження даних – від віртуальних машин до накопичувачів масивів зберігання.

Платформа постійно навчається, підкреслюють в HPE. У міру подальшої інтеграції InfoSight із продуктами й рішеннями HPE алгоритми штучного інтелекту одержать доступ до всі зростаючим обсягам даних.

Функціональність InfoSight може стати основою для автономних ЦОД, у яких конфігурування й системні настроювання виконуються без втручання експлуатаційного персоналу.

У компанії Tegile Systems, випускаючий гібридні ЦОД, використовують хмарне аналітичне ПЗ інтелектуального керування IntelliCare Cloud Analytics, для моніторингу завантаження ємності масивів зберігання, стану їхніх конфігурацій, «здоров'я» і продуктивності.

Хмарна система, що не вимагає установки серверних агентів, збирає дані з декількох тисяч масивів зберігання, установлених у замовників. Їхній аналіз дозволяє прогнозувати відмови компонентів, а також виникнення проблем із продуктивністю й ресурсами масивів.

DeepMind

Google застосовує технології штучного інтелекту у своїх центрах обробки даних для зниження енерговитрат, які виміряються мільйонами мегават-годинника в рік, а виходить, і для скорочення витрат.

Для цього в 2014 році була придбана компанія DeepMind, де створені програмний комплекс AlphaGo, що переміг професійного гравця в го, а також нейронна мережа, здатна навчатися відеоіграм. ПЗ DeepMind, розроблене для ЦОД Google, визначає оптимальні режими роботи системи охолодження й забезпечує керування устаткуванням у режимі реального часу.

Програмний комплекс DeepMind відслідковує більше сотні різних параметрів, що характеризують стан інфраструктури ЦОД або роблять на неї вплив. У їхньому числі – показники навантаження IT-устаткування й швидкості обертання вентиляторів у стійках, відомості про кондиціонери, градирні й теплообмінники, інформація про погоду й стан вікон у приміщеннях.

У процесі розробки цього ПЗ враховувалися дані про роботу ЦОД Google протягом декількох попереднього років. Алгоритми аналізу інформації й керування на основі отриманих виводів здатні до самостійного навчання, що дозволяє їм працювати в невизначені заздалегідь умовах.

У ЦОД установлена безліч датчиків, тому порівняльний аналіз поточних даних і гігантських масивів накопиченої «історичної» інформації, а також використання алгоритмів прогнозування нейронних мереж дозволяють ураховувати найменші нюанси постійно мінливої кліматичної обстановки в машинних залах. У результаті, як затверджують розроблювачі DeepMind, їм вдається пророчити очікуване значення PUE з точністю 99,6%.

Завдяки використанню рішень DeepMind на 40% скоротилися енерговитрати систем охолодження центрів обробки даних, у яких в основному встановлене промислове устаткування. Внесок саме цих систем в енергоспоживання інженерного комплексу ЦОД досить значний, і оптимізація

Для підтримки необхідної температури в його машинних залах використовується технологія прямого охолодження атмосферним повітрям, що дозволяє кліматичній системі працювати в режимі фрікулінгу більше 330 днів у році.

За керування відповідає програмне забезпечення з компонентами штучного інтелекту. На основі даних про температуру, погоду й відомостей

синоптиків це ПЗ оптимізує охолодження устаткування ЦОД і оцінює необхідність формування запасів холоду при прогнозованому потеплінні. Як очікується, середньорічне значення PUE у цьому ЦОД складе 1,3.

LitBit

У стартапі LitBit працюють над створенням цифрових помічників, наділених штучним інтелектом, яких у компанії називають штучними особистостями (artificial personae). Після навчання вони здобувають певні навички й здатні спростити роботу різних фахівців, зокрема, можуть здійснювати автоматизоване керування різноманітними операціями в центрах обробки даних.

Для їхнього навчання використовується розроблений в LitBit інтерфейс, що допомагає спілкуватися з artificial personae і постачати їхніми необхідними відомостями. Навчанням може займатися будь-який співробітник ЦОД, для цього не потрібне знання тонкощів програмування, спеціалізованих алгоритмів і інших подібних речей.

Розроблювачі пояснюють, що на базі їх рішень можуть створюватися цифрові помічники, що володіють інженерними навичками, які допоможуть виявити аномалії в роботі ЦОД і запобігти виникненню небезпечних ситуацій. Затверджується, що такі помічники мають функціональність, що дозволяє супроводжувати й доповнювати дії експлуатаційного персоналу, а в певних ситуаціях і заміняти деяких фахівців.

Програмні «персони» Litbit одержують відомості про зовнішній світ за допомогою технологій інфрачервоного зору, акустичного контролю й обробки звукової інформації. Накопичені первинні дані використовуються для подальшої обробки.

Приміром, цифрових помічників можна навчити прогнозувати й потім запобігати збої в роботі серверів і мережного устаткування. Для цього вони повинні аналізувати звуки, видавані встановленими в стійках джерелами живлення, і порівнювати отримані результати з акустичними шаблонами, що характеризують стабільні режими роботи цих пристроїв. А характеристики звуків і вібрацій фальшполу й стійок надають можливість оцінити стан апаратних

систем ЦОД і спрогнозувати деградацію їхньої продуктивності у випадку перевищення граничних значень досліджуваних параметрів.

Завдяки високій продуктивності «штучних особистостей», здатних обробляти сотні тисяч параметрів у секунду, з'являється можливість узгоджено управляти в реальному часі різними системами центрів обробки даних, а також накопичувати отримані дані й використовувати їх для наступного аналізу.

Цифрові помічники, пояснюють в LitBit, здатні взаємодіяти зі сторонніми службами, відправляти повідомлення через корпоративний мессенджер Slack або виконувати голосові команди Amazon Echo. Вони здійснюють онлайн-обробку природної мови й підтримують когнітивні обчислення.

Наприкінці 2025-го почалося впровадження технологій LitBit у канадській компанії ROOT Data Center, що надає послуги оренди площ у двох центрах обробки даних у Монреалі. Там мають намір досліджувати можливості «персон» LitBit для забезпечення безвідмовної роботи ЦОД.

На першому етапі планується застосовувати штучний інтелект помічника з іменем Алекс для моніторингу роботи дизель-генераторів. Установлені в корпусах дизель-генераторів мікрофони здатні вловлювати звукові відхилення від нормальних режимів роботи, що дозволяє прогнозувати подальше поводження цих систем і передбачати потенційні відмови.

Як очікується, за рахунок формування й навчання різних інженерних помічників експлуатаційного персоналу вдасться підвищити ефективність роботи ROOT Data Center.

Технологію LitBit мають намір використовувати й в CBRE Group. Ця найбільша у світі компанія зі штаб-квартирою в Лос-Анджелесі робить послуги в сфері комерційної нерухомості. Її фахівці приступилися до навчання цифрового помічника, якого назвали Remi (Risk Exposure Mitigation Intelligence): він повинен освоїти стандартні режими функціонування устаткування, установленого в 800 центрах обробки даних, розташованих у різних країнах миру.

Таке навчання, думають в CBRE Group, дозволить сконцентрувати досвід експлуатаційного персоналу всіх ЦОД у єдиній базі знань, доступ до якої (у тому

числі за допомогою мобільних пристроїв) одержить кожний фахівець служби експлуатації.

Wave2Wave

В американській компанії Wave2Wave розроблена роботизована система комутації волоконно-оптичних кабелів, що автоматизує формування перехресних з'єднань у вузлах обміну трафіком (meet-me room) центрів обробки даних.

Такі вузли (їх називають також пірінговими) забезпечують не тільки підключення ЦОД і орендарів їхніх сервісів до каналів зв'язку телекомунікаційних компаній, але й комутацію таких з'єднань, у тому числі для взаємодії з новими провайдерами послуг зв'язку й безпосереднього обміну даними між різними орендарями сервісів ЦОД.

Створена в Wave2Wave роботизована платформа перемикання оптичних з'єднань (Robotic Optical Management Engine, ROME) за кілька секунд здійснює оптичну крос-комутацію без участі людини.

Ця платформа розширює функціональність програмного керування мережною інфраструктурою, поширюючи дію методів SDN (Software-Defined Networking) на її нижній, фізичний рівень.

Пристрої ROME, розташовувані в 19-дюймових стійках, «прозорі» стосовно протоколів і швидкості передачі трафіка. Вузол їхнього логічного керування (Logical Control Unit) установлюється в шасі форм-фактора 1RU і працює на базі операційної системи реального часу й створеного для ROME програмного забезпечення.

Керування механічними компонентами ROME здійснюється контролерами Robotic Control Unit. Вони використовують системи із сервокеруванням, які переміщають із точністю до одного мікрометра дві роботизовані «руки», що захоплюють оптичні кабелі.

Основні моделі оптичних кросів Wave2Wave – ROME 250 і ROME 500 на 256 і 512 волоконно-оптичних з'єднань відповідно. Вони випускаються в

модифікаціях з одномодовими й багатомодовими (OM4) волоконно-оптичними кабелями.

Ці автоматизовані комутаційні панелі займають разом з вузлом логічного керування простір 11RU у стандартній стійці. За даними розроблювачів, установлені з'єднання зберігаються навіть при відключенні живлення пристроїв ROME.

У компанії Wave2Wave вважають, що робототехніка й штучний інтелект можуть застосовуватися для автоматизації керування різними системами центрів обробки даних. Приміром, ROME можна використовувати для зв'язку серверів і встановлюваних у стійках комутаторів top-of-rack, а також розміщати їх у різних місцях мережі ЦОД з дистанційним централізованим керуванням.

Romonet

Машинне навчання відкриває нові можливості в сфері керування центрами обробки даних. Аналіз величезного числа змінних і облік безлічі одночасно діючих факторів всі частіше виявляються нездійсненним завданням навіть для висококваліфікованих фахівців, що володіють значним досвідом експлуатації ЦОД. У британській компанії Romonet створили хмарну платформу прогностичної аналітики, що здатна аналізувати вартість активів ЦОД, оцінювати вплив на неї інфраструктурних змін і прогнозувати сумарну вартість володіння ЦОД (Total Cost of Ownership, TCO). Точність роботи прогностичної моделі Romonet, за даними компанії, досягає 97%. Згідно з опублікованими даними, в Intel застосовують технології Romonet, щоб продемонструвати переваги процесорів, здатних працювати при підвищеній температурі. Установка таких процесорів у серверах знижує TCO ЦОД. Romonet впроваджує у свою платформу технології машинного навчання й моделювання, щоб, використовуючи накопичені за кілька років дані й предиктивну аналітику, одержувати усе більше точні результати. Моделювання, за даними компанії, може здійснюватися без установки численних датчиків на підставі відомостей, що втримуються в проектній документації ЦОД.

Coolan

У каліфорнійському стартапі Coolan (компанія входить до складу Salesforce. com), створеному одним із засновників Facebook Open Compute Project, вирішили за допомогою машинного навчання й предиктивної аналітики підвищити рівень готовності центрів обробки даних, знизити число відмов і пов'язаних з ними простоїв. Розроблене в компанії рішення збирає, агрегує і зберігає дані, які характеризують стан серверів замовників, режими роботи системи електропостачання й т.п. Видавані системою рекомендації про необхідність заміни серверів і інших мір дозволяють більш точно управляти інфраструктурою. Salesforce придбала Coolan для підтримки програми уніфікації своїх ЦОД, що передбачає скорочення типів серверів і підвищення рівня автоматизації їхньої експлуатації.

Vigilent

У компанії Vigilent із Кремнієвої долини використовують штучний інтелект і машинне навчання для оптимізації в реальному часі температурних режимів у ЦОД і серверних приміщеннях. Навчання систем Vigilent починається з моменту їхнього запровадження в дію й триває в процесі подальшої експлуатації. Збір даних про температуру для ПЗ динамічного керування системою охолодження Vigilent Dynamic Cooling Management здійснюється ніздрюватою мережею бездротових датчиків, установлених у різних крапках машинних залів. Найважливішим компонентом цього ПЗ є модуль DCIM Toolkit.

У компанії пояснюють, що запропоновані її розроблювачами рішення здатні оптимізувати розподіл тепла в ЦОД, усунути крапки перегріву, підвищити утилизацію охолодного устаткування, до 40% потужності якого в середньому витрачається нераціонально. Рішення Vigilent, оптимизуюче співвідношення потужності охолодження й реальної IT-навантаження, використовується разом із системою Siemens Demand Flow для керування роботою холодильних машин, установлених у ЦОД. Крім того, компанія Schneider Electric уклала партнерську

угоду з Vigilent, щоб вмонтувати модуль Cooling Optimize у комплекс керування інфраструктурою ЦОД StruxureWare for Data Centers.

Mindi Technologies

У компанії Mindi Technologies, зареєстрованої два роки тому у Великобританії, розробляють систему Autopilot на основі технологій штучного інтелекту для прогнозування різного роду позаштатних подій у ЦОД, включаючи відмови IT- і інженерних комплексів, збої системного ПЗ, проблеми з енергопостачанням, інциденти безпеки. Спочатку фахівці Mindi Technologies мають намір зайнятися інтелектуальним балансуванням розподілу IT-ресурсів серверів, що одночасно підтримують трохи робітників навантажень, що дасть можливість прогнозувати потреби в ресурсах і забезпечити стабільність роботи додатків. У подальших планах – оптимізація завантаження серверів, що дозволяє скоротити число фізичних машин у ЦОД.

Amadeus IT Group

Amadeus IT Group, постачальник IT-сервісів для галузі авіаперевезень, має намір застосувати штучний інтелект платформи IBM Watson для моніторингу інфраструктури центра обробки даних, у якому встановлене більше 10 тис. серверів. У компанії стурбовані неефективністю ручного моніторингу непередбачених ситуацій і регулярних утруднень при усуненні проблем в умовах постійного розширення спектра розв'язуваних завдань і підвищення їхньої складності. Тому в Amadeus IT Group прагнуть максимально автоматизувати керування, а також використовувати штучний інтелект для складання прогнозів і своєчасного виправлення неполадок без залучення персоналу.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до

об’єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python https://www.python.org/ і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для налаштовуваних програм.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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

У цьому розділі проаналізуємо п'ять основних прогнозів спостережливості на 2026 рік.

1. Штучний інтелект перетворюється з союзника-копілота на автономного оперативного агента.

2. Інтелектуальна фільтрація телеметрії та пріоритизація даних для досягнення оптимізації витрат.

3. OpenTelemetry стає беззаперечним галузевим стандартом.

4. Спостережуваність поширюється не лише на інженерію, а й на бізнес та відповідність вимогам.

5. Стійкість робить доступність новим операційним показником.

Прогноз 1: ШІ еволюціонує від другого пілота до автономного агента

Сьогодні більшість платформ спостереження використовують штучний інтелект обмеженими, але важливими способами. Виявлення аномалій допомагає нам виявляти незвичайну поведінку. Розпізнавання шаблонів виявляє кореляції між метриками та журналами. Пропозиції щодо першопричин дозволяють інженерам зосередитися на потенційних проблемах.

Ці можливості функціонують як другий пілот, допомагаючи операторам-людям, водночас покладаючись на них у розслідуванні інцидентів та вживанні коригувальних заходів.

Ця модель буде значно розвинена у 2026 році. Штучний інтелект перетворить другого пілота на автономного агента. Ці агенти відстежуватимуть інциденти та співвідноситимуть сигнали з журналів, метрик, трас, даних топології та подій змін, визначаючи ймовірні тригери з мінімальною участю людини.

Що ще важливіше, вони зможуть автоматично запускати задані заходи щодо виправлення (перезапуск служб, відкат розгортань, масштабування ресурсів тощо). Ця адаптація зумовлена двома ключовими факторами:

– Дедалі складніші сучасні системи, які дедалі більше перевищують когнітивні межі людини.

– Розвиток генеративного та агентного штучного інтелекту, щоб системи могли самостійно міркувати, планувати та виконувати багатоетапні дії.

У інженерних та операційних командах ми побачимо глибокі наслідки:

– Середній час вирішення (MTTR) значно зменшиться, оскільки інциденти вирішуватимуться в режимі реального часу.

– Зменшення втоми під час чергування, оскільки менше сповіщень потребують втручання людини.

– Операції SRE та DevOps перейдуть від реактивного гасіння пожеж до проактивної інженерії надійності. Замість того, щоб реагувати на кожне сповіщення, інженери створюватимуть захисні бар'єри, розгортатимуть політики та визначатимуть межі довіри, які забезпечують експлуатаційну поведінку автономних систем. Людський досвід буде рушійною силою ключових знань там, де це найважливіше: проектування архітектури, планування стійкості та постійний розвиток.

Автономне реагування на інциденти стане очікуванням за замовчуванням, а не конкурентною перевагою. Ті організації, які не застосують цю модель, не зможуть встигати за ними.

Прогноз 2: Оптимізація витрат стимулює інтелектуальну фільтрацію даних.

Обсяги телеметричних даних зростають з нестійкою швидкістю. Кожен виклик мікросервісу, запит API, виведення на основі штучного інтелекту та фоновий процес створюють журнали, метрики та трасування.

Зокрема, агентні системи штучного інтелекту створюють величезну кількість телеметрії з високою кардинальністю, яка швидко перевантажує

звичайні конвеєри спостереження.

Давня практика «збирати все та аналізувати потім» більше не є розумною. Витрати на зберігання, прийом даних та запити продовжують зростати, часто швидше, ніж інвестиції в інфраструктуру. У 2026 році організації вперше зіткнуться зі справжньою вартістю спостережуваних даних у великих масштабах.

Галузь змінить свій підхід, орієнтований на цінність даних, зосереджуючись на цінності даних, а не на їхньому обсязі. Замість того, щоб без розбору використовувати всю телеметрію, платформи спостереження використовуватимуть такі стратегії, як адаптивна телеметрія, яка включатиме наступне:

– Інтелектуальна вибірка на основі стану системи та впливу на бізнес.

– Фільтрація перед завантаженням для видалення надлишкових та малоцінних даних.

– Динамічне маршрутизування критично важливої телеметрії до високопродуктивного сховища, а менш критичних даних – до менш витратних рівнів.

Це дозволяє організаціям платити за аналітичні дані, а не за шум, а інвестиції в спостережуваність безпосередньо корелюють з бізнес-пріоритетами.

З простого центру витрат ці платформи спостереження стануть інструментом отримання прибутку, який все більше узгоджуватиметься з практиками FinOps та методологіями оптимізації витрат.

Прогноз 3: OpenTelemetry стає беззаперечним стандартом.

Ця технологія зазнала експоненціального розвитку протягом останніх кількох років. У міру переходу компаній від повсюдного впровадження, до 2026 року вона перетвориться з універсального впровадження на фактично необхідний стандарт для телеметричних приладів у хмарах, серед постачальників та на різних платформах.

Зі зростанням популярності гібридних та мультихмарних стратегій в організаціях, власні інструменти будуть сприйматися як зобов'язання. Це

призведе до гнучкості, сумісності та довгострокової стійкості, створюючи потребу в стандартизації.

Прогноз 4: Спостережуваність поширюється на бізнес та дотримання вимог

Традиційно спостережуваність вважалася інженерною дисципліною, яка займалася лише часом безвідмовної роботи та продуктивністю. До 2027 року ця межа зникне. Дані телеметрії все частіше використовуватимуть для відстеження цілей рівня обслуговування (SLO), що відповідають бізнес-потребам:

– Затримка конверсії.

– Показники успішного оформлення замовлення.

– Час відповіді на пошук.

– Пороги відмов користувачів.

Водночас, зростаючі регуляторні вимоги до управління штучним інтелектом, конфіденційності даних та операційної прозорості вимагатимуть глибшого спостереження за системами, конвеєрами та моделями.

Для керівників вищої ланки та керівництва спостережливість стає стратегічним активом для прийняття рішень, який спрямовуватиме:

– Пріоритезація спринтів та планування дорожньої карти.

– Рішення щодо інвестицій в інфраструктуру та платформу.

– Розподіл бюджету та обґрунтування витрат.

– Управління ризиками та звітність про дотримання вимог.

Дані спостереження також будуть ключовими для спостереження, аудиту та відповідності моделі ШІ нормативним вимогам.

Замість вимірювання технічних результатів, надійність стає рушійною силою бізнес-результатів.

Прогноз 5: Стійкість замінює доступність як головний показник.

У розподілених, хмарних системах 100% доступність – це міф. Залежності збиваються, регіони страждають від збоїв, а сторонні сервіси створюють неминучий ризик. Доступність більше не є єдиним визначенням операційної

досконалості в цьому контексті.

До 2026 року найкращим показником успіху буде стійкість – здатність системи протистояти збоям, коректно деградувати та швидко відновлюватися, забезпечуючи при цьому хороший користувацький досвід.

Цей зсув призведе до глибших інвестицій у:

– Хаотична інженерія та тестування на впровадження відмов.

– Надійна ізоляція доменів відмов за допомогою розширеного картування залежностей для розуміння радіуса вибуху та каскадних відмов.

Спостережуваність, зосереджена на стійкості, підкреслює, як системи поводяться в умовах стресу, а не лише під час нормальної роботи.

Надійність, безпека (SecOps) та спостережуваність об'єднаються в єдину, уніфіковану операційну вимогу.

3.2 Розробка структурної схеми

Цифрова трансформація, про яку останнім часом стільки говорять, у значній мірі опирається на інфраструктуру центрів обробки даних. Однак для її успішного розгортання самі ЦОД також повинні мінятися. Ефективними технологічними інструментами їхнього розвитку й удосконалювання стають штучний інтелект, методи машинного навчання й Великі Дані.

Галузь центрів обробки даних стала невід'ємною реальністю миру сучасних інформаційних технологій. Цей сегмент глобального IT-ринку успішно розвивається й удосконалюється. В 2025 році, за даними компанії 451 Research, число комерційних ЦОД, що надають роздрібні й оптові послуги оренди (colocation) у різних країнах миру, наблизилося до п'яти тисяч.

Зі збільшенням числа ЦОД усе більше серйозним завданням стає підвищення рівня їхньої готовності, скорочення відмов і збоїв інженерного устаткування. Відповідно до аналітичних досліджень, збитки, які несуть

підприємства в результаті виникнення позаштатних ситуацій, обчислюються сотнями мільйонів доларів.

На ринку центрів обробки даних, як, втім, і в інших областях, затверджують експерти, процвітають у першу чергу ті компанії, яким у пошуках шляхів росту вдається вийти за рамки традиційних технологічних і організаційних рішень і знайти спосіб сполучити найчастіше суперечливі вимоги. Тому застосування новітніх технологій для підвищення ефективності, відказостійкості й зниження експлуатаційних витрат ЦОД буде розширюватися.

Штучний інтелект у ЦОД

Штучний інтелект наділяє машини здатністю виконувати певні інтелектуальні дії, на які раніше були можуть тільки люди. Завдяки застосуванню новітніх технологій, у тому числі подій, що відбуваються, що забезпечують візуальне сприйняття, і розпізнавання мови, такі машини можуть збирати й обробляти дані, а також інтерпретувати отримані результати для автоматичного прийняття операційних рішень.

У бізнесі, приміром, штучний інтелект може використовуватися для прогнозування – на основі даних з різних джерел – наслідків прийняття стратегічних бізнес-рішень і, таким чином, дозволяє оцінити реакцію ринку. У промисловості й військовій справі він може застосовуватися для організації взаємодії численних підключених до мережі пристроїв, включаючи дрони й роботи, у процесі спільного виконання ними колективних завдань.

Власники сучасних центрів обробки даних теж використовують штучний інтелект і машинне навчання (у всякому разі, намагаються робити це) для рішення цілком конкретних завдань. Такими завданнями є, у числі іншого, скорочення часу простою за рахунок прогнозування ризиків виникнення позаштатних ситуацій, оптимізація режимів роботи серверів і систем зберігання даних, зниження енерговитрат, підвищення ефективності комплексів охолодження й оптимізація температурних режимів у машинних залах, раціональне використання ресурсів експлуатаційного персоналу.

Компоненти штучного інтелекту можуть бути убудовані й у комерційно доступні платформи керування інфраструктурою ЦОД (Data Center Infrastructure Management, DCIM), що дозволяє спростити рішення деяких з перерахованих завдань і скоротити обсяг операційних витрат.

Застосування штучного інтелекту в ЦОД починає приносити плоди.

Відповідно до досліджень компанії Accenture, завдяки використанню штучного інтелекту, в 2035 році ріст економіки може скласти 14 трлн доларів, а показники рентабельності у всіх сферах господарської діяльності підвищаться в середньому на 38%.

В Gartner уважають, що майже третина центрів обробки даних, спроби яких впровадити штучний інтелект і машинне навчання виявляться безрезультатними, до 2027 року виявляться економічно неефективними.

У світлі подібних прогнозів впровадження штучного інтелекту й інших новітніх технологій для автоматизації процесів у галузі ЦОД є перспективним і затребуваним рішенням.

З ростом масштабів ЦОД, як ми вже відзначали, багато роблем уже неможливо вирішити вручну. «ЦОДобудування» іде по шляху створення програмноуправляемих, самооптимізизуючихся й тих, що самовідновлюються, центрів обробки даних, персонал яких звільняється від багатьох рутинних операцій. Внаслідок автоматизації, роботизації й стандартизації постійна присутність у машинних залах висококваліфікованих фахівців служб експлуатації ЦОД стає необов'язковим, і цілком природно виникає питання про можливе скорочення персоналу.

Однак численні дослідження аналітиків, опитування керівників IT-підприємств і центрів обробки даних свідчать про те, що кваліфікованих фахівців, що працюють в області інформаційних технологій, не вистачає. Тому, що вивільняються в ході автоматизації співробітники, не залишаться без справи, але їм, можливо, прийдеться одержувати нові знання.

В PricewaterhouseCoopers, приміром, уважають, що протягом найближчих п'яти років потрібно буде набагато більше фахівців в області аналітики й робототехніки.

Штучний інтелект і машинне навчання здатні значно підвищити ефективність центрів обробки даних. Сьогодні галузь ЦОД перебуває на самому початку шляху їхнього впровадження, зміни парадигми керування й розподіли завдань експлуатації між людьми й комп'ютерами.

Сьогодні за нашими оцінками більше 60% найбільших світових корпорацій уже працюють над своєю стратегією digital-трансформації, до того ж, за останній рік ми спостерігаємо великий ріст кількості запитів до нас серед українських компаній.

Бізнес в Україні традиційно прагне до високих технологій і проривних стратегій, але поки це втілюється в малих масштабах. Багато хто плутаються з термінологією, сприймаючи цифрову трансформацію як «знання якою технологією або сервісом замінити конкретний бізнес-процес у компанії» – я називаю цей підхід «клаптева автоматизація». Але основною проблемою, що ми зустрічаємо, консультуючи наших клієнтів, є відсутність відповідних кадрів – поки не вистачає людей, що володіють певними компетенціями й експертизою в цій області. Багато хто щось роблять, але не зрозуміло для чого стратегічно, немає розуміння із чого почати й т.д. Щоб перебороти це, компаніям необхідно реформувати структуру, виділити окремі департаменти усередині, які будуть займатися цифровою трансформацією. Лідерами по розвитку стають компанії, які містять у своїй ДНК більшу волю керівництва до розвитку й інноваційну культуру – бажання всієї команди постійно розвиватися й рухатися вперед.

Наша методологія розробки стратегії цифрової трансформації будується на декількох модулях, які присутні в бізнесі кожної сучасної компанії.

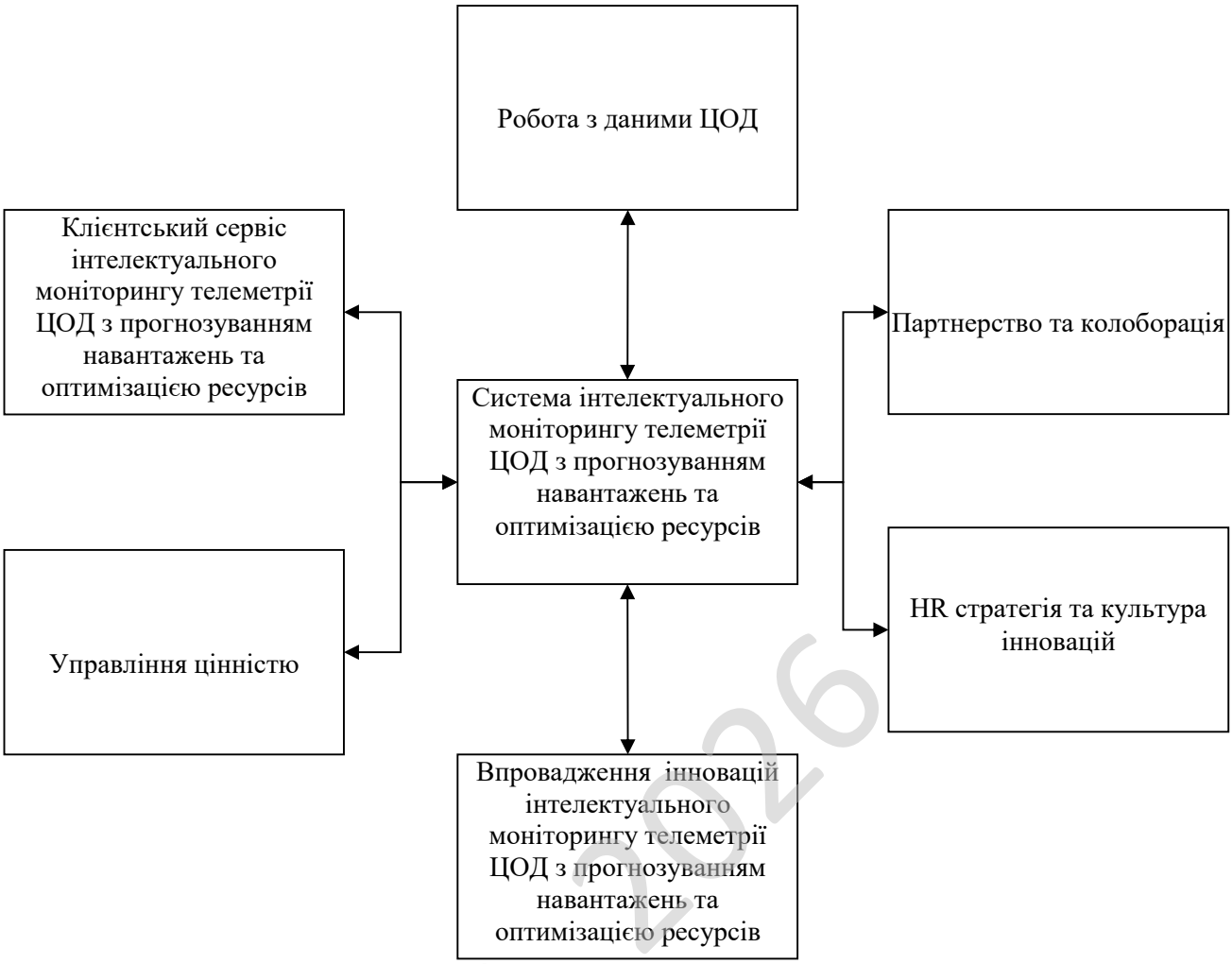

Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

У даній роботі у рамках системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів реалізований такий елемент штучного інтелекту, яка нейрона мережа.

Для моделювання на ЕОМ компонентів управляючої системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів (УС), сконструйованих з нейронів, була усвідомлена необхідність у спеціальному інструменті, що дозволяє за допомогою зручного графічного інтерфейсу створювати бібліотеки шаблонів блоків, будувати мережі

із блоків, побудованих по шаблонах, і прораховувати мережу з можливістю перегляду проміжних станів мережі, збору й аналізу статистики про роботу мережі з метою налагодження. При створенні (або виборі) інструмента використовувалися наступні критерії:

– відкритість, або специфікація й реалізація (generic) інтерфейсу й (процедур обробки) форматів даних, що дозволяють проводити модифікацію й нарощування функціональності системи не зачіпаючи ядра системи й з мінімальними витратами на модифікацію зв'язаних компонентів, інакше кажучи, мінімізація зв'язків між компонентами;

– гнучкість, можливості по конструюванню як можна більшого числа класів формальних моделей нейронів і мереж під всілякі додатки від моделей УС супутників і космічних апаратів до систем підтримки прийняття рішень і систем пророкування курсу цінних паперів;

– багатоплатформеність, максимальна незалежність від операційної системи;

– зручність і пристосованість до моделювання саме систем ААУ, простота у використанні й здатність ефективно працювати на відносно слабких ресурсах ЕОМ (класу персональних комп'ютерів), дешевина.

Аналіз наявних у наявності або доступних системах САПР і інших систем (наприклад, LabView або систем із класичними НМ), тим або іншим способом задовольняючих першим трьом критеріям, показав, що всі вони є або великоваговими, або занадто дорогими, або дуже погано пристосовані до моделювання систем автономного адаптивного управління системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів (ААУ) й об'єкта управління системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів (ОУ) з формальною моделлю нейрона або до роботи з мережами, що складаються з тисяч нейронів. Таким чином, виникла необхідність в інструменті

для науково-дослідних цілей, який би дозволяв перевіряти ідеї ААУ й створювати прототипи УС на нейронній мережі (НМ).

Організація обчислень у мережі

Після створення внутрішнього подання мережі в пам'яті у вигляді сукупності зв'язаних між собою елементів мережі, мережа готова до обчислень. Обчислення ініціюються з деякого обраної підмножини елементів мережі, називаних виходами мережі. Кожний елемент має свій метод, що реалізує операцію даної вершини мережі й результат якого інтерпретується як значення вихідного сигналу або значення виходу. Аргументами цього методу є значення виходів у входів елемента в попередні моменти часу, і, можливо, у сучасний момент. При цьому, природно, можливий нескінченний цикл у випадку неправильної специфікації мережі. На цей випадок у систему передбачається додати деякий попередній аналізатор коректності топології мережі. Розпараллелювання тут можливо при обчисленні значень виходів для елементів одного порядку, оскільки вони є незалежними.

Так як параметр часу в систему введений явно, існує необхідність у повідомленні кожному елементу про настання наступного такту обчислень. При одержанні такого повідомлення, кожний елемент виконує завершувальні операції для даного такту або може просто проігнорувати повідомлення. У реалізації механізму передачі повідомлень використаний об'єктний шаблон *Ланцюжок Оброблювачів* [15]. Суть його полягає в наступному. Припустимо існує деяка ієрархія класів або ланцюжок, де кожний попередній клас є батьківським для наступного, наприклад *ЕлементМережі* ⊃ *Нейрон* ⊃ *Нейрон2*. В *ЕлементаМережі* визначений метод обробки повідомлення *обробити_повідомлення(Повідомлення)*. У цьому методі в кожного класу при виклику визначається, чи може даний метод обробити дане повідомлення. Якщо так, то виконується обробка. Потім у кожному разі викликається метод обробки повідомлення батьківського класу, якщо він існує. Наприклад, метод обробки повідомлень в *ЕлементаМережі* збільшує лічильник тактів (лічильник часу) при

одержанні повідомлення *НаступнийТакт* (нащадок класу *Повідомлення*). Об'єкт *СередовищеЗКінцевимАвтоматом*, що є нащадком *ЕлементаМережі* й *КінцевогоАвтомата* при одержанні даного повідомлення виконує читання вхідного слова, і, природно, викликає обробку повідомлення для своїх батьківських класів.

Аналізатори роботи мережі

Для налагодження мереж часто необхідно знати різноманітну інформацію про стан мережі й окремі її елементи в деякі моменти часу. Мережі передбачаються гетерогенні, тобто, що складаються з різних елементів-екземплярів класів-нащадків *ЕлементаМережі*, і стан кожного елемента в деякі моменти часу може характеризуватися, загалом кажучи, деяким своїм набором параметрів, крім значення вихідного й вхідного сигналів. Наприклад, кінцевий автомат (КА) краще охарактеризувати станами, у якому перебуває автомат. Тому схема була обрана наступна: *ЕлементМережі* є похідним класом від *Літописець*, що має методи для запису об'єктів *Подія* в деяку часову послідовність *Історія*, що зберігає кожний *Літописець*. У процесі роботи мережі кожний *Літописці* записує в *Історію* один зі своїх або загальних об'єктів підкласу *Подія*. Наприклад, КА записує крім інших подія *СтанКА*, у якому є поле для вказівки стану КА в цей момент часу. Кожний (нащадок класу) *АналізаторРоботиМережі* вміє обробляти *Історії*, витягаючи звідти необхідну інформацію, і потім видаючи її в зручному виді на екран. У прикладі 5.8.1 наведений результат роботи програми, де зображена діаграма вихідних сигналів обраних *ЕлементовМережі*, імена яких виведені в першому рядку діаграми. Діаграма отримана як результат обробки *Історії* обраних *ЕлементовМережі АналізаторомРоботиМережі*.

Реалізація блоку оцінки стану (БОС)

Задача блоку оцінки стану в остаточному підсумку полягає в зіставленні вихідному вектору середовища W деякої оцінки або числа. Фактично, мова йде про завдання деякого функціонала над B^N, де N – розмірність вихідного вектора середовища W. Функціонал задається як таблична функція, значення якої

перераховуються в спеціальній секції файлу специфікації. Тут вказуються типи аргументу й самої табличної функції, може бути зазначене значення за замовчуванням, а потім перераховуються вхідні й відповідні вихідні значення.

Нейромережна реалізація цього блоку нами не створювалася. Передбачається, що її можна побудувати із двох простих підмереж, з яких одна являє собою нейро мережу, що розпізнає, що формує образи вектора ***in***, а інша – комбінаційну схему з нейронів, що видає сигнал на вихід з номером *out* відповідно до заданого функціонала.

Реалізація моделі середовища

Представимо середовище за допомогою КА Мура. Нехай КА має чотири стани s1, s2, s3, s4 і представлений діаграмою Мура на рисунку 3.6.

Файл специфікації для моделі середовища містить опис вихідного сигналу залежно від стану КА і опис самого КА (друга частина). П'ятий додатковий стан КА *initial* є ініціальним. Переходи вказуються у вигляді **(**<*вихідний стан*>**,** <*вхідне слово*>**) ->** <*кінцевий стан*>.

Крім намічених у цьому розділі напрямків розвитку системи, а саме створення конструктора мереж із графічним інтерфейсом, розширення мови специфікації мереж і ін., необхідні доробка й розробка нейромережних реалізацій бази знань (БЗ) і блоку прийняття рішень (БПР), розробка методів створення реальних додатків по отриманим за допомогою система побудови й дослідження нейронних мереж системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів (СПДНМ) специфікаціям мереж. Привабливим є також створення тривимірного візуалізатора БЗ. Візуалізація БЗ заснована на введенні топології в кінцевому просторі образів бази знань $\mathrm{B} = F \times Y \times F$ за допомогою відображення F і Y в R, таким чином, області в B відобразяться в області в R^3. Якщо образ $b_{ijk} \in \mathrm{B}$, $b_{ijk} = x_i^f(t-n) \wedge y_k(t-n) \wedge x_j^f(t)$ сформований, то він відображається крапкою кольору, що відповідає оцінці, що сформувалася, x_j^f образа . При цьому в просторі позначаться деякі кольорові області), що ілюструють закон управління.

На рисунку 3.2 представлена функціональна схема системи, під якою будемо розуміти середовище, у яку вкладений ОУ, у свою чергу утримуючий у собі УС. Як видно з рисунка, можна затверджувати, що УС управляє не тільки ОУ, але всією системою. Під середовищем у системі можна розуміти різні об'єднання об'єктів. Будемо називати *середовищем W* сукупність об'єктів, що лежать поза управляючою системою системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів (УС); *середовищем S* – сукупність об'єктів, що лежать поза об'єктом управління системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів (ОУ); *середовищем U* – всю систему.

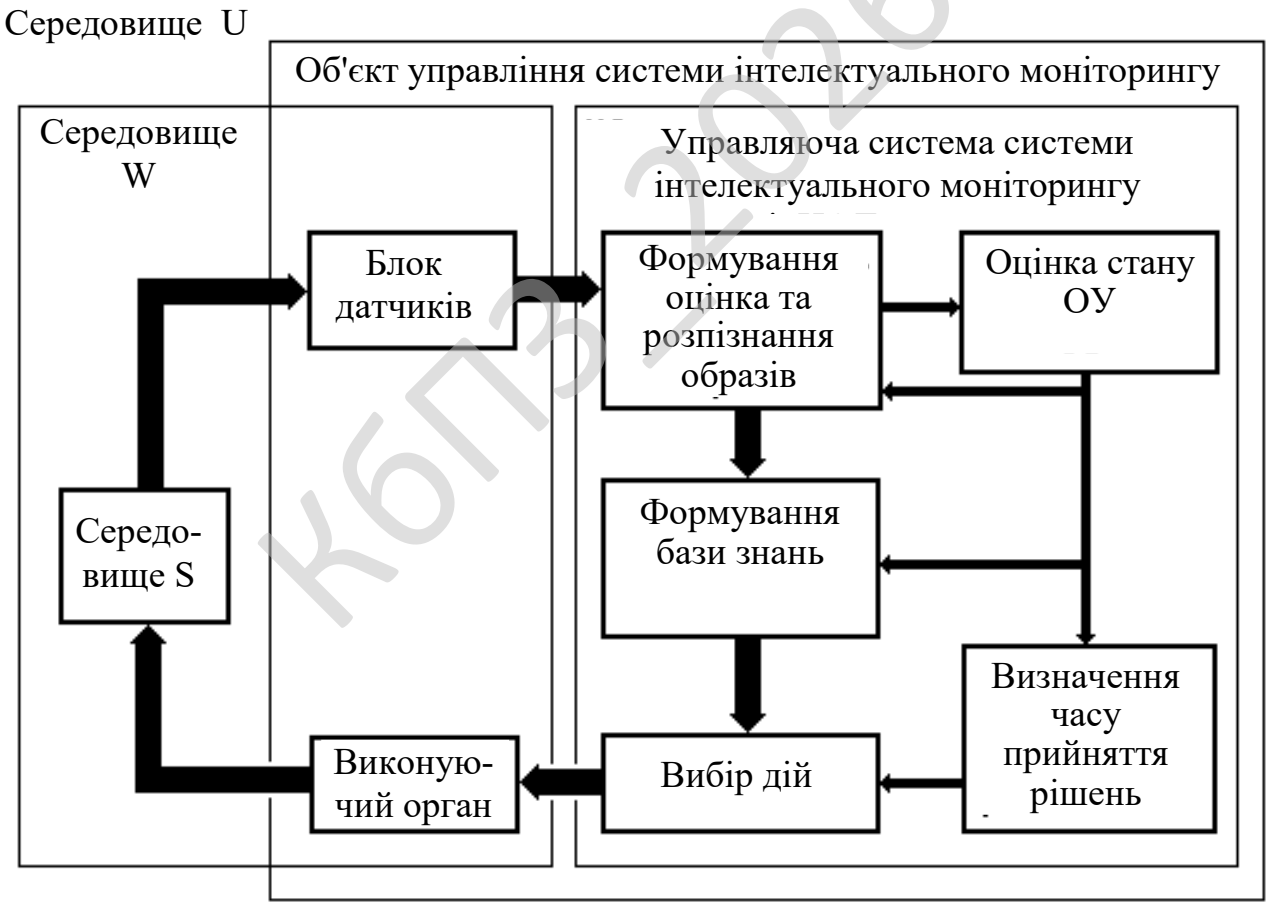

Рисунок 3.2 – Функціональна схема

Блок датчиків поставляє УС вхідну інформацію у вигляді двійкового вектора. Цей блок необхідний у реальних системах для сполучення середовища й УС, тому при моделюванні УС на ЕОМ не використовувався й ми не акцентуємо увагу на ньому в даній роботі.

Роботу блоку *формування й розпізнавання образів* (ФРО) можна представити в такий спосіб (докладний опис див. у роботах [3, 8]). У блоці ФРО на підставі апріорної інформації про можливі функціональні властивості середовища задані деякі об'єкти, назвемо їхніми *нейронами* (наприклад, нейрони спеціального виду, описані в роботі [8]), на які відображаються деякі класи часових-просторово-часових явищ, які потенційно можуть існувати в системі. Відображення задається топологією мережі. У класі, відображуваному на нейрон, виділяється підклас, що може сприйматися даним нейроном. Кожний нейрон може статистично аналізувати сприйманий їм підклас. Накопичуючи статистичну інформацію про сприйманий підклас, нейрон може прийняти рішення, чи є цей підклас випадковим або невипадковим явищем у системі. Якщо який-небудь нейрон приймає рішення, що відображуваний на нього підклас є невипадковою подією, то він переходить у деяке відмінне від вихідного ″навчене″ стан. Якщо нейрон навчений, то будемо говорити також, що сформовано образ, цей образ ідентифікується номером даного нейрона. Підклас явищ, сприйманий нейроном, і викликавший його навчання, тобто просторово-часові явища, що статистично вірогідно існують у системі, називається *прообразом* даного образа. Сформований образ може бути розпізнаний блоком ФРО, коли прообраз даного образа спостерігається БД. Блок ФРО вказує, які зі сформованих образів розпізнані в сучасний момент. Одночасно із цим розпізнані образи беруть участь у формуванні образів більше високих порядків, тобто має місце агрегування й абстрагування образів.

Блок *формування бази знань* [4-6] (БЗ) призначений для автоматичного подання емпірично знайдених УС знань про функціональні властивості системи. Елементарною конструкцією бази знань (БЗ) у методі ААУ є статистично

достовірна відомість про те, як певна дія Y_j впливає на прообраз певного сформованого образа. *Дією* Y_j названа підмножина *множини припустимих впливів*, елементи якого абсолютно ідентичні для УС по їхньому впливі на сформовані образи. Непуста відомість може мати одне із двох значень: або дія Y_j тягне розпізнавання образа O_i , або дія Y_j тягне витиснення образа O_i. За допомогою БЗ можна бачити, як конкретна дія впливає на всю сукупність сформованих образів.

Блок оцінки стану [7] (БОС) виробляє інтегральну оцінку якості стану ОУ S_t. Оцінка S_t використовується для розрахунку оцінки (ваги) p_i кожного зі знову сформованих образів деяким статистичним способом. У свою чергу, S_t функціонально залежить від оцінок p_i розпізнаних образів. Є деяка множина споконвічна сформованих і оцінених образів. Оцінка S_t використовується також для розрахунку темпу прийняття рішень.

Блок *вибір дії* [4-6] або, надалі, *блок прийняття рішень* (БПР) реалізує процедуру ухвалення рішення, засновану на аналізі поточної ситуації, цільових функцій, умісту БЗ, а також оцінки поточного значення оцінки S_t. Фактична інформація про поточну ситуацію представлена множиною образів, розпізнаних у сучасний момент блоком ФРО, а інформація про якість поточного стану представлена оцінкою S_t. Множина розпізнаних образів визначає в БЗ той її розділ, що адекватний поточної ситуації (ті знання, які дійсні в поточних умовах). Відповідно до цільової функції, що припускає прагнення УС до поліпшення якості стану ОУ, УС вибирає по БЗ ту дію, що має максимальну суму оцінок викликуваних і образів, що витісняються. Із множини вихідних впливів, що відповідає обраній дії Y_j, конкретне вихідний вплив вибирається випадковим способом, що відповідає другій цільовій функції, що передбачає прагнення до одержання нових знань.

Блок *визначення часу ухвалення рішення* визначає глибину перегляду БЗ залежно від поточної оцінки S_t. Чим вище значення S_t, тим більше образів (у порядку убування модуля їхньої ваги) може врахувати УС при ухваленні

рішення, тим менше темп прийняття рішень. При моделюванні цей блок не використовувався й у даній роботі розглядатися не буде.

У УС можуть бути засоби для апріорного аналізу наслідків альтернативних обираних дій на кілька кроків уперед.

Такий загалом алгоритм управління, реалізований УС у методі ААУ. Основні властивості процесу управління полягають у тому, що УС автоматично накопичує емпіричні знання про властивості пред'явленого їй об'єкта управління й приймає рішення, опираючись на накопичені знання.

Якість управління росте в міру збільшення об'єму накопичених знань. Помітимо також, що управління складається не в тому, що УС реагує на вхідну інформацію (у певному змісті – негативний зворотний зв'язок), а в тому, що УС постійно активно шукає можливий у поточних умовах спосіб поліпшити стан ОУ (позитивний зворотний зв'язок).

Тим самим УС ААУ має внутрішню активність.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

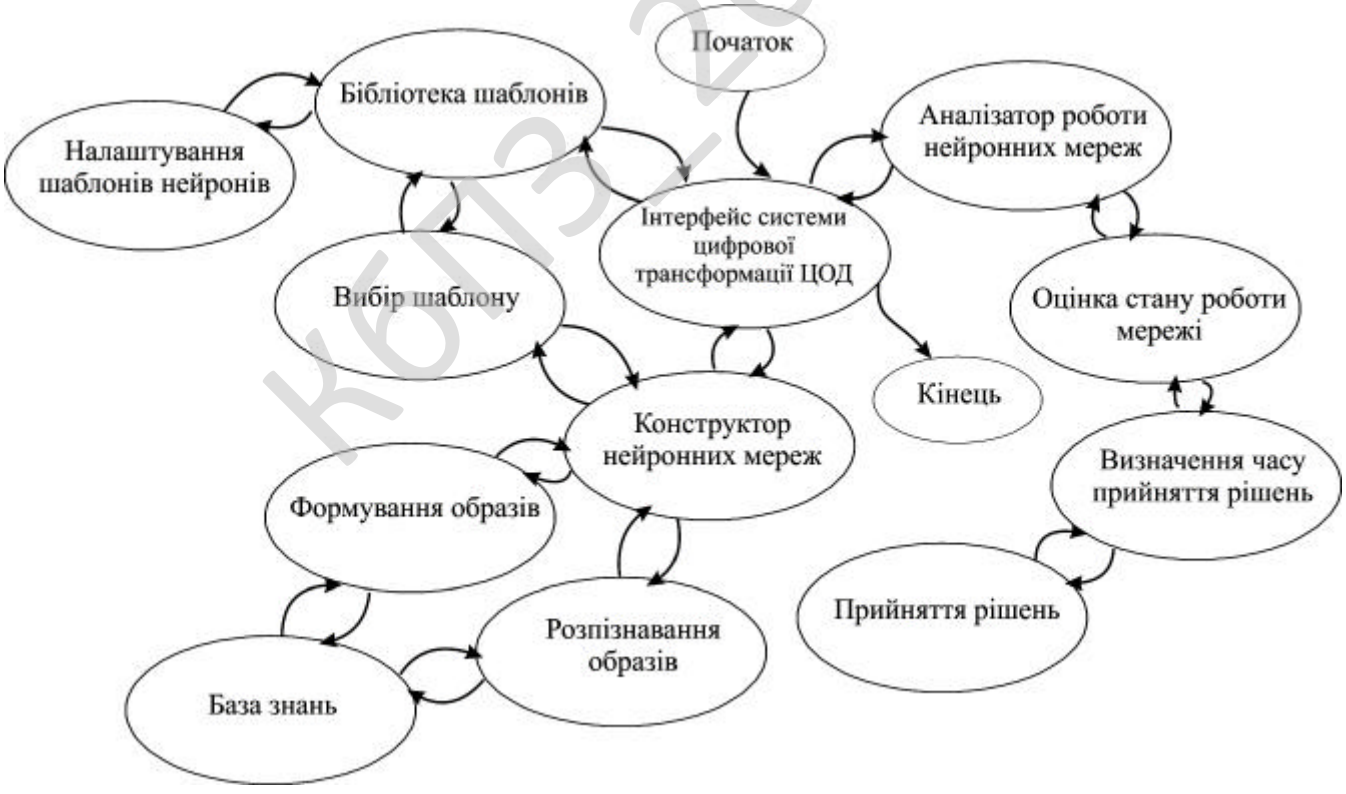

Рисунок 3.3 – Діаграма взаємодії процесів

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

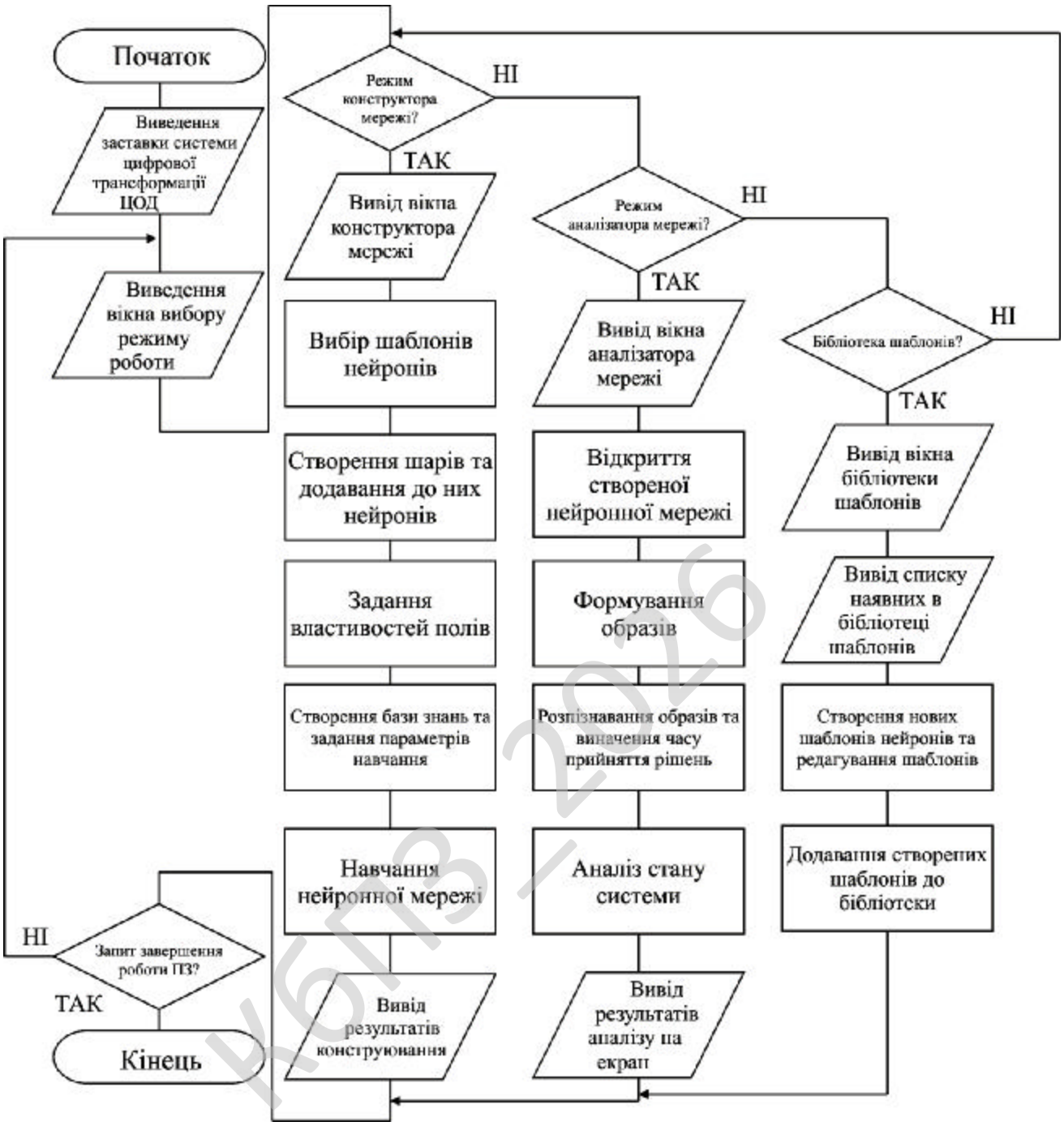

Рисунок 4.1 – Блок-схема основної програми

Було використано наступні підходи UML: Діаграма об'єктів; Діаграма розгортання.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок

діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

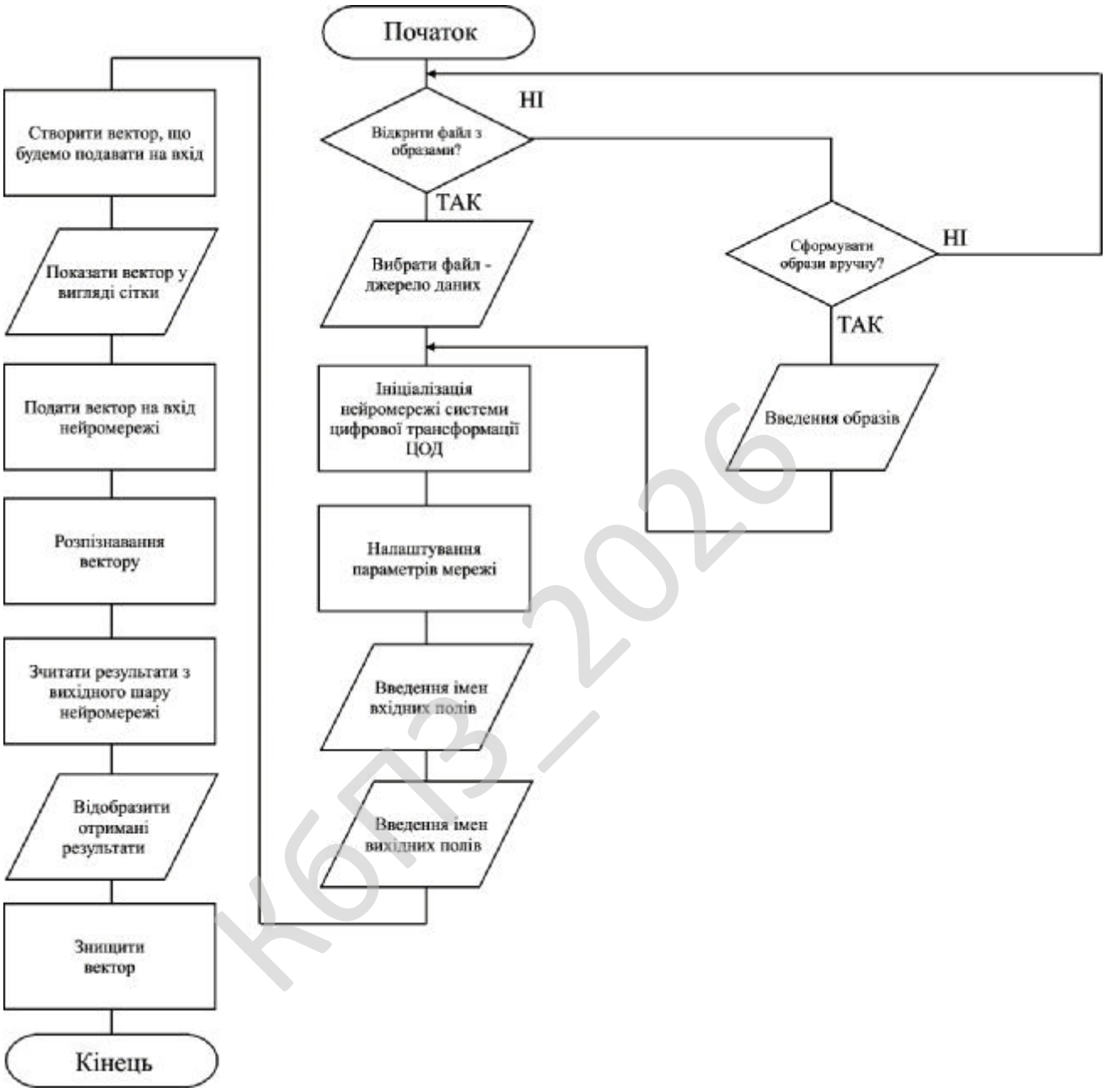

Рисунок 4.2 – Блок-схема роботи підпрограми

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Подійно-орієнтована архітектура (Event-driven architecture, надалі EDA) – шаблон архітектури програмного забезпечення, який призначений для створення подій, їх виявлення, споживання і реагування на них.

Подія може бути визначена як значна зміна стану. Наприклад, коли споживач купує автомобіль, стан автомобіля змінюється з "на продаж" до "продано". Архітектура системи дилера автомобілів може трактувати цю зміну стану як подію, поява якої може стати відомою іншим програмам даної архітектури.

З формальної точки зору, те, що виробляється, публікується, поширюється, виявляється і споживається (як правило, асинхронно) є повідомленням, яке називають сповіщенням про подію (або нотифікацією), а не самою подією, яка є зміною стану, що викликає появу повідомлення.

Події не подорожують, вони просто відбуваються. Проте термін подія часто використовується метонімічно для позначення самого нотифікаційного повідомлення, що може призвести до певної плутанини.

Цей архітектурний шаблон може застосовуватися при проектуванні і реалізації ПЗ і систем, які передають події між слабкозв'язаними компонентами програмного забезпечення і сервісами (службами).

Подійно-орієнтована система як правило складається з емітерів подій (або агентів) і споживачів подій (або стоків).

Стоки несуть відповідальність за здійснення реагування на появу події. Реакція не завжди може бути повністю забезпечена самим стоком. Наприклад, стік, може бути відповідальним лише за фільтрацію, трансформацію і відправку події до іншого компонента або він може забезпечити повністю самостійну реакцію на таку подію. Перша категорія стоків може бути заснована на традиційних компонентах, таких як проміжне програмне забезпечення, орієнтоване на обробку повідомлень (message oriented middleware, MOM), в той час, як друга категорія стоків (самостійна реакція в режимі он-лайн) може вимагати більш придатної платформи (фреймворку) для виконання транзакцій.

Розробка ПЗ і систем в подійно-орієнтованій архітектурі дозволяє їм бути сконструйованими способом, який більш відповідає вимогам до їх створення, оскільки такі системи в більшій мірі пристосовуються до непередбачуваних і асинхронних середовищ.

Подійно-орієнтована архітектура (EDA) може доповнювати сервісно-орієнтовану архітектуру (SOA), оскільки сервіси (служби) можуть бути активовані тригерами, які ініціюються при настанні подій.

Ця парадигма особливо корисна, коли стік не забезпечує власного виконання будь-яких дій.

Подійно-орієнтована SOA (SOA-2) розвиває архітектури SOA і EDA для забезпечення більш глибокого і надійного рівня сервісів за рахунок використання раніше невідомих причинно-наслідкових зв'язків, щоб сформувати новий шаблон

подій. Цей новий шаблон бізнес-аналітики дає поштовх до небаченого раніше зростання рівня автоматизації підприємства за рахунок привнесення додаткової цінної інформації в описану раніше модель діяльності.

Обчислювальна техніка та сенсорні пристрої (сенсори, датчики, контролери) можуть виявляти зміни стану об'єктів або умов і створювати події, які потім можуть бути оброблені сервісом (службою) або системою.

Подія може складатися з двох частин: заголовка події та тіла події. Заголовок події може включати в себе інформацію таку як, наприклад, назва події, часова мітка події і тип події. Тіло події – це частина, яка описує факт, що стався в дійсності. Тіло події не слід плутати з шаблоном або логікою, яка може бути застосована як реакція на саму події.

Архітектура, керована подіями, складається з чотирьох логічних рівнів (шарів). Вона починається з виявлення факту, його технічного подання у формі події і закінчується непустою множиною реакцій на цю подію.

Генератор подій.

Першим логічним шаром є генератор подій, який виявляє факт і представляє цей факт подією. Оскільки фактом може бути практично все, що може бути сприйнято, то ним може бути і генератор подій. Наприклад, генератором може бути клієнт електронної пошти, система електронної комерції або певний тип датчика.

Перетворення різних даних, отриманих від датчиків, в єдину стандартизовану форму даних, які можуть бути оцінені, є основною проблемою при розробці та реалізації цього шару. Однак, враховуючи, що подія є строго декларативною, можна легко застосовувати будь-які операції трансформації, тим самим усуваючи необхідність забезпечення високого рівня стандартизації.

Канал подій.

Канал подій – це механізм, через який інформація від генератора подій передається до обробника подій (event engine) або стоку.

Це може бути з'єднання TCP/IP або вхідний файл будь-якого типу (простий текст, формат XML, e-mail тощо). В один і той же час може бути відкрито кілька каналів подій. Як правило, оскільки обробник подій повинен працювати в режимі, наближеному до реального часу, канали подій зчитуються асинхронно. Події зберігаються в черзі, очікуючи наступної обробки механізмом обробки подій.

Механізм обробки подій.

Механізм обробки подій (event processing engine) є місцем, де подія ідентифікується і вибирається відповідна реакція на нього, яка потім виконується. Це також може призвести до породження ряду тверджень. Якщо подія, яка надійшла до механізму обробки подій, є наприклад такою «Запаси продукту ID досягли нижнього допустимого рівня», це може ініціювати, наприклад, такі реакції як «Замовити продукт ID» і «Сповістити персонал».

Наступна подійно-орієнтована дія (післядія).

Щодо того, як можуть проявлятися наслідки події, слід відмітити, що вони можуть проявитись багатьма різними способами і у різноманітних формах (наприклад, повідомлення електронної пошти, надіслане комусь, або ПЗ, що виводить деяке попередження на екран). Залежно від рівня автоматизації, який забезпечується стоком (механізмом обробки подій), ці дії можуть виявитись зайвими.

Є три основні стилі обробки подій: простий, потоковий і складний. Часто ці три стилі використовуються спільно у розвинутій подійно-орієнтованій архітектурі.

Проста обробка подій.

Проста обробка подій стосується подій, які безпосередньо належать до специфічних вимірних змін умов. У випадку простої обробки подій, мають справу з появою відомих подій, що ініціюють післядію (післядії). Проста обробка подій зазвичай використовується для управління потоком робіт в реальному часі, скорочуючи тим самим час затримки і вартість робіт.

Наприклад, прості події можуть створюватись (породжуватись) датчиком, що виявляє зміну тиску в шині або температуру навколишнього середовища.

Обробка потоку подій.

При обробці потоку подій (event stream processing, далі ESP) відбуваються як звичайні, так і відомі події. Звичайні події (заявки, передачі RFID) перевіряються на те, чи є вони відомими, і передаються інформаційним передплатникам. Обробка потоку подій зазвичай використовується для управління потоком інформації в реальному часі і на рівні підприємства, що дозволяє своєчасно приймати рішення.

Обробка складних подій.

Обробка складних подій (Complex event processing (CEP)) дозволяє за шаблонами простих і звичайних подій проводити аналіз того, чи наступила складна подія. Обробка складних подій полягає в оцінюванні взаємного впливу подій і в наступному виконанні дій. При цьому, типи подій (відомих або звичайних) можуть перетинатись, а події можуть виникати протягом тривалого періоду часу.

Кореляція подій може бути причинною, тимчасовою або просторовою. CEP вимагає використання складних інтерпретаторів подій, визначення і підбору шаблонів подій, а також відповідних кореляційних методів. Обробка складних подій зазвичай використовується для виявлення і реагування на аномальну поведінку, загрози і можливості у бізнесі.

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому

повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скриншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько $50 за користувача для локального і $7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

– проект використовує ліцензії, схвалені Open Source Initiative;

– Вихідний код проекту доступний для скачування;

– у проекту є публічно доступна веб-сайт;

– програмне забезпечення від Atlassian є на веб-сайті проекту.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ДСТУ 28147:2009, що є класичним алгоритмом симетричного шифрування на основі мережі Фейстеля (рисунок 4.4). Даний алгоритм шифрує інформацію блоками по 64 біта (такі алгоритми називаються "блоковими"). Зміст мережі Фейстеля полягає в тому, що блок шифруемої інформації розбивається на два або більше субблоків, частина яких обробляється за певним законом, після чого результат цієї обробки накладається (операцією побітового додавання за модулем 2) на необроблювані субблоки. Потім субблоки міняються місцями, після чого обробляються знову й т.д. певне для кожного алгоритму число раз – раундів.

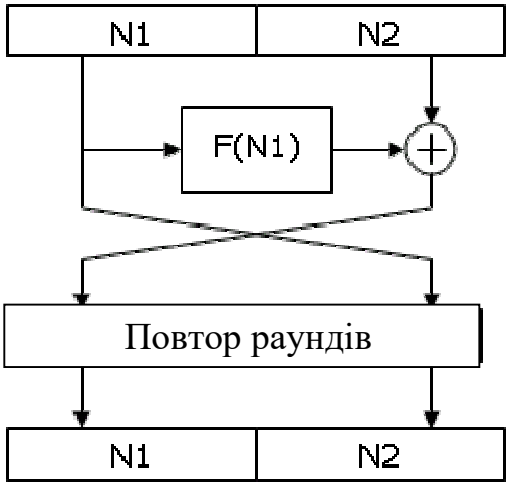

Рисунок 4.3 – Мережа Фейстеля

Основна відмінність алгоритмів симетричного шифрування друг від друга складається саме в різних функціях обробки субблоків. Дана функція часто називається "основним криптографічним перетворенням", оскільки саме вона несе основне навантаження при шифруванні інформації. Основне перетворення алгоритму ДСТУ 28147:2009 є досить простим, що забезпечує високу швидкодію алгоритму; у ньому виконуються наступні операції (рисунок 4.5).

Рисунок 4.4 – Основне перетворення алгоритму ДСТ 28147:2009

1. Додавання субблоку з певним фрагменом ключа шифрування за модулем 2^{32}. Кх – це 32-бітна частина ("підключ") 256-бітного ключа шифрування, якому можна представити як конкатенацію 8 підключів: К = $К_0К_1К_2К_3К_4К_5К_6$ $К_7$. Залежно від номера раунду й режиму роботи алгоритму (про їх – нижче), для даної операції вибирається один з підключів.

2. Таблична заміна. Для її виконання субблок розбивається на 8 4-бітних фрагментів, кожний з яких прогоняється через свою таблицю заміни. Таблиця заміни містить у певній послідовності значення від 0 до 15 (тобто всі варіанти значень 4-бітні фрагменти даних); на вхід таблиці подається блок даних, числове подання якого визначає номер вихідного значення. Наприклад, подається значення 5 на вхід наступної таблиці: "13 0 11 74 91 10 143 5 122 15 8 6". У результаті на виході виходить значення 9 (оскільки 0 заміняється на 13, 1 – на 0, 2 – на 11 і т.д.).

3. Побітове циклічне зрушення даних усередині субблока на 11 біт уліво.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку зображено розроблене у бакалаврської дипломної роботі програмне забезпечення системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Функціональних кнопок ПЗ.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Розділу обрання групи обробки ЦОД.

– Розділу виведення результату роботи системи.

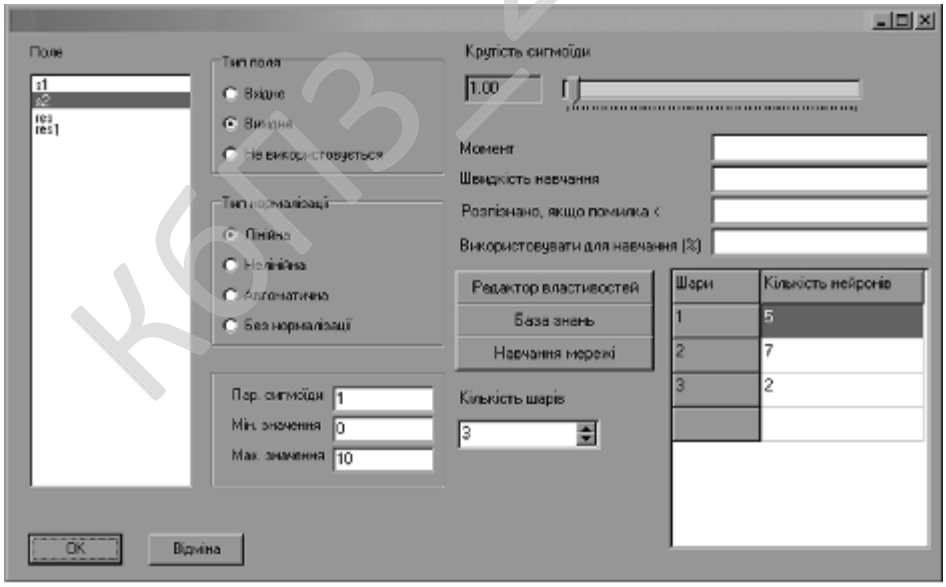

Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

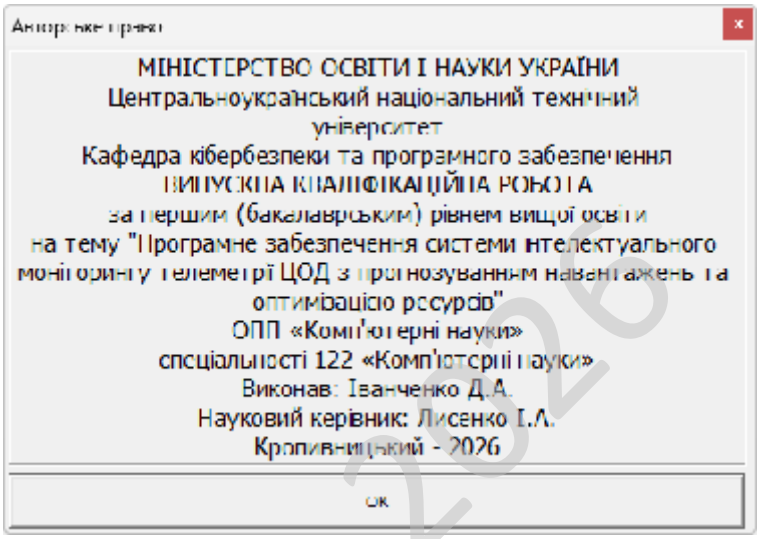

Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних

вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

– Встановлення та активація.

– Деактивація.

– Адаптація.

– Обновлення.

– Вмонтування.

– Відстежування версій.

– Видалення.

– Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути

невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

– Досліджена система інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів.

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 Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

системи інтелектуального моніторингу телеметрії ЦОД з прогнозуванням навантажень та оптимізацією ресурсів. 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.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 28147:2009.

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. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.

2. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка»* (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

3. 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.

4. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

5. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

6. 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.

7. 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.

8. 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.

9. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

10. 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.

11. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.

12. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

13. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

14. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного

захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

15. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

16. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

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

18. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

19. 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.

20. 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.

21. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T),* Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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

35. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у

комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки.* №4. С. 103-110. 2020.

36. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

37. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

38. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія.* – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

39. Смірнов О.А., Дрєєва Г.М., Дрєєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки.* № 2(33). с. 161-172, 2019.

40. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

41. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

42. Смірнов О.А., Дрєєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

43. Дрєєва Г.М., Смірнов О.А., Дрєєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

44. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дрєєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

45. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Конопліцька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

46. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

47. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

48. Смірнов О.А., Смірнов С.А., Дідик А.К., Дрєєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". - Випуск 5 (142). - Х.: ХУПС - 2016. - С. 148-152.

49. Смірнов О.А., Смірнов С.А. Дідик А.К., Дрєєв О.М. Моделі системи нейромережевих експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дрєєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових

праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. - 2016. - С. 121-127.

51. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. - Випуск 2 (46) - Х.: ХУПС - 2016. - С. 146-149.

52. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

53. Смірнов О.А., Кавун С.В., Коваленко О.В., Дрєєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

54. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дрєєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

55. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

56. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

57. Смірнов О.А., Євсеєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. Навчальний посібник. – Кіровоград: КНТУ 2012. – 454 с