МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ ТА РАДІОЕЛЕКТРОННИХ СИСТЕМ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

___________ Віктор ГНАТЮК
"_____" ______________ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

Тема: «Метод оцінки рівня кібербезпеки підприємства»

Виконавець: ________________________ Ілля ВОЙТЮК
(підпис)

Керівник: ________________________ Олександр ПУЗИРЕНКО
(підпис)

Консультанти з окремих розділів пояснювальної записки:

Консультант розділу «Охорона праці» ________________ Катерина КАЖАН
(підпис)

Консультант розділу «Охорона навколишнього середовища»
________________________ Лариса ЧЕРНЯК
(підпис)

Нормоконтролер: ________________________ Богдан ЧУМАЧЕНКО
(підпис)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Факультет аеронавігації, електроніки та телекомунікацій
Кафедра телекомунікаційних та радіоелектронних систем
Спеціальність 172 «Електронні комунікації та радіотехніка»
Освітньо-професійна програма «Телекомунікаційні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри

____________ Віктор ГНАТЮК
"_____" ______________ 2025 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Войтюка Іллі Олексійовича
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема кваліфікаційної роботи: «Метод оцінки рівня кібербезпеки підприємства»
затверджена наказом ректора від «02» вересня 2025 р. № 1672 /ст

2. Термін виконання роботи: з 29.09.2025 р. по 31.12.2025 р.

3. Вихідні дані до роботи:. доменно-критеріальна модель оцінювання кібербезпеки підприємства та вимоги до застосунку.

4. Зміст пояснювальної записки: аналіз поняття кібербезпеки підприємства, механізмів забезпечення та методів оцінювання; розробка методу інтегральної оцінки; проєктування та реалізація MVP-системи оцінювання.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:. схеми, рисунки, таблиці

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Розробити деталізований зміст розділів кваліфікаційної роботи	29.09.2025-30.09.2025	Виконано
2	Вступ	01.10.2025-03.10.2025	Виконано
3	Дослідження поняття кібербезпеки підприємства	04.10.2025-14.10.2025	Виконано
4	Розробка методу оцінки рівня кібербезпеки підприємства	15.10.2025-26.10.2025	Виконано
5	Програмна реалізація методу оцінки рівня кібербезпеки підприємства	27.10.2025-16.11.2025	Виконано
6	Охорона праці	17.11.2025-30.11.2025	Виконано
7	Охорона навколишнього середовища	01.12.2025-14.12.2025	Виконано
8	Усунення недоліків та захист кваліфікаційної роботи	15.12.2025-31.12.2025	Виконано

7. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	к.т.н., доц. Катерина КАЖАН		
Охорона навколишнього середовища	д.т.н., доц. Лариса ЧЕРНЯК		

8. Дата видачі завдання: «01» вересня 2025 р.

Керівник кваліфікаційної роботи ____________________ Олександр ПУЗИРЕНКО
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання ____________________ Ілля ВОЙТЮК
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Кваліфікаційна робота «Метод оцінки рівня кібербезпеки підприємства» містить 129 сторінок, 24 рисунки, 18 таблиць, 64 використаних джерела.

МЕТОД ОЦІНКИ РІВНЯ КІБЕРБЕЗПЕКИ ПІДПРИЄМСТВА; КІБЕРБЕЗПЕКА, ДЕРЕВО РІШЕНЬ, КІБЕРБЕЗПЕКА.

Об’єкт дослідження – процеси забезпечення кібербезпеки підприємства в умовах використання інформаційно-комунікаційних систем і цифрових сервісів.

Предмет дослідження – методи та моделі оцінювання рівня кібербезпеки підприємства на основі доменно-критеріальної структури, вагового узгодження показників і обробки невизначених лінгвістичних оцінок з подальшим формуванням рекомендацій.

Мета кваліфікаційної роботи – розробити й обґрунтувати метод оцінки рівня кібербезпеки підприємства, який на основі відповідей з інтерфейсу опитування розраховує інтегральний показник та формує контекстно релевантні рекомендації і дерево рішень для підвищення цього показника.

Метод дослідження – истемний аналіз і порівняльний аналіз підходів; побудова доменно-критеріальної моделі; багатокритеріальне прийняття рішень з визначенням ваг методом AHP; нечітка логіка для обробки лінгвістичних відповідей; математичне моделювання агрегації в інтегральний бал; прототипування та валідація ідеї.

Матеріали кваліфікаційної роботи рекомендується використовувати при самооцінюванні або проходження аудиту з кібербезпеки підприємства, формуванні пріоритезованого плану підвищення рівня кібербезпеки, підготовці управлінських рішень щодо інвестицій у захист.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

AHP (Analytic Hierarchy Process) – метод аналізу ієрархій, що визначає ваги доменів та критеріїв.

API (Application Programming Interface) – прикладний програмний інтерфейс.

BCP (Business Continuity Plan) – план безперервності бізнесу.

CIS (Center for Internet Security) –ініціатива практик кіберзахисту CIS Controls.

CSF (Cybersecurity Framework) – фреймворк кібербезпеки NIST CSF 2.0.

D1–D12 – коди доменів оцінювання.

DDoS (Distributed Denial of Service) – розподілена відмова в обслуговуванні.

DoS (Denial of Service) – відмова в обслуговуванні.

DR (Disaster Recovery) – аварійне відновлення.

HTTP (HyperText Transfer Protocol) – протокол передавання даних у вебі.

IAM (Identity and Access Management) – управління ідентифікацією та доступами.

IDS (Intrusion Detection System) – система виявлення вторгнень.

IEC (International Electrotechnical Commission) – Міжнародна електротехнічна комісія.

IR (Incident Response) – реагування на інциденти.

ISCM (Information Security Continuous Monitoring) – безперервний моніторинг інформаційної безпеки.

ISMS (Information Security Management System) – система управління інформаційною безпекою.

ISO (International Organization for Standardization) – Міжнародна організація зі стандартизації.

ISO/IEC 27001 – стандарт вимог до системи управління інформаційною безпекою.

ISO/IEC 27002 – стандарт настанов та контролів інформаційної безпеки.

ISO/IEC 27005 – стандарт управління ризиками інформаційної безпеки.

JSON (JavaScript Object Notation) – формат обміну даними.

JSONB (JSON Binary) – бінарний тип JSON у PostgreSQL.

JML (Joiner–Mover–Leaver) – процеси прийому, зміна ролі та звільнення працівників.

L0 – L4 – рівні оцінювання зрілості критерію.

MFA (Multi-Factor Authentication) – багатофакторна автентифікація.

MVP (Minimum Viable Product) – мінімально життєздатний продукт.

N/A (Not Applicable) – не застосовується.

NIST (National Institute of Standards and Technology) – Національний інститут стандартів і технологій США.

Nginx – вебсервер та reverse proxy.

ORM (Object-Relational Mapping) – об’єктно-реляційне відображення.

PAM (Privileged Access Management) – управління привілейованими доступами.

PostgreSQL – реляційна система керування базами даних.

REST (Representational State Transfer) – архітектурний стиль побудови веб-API.

SaaS (Software as a Service) – програмне забезпечення як сервіс.

Score – інтегральний показник рівня кібербезпеки.

SQL (Structured Query Language) – мова структурованих запитів.

TFN (Triangular Fuzzy Number) – трикутне нечітке число.

VPN (Virtual Private Network) – віртуальна приватна мережа.

uvicorn – ASGI-сервер для запуску backend-застосунку.

БД – база даних.

ІКС – інформаційно-комунікаційні системи.

КБ – кібербезпека.

ВСТУП

Актуальність теми. Підприємства критично залежать від інформаційно-комунікаційних систем і цифрових сервісів, тому кібератаки та інциденти інформаційної безпеки безпосередньо впливають на безперервність бізнес-процесів, збереження даних і виконання договірних та регуляторних вимог. На практиці керівництву потрібні вимірювані показники, які дозволяють визначати поточний рівень кібербезпеки, відстежувати його динаміку та обґрунтовувати пріоритети впровадження заходів захисту. Існуючі підходи часто мають розрив між анкетуванням, формалізованим розрахунком інтегрального показника та подальшими керованими рекомендаціями, а також не забезпечують достатньої відтворюваності результатів у програмному вигляді. Це зумовлює актуальність розроблення методу оцінки рівня кібербезпеки підприємства, який поєднує доменно-критеріальну структуру, вагове узгодження та обробку лінгвістичних оцінок з подальшим формуванням рекомендацій і реалізацією у вигляді прикладної системи.

Зв’язок роботи з науковими програмами, планами, темами. Державний науково-дослідний проєкт молодих вчених МОН України «Методи побудови захищених багатошарових стільникових мереж 5G/6G на основі використання алгоритмів штучного інтелекту для моніторингу об'єктів критичної інфраструктури країни» (№ 0124U000197).

Мета і завдання дослідження. Розробити метод оцінки рівня кібербезпеки підприємства та реалізувати MVP-застосунок, який забезпечує проходження опитування, розрахунок інтегрального показника і формування структурованих рекомендацій щодо підвищення рівня кібербезпеки. Для досягнення поставленої мети вирішуються такі наукові завдання:

1. Дослідити поняття кібербезпеки підприємства, механізми її забезпечення та підходи до оцінювання рівня кібербезпеки та кіберзрілості з визначенням вимог до методу оцінки.

2. Сформувати доменно-критеріальну модель оцінювання та визначити структуру доменів і критеріїв із прив'язкою до визнаних рамок і стандартів.

3. Розробити математичний апарат оцінювання, що включає визначення ваг доменів, перетворення лінгвістичних відповідей та розрахунок інтегрального показника.

4. Впровадити механізми підвищення достовірності та коректності оцінки: коефіцієнт доказовості, перенормування ваг за певних умов.

5. Реалізувати MVP інформаційної системи у вигляді веб-додатку.

Об'єктом дослідження є процеси забезпечення кібербезпеки підприємства в умовах використання інформаційно-комунікаційних систем і цифрових сервісів.

Предметом дослідження – методи та моделі оцінювання рівня кібербезпеки підприємства на основі доменно-критеріальної структури, вагового узгодження показників і обробки невизначених лінгвістичних оцінок з подальшим формуванням рекомендацій.

Методи досліджень. Системний аналіз і узагальнення наукових джерел і стандартів у сфері кібербезпеки та управління ризиками; порівняльний аналіз підходів оцінювання; метод аналізу ієрархій AHP для визначення вагових коефіцієнтів; методи нечіткої логіки із застосуванням трикутних нечітких чисел та дефазифікації для інтерпретації лінгвістичних оцінок; математичне моделювання та алгоритмізацію для розрахунку інтегрального показника; методи проєктування ПЗ та прототипування для реалізації MVP у вигляді клієнт-серверної системи з реляційною базою даних.

Наукова новизна та практичне значення отриманих результатів.

Наукова новизна отриманих результатів:

1. Розроблено метод інтегральної оцінки рівня кібербезпеки підприємства, який формалізує доменно-критеріальну структуру оцінювання, вагове узгодження важливості доменів методом AHP та інтерпретацію лінгвістичних відповідей через нечіткі множини з отриманням інтегрального показника.

2. Обґрунтовано і реалізовано механізми підвищення коректності самооцінки в межах методу: коефіцієнт доказовості, обробку даних із перенормуванням ваг, штрафні правила для критичних умов, а також формування структурованих рекомендацій як вихідного артефакту оцінювання.

Практичне значення отриманих результатів:

1. Реалізовано MVP-застосунок для оцінювання кібербезпеки підприємства, який забезпечує повний цикл опитування, обробки даних та формулювання рекомендацій у контейнеризованій архітектурі.
2. Розроблено модель даних із версійністю опитувальника та історизацією запусків, що забезпечує відтворюваність результатів, повторний перерахунок та накопичення історії оцінювань для подальшого аналізу і використання в управлінській практиці.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ ПОНЯТТЯ КІБЕРБЕЗПЕКИ ПІДПРИЄМСТВА

1.1. Дослідження поняття кібербезпеки підприємства

У сучасній цифровій економіці підприємства фактично функціонують як соціотехнічні системи, що поєднують в собі людей, процеси та технології, а критичні бізнес-функції залежать від мереж, хмарних сервісів, корпоративних інформаційних систем, цифрових ланцю

гів постачання, від налаштованих операційних та індустріальних технологій тощо. В таких умовах наслідок від кіберзагрози може перерости з технічної проблеми відділу кібербезпеки до повноцінного системного ризику для безперервності бізнесу, репутації, відповідності регуляторним вимогам і здатності організації масштабуватися і розвиватися. Таким чином, з’являється необхідність науково обгрунтувати поняття кібербезпеки підприємства та узгодити термінологію для подальшого напрацювання методу її оцінки. Це теж є своєрідною задачею, оскільки сучасні стандарти кібербезпеки використовують терміни не стандартизовано, що спричиняє певні розбіжності у трактуванні базових понять [1].

Термін кіберпростір зазвичай розуміють як середовище взаємодії цифрових ресурсів, тобто мереж, систем, сервісів та пристроїв, де відбувається обробка й передавання даних. Відповідно, кібератака є навмисною спробою втрутитися, що спрямована на порушення нормальної роботи, компрометацію даних або отримання несанкціонованого доступу до ресурсів у цьому середовищі.

Наприклад Національний інститут стандартів і технологій визначає поняття кібербезпека як здатність захищати або обороняти використання кіберпростору від кібератак [2]. Це визначення важливе тим, що фіксує кібербезпеку саме як здатність організації підтримувати прийнятний рівень захищеності в умовах активної протидії.

Для підприємства, однак, цього недостатньо: потрібно уточнити що саме захищається і якою ціною, враховуючи кризовий менеджмент та пріоритети бізнесу.

Тому в академічних роботах і прикладних фреймворках кібербезпека розкривається через очікувані результати та процеси управління ризиками. До прикладу, у NIST Cybersecurity Framework версії 2.0 ядро результатів поділене структурно за функціями Govern, Identify, Protect, Detect, Respond, Recover, де Govern, тобто управління, підкреслює організаційно-управлінську природу кібербезпеки, а не лише технічну [3].

Актуальність забезпечення належного рівня кібербезпеки для підприємств стрімко зросла в умовах цифрової трансформації бізнесу. За недавніми оцінками, понад 53% організацій у світі зазнали хоча б однієї успішної кібератаки протягом року, що призводило до простоїв у роботі, компрометації конфіденційних даних та значних фінансових втрат [4]. Автори дослідження зауважують, що ефективна система кібербезпеки повинна спиратися не лише на технологічні рішення, але й на організаційні та людські фактори [5]. Зокрема, є необхідність у використанні міждисциплінарного підходу: поєднання технічних, організаційних та правових заходів для побудови цілісної стратегії безпеки підприємства [6]. Ігнорування таких факторів, як корпоративна культура безпеки, обізнаність персоналу чи наявність чітких політик, може спотворити ефективність навіть найдосконаліших технічних засобів захисту [4]. Отже, поняття кібербезпеки підприємства виходить за рамки суто інформаційно-технологічної безпеки, воно включає управління процесами і людьми з метою мінімізації кіберризиків.

Варто розглянути і контекст сьогодення. В Україні питання кібербезпеки підприємств набуло надзвичайно особливої ваги у зв’язку з веденням гібридної війни та масованими кібератаками на критичну інфраструктуру. Україна стикається з підвищеними ризиками в кіберпросторі, і це стосується як державних установ, так і бізнес-сектору [6]. До прикладу, дослідження стану кібербезпеки закладів вищої освіти України виявило, що найбільш поширеними інцидентами є зараження вірусами та шкідливим ПЗ – 39.4% випадків, злам акаунтів у навчальних онлайн-системах – 26.8%, фішингові атаки – 14.1% та витоки персональних даних – 12.7%. При цьому майже половина опитаних, а точніше 49.3%, оцінила свій рівень обізнаності з кібербезпекою як середній, а понад 28% – як низький [6], що показує наявність суттєвих прогалин у освітньому та організаційному аспектах безпеки у статистичному розрізі освітян. Відслідковується і загальносвітова тенденція: людський фактор та обізнаність користувачів є критично важливими елементами кібербезпеки нарівні з технологічними рішеннями. В умовах війни перед бізнесом стоїть задача посилювати кіберзахист не лише заради власної інформаційної безпеки, але й з міркувань національної безпеки. Українська наукова спільнота приділяє особливу увагу питанням захисту критичної інформаційної інфраструктури та оцінки потенційних наслідків кібератак. Зокрема, досліджується вплив кібератак на безперервність бізнес-процесів, репутацію, фінансову стійкість підприємств та національну безпеку загалом [7]. Таким чином, поняття кібербезпеки українського підприємства включає додатковий шар викликів, пов’язаних з протидією високорівневим загрозам у кіберпросторі в умовах гібридної війни.

Щоб уникнути термінологічної плутанини, доцільно розмежувати інформаційну безпеку та кібербезпеку. У практиці менеджменту безпеки інформаційна безпека зазвичай фокусується на захисті інформації та процесів її обробки з позицій класичної тріади CIA: confidentiality (конфіденційність), integrity (цілісність), availability (доступність). У стандартизації головним інструментом організаційного впровадження виступає ISMS (Information Security Management System) – система управління інформаційною безпекою. ISO/IEC 27001 визначає вимоги до побудови та постійного вдосконалення ISMS, придатної для організацій будь-якого розміру й сфери діяльності [8].

Рис. 1.1. Зображення класичної тріади CIA

Ризикоорієнтований характер підходу тут є ключовим, адже управління безпекою розглядається як безперервний цикл, а не разове впровадження контролю. Стандарт ISO/IEC 27005, який адаптує загальні принципи ISO 31000 до контексту інформаційної безпеки, прямо описує повний цикл ризик менеджменту: оцінювання, оброблення, комунікація, моніторинг і перегляд. Це важливо методологічно для подальшої формалізації оцінки рівня кібербезпеки підприємства [9]. Також ISO/IEC 27000 задає базову термінологію та огляд сімейства стандартів ISMS, що є необхідним для коректного використання понять у подальшій роботі [10].

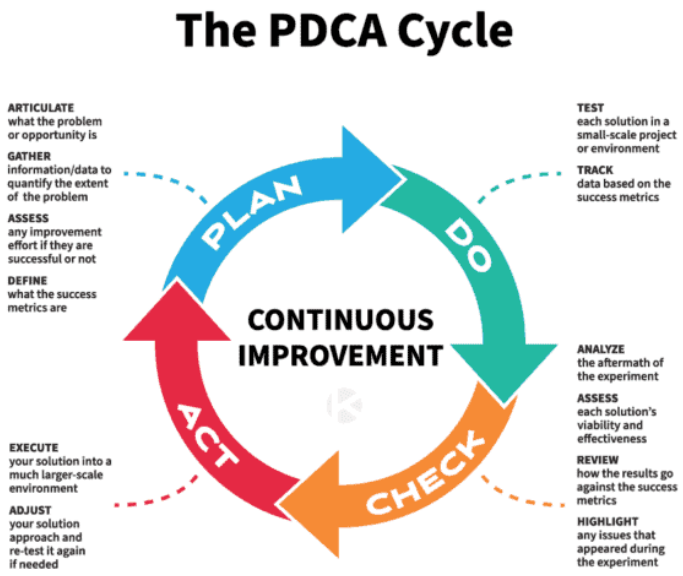

Рис. 1.2. Цикл Демінга, Плануй-Роби-Перевіряй-Роби

Отже, кібербезпека підприємства у межах цієї роботи доцільно трактувати як інтегровану організаційну здатність забезпечувати прийнятні рівні конфіденційності, цілісності та доступності критичних активів і сервісів підприємства в кіберпросторі, одночасно підтримуючи керованість ризиків, відповідність вимогам і готовність до інцидентів, враховуючі усі фактори.

У сучасних дослідженнях дедалі частіше підкреслюється, що виключно технократичні оцінки погано пояснюють реальну кіберстійкість організацій, бо інциденти виникають на стику політик, поведінки персоналу, процесів і технологій. Показовою є робота з розробки та пілотування моделі зрілості BYOD-безпеки у лікарнях, де структура оцінювання прямо включає технічні, політичні та людські виміри і будується за рівнями зрілості [11]. Дане дослідження підтримує тезу, що оцінка кібербезпеки підприємства має бути багатовимірною, а інструмент збору даних повинен охоплювати не лише технічні контролі, а й управлінські та поведінкові практики.

В українському науковому контексті актуальність інтелектуалізації підходів також підтверджується дослідженнями щодо застосування нейросимвольних моделей для виявлення й протидії кіберзагрозам у критичних кіберфізичних системах [12]. Частіше подібні роботи фокусуються на детекції та реагуванні, але вони й підсилюють загальну аргументацію – для складних доменів потрібні гібридні, контекстні підходи, що комбінують формальні правила й адаптивні моделі.

Окремо слід пояснити поняття кіберстійкості. У літературі кіберстійкість зазвичай описує здатність системи або організації передбачати, витримувати, відновлюватися та адаптуватися до несприятливих умов, атак або компрометацій. Одне з наукових джерел формалізує термін таким чином, що кіберстійкість розглядається як багатокомпонентна властивість, тісно пов’язана з безперервністю та адаптивністю [13]. Для підприємства це означає, що високий рівень кібербезпеки не досягається виключно наявністю певних контролів, а має включати готовність і спроможність підтримувати критичні функції під час інцидентів і після них, що лягає в основу напрацювання методу з подальшою програмною реалізацією. Задача не лише виміряти стан, а й сформувати траєкторію покращення через надання певних рекомендацій та дерева рішень.

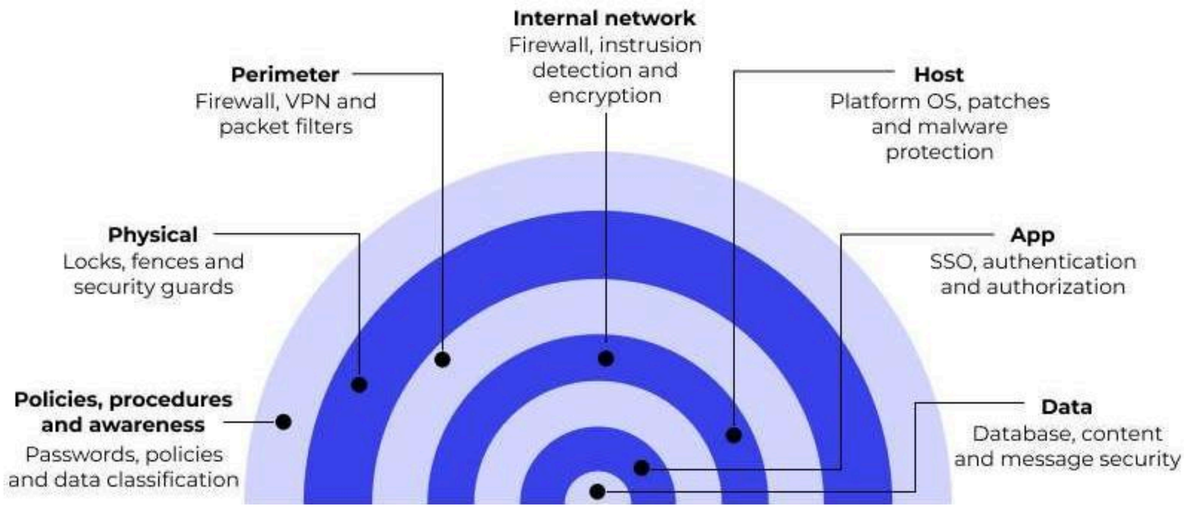

Рис. 1.3. Різні рівні та шари кібербезпеки

Поняття рівня кібербезпеки підприємства в академічному сенсі доцільно інтерпретувати як ступінь досягнення визначених результатів кібербезпеки та ступінь зрілості практик, що забезпечують такі результати. Важливим стає термін maturity model (модель зрілості) – формалізований опис еволюції практик за рівнями, наприклад, від ad-hoc рішень до оптимізованих систем, часто поділений на певні домени. Тут термін доменів використовується в розумінні груп однорідних практик, тобто домен управління, політики доступу, реагування.

Моделі зрілості широко застосовуються для оцінювання готовності організацій, але водночас демонструють неоднорідність доменів і підходів до вимірювання [14]. До речі, оглядовий SoK-аналіз обмежень Cybersecurity Capability Maturity Models підкреслює типові прогалини: надмірну орієнтованість на complience, недостатню інтеграцію людських і організаційних факторів та складність коректної кількісної агрегації [15]. Власне тому обгрунтована необхідність в методі, що зберігає структурність доменів і рівнів, але забезпечує коректну кількісну інтерпретацію та може після аналізу пропонувати практичні рекомендації, ще й з урахуванням контексту підприємства.

У сучасних методологіях оцінювання кіберризиків та спроможностей використовують різні прояви опитування як інструмент стандартизованого збору даних про практики, процеси й наявні контролі, особливо коли організація не може надати повні телеметричні дані або коли оцінюються управлінські практики. У PCSRA – методології прогнозованого оцінювання ризику початковим етапом розглядається використання мінімального за обсягом опитувальника, сформульованого зрозумілою для персоналу підприємства мовою. Такий інструмент дозволяє швидко отримати базову картину стану безпеки та визначити пріорітетні напрями для подальшого поглибленого аналізу та управлінських рішень [16].

Далі виникає задача агрегації відповідей у формалізований показник. У наукових роботах це часто вирішується шляхом використання багатокритеріальних підходів та моделі невизначеності. Показовою є концепція Fuzzyfortify, де поєднано Fuzzy AHP, тобто нечіткий метод аналізу ієрархій для визначення ваг критеріїв, та нечітке виведення для інтегральної оцінки ризику у складному середовищі [17]. Сучасні динамічні моделі кіберризику комбінують різні математичні моделі. Наприклад, інтеграція марковських ланцюгів, що описсують переходи між станами атаки або компрометації і байєсівських мереж, які відображають імовірні причинно-наслідкові залежності, для отримання чутливих до фіксації часу оцінок ризику [18]. Таким чином демонструється підхід до оцінювання, який має бути не просто середнім значенням за показниками, а математично обґрунтованим механізмом інтеграції багатьох критеріїв враховуючи різні фактори та невизначеність.

Нарешті компонента інтелектуальних рекомендацій та дерева рішень логічно випливає з того, що оцінка кібербезпеки мала би не лише констатувати стан, а й полегшувати прийняття управлінського рішення: що робити далі і в якій послідовності, з урахуванням галузі, масштабу і ресурсних обмежень. Саме це зближує оцінювання з практиками NIST CSF 2.0, де результати організовані так, щоб бути застосовними для управління ризиками й пріоритизації заходів на рівні керівництва та управління процесів [3].

1.2. Аналіз механізмів забезпечення кібербезпеки підприємства

Кібербезпека підприємства не зводиться до набору технічних засобів захисту. У сучасних підходах вона трактується як керована організаційна здатність, що формується через систему взаємопов’язаних механізмів: управлінських (governance), процесних (risk management, change management, incident management), кадрових та технічних, тобто контролі, моніторинг, реагування. Така багаторівнева природа добре корелює з логікою стандарту NIST CSF 2.0, де результати кібербезпеки організовані навколо функцій Govern–Identify–Protect–Detect–Respond–Recover, а Govern визначає рамку відповідальностей, політик, нагляду й кризові стратегії для всіх інших функцій.

Рис. 1.4. Візуалізація функціональної моделі CSF 2.0

Кібербезпекове управління у контексті підприємства означає систему управлінських рішень і правил, які визначають:

- цілі безпеки та допустимий рівень ризику,
- ролі, відповідальності та повноваження,
- політики та контроль їх виконання,
- пріоритизацію інвестицій у безпеку відповідно до бізнес-цілей.

У CSF 2.0 прямо підкреслюється, що управлінські обговорення включають узгодження стратегій управління ризиками, зокрема й ризиками ланцюга постачання, визначення ролей і політик та організаційний нагляд [3].

Для механізмів забезпечення кібербезпеки це принципово, оскільки безпека стає керованим циклом, яке включає планування, реалізацію, перевірку та поліпшення. Важливо також, що ISO/IEC 27005 надає структурований підхід до управління ризиками інформаційної безпеки, тобто ідентифікацію, оцінювання та оброблення, підтримуючи реалізацію ISMS на ризикоорієнтованій основі.

В корні майбутнього програмного рішення governance логічно перетворюється на домен критеріального опитування типу Security Policies & Governance, тобто перевіряє наявність політик, призначення ролей, регулярність перегляду, прив’язкц до ризиків, контроль виконання. Таким чином оцінюватиметься не наявність документу, а статус роботи налаштованого управлінського механізму.

Термін ризик у кібербезпеці доцільно пояснювати як поєднання або функцію ймовірності небажаної події та наслідків для підприємства. NIST SP 800-30 Rev.1 формалізує ризик-оцінювання як частину загального управління ризиками на різних рівнях, як організації, як бізнес-процесу, як інформаційної системи та як основу для управлінських рішень щодо вектору дій у відповідь на виявлені ризики [19].

Практична реалізація ризикоорієнтованості здійснюється через контролі (security controls) – організаційні, операційні та технічні заходи або контрзаходи, які зменшують імовірність успішної атаки, обмежують її наслідки або підвищують здатність до виявлення й відновлення. Каталог NIST SP 800-53 Rev.5 описує контролі як гнучкі та налаштовувані елементи, що реалізуються у межах організаційного процесу управління ризиками [20].

Ми поступово підходимо до формулювання ідеї нового методу: відповіді в опитувальнику фактично є якісно-кількісними спостереженнями про стан механізмів і контролів, а математична формула – спосіб коректної агрегації під невизначеністю, що узгоджується з сучасними нечіткими моделями оцінювання, як наприклад у раніше згаданому Fuzzyfortify.

Технічна площина механізмів забезпечення кібербезпеки підприємства полягає в тому, щоб більшість базових ризиків нейтралізувалися стандартними й повторюваними практиками: інвентаризація активів, керування конфігураціями, контроль облікових записів, сегментація мережі, резервне копіювання, захист електронної пошти й вебу. Вичерпним прикладом це добре відображає CIS Critical Security Controls, які позиціонуються як пріоритизований набір захисних заходів проти найпоширеніших атак. Вони мають практичну цінність як мінімальна необхідна база, що стосується малого та середнього бізнесу та невеликих організацій [21].

На рівні доменної моделі для критеріального опитування це зазвичай розкладається на домени на кшталт Asset & Configuration Management, Vulnerability Management, Access Control, Backup & Recovery. Оцінювання має враховувати не лише наявність технологій, а й керованість: чи визначені власники процесів за RACI моделлю, чи є регламент змін, чи вимірюється виконання, чи ведеться доказова база.

Окремим критичним механізмом є безперервний моніторинг безпеки. NIST SP 800-137 визначає ISCM як підтримання постійної обізнаності щодо стану безпеки, вразливостей і загроз з метою підтримки рішень управління ризиками [22]. У прикладному сенсі це неможливо без якісного лог-менеджменту, чому присвячені рекомендації NIST SP 800-92 [23].

Для майбутньої реалізації це має пряме відношення, ми маємо врахувати що частина опитування повинна зібрати дані не лише стосовно наявності засобів моніторингу, а й процесну зрілість – хто переглядає події, які метрики або сповіщення, яка частота, як відпрацьовуються спрацювання, чи є ретроспективний аналіз. В цій зоні застосування штучного інтелекту для пояснення необхідних мір може бути найбільш ефективним: наприклад, запропонувати мінімальний набір журналів і подій для галузевого контексту, або запропоноване дерево рішень, яке покроково пояснення побудову логування та реагування при обмежених ресурсах.

Інцидент кібербезпеки – це подія або серія подій, що порушують політики безпеки або загрожує конфіденційності, цілісності чи доступності активів. Механізм реагування полягає у здатності підприємства діяти за попередньо визначеним сценарієм: підготовка, виявлення, аналіз, стримування, усунення, відновлення, післяінцидентна ретроспектива.

Актуальною тут є NIST SP 800-61 Rev.3 від 2025 року, який прямо позиціонує реагування на інциденти як частину діяльності з управління кіберризиками у логіці CSF 2.0 та надає рекомендації для підвищення ефективності виявлення, реагування й відновлення [24]. Відповідно, можемо окреслити що майбутнє рішення може не лише оцінювати, а й адаптувати шаблони під контекст підприємства та пропонувати дерево кроків, які підвищують оцінку домену. Треба тримати в фокусі різницю між простим самоаудитом і якісно сформульованим науковим оцінюванням. NIST SP 800-53A Rev.5 пояснює, що оцінювання контролів не має бути чек-листом та формальною процедурою заради аудиту та звітування, натомість ключовим механізмом перевірки, що обрані контролі реально реалізовані та досягають поставлених цілей [25].

Оскільки підприємство є соціотехнічною системою, механізми кібербезпеки мають включати людський вимір: навчання, усвідомленість, дотримання політик, відповідальність за безпечну поведінку. Систематичний огляд моделей зрілості інформаційної обізнаності (ISA maturity) показує, що зрілість у цій сфері описується еволюцією організаційних практик, вимірюваністю й інтеграцією в процеси [26]. Це означає необхідність виокремити домен на кшталт Security Awareness & Culture, який оцінює регулярність, рольову диференціацію з точки зору доступів, симуляції фішингу, фіксація проходження, метрики та коригувальні дії.

Для більшості підприємств частина критичних процесів залежить від хмарних провайдерів, підрядників, SaaS або ERP, постачальників обладнання та сервісів. Тому кіберризики ланцюга постачання (C-SCRM) мають бути вбудовані у загальне управління ризиками. NIST SP 800-161 Rev.1 описує практики управління кіберризиками в ланцюгу постачання, а NIST випустив окремі швидкі рекомендації, які показують, як використовувати CSF для побудови C-SCRM спроможностей [27]. У європейському регуляторному полі ENISA також підкреслює необхідність політик і практик supply chain security в межах вимог управління кіберризиками [28]. Цей механізм корисно відобразити і в нашому методі певним окремим доменом Third-Party & Supply Chain Security, де оцінюються вимоги до підрядників, контроль доступів, оцінка критичності постачальників, процеси погодження й моніторингу.

Для України, з огляду на умови гібридної війни та підвищену активність кібератак проти різних секторів, питання механізмів кібербезпеки має додатковий вимір: необхідність швидкої адаптації та поєднання формальних політик із інтелектуальними інструментами виявлення або протидії.

Отже, механізми забезпечення кібербезпеки підприємства утворюють цілісну систему, де:

- governance та ISMS задають рамку відповідальностей і циклу поліпшення;
- ризикоорієнтоване управління визначає пріоритети контролів і дозволяє формалізувати оцінку;
- технічні контролі, моніторинг і реагування дають вимірювані операційні результати;
- оцінювання контролів має спиратися на доказовість, а не лише самодекларацію;

- людський фактор і supply chain формують системні ризики, які не можна забезпечити лише на технічному рівні.

1.3. Дослідження методів оцінки рівня кібербезпеки підприємства

Під методом оцінки рівня кібербезпеки підприємства у цьому дослідженні розумітимемо формалізовану процедуру, яка перетворює спостережувані характеристики у вимірюваний результат: профіль відповідності, рівень зрілості, кількісний показник ризику або інтегральний бал. Важливо розрізняти: оцінювання як вимірювання стану (assessment), аудит як перевірку відповідності вимогам (audit), сертифікацію як формальний результат незалежної перевірки. Звідси випливає, що частина підходів дає полярний результат, тобто так або ні, або перелік невідповідностей, орієнтовним на аудит, а інша — числовий бал або рівень та траєкторію покращення.

Найпоширеніша практична лінія оцінювання це відповідність вимогам. Вона опирається на стандарти та фреймворки, де рівень безпеки інтерпретується як ступінь реалізації контрольних практик. Показовий приклад — NIST Cybersecurity Framework 2.0, де результати організовано за функціями Govern, Identify, Protect, Detect, Respond, Recover; фреймворк не нав'язує єдиної формули, але задає структуровані очікувані результати, що зручно для побудови профілів і gap-аналізу [3].

Інша домінантна основа це ISO/IEC 27001:2022, яка має аналог в Україні: гармонізований стандарт ДСТУ ISO/IEC 27001:2023. Його логіка полягає у створенні та безперервному поліпшенні ISMS, де ризикоорієнтованість закладена як принцип управління. Дослідницькі роботи, що аналізують оновлення ISO/IEC 27001:2022 та його застосування в управлінні кібербезпекою підприємства, підтверджують тенденцію до зближення ISO-логіки з підходами NIST, в моменті узгодження контрольних доменів і термінології управління ризиком.

Для прикладної самооцінки малого та середнього бізнесу часто використовують також CIS Critical Security Controls v8.1, що є пріоритизованим набором safeguard, який зручний коротким списком першочергових дій і добре піддається перевірці виконання.

Сильні сторони комплаєнс-підходів полягають у високій порівнюваності результатів між компаніями, аудитопридатністю і юридичній інтерпретованості, сильним зв'язком із governance. Слабкими сторонами в контексті задачі створення інтегральної оцінки рівня кібербезпеки є те, що результат часто дискретний або фрагментований, складно коректно перетворити чекліст на певне число без додаткової моделі ваг і невизначеності, низька чутливість до контексту, тобто ймовірніше не враховується галузь, розмір підприємства, загрози, що існують тут і зараз, якщо не доповнювати модель.

Модель зрілості це шкала рівнів, яка відображає, наскільки системно й керовано організація виконує практики кіберзахисту. Систематичні огляди 2025 року показують, що ландшафт моделей зрілості є фрагментованим: від універсальних до доменних або секторо-специфічних. При цьому типова реалізація це звичайне критеріальне самооцінювання з подальшим агрегуванням у рівні [29]. З наукової точки зору важливо, що сучасні maturity-підходи еволюціонують від простих рівнів до формалізованих багатокритеріальних процедур, зокрема з нечіткими перевагами та AHP-подібними механізмами, аби краще працювати з неоднозначними відповідями та неповними даними.

Перевага моделей зрілості у тому, що вони природно підтримують ідею створення дорожньої карти покращень. Обмеження в тому, що типова зрілість все ще часто є: надто суб'єктивною, наприклад умовно, найвищий рівень 4, не пояснює, чому і що саме робити далі, та слабко динамічною, бо замало враховуються зміни загроз і часу.

Ризикоорієнтовані підходи трактують результат оцінювання як величину ризику, що пов'язує загрози, вразливості, ймовірності та наслідки. Класичне формальне визначення ризику в кількісному аналізі часто подають як набір трійок: сценарій, ймовірність, наслідок. Це дозволяє явно відображати невизначеність і множинність сценаріїв [30]. У практиці підприємств цей клас методів дає сильний зв'язок із бізнес-наслідками у фінансах, безперервності процесів, репутації, але стикається з методологічною проблемою: точні оцінки ймовірності або впливу рідко доступні, особливо для малого бізнесу, де немає історичних даних інцидентів і команди аналітиків.

Отже, ризик-методи найкраще працюють, коли їх підкріплюють статистикою, або доповнюють механізмами, що формалізують експертні лінгвістичні оцінки, що дозволяють нам нечіткі та MCDM-підходи.

Багатокритеріальне прийняття рішень (MCDM) – клас методів, де результат є агрегуванням багатьох критеріїв із вагами. Для нашого методу і програмної реалізації, де результат опитування крізь певний набір формул отримує свій числовий бал, критичною є задача визначення ваг доменів та критеріїв в них. Саме її класично вирішує AHP (Analytic Hierarchy Process): критерії структуризуються ієрархічно, ваги визначаються попарними порівняннями, а якість експертних суджень контролюється показником узгодженості [31]. Однак опитування з кібербезпеки часто містять нечіткі судження, наприклад «частково впроваджено», «залежить від підрядника», «нерегулярно». Для цього застосовують нечіткі множини: на відміну від бінарної належності, елемент може належати множині з мірою від 0 до 1, що дає математичну основу для обробки лінгвістичних значень [32].

У дослідженнях ризикооцінювання це реалізують як нечіткі MCDM системи: наприклад, моделі, що поєднують MCDM із нечіткою логікою для оцінки ризику в контексті ISMS та ISO-підходів. Інший напрям – нечіткі моделі аналізу ризику, які формалізують причинно-наслідкові ланцюги подій та дають числові результати при дефіциті точних даних [33]. У прикладній кібербезпеці існують і гібриди Fuzzy з AHP для отримання інтегральних оцінок і ранжування покращень.

Показовий тренд це інтеграція AHP із нечіткими процедурами та операторами агрегування у задачах вимірювання зрілості кібербезпеки, що насправді є близьким до рішення, яке ми шукаємо [34].

Суть переваги цього класу методів в тому, що вони одночасно дають числовий бал, коректно працюють з невизначеністю у відповідях, дозволяють вбудувати ваги доменів та галузеву специфіку, не вимагаючи великих масивів історичних даних.

Найвагоміше обмеження полягає у методично обґрунтованому налаштуванні шкал, функцій належності й правил агрегування.

Коли мета дослідити не лише поточний стан, а й динаміку ризику, наприклад зміна експозиції через нові вразливості, топологію мережі, ланцюги атак, використовують ймовірнісні графові моделі. Баєсівська мережа це графова модель причинно-ймовірнісних залежностей, що дозволяє оновлювати ймовірності при надходженні нових свідчень. Її теоретичну основу систематизовано в класичній праці Пірла про ймовірнісне міркування та мережі правдоподібного висновування [35].

У 2025 році з’являються прикладні моделі, що поєднують марковські ланцюги, які використовуються для моделювання переходів між станами атаки або компрометації, та баєсівські мережі для залежностей між вразливостями, конфігураціями і подіями, роблячи оцінювання ризику time-sensitive, в сенсі динамічної оцінки кіберризику [36].

Перевага цього класу у високій аналітичний точності у складних системах та можливість обґрунтовано моделювати ланцюги атак. Недолік у достатньо високих вимогах до вхідних даних і моделі середовища, що у випадку малого та середнього бізнесу часто робить такі методи надмірно важкими для імплементації.

Прогнозування інцидентів (incident forecasting) це клас підходів, де оцінка включає не лише теперішній стан, а й найбільш потенційний, використовуючи дані про події, вразливості та активність противника. У науковій літературі ця лінія розвивається на перетині Machine Learning, Data Mining та кібербезпеки; ранні оглядові роботи про data-driven prediction систематизують методи та їхні обмеження, зокрема проблеми даних, узагальнення і перенавчання [37].

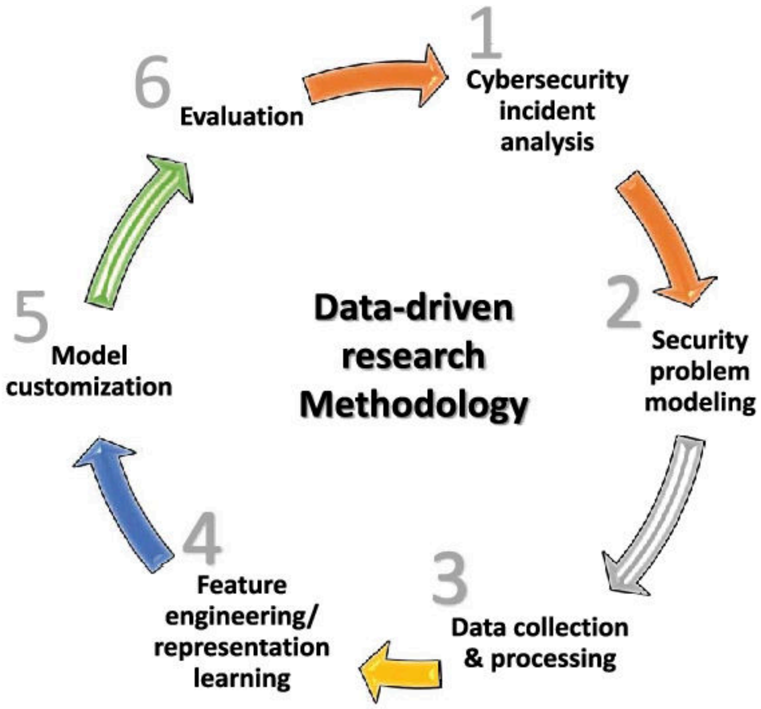

Рис. 1.5. Методологія дослідження, керованих даними

Для підсилення прогнозу активно використовують Cyber Threat Intelligence (CTI) – структуровані відомості про противників, тактики або техніки та індикатори компрометації. Огляд CTI-досліджень показує, що ефективність CTI залежить від якості процесів збирання, нормалізації та здатності інтегрувати її у рішення з управління ризиком.

Існує й показовий приклад інтеграції, де оцінка трансформується у певний прогноз: методологія поєднує прогнозування технік, на кшталт MITRE ATT&CK, і кількісну оцінку контролів через автоматизоване мапування й категоризацію ризику [38]. Це концептуально близько до задуму, що ШІ міг би на основі відповідей з опитування та врахування контексту підприємства надавати рекомендації і формувати дерево рішень, але тут існує наукова проблема: пояснюваність і відтворюваність рекомендацій. Без прозорого послідовного зв’язку, де відповідь класифікується за певним правилом і з цього випливає певна рекомендація, рішення можуть бути недовіреними для аудиту.

Задачею нашого методу є поєднання формальної моделі оцінювання, створення дерева рішень, як навігації для покращення, та ШІ-інтерпретації з урахуванням контексту підприємства. Цю ідею добре підтримує напрям нейросимвольних підходів, що комбінує в собі нейронний та символічний, що дає баланс між точністю та пояснюваністю. Це дозволяє оперувати мовою, даними, правилами, логічними структурами та графами знань. Для критичних інфраструктур такі підходи досліджують як спосіб виявлення та запобігання загрозам із підвищенням інтерпретованості висновків [39].

Для постановки задачі це означає, що символьна частина може бути реалізована як контрольна модель: спираючись на стандарти та практики NIST та CIS, ми можемо виокремити потрібні домени та критерії, а з них формулювати вагові коефіцієнти, певні правила та створювати гілки дерева рішень. Нейронна частина може виступити модулем, що перетворює контекст галузі, враховує розмір підприємства, регуляторні вимоги, критичні активи у персоналізовані пріоритети та пояснення.

Порівняння має бути не лише описовим, а й конструктивним, щоб наочно можна було вивести вимоги до нового методу. Таким чином доцільно оцінити підходи за низкою наскрізних критеріїв: кількісність, робота з невизначеністю, динамічність у часі, адаптація до контексту, аудитопридатність, автоматизація, дієвість рекомендацій.

Таблиця 1.1

Порівняльний аналіз методів оцінки кібербезпеки підприємства

Клас методів	**Кількісний інтегральний показник**	**Невизначеність у відповідях**	**Динамічність у часі**	**Адаптація до контексту**	**Аудит або стандарти**	**Рекомендації наступних кроків**
Комплаєнс за стандартами ISO, NIST, CIS	обмежено, частіше це чекліст	слабко	слабко	середньо	високо	середньо
Моделі зрілості	середньо або високо	середньо	слабко або середньо	середньо	середньо	середньо
Ризикооцінювання	високо, ризик як величина	середньо, але складно калібрувати	середньо	середньо	високо	середньо
Поєднані FuzzyAHP з MCDM	високо	високо	середньо	високо	середньо	високо, бо можна ранжувати пріоритети

Закінчення таблиці 1.1

Баєсова мережа або Марківські ланцюги	високо	високо	високо	низько або середньо	середньо	середньо
Machine Learning, CTI, прогнозування	високо, за наявності даних	середньо	високо	низько або середньо	середньо	середньо
Нейросимвольні	залежить від формалізації	високо	середньо або високо	високо	високо, за мапуванням до стандартів	Високо, бо має дерево рішень з поясненням

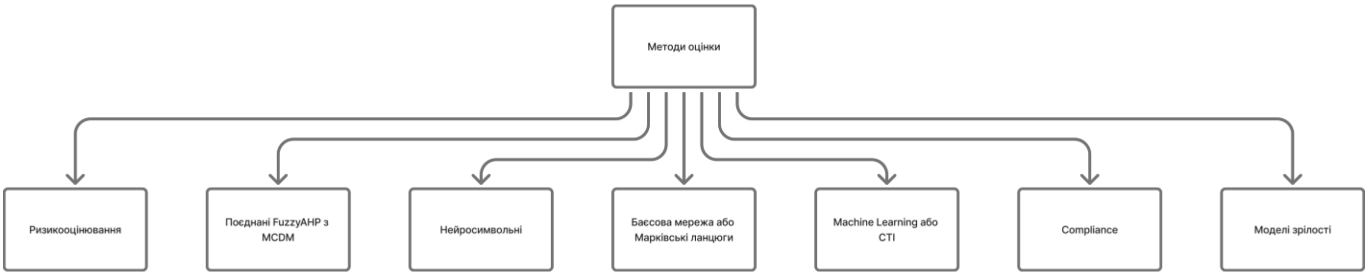

Рис. 1.6. Методи оцінки розглянуті в аналізі

Враховуючи порівняльний аналіз у табл. 1.1. робимо висновок, що жоден клас з проаналізованих підходів самостійно не закриває вимоги кількісної оцінки, невизначеності анкетних відповідей, галузевої контекстуалізації та пояснюваних рекомендацій. Тому методологічно обґрунтованою виглядає гібридна конструкція:

1. Опитування за доменами, що виступить в якості інструменту збору даних, узгоджений зі структурою NIST CSF 2.0 та стандарту ISO 27001.

2. Аналіз Ієрархій застосуємо для визначення ваги доменів та критеріїв із контролем узгодженості суджень, що дасть строгий механізм для різних галузей.

3. Нечітка логіка для перетворення лінгвістичних відповідей у числові значення без втрати змісту.

4. Нейросимвольний модуль рекомендацій, які перетікають у дерево рішень та враховують політики, плани реагування, рівні доступів, того. Залишаємо потенціал на інтеграцію пояснень з використанням ШІ, оптимізація під контекст підприємства, з можливістю завантажити артефакти і перерахувати оцінку після впроваджень. В даній роботі термін артефакти використовуватиметься в розумінні набору певних

документів, затверджених політик, правил, які певним чином регламентовані або задокументовані.

Потенційним покращенням в майбутньому може стати інтеграція сигналів часу з інтеграцією певних індикаторів, що підвищило би чутливості оцінки до актуальних загроз, навіть у режимі реального часу.

1.4. Постановка задач дослідження

Проведений у попередніх підрозділах аналіз показав, що кібербезпека підприємства є соціотехнічною керованою здатністю, яка проявляється через результати у доменах управління, захисту, виявлення, реагування та відновлення. Така логіка узгоджується з функціональною структурою NIST CSF 2.0. Одночасно стандарти сімейства ISO та IEC 2700x підкреслюють ризикоорієнтовану побудову та безперервне вдосконалення ISMS, що важливо для інтерпретації рівня кібербезпеки як динамічного стану, який можна вимірювати і цілеспрямовано підвищувати.

1.4.1. Наукова проблема та суперечність

Практика оцінювання кібербезпеки підприємств переважно реалізується як комплаєнс або аудит за стандартами, або самооцінюванням зрілості за доменними опитуваннями, або як кількісна оцінка ризику, що потребує даних. У реальних умовах більшості підприємств, особливо малого та середнього бізнесу, виникає суперечність: потрібен інтегральний, порівнюваний і керований показник, який легко отримати, але водночас він має бути математично коректним за наявності невизначених відповідей і повинен одразу породжувати рекомендації та траєкторію покращення. Саме ця суперечність і становить ядро наукової проблеми: як побудувати метод оцінювання, що одночасно є кількісним, контекстно-адаптивним, придатним до аудиту, придатним для автоматизації у вигляді застосунку.

1.4.2. Об'єкт і предмет дослідження

Об'єктом дослідження є процеси та механізми забезпечення кібербезпеки підприємства як соціотехнічної системи.

Предметом дослідження є методи та моделі кількісної оцінки рівня кібербезпеки підприємства на основі доменної структури критеріїв, їх вагового узгодження та обробки невизначених лінгвістичних оцінок із подальшим формуванням рекомендацій.

1.4.3. Мета дослідження

Мета дослідження полягає у розробленні та обґрунтуванні методу оцінки рівня кібербезпеки підприємства, який на основі відповідей, отриманих через інтерфейс опитування, забезпечує розрахунок інтегрального показника, а також формує контекстно релевантні рекомендації й дерево рішень для підвищення цього показника.

1.4.4. Завдання дослідження

Для досягнення мети необхідно розв'язати такі взаємопов'язані завдання:

1. Сформувати доменну модель оцінювання кібербезпеки підприємства із прив'язкою до результатів та практик NIST CSF 2.0 і до ризикоорієнтованої логіки ISO/IEC 27001/27005, щоб забезпечити змістову валідність.

2. Розробити опитування як інструмент збору даних, визначивши шкали відповідей, правила доказовості, та процедуру врахування контексту підприємства.

3. Побудувати математичну модель агрегації: визначити ваги доменів та критеріїв методом AHP, визначити нечіткі множини та функції належності для лінгвістичних відповідей та правила перетворення в числові значення, вивести формулу інтегрального показника рівня кібербезпеки.

4. Сформувати механізм рекомендацій і дерева рішень, який пов'язує критичні критерії з пріоритетними заходами, шаблонами артефактів та обґрунтуванням, використовуючи нейросимвольний підхід як компроміс між гнучкістю й відтворюваністю.

5. Реалізувати MVP застосунку: створити інтерфейс опитування, можливість розрахувати оцінку, вивід рекомендацій дерева рішень.

ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі обґрунтовано, що кібербезпека підприємства є багатовимірною соціотехнічною здатністю, що розглядається крізь призму управління, ризикоорієнтованості, наявності й керованості контролів, спроможностей моніторингу, реагування та відновлення. Фреймворк NIST CSF 2.0 задає структурну модель результатів безпеки, придатну для формування доменів оцінювання, тоді як стандарти ISO/IEC 27001 та ISO/IEC 27005 забезпечують методологічний фундамент ризикоорієнтованого управління й безперервного вдосконалення ISMS.

Порівняльний аналіз підходів до оцінювання показав, що методи комплаєнсу мають високу аудитопридатність, але слабко дають інтегральний показник і контекстну адаптацію; моделі зрілості природньо підтримують дорожню карту розвитку, але часто страждають від суб’єктивності та низької математичної строгості; кількісні моделі ризиків потребують даних і складні для малого бізнесу. Найбільш перспективним для задачі дисертації є гібридний підхід із використанням AHP для визначення ваг та нечіткої логіки для обробки лінгвістичних відповідей, доповнений нейросимвольним модулем рекомендацій і дерева рішень, що забезпечує вимірювання та кероване покращення стану кібербезпеки.

У результаті сформульовано мету та систему завдань дослідження, що виступить підгрунтям до розробки математичної моделі, тобто формули інтегральної оцінки, проектування структури опитувальника, побудови механізму рекомендацій та дерева рішень і реалізації MVP застосунку.

РОЗДІЛ 2
РОЗРОБКА МЕТОДУ ОЦІНКИ РІВНЯ КІБЕРБЕЗПЕКИ ПІДПРИЄМСТВА

2.1. Розробка комплексної моделі оцінювання рівня кібербезпеки підприємства

Під комплексною моделлю оцінювання надалі розумітимемо формалізований опис того, що саме вимірюється, звідки беруться дані, як ці дані перетворюються на числовий результат, і як результат породжує керовані управлінські дії. Це підхід відповідатиме ідеї outcome-driven оцінювання, що характерно для NIST CSF 2.0, де структура функцій і категорій задає структуру для вимірюваних результатів, але не нав'язує єдиного способу обчислення [3].

У моделі вводимо чотири рівні абстракції:

1. Домени, позначимо їх літерою D – верхній рівень структурування, який відображає логіку управління та технічних або організаційних практик;

2. Критерії, позначимо літерою C – конкретні вимірювані ознаки всередині домену. До прикладу, регулярний перегляд прав доступу, сегментація мережі, тестування резервних копій, наявність плану реагування на інциденти.

3. Контекст підприємства, позначимо літерою K – параметри, що впливають на пріоритетність доменів і критеріїв. Серед них: галузь, розмір, критичні активи, регуляторні вимоги, модель використання хмарних сервісів.

4. Виходи, позначимо літерою O — інтегральний бал $S \in [1;100]$, профіль доменів, перелік вузьких місць (bottlenecks) та дерево рішень і рекомендацій.

Ключова вимога до майбутнього рішення інтерпретується як нормалізований інтегральний показник, де 100 означає досягнення цільового профілю, по суті сигналізує про зрілість за всіма критеріями з урахуванням вагових коефіцієнтів, а значення ближче до 1 говорить про критичну недостатність практик.

Щоб оцінка мала змістову валідність, домени доцільно виокремлювати на визнаних структурах: NIST CSF 2.0 забезпечує функціональний розподіл outcomes і природно підтримує домен Govern як управлінський шар. ISO/IEC 27002:2022 дає еталонний перелік контрольних практик та імплементаційних настанов, які можуть бути використані як джерело критеріїв. ISO/IEC 27005 задає повний цикл керування ризиками в контексті інформаційної безпеки, починаючи з оцінки та обробки, завершуючи комунікацією, моніторингом та переглядом. Це необхідно для прямої залежності і наслідку управлінських рішень від отриманої оцінки. CIS Controls v8.1 корисний як каталог практичних дій, орієнтований на пріоритизацію та вимірюваність safeguards.

Перед створенням моделі важливо уточнити, що національний стандарт ДСТУ ISO/IEC 27001:2023 є прийнятим як ідентичний ISO/IEC 27001:2022, що підтверджується наказом про прийняття національних стандартів. Це дозволяє легітимно використовувати термінологію ISO в моделі та коректно посилатися на вимоги в національному середовищі.

Майбутній формат опитування має бути повноцінним інструментом формалізації стану контролів. Щоб забезпечити відтворюваність, кожен критерій повинен мати:однозначний зміст, тобто що саме ми будемо перевіряти, шкалу відповіді, тобто як саме вимірюємо, правила інтерпретації, та мати потенціал на впровадження можливості зчитати ознаку доказовості, тобто чи існує певний артефакт чи політика, що підтверджує відповідь.

Практично доцільно застосувати лінгвістичну шкалу з 5 рівнями, починаючи від «не впроваджено» до «повністю і регулярно виконується», оскільки вона достатньо гнучка для самооцінки і водночас зручна для подальшого нечіткого відображення у числа.

На практиці, для MVP доцільно зробити домени не надто дрібними, але й не надто широкими. Раціонально буде прив’язатися до референтних структур: NIST CSF 2.0 задає Govern–Identify–Protect–Detect–Respond–Recover верхним рівнем очікування від організації; ISO/IEC 27002:2022 на рівні контролів групує практики у організаційні, орієнтовні на персонал, фізичні, технологічні; CIS Controls v8.1 задає 18 пріоритизованих контролів, які придатні для організацій меншого розміру, і зручні для мапування рекомендацій на конкретні safeguard-и.

Це зводиться до створення збалансованого варіанту із 12 доменів, який покриватиме і відповідність політикам, налаштуванням рівнів доступів, планів реагування, переглядатиме моніторинг та справність організації до подальшого відновлення в разі кіберінциденту відновлення.

Таблиця 2.1

Домени методу та мапування до стандартів

Домен	Призначення в оцінюванні	Функція CSF 2.0	Тип контролів ISO/IEC 27002	Опорні CIS Controls
D1 Governance & Policies	політики, ролі, контроль виконання	Govern	Organizational	17, 18 та governance акценти v8.1
D2 Risk Management & Compliance	ризики, відповідність вимогам, безперервне поліпшення	Govern та Identify	Organizational	18, 15
D3 Asset & Service Inventory	об’єкт захисту: активи, ПЗ, SaaS, дані	Identify	Organizational та Technological	1, 2
D4 Identity & Access Management	доступи, MFA, least privilege, PAM	Protect	Technological	5, 6
D5 Awareness & HR Security	людський фактор: навчання, JML, дисципліна	Govern та Protect	People	14
D6 Data Protection & Privacy	захист даних, шифрування, зберігання, приватність	Protect	Organizational та Technological	3
D7 Secure Config & Patch Vulnerability Management	базові конфігурації, патчі, уразливості	Protect	Technological	4, 7
D8 Network Security	периметр, фаєрвол, сегментація, VPN	Protect та Detect	Technological	12, 13
D9 Endpoint & Application Security	захист endpoint, базова безпека застосунків	Protect та Detect	Technological	10, 16
D10 Logging & Monitoring	логування, алерти, моніторинг ISCM	Detect	Technological	8, 13
D11 Incident Response	план IR, ролі, тренування, lessons learned	Respond	Organizational	1
D12 Backup, Recovery, Resilience	резервування, тест відновлення, BCP/DR	Recover	Technological та Organizational	11

Критерій це ознака виконання практики. Індикатор – конкретизація критерію, яку дослівно можна запитати та перевірити. До прикладу «чи проводиться перегляд доступів не рідше раз на квартал?». У прикладному оцінюванні безпеки слабким місцем самооцінки є свідомово завищені бажані відповіді, тому в модель вводиться поняття доказовості, тобто перевіряється існування артефакту, який підтверджує відповідь. У методі доказ грає коригуючу роль, є мультиплікатором, який зменшує внесок критерію, якщо відповідь не підтверджується жодним артефактом.

Наведемо далі достатній набір критеріїв для MVP, пропонується напрацювати 6 критеріїв на кожен домен, отримавши в цілому 72 питання. Це компромісне число, за яким можна буде отримати стійкий інтегральний бал, визначити слабкі місця та коректно формувати дерево рішень. Тому відобразимо їх у таблицях 2.2 – 2.13.

Таблиця 2.2

D1 Governance & Policies

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D1.1	Чи існує затверджена політика кібербезпеки або пакет політик, актуальна за останні 12 міс.?	PDF політики, дата перегляду	CIS 17/18; Org
D1.2	Чи визначені ролі та відповідальність, хто власник процесу впровадження кібербезпеки, хто адміністратор, відповідальний за інциденти?	матриця RACI, наказ чи опис посад	CIS 17; Org
D1.3	Чи є процес управління винятками, exception management, з термінами та власниками?	реєстр винятків	CIS 18; Org
D1.4	Чи визначено мінімальні вимоги до постачальників або аутсорсу?	шаблон договору, чекліст	CIS 15; Org
D1.5	Чи проводиться менеджмент рев'ю стану кібербезпеки за KPI або ризиками хоча б щоквартально?	протокол зустрічі, KPI	CIS 17/18; Org
D1.6	Чи існує затверджена політика резервування, плану дій у відповідь на інцидент, логування?	набори політик	CIS 11/17/8; Org

Таблиця 2.3

D2 Risk Management & Compliance

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D2.1	Чи проводиться регулярна оцінка ризиків хоча б 1 раз/рік або при змінах?	реєстр ризиків	CIS 18; Org

Закінчення таблиці 2.3

D2.2	Чи є план обробки ризиків з власниками й строками?	план обробки ризиків	CIS 18; Org
D2.3	Чи визначено вимоги відповідності та відповідальних?	матриця відповідності complience	CIS 18/15; Org
D2.4	Чи виконується внутрішній контроль раз на півроку?	чекліст, звіт	CIS 18; Org
D2.5	Чи оцінюються ризики третіх сторін: постачальники ПЗ, хостинг, аутсорс сервісів чи процесів?	реєстр постачальників та оцінка ризику	CIS 15; Org
D2.6	Чи є мінімальна модель ризик-метрик?	модель оцінка ризиків	CIS 18; Org

Таблиця 2.4

D3 Asset & Service Inventory

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D3.1	Чи проводиться інвентаризація та ототожнення пристроїв з власниками?	список активів	CIS 1; Org та Tech
D3.2	Чи ведеться облік ПЗ та версій, в тому числі критичних застосунків?	список ПЗ, що використовується	CIS 2
D3.3	Чи є перелік хмарних сервісів, SaaS та адміністраторів доступу?	SaaS list	CIS 2/5
D3.4	Чи описані критичні бізнес-процеси та їх IT-залежності?	список процесів	CIS 18; Org
D3.5	Чи визначено критичні активи для пріоритизації захисту?	перелік	CIS 18; Org
D3.6	Чи виявляється неавторизовані сервіси або ПЗ хоча б періодично?	звіт перевірки	CIS 2

Таблиця 2.5

D4 Identity & Access Management (IAM)

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D4.1	Чи унікальні облікові записи кожного працівника з правами адміністратора?	налаштування IAM	CIS 5/6
D4.2	Чи ввімкнено MFA для адмінів і віддаленого доступу?	скрін MFA-політики	CIS 6
D4.3	Чи застосовані мінімально необхідні привілеї ролям та групам?	RBAC-матриця	CIS 6
D4.4	Чи є процес створення, зміни та видалення доступів?	процедура JML	CIS 5
D4.5	Чи виконуються регулярні перегляди доступів квартально чи раз на півроку?	акт перегляду	CIS 6

Закінчення таблиці 2.5

D4.6	Чи привілеї локальних адмінів контрольвані та обмежені в часі?	Налаштування або політика	CIS 6

Таблиця 2.6

D5 Awareness & HR Security

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D5.1	Чи проходять співробітники базове навчання з КБ щороку?	журнал навчання	CIS 14; People
D5.2	Чи є правила дозволеного використання корпоративних акаунтів, девайсів, ПЗ тощо?	політика AUP або BYOD	CIS 14
D5.3	Чи проводяться фішинг-тести або принаймні інструктажі щодо фішингу?	звіт або план	CIS 14
D5.4	Чи є формалізований офбординг?	чекліст	CIS 5
D5.5	Чи визначені вимоги до паролів/менеджерів паролів для персоналу?	policy	CIS 6/14
D5.6	Чи визначено порядок повідомлення про підозрілі події?	інструкція	CIS 17

Таблиця 2.7

D6 Data Protection & Privacy Controls

ID	Питання	Приклад доказу	Мапування до CIS або ISO
D6.1	Чи класифіковано дані наприклад на публічні, внутрішні та конфіденційні?	політика класифікації даних	CIS 3
D6.2	Чи використовується шифрування TLS для критичних сервісів?	налаштування TLS	CIS 3
D6.3	Чи використовується шифрування на носії корпоративних даних?	BitLocker або налаштування бази даних	CIS 3
D6.4	Чи є політика зберігання та видалення даних?	retention policy	CIS 3
D6.5	Чи контролюється доступ до чутливих даних, наприклад обмеження групами?	ACL або RBAC	CIS 3/6
D6.6	Чи ведеться базова документація приватності персональних даних та хто має до цього доступ?	data map	CIS 3

Таблиця 2.8

D7 Secure Configuration & Patch and Vulnerability Management

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D7.1	Чи є базові secure baselines для ОС, мережевого обладнання або хмарних сервісів?	baseline doc	CIS 4
D7.2	Чи визначені формально визначені строки для патчів?	patch policy	CIS 7
D7.3	Чи застосовуються автоматичні оновлення або централізований менеджмент патчів?	MDM або WSUS	CIS 7
D7.4	Чи проводиться сканування вразливостей квартально?	vuln scan report	CIS 7
D7.5	Чи є контроль змін конфігурацій для критичних систем?	change log	CIS 4/18
D7.6	Чи обмежено використання застарілих протоколів або слабких налаштувань?	hardening evidence	CIS 4

Таблиця 2.9

D8 Network Security

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D8.1	Чи є керований налаштований фаєрвол?	firewall ruleset	CIS 12
D8.2	Чи є сегментація на офісний, серверний та гостьовий Wi-Fi?	network diagram	CIS 12
D8.3	Чи захищений віддалений доступ?	VPN config	CIS 6/12
D8.4	Чи налаштовано безпечний Wi-Fi?	Wi-Fi settings	CIS 12
D8.5	Чи є базовий захист DNS або веб-доступу?	DNS policy	CIS 9/12
D8.6	Чи контролюється вихідний трафік для критичних систем?	rules	CIS 13

Таблиця 2.10

D9 Endpoint & Application Security

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D9.1	Чи встановлено й оновлюється AV або EDR на робочих станціях?	endpoint dashboard	CIS 10
D9.2	Чи ввімкнено шифрування дисків на ноутбуках?	BitLocker status	CIS 3/10
D9.3	Чи є MDM та політики для мобільних телефонів або ноутбуків чи комп'ютерів?	MDM policy	CIS 10
D9.4	Чи застосовується контроль запуску ПЗ хоча б для адмінів?	policy	CIS 10
D9.5	Якщо є веб-додаток, то чи є базові практики secure SDLC?	repo rules	CIS 16

Закінчення таблиці 2.10

D9.6	Чи ізольовані адмін-операції у окремий профіль, пристрій, віртуальна операційна система?	procedure	CIS 6/10

Таблиця 2.11

D10 Logging, Monitoring & Detection

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D10.1	Чи ввімкнено логування ключових систем IAM, серверів, фаєрволів?	перелік джерел	CIS 8
D10.2	Чи є централізація логів для критичних джерел?	архітектура	CIS 8
D10.3	Чи визначено ретенцію логів і захист від підміни?	retention settings	CIS 8
D10.4	Чи є синхронізація часу на ключових системах?	NTP policy	CIS 8
D10.5	Чи налаштовано алерти для базових подій?	alert rules	CIS 13
D10.6	Чи є регулярний щотижневий огляд алертів та логів із відповідальним?	журнал огляду	CIS 13

Таблиця 2.12

D11 Incident Response (IR)

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D11.1	Чи існує план реагування з ролями та контактами?	IR plan	CIS 17
D11.2	Чи є runbooks для типових інцидентів, наприклад фішингу, вірусів з вимогами, витоків даних?	2–3 runbooks	CIS 17
D11.3	Чи визначено процес ескалації та комунікації внутрішньої та з зовнішніми стейкхолдерами?	escalation matrix	CIS 17
D11.4	Чи проводились tabletop/exercises за останні 12 міс.?	протокол вправи	CIS 17
D11.5	Чи визначено порядок збору та збереження доказів?	evidence handling	CIS 17
D11.6	Чи є ретроспектива після кіберінцидентів та здійснення коригувальних дій?	PIR report	CIS 17

Таблиця 2.13

D12 Backup, Recovery & Resilience

ID	Питання	Приклад доказу	Відповідність до CIS або ISO
D12.1	Чи є стратегія резервування?	backup policy	CIS 11
D12.2	Чи є резервні та незмінні копії для захисту від ransomware?	налаштування immutability	CIS 11

Закінчення таблиці 2.13

D12.3	Чи тестувалося відновлення за останні 6–12 міс.?	звіт тесту	CIS 11
D12.4	Чи визначені RPO або RTO для критичних процесів?	BIA або таблиця	CIS 11/18
D12.5	Чи є DR або BCP-процедури для критичних сервісів?	DR doc	CIS 11
D12.6	Чи ведеться журнал резервних копій, помилок і контроль виконання?	backup logs	CIS 11

Для анкетного формату найкраще працює лінгвістична шкала, де користувач оцінює ступінь впровадження практики. Але лінгвістичні значення не є точними числами, тому ми застосуємо нечіткі множини. Концепт нечітких множин, наприклад належності від 0 до 1 замість полярної відповіді так чи ні введений Заде і є базою для подальшої дефазифікації [32].

2.2. Розробка методу оцінки рівня кібербезпеки підприємства

AHP (Analytic Hierarchy Process) – метод багатокритеріального прийняття рішень, у якому складну задачу декомпозують в послідовну ієрархію: ціль, домени, критерії, після чого ваги визначаються через попарні порівняння. Базове обґрунтування AHP та його застосування як процедури пріоритизації викладено у класичній праці Сааті [31].

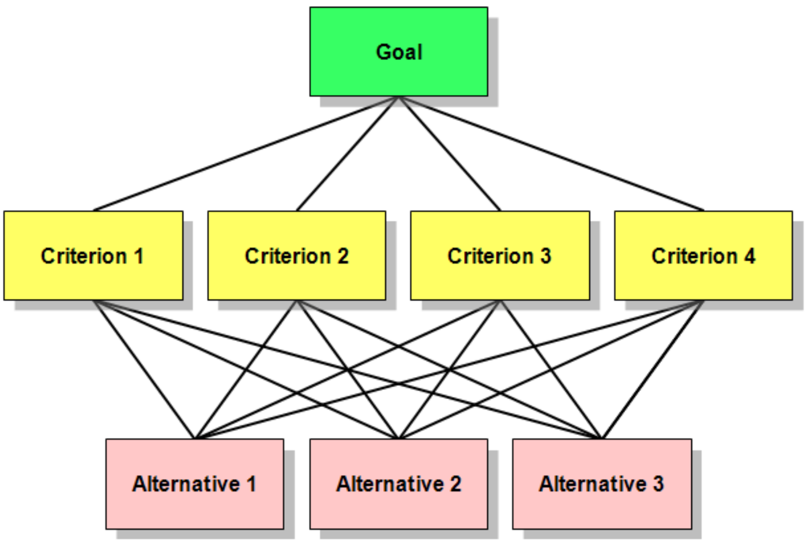

Рис. 2.1. Класичний приклад методу аналізу ієрархій

Метод пропонується спершу описати як формальну процедуру з чіткими змінними, що надалі допоможе з предметнішим описом розробки:

- $\mathrm{D} = d_1, \dots, d_m$ – множина доменів, в нашому методі 12 методів.
- Для кожного домену d задано набір критеріїв ($C_d = c_{d,1}, \dots, c_{d,n_d}$.
- w_d – вага домену, відображає важливість домену в інтегральному показнику, що в сумі зводиться до одиниці, тобто $\sum_d w_d = 1$.
- $v_{d,i}$ – вага критерію всередині домену, вона може бути рівною або теж визначатися методом аналізу ієрархій, $\sum_i v_{d,i} = 1$ для кожного d.
- Відповідь користувача на критерій $c_{d,i}$ подається лінгвістично та перетворюється на нормовану оцінку $x_{d,i} \in [0,1]$.
- $e_{d,i} \in (0,1]$ – коефіцієнт доказовості.

Тоді скоригована оцінка критерію отримуватиметься за формулою:

$$\widetilde{x_{d,\iota}} = e_{d,i} \cdot x_{d,i}. \tag{2.1}$$

AHP добре підходить для кібербезпеки, бо експертам легше порівнювати, ніж одразу задавати абсолютні числа. Експерт формує матрицю $\mathrm{A} = [a_{ij}]$, де a_{ij} це перевага домену d_i над d_j. Далі ваги w знаходяться як нормований головний власний вектор.

Проблема в методі аналізу ієрархій полягає в тому, що експерт може бути суперечливим, наприклад «A важливіше B», «B важливіше C», але «C важливіше A». Це контролюється через індекс узгодженості та коефіцієнт узгодженості CR. Вони є показниками, що вимірюють ступіть збігу думок або оцінок групи експертів чи даних. Якщо CR перевищує поріг, то матрицю попарних порівнянь треба переглянути або усереднити груповою процедурою.

Якщо ваги задають кілька експертів, використовується агрегація, що часто відбувається шляхом визначення геометричного середнього елементів матриці, щоб зменшити суб’єктивність одного респондента.

У нечіткій логіці існує термін лінгвістичної змінної, застосовний в нашій роботі. Ми можемо використати її для визначення рівню впровадження, що приймає значення від умовного низького до високого, які моделюються нечіткими множинами з функцією належності.

В розробці методу доцільно застосувати трикутні нечіткі числа $\mathrm{l}, \mathrm{m}, \mathrm{u}$, відповідно нижня межа, найімовірніше значення та верхня межа. Вони є доволі типовим інструментом у fuzzy-AHP. Наприклад, у роботі Чанга описано підхід до fuzzy AHP саме через використання трикутних нечітких чисел у попарних порівняннях [40].

У нашому методі доцільно зафіксувати 5 рівнів, це буде основою, яку в майбутньому можна буде калібрувати емпірично:

- L0, не впроваджено: TFN $(0.00,\ 0.00,\ 0.10)$
- L1, впроваджується епізодично: $(0.05,\ 0.20,\ 0.35)$
- L2, частково впроваджено: $(0.30,\ 0.50,\ 0.70)$
- L3, впроваджено й підтримується: $(0.65,\ 0.80,\ 0.95)$
- L4, оптимізовано та вимірюється: $(0.90,\ 1.00,\ 1.00)$

Гіпотезою та основною ідеєю саме таких значень є більша невизначеність в нижніх рівнях впровадження, та відповідно вузьчого інтервалу значень, де присутня більша визначеність і ймовірніше присутні артефакти в якості доказів.

Дефазифікація, термін що позначає перетворення нечіткого значення у чітке число. Для TFN часто беруть центроїд або середнє значення:

$$x = \frac{l + m + u}{3}. \tag{2.2}$$

У навчальній та інженерній літературі з нечітких систем центроїдна дефазифікація подається як базовий і широко застосовний варіант для отримання скалярного результату.

Перед моделлю нашого методу також стоїть задача бути реалістичною для малого та середнього бізнесу, де не всі критерії у доменах застосовні відповідно до контексту. Тому ми не перетворюємо опитування на аудит, але вводимо м'яку корекцію через коефіцієнт доказовості:

- $e_{d,i} = 1.0$, якщо користувач надав артефакт або система отримала сигнал автоматично. До прикладу, у організації що проходить опитування присутній журнал резервних копій, реалізована мультифакторна ауетентифікація у співробітників;
- $e_{d,i} = 0.85$, якщо є непрямі підтвердження або часткова доказовість;
- $e_{d,i} = 0.70$, якщо підтверджень немає.

Сенс такого підходу узгоджується з стандартом NIST SP 800-53A, де свідчиться, що оцінювання контролів має підтверджувати реалізацію і досягнення цілей, а не бути процедурою, що існує лише на папері [25].

Виведемо оцінку домену, з урахуванням ваг критеріїв:

$$S_d = \sum_{i=1}^{n_d} v_{d,i} \cdot \widetilde{x_{d,\iota}}, \quad \sum_{i} v_{d,i} = 1 \tag{2.3}$$

Виведемо загальну оцінку:

$$S = \sum_{d=1}^{m} w_d \cdot S_d, \quad \sum_{d} w_d = 1, \quad S \in [0,1]. \tag{2.4}$$

Нормалізація у оцінку від 1 до 100:

$$Score = max\{1, \text{round}(100 \cdot S)\}. \tag{2.5}$$

У кібербезпеці існують практики, відсутність яких створює настільки великий ризик, що середній бал може бути небезпечним самообманом. До прикладу, відсутність резервного копіювання з відтвореним тестом відновлення, повна відсутність MFA, відсутність плану реагування на інциденти.

Тому вводиться функція штрафу Р $\in (0,1]$, яка знижує результат при провалі критичних критеріїв:

$$S' = S \cdot P. \tag{2.6}$$

Наведемо сценарій застосування, якщо домен D12 Backup & Recovery нижчий за поріг $S_{D12} < 0.3$, то Р $= 0.7$; якщо одночасно D11 Incident Response < 0.3, то Р $=$ 0.55, тощо. Цей елемент важливий, бо сигналізує про існування критичних точок. Тобто, без відновлення й процедур реагування організація не може вважатися зрілою.

Контекстна адаптація є також критичною для нашого методу. Вона має бути формалізовано окресленою. Домени та критерії застосовні однакові, але ваги і пріоритетність рекомендацій змінюються відповідно.

Для цього вводимо множники $k_d(K)$ залежно від контексту К:

$$w'_d = \frac{w_d \cdot k_d(K)}{\sum_j w_j \cdot k_j(K)} \tag{2.7}$$

Як опорну ідею для малого та середнього розміру організацій можна використати CIS Implementation Groups: IG1 визначається як «essential cyber hygiene» і описується як базовий мінімум для всіх організацій; IG2 та IG3 додають вимоги залежно від ризику та ресурсів. Випливає лінійний механізм, де для малого бізнесу алгоритм рекомендацій за замовчуванням фокусується на IG1-діях, і з збільшенням розміру організацій рекомендації стають гострішими та вимогливіший [41].

Сформулюємо повну методику розрахунку у вигляді чіткої процедури:

Фіксується множина доменів D і критеріїв $D_x \in [1,12]$.

Обирається лінгвістична шкала L0 - L4 і її нечітке відображення TFN.

Задаються правила дефазифікації, в нашому методі центроїд та середнє є однаковим значення.

Далі визначаємо ваги доменів через метод аналізу ієрархій:

Експерти формують матрицю попарних порівнянь А $= [a_{ij}]$.

Обчислюється вектор ваг w, нормований $\sum w_i = 1$.

Перевіряється узгодженість:

$$\mathrm{CI} = \frac{\lambda_{max} - n}{n - 1}, \quad C\mathrm{R} = \frac{CI}{RI}. \tag{2.8}$$

Порогове правило $\mathrm{CR} \leq 0.1$ широко використовується як індикатор прийнятної узгодженості. RI, індекс випадковості, береться з таблиць Сааті; В нашому випадку достаньо буде навести RI для $\mathrm{n} = 3 \dots 10$.

Визначаємо ваги критеріїв у межах домену, тут є два коректні варіанти:

- Достатній, $v_{d,i} = 1/n_d$, тобто визначається рівновагомість, а деталізація ваг переноситься на дерево рішень через Δ.
- Покращений, з окремим AHP усередині кожного домену, особливо для D4, D10, D11, D12.

Наступним кроком збираємо відповіді та докази. Користувач дає відповідь L0 - L4 та за можливості додає артефакт для доказовості. Після цього ми застосовуємо нечітке відображення та дефазифікацію.

Відповідний L перетворюється у TFN (l, m, u).

Отримується чітке число для подальшого розрахунку $\mathrm{x} \in [0,1]$ через дефазифікацію.

Відбувається корекція, що спирається на наявність доказу

$$\widetilde{\mathrm{x}_{\mathrm{d,\iota}}} = \mathrm{e}_{\mathrm{d,i}} \cdot \mathrm{x}_{\mathrm{d,i}}, \quad \mathrm{e} \in \{1.0, 0.85, 0.7\}$$

Обчислюємо оцінки доменів:

$$S_d = \sum_{i=1}^{n_d} v_{d,i} \widetilde{x_{d,\iota}}, \quad S_d \in [0,1]. \tag{2.10}$$

Контекстна адаптація вагових коефіцієнтів відповідно до організації. Вводяться множники $k_d(K)$ і проводиться перенормування:

$$w'_d = \frac{w_d k_d(K)}{\sum_j w_j k_j(K)}. \tag{2.11}$$

Отримуємо інтегральну оцінку:

$$\mathrm{S} = \sum_{d=1}^{m} w'_d S_d\,, \quad S \in [0,1]. \tag{2.12}$$

Вводимо штрафні критерії для критичних провалів. Щоб уникнути невідповідно-високого результату при провалі базових практик, вводимо P $\in (0,1]$ і:

$$S' = \mathrm{S} \cdot P. \tag{2.13}$$

Далі переводимо у шкалу оцінки від 1 до 100:

$$\mathrm{Score} = \max 1,\mathrm{round}(100 \cdot S^*)\,. \tag{2.14}$$

Виявляємо слабкі місця:

$$\Delta_{d,i} = w'_d * v_{d,i}\left(x^{target} - \widetilde{x_{d,\imath}}\right). \tag{2.15}$$

Топ-N критеріїв за Δ і є головними точками для втручання.

Розберемо повний розрахунок інтегрального показника оцінки за нашим методом. Щоб продемонструвати весь ланцюжок обчислень компактно, розглянемо на трьох доменах:

- D4 Identity & Access Management;
- D10 Logging, Monitoring & Detection;

- D12 Backup, Recovery & Resilience.

Ваги доменів, припустимо, отримані методом аналізу ієрархій з $\Sigma = 1$:

$$w_{D4} = 0.30, \quad w_{D10} = 0.35, \quad w_{D12} = 0.35. \tag{2.16}$$

Критерії всередині доменів для демонстрації використовуватимемо в кількості трьох із рівними вагами:

$$v_{d,i} = 1/3. \tag{2.17}$$

Оскільки відповіді є лінгвістичними, ми трактуємо їх як нечіткі значення на основі апарату нечітких множин. Задамо TFN та застосуємо дефазифікацію центроїдом для трикутних нечітких чисел:

$$\mathrm{x} = \frac{l + m + u}{3}. \tag{2.18}$$

Таблиця 2.14

Дефазифікація рівня відповідності через трикутні нечіткі числа

Рівень	TFN $(\boldsymbol{l}, \boldsymbol{m}, \boldsymbol{u})$	Дефазифіковане $\boldsymbol{x}$
L0 не впроваджено	(0.00, 0.00, .10)	0.0333
L1 епізодично	(0.05, 0.20, 0.35)	0.2000
L2 частково	(0.30, 0.50, 0.70)	0.5000
L3 впроваджено	(0.65, 0.80, 0.95)	0.8000
L4 оптимізовано	(0.90, 1.00, 1.00)	0.9667

Коефіцієнт доказовості:

$$\mathrm{e} \in 1.00,\ 0.85,\ 0.70, \tag{2.19}$$

де 1.00 є фактом наявності артефакту чи іншого підтвердження, 0.85 це часткова відповідність, а 0.70 де підтвердження відсутнє.

Скоригована оцінка критерію:

$$\tilde{x}_{d,i} = e_{d,i} \cdot x_{d,i}. \tag{2.20}$$

Випадковим чином обираємо вихідні відповіді і перетворимо їх у числа.

D4 IAM, вага домену 0.30:

Таблиця 2.15

Розрахунок відповідей для домену D4

Критерій	**Відповідь**	$\boldsymbol{x}$	**Доказ** $\boldsymbol{e}$	$\widetilde{\boldsymbol{x}} = \boldsymbol{e} * \boldsymbol{x}$
D4.2 MFA для адмінів та віддаленого доступу	L2	0.5000	0.85	0.4250
D4.5 Регулярний перегляд доступів	L1	0.2000	0.70	0.1400
D4.4 Процес JML, joiner-mover-leaver	L3	0.8000	1.00	0.8000

Оцінка домену:

$$S_{D4} = \frac{0.4250 + 0.1400 + 0.8000}{3} = 0.4550. \tag{2.21}$$

D10 Logging & Monitoring, вага домену 0.35:

Таблиця 2.16

Розрахунок відповідей для домену D10

Критерій	**Відповідь**	$\boldsymbol{x}$	**Доказ** $\boldsymbol{e}$	$\widetilde{\boldsymbol{x}} = \boldsymbol{e} * \boldsymbol{x}$
D10.1 Логування ключових джерел	L3	0.8000	0.85	0.6800
D10.2 Централізація логів	L1	0.2000	1.00	0.2000
D10.5 Алерти для критичних подій	L0	0.0333	0.70	0.0233

$$S_{D10} = \frac{0.6800 + 0.2000 + 0.0233}{3} = 0.3011. \tag{2.22}$$

D12 Backup & Recovery, вага домену 0.35:

Таблиця 2.17

Розрахунок відповідей для домену D12

Критерій	**Відповідь**	$\boldsymbol{x}$	**Доказ** $\boldsymbol{e}$	$\boldsymbol{\tilde{x} = e * x}$
D12.1 Політика та стратегія резервування	L3	0.8000	1.00	0.8000
D12.2 Offline та immutable копії	L0	0.0333	0.70	0.0233
D12.3 Тест відновлення	L1	0.2000	0.70	0.1400

$$S_{D12} = \frac{0.8000 + 0.0233 + 0.1400}{3} = 0.3211. \quad (2.23)$$

Обрахуємо інтегральну оцінку без штрафу:

$$\mathrm{S} = \sum_{d} w_d S_d = 0.30 \cdot 0.4550 \; + \; 0.35 \cdot 0.3011 \; + \; 0.35 \cdot 0.3211. \quad (2.24)$$

$$\mathrm{S} = 0.1365 + 0.105385 + 0.112385 = 0.35427. \quad (2.25)$$

Попередній бал:

$$\mathrm{Score}_{raw} = \mathrm{round}(100 \cdot S) = \mathrm{round}(35.427) = 35. \quad (2.26)$$

Далі вирахуємо штраф для критичних провалів. У методі потрібен механізм, який не дозволяє отримати звичайний середній бал, коли провалені базові функції, без яких кіберстійкість низька. Задамо просте правило штрафу для прикладу обрахунку.

Якщо $\widetilde{x_{D12.3}} < 0.6$, тобто немає або не тестується відновлення, то $P_1 = 0.70$.

Якщо $\widetilde{x_{D10.5}} < 0.6$, тобто немає алертів, то $P_2 = 0.85$.

Загальний штраф має бути мультиплікативний, оскільки сигналізує про накопичення критичних слабкостей:

$$\mathrm{P} = P_1 \cdot P_2 = 0.70 \cdot 0.85 = 0.595. \quad (2.27)$$

Скоригована оцінка:

$$S^* = S \cdot P = 0.35427 \cdot 0.595 = 0.21079. \tag{2.28}$$

Підсумковий бал:

$$Score = max\{1, round(100 \cdot S^*)\} = round(21.079) = 21 \tag{2.29}$$

Інтерпретація цього прикладу така, що оцінка 35, отримана попередньо, не відображає реалістичний стан, оскільки існують критичні провали у двох критеріях, тому фінальна оцінка знижується до 21 зі 100.

Обчислимо тепер критичні місця, які є потенційними точками росту. Використовуємо маржинальний потенціал покращення:

$$\Delta_{d,i} = w_d \cdot v_{d,i} \cdot \left(x^{target} - \widetilde{x_{d,i}}\right), \quad x^{target} = 1. \tag{2.30}$$

Оскільки $v_{d,i} = 1/3$, рахуємо ключові Δ:

1. D12.2 Офлайн- та незмінні копії:

$$\Delta = 0.35 \cdot \frac{1}{3} \cdot (1 - 0.0233) \approx 0.11395 \tag{2.31}$$

2. D10.5 Алерти:

$$\Delta = 0.35 \cdot \frac{1}{3} \cdot (1 - 0.0233) \approx 0.11395 \tag{2.32}$$

3. D12.3 Тест відновлення:

$$\Delta = 0.35 \cdot \frac{1}{3} \cdot (1 - 0.1400) \approx 0.10033 \tag{2.33}$$

4. D10.2 Централізація логів:

$$\Delta = 0.35 \cdot \frac{1}{3} \cdot (1 - 0.2000) \approx 0.09333 \tag{2.34}$$

5. D4.5 Перегляд доступів:

$$\Delta = 0.30 \cdot \frac{1}{3} \cdot (1 - 0.1400) = 0.08600 \tag{2.35}$$

Висновок, що можемо побачити з цих результатів для дерева рішень, це те що найкращим першим кроком за ефективністю буде побудова мінімальної системи сповіщень та підсилення резервних копій, як офлайн так і незмінних, додатково звернути увагу на тестування відновлення. Це, до речі, прямо відповідає кращим практикам з стандартів CIS.

У нашому застосунку є потенціальним покращення ранжування дії за співвідношенням імпакту до зусилля, але математичне ядро імпакту є сама різниця внеску, тобто Δ.

В реальних підприємствах частина критеріїв може бути не застосовною, або Not Applicable, далі N/A. Якщо їх рахувати як умовний нуль, то ми отримаємо штучне заниження фінального результату, власне оцінка буде вважатися спотвореною. З типових прикладів: компанія може принципово не використовувати віддалений доступ; не мати веб-застосунку; мати заборону на BYOD тощо. Тому N/A трактуватиметься як виключення певного критерію з нормування оцінки.

Нехай $A_d \subseteq C_d$ множина застосовних критеріїв домену d. Тоді:

1. Для всіх $c_{d,i} \notin A_d$, тобто N/A, критерій не входитиме у суму;

2. Ваги критеріїв перенормовуються на A_d:

$$v'_{d,i} = \begin{cases} \frac{v_{d,i}}{\sum_{j \in A_d} v_{d,j}}, & i \in A_d, \\ 0, & i \notin A_d. \end{cases} \tag{2.36}$$

3. Оцінка домену обчислюється так:

$$S_d = \sum_{i \in A_d} v'_{d,i} \cdot \tilde{x}_{\{d,i]}. \tag{2.37}$$

Сенс полягає в тому, що домен оцінюється лише за тим, що реально є релевантним для організації, відповідно оцінка є наближеною до реальності, що є і основною метрикою успіхою нашого методу.

Іноді весь домен може бути N/A, це радше виключний випадок, але можливий крайній випадок, який варто врахувати. Тому реалізація такої в MVP для дуже специфічних доменів є релевантною. Тоді рекомендований підхід:

- встановити $w_d = 0$ для такого домену;
- перенормувати доменні ваги на множину застосовних доменів D':

$$w'_d = \begin{cases} \frac{w_d}{\sum_{j \in D'} w_j}, & d \in D', \\ 0, & d \notin D'. \end{cases} \tag{2.38}$$

Варто зазначити, що пропуск відповіді в жодному разі не N/A. В такому випадку є два коректні підходи:

Підхід 1: якщо критерій застосовний, але відповідь відсутня, це варто трактувати як L0 з низькою доказовістю, наприклад $e = 0.7$. Це стимулює заповнення та не завищує результат.

Підхід 2: якщо частка пропусків перевищує поріг, наприклад 10%, оцінка не видається, а повертається статус «отриманих даних недостатньо». Звичайно цей підхід є кращим з точки зору якості, але для нашого MVP він лише збільшить навантаження без дійсного результату.

У нашому методі буде використовуватися перший метод, хоча другий підхід є суттєвим покращенням для майбутніх ітерацій програмної реалізації.

2.3. Розробка рекомендацій щодо підвищення рівня кібербезпеки

У нашому методі рекомендації мають бути певним деревом рішень, яке вказує що, якщо критерій або домен слабкий, то система пропонує конкретний крок або перевіряє вузьке місце глибше. Далі вона формує необхідний артефакт, певну політику чи план, і після впровадження дає можливість перерахувати бал.

Слабким місцем або bottleneck у методі доцільно розуміти критерій або набір критеріїв, який має найбільший потенціал підвищення інтегрального балу за умови покращення.

Для критерію $c_{d,i}$ вводимо маржинальний потенціал покращення:

$$\Delta_{d,i} = w'_d \cdot v_{d,i} \cdot \left(x_{d,i}^{target} - \tilde{x}_{d,i}\right). \tag{2.39}$$

Формула прямо випливає з лінійної моделі агрегації у рівняннях 2.3 та 2.4 і дає прозору інтерпретацію чому на певний критерій звертається увага. Саме прозорість потім дозволяє побудувати пояснювані рекомендації.

Далі система ранжує критерії за $\Delta_{d,i}$ і вибирає топ N-ої кількості для генерації плану покращень.

Дерево рішень у методі доцільно трактувати як символьну модель прийняття рішень: граф, де вузли є перевіркою умов, ребра це варіанти стану, листя це рекомендовані дії або артефакти. Це відповідає класичному розумінню дерев рішень, описаному у науковій літературі, що досліджує машинне навчання [42].

На відміну від дерева на основі машинного навчання, яке вчиться на певному датасеті, у нашому MVP дерево спирається на правила, тобто задається експертно та прив’язується до фреймворків ISO, CIS, NIST. Перевага такого підходу відтворюваності відповідей, аудитопридатності та пояснюваності.

Розглянемо домен D10 Monitoring & Detection:

- Вузол 1: «чи є централізоване логування критичних систем?»

- якщо «ні» слідує рекомендація A: «увімкнути логування та визначити перелік джерел», надається шаблон політики логування та мінімальний список подій;
- якщо «так» то переходить у вузол 2: «чи є алерти або кореляції для критичних подій?»
    - якщо «ні» то слідує рекомендація B: «налаштувати базові правила виявлення»
    - якщо «так» то переходить у вузол 3: «чи є процес реагування на алерти?», перетікає до наступного домену D11 Incident Response

Щоб рекомендації не стали довільними порадами від ШІ, кожен лист дерева має містити три компоненти:

- конкретну дію з мінімальним набором кроків;
- артефакт, вартий уваги: політика, процедура, чекліст, план реагування, тощо;
- посилання на контрольні практики ISO/IEC 27002 або CIS Safeguards, тощо.

Наприклад, якщо провалено Incident Response, дерево повинно приводити до формалізації плану реагування.

Оскільки підприємства мають обмежені ресурси, рекомендації мають бути оптимізовані не лише на максимальний $\Delta_{d,i}$, а й на реалістичність впровадження. Для цього вводиться оцінка витраченого ресурсу effort та вартості для кожної дії $\mathrm{cost(a)}$. Тоді обрахунок відбувається використовуючи відношення:

$$\text{Priority}(a) = \frac{\Delta(a)}{cost(a)}. \tag{2.40}$$

Таким чином наше дерево рішень істостно нагадуватиме практичний беклог для бізнесу, бо зрозуміло що робити спочатку, щоб найшвидше підняти оцінку.

Роль ШІ у цій системі доцільно позиціонувати як інтерпритатор рекомендацій. Він пояснює чому впровадження певної політики має ваговий пріорітет над іншою. Додатково, він здатний згенерувати необхідний для організації артефакт, який прискорить впровадження. Проте структура рішення, до прикладу критичність певних критеріїв, лишається у частині дерева рішень. Таким чином залишається пояснюваність і можливість довести, що рекомендація не є випадковою або галюцинаціями штучного інтелекту.

Наведемо приклад як працюватиме метод. Припустимо, що домен D4 Access має вагу $w'_4 = 0.15$, і всередині нього є критерій «MFA для адмін-доступу» з $v_{4,1} = 0.25$. Користувач відповів «частково», чому відповідає L2, але не навів доказу, з чого слідує, що $e = 0.7$. З цього слідує:

- L2 відповідає TFN $(0.30{,}0.50{,}0.70)$
- дефазифікація: $x = (0.30 + 0.50 + 0.70)/3 = 0.50$
- перевірка на доказовість: $\tilde{x} = 0.7 * 0.50 = 0.35$

Якщо ціль $x^{target} = 1$, то потенціал:

$$\Delta = 0.15 \cdot 0.25 \cdot (1 - 0.35) = 0.024375. \tag{2.41}$$

Число зрозуміло інтерпритується: впровадження MFA на всіх платформах організації та підтвердження його існування дає відчутний приріст загального балу, тому система підніме цю дію в пріоритет.

У методі дерево рішень є інтерпретованою символьною структурою, що не обов'язково має бути ML-моделлю.

Практично дерево можна описати структурою вузла:

- перевірка виду $S_d <$ або $\widetilde{x_{d,\iota}} <$;
- рекомендовані дії та артефакти;
- перехід на уточнення, якщо умова істинна/хибна;
- прив'язка до стандарту CIS, ISO або CSF.

Наведемо декілька прикладів гілок, які рухаються за нашим методом в дереві рішень за доменом Network Security & Firewall D8:

Вихідне правило: якщо $S_{D8} < 0.6$ то маємо перейти до деталізації.

1. Вузол 1: $\widetilde{x_{D8.1}} < 0.6$, що стосується наявності керованого firewall та правил.

Action 1: увімкнути фаєрвол на периметрі та в хмарі; запровадити принцип мінімально необхідних портів; заборонити «ANY-ANY» для inbound.

Артефакт: «Firewall ruleset policy» та чекліст ревізії правил, який має відбуватися квартально.

Відношення: CIS 12, мережева інфраструктура та контролі периметру.

2. Вузол 2: якщо firewall є, але $\widetilde{x_{D8.2}} < 0.6$.

Action 2: розділити гостьову Wi-Fi мережу, офіс та сервери; мінімізувати «плоскість» мережі, тобто сегментувати користувачів та пристрої за підмережами.

Артефакт: мережевий діаграмний опис та правила між сегментами.

Відношення: CIS 12.

3. Fallback: якщо D8 задовільняється повністю, але S_{D10} низький це значить, що вузьке місце у виявленні, а не у фаєрволі, значить пошук критичної точки далі продовжиться у гілці D10.

У межах запропонованого методу рекомендація трактується як керована дія, яка пов'язана з конкретним критерієм опитувальника, має очікуваний вплив на часткову оцінку $\widetilde{x_{d,i}}$, а отже на інтегральний показник, і може бути підтверджена артефактом. Така логіка корелює з ідеями NIST CSF 2.0 про outcomes як основу керування ризиком та підходу ISO/IEC 27001 щодо побудови ISMS із вимогою встановлювати, підтримувати та постійно поліпшувати систему управління безпекою. Для наповнення рекомендацій практичним змістом використовується каталог Safeguards CIS Controls v8.1, який прямо позиціонується як пріоритизований набір захисних дій.

Відповідно, набір рекомендацій у застосунку доцільно розглядати як задачу локальної оптимізації: за обмежених ресурсів підібрати такі дії, щоб максимізувати очікуване зростання результуючої оцінки. На рівні методу це відображається через величину $\Delta_{d,i}$, яка є маржинальним потенціалом, а також через оцінку зусилля для кожної дії.

Щоб дерево рішень було відтворюваним, воно має спиратися на базу знань. Це певний контрольований набір пов'язаних дій, артефактів, випливаючих ефектів та їх відношення до стандартів. У найпростішому вигляді вводиться множина дій $A = \{a\_1, \ldots, a\_k\}$ для кожної з яких фіксується структура:

- зв’язок із критеріями, $link(a) \subseteq c_{d,i}$;
- цільовий ефект, тобто певне очікуване підвищення $\widetilde{x_{d,\iota}}$;
- артефакт, який підвищує доказовість $e_{d,i}$ і робить результат придатним до перевірки;
- відношення на відповідні Safeguards CIS Controls v8.1 та, за потреби, на типи контролів ISO/IEC 27002;
- оцінка зусилля $cost(a)$, наприклад шкала від 1 до 5, та за можливості, певні враховані передумови.

Така база знань має роль словника, який забезпечує процес створення рекомендацій, що завжди прив’язані до структури методу та зовнішніх еталонів та стандартів.

Після обчислення $\Delta_{d,i}$ для кожного критерію, який був зроблений в попередньому розділі, формується кандидатний набір дій:

$$A^* = a \in A\colon\ link(a) \cap TopN(\Delta) \neq \emptyset. \tag{2.42}$$

Щоб перетворити це на практичний беклог доцільно визначити очікуваний ефект дії як прогноз приросту інтегральної оцінки. Нехай дія підвищує набір критеріїв $c_{d,i}$ до нових значень $\widetilde{x'}_{d,i}$. Тоді лінійна модель дозволяє оцінити приріст:

$$\Delta(a) \approx \sum_{(d,i)\in \mathrm{link}(a)} w'_d v'_{d,i} \left(\tilde{x}'_{d,i} - \tilde{x}_{d,i}\right). \tag{2.43}$$

Далі вводиться критерій пріоритизації у відношенні ефекту та зусилля:

$$Priority(a) = \frac{\Delta(a)}{cost(a)}. \tag{2.44}$$

Таким чином, метод породжує доволі керований план з очевидним пріорітетом, відсортованим списком, з певним обґрунтуванням.

Дерево рішень у системі виконує роль ієрархічної інтерпретованої моделі правил, що послідовно звужує проблему від доменного рівня до конкретної дії. Інтерпретованість дерев рішень та їх природа, що походить від правил, є класично описаними в літературі з індукції [42] та у монографії CART, де дерево розглядається як дерево-структуровані правила класифікації та регресії [43]. У нашому випадку дерево не навчається на даних, навпаки сконструйоване експертно. Хоча оформленим воно залишається у формі, тотожній інтерпретованим правилам прийняття рішень.

Щоб дерево не перетворилося на хаотичний набір вузлів та гілок, доцільно закласти трирівневу логіку:

1. Шар критичних умов, який здійснює перевірки, які мають пріоритет над усім іншим, бо впливають на штрафи P і фактично обмежують «зрілість», наприклад відсутнім тестом відновлення, мінімальних алертів, базового IR-плану.

2. Шар доменного розгалуження, який обирає домен для втручання за найбільшими Δ та/або найнижчим S_d.

3. Шар уточнення, який є послідовністю точкових перевірок на рівні конкретних критеріїв, які приводять до конкретних дій та артефактів.

Така структура з декількох шарів гарантуватиме методологічність відпрацювання, оскільки вона поетапно опрацьовує усі дані з пропрацьованою логікою, що унеможливлює появу випадкових відповідей або галюцинацій, які можуть з'являтися при опрацюванні даних ШІ.

Кожний вузол n дерева рішень задається четвіркою:

$$n = \langle condition,\ explain,\ actions,\ next \rangle, \text{де} \tag{2.45}$$

- condition: булева умова, наприклад $\widetilde{x_{d,i}} < \tau$, $S_d < \tau_d$, або докази відсутні;
- explain: пояснення важливості зміни;
- actions: перелік дій із бази знань, опційно з пропонованими артефактами;
- next: перехід до наступного вузла або завершення гілки.

Пороги τ доцільно фіксувати як частину методу, наприклад 0.6 для мінімально прийнятного стану певного критерію, або робити адаптивними через профіль пріоритизації. В такому випадку це можуть бути різні рівні жорсткості для IG1/IG2/IG3 без прив'язки до конкретної індустрії.

Припустимо, після розрахунку отримано певну кількість критичних місць, пройдемося по умовно трьох найпріорітетніших: $\tilde{x}_{D_{10.5}}$, тобто немає алертів, $\widetilde{x_{D12.3}}$, тобто не тестується відновлення, $\widetilde{x_{D4.5}}$, не переглядаються доступи.

Тоді дерево починає з шару критичних умов:

Вузол 1, Recovery: якщо $\widetilde{x_{D12.3}} < 0.6$ це приводить до рекомендації провести тестування відновлення та додати артефакт звіту відповідного тестування. Відповідно таке рішення підвищує і $\tilde{x}$, і e, після чого система перераховує P і Score.

Після закриття вузлу 1 дерево переходить у Detect:

Вузол 2, Detection: якщо $\widetilde{x_{D10.5}} < 0.6$ слідує рекомандація налаштувати базові алерти та додати артефакт переліку правил та алертів, гарною доданою цінністю була би і процедура реагування на алерт, тобто пов'язування з іншими доменами.

Далі дерево переходить до Protect & Access:

Вузол 3, Access reviews: якщо $\widetilde{x_{D4.5}} < 0.6$ то слідує рекомендаціє запровадити регулярний перегляд доступів та додати артефакт акту перегляду доступів, також пропонується створити шаблон матриці ролей. Після виконання система підвищує рівень доказовості, тобто доказовості, та перераховує Score.

Функціонально застосунок реалізуватиме керований цикл:

1. первинне анкетування з отриманням оцінки $Score_0$;

2. генерація беклогу дій A^* та дерева рішень;

3. користувач обирає дію, система запитує артефакт або пропонує згенерувати шаблон, пояснює критерії перевірки виконання;

4. повторне анкетування або часткове оновлення відповідей та додавання доказів, система згенеровує наступну ітерацію оцінки $Score_1$;

5. повторне оновлення пріорітетних слабких місць; цикл триває.

Щоб забезпечити відтворюваність і аудитопридатність результатів оцінювання, програмна реалізація має зберігати не лише підсумковий Score, а й версію пройденого опитування, вагові коефіцієнти, TFN інтерпритацію, усі доказові артефакти, умови штрафів, а також маршрут проходження дерева рішень.

У межах MVP достатньо описати та реалізувати 9 базових сутностей:

1. Опис опитування;

2. Домен та критерій;

3. Набір вагових коефіцієнтів за доменами та внутрішньо-доменні за критеріями;

4. Набір відповідей користувача;

5. Артефакт-підтвердження;

6. Акт розрахунку, які проміжні значення, штрафи, фінальний Score;

7. Дія з ресурсозатратністю, ефектом, артефактами, відношенням до стандартів;

8. Дерево рішень та логіка переходів між вузлами;

9. Результат рекомендацій, слабкі місця, пріорітетні дії, шлях дерева.

2.4. Обґрунтування вибору технологій для програмної реалізації методу

Для забезпечення реалізації цього методу у програмній реалізації пропонується сформулювати технічні вимоги:

- версійність опитування та відтворюваність розрахунків, збереження TFN, ваг, умов штрафів;
- обчислювальний модуль з AHP-вагами, інтерпритація нечітких та чітких чисел в обидві сторони, N/A-перенормування, оцінка та пошук слабких місць;
- Рушій для дерева рішень;
- Сховище доказів у файлах та артефактах;
- Генерація артефактів, шаблонів політик, планів IR, шаблонів звітів;

Враховуючи технічні вимоги, опишемо верхньорівневу архітектуру програмного рішення разом з технологіями, що будуть застосовуватися в подальшій розробці:

Frontend на React та TypeScript буде практичним рішення для реалізації оаитування як динамічної форми з валідацією, збереженням прогресу та відображенням профілю доменів.

Backend на Python та FastAPI відповідатимуть за математичне ядро, яке реалізовуватиме AHP-ваги, нечітке відображення TFN та дефазифікація, N/A-перенормування, систему штрафів та врахування критичних умов, слабкі місця, формування рекомендацій і проходження дерева рішень. Вибір Python є вичерпним для закриття потреби швидкої реалізації математичних модулів враховуючи існування багатьох готових математичних бібліотек та забезпечить прозоре тестування формул.

База даних буде на PostgreSQL забезпечувати реляційні зв’язки, підтримується JSONB для версійних схем опитування, забезпечується збереження усіх станів проходження опитування та слідуванню рекомендацій послідовно.

Контейнеризація на Docker, що забезпечить відтворюваність середовища на більшості систем, є простою в розгортанні та демонстрації MVP.

ВИСНОВКИ ДО РОЗДІЛУ 2

У розділі 2 розроблено формалізований метод оцінки рівня кібербезпеки підприємства, який переходить від описової оцінки до кількісної, відтворюваної процедури з інтегральним результатом Score ∈ [1;100]. Змістова валідність структури оцінювання забезпечена доменною організацією, узгодженою з функціональною моделлю NIST CSF 2.0. У такий спосіб метод вимірює ступінь досягнення результатів, релевантних управлінню ризиком і стійкості підприємства.

Ключовою науково-прикладною складовою методу є гібридизація багатокритеріального підходу з обробкою невизначеності, ваги доменів та критеріїв можуть визначатися формально, лінгвістичні відповіді опитування інтерпретуються через нечіткі множини та перетворюються у нормовані значення за процедурою дефазифікації. Це дозволяє математично коректно працювати з типовими для самооцінки формулюваннями «частково/епізодично/підтримується», не зводячи їх до полярних відповідей «так» або «ні».

Для підвищення достовірності самооцінки у метод введено поняття доказовості як м’який коригувальний механізм: одна і та сама лінгвістична відповідь має різну вагу довіри залежно від наявності артефактів підтвердження, що може виступати в різних випадках політикою, звітом тесту відновлення, скріном налаштувань, логаи тощо. Концептуально це узгоджується з підходом NIST до оцінювання контролів як перевірки їх фактичної реалізації та досягнення цілей, а не як формальної декларації. Одночасно запропоновано механізм перенормування ваг критеріїв та доменів з огляду на релевантність домену чи критерію до контексту підприємтва, що запобігає штучному заниженню Score при наявності нерелевантних питань і робить метод придатним для різних організаційних моделей.

Щоб інтегральний бал не ставав штучно усередненим, коли провалені базові спроможності виявлення, реагування або відновлення, у метод включено штрафи та критичні умови для виключно необхідних критеріїв. Такий підхід управлінськи обґрунтований тим, що без мінімальних реалізацій певних доменів навіть висока зрілість інших доменів не гарантує стійкості. Score таким чином у методі відображає ступінь впровадження контролів та структурну наявність критичних здібностей, які визначають практичну кіберстійкість.

У підрозділі 2.3 запропоновано перетворення результатів оцінювання на кероване підвищення рівня кібербезпеки, обчислюються вузькі місця через маржинальний потенціал Δ і формуються рекомендації через дерева рішень, що інтерпретовані як базові правила «якщо-то». Інтерпретованість дерева рішень важлива методологічно: вона забезпечує пояснюваність і відтворюваність траєкторії оцінка-дія-переоцінка у межах циклу постійного покращення, притаманного управлінню безпекою. Також у розділі зафіксовано формат даних і продемонстровано сценарій проходження дерева з послідовним усуненням критичних провалів та повторним перерахунком Score.

З практичної точки зору, розділ 2 доводить готовність методу до програмної реалізації, є визначена структура доменів та критеріїв, шкали, формули агрегації, правила N/A та штрафів, механізм пріорітетності слабких місць і формалізований формат бази знань рекомендацій. Візія реалізації у межах цього розділу обмежена MVP-стеком: React з TypeScript для інтерфейсу опитування, FastAPI для реалізації рушія обрахунків та виконання правил, PostgreSQL як надійне сховище версійності та слідів обчислень, файлове зберігання артефактів-доказів на сервері, контейнеризація Docker для відтворюваного розгортання.

Підсумовуючи, розділ 2 формує повний теоретично-методичний фундамент для подальшої програмної реалізації з чітко описаними поетапними процедурами, зрозуміла послідовність дій, логіка дерева рішень і повторних ітерацій оцінювання.

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ОЦІНКИ РІВНЯ КІБЕРБЕЗПЕКИ ПІДПРИЄМСТВА

В цьому розділі описано програмну реалізацію запропонованого у розділі 2 методу оцінювання від побудови архітектури і моделі даних до ключових механізмів обчислення Score та формування рекомендацій. Реалізацію орієнтовано на MVP, тобто на мінімально достатню функціональність, яка дозволяє пройти опитування у веб-інтерфейсі, зберегти відповіді, обчислити інтегральний показник та сформувати список пріоритетних покращень.

Система спроєктована трирівнево, frontend з React та TypeScript, backend з FastAPI та SQLAlchemy та PostgreSQL як єдине джерело даних. Компоненти готуються до контейнерного розгортання з використанням Docker. У frontend використано Nginx як сервер статичних файлів та reverse proxy для API-запитів. Важливо зафіксувати, що в даному розділі ми реалізовуємо MVP, яка може вар'юватися від теоретичної моделі, розробленої в попередньому розділі. Метод допускає використання нечіткої логіки, дефазифікації, штрафів та критичних умов для певних критеріїв. У наявній реалізації рівні L0 - L4 відображаються у дискретну шкалу 0–100 із кроком 25, застосовується коефіцієнт доказовості і перенормування ваг при N/A. Така реалізація покриває нашу ціль в створенні MVP, але залишає простір для розширення до повної математичної моделі.

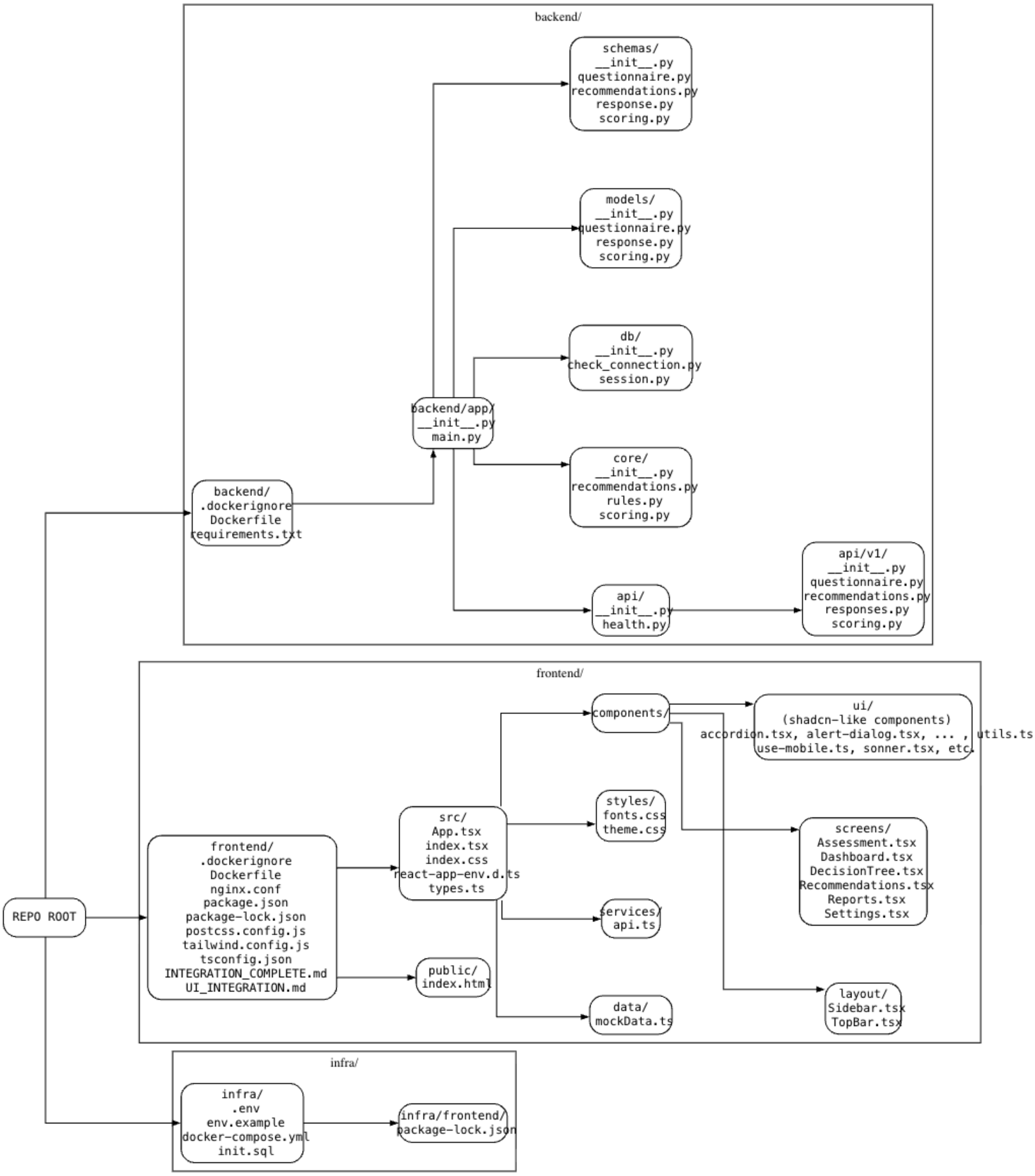

Рис. 3.1. Структурна схема програмної реалізації

3.1. Архітектура програмної реалізації MVP

Реалізація побудована як контейнеризована клієнт-серверна система з трьома логічними складовими: веб-клієнтом, сервером застосунку та базою даних

PostgreSQL. Під «контейнеризацією» розуміється пакування застосунків разом із залежностями в ізольовані середовища виконання, що забезпечує відтворюваність запуску на різних машинах. Для користувача система виглядає як єдиний веб-додаток, але внутрішньо розподілена за відповідальністю: інтерфейс збирає дані, сервер виконує валідацію й обчислення, база даних забезпечує довготривале збереження версій опитування, відповідей та результатів.

У файлі docker-compose визначено три сервіси: postgres, backend, frontend, їхні порти, змінні оточення, залежності запуску та спільну мережу cyberscore_network. PostgreSQL ініціалізується SQL-скриптом, який монтується в стандартну директорію docker-entrypoint-initdb.d, що гарантує створення таблиць і початкових даних при першому старті контейнера. Це прямо реалізовано через volume-мапінг ./init.sql:/docker-entrypoint-initdb.d/init.sql:ro. Volume-мапінг це механізм, що створює прямий зв'язок між локальною файловою системою користувача та ізольованим середовищем контейнера, завдяки чому база даних розгортається з готовою структурою без необхідності налаштовувати її вручну. Параметр ro налаштовує read-only режим, аби контейнер міг лише зчитати інструкції з файлу, але не змінювати його.

У Frontend використано зв'язку React та TypeScript. React задає компонентну модель побудови інтерфейсу, а TypeScript додає статичну типізацію, що зменшує ризик помилок під час інтеграції з API та моделювання даних доменів та критеріїв. У звичайному режимі роботи веб-клієнт віддається через Nginx. У цьому проєкті Nginx виконує роль веб-сервера статичних файлів і reverse proxy для API. Reverse proxy означає, що браузер звертається до одного домену або хоста, а Nginx перенаправляє частину запитів, наприклад /api/*, на Backend. Це спрощує мережеву модель і практично усуває типові проблеми міждоменної політики браузера.

```
server {
    listen 80;
    server_name localhost;
    root /usr/share/nginx/html;
    index index.html;

    gzip on;
    gzip_vary on;
    gzip_min_length 1024;
    gzip_types text/plain text/css text/xml text/javascript application/x-
javascript application/xml+rss application/json;

    location /api/ {
        proxy_pass http://backend:8000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }

    location / {
        try_files $uri $uri/ /index.html;
        add_header Cache-Control "no-cache, no-store, must-revalidate";
        add_header Pragma "no-cache";
        add_header Expires "0";
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2|ttf|eot)$ {
        expires 1y;
        add_header Cache-Control "public, immutable";
    }

    location /health {
        access_log off;
        return 200 "healthy\n";
        add_header Content-Type text/plain;
    }
}
```

Рис. 3.2. Конфігурація файлу nginx.conf

Backend реалізований на FastAPI із використанням SQLAlchemy як ORM. ORM (Object-Relational Mapping) – це шар, який відображає таблиці БД у класи, а рядки в об’єкти, що дозволяє працювати з даними на рівні мови програмування, зберігаючи транзакційність і цілісність. FastAPI використано як шар API, який приймає HTTP-запити, валідовує дані та викликає логіку обрахунку оцінки та створення рекомендацій. PostgreSQL виступає єдиним сховищем даних. Backend фактично реалізовано модульно, є окремий виділений API-рівень app/api/, ядро з головними калькуляціями app/core/, ORM-моделі app/models/, схеми серіалізації та валідації app/schemas/ та конфігурацію БД app/db/. Цей підхід знижує зв'язність компонентів, що зменшує ризик випадкової помилки на етапі розробки.

3.2. Проєктування даних і модель бази даних на PostgreSQL

Модель даних підпорядкована головному сценарію: система має зберігати версійну структуру опитування, набори відповідей користувача, результати обчислення Score і результати генерації рекомендацій. Версійність є принциповою: вона дозволяє порівнювати оцінювання в часі навіть тоді, коли структура критеріїв змінюється.

В Backend ця модель реалізована через SQLAlchemy-класи:

- QuestionnaireDefinition представляє версію, має атрибути опису та час створення, а також зв'язок із доменами;
- Domain представляє домен D1..D12, кожен домен має власний код, назвою, вагу та опис. Домен прив'язаний до версії опитувальника і містить список критеріїв;
- Criterion представляє критерій у межах домену, тобто ідентифікатор, назва, вага в домені, опис і прив'язка до версії;
- ResponseSet є контейнером відповідей користувача для конкретної версії опитувальника;
- Response є відповіддю на конкретний критерій: рівень L0 - L4, статус доказовості «FULL/PARTIAL/NONE» і ознака застосовності, яка реалізує N/A.
- ScoringRun фіксує результат обчислення: інтегральний Score та доменні оцінки у JSON-полі, що дозволяє зберігати словник {"D1": ..., "D2": ...} без створення додаткових таблиць.

- RecommendationRun зберігає сформований набір рекомендацій як JSON-масив структурованих об’єктів.

Зберігання доменних оцінок і рекомендацій у JSON не руйнує нормальну форму даних у критичних місцях, бо ключові сутності залишаються реляційними, а похідні результати є фактично знімками розрахунку. Таким чином забезпечується як відтворюваність Score у часі, так і достатня гнучкість для еволюції формату рекомендацій без міграції схеми на кожному кроці.

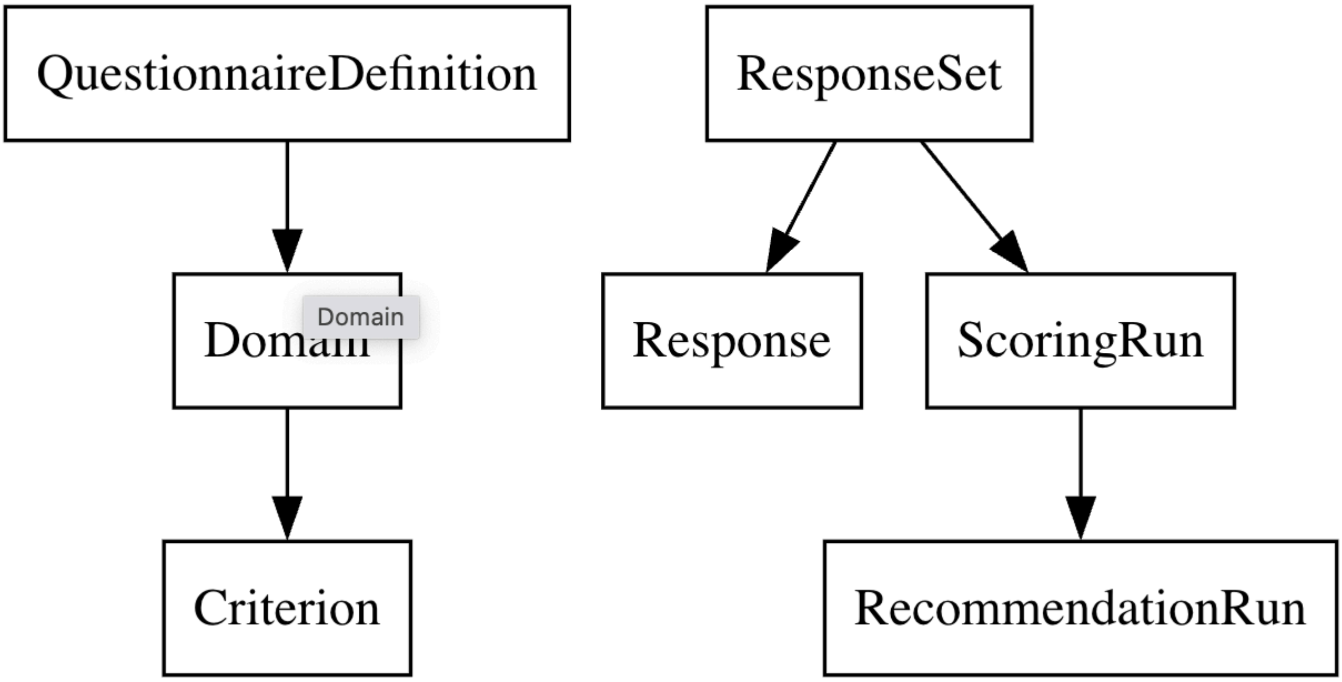

Рис. 3.3. ER-діаграма бази даних

3.3. Реалізація backend на FastAPI

Backend реалізує сервер застосунку як REST API на базі FastAPI та забезпечує повний цикл обробки результатів опитування. Від отримання версійної структури критеріїв до збереження відповідей, розрахунку інтегрального показника Score і генерації пріоритетних рекомендацій. Архітектурно серверна частина побудована за принципом розділення відповідальності: шар API відповідає за прийом HTTP-запитів і валідацію, шар доступу до даних за транзакції та цілісність у PostgreSQL, а ядро калькуляції за нормалізацію ваг, математичне агрегування та формування рекомендацій.

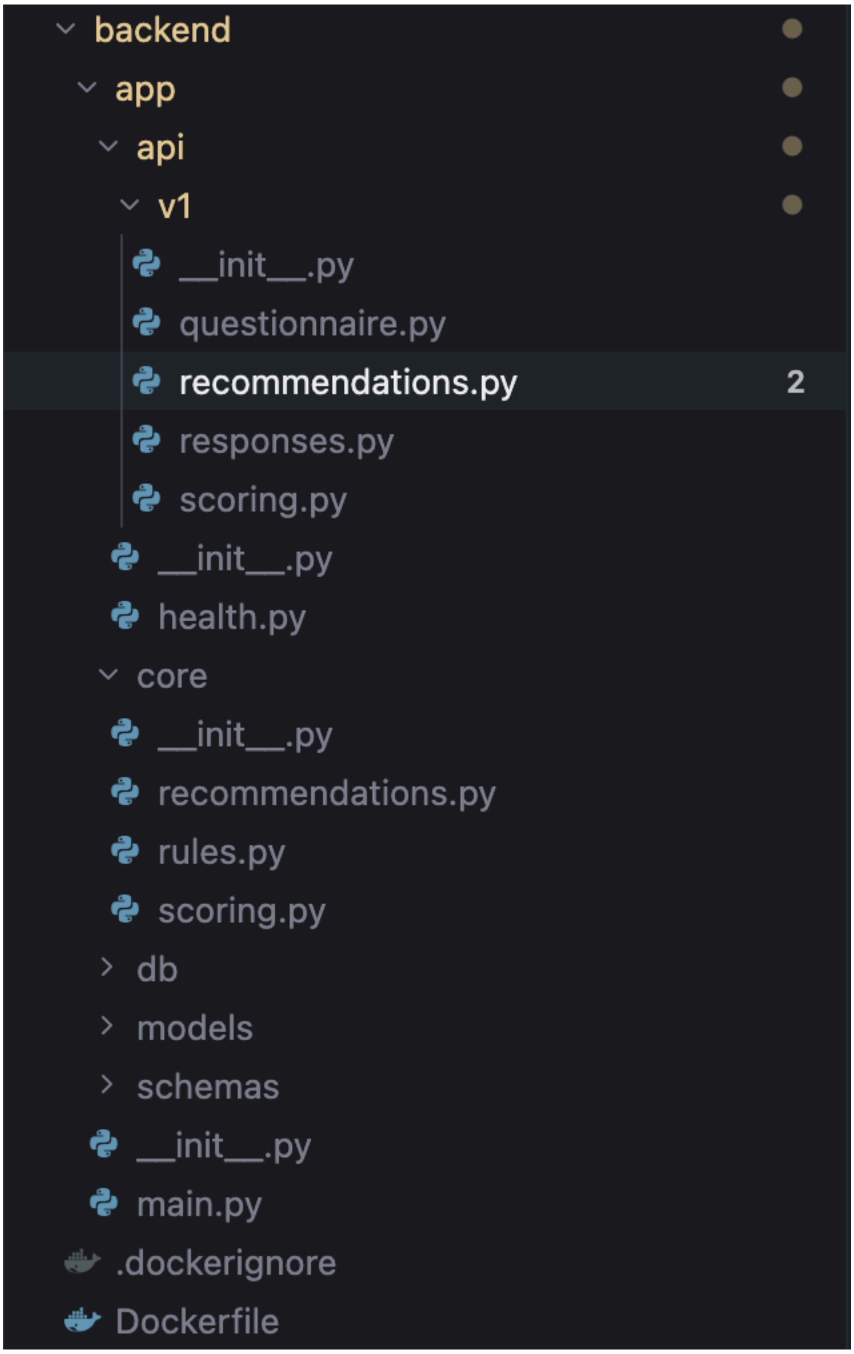

Рис. 3.4. Файлова структура бекенду

Головним файлом є main.py, де ініціалізується FastAPI-застосунок, підключається CORS-middleware та реєструються маршрути. Додатково реалізовано startup-перевірку підключення до БД з повторними спробами. Така перевірка важлива для контейнеризованого запуску, коли база даних може стартувати повільніше за застосунок.

```python
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from sqlalchemy import text
from app.api import health, v1
from app.db.session import engine, Base
import os
import time

app = FastAPI(
    title="CyberScore API",
    description="API для системи оцінки кібербезпеки підприємства",
    version="1.0.0"
)

cors_origins = os.getenv("CORS_ORIGINS",
"http://localhost:3000").split(",")
app.add_middleware(
    CORSMiddleware,
    allow_origins=cors_origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

app.include_router(health.router, tags=["Health"])
app.include_router(v1.router, prefix="/api/v1", tags=["API v1"])

@app.on_event("startup")
async def startup_event():
    """Подія запуску додатку."""
    print("CyberScore API запущено")
    max_retries = 10
    for i in range(max_retries):
        try:
            with engine.connect() as conn:
                conn.execute(text("SELECT 1"))
            print("Підключення до БД успішне!")
            break
        except Exception as e:
            if i < max_retries - 1:
                print(f"⏳ Очікування підключення до БД... (спроба
{i+1}/{max_retries})")
                time.sleep(2)
            else:
                print(f"Попередження: не вдалося підключитися до БД: {e}")
                print("Таблиці мають бути створені через init.sql")

@app.on_event("shutdown")
async def shutdown_event():
    print("CyberScore API зупинено")
```

Рис. 3.5. Код файлу main.py

В застосунку API реалізує три основні сценарії:

1. отримання структури опитування за версією;
2. створення набору відповідей;
3. запуск обчислення Score та отримання рекомендацій.

Перший сценарій реалізовано endpoint-ом GET /api/v1/questionnaire/{version}. Він читає з БД сутність QuestionnaireDefinition і підвантажує домени з критеріями. На стороні Frontend це дозволяє динамічно будувати форму опитування без зашитого переліку питань у коді інтерфейсу.

```python
from fastapi import APIRouter, Depends, HTTPException
from sqlalchemy.orm import Session, joinedload
from app.db.session import get_db
from app.models.questionnaire import QuestionnaireDefinition, Domain,
Criterion
from app.schemas.questionnaire import QuestionnaireResponse

router = APIRouter()


@router.get("/{version}", response_model=QuestionnaireResponse)
async def get_questionnaire(version: str, db: Session = Depends(get_db)):
    questionnaire = db.query(QuestionnaireDefinition).filter(
        QuestionnaireDefinition.version == version
    ).first()

    if not questionnaire:
        raise HTTPException(status_code=404, detail=f"Questionnaire
version {version} not found")

    domains = db.query(Domain).options(
        joinedload(Domain.criteria)
    ).filter(
        Domain.questionnaire_version == version
    ).all()

    questionnaire.domains = domains

    return questionnaire
```

Рис. 3.6. Код файлу questionnaire.py

Другий сценарій реалізовано endpoint-ом POST /api/v1/responses. Запит містить questionnaire_version та список відповідей по критеріях. Сервер створює ResponseSet, а потім вставляє пов’язані Response. Такий підхід має певні переваги: по-перше, у БД зберігається фактичний набір відповідей як артефакт аудиту; по-друге, оцінювання стає повторюваною операцією, його можна виконувати повторно, навіть якщо формула змінюється.

```python
@router.post("", response_model=ResponseSetResponse)
async def create_response_set(request: ResponseSetCreate, db: Session =
Depends(get_db)):

    response_set = ResponseSet(
        questionnaire_version=request.questionnaire_version
    )
    db.add(response_set)
    db.flush()

    for response_data in request.responses:
        response = Response(
            response_set_id=response_set.id,
            criterion_id=response_data.criterion_id,
            level=response_data.level,
            evidence_status=response_data.evidence_status,
            applicable=response_data.applicable
        )
        db.add(response)

    db.commit()
    db.refresh(response_set)

    return response_set
```

Рис. 3.7. Фрагмент коду з функцією create_response_set

Третій сценарій реалізовано endpoint-ом POST /api/v1/scoring/run, який приймає ідентифікатор response_set_id. Сервер читає всі відповіді, читає домени та критерії відповідної версії, виконує перенормування ваг з урахуванням N/A і обчислює доменні та інтегральну оцінки. Результат записується як ScoringRun, після чого може бути використаний для генерації рекомендацій.

```python
@router.post("/run", response_model=ScoringResponse)
async def run_scoring(request: ScoringRequest, db: Session =
Depends(get_db)):

    response_set = db.query(ResponseSet).filter(ResponseSet.id ==
request.response_set_id).first()
    if not response_set:
        raise HTTPException(status_code=404, detail=f"ResponseSet
{request.response_set_id} not found")

    responses = db.query(Response).filter(Response.response_set_id ==
request.response_set_id).all()
    if not responses:
        raise HTTPException(status_code=400, detail="No responses found
for this ResponseSet")

    domains = db.query(Domain).filter(
        Domain.questionnaire_version == response_set.questionnaire_version
    ).all()

    if not domains:
        raise HTTPException(
            status_code=404,
            detail=f"No domains found for questionnaire version
{response_set.questionnaire_version}"
        )

    normalized_criterion_weights = normalize_weights(domains, responses)
    normalized_domain_weights = get_domain_weights(domains, responses)

    scores = calculate_total_score(
        domains,
        responses,
        normalized_domain_weights,
        normalized_criterion_weights
    )

    total_score = scores.pop("total")
    domain_scores = scores

    scoring_run = ScoringRun(
        response_set_id=request.response_set_id,
        total_score=total_score,
        domain_scores=domain_scores
    )
    db.add(scoring_run)
    db.commit()
    db.refresh(scoring_run)

    return ScoringResponse(
        scoring_run_id=scoring_run.id,
        total_score=float(total_score),
        domain_scores={k: float(v) for k, v in domain_scores.items()},
        created_at=scoring_run.created_at
    )
```

Рис. 3.8. Фрагмент коду з функією run_scoring

Підтримка N/A реалізована через поле applicable в кожній відповіді. Це дозволяє позначити критерій як нерелевантний для конкретної організації. Однак просте виключення критерію створює математичну проблему, бо сума ваг у домені та між доменами перестає дорівнювати 1, а отже порівнянність Score порушується. Для розв’язання цього використано перенормування.

Функція normalize_weights обчислює нові ваги критеріїв у межах кожного домену, ваги застосовних критеріїв масштабуються так, щоб їх сума дорівнювала 1, а N/A-критерії отримували вагу 0. Аналогічно get_domain_weights обчислює ваги доменів, виключаючи домени, де немає жодного застосовного критерію. Це зберігає коректність агрегування: інтегральний Score завжди є зваженою сумою доменних оцінок за коректно нормованими коефіцієнтами.

У реалізації модуля обрахунку core/scoring.py рівні зрілості L0 - L4 відображено в дискретну числову шкалу: 0, 25, 50, 75, 100. Далі застосовується коефіцієнт доказовості FULL=1.0, PARTIAL=0.85, NONE=0.70, який зменшує внесок критерію при відсутності або неповноті підтверджувальних артефактів. Після цього значення множиться на нормовану вагу критерію, сумується в оцінку домену, а оцінки доменів агрегуються в загальний Score через нормовані ваги доменів.

Цей механізм відтворює основну інтуїцію методу розроблену у другому розділі: користувацька відповідь має бути зважена, доказовість впливає на довіру до самооцінки, N/A не впливає на остаточну оцінку. У MVP реалізовано детермінований варіант оцінки, який забезпечує працездатний цикл і створює фундамент для подальшого переходу до використання TFN та фази-дефазифікації, штрафів та критичних умов без зміни структури даних.

Модуль рекомендацій core/recommendations.py реалізує практичне продовження оцінювання: він перетворює результат у список дій з оцінкою потенційного впливу та складності. Це працює наступним чином: для кожного застосовного критерію визначається цільовий рівень, зазвичай на 1–2 рівні вище поточного, обмежуючись максимум в L4. Далі impact оцінюється як різниця між поточним і цільовим значеннями, помножена на фактор доказовості і на ваги домену та критерію. У результаті отримується оцінка, наскільки підвищення конкретного критерію може збільшити загальний Score. Необхідний ресурс оцінюється евристично за розривом рівнів, початковою точкою і станом доказів.

Такий підхід у MVP дозволяє рекомендаціям стати керованими та порівнюваними: система може ранжувати кроки і показувати, що саме дає найбільший приріст Score. Надалі, при переході до повної математичної моделі, потенційний приріст можна зробити точнішим, наприклад, через перерахунок Score при «симуляції» покращення критерію, не змінюючи API-контракту рекомендацій.

Генерація рекомендацій доступна як окремий endpoint, який прив’язаний до конкретного ScoringRun та зберігає результат у RecommendationRun. Таким чином рекомендації також стають відтворюваними артефактами: можна аналізувати, як змінювались поради між проходженнями або при зміні версії методу.

```
def calculate_criterion_impact(
    criterion: Criterion,
    current_level: int,
    target_level: int,
    evidence_status: str,
    domain_weight: float,
    criterion_weight: float
) -> float:
    if current_level >= target_level:
        return 0.0

    current_value = LEVEL_TO_VALUE.get(current_level, 0.0)
    target_value = LEVEL_TO_VALUE.get(target_level, 0.0)
    evidence_factor = EVIDENCE_FACTORS.get(evidence_status, 0.70)

    score_diff = (target_value - current_value) * evidence_factor

    impact = score_diff * criterion_weight * domain_weight

    return round(impact, 2)
```

Рис. 3.9. Функція calculate_criterion_impact

3.4. Реалізація Frontend та інтеграція з API

Клієнтська частина застосунку реалізована як Single Page Application – односторінковий вебзастосунок, у якому навігація між екранами здійснюється без повного перезавантаження сторінки. SPA-підхід зменшує затримки під час переходів, дозволяє повторно використовувати спільні UI-компоненти, наприклад навігацію, панель пошуку, профіль користувача. Також спрощує реалізацію живих інтерактивних сценаріїв, наприклад опитування зберігає стан заповнених полей, відбувається миттєвий перерахунок прогнозу Score, перегляд документів, візуалізація дерева рішень тощо.

Інтерфейс побудований за принципом інформаційної архітектури керування оцінюванням: користувач рухається від огляду поточного стану на сторінці Dashboard до заповнення оцінювання на сторінці Assessment, перевіряє отримання і пріоритизацію покращень на Recommendations, може подивитися на навігацію причинно-наслідкової логіки на сторінці Decision Tree. На сторінці Reports можна завантажити або згенерувати артефакти.

У боковій панелі відображено головні модулі системи: Dashboard, Assessment, Recommendations, Decision Tree, Reports, Settings. Верхня панель містить глобальний пошук, який орієнтований на швидку навігацію по сутностях доменної моделі: домени, критерії, рекомендації та вузли дерева рішень. Пошук трактовано як клієнтську функцію фільтрації в межах завантажених даних з можливістю масштабування до серверного пошуку, коли даних стане більше. Праворуч у верхній панелі розміщені сповіщення та профіль користувача.

Екран Dashboard виконує роль інформаційної панелі системи. Центральним елементом є загальна оцінка кіберстійкості, яку користувач отримує після проходження оцінювання. Поруч відображається зміна Score відносно попереднього запуску та час останнього розрахунку. Візуальна динаміка у вигляді короткого графіка допомагає правильно інтерпретувати дані, бо користувач бачить декілька джерел даних та траєкторію змін.

Нижче розміщено блок Domain Scores з набором карток доменів D1…D12 з числовими значеннями. Модуль доменів є критичним з погляду методики: він показує «профіль» кібербезпеки, тобто нерівномірність зрілості між напрямами. Картковий формат скорочує когнітивне навантаження: замість перегляду десятків критеріїв користувач одразу ідентифікує домени з низькими значеннями, які потребують пріоритетного втручання. Додаткові маркери стану на картках, такі як індикатори та позначки, виконують роль швидкої класифікації ризику домену.

Внизу екрана відображено Recent Assessment Runs, це список останніх запусків оцінювання з ID, датою, Score та версією. Система зберігає історію розрахунків і дозволяє порівнювати результати між різними проходженнями.

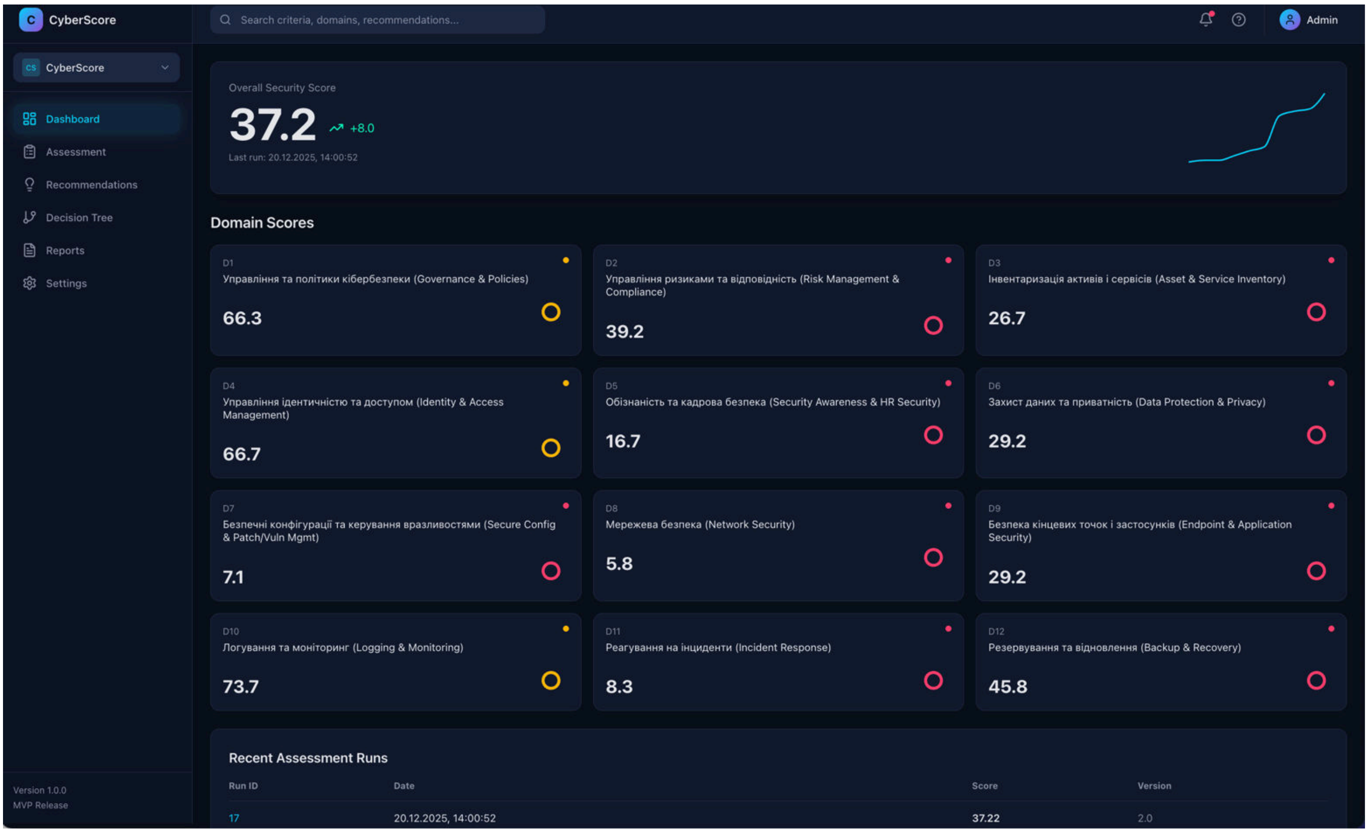

Рис. 3.10. Сторінка Dashboard

З точки зору взаємодії з Backend, Dashboard лише виконує функцію представлення, він оперує останнім ScoringRun для організації або користувача та відображає агреговані значення без додаткових обчислень на стороні браузера.

Модуль Assessment є точкою формування вхідних даних для математичного ядра. Інтерфейс організований навколо доменів і критеріїв: користувач послідовно задає рівні відповідності L0 - L4, зазначає застосовність N/A і, за можливості, статус доказовості, в межах «відсутності», «часткової присутності» або «повної».

Після заповнення оцінювання користувач запускає обчислення. У цей момент клієнтська частина:

1. формує структурований набір відповідей ResponseSet;
2. надсилає його на сервер для збереження;
3. ініціює розрахунок Score та отримує результат для відображення.

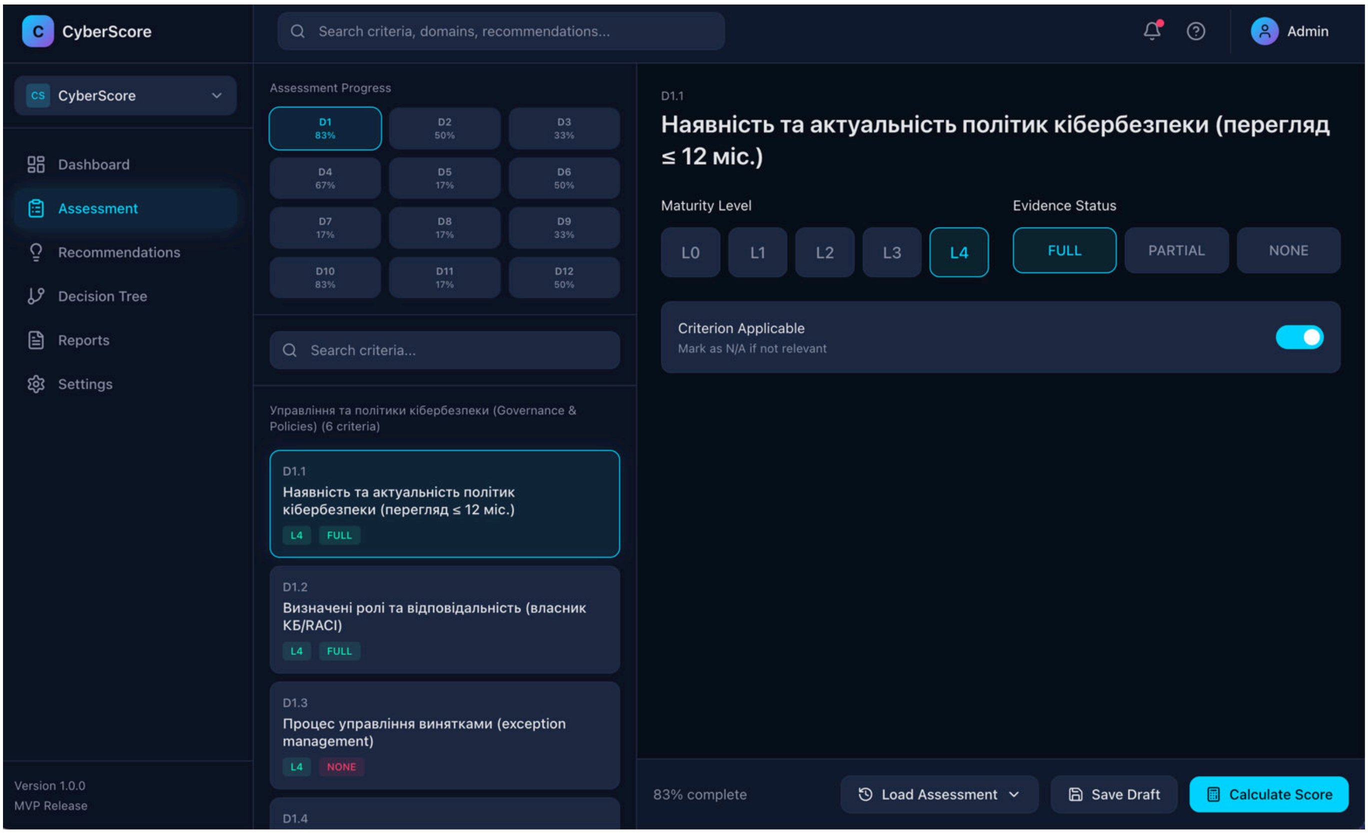

Рис. 3.11. Сторінка Assessment

```
export const submitResponses = async (data: ResponseSetCreate):
Promise<ResponseSetResponse> => {
  const response = await api.post('/v1/responses', data);
  return response.data;
};

export const runScoring = async (responseSetId: number):
Promise<ScoringResult> => {
  const request: ScoringRequest = { response_set_id: responseSetId };
  const response = await api.post('/v1/scoring/run', request);
  return response.data;
};
```

Рис. 3.12. Фрагмент коду функцій submitResponses та runScoring

Модуль Recommendations реалізує перехід від оцінки до керованого покращення. Він відображає список рекомендованих дій, впорядкований за пріоритетом. Кожна рекомендація представлена як картка, де зібрано:

- назву дії, сформульовану як практичний крок;
- прив’язку до домену та критерію з ідентифікатором;
- очікуваний вплив і складність та ресурсозатратність;
- цільову зміну рівня.

Користувач формує набір обраних заходів, використовуючи перемикач на картках, а система в реальному часі перераховує прогнозовану оцінку, показуючи очікуване зростання інтегрального показника при імплементації вибраних дій. Таким чином інтерфейс підтримує керування портфелем покращень: користувач балансує між впливом і зусиллями, отримуючи кількісну інтерпретацію рішень. Крім цього, на картках присутні кнопки генерації артефактів.

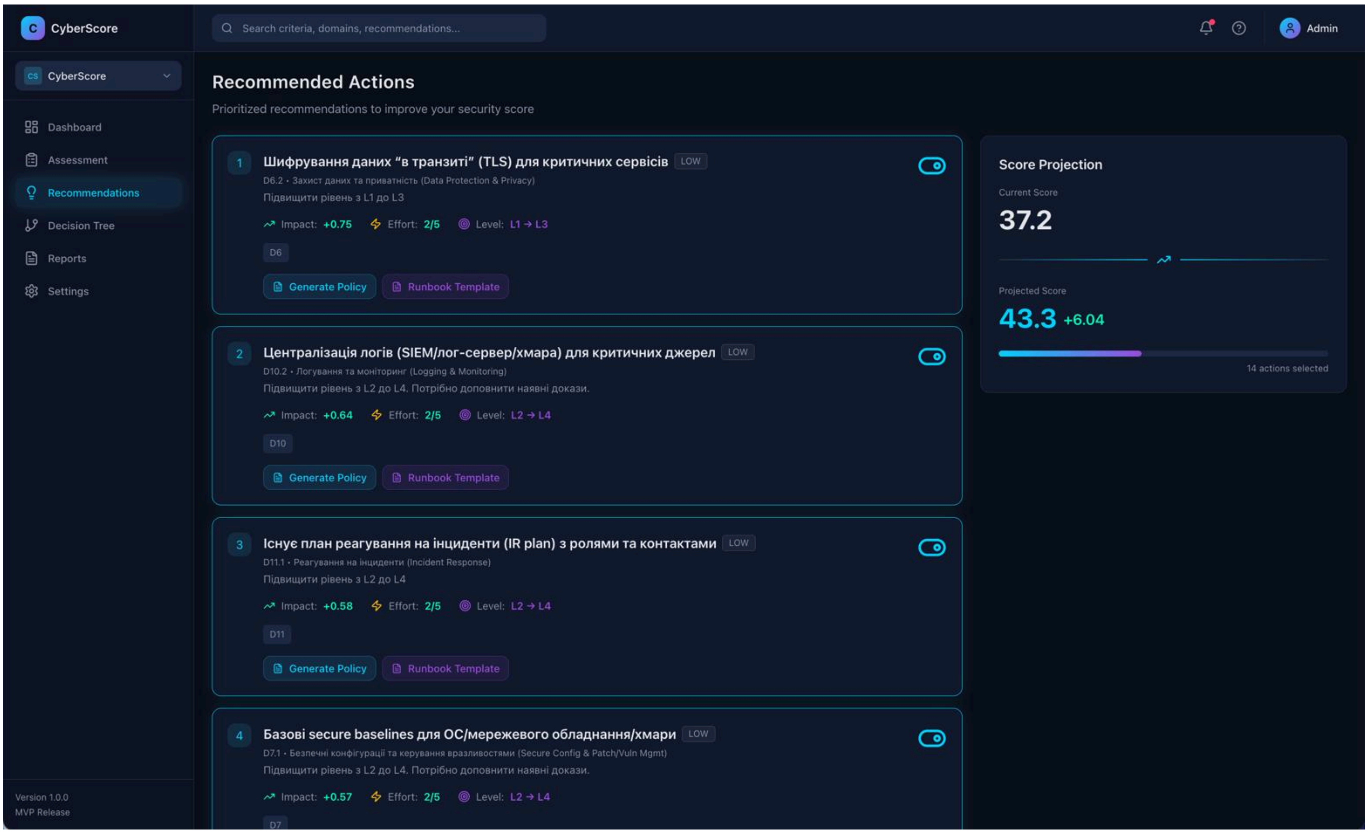

Рис. 3.13. Сторінка Recommendations

З точки зору інтеграції, Recommendations використовує дані останнього розрахунку ScoringRun і результат генерації рекомендацій RecommendationRun. Перерахунок прогнозу Score може виконуватись або клієнтською симуляцією, на основі impact, або серверною симуляцією, коли Backend обчислює projected_score для обраного набору дій.

Сторінка Decision Tree відображає логіку оцінювання і вплив критичних умов у вигляді графа, тобто дерева рішень, з вузлами та ребрами. Під вузлом тут розуміється елемент логіки, певна умова, розрахунок, застосування штрафу, а ребра задають напрям переходу між етапами.

Інтерфейс реалізує інтерактивне полотно з можливістю масштабування, пошуку вузлів і вибору елементів. При кліку на вузол праворуч відкривається панель деталізації, де показано тип вузла, його значення у процесі обчислення та пояснення важливості. Дерево підсвічує певний шлях, який актуальний саме для поточного набору відповідей. До прикладу, при невиконанні критичної умови активується гілка із застосуванням штрафу та поясненням наслідків для інтегральної оцінки. У результаті користувач бачить не тільки фінальний Score, а й причинно-наслідкову траєкторію його формування.

Крім пояснення, Decision Tree уніфікує логіку рекомендацій: вузли, які ведуть до штрафу або знижують доменний показник, мають пов’язані дії, і через інтерфейс користувач переходить до відповідних рекомендацій або одразу ініціює генерацію потрібного артефакту.

Модуль Reports концентрує всі створені системою або користувачем документи: політики, плани, SOP, runbooks, підсумкові звіти оцінювання. Інтерфейс побудований як двопанельна структура: ліворуч список доступних документів із метаданими, а праворуч панель для перегляду вмісту без завантаження.

Користувач може завантажити окремий документ, оглянути його перед завантаженням, а також завантажити усе архівом.

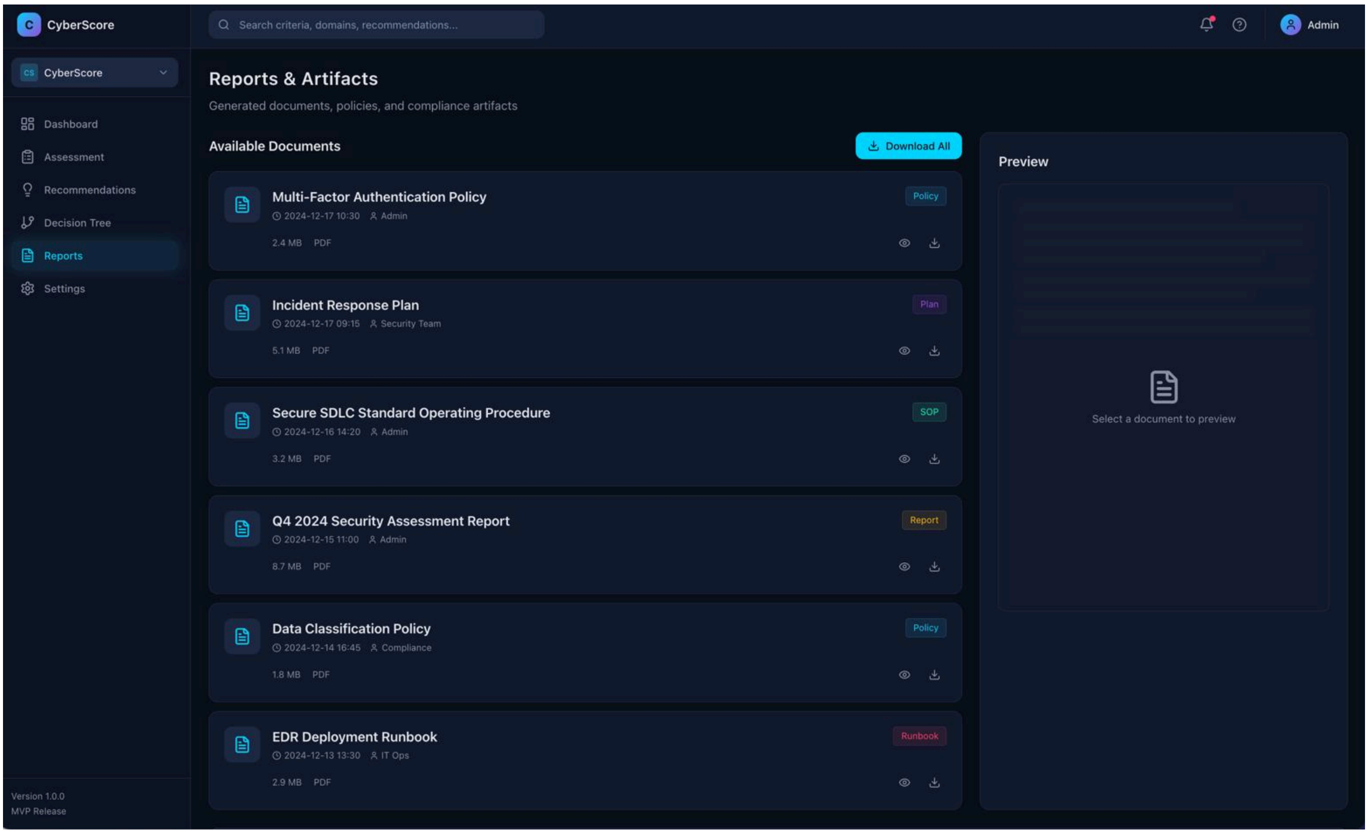

Рис. 3.14. Сторінка Reports & Artifacts

Сторінка налаштувань класично призначена для персоналізації та операційного керування. У блоці Profile користувач редагує базові атрибути. Окремим блоком винесені сповіщення, це набір перемикачів, що визначають, по рякі події системи дізнається користувач: завершення оцінювання, суттєві зміни Score, провали критичних гейтів, регулярні підсумки. Це безпосередньо пов’язано з функцією повторних оцінювань та контролю прогресу. Блок Security в інтерфейсі надає точки входу для безпекових операцій.

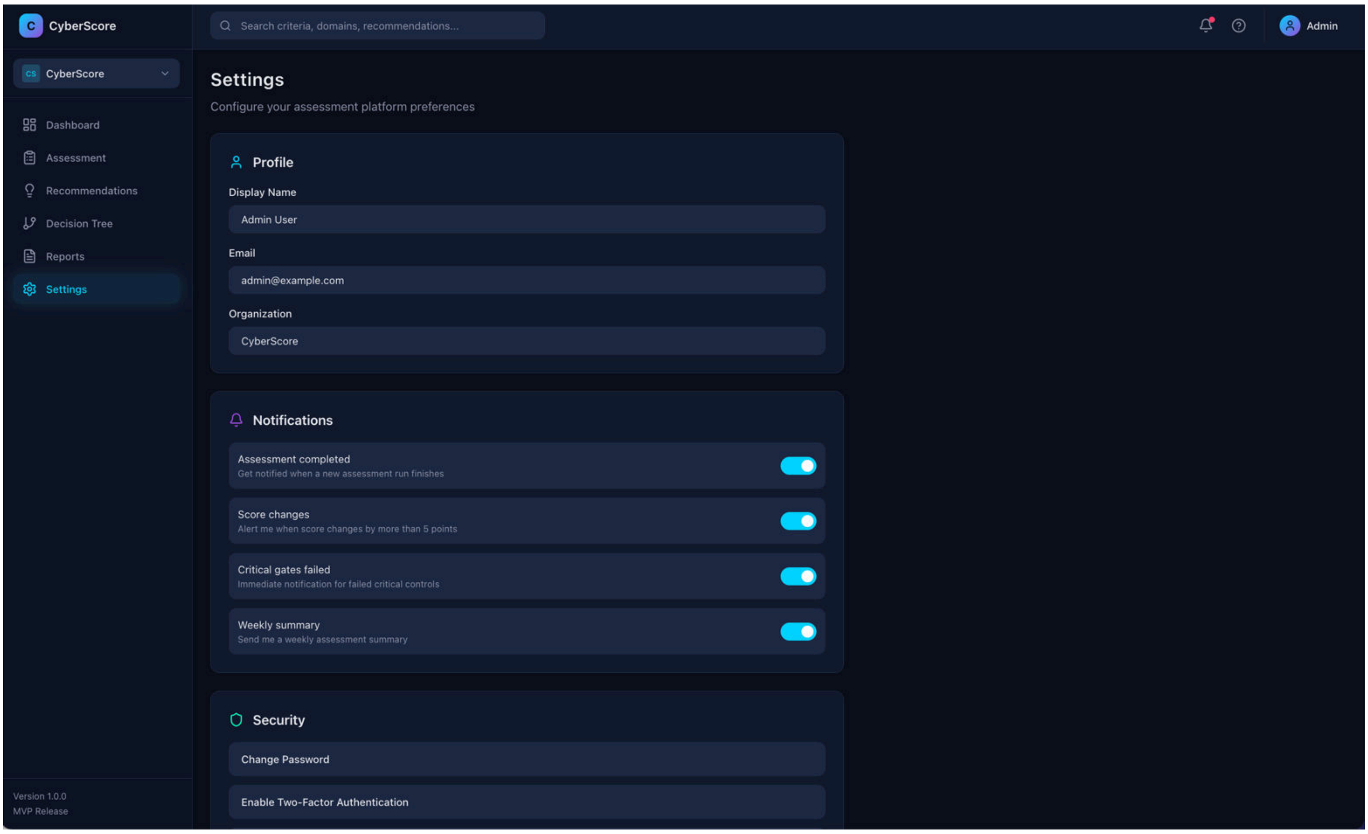

Рис. 3.15. Сторінка налаштувань

Фронтенд працює з чітко визначеними сутностями: версія опитування, домени та критерії, набір відповідей, оцінка з розрахунку, рекомендації, артефакти. На стороні інтерфейсу зберігається лише стани взаємодії, а всі значущі артефакти фіксуються на сервері.

3.5. Сценарій застосування системи

Цілісний сценарій роботи MVP можна описати як послідовність стабільних станів даних у БД.

На першому кроці користувач відкриває веб-інтерфейс, і Frontend запитує структуру опитувальника за конкретною версією. Backend повертає версію, домени та критерії з вагами. Користувач заповнює анкету, для кожного критерію задаючи рівень L0 - L4, ознаку застосовності та статус доказовості. Вже на цьому етапі формується семантично багатий набір даних.

Далі Frontend надсилає масив відповідей у Backend для створення ResponseSet. У БД з'являється запис «проходження», а всі відповіді зберігаються як рядки таблиці responses. На цьому кроці система отримує перший артефакт аудиту: можна відтворити, які саме відповіді були дані, незалежно від подальших обчислень.

Після збереження відповідей користувач ініціює розрахунок Score. Backend завантажує домени та критерії відповідної версії, переносить відповіді в scoring.py, виконує перенормування ваг для N/A і обчислює доменні та інтегральну оцінку. Результат фіксується в scoring_runs. З цього моменту оцінювання стає статистикою, а система може будувати тренди, порівняння, повторні рекомендації. До кожного проходження дозволено повернутися та провести повторний розрахунок, завантаживши відповідний стан.

Після отримання Score система формує рекомендації. Вони зберігаються в recommendation_runs, і Frontend може відображати їх як перелік кроків із пріоритетом, очікуванним впливом та підказками чому це важливо.

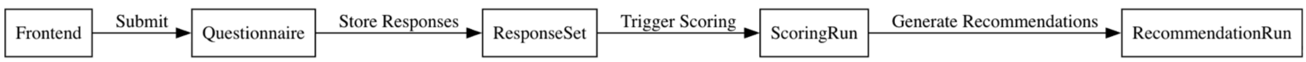

Рис. 3.16. Діаграма послідовності виконання дій

3.6. Контейнеризація та розгортання MVP

Контейнеризація реалізована через окремі Dockerfile для Frontend і Backend. У Frontend використано multi-stage build, спочатку збирається production білд React, потім результат копіюється в Nginx образ, який віддає статичні файли.

Backend-контейнер будується на базі Python образу, встановлює залежності, копіює код і запускає uvicorn. Також додано healthcheck, який періодично перевіряє доступність endpoint-у /health. Це дозволяє оркестратору Compose визначати, чи сервіс працює в реальному часі.

Для повноцінного запуску MVP у відповідності до описаної архітектури функціонально створений файл docker-compose, що має compose-конфігурацію, яка:

1. підніме три сервіси: frontend, backend, postgres;
2. об’єднає їх у спільну мережу;
3. задасть креденшели PostgreSQL;
4. за потреби підключить volume для збереження даних БД.

```yaml
services:
  postgres:
    image: postgres:15-alpine
    container_name: cyberscore_postgres
    environment:
      POSTGRES_DB: ${POSTGRES_DB:-cyberscore}
      POSTGRES_USER: ${POSTGRES_USER:-cyberscore_user}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD:-cyberscore_password}
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql:ro
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER:-
cyberscore_user}"]
      interval: 5s
      timeout: 5s
      retries: 5
    networks:
      - cyberscore_network

  backend:
    build:
      context: ../backend
      dockerfile: Dockerfile
    container_name: cyberscore_backend
    environment:
      DATABASE_URL: ${DATABASE_URL:-
postgresql://cyberscore_user:cyberscore_password@postgres:5432/cyberscore}
      CORS_ORIGINS: ${CORS_ORIGINS:-http://localhost:3000}
      ENVIRONMENT: ${ENVIRONMENT:-development}
    ports:
      - "${BACKEND_PORT:-8000}:8000"
    depends_on:
      postgres:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localhost:8000/health"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 30s
    networks:
      - cyberscore_network
    restart: unless-stopped

  frontend:
    build:
      context: ../frontend
      dockerfile: Dockerfile
    container_name: cyberscore_frontend
    ports:
      - "${FRONTEND_PORT:-3000}:80"
    depends_on:
      - backend
    networks:
      - cyberscore_network

networks:
  cyberscore_network:
    driver: bridge

volumes:
  postgres_data:
```

Рис. 3.17. Файл docker-compose.yml

ВИСНОВКИ ДО РОЗДІЛУ 3

У третьому розділі реалізовано MVP-застосунок, який переводить метод оцінки рівня кібербезпеки підприємства з теоретичної моделі в практичну систему. Реалізація підтримує базовий цикл завантаження версійної структури опитування, введення відповідей, збереження результатів, обчислення інтегрального показника Score $\in [1;100]$, формування профілю доменних оцінок та генерацію пріоритетних рекомендацій. Таким чином, метод стає відтворюваним і придатним для апробації.

Архітектура побудована як контейнеризована система з розділенням компонентів: Frontend забезпечує інтерфейс взаємодії та візуалізацію результатів, Backend реалізує API та обчислювальну логіку, PostgreSQL зберігає версії опитування, відповіді й історію оцінювань. Таке розділення спрощує подальшу розбудову системи: зміни у формулі скорингу або правилах рекомендацій можуть впроваджуватися на сервері без перебудови інтерфейсу, а Frontend зберігає роль зрозумілого інтерфейсу для клієнта.

У моделі даних коректно відокремлено первинні та похідні сутності: відповіді користувача зберігаються окремо від результатів розрахунку, а результати оцінювання та рекомендації фіксуються як окремі запуски. Це забезпечує історизацію, порівнюваність у часі та можливість повторного перерахунку при зміні формули або параметрів оцінювання. Backend-ядро реалізує ключові елементи методу MVP-рівня: ваговий підхід, підтримку N/A через перенормування та врахування evidence-фактору, а модуль рекомендацій формує пріорітетний перелік дій із прогнозованим впливом на Score.

Frontend реалізує керований сценарій використання, де на Dashboard користувач бачить інтегральну оцінку та доменний профіль, у модулі Recommendations пріоритетні дії та прогноз зміни Score при їх виборі; у Reports простір для роботи з артефактами, у Settings централізовані налаштування. У підсумку, інтерфейс забезпечує зрозумілу інтерпретацію результатів та фокусує користувача на покроковому підвищенні показника за методом.

Разом із тим, аналіз реалізації показує, що частина функціональності перебуває на рівні MVP і потребує доведення до повної відповідності логіці розділу 2 та інтерфейсним модулям. Це не суперечить результатам розділу 3, а визначає чіткі напрямки наступної ітерації, які логічно переходять у підсумкові висновки роботи.

Подальший розвиток MVP до повної версії системи логічно зосередити на кількох послідовних кроках. Насамперед потрібно забезпечити повну відтворюваність розгортання, додавши керований механізм ініціалізації бази даних , щоб система запускалася на чистому середовищі без ручної підготовки таблиць і первинних даних опитування. Після цього доцільно поглибити обчислювальне ядро до повної моделі, описаної в розділі 2, з фіксацією проміжних параметрів розрахунку в результатах запуску оцінювання, що підвищить прозорість та аудитопридатність Score. Паралельно варто зробити прогноз Score точнішим, перенісши Score Projection у серверний перерахунок для довільного набору обраних дій, щоб прогноз ґрунтувався на симуляції оновлених значень критеріїв, а не лише на сумарному impact.

Наступним важливим кроком є реалізація Decision Tree як серверної функції: формалізувати дерево правил, додати endpoint-и для отримання дерева та активного шляху на основі поточних відповідей і результатів скорингу, а також зв’язати вузли дерева з конкретними рекомендаціями й штрафними умовами, щоб інтерфейс відображав не лише граф, а й керовану логіку підвищення показника. Після цього природно розширити модуль Reports та Artifacts, реалізувавши повний доказовий контур: генерацію документів за шаблонами, збереження метаданих, перегляд та завантаження та прив’язку артефактів до критеріїв і рекомендацій, що дозволить підтверджувати реальне виконання заходів.

Оскільки система працює з чутливою інформацією, наступним шаром має стати безпековий контур застосунку, а саме аутентифікація, ролі, базове журналювання дій та серверні налаштування користувача. Завершальним елементом інженерного доведення є запровадження системного тестування модульних тестів для нормалізації, обрахунку оцінки й рекомендацій, а також регресійних сценаріїв, які гарантуватимуть стабільність результатів при зміні формул, ваг або версій опитувальника.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ

У межах магістерської роботи об'єктом охорони праці є робоче місце фахівця з кібербезпеки або аналітика SOC та інженера-розробника, який забезпечує експлуатацію створеної системи оцінювання і виконує типові операції з інформаційно-комунікаційними системами: моніторинг подій, аналіз журналів, адміністрування сервісів, аудит, реагування на інциденти, а також супровід веб-застосунку та роботу з документацією. Така діяльність майже завжди пов'язана з тривалою роботою за ПК, високим рівнем інтелектуального навантаження й відповідальності за безперервність процесів, що робить критичними питання ергономіки, мікроклімату, освітлення, електро- та пожежної безпеки.

Закон України «Про охорону праці» є системою правових, соціально-економічних, організаційно-технічних та санітарно-гігієнічних заходів і засобів, спрямованих на збереження життя, здоров'я і працездатності працівника у процесі трудової діяльності [44].

У контексті цієї роботи охорона праці напряму впливає і на якість результатів оцінювання кібербезпеки: перевтома, погане освітлення чи незручна поза підвищують імовірність помилок при заповненні опитування та інтерпретації звітів, що може спотворити інтегральний бал і управлінські рішення.

Робоче місце фахівця з кібербезпеки переважно знаходиться в офісних приміщеннях, це кабінети, open-space, кімнати моніторингу, рідше у технічних зонах, наприклад серверні або комутаційні. За характером навантажень домінують сидяча робота, зорова напруга, повторювані рухи кистей в роботі з клавіатурою та мишкою, а також підвищене когнітивне та психоемоційне навантаження: інциденти, дедлайни, чергування в наднормований час.

З урахуванням чинних санітарних підходів і гігієнічної класифікації праці виробничі фактори доцільно групувати так:

1) Фізичні фактори виробничого середовища

Мікроклімат, тобто температура, відносна вологість, швидкість руху повітря. Для операторських робіт та залів обчислювальної техніки санітарні норми прямо вказують на необхідність підтримання оптимальних параметрів, зокрема для операторських робіт, пов’язаних із нервово-емоційним напруженням [45].

Світлове середовище важливе, оскільки недостатня або надмірна освітленість, відблиски на екрані, нерівномірність освітлення, дискомфортна яскравість у полі зору може викликати негативні наслідки. Загальні вимоги до природного і штучного освітлення для будівель та робочих місць регламентуються ДБН В.2.5-28:2018, зокрема визначено мінімальні вимоги до освітленості робочих місць із постійним перебуванням людей та підходи до нормування [46].

Шум провокований робочою технікою, серверним обладнанням поруч, переговорами в корпоративних просторах. Для задач кібербезпеки шум є не лише фактором комфорту, але й джерелом додаткового психофізіологічного навантаження, наслідком є втрата концентрації при аналізі інцидентів.

Електромагнітні та електростатичні поля провоковані блоками живлення, кабельними трасами, мережевим обладнанням, Wi-Fi точками. У практиці офісів рівні зазвичай не перевищують допустимі, але при щільному розміщенні обладнання фактор потребує контролю в рамках оцінки умов праці.

2) Небезпечні фактори

Ураження електричним струмом через пошкоджену ізоляцію, несправність мережевих фільтрів, некоректне підключення UPS, втручання в комутаційні шафи без допуску. Норми безпечної експлуатації електроустановок споживачів визначені відповідними правилами [47].

Пожежна небезпека: перевантаження електромережі, короткі замикання, перегрів блоків живлення, використання несертифікованих зарядних пристроїв, порушення правил прокладання кабелів, захаращення евакуаційних шляхів. Вимоги пожежної безпеки для будівель і приміщень встановлюються Правилами пожежної безпеки в Україні [48].

Травмонебезпечні фактори в офісі: спотикання через кабелі, падіння через слизькі покриття, травми при переміщенні обладнання, це монітори, системні блоки, UPS, ризики в серверній через обмежений простір, гарячі поверхні, велику кількість кабелів.

3) Фактори трудового процесу

Статичні навантаження через тривале сидіння, напруження шийно-плечової зони, фіксована поза; локальні динамічні навантаження, через повторювані рухи кистей, що підвищують ризик порушень опорно-рухового апарату.

Зорове навантаження через роботу з багатьма інтерфейсами одночасно, дашборди, SIEM, тикетинг, журнали, робота з дрібними елементами інтерфейсу, часте перемикання контекстів.

Нервово-емоційна напруженість: реагування на інциденти, дефіцит часу, чергування, відповідальність за доступність сервісів та коректність рекомендацій системи оцінювання. Підходи до оцінювання важкості та напруженості праці та принцип захисту часом формалізовані у державних санітарних нормах (гігієнічна класифікація праці [49].

Зауваження щодо нормативної бази для ПК: При формуванні вимог до робочого місця та інтерфейсу в актуальному полі доцільно спиратися на чинні стандарти ергономіки, серія ДСТУ ISO 9241, будівельні норми освітлення та санітарні норми мікроклімату. [50].

Враховуючи специфіку кібербезпекових задач і експлуатації розробленого ПЗ, заходи охорони праці доцільно будувати як комбінацію інженерних і організаційних рішень, орієнтованих на зниження втоми, помилок оператора та ризиків аварійних ситуацій.

Базові вимоги до компонування робочого місця та робочої пози для офісної роботи з відеотерміналами визначаються стандартом ДСТУ ISO 9241-5, а вимоги до характеристик робочого середовища, тобто чинники, що впливають на комфорт і ефективність відповідно ДСТУ ISO 9241-6 [51].

Для типового робочого місця кіберфахівця рекомендовано забезпечити:

- Регульоване крісло, з можливістю керувати висотою сидіння, нахилом спинки, поперековою підтримкою, за потреби з підставкою для ніг;

- Нейтральну позу: опора попереку; відсутність піднятих плечей; передпліччя мають опиратися на стіл або підлокітники без надмірного згинання зап’ястка.
- Раціональне розміщення засобів введення: клавіатура і миша в зоні досяжності без надмірного відведення плеча, бажана однакова висота площин для введення й письма.
- Розміщення моніторів має відбуватися так, щоб основний екран був у центральній зоні; верхня межа активної області не повинна примушувати до постійного закидання голови; при двох моніторах або симетрія, або чітко визначений головний монітор із мінімізацією поворотів голови.

Окремо для нашого програмного рішення ергономіка робочого місця доповнюється вимогами до зручності читання: достатній розмір шрифту, уникнення дрібних клікабельних елементів, логічне групування питань за доменами, обмеження візуального шуму. Це знижує ризик помилок при введенні відповідей і підвищує надійність результату оцінки, що є важливим для методичної частини дипломної роботи.

З огляду на вимоги до режиму праці та відпочинку, на рівні підприємства мають бути визначені: перерва для відпочинку і харчування, також локальні регламенти коротких перерв та мікропауз для зменшення статичного й зорового навантаження Загальна норма щодо перерви для відпочинку і харчування встановлена КЗпП України, ст. 66 [52].

Для робіт із високою часткою екранного часу світлове середовище є ключовим фактором зорової працездатності та профілактики головного болю та перевтоми. ДБН В.2.5-28:2018 задає підходи до нормування освітлення й мінімальні вимоги до освітленості робочих місць із постійним перебуванням людей, а також правила організації систем освітлення з урахуванням якості світла [46].

Практичні заходи для офісу моніторингу:

- розміщення робочих місць відносно вікон так, щоб уникати вікна в прямому полі зору та відблисків на екрані;
- застосування жалюзів та штор при високих контрастах;

- використання рівномірного загального освітлення;
- контроль появи відблисків у типових напрямках погляду, при роботі з SIEM-дашбордами, формами опитування, звітами.

Для операторських робіт, пов’язаних із нервово-емоційним напруженням, що характерно для SOC та реагування на інциденти, санітарні норми мікроклімату прямо орієнтують на підтримання оптимальних параметрів; у документі окремо наведено значення для таких приміщень, зокрема кабінетів, пультів, постів керування, залів обчислювальної техніки [45].

Технічні та організаційні заходи:

- налаштування кондиціювання без локальних струменів холодного повітря на робоче місце;
- підтримання адекватної вологості, оскільки сухе повітря підсилює дискомфорт очей при роботі з екраном;
- окремі вимоги для серверних: контроль температури та доступу, організація робіт так, щоб персонал не перебував там довше необхідного часу.

На відміну від звичайних офісних задач, кібербезпекові роботи включають періоди різкого росту навантаження, коли помилка може призвести до істотних наслідків. У таких умовах заходи охорони праці повинні враховувати:

- керування навантаженням, чергування ролей у команді, дублювання критичних рішень, правило «другої пари очей» для небезпечних змін;
- регламент комунікацій у каналах ескалації, зрозумілі чек-листи дій;
- профілактика вигорання, дотошна реалізація планування відпусток, ротація нічних змін, контроль понаднормових робіт.

Обґрунтуванням таких заходів слугує підхід гігієнічної класифікації праці, де прямо враховуються інтелектуальні, сенсорні та емоційні навантаження, а також режим роботи, і допускається застосування захисту часом як інструменту зниження ризиків при перевищенні нормативів [49].

Для забезпечення виконання вимог охорони праці роботодавець організовує навчання та перевірку знань з питань охорони праці відповідно до Типового положення НПАОП 0.00-4.12-05. Враховуючи контекст це важливо щонайменше у частині:

- первинного та повторного інструктажу для офісних працівників;
- цільових інструктажів перед роботами в серверній та електрощитовій або при підключенні UPS або мережевого обладнання;
- фіксації відповідальності та порядку дій при аварійних ситуаціях.

Пожежна безпека в офісній IT-інфраструктурі визначається концентрацією електрообладнання, зокрема ПК, моніторів, зарядних пристроїв, мережевих комутаторів, UPS, наявністю кабельних ліній і тривалим режимом роботи частини пристроїв. Організація пожежної безпеки здійснюється відповідно до Правил пожежної безпеки в Україні, Наказ МВС №1417 [48].

Основні профілактичні заходи:

- вільні евакуаційні шляхи, доступні вогнегасники, видимі плани евакуації;
- заборона використання несправних або саморобних подовжувачів, недопущення перевантаження мережі;
- впорядкування кабельного господарства, захист від механічних пошкоджень, відсутність тимчасових з’єднань;
- порядок дій при загорянні: повідомлення відповідних служб та керівництва, зупинка робіт, знеструмлення ділянки за умови уникнення ризиків, організована евакуація, застосування первинних засобів пожежогасіння лише за відсутності загрози життю.

Хоча більшість робіт виконується за робочою станцією, у межах експлуатації ІКС, частково, нашого програмного комплексу, можливі підключення або перекомунікації мережевого обладнання, роботи з UPS, заміна блоків живлення, підключення серверів у стійці. Такі роботи мають виконуватися з дотриманням Правил безпечної експлуатації електроустановок споживачів, НПАОП [47].

Практичні вимоги в нашому випадку:

- використання справних мережевих фільтрів та заземлених розеток;
- заборона робіт з електрообладнанням при видимих дефектах кабелів, вилок, розеток;

- обмеження непрофільних робіт для фахівців, які не мають відповідного допуску;
- організація доступу до серверної, зокрема контроль ключів та допусків, освітлення й проходів між стійками;
- мінімізація робіт в «горящому» темпі та планування технічних вікон.

Практичні рекомендації для впровадження й безпечної експлуатації системи оцінювання кібербезпеки варто оформити як керований цикл: Підготовка, Оцінювання, Валідація, План дій, Виконання, Контроль, Повторна оцінка із чіткими ролями та ритмом: призначте власника процесу, встановіть регулярність 1 раз на квартал з фіксацією версії опитування й дати; саме проходження плануйте як сесію 60–90 хв у час без піків інцидентів, із застосуванням правила «2 пар очей», а після отримання оцінки формуйте пакет впровадження за 3 робочі дні. Найпріорітетнішнішу п’ятірку рекомендацій з очікуваним ефектом, з призначенням відповідального та закладеним потрібним ресурсом або бюджетом; реалізацію рекомендацій розбийте на 30, 90 або 180 днів, з огляду на складність впровадження, а контроль виконання закріпіть як регулярний чек-пойнт 15 хв щотижня з актуалізацією статусів і ризиків.

Щоб мінімізувати помилки заповнення та хибні управлінські висновки через втому, введіть «захист часом» і стандарти екранної роботи: мікропаузи 5 хв кожні 50–60 хв та правило 20-20-20: кожні 20 хв 20 секунд дивитися на відстань ~6 м; робоче місце налаштуйте енергономічно: відстань до монітора 50-70 см, верхня межа екрана на рівні очей або трохи нижче, клавіатура і миша на одному рівні з опорою для передпліч; середовище підтримуйте стабільним, орієнтирами виступатимуть температура в діапазоні 22-24°C, вологість 40-60%, без прямого потоку холодного повітря, а освітлення доведіть до нормативної якості, здійсніть перевірку люксметром, усуньте відблиски перестановкою, жалюзями або замінною світильників. Зафіксуйте правило: якщо виконання оцінювання припадає на нічне чергування або період після інциденту — результати проходять додаткову валідацію або переносяться.

Щоб знизити електро- та пожежні ризики, стандартизовано закрийте побутові джерела аварійності: використовуються лише справні сертифіковані подовжувачі та UPS, забороняються тимчасові з’єднання та перевантаження ліній, впроваджується кабель-менеджмент без перетискань і кабелів у проходах, евакуаційні шляхи залишаються вільними, первинні засоби пожежогасіння мають бути доступними; роботи в серверній та електрощитовій виконують тільки особи з відповідним допуском та після інструктажу, а всі зміни, що можуть вплинути на доступність мережі або живлення, плануються як технічні вікна з коротким чек-листом дій і відкату.

ВИСНОВКИ ДО РОЗДІЛУ 4

Узагальнюючи, запропонований підхід до застосування системи оцінювання рівня кібербезпеки доцільно розглядати як регулярний керований процес, де якість результатів залежить від методики скорингу, умов праці та дисципліни експлуатації. Регламентація ритму оцінювань квартально чи піврічно, формалізована валідація результатів і планування виконання рекомендацій у горизонті 30, 90, 180 днів забезпечують доступний екологічний підхід до використання інструмента управління кіберстійкістю. Одночасно, впровадження вимірюваних ергономічних і санітарно-гігієнічних заходів знижує перевтому та імовірність помилок під час заповнення й інтерпретації звітів, підвищуючи достовірність інтегрального балу та рекомендацій. Дотримання електро- та пожежної безпеки, порядок у кабельному господарстві й допуски до робіт у технічних приміщеннях мінімізують інциденти інфраструктурного характеру, що безпосередньо впливають на безперервність роботи ІКС і на стабільність процесу оцінювання.

РОЗДІЛ 5
ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Цифровізація бізнес-процесів і масове впровадження інформаційно-комунікаційних систем трансформували екологічний профіль ІТ-діяльності, тепер вона більше не є нематеріальною, оскільки супроводжується споживанням електроенергії, використанням інженерної інфраструктури, типу охолодження, резервного живлення, а також утворенням відходів електричного та електронного обладнання і відпрацьованих батарей та акумуляторів. Базові правові засади державної політики у сфері довкілля, обов'язок запобігання шкоді та принципи раціонального природокористування визначаються Законом України «Про охорону навколишнього природного середовища» [53].

У межах цієї магістерської роботи, присвяченої методу оцінки рівня кібербезпеки підприємства та його програмній реалізації, екологічна складова має прикладний характер з таких причин:

- практичне застосування методу передбачає регулярні цикли оцінювання, зберігання результатів у наборах відповідей, розрахунках score, сформованих рекомендаціях, а в організаційному масштабі враховує підтримку ІТ-інфраструктури, що забезпечує доступність і цілісність сервісу;
- кібербезпека як функція підприємства часто ініціює додаткові засоби контролю та моніторингу, що може збільшувати обсяги даних і навантаження на обчислювальні ресурси;
- вимоги до надійності у реальних умовах можуть призводити до розширення інфраструктури: UPS, дублювання вузлів. Опосередковано до збільшення енергоспоживання та майбутніх потоків ВЕЕО, тобто відходів електричного та електронного обладнання.

Таким чином, екологічне забезпечення ІТ-рішень у сфері кібербезпеки доцільно розглядати як частину відповідального управління підприємством: мінімізувати ресурсні витрати, дотримуватись законодавчих вимог щодо відходів, а також узгоджувати екологічні заходи з вимогами охорони праці та пожежної безпеки [54].

Для підприємства, що експлуатує ІКС, ключовими є такі регуляторні рамки:

1. Загальні засади охорони довкілля

Закон України «Про охорону навколишнього природного середовища» встановлює принципи запобігання шкоді, екологічної безпеки та відповідальності за порушення природоохоронних вимог [53].

2. Енергоефективність та непрямий вплив через електроенергію

Організаційні та правові основи підвищення енергоефективності визначені Законом України «Про енергетичну ефективність». Для ІТ-інфраструктури це транслюється у вимоги та очікування щодо раціонального використання енергії, оптимізації навантажень та управління енергоспоживанням [55].

Непрямий вплив на атмосферне повітря через генерацію електроенергії в енергосистемі корелює із загальними вимогами охорони атмосферного повітря, визначеними Законом України «Про охорону атмосферного повітря» [56].

3. Управління відходами, зокрема ВЕЕО та батареями і акумулятори

Закон України «Про управління відходами» задає ієрархію управління відходами: пріоритет запобігання, підготовки до повторного використання, рециклінгу тощо; а також визначає обов’язки утворювачів відходів щодо належного поводження, обліку й передачі відходів уповноваженим суб’єктам [57].

Практичний контур державного обліку та звітності деталізується, зокрема, наказом Міндовкілля про затвердження порядку ведення державного обліку відходів та подання звітності і типової форми обліку [58].

Також впроваджується інституційна цифрова складова, певна інформаційна система управління відходами, що регулюється постановою КМУ [59].

4. Охорона праці та пожежна безпека на перетині з екологічними аспектами

Безпечне поводження з електронікою, батареями та накопичувачами даних одночасно є питанням екології й безпеки праці. Загальні обов’язки роботодавця щодо безпечних умов праці визначає Закон України «Про охорону праці» [54].

Вимоги до організації пожежної безпеки на підприємствах регламентуються Правилами пожежної безпеки в Україні [60].

5. Добровільні стандарти управління

Як організаційний інструмент системного зниження впливів доцільно орієнтуватися на:

- ДСТУ ISO 14001:2015, що стосується систем екологічного управління, для управління екологічними аспектами за циклом постійного поліпшення [61].
- ДСТУ ISO 50001:2020, що стосується систем енергетичного менеджменту, для управління енерговикористанням та енергорезультативністю [62].

Окремо, для підприємств із європейськими контрагентами або експортною діяльністю важливо враховувати підходи ЄС до ВЕЕО та батарей: Директива WEEE 2012/19/EU, про вимоги щодо роздільного збирання та належної обробки електронних відходів та Регламент 2023/1542 щодо батарей і відходів батарей [63, 64].

З урахуванням того, що розроблений у роботі метод реалізований як програмний продукт із типовою триланковою структурою, екологічні аспекти виникають у трьох «шарах» життєвого циклу:

1) Етап розробки та тестування

Екологічний вплив тут переважно опосередкований: робочі станції розробників, тестові середовища, збільшення компіляцій та збірок, трафік до репозиторіїв. Для зниження впливу актуальні практики lean-розробки: використання контейнерів для відтворюваності середовища без множинних стендів, оптимізація CI-циклів, обмеження зайвих тестових артефактів.

2) Етап впровадження та експлуатації

Система може працювати на існуючих серверах підприємства або у віртуалізованому середовищі. Основні аспекти:

- енергоспоживання серверів, мережевого обладнання, UPS, охолодження, що виступає непрямим впливом на енергосистему;
- обсяг даних у БД, політики зберігання та резервного копіювання;

- підтримка доступності, резервування, дублювання, часті оновлення потенційно збільшує матеріальну базу.

3) Етап оновлення та списання обладнання та матеріалів

Саме тут формується найбільш видимий екологічний ризик – ВЕЕО, з використаних ПК, серверів, мережевого обладнання, носіїв, акумуляторів UPS, витратні матеріали. Правові наслідки для підприємства визначаються законодавством у сфері управління відходами та підзаконними актами щодо обліку та звітності [57].

Джерела, тобто робочі станції персоналу, сервери або віртуальні хостинги застосунку, PostgreSQL, мережеве обладнання, UPS, кондиціювання та вентиляція технічних приміщень.

Механізм впливу збільшення електроспоживання призводить до пропорційного зростання непрямих викидів в енергосистемі, а також до підвищення тепловиділень і потреби в охолодженні. Законодавча рамка раціонального використання енергії закріплена у Законі України «Про енергетичну ефективність», а загальна рамка охорони повітря — у Законі України «Про охорону атмосферного повітря» [55].

Серед локальних фізичних чинники є шум та вібрація відвентиляторів серверів і кліматичного обладнання, локальні тепловиділення, електромагнітні поля в зоні силового та мережевого обладнання. Особливість чинників в тому, що вони мають переважно локальний характер і тісніше пов’язані з охороною праці та безпечними умовами експлуатації, ніж з класичною екологією. Водночас вони непрямо впливають на довкілля через потребу в додатковому охолодженні та енергії. Закон України «Про охорону праці» та Правила пожежної безпеки в Україні задають рамку організаційних вимог до таких зон.

Для нашого рішення найбільш вагомими та керованими впливами є енергоспоживання і управління ВЕЕО та батареями, оскільки вони мають чітке правове регулювання і довгострокові наслідки.

У межах ієрархії управління відходами, визначеної Законом України «Про управління відходами», пріоритетом є запобігання та підготовка до повторного використання.

Практичні заходи для підприємства, що застосовує метод оцінки кібербезпеки:

- вибір енергоефективного та ремонтопридатного обладнання, наприклад модульних конфігурацій, апгрейду RAM та SSD, замінних вентиляторів та БЖ, щоб зменшити темп списання і майбутні обсяги ВЕЕО.
- типізація моделей для спрощення ремонту, взаємозамінності компонентів і повторного використання.
- Архітектурна оптимізація, у контексті реалізації нашого застосунку пріоритетом є розміщення сервісу на наявних ресурсах підприємства з мінімізацією дублювання середовищ. Це корелює з енергоефективністю як організаційним принципом.

Враховуючи Закон України «Про енергетичну ефективність», доцільно закласти організаційні та технічні підходи до зниження енерговитрат ІТ-сервісу [55].

Рекомендовані дії в контексті програмної реалізації методу:

- Консолідація та віртуалізація, контейнеризація замість виділеного «заліза» під кожну функцію;
- Раціональні політики зберігання для результатів оцінювання, логів доступу та резервних копій: зберігати рівно стільки, скільки потрібно для аудиту та аналізу трендів і вимог підприємства, уникаючи неконтрольованого росту БД;
- Планування ресурсомістких операцій у часові вікна з меншим навантаженням;
- Енергетичний менеджмент як система. Орієнтиром може слугувати ДСТУ ISO 50001:2020 та практика впровадження енергоменеджменту, закріплена відповідними урядовими рішеннями для організаційного сектору.

Особливістю ІТ-відходів у кібербезпеці є наявність носіїв даних, які не можна передавати на повторне використання без контрольованого стирання. Тому екологічна політика має бути узгоджена з безпековою політикою.

Законодавство у сфері управління відходами вимагає належного поводження з відходами, а також обліку та звітності; ці вимоги деталізуються профільними порядками та формами [57].

Рекомендована організація потоків на підприємстві:

- окремі місця для електронних плат і кабелів, батарей та акумуляторів, картриджів і витратних матеріалів;
- маркування, інструктаж персоналу, заборона змішування різних фракцій, визначення відповідальних осіб;
- передача відходів лише суб’єктам, які мають належні дозвільні документи на операції з відходами, із документуванням;
- ведення обліку та звітності у формах і за правилами, визначеними уповноваженим органом.

Для батарей та кумуляторів додатково критичні вимоги пожежної безпеки: недопущення механічних пошкоджень, контроль умов складування, мінімізація ризиків загоряння, відповідно до Правил пожежної безпеки в Україні [60].

Для закріплення практик у вигляді процесу, доцільно впроваджувати елементи системи екологічного управління за ДСТУ ISO 14001:2015. У контексті нашого застосунку це може бути реалізовано через:

- реєстр екологічних аспектів для IT-функції;
- KPI за річним споживанням електроенергії IT-вузла, обсяг БД та резервних копій, частка повторного використання обладнання, маса та кількість ВЕЕО переданих на ліцензовану обробку;
- план заходів з енергооптимізації, політики, графік оновлення та ремонту, навчання персоналу.

Практично пропонується слідувати декількома рекомендованими шляхами. В домені енергики та цифрової гігієни інфраструктури почніть із вимірювання базової лінії: за 7–14 днів зніміть середнє споживання kWh/добу та пікові навантаження, і зафіксуйте вихідні характеристики комплектуючих для контейнерів та PostgreSQL. Далі протягом 30-60 днів введіть технічні обмеження, що дають швидкий ефект:, мінімізуйте кількість постійно увімкнених тестових стендів до, поставте обмеження на використання ресурсів для контейнерів, оптимізуйте БД, і застосуйте політики зберігання: результати оцінювання мають зберігатися 24 міс., логи доступу та API 90 днів, деталізована телеметрія 30 днів з агрегацією метрик. Цільові KPI на квартал зміна у 10-15% середнього енергоспоживання та 20-30% темпу приросту сховища.

Паралельно, впродовж першого місяця, створіть інвентаризаційний реєстр серверів, ПК, мережі, UPS, носіїв із датою введення, станом і планом ремонту та заміни; у політиці закупівель зафіксуйте вимогу ремонтопридатності та уніфікації моделей. Для зменшення списань запровадьте перерозподіл техніки за класами задач і обов’язкову санітизацію носіїв перед повторним використанням. KPI в цьому домені ≥70% обладнання після виведення з експлуатації має йти на ремонт та внутрішнє перевикористання або обробку відходів.

З ВЕЕО та батареями пропонується зробити роздільний збір, впровадити безпечне зберігання, передачу та документування. Протягом 3 місяця впровадження оформіть процес поводження з відходами, організуйте 3 потоки, наприклад електроніка та кабелі; батареї та UPS; картриджі, у марковані контейнери й розробіть коротку інструкцію персоналу, заборону змішування фракцій, відповідальних осіб і журнал обліку. Передавайте ВЕЕО та батареї лише уповноваженим операторам із актами та накладними; вивезення плануйте щоквартально або при 80% заповненні контейнерів; для акумуляторів забезпечте умови, що мінімізують ризик пошкодження та займання. KPI в цьому домені 0% змішаних фракцій, 100% передачі з документами, відсутність інцидентів при тимчасовому зберіганні.

ВИСНОВКИ ДО РОЗДІЛУ 5

У підсумку, екологічні наслідки впровадження та експлуатації розробленої системи оцінки рівня кібербезпеки визначаються не самим програмним забезпеченням, а ресурсами, які його підтримують: електроспоживанням серверів, мережі, UPS та життєвим циклом IT-обладнання. Найбільш керованими й нормативно чітко регламентованими є саме потоки електронних відходів і відпрацьованих акумуляторів, тому підприємству доцільно будувати практику на вимогах Закону України «Про охорону навколишнього природного середовища» та Закону України «Про управління відходами» із пріоритетом запобігання утворенню відходів, повторного використання, роздільного збирання, документованої передачі уповноваженим суб’єктам і ведення обліку та звітності. Одночасно, скорочення непрямого впливу через електроенергію забезпечується організаційними та технічними заходами енергоефективності, що узгоджується із вимогами у сфері енергоефективності та підходами енергоменеджменту. Запропонований у роботі 90-денний план із KPI у споживанні, обсягах зберігання, часткою повторного використання та переробки, відсутності змішування фракцій переводить екологічні вимоги в керований процес, який одночасно підтримує цілі кібербезпеки і знижує ризики для персоналу та інфраструктури під час поводження з батареями і технікою.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано прикладну науково-технічну проблему формалізованого, відтворюваного та придатного до програмної реалізації оцінювання рівня кібербезпеки підприємства за результатами опитування з подальшим формуванням керованих рекомендацій щодо підвищення інтегрального показника. Значення розв’язаної проблеми для практики полягає в тому, що підприємства отримують інструмент системної самооцінки, який дозволяє перевести якісні судження про стан захисту в числовий вимір, зафіксувати результат у часі, визначити слабкі місця та отримати структуровані кроки покращення. Значення для науки полягає у формалізації процедури агрегації багатьох критеріїв під невизначеністю, обґрунтуванні використання багатокритеріальних методів вагового узгодження та нечіткої інтерпретації лінгвістичних оцінок у задачі оцінювання кібербезпеки підприємства, а також у відтворюваному представленні методу через модель даних та програмний прототип.

У першому розділі систематизовано понятійний апарат і досліджено кібербезпеку підприємства як соціотехнічну область, де результат захисту визначається сукупністю організаційних, процесних і технічних практик, а оцінювання має враховувати як наявність контролів, так і їх фактичну реалізацію. Виконано аналіз механізмів забезпечення кібербезпеки підприємства та показано доцільність ризикоорієнтованого підходу у прийнятті управлінських рішень. Проведено огляд і глибокий порівняльний аналіз підходів до оцінювання рівня кібербезпеки та зрілості, у результаті якого визначено вимоги до методу, що розробляється: вимірюваність, прозорість обчислень, узгодженість структури оцінювання з визнаними рамками, придатність до самооцінки, чутливість до критичних провалів, можливість відтворення розрахунків і перерахунку результатів при оновленні методики. У межах постановки задач сформульовано виходи методу як інтегральний бал у шкалі $S \in [1;100]$, профіль доменних оцінок, перелік вузьких місць і керована логіка подальших дій через рекомендації та дерево рішень.

У другому розділі розроблено формалізований метод оцінки рівня кібербезпеки підприємства, який визначає структуру оцінювання, процедуру збору даних, правила їх інтерпретації та обчислення інтегрального показника. Структуру оцінювання побудовано у вигляді доменно-критеріальної моделі з 12 доменів, узгоджених із визнаними підходами до управління кібербезпекою та каталогами практик. Доменна частина методу включає: D1 Governance & Policies, D2 Risk Management & Compliance, D3 Asset & Service Inventory, D4 Identity & Access Management (IAM), D5 Awareness & HR Security, D6 Data Protection & Privacy Controls, D7 Secure Configuration & Patch and Vulnerability Management, D8 Network Security, D9 Endpoint & Application Security, D10 Logging, Monitoring & Detection, D11 Incident Response (IR), D12 Backup, Recovery & Resilience. Для кожного домену визначено набір критеріїв, через які збираються відповіді користувача, а внесок кожного домену в інтегральний показник задається ваговим коефіцієнтом, нормованим так, щоб сума ваг дорівнювала одиниці.

Ключовим результатом другого розділу є поєднання вагового узгодження важливості компонентів оцінювання та коректного опрацювання лінгвістичних відповідей. Ваги доменів визначаються методом аналізу ієрархій AHP, де експертні попарні порівняння оформлюються у матрицю, на основі якої обчислюється вектор ваг. У методі передбачено перевірку узгодженості експертних суджень як обов’язковий елемент обґрунтування вірогідності результату. Вхідні дані від користувача формуються у лінгвістичній шкалі рівнів L0 – L4, що відповідає практиці анкетування у сфері кібербезпеки та дозволяє користувачу обирати зрозумілі градації без потреби у точних числових вимірах. Для перетворення лінгвістичних оцінок у числовий простір метод використовує нечіткі множини й трикутні нечіткі числа TFN, а також правило дефазифікації через центроїд для отримання чіткого значення, придатного до подальших обчислень. Отримані числові оцінки агрегуються на рівні доменів з урахуванням ваг критеріїв, а потім інтегруються в загальний Score.

У метод введено механізм підвищення достовірності результатів самооцінки через коефіцієнт доказовості, що коригує оцінку критерію залежно від наявності підтвердних артефактів. Використано три фіксовані значення коефіцієнта: 1.00 для підтвердженої відповідності, 0.85 для часткової відповідності, 0.70 для відсутності підтвердження. Це рішення зменшує ризик завищення показника при декларативному заповненні анкети та забезпечує узгодженість між заявленим рівнем і фактом наявності доказів. Для коректної роботи з критеріями, які не застосовуються до конкретного підприємства, передбачено ознаку застосовності N/A та механізм перенормування ваг, що перерозподіляє вплив тільки між релевантними критеріями, зберігаючи інтерпретованість інтегрального результату. Для запобігання ситуаціям, коли високі значення окремих компонентів приховують критичні провали в базових практиках, у методі введено штрафні правила для критичних умов, які знижують підсумковий Score за наявності зафіксованих провалів за визначеними критичними критеріями. Завершальним етапом обчислень є приведення інтегрального значення до шкали [1;100] та формування профілю доменних оцінок, що дозволяє відобразити структуру сильних і слабких сторін. Також сформульовано процедуру виявлення слабких місць за критеріями та доменами як основи для подальшого формування рекомендацій.

У третьому розділі реалізовано MVP застосунку, який відображає розроблений метод у вигляді працюючої інформаційної системи та забезпечує повний цикл опитування, збереження відповідей, розрахунку Score, формування рекомендацій, відображення результатів. Архітектурне рішення побудовано як контейнеризована трирівнева система з трьома сервісами: Frontend, Backend і база даних, що забезпечує відтворюваність розгортання, простоту запуску та повторюваність демонстрації результатів. Frontend реалізовано на React та TypeScript і забезпечено вебсервером Nginx для віддачі статичних ресурсів та роботи з HTTP-викликами до API. Backend реалізовано на Python та FastAPI з REST API для отримання структури опитування, приймання відповідей, виконання обчислень та генерації рекомендацій. Персистентне зберігання даних реалізовано у PostgreSQL, включно з версійністю опитування, відповідями, результатами обрахунку оцінки та рекомендаціями. Це забезпечує відтворюваність розрахунків і можливість повторного аналізу результатів на основі зафіксованих у базі даних вхідних значень.

У моделі даних забезпечено коректне відокремлення первинних і похідних сутностей. Первинні сутності описують структуру опитування та відповіді користувача, похідні сутності фіксують результати обчислень і сформовані рекомендації. Впроваджено сутність QuestionnaireDefinition для версійності опитувальника, сутності Domain і Criterion для доменно-критеріальної структури оцінювання, сутність ResponseSet як контейнер проходження опитування для конкретної версії, сутність Response як відповідь на конкретний критерій з рівнем L0 – L4, ознакою застосовності N/A та статусом доказовості, сутність ScoringRun для фіксації інтегрального Score і профілю доменних оцінок, сутність RecommendationRun для збереження сформованих рекомендацій у вигляді структурованих об’єктів. Для збереження доменних оцінок і рекомендацій використано JSON та JSONB подання, що зберігає гнучкість структури виходів без втрати реляційної цілісності основних даних та дозволяє відтворити результат розрахунку для конкретного набору відповідей.

Backend реалізує три основні прикладні сценарії роботи: отримання структури опитування за версією, створення набору відповідей і запуск обчислення інтегрального показника з формуванням рекомендацій. Отримання структури опитування реалізовано як запит, який повертає домени, критерії та їх метадані для побудови форми у клієнтській частині без жорстко зафіксованих у фронтенді переліків. Створення ResponseSet забезпечує запис відповідей у базу даних з прив’язкою до версії опитування та дозволяє зберігати історію проходжень. Запуск обчислення формує ScoringRun, де фіксується результат обчислення Score і доменні оцінки, а також формує RecommendationRun з пріоритетним переліком рекомендацій. Реалізація на сервері робить обчислення детермінованими, централізовано керованими та однаковими для всіх клієнтів, що важливо для коректної апробації методу та контролю змін у формулі, вагах і правилах.

Frontend реалізує керований сценарій використання системи і зосереджений на коректному зборі вхідних даних та зрозумілому поданні виходів методу. На Dashboard реалізовано відображення інтегрального показника Score, блоку Domain Scores у вигляді набору карток доменів для візуального зіставлення рівнів зрілості між напрямами, а також блоку історії оцінювань у вигляді списку запусків, що дозволяє відстежувати динаміку результатів між проходженнями. Модуль Assessment реалізує формування вхідних даних для математичного ядра: для кожного критерію користувач задає рівень L0 – L4, застосовність N/A та статус доказовості у встановлених значеннях, після чого запускає обчислення. Клієнтська частина формує структурований ResponseSet, надсилає його на сервер для збереження і ініціює розрахунок Score та отримання результату для відображення. Модуль Recommendations реалізує подання сформованих рекомендацій у вигляді пріоритетного переліку карток, де для кожної рекомендації відображено формулювання практичної дії, прив’язку до домену та критерію, атрибути очікуваного впливу, складності та ресурсозатратності, а також цільову зміну рівня. Інтерфейс підтримує відбір рекомендацій користувачем для подальшої реалізації та закріплює керованість підвищення показника шляхом фіксації пріоритетів. Реалізовано сторінку Decision Tree як компонент візуалізації логіки оцінювання і впливу рішень на підсумковий результат, де вузли відображають кроки оцінювання та переходи між станами при зміні умов, а ребра задають напрям переходу між кроками. Також реалізовано функціональність роботи з документами, що дозволяє завантажувати окремі документи, переглядати їх перед завантаженням і отримувати архів, а сторінка налаштувань реалізує типову персоналізацію та керування сповіщеннями про події системи в межах наявної структури інтерфейсу.

Якісні результати, отримані в роботі, полягають у створенні методично узгодженої процедури оцінювання, яка має чітко визначені вхідні дані, проміжні перетворення та вихідні артефакти. Метод забезпечує прозору інтерпретацію результатів на рівні доменів і дозволяє переходити від діагностики стану до керованого планування покращень. Програмний прототип забезпечує цілісність життєвого циклу оцінювання через збереження версії опитування, відповідей, запусків скорингу та рекомендацій, що робить оцінку відтворюваною і придатною для накопичення історичних даних. Архітектурні рішення, модель даних та REST API забезпечують розширюваність опитувальника через версійність і можливість модернізації формули скорингу на серверній стороні без зміни клієнтської логіки збору даних.

Кількісні показники здобутих результатів включають доменну структуру оцінювання з 12 доменів, уніфіковану шкалу рівнів L0 – L4 для відповідей на критерії, інтегральний показник Score у шкалі $S \in [1;100]$, фіксований коефіцієнт доказовості з трьома рівнями 1.00, 0.85 та 0.70, механізм обробки застосовності N/A із перенормуванням ваг, а також реалізовану контейнеризовану систему з трьома сервісами Frontend, Backend та бази даних. У програмній частині реалізовано три ключові прикладні сценарії API, що охоплюють повний цикл оцінювання, а результати обчислень і рекомендації фіксуються у вигляді окремих запусків, що підтримує багатократне повторення оцінювання та порівняння результатів у часі.

Вірогідність отриманих результатів обґрунтована методичними та інженерними рішеннями, реалізованими у роботі. Метод базується на доменно-критеріальній структурі, узгодженій із визнаними рамками та стандартами у сфері кібербезпеки та управління ризиками, що підтримує змістову валідність оцінювання. Вагові коефіцієнти доменів визначаються AHP з передбаченою перевіркою узгодженості, що знижує ймовірність випадкових або суперечливих експертних суджень у структурі ваг. Лінгвістичні відповіді перетворюються в числові значення через формалізовані нечіткі множини та дефазифікацію, що забезпечує стабільність перетворення та однаковість розрахунків. Коефіцієнт доказовості формалізує вплив підтвердження на оцінку критерію і зменшує похибку, пов’язану з суб’єктивністю самооцінки. Відтворюваність результатів забезпечена тим, що в базі даних фіксується версія опитування, набір відповідей і результати запусків скорингу та рекомендацій, а обчислення виконуються на сервері в контрольованому середовищі. Контейнеризація відтворює конфігурацію середовища виконання та зменшує ризики, пов’язані з різницею інсталяцій і залежностей.

Рекомендації щодо наукового використання здобутих результатів полягають у застосуванні розробленої формалізації як основи для подальшої валідації на вибірках підприємств і накопичення емпіричних даних для уточнення ваг, правил штрафів і параметрів рекомендацій у межах уже реалізованої моделі версійності. Формальна процедура методу, збереження запусків скорингу та рекомендацій, а також відокремлення первинних і похідних даних створюють умови для дослідження стабільності оцінки, чутливості інтегрального показника до змін окремих критеріїв і формування довірчих інтервалів на основі повторних проходжень.

Рекомендації щодо практичного використання здобутих результатів полягають у впровадженні MVP як інструмента первинної діагностики стану кібербезпеки підприємства та підтримки управлінського циклу покращень. Система придатна для регулярного повторного проходження опитування з фіксацією результатів, що дозволяє відстежувати динаміку Score у часі і вимірювати ефект від реалізованих заходів. Профіль доменних оцінок і виявлені вузькі місця забезпечують основу для пріоритезації робіт за напрямами D1–D12, а модуль Recommendations формує структурований перелік дій з атрибутами пріоритету, очікуваного впливу та ресурсності, що може використовуватись як основа плану підвищення рівня кібербезпеки. Збереження доказовості у відповідях дозволяє використовувати результати як матеріал для внутрішніх перевірок і підготовки управлінської звітності з прив’язкою до підтвердних артефактів. Контейнеризоване розгортання забезпечує швидке впровадження прототипу в тестовому або навчальному середовищі без складної інсталяції залежностей та дозволяє застосовувати матеріали роботи у навчальному процесі для демонстрації повного циклу оцінювання з програмною реалізацією.

Загалом виконана робота завершила побудову методичної та програмної основи для оцінювання рівня кібербезпеки підприємства, забезпечила кількісне представлення результатів через Score у шкалі [1;100], надала інструментарій доменного аналізу, фіксації історії проходжень і формування рекомендацій, а також створила відтворюваний MVP, який може бути використаний у практичній діяльності підприємств і як база для подальших досліджень у межах визначеної предметної області.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Boyes, Hugh & Higgins, Matthew. An Overview of Information and Cyber Security Standards [Електронний ресурс]. – URL: https://www.researchgate.net/publication/384271593_An_Overview_of_Information_and_Cyber_Security_Standards (дата звернення: 17.12.2025).
2. National Institute of Standards and Technology (NIST). Computer Security Resource Center (CSRC). Cyber Security [Електронний ресурс]. / National Institute of Standards and Technology. – Glossary. – URL: https://csrc.nist.gov/glossary/term/cyber_security (дата звернення: 17.12.2025).
3. National Institute of Standards and Technology (NIST). The NIST Cybersecurity Framework (CSF) 2.0 (NIST CSWP 29) [Електронний ресурс]. – February 26, 2024. – DOI: 10.6028/NIST.CSWP.29. – URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf (дата звернення: 17.12.2025).
4. L. Liyanage, A. Gamagedara Arachchilage, G. Russello. SoK: Identifying Limitations and Bridging Gaps of Cybersecurity Capability Maturity Models (CCMMs) [Електронний ресурс] / Lasini Liyanage, Nalin Asanka Gamagedara Arachchilage, Giovanni Russello. – URL: https://ar5iv.labs.arxiv.org/html/2408.16140 (дата звернення: 17.12.2025).
5. Büyüközkan, Gülçin & Güler, Merve. Cybersecurity maturity model: Systematic literature review and a proposed model [Електронний ресурс]. – URL: https://www.researchgate.net/publication/390379156_Cybersecurity_maturity_model_Systematic_literature_review_and_a_proposed_model (дата звернення: 17.12.2025).
6. Zybin, Serhii & Bondarchuk, Oleg & Piroh, Alex & Suprun, Olha & Kyshakevych, Svitlana. Cybersecurity on Ukrainian Higher Education: Threats and protection measures [Електронний ресурс]. – URL:

https://www.researchgate.net/publication/393481181_Cybersecurity_on_Ukrainian_Higher_Education_Threats_and_protection_measures (дата звернення: 17.12.2025).
7. Dreis, Yurii & Korchenko, Oleksandr. Analysis of methods and models for assessing the consequences of the loss information with limited access, its value and aging [Електронний ресурс]. – URL: https://www.researchgate.net/publication/385910053_Analysis_of_methods_and_models_for_assessing_the_consequences_of_the_loss_information_with_limited_access_its_value_and_aging (дата звернення: 17.12.2025).
8. International Organization for Standardization (ISO). ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements [Електронний ресурс]. – 2022-10. – URL: https://www.iso.org/standard/27001 (дата звернення: 17.12.2025).
9. International Organization for Standardization (ISO). ISO/IEC 27005:2022 Information security, cybersecurity and privacy protection — Guidance on managing information security risks [Електронний ресурс]. – 2022-10. – URL: https://www.iso.org/standard/80585.html (дата звернення: 17.12.2025).
10. International Organization for Standardization (ISO). ISO/IEC 27000 (OBP) — Security techniques — Information security management systems — Overview and vocabulary. – [Електронний ресурс] / International Organization for Standardization – URL: https://www.iso.org/obp/ui/#iso:std:iso-iec:27000 (дата звернення: 17.12.2025).
11. Wani J., Mendoza A., Gray K. A Sociotechnical Approach to Bring-Your-Own-Device Security in Hospitals: Development and Pilot Testing of a Maturity Model Using Mixed Methods Action Research [Електронний ресурс] // JMIR Human Factors. – 13 Aug 2025. – DOI: 10.2196/71912. – URL: https://humanfactors.jmir.org/2025/1/e71912 (дата звернення: 17.12.2025).
12. Yevdokymov S. Neuro-symbolic models for ensuring cybersecurity in critical cyber-physical systems [Електронний ресурс] // JCPEE. – 2024. – Vol. 14, No. 1. – P. 42–50. – DOI: 10.23939/jcpee2024.01.042. – URL: https://science.lpnu.ua/jcpee/all-

volumes-and-issues/volume-14-number-1-2024/neuro-symbolic-models-ensuring-cybersecurity (дата звернення: 17.12.2025).

13. Björck, F., Henkel, M., Stirna, J., Zdravkovic, J. Chapter 10.1007/978-3-319-16486-1_31 [Електронний ресурс]. – URL: http://link.springer.com/chapter/10.1007/978-3-319-16486-1_31 (дата звернення: 17.12.2025).
14. Büyüközkan G., Güler M. Cybersecurity maturity model: Systematic literature review and a proposed model [Електронний ресурс] / G. Büyüközkan, M. Güler // Technological Forecasting and Social Change. – 2025. – Vol. 213. – DOI: 10.1016/j.techfore.2025.123996. – URL: https://www.sciencedirect.com/science/article/abs/pii/S0040162525000277 (дата звернення: 17.12.2025).
15. L. Liyanage, A. Gamagedara Arachchilage, G. Russello. SoK: Identifying Limitations and Bridging Gaps of Cybersecurity Capability Maturity Models (CCMMs) [Електронний ресурс] / Lasini Liyanage, Nalin Asanka Gamagedara Arachchilage, Giovanni Russello. – URL: https://arxiv.org/html/2408.16140v1 (дата звернення: 17.12.2025).
16. Daniel Jorge Ferreira, Nuno Mateus-Coelho, Henrique S. Mamede. Methodology for Predictive Cyber Security Risk Assessment [Електронний ресурс]. – URL: https://www.sciencedirect.com/science/article/pii/S1877050923004581?utm_source (дата звернення: 17.12.2025).
17. Hersyah M. H. et al. Fuzzyfortify: a multi-attribute risk assessment for multi-factor authentication and cloud container orchestration [Електронний ресурс] / M. H. Hersyah, M. D. Hossain, Y. Taenaka, Y. Kadobayashi // Frontiers in Computer Science. – 2025. – Vol. 7. – DOI: 10.3389/fcomp.2025.1557918. – URL: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1557918/full (дата звернення: 17.12.2025).
18. Cheimonidis, Pavlos & Rantos, Konstantinos. A Proactive and Time-Sensitive Cyber Risk Assessment Model Integrating Markov Chains and Bayesian Networks [Електронний ресурс]. – 2024. – URL: https://www.researchgate.net/publication/392248629_A_Proactive_and_Time-

Sensitive_Cyber_Risk_Assessment_Model_Integrating_Markov_Chains_and_Bayesian_Networks (дата звернення: 17.12.2025).

19. NIST. Guide for Conducting Risk Assessments [Електронний ресурс] / National Institute of Standards and Technology. – 2012. – (NIST Special Publication 800-30 Rev. 1). – URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf (дата звернення: 17.12.2025).
20. NIST. Security and Privacy Controls for Information Systems and Organizations [Електронний ресурс] / National Institute of Standards and Technology. – 2020. – (NIST Special Publication 800-53 Rev. 5). – URL: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf (дата звернення: 17.12.2025).
21. CIS. CIS Critical Security Controls Version 8 [Електронний ресурс] / Center for Internet Security. – URL: https://www.cisecurity.org/controls/v8 (дата звернення: 17.12.2025).
22. NIST. Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations [Електронний ресурс] / National Institute of Standards and Technology. – 2011. – (NIST Special Publication 800-137). – URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-137.pdf (дата звернення: 17.12.2025).
23. NIST. Guide to Computer Security Log Management [Електронний ресурс] / National Institute of Standards and Technology. – 2006. – (NIST Special Publication 800-92). – URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-92.pdf (дата звернення: 17.12.2025).
24. NIST. Computer Security Incident Handling Guide [Електронний ресурс] / National Institute of Standards and Technology. – 2024. – (NIST Special Publication 800-61 Rev. 3, Final). – URL: https://csrc.nist.gov/pubs/sp/800/61/r3/final (дата звернення: 17.12.2025).

25.NIST. Assessing Security and Privacy Controls in Information Systems and Organizations [Електронний ресурс] / National Institute of Standards and Technology. – 2022. – (NIST Special Publication 800-53A Rev. 5). – URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53Ar5.pdf (дата звернення: 17.12.2025).
26.IJASEIT. Article [Електронний ресурс] // International Journal on Advanced Science, Engineering and Information Technology. – URL: https://ijaseit.insightsociety.org/index.php/ijaseit/article/view/20234 (дата звернення: 17.12.2025).
27.NIST. Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations [Електронний ресурс] / National Institute of Standards and Technology. – 2022. – (NIST Special Publication 800-161 Rev. 1). – URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf (дата звернення: 17.12.2025).
28.European Union Agency for Cybersecurity (ENISA). Technical Implementation Guidance on Commission Implementing Regulation (EU) 2024/2690 of 17 October 2024. – June 2025, Version 1.0. – Luxembourg: Publications Office of the European Union, 2025. – DOI: 10.2824/2702548. – URL: https://www.enisa.europa.eu/sites/default/files/2025-06/ENISA_Technical_implementation_guidance_on_cybersecurity_risk_management_measures_version_1.0.pdf (дата звернення: 17.12.2025).
29.G. Büyüközkan, M. Güler.. Cybersecurity maturity model: Systematic literature review and a proposed model [Електронний ресурс] / G. Büyüközkan, M. Güler. – 2025. – URL: https://ideas.repec.org/a/eee/tefoso/v213y2025ics0040162525000277.html (дата звернення: 17.12.2025).
30.Kaplan S., Garrick B. J. On The Quantitative Definition of Risk [Електронний ресурс] / S. Kaplan, B. J. Garrick // Risk Analysis. – 1981. – Vol. 1, Issue 1. – DOI: 10.1111/j.1539-6924.1981.tb01350.x. – URL:

https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1539-6924.1981.tb01350.x (дата звернення: 17.12.2025).

31. Saaty T. L. The Analytic Hierarchy Process [Електронний ресурс] / T. L. Saaty. – McGraw-Hill, 1980. – URL: https://books.google.com.ua/books/about/The_Analytic_Hierarchy_Process.html?id=Xxi7AAAAIAAJ (дата звернення: 17.12.2025).
32. Zadeh L. A. Fuzzy sets [Електронний ресурс] / L. A. Zadeh // Information and Control. – 1965. – Vol. 8, Issue 3. – DOI: 10.1016/S0019-9958(65)90241-X. – URL: https://www.sciencedirect.com/science/article/pii/S001999586590241X (дата звернення: 17.12.2025).
33. Ana Paula Henriques de Gusmão, Lúcio Camara e Silva, Maisa Mendonça Silva, Thiago Poleto, Ana Paula Cabral Seixas Costa. Information security risk analysis model using fuzzy decision theory [Електронний ресурс]. – Elsevier. – URL: https://www.sciencedirect.com/science/article/abs/pii/S0268401215000900 (дата звернення: 17.12.2025).
34. Merve Güler, Gülçin Büyüközkan. Cybersecurity maturity assessment using an incomplete hesitant fuzzy AHP method and Bonferroni means operator. [Електронний ресурс]. – Elsevier, 2025. – URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417425008905(дата звернення: 17.12.2025).
35. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference [Електронний ресурс] / J. Pearl. – Morgan Kaufmann, 1988. – URL: https://www.sciencedirect.com/book/monograph/9780080514895/probabilistic-reasoning-in-intelligent-systems (дата звернення: 17.12.2025).
36. Cheimonidis, Pavlos & Rantos, Konstantinos. A Proactive and Time-Sensitive Cyber Risk Assessment Model Integrating Markov Chains and Bayesian Networks [Електронний ресурс]. – 2024. – URL: https://www.researchgate.net/publication/392248629_A_Proactive_and_Time-Sensitive_Cyber_Risk_Assessment_Model_Integrating_Markov_Chains_and_Bayesian_Networks (дата звернення: 17.12.2025).

37. Sun N., Zhang J. Data-Driven Cybersecurity Incident Prediction [Електронний ресурс] / N. Sun, J. Zhang. – URL: https://www.semanticscholar.org/paper/Data-Driven-Cybersecurity-Incident-Prediction%3A-A-Sun-Zhang/6f3131c90ccad03ef5992a749dd667aec831e42a(дата звернення: 17.12.2025).
38. Cue, Hayat & Bourlai, Thirimachos & Lupo, Mark. A Unified Assessment Methodology for Incident Forecasting with Cyber Threat Intelligence Integration [Електронний ресурс]. / PP. 1-1. 10.1109/ACCESS.2025.3596252. – URL: https://researchgate.net/publication/394376677_Proactive_Cyber_Resilience_A_Unified_Assessment_Methodology_for_Incident_Forecasting_with_Cyber_Threat_Intelligence_Integration(дата звернення: 17.12.2025).
39. Yevdokymov S. Neuro-symbolic models for ensuring cybersecurity in critical cyber-physical systems [Електронний ресурс] // JCPEE. – 2024. – Vol. 14, No. 1. – P. 42–50. – DOI: 10.23939/jcpee2024.01.042. – URL: https://science.lpnu.ua/jcpee/all-volumes-and-issues/volume-14-number-1-2024/neuro-symbolic-models-ensuring-cybersecurity (дата звернення: 17.12.2025).
40. Da-Yong Chang. Applications of the extent analysis method on fuzzy AHP [Електронний ресурс]. – URL: https://www.sciencedirect.com/science/article/pii/0377221795003002 (дата звернення: 17.12.2025).
41. Center for Internet Security (CIS). Implementation Groups (IG1) [Електронний ресурс]. – URL: https://www.cisecurity.org/controls/implementation-groups/ig1 (дата звернення: 17.12.2025).
42. J.R. Quinlan. Induction of Decision Trees. Machine Learning [Електронний ресурс]. – Springer. – DOI: 10.1023/A:1022643204877. – URL: https://link.springer.com/article/10.1023/A%3A1022643204877(дата звернення: 17.12.2025).
43. Breiman L. et al. Classification and Regression Trees [Електронний ресурс] / L. Breiman, J. Friedman, R. Olshen, C. Stone. – Routledge, 1984. – URL: https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-

regression-trees-leo-breiman-jerome-friedman-olshen-charles-stone(дата звернення: 17.12.2025).

44. Верховна Рада України. Закон України № 2694-XII [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/laws/main/2694-12 (дата звернення: 17.12.2025).
45. Верховна Рада України. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/rada/show/va042282-99 (дата звернення: 17.12.2025).
46. ДБН В.2.5-28-2018 [Електронний ресурс]. – URL: https://e-construction.gov.ua/laws_detail/3074958732556240833 (дата звернення: 17.12.2025).
47. Верховна Рада України. Наказ про Про затвердження Правил безпечної експлуатації електроустановок споживачів [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/laws/main/z0093-98 (дата звернення: 17.12.2025).
48. Верховна Рада України. Наказ Про затвердження Правил пожежної безпеки в Україні [Електронний ресурс]. – URL: http://zakon.rada.gov.ua/laws/show/z0252-15 (дата звернення: 17.12.2025).
49. Верховна Рада України. Наказ Про затвердження Державних санітарних норм та правил [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/laws/main/z0472-14 (дата звернення: 17.12.2025).
50. Верховна Рада України. Постанова Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/rada/show/v0007282-98 (дата звернення: 17.12.2025).
51. ДСТУ ISO 9241-5:2004. Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 5. Вимоги до компонування робочого місця та до робочої пози [Електронний ресурс]. – URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=53291 (дата звернення: 17.12.2025).
52. Кодекс законів про працю України [Електронний ресурс] : Закон України від 10.12.1971 № 322-VIII. – URL: https://zakon.rada.gov.ua/laws/main/322-08 (дата звернення: 17.12.2025).

53. Закон України "Про охорону навколишнього природного середовища" [Електронний ресурс] : від 25.06.1991 № 1264-XII. – URL: https://zakon.rada.gov.ua/go/1264-12 (дата звернення: 17.12.2025).
54. Верховна Рада України. Закон України № 2694-12 [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/laws/show/2694-12 (дата звернення: 17.12.2025).
55. Закон України "Про енергетичну ефективність" [Електронний ресурс] : від 21.10.2021 № 1818-IX. – URL: https://zakon.rada.gov.ua/laws/show/1818-20 (дата звернення: 17.12.2025).
56. Закон України "Про охорону навколишнього природного середовища" [Електронний ресурс] : від 25.06.1991 № 1264-XII (посилання на редакцію 2707-12). – URL: https://zakon.rada.gov.ua/go/2707-12 (дата звернення: 17.12.2025).
57. Закон України "Про управління відходами" [Електронний ресурс] : від 20.06.2022 № 2320-IX. – URL: https://zakon.rada.gov.ua/laws/show/2320-20 (дата звернення: 17.12.2025).
58. Верховна Рада України. Про затвердження Порядку ведення державного обліку відходів та подання звітності та Типової форми обліку відходів (редакція від 26.11.2024) [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/laws/show/z0090-25/ed20241126 (дата звернення: 17.12.2025).
59. Постанова Кабінету Міністрів України "Про затвердження Порядку створення та адміністрування інформаційної системи управління відходами" [Електронний ресурс] : від 05.12.2023 № 1279. – URL: https://zakon.rada.gov.ua/go/1279-2023-%D0%BF (дата звернення: 17.12.2025).
60. Правила пожежної безпеки в Україні [Електронний ресурс] : Наказ МВС України від 30.12.2014 № 1417 (зареєстровано в Мінюсті № z0252-15). – URL: https://zakon.rada.gov.ua/go/z0252-15(дата звернення: 17.12.2025).
61. Верховна Рада України. Наказ Про внесення змін до наказу № 203 від 21 грудня 2015 р. [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/go/v0221774-15 (дата звернення: 17.12.2025).

62. Верховна Рада України. Наказ Про прийняття та скасування національних стандартів [Електронний ресурс]. – URL: https://zakon.rada.gov.ua/go/v0104774-20 (дата звернення: 17.12.2025).
63. Directive 2012/19/EU of the European Parliament and of the Council of 4 July 2012 on waste electrical and electronic equipment (WEEE) [Електронний ресурс]. – URL: https://eur-lex.europa.eu/eli/dir/2012/19/oj/eng (дата звернення: 17.12.2025).
64. Regulation (EU) 2023/1542 of the European Parliament and of the Council of 12 July 2023 concerning batteries and waste batteries [Електронний ресурс]. – URL: https://eur-lex.europa.eu/eli/reg/2023/1542/oj/eng(дата звернення: 17.12.2025).

Додаток А

Фрагмент коду responses.py

```python
from fastapi import APIRouter, Depends, HTTPException
from sqlalchemy.orm import Session
from typing import List
from app.db.session import get_db
from app.models.response import ResponseSet, Response
from app.models.scoring import ScoringRun
from app.schemas.response import ResponseSetCreate, ResponseSetResponse,
ResponseSetWithResponses

router = APIRouter()

@router.post("", response_model=ResponseSetResponse)
async def create_response_set(request: ResponseSetCreate, db: Session =
Depends(get_db)):

    response_set = ResponseSet(
        questionnaire_version=request.questionnaire_version
    )
    db.add(response_set)
    db.flush()

    for response_data in request.responses:
        response = Response(
            response_set_id=response_set.id,
            criterion_id=response_data.criterion_id,
            level=response_data.level,
            evidence_status=response_data.evidence_status,
            applicable=response_data.applicable
        )
        db.add(response)

    db.commit()
    db.refresh(response_set)

    return response_set

@router.get("/latest", response_model=ResponseSetResponse)
async def get_latest_response_set(db: Session = Depends(get_db)):
    response_set =
db.query(ResponseSet).order_by(ResponseSet.created_at.desc()).first()
    if not response_set:
        raise HTTPException(status_code=404, detail="No response sets
found")
    return response_set

@router.get("", response_model=List[ResponseSetResponse])
async def get_all_response_sets(limit: int = 50, db: Session =
Depends(get_db)):
    response_sets =
db.query(ResponseSet).order_by(ResponseSet.created_at.desc()).limit(limit)
.all()
    result = []
    for rs in response_sets:
        scoring_run = db.query(ScoringRun).filter(
            ScoringRun.response_set_id == rs.id
        ).order_by(ScoringRun.created_at.desc()).first()
        result.append(ResponseSetResponse(
            id=rs.id,
            questionnaire_version=rs.questionnaire_version,
            created_at=rs.created_at,
            total_score=float(scoring_run.total_score) if scoring_run else
None
        ))
    return result

@router.get("/{response_set_id}", response_model=ResponseSetWithResponses)
async def get_response_set_with_responses(response_set_id: int, db:
Session = Depends(get_db)):
    response_set = db.query(ResponseSet).filter(ResponseSet.id ==
response_set_id).first()
    if not response_set:
        raise HTTPException(status_code=404, detail=f"ResponseSet
{response_set_id} not found")

    responses = db.query(Response).filter(Response.response_set_id ==
response_set_id).all()

    return ResponseSetWithResponses(
        id=response_set.id,
        questionnaire_version=response_set.questionnaire_version,
        created_at=response_set.created_at,
        responses=responses
    )
```

Додаток Б

Фрагмент коду /api/v1/scoring.py

```python
from fastapi import APIRouter, Depends, HTTPException
from sqlalchemy.orm import Session
from typing import List
from app.db.session import get_db
from app.models.response import ResponseSet, Response
from app.models.scoring import ScoringRun
from app.schemas.response import ResponseSetCreate, ResponseSetResponse,
ResponseSetWithResponses

router = APIRouter()

@router.post("", response_model=ResponseSetResponse)
async def create_response_set(request: ResponseSetCreate, db: Session =
Depends(get_db)):

    response_set = ResponseSet(
        questionnaire_version=request.questionnaire_version
    )
    db.add(response_set)
    db.flush()

    for response_data in request.responses:
        response = Response(
            response_set_id=response_set.id,
            criterion_id=response_data.criterion_id,
            level=response_data.level,
            evidence_status=response_data.evidence_status,
            applicable=response_data.applicable
        )
        db.add(response)

    db.commit()
    db.refresh(response_set)

    return response_set

@router.get("/latest", response_model=ResponseSetResponse)
async def get_latest_response_set(db: Session = Depends(get_db)):
    response_set =
db.query(ResponseSet).order_by(ResponseSet.created_at.desc()).first()
    if not response_set:
        raise HTTPException(status_code=404, detail="No response sets
found")
    return response_set

@router.get("", response_model=List[ResponseSetResponse])
async def get_all_response_sets(limit: int = 50, db: Session =
Depends(get_db)):
    response_sets =
db.query(ResponseSet).order_by(ResponseSet.created_at.desc()).limit(limit)
.all()
    result = []
    for rs in response_sets:
        scoring_run = db.query(ScoringRun).filter(
            ScoringRun.response_set_id == rs.id
        ).order_by(ScoringRun.created_at.desc()).first()
        result.append(ResponseSetResponse(
            id=rs.id,
            questionnaire_version=rs.questionnaire_version,
            created_at=rs.created_at,
            total_score=float(scoring_run.total_score) if scoring_run else
None
        ))
    return result

@router.get("/{response_set_id}", response_model=ResponseSetWithResponses)
async def get_response_set_with_responses(response_set_id: int, db:
Session = Depends(get_db)):
    response_set = db.query(ResponseSet).filter(ResponseSet.id ==
response_set_id).first()
    if not response_set:
        raise HTTPException(status_code=404, detail=f"ResponseSet
{response_set_id} not found")

    responses = db.query(Response).filter(Response.response_set_id ==
response_set_id).all()

    return ResponseSetWithResponses(
        id=response_set.id,
        questionnaire_version=response_set.questionnaire_version,
        created_at=response_set.created_at,
        responses=responses
    )
```

Додаток В

Фрагмент коду rules.py

```
from typing import List, Dict
from app.models.questionnaire import Domain, Criterion
from app.models.response import Response


def normalize_weights(domains: List[Domain], responses: List[Response]) ->
Dict[int, Dict[int, float]]:

    response_map = {r.criterion_id: r for r in responses}

    criteria_by_domain: Dict[int, List[Criterion]] = {}
    for domain in domains:
        criteria_by_domain[domain.id] = list(domain.criteria)

    normalized_weights: Dict[int, Dict[int, float]] = {}

    for domain in domains:
        domain_criteria = criteria_by_domain[domain.id]

        applicable_criteria = [
            c for c in domain_criteria
            if response_map.get(c.id,
Response(applicable=True)).applicable
        ]

        if not applicable_criteria:
            normalized_weights[domain.id] = {
                c.id: float(c.weight) for c in domain_criteria
            }
            continue

        total_applicable_weight = sum(float(c.weight) for c in
applicable_criteria)

        normalized_weights[domain.id] = {}
        for criterion in domain_criteria:
            if criterion.id in [c.id for c in applicable_criteria]:
                normalized_weight = float(criterion.weight) /
total_applicable_weight if total_applicable_weight > 0 else 0.0
                normalized_weights[domain.id][criterion.id] =
normalized_weight
            else:
                normalized_weights[domain.id][criterion.id] = 0.0

    return normalized_weights


def get_domain_weights(domains: List[Domain], responses: List[Response]) -
> Dict[int, float]:

    response_map = {r.criterion_id: r for r in responses}

    applicable_domain_weights: Dict[int, float] = {}
    total_weight = 0.0

    for domain in domains:
        has_applicable = any(
            response_map.get(c.id, Response(applicable=True)).applicable
            for c in domain.criteria
        )

        if has_applicable:
            applicable_domain_weights[domain.id] = float(domain.weight)
            total_weight += float(domain.weight)

    normalized_domain_weights: Dict[int, float] = {}
    for domain in domains:
        if domain.id in applicable_domain_weights and total_weight > 0:
            normalized_domain_weights[domain.id] =
applicable_domain_weights[domain.id] / total_weight
        else:
            normalized_domain_weights[domain.id] = 0.0

    return normalized_domain_weights
```

Додаток Г

Фрагмент коду core/scoring.py

```python
from typing import List, Dict
from app.models.questionnaire import Domain, Criterion
from app.models.response import Response
from app.core.rules import normalize_weights, get_domain_weights


EVIDENCE_FACTORS = {
    "FULL": 1.0,
    "PARTIAL": 0.85,
    "NONE": 0.70
}


LEVEL_TO_VALUE = {
    0: 0.0,   # L0
    1: 25.0,  # L1
    2: 50.0,  # L2
    3: 75.0,  # L3
    4: 100.0  # L4
}


def calculate_criterion_score(criterion: Criterion, response: Response,
normalized_weight: float) -> float:
   if not response.applicable:
        return 0.0

    level_value = LEVEL_TO_VALUE.get(response.level, 0.0)
    evidence_factor = EVIDENCE_FACTORS.get(response.evidence_status, 0.70)


    criterion_score = level_value * evidence_factor


    weighted_score = criterion_score * normalized_weight

    return weighted_score


def calculate_domain_score(domain: Domain, responses: List[Response],
normalized_criterion_weights: Dict[int, float]) -> float:
    domain_score = 0.0


    response_map = {r.criterion_id: r for r in responses}


    for criterion in domain.criteria:
        response = response_map.get(criterion.id)
        if not response:
            continue

        normalized_weight = normalized_criterion_weights.get(criterion.id,
0.0)
        criterion_score = calculate_criterion_score(criterion, response,
normalized_weight)
        domain_score += criterion_score

    return min(100.0, max(0.0, domain_score))


def calculate_total_score(
    domains: List[Domain],
    responses: List[Response],
    normalized_domain_weights: Dict[int, float],
    normalized_criterion_weights: Dict[int, Dict[int, float]]
) -> Dict[str, float]:

    domain_scores: Dict[str, float] = {}


    for domain in domains:
        domain_score = calculate_domain_score(
            domain,
            responses,
            normalized_criterion_weights.get(domain.id, {})
        )
        domain_scores[domain.domain_code] = domain_score


    total_score = 0.0
    for domain in domains:
        domain_weight = normalized_domain_weights.get(domain.id, 0.0)
        domain_score = domain_scores.get(domain.domain_code, 0.0)
        total_score += domain_score * domain_weight

    result = {"total": round(total_score, 2)}
    result.update({code: round(score, 2) for code, score in
domain_scores.items()})

    return result
```

Додаток Г

Фрагмент коду функції generate_recomendations

```python
def generate_recommendations(
    domains: List[Domain],
    responses: List[Response],
    domain_scores: Dict[str, float],
    normalized_domain_weights: Dict[int, float],
    normalized_criterion_weights: Dict[int, Dict[int, float]]
) -> List[Dict]:

    recommendations = []
    response_map = {r.criterion_id: r for r in responses}

    for domain in domains:
        domain_score = domain_scores.get(domain.domain_code, 0.0)
        domain_weight = normalized_domain_weights.get(domain.id, 0.0)

        if domain_score >= 100.0:
            continue

        criterion_weights = normalized_criterion_weights.get(domain.id,
{})

        for criterion in domain.criteria:
            response = response_map.get(criterion.id)
            if not response or not response.applicable:
                continue

            current_level = response.level
            target_level = determine_target_level(current_level,
response.evidence_status)

            if current_level >= target_level:
                continue

            criterion_weight = criterion_weights.get(criterion.id, 0.0)

            impact = calculate_criterion_impact(
                criterion,
                current_level,
                target_level,
                response.evidence_status,
                domain_weight,
                criterion_weight
            )

            effort = calculate_effort(current_level, target_level,
response.evidence_status)

            description = f"Підвищити рівень з L{current_level} до
L{target_level}"
            if response.evidence_status == 'NONE':
                description += ". Спочатку необхідно зібрати докази
відповідності."
            elif response.evidence_status == 'PARTIAL':
                description += ". Потрібно доповнити наявні докази."

            recommendation = {
                "criterion_id": criterion.id,
                "criterion_code": criterion.criterion_code,
                "criterion_name": criterion.name,
                "domain_code": domain.domain_code,
                "domain_name": domain.name,
                "current_level": current_level,
                "target_level": target_level,
                "impact": impact,
                "effort": effort,
                "description": description,
                "priority": determine_priority(impact, effort)
            }

            recommendations.append(recommendation)

    recommendations.sort(
        key=lambda x: (
            {'high': 0, 'medium': 1, 'low': 2}[x['priority']],
            -x['impact']
        )
    )

    return recommendations
```