

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

«До захисту допущено»
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
(підпис)

“ ____ ” червня 2026 р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Системне програмування та спеціалізовані комп'ютерні системи»
Спеціальність 123 «Комп'ютерна інженерія»**

на тему: Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR

Виконав: студент IV курсу, групи КВ-23
Домуці Дмитро Дмитрович

(підпис)

Керівник: доц.каф.СПСКС, к.т.н.
Потапова Катерина Романівна

(підпис)

Консультант з нормоконтролю: доц.каф.СПСКС, к.т.н.
Клятченко Ярослав Михайлович

(підпис)

Рецензент, доц.каф.ПМ, к.ф.-м.н.
Вовк Лілія Борисівна

(підпис)

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.
Студент

(підпис)

Київ – 2026 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Освітньо-професійна програма

**«Системне програмування та спеціалізовані
комп'ютерні системи»**

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

____ Віталій РОМАНКЕВИЧ
(підпис)

“ ____ ” червня 2026 р.

ЗАВДАННЯ

на дипломний проєкт студента

Домуці Дмитру Дмитровичу

1. Тема проєкту «Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR», керівник проєкту Потапова К.Р., кандидат технічних наук, доцент кафедри системного програмування і спеціалізованих комп'ютерних систем, затверджені наказом по університету від « ____ » травня 2026 р. № ____ -С

2. Термін подання студентом проєкту ____ червня 2026 р.

3. Вихідні дані до проєкту див. Технічне завдання

4. Зміст пояснювальної записки:

- 1) Аналіз існуючих рішень;
- 2) Математичне та алгоритмічне забезпечення системи;
- 3) Програмна реалізація інформаційно-аналітичного комплексу;
- 4) Тестування предиктивного ядра та верифікація функціональних модулів.

5. Перелік графічного матеріалу:

- 1) Інформаційно-аналітична система;

- 2) Реляційна база даних;
- 3) Алгоритм прогнозування кумулятивної втоми;
- 4) Алгоритм генерації адаптивних тренувальних протоколів.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я. М., доцент кафедри СПіСКС, к.т.н., доцент		

7. Дата видачі завдання «1» жовтня 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	15.11.2025	
2.	Аналіз існуючих рішень	01.12.2025	
3.	Розробка технічного завдання	18.12.2025	
4.	Підготовка матеріалів першого розділу дипломного проєкту	21.01.2026	
5.	Розробка програмного забезпечення	18.02.2026	
6.	Підготовка матеріалів другого розділу дипломного проєкту	14.03.2026	
7.	Підготовка матеріалів третього розділу дипломного проєкту	05.04.2026	
8.	Підготовка матеріалів четвертого розділу дипломного проєкту	18.04.2026	
9.	Підготовка графічної частини дипломного проєкту	04.05.2026	
10.	Оформлення документації дипломного проєкту	16.05.2026	
11.	Попередній огляд матеріалів диплому на кафедрі	28.05.2026	

Студент

(підпис)

Дмитро ДОМУЩІ

Керівник проєкту

(підпис)

Катерина ПОТАПОВА

АНОТАЦІЯ

Кваліфікаційна робота містить пояснювальну записку обсягом 67 с. (14 рис., 1 табл., 26 джерел, 5 додатків).

Об'єктом розробки є Basketball AI Injury Prevention System (BAIPS) — мікросервісний програмний комплекс адаптивного керування тренувальним навантаженням атлетів на основі рекурентної нейромережі LSTM та концепції ACWR.

У роботі розроблено:

- нелінійну мультиплікативну модель деградації амортизаційного ресурсу спортивного взуття з урахуванням маси атлета, типу ігрового покриття та призначення взуття;
- чотиришарову LSTM-архітектуру BasketballFatigueNet для предиктивного оцінювання ризику перевтоми на 7-денних вікнах телеметрії;
- метод гібридного постобчислювального коригування прогнозів через показник варіабельності серцевого ритму (HRV, rMSSD);
- стохастичний модуль Drill Engine для генерування адаптивних тренувальних протоколів за п'ятьма ігровими позиціями.

Серверну частину реалізовано на FastAPI з інтеграцією TensorFlow; клієнтський застосунок розроблено за специфікацією PWA на Vue.js 3 з підтримкою автономного кешування. СУБД — PostgreSQL. Модель верифіковано на синтетичному датасеті (50 000 записів, MAE = 0,0227) та впроваджено в навчально-тренувальний процес гімназії «Кларіс Вербіс».

Ключові слова: превенція травматизму, LSTM, ACWR, варіабельність серцевого ритму, FastAPI, Vue.js, PWA, PostgreSQL, спортивна аналітика, баскетбол.

ABSTRACT

The qualification work comprises an explanatory note (67 p., 14 fig., 1 table, 26 references, 5 appendices).

The object of development is the Basketball AI Injury Prevention System (BAIPS) — a microservice software complex for adaptive management of athletes' training loads, based on an LSTM recurrent neural network and the ACWR concept.

The following components were developed:

- a nonlinear multiplicative footwear degradation model accounting for athlete body mass, surface type, and shoe category;
- a four-layer BasketballFatigueNet LSTM architecture for 7-day telemetry-window fatigue risk prediction;
- a hybrid post-computation correction method using heart rate variability (HRV, rMSSD);
- a stochastic Drill Engine module for position-specific training protocol generation across five basketball positions.

The backend is built on FastAPI with TensorFlow inference integration; the frontend is a Vue.js 3 Progressive Web App with Service Worker offline caching. Data persistence uses PostgreSQL. The model was validated on a synthetic dataset (50,000 records, MAE = 0.0227) and deployed at Claris Verbis gymnasium.

Keywords: injury prevention, LSTM, ACWR, heart rate variability, FastAPI, Vue.js, PWA, PostgreSQL, sports analytics, basketball.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.002 ТЗ	Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR Технічне завдання	4		
	A4	ІАЛЦ.045420.003 ТП	Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR Відомість технічного проєкту	2		
	A4	ІАЛЦ.045420.004 ПЗ	Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR Пояснювальна записка	67		
			ІАЛЦ.045420.001 ОА			
Змін.	Арк.	№ докум.	Підпис	Дата		
Розробив	Домуші Д.Д.				Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR Опис альбому	
Перевірив	Потапова К.Р.					
Консулт.						
Н. контроль	Клятченко Я.М.					
Зав. каф.	Романкевич В. О.					
					Літ. Аркуш Аркушів	
					1 2	
					КПІ ім. Ігоря Сікорського, ФПСМ KB-23	

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.005 Д1	Інформаційно-аналітична система.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.006 Д2	Реляційна база даних.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.007 Д3	Алгоритм прогнозування кумулятивної втоми.	1		
			Схема алгоритму			
	A4	ІАЛЦ.045420.008 Д4	Алгоритм генерації адаптивних протоколів.	1		
			Схема алгоритму			
		Диск CD-ROM	Текст пояснювальної записки.	1		
			Графічний матеріал			

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.001 ОА

Арк.
2

ЗМІСТ

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОБОТИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1 Вимоги до програмного продукту, що розробляється	3
5.2 Рекомендовані вимоги до апаратного забезпечення	3
5.3 Вимоги до мінімального програмного забезпечення	3
6. ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.045420.002 ТЗ		
Змін	Арк.	№ докум.	Підпис	Дата			
Розробив					Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR <i>Технічне завдання.</i>	Літ.	Аркуш
Перевірив							
						1	4
Н. контроль	Клятченко Я.М.					КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23	
Затвердив	Романкевич В.О.						

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Назва розробки: «Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR».

Галузь застосування: спортивна медицина, біомеханіка, професійний та аматорський спорт (зокрема баскетбол), автоматизація індивідуального планування та моніторингу тренувального процесу.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Метою дипломного проєкту є розроблення інформаційно-аналітичного комплексу для предиктивного оцінювання кумулятивної втоми спортсменів і мінімізації ризиків травматизму на основі рекурентної нейромережі LSTM та концепції ACWR.

4. ДЖЕРЕЛА РОБОТИ

Джерелом інформації є науково-технічна література з методів глибокого машинного навчання, медичні дослідження у галузі спортивної фізіології (зокрема парадокс тренувального навантаження та концепція ACWR), технічна документація до фреймворків FastAPI та Vue.js, документація СУБД PostgreSQL, публікації у фахових періодичних виданнях та електронні статті у мережі Інтернет.

					ІАЛЦ.045420.002 ТЗ	Арк.
						2
Зм	Лист	№ докум.	Підп.	Дата		

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- можливість моніторингу фізіологічного стану та тренувального навантаження користувача;
- можливість прогнозування ризику травм на основі біометричних даних (LSTM);
- можливість автоматичної генерації адаптивних тренувальних планів;
- можливість динамічного розрахунку зносу спортивного екіпірування;
- можливість автономної (офлайн) роботи клієнтського додатка.

5.2 Рекомендовані вимоги до апаратного забезпечення

- процесор: архітектура x86-64 (наприклад, Intel Core i5, AMD Ryzen 5 або аналогічні);
- оперативна пам'ять: не менше 4 ГБ (рекомендовано 8 ГБ для роботи нейромережі);
- вільний простір на диску: не менше 10 ГБ;
- наявність доступу до мережі Internet (GPRS, EDGE, 3G, 4G);

5.3 Вимоги до програмного та апаратного забезпечення користувача

- Операційна система: iOS, Android, Windows, macOS або Linux;
- Програмне забезпечення: сучасний веббраузер (Google Chrome, Safari, Mozilla Firefox) із підтримкою технології Service Worker;

Підключення до мережі Інтернет (для первинного завантаження та подальшої синхронізації даних із сервером).

					ІАЛЦ.045420.002 ТЗ	Арк.
						3
Зм	Лист	№ докум.	Підп.	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту
1.	Вивчення літератури за тематикою проєкту	15.11.2025
2.	Аналіз існуючих рішень	01.12.2025
3.	Розроблення та узгодження технічного завдання	18.12.2025
4.	Підготовка матеріалів першого розділу дипломного проєкту	21.01.2026
5.	Розробка програмного забезпечення	18.02.2026
6.	Підготовка матеріалів другого розділу дипломного проєкту	14.03.2026
7.	Підготовка матеріалів третього розділу дипломного проєкту	05.04.2026
8.	Підготовка графічної частини дипломного проєкту	18.04.2026
9.	Підготовка матеріалів четвертого розділу дипломного проєкту	04.05.2026
10.	Оформлення документації дипломного проєкту	16.05.2026
11.	Попередній огляд матеріалів диплому на кафедрі	28.05.2026

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.004 ПЗ	Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR	67		
			Пояснювальна записка			
	A4	ІАЛЦ.045420.005 Д1	Інформаційно-аналітична система.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.006 Д2	Реляційна база даних.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.007 Д3	Алгоритм прогнозування кумулятивної втоми.	1		
			Схема алгоритму			
	A4	ІАЛЦ.045420.008 Д4	Алгоритм генерації адаптивних тренувальних протоколів.	1		
			Схема алгоритму			
			ІАЛЦ.045420.003 ТП			
Змін.	Арк.	№ докум.	Підпис	Дата		
Розробив	Домуші Д.Д.				Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR Відомість технічного проєкту Літ. Аркуш Аркушів 1 2	
Перевірів	Потапова К.Р.					
Консульт.						
Н. контроль	Клятченко Я.М.					
Зав. каф.	Романкевич В. О.					
					КПІ ім. Ігоря Сікорського, ФПСПМ KB-23	

[illegible]

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Інформаційно-аналітична система превенції спортивних травм на
основі рекурентної нейромережі LSTM та концепції ACWR»

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	4
ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	8
1.1 Проблематика спортивного травматизму та концепція ACWR.....	8
1.1.1 Аналіз компресійних та кінематичних навантажень у професійному баскетболі..	8
1.1.2 Математичні засади моделі співвідношення гострого та хронічного навантажень (ACWR)	10
1.2 Аналіз існуючих інформаційних систем моніторингу тренувальних навантажень	12
1.2.1 Платформа предиктивної аналітики Zone7	12
1.2.2 Платформа носитної телеметрії та GPS-трекінгу Catapult Sports	13
1.2.3 Комплексна система командного моніторингу Kitman Labs	14
1.3 Аналіз підходів до використання глибокого машинного навчання у спортивній медицині.....	15
1.3.1 Класичні нейромережі прямого поширення (FFNN) та їх обмеження при обробці послідовностей	15
1.3.2 Одномірні згорткові нейронні мережі (1D-CNN) та їх специфіка в аналізі телеметрії	16
1.3.3 Рекурентні архітектури (RNN) та переваги шарів LSTM для моделювання кумулятивного стресу	18
1.4 Обґрунтування теми дипломного проєкту та постановка задачі.....	20
Висновки до розділу 1	25
РОЗДІЛ 2. МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	26
2.1 Математична модель предиктивної оцінки деградації спортивного екіпірування	26
2.2 Архітектура та математичний апарат рекурентної нейромережі LSTM	29

					ІАЛЦ.045420.004 ПЗ		
Зм.	Арк.	№ докум.	Підп.	Дата			
Розроб.		Домуші Д.Д.			Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR Пояснювальна записка.	Літ.	Аркуш
Перевір.		Потапова К.Р.					1
							64
Н. контр.		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПСІМ, КВ-23	
Затв.		Романкевич В.О.					

2.3	Метод гібридного постобчислювального коригування прогнозів на основі HRV	32
2.4	Стохастичний алгоритм генерації тренувальних планів (метод Фішера-Єйтса)	35
	Висновки до розділу 2	38
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОГО КОМПЛЕКСУ		39
3.1	Вибір та обґрунтування інструментальних засобів розробки (FastAPI, TensorFlow)	39
3.2	Проектування логічної структури реляційної бази даних PostgreSQL ..	42
3.3	Розробка мікросервісної бекенд-архітектури та інтеграція ІІІ-ядра.....	44
3.4	Реалізація кросплатформного клієнтського інтерфейсу за технологією PWA	46
	Висновки до розділу 3	51
РОЗДІЛ 4. ТЕСТУВАННЯ ПРЕДИКТИВНОГО ЯДРА ТА ВЕРИФІКАЦІЯ ФУНКЦІОНАЛЬНИХ МОДУЛІВ СИСТЕМИ		53
4.1	Методика генерації синтетичних наборів даних та їх нормалізація	53
4.2	Навчання та валідація предиктивної моделі глибокого навчання	56
4.3	Аналіз прогностичної здатності та оцінка похибки алгоритму.....	58
4.4	Верифікація REST API та функціональних модулів системи	61
	Висновки до розділу 4	61
ВИСНОВКИ.....		63
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ		65

ДОДАТКИ

Додаток А. Копії графічного матеріалу

- ІАЛЦ.045420.005 Д1.
- ІАЛЦ.045420.006 Д2.
- ІАЛЦ.045420.007 Д3.

					ІАЛЦ.045420.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		2

- ІАЛЦ.045420.008 Д4.

Додаток Б. Фрагменти програмного коду

Додаток В. Презентація дипломного проєкту

Додаток Г. Копія тез конференції

Додаток Д. Копія довідки про впровадження

					ІАЛЦ.045420.004 ПЗ	Арк.
						3
Зм	Лист	№ докум.	Підп.	Дата		

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

СУБД — система управління базами даних (Database Management System).

ПЗ — пояснювальна записка.

ШІ — штучний інтелект.

1D-CNN (One-Dimensional Convolutional Neural Network) — одновимірна згорткова нейронна мережа.

ACWR (Acute:Chronic Workload Ratio) — коефіцієнт співвідношення гострого та хронічного тренувального навантаження.

API (Application Programming Interface) — інтерфейс прикладного програмування.

ASGI (Asynchronous Server Gateway Interface) — асинхронний інтерфейс серверного шлюзу.

BAIPS (Basketball AI Injury Prevention System) — Basketball AI Injury Prevention System; повна англомова назва розробленого інформаційно-аналітичного комплексу превенції спортивних травм.

BPTT (Backpropagation Through Time) — алгоритм зворотного поширення похибки крізь часові кроки.

EVA (Ethylene Vinyl Acetate) — етиленвінілацетат; пінний матеріал амортизаційної міжпідшви спортивного взуття.

FFNN (Feed-Forward Neural Network) — нейронна мережа прямого поширення.

GPS (Global Positioning System) — глобальна система супутникової навігації.

HRV (Heart Rate Variability) — варіабельність серцевого ритму.

HTTP / HTTPS (Hypertext Transfer Protocol / Secure) — протокол передачі гіпертексту / захищена версія.

IMU (Inertial Measurement Unit) — інерційний вимірювальний пристрій.

JWT (JSON Web Token) — стандарт захищеної передачі даних між сторонами у форматі JSON.

					ІАЛЦ.045420.004 ПЗ	Арк.
						4
Зм	Лист	№ докум.	Підп.	Дата		

LSTM (Long Short-Term Memory) — архітектура рекурентної нейронної мережі з довгою короткочасною пам'яттю.

MAE (Mean Absolute Error) — середня абсолютна похибка.

MSE (Mean Squared Error) — середньоквадратична похибка.

NBA (National Basketball Association) — Національна баскетбольна асоціація США.

ORM (Object-Relational Mapping) — об'єктно-реляційне відображення.

PWA (Progressive Web Application) — прогресивний вебзастосунок.

REST (Representational State Transfer) — архітектурний стиль взаємодії компонентів розподіленої системи.

RNN (Recurrent Neural Network) — рекурентна нейронна мережа.

RPE (Rating of Perceived Exertion) — суб'єктивна оцінка сприйнятого фізичного навантаження (шкала Борга).

sRPE (Session Rating of Perceived Exertion) — метод оцінки сесійного тренувального навантаження на основі RPE та тривалості заняття.

UWB (Ultra-Wideband) — надширокосмугова радіотехнологія для локального позиціонування в приміщеннях.

W3C (World Wide Web Consortium) — міжнародна організація зі стандартизації вебтехнологій.

					ІАЛЦ.045420.004 ПЗ	Арк.
						5
Зм	Лист	№ докум.	Підп.	Дата		

ВСТУП

Статистика травматизму в NBA свідчить, що понад 65% зареєстрованих пошкоджень є безконтактними, тобто спричиненими не фізичними зіткненнями, а хронічним перевищенням адаптаційного ресурсу тканин унаслідок некерованого накопичення тренувального навантаження [6]. Цей факт ставить задачу автоматизованого предиктивного моніторингу фізіологічного стану атлетів в основу сучасної спортивної інженерії. Обсяги біометричної телеметрії, якими оперують професійні клуби та індивідуальні спортсмени, безперервно зростають і зумовлюють потребу в інтелектуальних засобах опрацювання інформації.

Проте традиційні підходи, що ґрунтуються виключно на суб'єктивному досвіді тренерського штабу або дорогих апаратних комплексах, не здатні забезпечити гнучкого та точного аналізу через нелінійний, накопичувальний характер фізіологічних реакцій організму. Класичні математичні моделі дозволяють розраховувати базові коефіцієнти поточних навантажень, однак цього недостатньо для врахування кумулятивного ефекту втоми в межах тривалих мікроциклів. Предиктивне оцінювання ризику травматизму потребує обробки багатовимірних часових рядів, де фізіологічні декомпенсації мають прихований(латентний) характер. Для вирішення цієї задачі найбільш адекватним інструментом є рекурентні нейронні мережі з довгою короткочасною пам'яттю, здатні акумулювати передісторію стану спортсмена впродовж тривалих часових проміжків, долаючи ефект згасання градієнта, притаманний класичним рекурентним архітектурам.

Цільове середовище використання — спортивний зал з нестабільним мережевим покриттям, яке вимагає двох архітектурних рішень: ASGI-сервер із неблокуючою обробкою запитів під час інференсу нейромережі та PWA-клієнт із Service Worker для автономного функціонування інтерфейсу.

					ІАЛЦ.045420.004 ПЗ	Арк.
						6
Зм	Лист	№ докум.	Підп.	Дата		

Метою дипломного проєкту є розроблення мікросервісного програмного комплексу адаптивного керування тренувальним навантаженням із застосуванням методів глибокого навчання, що автоматизує процес превенції спортивного травматизму та оптимізує планування тренувальних занять. Розроблена система реалізує предиктивний аналіз та прескриптивне керування на трьох рівнях: на рівні обчислювального ядра серверної частини — шляхом генерування прогнозів навченою нейромережею LSTM; на рівні гібридної постобчислювальної адаптації результатів III за допомогою об'єктивних біометричних коефіцієнтів варіабельності серцевого ритму; та в межах кросплатформного клієнтського PWA-інтерфейсу з підтримкою автономного кешування даних. У цьому дипломному проєкті практично верифіковано гіпотезу: відкрита LSTM-архітектура на стандартному хмарному сервері здатна замінити дорогі пропрієтарні платформи у задачах превенції безконтактного травматизму.

Апробація результатів роботи. Основні результати та ключові технічні рішення дипломного проєкту оприлюднено на VII Міжнародній науково-практичній конференції «European science and innovation congress» (1–3 червня 2026 р., м. Барселона, Іспанія) [26].

					ІАЛЦ.045420.004 ПЗ	Арк.
						7
Зм	Лист	№ докум.	Підп.	Дата		

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Перший розділ вирішує дві задачі: виявити технічний та економічний розрив між наявними комерційними системами і встановити, чому LSTM є архітектурно оптимальним вибором для моделювання кумулятивного стресу на часових рядах.

1.1 Проблематика спортивного травматизму та концепція ACWR

Дослідження превентивних методів у комп'ютерній інженерії спорту вимагає чіткої формалізації фізичних та біомеханічних процесів, що відбуваються під час навантаження на опорно-руховий апарат атлета. Без розуміння кінематичних факторів та математичних закономірностей накопичення втоми неможливо побудувати адекватну предиктивну модель машинного навчання, здатну трансформувати сирі метрики у точні індекси ризику.

1.1.1 Аналіз компресійних та кінематичних навантажень у професійному баскетболі

Баскетбол характеризується надзвичайно високими вимогами до фізичних і фізіологічних кондицій гравців [22]. Ігровий процес побудований на безперервному чергуванні вибухових прискорень, миттєвих гальмувань, різких змін напрямку руху та серій стрибків із жорстким приземленням. Усі ці рухові акти генерують значні кінематичні навантаження, що передаються крізь стопу на гомілковостопні та колінні суглоби, а також на хребетний стовп атлета. Під час приземлення після стрибка сила вертикального удару, що діє на суглобові структури та міжхребцеві диски, може у 4-6 разів перевищувати масу тіла спортсмена, формуючи критичні зони механічного напруження.

					ІАЛЦ.045420.004 ПЗ	Арк.
						8
Зм	Лист	№ докум.	Підп.	Дата		

Згідно з профільною медико-статистичною аналітикою професійних ліг, безконтактні травми нижніх кінцівок (зокрема, розриви передньої хрестоподібної зв'язки, пошкодження менісків і стресові переломи плеснових кісток) становлять понад 65% від загальної кількості пошкоджень [6]. Фундаментальною причиною виникнення таких патологій є феномен «Overuse Injury» — травми від надмірного кумулятивного навантаження на біологічні тканини. Механізм деградації структур має виражений накопичувальний характер: кожне інтенсивне тренування чи матч спричинює мікроскопічні ушкодження у м'язових волокнах, сухожиллях та кісткових структурах. За умов адекватного відпочинку організм активує процеси регенерації, що призводить до суперкомпенсації. Проте, якщо частота, тривалість та інтенсивність тренувальних сесій перевищують адаптаційний потенціал спортсмена, мікротравми починають накопичуватися швидше, ніж відбувається їх відновлення. Це призводить до хронічного запалення, зниження еластичності зв'язок та утворення мікротріщин у кісткових структурах. На певному етапі відбувається деградація біомеханічної міцності тканини, і звичайний вибуховий рух (наприклад, різкий кроссовер чи стрибок) завершується механічним руйнуванням (травмою).

Критичним аспектом, що ігнорується більшістю загальних фітнес-систем (зокрема WHOOP або Oura), є позиційна специфіка зовнішнього навантаження. Сучасні комерційні пристрої зосереджені переважно на кардіоваскулярних показниках (загальний пульс, варіабельність), проте вони не диференціюють латеральне вибухове навантаження розігруючого захисника (Point Guard) від вертикальної компресії та постійної силової боротьби центрального гравця (Center). У гравців периметра переважають травми гомілковостопного суглоба через різкі зміни вектора руху, тоді як «бігмени» зазнають хронічних компресійних пошкоджень колін та хребта через систематичні стрибки під кошиком.

					ІАЛЦ.045420.004 ПЗ	Арк.
						9
Зм	Лист	№ докум.	Підп.	Дата		

Звідси конструктивна вимога: система повинна диференціювати навантаження за ігровою позицією, а не зводити його до єдиного кардіоваскулярного індексу.

1.1.2 Математичні засади моделі співвідношення гострого та хронічного навантажень (ACWR)

Для кількісного оцінювання динаміки втоми використовується ACWR (Acute:Chronic Workload Ratio) — концепція Т. Габбетта [1], яка встановлює, що не абсолютне навантаження, а різкість його зміни є основним предиктором безконтактної травми. Математичне підтвердження цього зв'язку [1, 16, 17] дозволяє сформулювати кількісні порогові значення ризику.

Коефіцієнт ACWR кількісно виражає баланс між поточною втомою та рівнем хронічної підготовленості. Для його обчислення без інвазивного обладнання застосовано адаптований метод Session-RPE [4]: суб'єктивна оцінка важкості тренування за 10-бальною шкалою Борга. З метою інтеграції біомеханічної специфіки баскетболу та диференціації рухової активності внутрішнє добове навантаження L для конкретного дня обчислюється за формулою:

$$L = RPE \times t \times k_{pos} \quad (1.1)$$

де RPE — суб'єктивна оцінка напруження (від 1 до 10); t — фактична тривалість тренувальної сесії, хв; k_{pos} — емпіричний ваговий коефіцієнт ігрового амплуа, що кількісно відображає різницю в енерговитратах та механічному навантаженні на суглобовий апарат між гравцями периметра та трисекундної зони.

Розрахунок самого індексу ACWR базується на моделі кумулятивного ковзного середнього (Coupled Rolling Average). Він визначається як відношення середнього акумульованого навантаження за поточний мікроцикл (7 діб) до середнього базового навантаження за попередній мезоцикл (28 діб):

					ІАЛЦ.045420.004 ПЗ	Арк.
						10
Зм	Лист	№ докум.	Підп.	Дата		

$$ACWR = \frac{\frac{1}{7} \sum_{i=1}^7 L_i}{\frac{1}{28} \sum_{j=1}^{28} L_j} \quad (1.2)$$

де L_i та L_j — розраховані показники внутрішнього навантаження за відповідну добу. Чисельник виступає математичним вираженням гострої втоми за 7 діб, а знаменник відображає усереднену адаптаційну базу спортсмена за 28 діб [1].

Інтерпретація отриманого значення ACWR здійснюється за концепцією «зон ризику» Т. Габбетта [1]. значення індексу в діапазоні $[0,8; 1,3]$ формує оптимальну зону («Sweet Spot»), за якої ймовірність отримання безконтактної травми м'яких тканин є мінімальною (менше 10%) [1, 16]. Якщо гостре навантаження різко зростає і показник перевищує критичну межу 1,5, настає «зона підвищеної небезпеки» (Danger Zone), де ризик травмування зростає за нелінійним законом.

Графічну інтерпретацію зон ризику за концепцією Габбетта [1] наведено на рисунку 1.1.

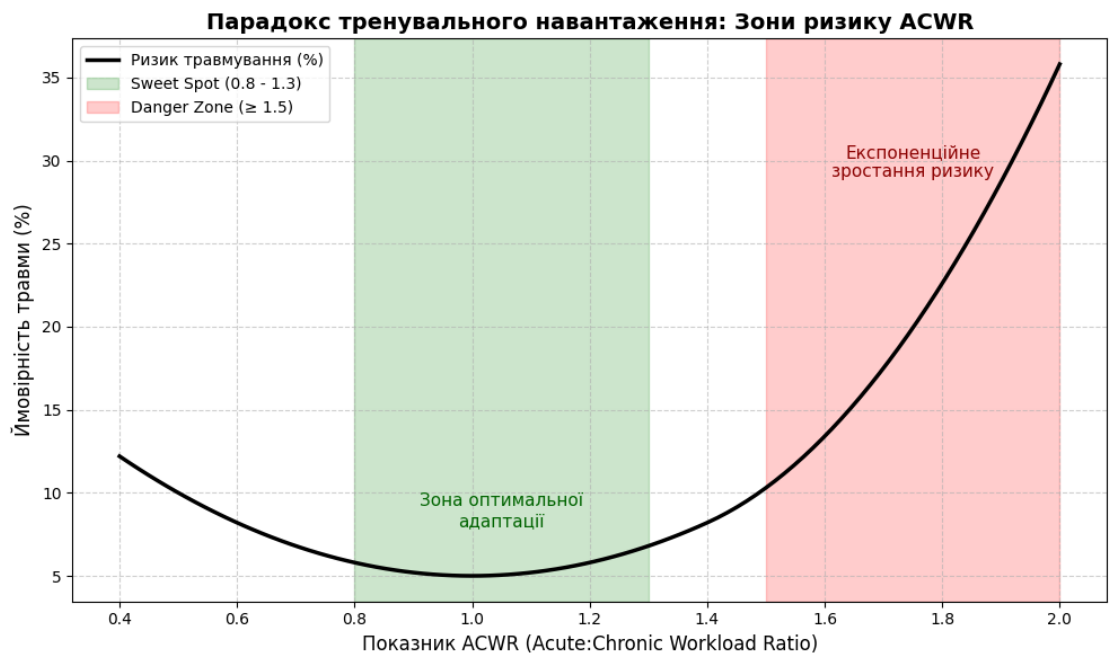


Рисунок 1.1 – Зони ризику безконтактного травматизму залежно від значення коефіцієнта ACWR

Додатково алгоритмічна логіка предиктивного ядра враховує емпіричне «правило 10%» [1, 24], згідно з яким приріст тижневого навантаження понад 15% зумовлює зростання ймовірності травмування від 21% до 49% [1]. Визначені порогові значення та виявлені закономірності є базовими тригерами для функціонування рекомендаційного модуля розроблюваної системи під час динамічної маршрутизації тренувальних планів та протоколів відновлення.

1.2 Аналіз існуючих інформаційних систем моніторингу тренувальних навантажень

Ефективна превенція травматизму потребує автоматизованого збору та консолідації значної кількості різномірних параметрів. У сучасній індустрії HealthTech та Sports Analytics функціонує ряд розгорнутих комерційних рішень, що намагаються розв'язати проблему контролю стану атлетів. Для визначення оптимального архітектурного напрямку власної розробки варто розглянути архітектуру, логіку роботи та ключові обмеження провідних світових систем моніторингу.

1.2.1 Платформа предиктивної аналітики Zone7

Zone7 — хмарна аналітична платформа, що агрегує дані медичних карток, лабораторних аналізів і комерційних смарт-годинників та генерує попередження про підвищений ризик травми [13]. Ключове обмеження — повністю закритий пропрієтарний алгоритм без можливості аудиту або адаптації. Обчислювальне ядро платформи базується на закритих пропрієтарних алгоритмах штучного інтелекту, які будують складні багатовимірні кореляційні матриці, виявляють аномалії у фізіологічних трендах та автоматично надсилають сповіщення тренерському штабу про критичний рівень втоми конкретного гравця.

					ІАЛЦ.045420.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		12

Однак Zone7 — закрыта пропрієтарна платформа. Алгоритм недоступний для аудиту, а вектор ознак фіксований: налаштування під позиційну специфіку баскетболу (наприклад, диференціація латерального навантаження Point Guard і вертикальної компресії Center) технічно неможливе. Комерційна модель передбачає дорогу B2B-ліцензію та обов'язкову участь сертифікованих аналітиків, що робить Zone7 апріорі недоступною для аматорського сегмента. Окрім того, хмарна архітектура Zone7 не підтримує локального рекомендаційного планування та повністю ігнорує зовнішні механічні фактори ризику травматизму, зокрема ступінь деградації амортизаційного ресурсу спортивного взуття.

1.2.2 Платформа носитної телеметрії та GPS-трекінгу Catapult Sports

Спортивно-аналітичний комплекс Catapult Sports є індустріальним стандартом у галузі реєстрації зовнішніх фізичних навантажень у командних видах спорту [14]. Архітектура цього рішення ґрунтується на жорсткій інтеграції спеціалізованого апаратного забезпечення та клієнтських програмних модулів. Під час тренувальних сесій спортсмени екіпіруються нагрудними пасами, до яких інтегровано високочастотні інерційні вимірювальні пристрої (IMU) — триосьові акселерометри, гіроскопи та магнітометри, а також приймачі супутникової навігації (GPS/GLONASS). Серверне програмне забезпечення Catapult у реальному часі опрацьовує потік телеметрії та розраховує комплексні механічні індекси, зокрема показник PlayerLoad (сумарний вектор прискорення у тривимірному просторі) та метаболічну потужність атлета.

Незважаючи на високу точність фіксації кінематичних параметрів, комплекс Catapult Sports має низку критичних архітектурних та інфраструктурних обмежень:

- **апаратна залежність:** програмний продукт є монолітним доповненням до фірмового апаратного забезпечення, що унеможливорює

					ІАЛЦ.045420.004 ПЗ	Арк.
						13
Зм	Лист	№ докум.	Підп.	Дата		

використання платформи без придбання дороговартісних нагрудних датчиків;

- **інвазивність:** необхідність постійного механічного носіння апаратних трекерів під час контактної боротьби на майданчику створює дискомфорт та підвищує ризик отримання механічних ушкоджень від самого пристрою;
- **екранування сигналу в приміщеннях:** у критичних для баскетболу умовах закритих спортивних залів супутниковий GPS-сигнал повністю блокується, що змушує розгортати складні та дорогі системи локального позиціонування (LPS) на базі радіомаяків Ultra-Wideband (UWB), геометрично збільшуючи інфраструктурний бюджет;
- **відсутність прескриптивного виводу:** програмні інструменти Catapult функціонують виключно в межах дескриптивної аналітики, констатуючи факт виконаного навантаження, але не маючи вбудованого III-модуля для автоматичного генерування адаптивних коригуючих планів відновлення.

1.2.3 Комплексна система командного моніторингу Kitman Labs

Інформаційна система Kitman Labs розроблена як централізований аналітичний хаб для оптимізації операційних процесів спортивних організацій, ведення медичного документообігу та збору експертних оцінок [15]. Програмний комплекс надає кросплатформні інтерфейси для щоденного заповнення спортсменами суб'єктивних анкет (Wellness Questionnaire) та реєстрації клінічних звітів фізіотерапевтів. Головною метою системи є об'єднання розрізненої інформації про травми та візуалізація загальних графіків готовності команди для управлінського персоналу клубу.

Аналіз логічної структури Kitman Labs свідчить, що аналітичний рушій платформи ґрунтується переважно на класичній лінійній статистиці,

					ІАЛЦ.045420.004 ПЗ	Арк.
						14
Зм	Лист	№ докум.	Підп.	Дата		

кореляційному аналізу та простих регресійних моделях. Система не має засобів повноцінної обробки багатовимірних нелінійних часових послідовностей біометрії за допомогою методів глибокого машинного навчання, що знижує точність довгострокового предиктивного аналізу. Архітектура платформи має монолітну хмарну прив'язку та не містить інструментів локального кешування або Service Worker, що унеможлиблює автономне функціонування клієнтських інтерфейсів на спортивних майданчиках за відсутності стабільного мережевого покриття. У системі також відсутній математичний апарат для нівелювання проблеми «холодного старту» при реєстрації нових користувачів, що призводить до алгоритмічного упередження (bias) на початкових етапах експлуатації.

1.3 Аналіз підходів до використання глибокого машинного навчання у спортивній медицині

Для виявлення патернів накопичення втоми у тижневих фізіологічних рядах класична лінійна статистика є недостатньою. Необхідні методи, що зберігають хронологічний контекст. Систематичний огляд Claudino et al. [2] підтверджує, що серед 17 досліджень з глибокого навчання в спортивній медицині рекурентні архітектури демонструють найвищу точність у задачах багатокрокового прогнозу. Проте вибір конкретної топології штучної нейронної мережі є критичним архітектурним завданням, оскільки математична структура шарів безпосередньо визначає здатність системи точно відтворювати складні нелінійні закономірності живого організму.

1.3.1 Класичні нейромережі прямого поширення (FFNN) та їх обмеження при обробці послідовностей

Традиційним підходом до розв'язання задач розпізнавання патернів, класифікації та регресії у комп'ютерних науках тривалий час слугувало

					ІАЛЦ.045420.004 ПЗ	Арк.
						15
Зм	Лист	№ докум.	Підп.	Дата		

використання штучних нейронних мереж прямого поширення (Feed-Forward Neural Networks, FFNN), яскравим представником яких є багатошаровий перцептрон (MLP)[19]. Математичний апарат таких мереж передбачає, що вектор вхідних ознак x трансформується послідовно через систему прихованих шарів, де кожен нейрон обчислює зважену суму своїх входів, додає зміщення (bias) та пропускає результат крізь нелінійну функцію активації (Sigmoid або ReLU).

Проте системний аналіз свідчить, що архітектура FFNN є принципово неефективною для задач предиктивного моніторингу тренувального навантаження. Ключовий недолік мереж прямого поширення полягає у повній відсутності механізмів внутрішнього збереження стану (пам'яті): кожен вхідний вектор телеметрії обробляється як ізольована подія, позбавлена хронологічного контексту.

У спортивній фізіології стан опорно-рухового апарату атлета є результатом тривалого накопичувального процесу: мікротравми тканин, виснаження центральної нервової системи та адаптаційні реакції мають виражений кумулятивний характер. Спроба адаптувати FFNN до обробки часових рядів шляхом об'єднання добових показників у єдиний плоский вектор (flattening) призводить до виникнення «прокляття розмірності» (curse of dimensionality — надмірне зростання кількості параметрів). Кількість вагові коефіцієнтів мережі зростає в геометричній прогресії, що за умов обмеженого обсягу реальних клінічних даних неминуче спричинює перенавчання моделі (overfitting) та втрату узагальнювальної здатності.

1.3.2 Одномірні згорткові нейронні мережі (1D-CNN) та їх специфіка в аналізі телеметрії

Як альтернатива класичним перцептронам у задачах аналізу послідовностей розглядаються згорткові нейронні мережі (CNN). На відміну від двовимірних архітектур (2D-CNN), що застосовуються в комп'ютерному зорі, для

					ІАЛЦ.045420.004 ПЗ	Арк.
						16
Зм	Лист	№ докум.	Підп.	Дата		

аналізу часових рядів фізіологічних метрик використовуються одномірні згорткові мережі (1D-CNN). Математична суть одномірної згортки полягає в переміщенні спеціалізованого ядра (фільтра) вздовж часової осі вхідного тензора, що дозволяє автоматично отримувати локальні ознаки та інваріантні патерни (наприклад, раптові сплески інтенсивності або критичні падіння тривалості сну).

Головною перевагою архітектури 1D-CNN є висока обчислювальна продуктивність. Оскільки операції згортки над різними ділянками часового ряду є математично незалежними, процес навчання та застосування мережі може бути ефективно розпаралелений на обчислювальних ядрах графічних прискорювачів (GPU) [8], що суттєво скорочує загальний час обробки.

Проте попри ефективність виділення локальних біомеханічних ознак, одномірні згорткові мережі мають суттєві обмеження при моделюванні довгострокових фізіологічних трендів. Архітектура 1D-CNN оперує жорстко фіксованим розміром рецептивного поля, визначеним розміром ядра та кількістю шарів. Це означає, що мережа здатна фіксувати взаємозв'язки між метриками лише у межах коротких локальних вікон. Для коректного обчислення спортивних індексів на кшталт ACWR системі необхідно безперервно утримувати контекстний зв'язок між гострим мікроциклом (7 діб) та хронічним мезоциклом (28 діб). Спроби розширити рецептивне поле через механізми розширеної згортки (dilated convolutions) частково нівелюють цю проблему, однак роблять топологію мережі надто складною для розгортання в легковагових мобільних середовищах. На практиці жорстка межа рецептивного поля перетворюється на системну перешкоду: 28-денна хронічна база ACWR вимагає контексту, що виходить за межі будь-якого розумного розміру ядра 1D-CNN.

1.3.3 Рекурентні архітектури (RNN) та переваги шарів LSTM для моделювання кумулятивного стресу

Для побудови математично точних моделей, орієнтованих на роботу з послідовними даними, найбільш обґрунтованим інженерним рішенням є використання рекурентних нейронних мереж (Recurrent Neural Networks, RNN). Фундаментальна особливість топології RNN полягає в наявності зворотних циклічних зв'язків між нейронами: під час обробки поточного елемента часового ряду мережа приймає на вхід не лише поточний вектор ознак, а й свій прихований стан з попереднього кроку, що функціонує як динамічний механізм внутрішньої пам'яті.

Проте при практичному застосуванні класичних архітектур RNN (мереж Елмана або Джордана)[18] для аналізу тривалих спортивних циклів виникає серйозна математична перешкода — ефект згасання або вибуху градієнта (vanishing/exploding gradient). Під час процедури зворотного поширення похибки в часі (Backpropagation Through Time, BPTT) градієнт функції втрат обчислюється через послідовне множення матриць ваг. Якщо ці ваги є меншими за одиницю, значення градієнта зменшується експоненційно з кожним кроком назад у часі, наближаючись до нуля. Унаслідок цього мережа втрачає здатність коригувати вагові коефіцієнти на основі подій, що відбулися на початку часового вікна, фактично «забуваючи» передісторію тренувань.

Для подолання цього обмеження в межах дипломного проєкту обрано архітектуру рекурентної нейромережі з довгою короткочасною пам'яттю (Long Short-Term Memory, LSTM) [3] як обґрунтований інструмент для моделювання кумулятивної втоми. Математичний каркас блока LSTM містить спеціалізовану клітинну лінію стану (cell state), вздовж якої інформація може безперешкодно протікати крізь час. Керування процесами запису, стирання та зчитування інформації здійснюється системою трьох нелінійних вентилів [3]:

- **вентиль забування (forget gate)** — аналізує поточний вхід та попередній прихований стан, визначаючи за допомогою сигмоїдної функції активації, яку частку застарілої інформації про накопичений стрес необхідно видалити з пам'яті клітини;
- **вентиль введення (input gate)** — визначає, які нові фізіологічні параметри поточного дня доцільно зафіксувати у стані клітини, паралельно генеруючи вектор нових кандидатів за допомогою гіперболічного тангенса;
- **вихідний вентиль (output gate)** — формує фінальний прихований стан поточного кроку, що виступає предиктивним значенням ризику втоми та передається на наступну ітерацію обчислень.

Вентильна структура ефективно ізолює лінійний потік градієнта, усуваючи математичну першопричину його згасання. У результаті нейромережа здатна стабільно обробляти тривимірний тензор стану спортсмена, виявляючи складні нелінійні довгострокові взаємозв'язки між дефіцитом сну, порушеннями стану вегетативної нервової системи [7] та ризиком виникнення безконтактних спортивних травм.

Схематичне порівняння розглянутих архітектур наведено на рисунку 1.2.

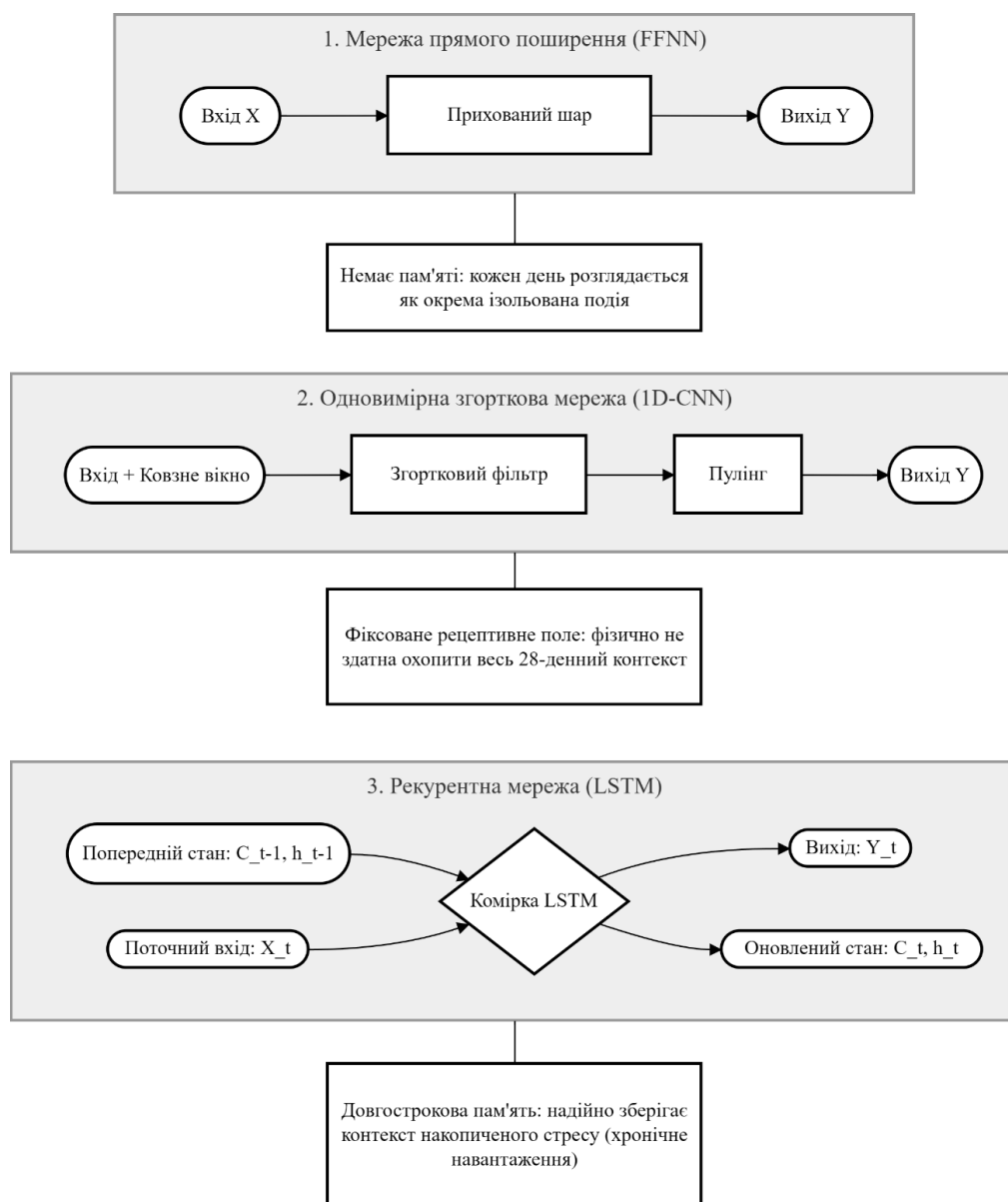


Рисунок 1.2 – Порівняльна схема трьох неймережевих архітектур у контексті аналізу фізіологічних часових рядів

1.4 Обґрунтування теми дипломного проєкту та постановка задачі

Проведений системний аналіз науково-технічної літератури, медико-біомеханічних досліджень та сучасного ринку програмних продуктів HealthTech дозволяє констатувати, що в галузі спортивної інженерії зберігається суттєвий технологічний розрив між рішеннями різного рівня. Сучасні інструменти превенції травматизму та контролю навантажень характеризуються або

надмірною інфраструктурною і фінансовою складністю, або критичною обмеженістю аналітичного апарату.

Масові фітнес-застосунки ігнорують позиційну специфіку навантажень і оперують лише кардіоваскулярними індексами. Водночас елітні комерційні системи дають точніший аналіз, але ціна апаратної залежності (датчики IMU, локальні UWB-мережі) виключає їх зі студентського та аматорського сегменту. Системи командного менеджменту на кшталт Kitman Labs покладаються на лінійну статистику, що принципово недостатньо для нелінійних накопичувальних процесів. Зовнішній чинник ризику (деградація піни взуття) залишається поза увагою всіх трьох категорій.

Для наочного узагальнення результатів проведеного аналізу, зіставлення функціональних можливостей існуючих рішень та обґрунтування необхідності розробки нової архітектури, ключові параметри систем зведено у таблицю 1.1.

					ІАЛЦ.045420.004 ПЗ	Арк.
						21
Зм	Лист	№ докум.	Підп.	Дата		

Таблиця 1.1 – Порівняльний аналіз систем моніторингу та превенції травматизму

Функціональний критерій	Масові фітнес-додатки (Nike Run, Strava)	Професійні трекери та платформи (Catapult, Zone7) [13, 14]	Розроблюваний мікросервісний комплекс
Апаратні вимоги	Смартфон або базовий смарт-годинник масового сегмента	Дорогочасні носимі датчики (IMU), локальні мережі UWB/LPS.	Доступний смартфон (введення sRPE та базової телеметрії сну)
Аналіз та прогнозування втоми	Базовий підрахунок калорій, відсутній предиктивний аналіз	Високоточний моніторинг, заснований на закритих хмарних алгоритмах ШІ.	Предиктивний аналіз нелінійних трендів мережею LSTM [3]
Моніторинг стану екіпірування	Спрощений лінійний підрахунок за кілометражем або часом	Повністю відсутній у більшості комерційних систем.	Адаптивна мультиплікативна модель (вага, покриття, тип взуття) [5]
Парадигма керування	Статичні шаблони планів без зворотного зв'язку	Дескриптивна аналітика (констатація факту виснаження без генерації рішень).	Прескриптивна аналітика — автоматична реконструкція планів модулем Drill Engine
Цільова аудиторія та модель	Масовий любительський ринок (B2C)	Елітні професійні клуби та організації (дорога B2B ліцензія).	Демократизований сегмент: напівпрофесіонали, любителі (B2C/B2B)

Аналіз даних таблиці 1.1 дозволяє обґрунтувати архітектурні переваги запропонованого рішення:

1. **Апаратна незалежність.** На відміну від систем класу Catapult, розроблюваний комплекс замінює апаратну залежність алгоритмічною ефективністю. Замість розгортання локальних радіомаяків у закритих приміщеннях система використовує математично модифікований метод Session-RPE [4], оцифровуючи обсяг виконаної роботи засобами програмних алгоритмів, що функціонують на стандартних мобільних пристроях.
2. **Предиктивне ядро аналізу втоми.** Аналітичний модуль ґрунтується на рекурентній неймережі LSTM [3], що дозволяє утримувати довгостроковий контекст часового ряду (взаємозв'язок між гострою втомою та хронічною тренованістю), забезпечуючи точність прогнозування ризику травм у межах прозорості відкритої архітектури.
3. **Адаптивний моніторинг інвентарю.** На противагу лінійному обліку зносу в Strava, впроваджено математичну модель із мультиплікативними ваговими коефіцієнтами, що враховує антропометричні параметри гравця та тип покриття майданчика. Це дозволяє завчасно попереджати про критичну деградацію амортизаційної піни EVA [5], яка є прямим каталізатором компресійних пошкоджень суглобів.
4. **Перехід до прескриптивної аналітики.** Якщо більшість комерційних систем лише фіксують факт перевтоми (дескриптивний підхід), розроблюваний комплекс реалізує прескриптивну парадигму: інтелектуальний модуль «Drill Engine» автоматично генерує адаптивний тренувальний контент залежно від ігрового амплуа (PG, SG, SF, PF, C) або формує протоколи відновлення при фіксації критичного рівня ризику.

Науково-практична цінність роботи полягає в емпіричному підтвердженні такої гіпотези: $MAE = 0,0227$ на синтетичному датасеті без апаратних трекерів

					ІАЛЦ.045420.004 ПЗ	Арк.
						23
Зм	Лист	№ докум.	Підп.	Дата		

доводить, що відкрита LSTM-архітектура на безкоштовному хмарному сервері є достатньою для предиктивного контролю навантаження в аматорському баскетболі без жодного пропрієтарного обладнання.

Для реалізації визначеної мети в межах дипломного проєкту необхідно спроектувати, програмно реалізувати та експериментально дослідити клієнт-серверний мікросервісний комплекс, виконавши такі технічні завдання:

- розробити логічну схему реляційної бази даних у СУБД PostgreSQL[11], впровадивши механізми пулінгу з'єднань для забезпечення стабільної взаємодії з ШІ-ядром у хмарному середовищі, та реалізувати транзакційну цілісність даних засобами ORM із механізмами каскадного видалення артефактів;
- сконструювати та навчити рекурентну нейронну мережу з архітектурою LSTM [3], налаштовану на обробку тривимірних тензорів багатовимірних часових рядів (7-денних вікон sRPE, тривалості сну та занять), інтегрувавши шари Dropout для запобігання перенавчанню;
- розробити алгоритм динамічного постобчислювального коригування результатів тензорного виводу нейромережі за допомогою нормалізованого коефіцієнта варіабельності серцевого ритму (HRV, параметр rMSSD) [7], що забезпечить миттєву реакцію системи на гострий фізіологічний стрес;
- створити рекомендаційний модуль асинхронного генерування тренувального та відновлювального контенту, диференційованого за п'ятьма ігровими позиціями;
- впровадити стохастичний алгоритм Фішера-Єйтса [9] для усунення проблеми «холодного старту» та запобігання алгоритмічному упередженню при реєстрації нових користувачів;
- реалізувати математичний модуль обчислення деградації амортизаційних властивостей взуття методом лінійної інтерполяції

індивідуальних вагових коефіцієнтів та експлуатаційних штрафів покриття майданчика;

— побудувати бекенд-інтерфейс на базі асинхронного фреймворку FastAPI [10] з інтеграцією Keep-Alive маршрутів для усунення затримок першого відгуку, та реалізувати кросплатформний клієнтський застосунок за технологією Progressive Web App (PWA) з підтримкою автономного функціонування та локального кешування даних за допомогою Service Worker [12].

Висновки до розділу 1

У першому розділі виконано комплексний аналіз предметної галузі та сформовано інженерну постановку задачі. Встановлено, що безконтактні травми нижніх кінцівок становлять понад 65% від загальної кількості пошкоджень у баскетболі та є прямим наслідком некерованого накопичення кумулятивного тренувального навантаження. Досліджено математичний апарат концепції ACWR і доведено, що зона оптимального балансу між гострим і хронічним навантаженням лежить у діапазоні 0.8–1.3.

Критичний огляд існуючих комерційних систем (Zone7, Catapult Sports та Kitman Labs) виявив спільний структурний недолік: всі три або залежать від дорогого пропрієтарного обладнання, або обмежуються дескриптивною аналітикою без генерації рекомендацій, або ігнорують фізичний стан екіпірування спортсмена.

Порівняльний аналіз нейромережових архітектур (FFNN, 1D-CNN, LSTM) встановив, що лише LSTM здатна зберігати довгостроковий контекст, необхідний для розрахунку 28-денної хронічної бази ACWR, не страждаючи від ефекту згасання градієнта. Це обґрунтовує вибір LSTM як обчислювального ядра BAIPS і визначає перелік технічних завдань, що реалізуються в наступних розділах.

					ІАЛЦ.045420.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		25

РОЗДІЛ 2. МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Другий розділ формалізує математичний фундамент чотирьох ключових алгоритмічних модулів системи. На основі аналізу біомеханічних характеристик спортивного взуття розроблено нелінійну мультиплікативну модель деградації амортизаційного ресурсу з трьома незалежними коефіцієнтами. Архітектуру рекурентної нейромережі LSTM описано через математичний апарат вентильних механізмів та сформовано специфікацію вхідного тензора за методом ковзного вікна. Додатково запропоновано метод гібридного постобчислювального коригування прогнозів через показник HRV та обґрунтовано застосування стохастичного алгоритму Фішера-Сйтса для модуля прескриптивної аналітики.

2.1 Математична модель предиктивної оцінки деградації спортивного екіпірування

У баскетболі гравці систематично виконують стрибки, різкі зупинки та зміни напрямку руху, що створює значне компресійне навантаження на суглоби нижніх кінцівок. Спортивне взуття (зокрема його амортизаційна міжпідшва із пінного матеріалу) є першою лінією захисту від цих ударів. Проте з часом піна ущільнюється й втрачає пружність, підвищуючи ризик безконтактних травм коліна та гомілковостопного суглоба.

Більшість існуючих систем обліку взуття спираються на лінійне накопичення часу або дистанції без поправки на індивідуальні характеристики спортсмена. Цей підхід є неточним: 90-кілограмовий центровий зношує підшву значно швидше, ніж 70-кілограмовий захисник за однакової кількості годин. У розробленій системі реалізовано алгоритм адаптивного розрахунку ресурсу взуття, який автоматично коригує еталонний показник довговічності залежно від

					ІАЛЦ.045420.004 ПЗ	Арк.
						26
Зм	Лист	№ докум.	Підп.	Дата		

маси спортсмена, типу ігрового покриття та функціонального призначення взуття.

Максимально допустимий безпечний ресурс пари визначається за формулою:

$$E_{max} = E_{base} \cdot k_{weight} \cdot k_{surface} \cdot k_{type} \quad (2.1)$$

де E_{max} — розрахунковий безпечний ресурс взуття у годинах; E_{base} — базовий еталонний ресурс, прийнятий рівним 80 годинам активного використання; k_{weight} — коефіцієнт, що враховує масу тіла спортсмена; $k_{surface}$ — коефіцієнт деградації від типу ігрового покриття; k_{type} — коефіцієнт спеціалізації взуття.

Щоб уникнути різкого стрибка в розрахунках, маса враховується через лінійну залежність. Пороговим значенням обрано 85 кг: за цієї масі коефіцієнт $k_{weight} = 1$ і ресурс не зменшується. За кожен кілограм понад цей поріг ресурс знижується на 1 %:

$$k_{weight} = 1.0 - (m - 85) \cdot 0.01 \quad (2.2)$$

де m — фактична маса тіла користувача у кілограмах.

Коефіцієнт ігрового покриття $k_{surface}$ відображає жорсткість поверхні майданчика та її вплив на темп зношення підошви. У системі закладено такі показники: для вуличного асфальту $k_{surface} = 0.5$; для комбінованих покриттів $k_{surface} = 0.75$; для гумового покриття (тартан) $k_{surface} = 0.8$; для баскетбольного паркету в залі $k_{surface} = 1.0$.

Коефіцієнт k_{type} враховує конструктивну адаптованість кросівок до баскетбольних рухів — зокрема до різких латеральних зупинок і приземлень. Спеціалізоване баскетбольне взуття отримує $k_{type} = 1$. Неспеціалізоване взуття (наприклад, бігові кросівки) штрафується на 30 % через недостатню бокову підтримку гомілковостопного суглоба: $k_{type} = 0.7$.

Поточний відсоток зношеності взуття оновлюється після кожного тренування, яке спортсмен реєструє в системі:

$$D = D_{init} + \left(\frac{t_{played}}{E_{max}} \right) \cdot 100 \quad (2.3)$$

де D — поточний розрахунковий відсоток зношеності взуття; D_{init} — початковий відсоток зношеності, вказаний користувачем під час реєстрації пари (актуально для вживаного взуття); t_{played} — сумарна тривалість тренувань у цьому взутті у годинах, що береться з бази даних.

Практична перевага цього підходу полягає в індивідуалізації сповіщень. Важкий гравець, що тренується переважно на вуличному асфальті, отримає попередження про необхідність заміни взуття значно раніше, ніж легкий спортсмен, який грає виключно в залі. Саме такий облік є першим, фізичним рівнем превенції травматизму в розробленій системі.

Алгоритм адаптивного розрахунку ресурсу взуття наведено на рисунку 2.1.

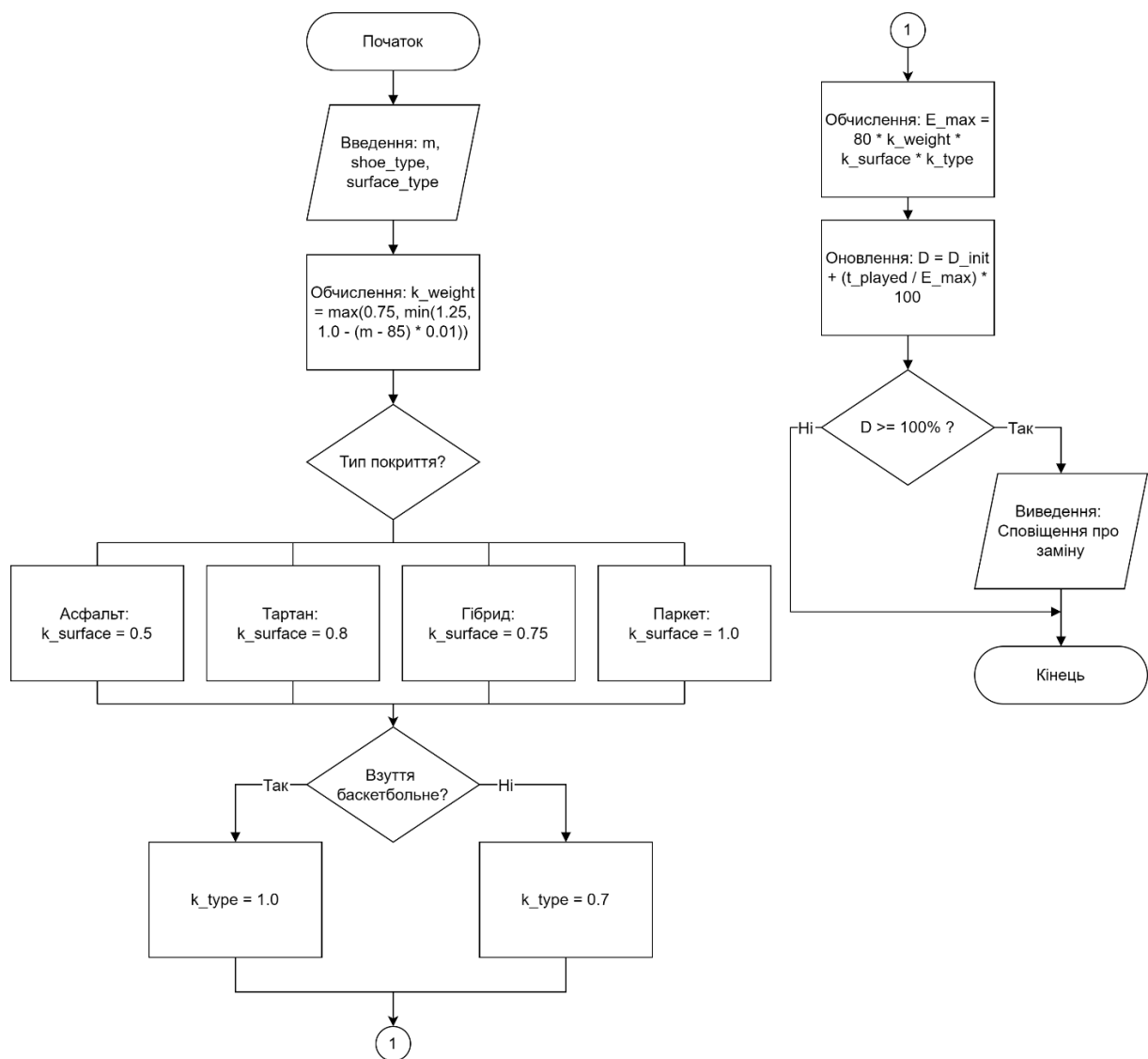


Рисунок 2.1 – Блок-схема алгоритму адаптивного розрахунку деградації амортизаційного ресурсу спортивного взуття

2.2 Архітектура та математичний апарат рекурентної нейромережі LSTM

Процес накопичення фізіологічної втоми є за своєю природою послідовним: стан організму спортсмена сьогодні визначається не лише вчорашнім тренуванням, а й усім тижневим ритмом навантажень і відновлення. Класичні нейронні мережі прямого поширення обробляють кожен вхідний вектор як незалежну подію, тому вони не підходять для цього завдання. Традиційні

рекурентні мережі (RNN) мають пам'ять, але страждають від математичної проблеми згасання градієнта: при навчанні на тривалих послідовностях сигнал про похибку «зникає» і мережа фактично перестає навчатися на ранніх даних.

Обчислювальним ядром розробленої системи виступає архітектура LSTM (Long Short-Term Memory — рекурентна мережа з довгою короткочасною пам'яттю). Її ключова конструктивна особливість — це спеціальна «лінія стану клітини» (cell state), що проходить крізь увесь часовий ряд і дозволяє мережі зберігати важливу інформацію без її спотворення. Доступ до цієї лінії регулюється системою логічних вентилів. Центральним з них є вентиль забування, що визначає, яку частку накопиченого стресу з попередніх днів залишити, а яку відкинути:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

де f_t — вектор вентиля забування (значення від 0 до 1, де 0 — «повністю забути», 1 — «повністю зберегти»); σ — сигмоїдна функція активації; W_f — матриця вагових коефіцієнтів вентиля; h_{t-1} — прихований стан мережі з попереднього дня (накопичений контекст стресу); x_t — вхідний вектор поточного дня; b_f — вектор зміщення (bias).

Вхідний вектор стану спортсмена x_t формується методом ковзного вікна розміром 7 діб. Для кожного дня мережа отримує три параметри. З урахуванням поправки на ігровий стрес вектор одного дня має такий вигляд:

$$x_t = [t_{sleep}, \min(RPE \cdot k_{game}, 10), t_{duration} \cdot k_{game}] \quad (2.5)$$

де t_{sleep} — тривалість нічного сну у годинах; RPE — суб'єктивна оцінка важкості навантаження за 10-бальною шкалою Борга; $t_{duration}$ — фактична тривалість фізичної активності у хвилинали; k_{game} — коефіцієнт ігрового стресу ($k_{game} = 1.25$ для офіційних матчів; $k_{game} = 1.0$ для тренувань та відновлення); значення $RPE \cdot k_{game}$ обмежується максимумом 10 — верхньою межею шкали Борга.

Перед подачею до мережі всі вхідні дані нормалізуються за методом Min-Max, який приводить значення кожного показника до діапазону від 0 до 1:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.6)$$

де X_{norm} — нормалізоване значення показника; X — фактичне значення показника; X_{min} та X_{max} — мінімальне та максимальне значення цього показника у навчальній вибірці.

Програмна архітектура нейромережі складається з чотирьох шарів. Перший — рекурентний шар LSTM з 32 нейронами та функцією активації ReLU. Одразу після нього розміщено шар регуляризації Dropout, який під час кожної ітерації навчання випадково вимикає 20 % зв'язків. Це запобігає перенавчанню моделі. Третій шар — повнозв'язний (Dense) з 16 нейронами та функцією активації ReLU. Завершує ланцюжок вихідний нейрон із сигмоїдною функцією активації, що математично обмежує результат діапазоном від 0 до 1:

$$f(z) = \frac{1}{1 + e^{-z}}$$

де $f(z)$ — розрахований базовий показник ризику перевтоми (у діапазоні від 0 до 1); z — зважена сума виходів нейронів попереднього шару.

Мережу скомпільовано з оптимізатором Adam зі швидкістю навчання 0.001.

Архітектуру рекурентної нейромережі BasketballFatigueNet наведено на рисунку 2.2.

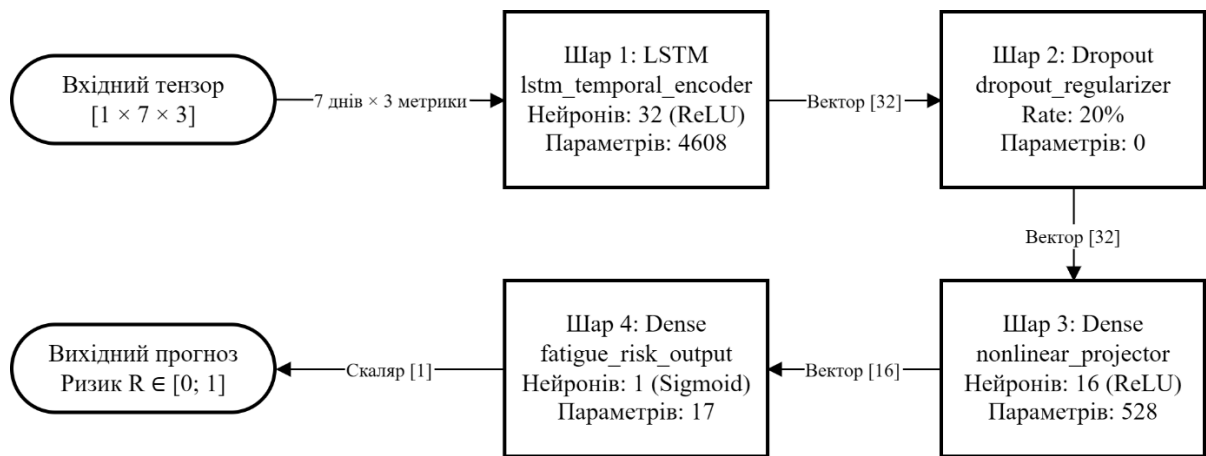


Рисунок 2.2 – Архітектура рекурентної нейромережі BasketballFatigueNet для предиктивної оцінки ризику перевтоми

Навчання тривало 15 епох із розміром пакета 64 зразків та контрольною вибіркою у 20% від загального набору даних. Точність прогнозу вимірюється метрикою середньої абсолютної похибки (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.8)$$

де n — кількість спостережень у вибірці; y_i — еталонне (фактичне) значення ризику; $\hat{y}_i$ — значення, розраховане нейромережею.

Показник варіабельності серцевого ритму (HRV) свідомо виключено з архітектури мережі. Він інтегрується як зовнішній модифікатор на етапі постобробки, що описано у наступному підрозділі.

2.3 Метод гібридного постобчислювального коригування прогнозів на основі HRV

Варіабельність серцевого ритму (HRV, параметр rMSSD) є одним із найбільш інформативних маркерів відновлення центральної нервової системи. Низьке значення HRV після напруженого тижня вказує на перевтому значно виразніше, ніж будь-яке суб'єктивне оцінювання. Проте інтегрувати HRV

безпосередньо у вхідний тензор LSTM технічно ризиковано: спортсмени вимірюють цей показник нерегулярно, і будь-який пропущений день призведе до помилки або суттєвого спотворення прогнозу.

Для вирішення цієї суперечності в системі реалізовано метод гібридного постобчислювального коригування. Алгоритм сканує записи за останні сім днів у зворотному хронологічному порядку та знаходить найсвіжіше дійсне вимірювання HRV. Якщо жодного вимірювання немає, то модифікатор не застосовується, і система повертає базовий прогноз LSTM без змін.

За базовий орієнтир у системі прийнято значення $HRV = 55$ мс — типовий показник для активних спортсменів[7, 21]. Кожна мілісекунда відхилення від цієї норми змінює ризик на 0.8 %:

$$M_{raw} = 1 - (HRV_{latest} - 55) \cdot 0.008 \quad (2.9)$$

де M_{raw} — попередній коефіцієнт модифікації ризику; HRV_{latest} — останнє зафіксоване значення варіабельності серцевого ритму у мілісекундах.

Щоб аномальне чи помилкове вимірювання пульсу не призводило до нереалістичних змін у прогнозі, вплив модифікатора жорстко обмежено: максимальна корекція у будь-який бік не перевищує 25 %:

$$M_{final} = \max(0,75, \min(1,25, M_{raw})) \quad (2.10)$$

де M_{final} — фінальний обмежений коефіцієнт впливу HRV; $\max$ та $\min$ — математичні функції обмеження нижньої та верхньої меж.

На фінальному кроці базовий прогноз мережі множиться на цей коефіцієнт. Результат додатково відтинається у межах від 0 до 1, щоб значення залишалось коректним відсотком ймовірності:

$$R_{final} = \max(0, \min(R_{lstm} \cdot M_{final}, 1)) \quad (2.11)$$

де R_{final} — фінальний предиктивний показник ризику перевтоми; R_{lstm} — базовий ризик, розрахований архітектурою LSTM; M_{final} — гібридний коефіцієнт впливу серцево-судинних показників.

Практично це означає таке. Якщо LSTM розраховала ризик на рівні 0.60 (60%), а поточне HRV спортсмена становить 40 мс (значно нижче норми), алгоритм обчислює: $M_{raw} = 1 - (40 - 55) \cdot 0.008 = 1.12$. Оскільки 1.12 знаходиться в допустимому діапазоні [0.75; 1.25], фінальний ризик складе $0.60 \cdot 1.12 = 0,672$, тобто підвищиться до 67.2%. Такий механізм відображає реальну фізіологічну картину: кількісні показники навантаження можуть виглядати помірними, але пригнічений стан нервової системи вказує на вищий фактичний ризик[7, 21].

Залежність коефіцієнта коригування від значення HRV у допустимих межах наведено на рисунку 2.3.

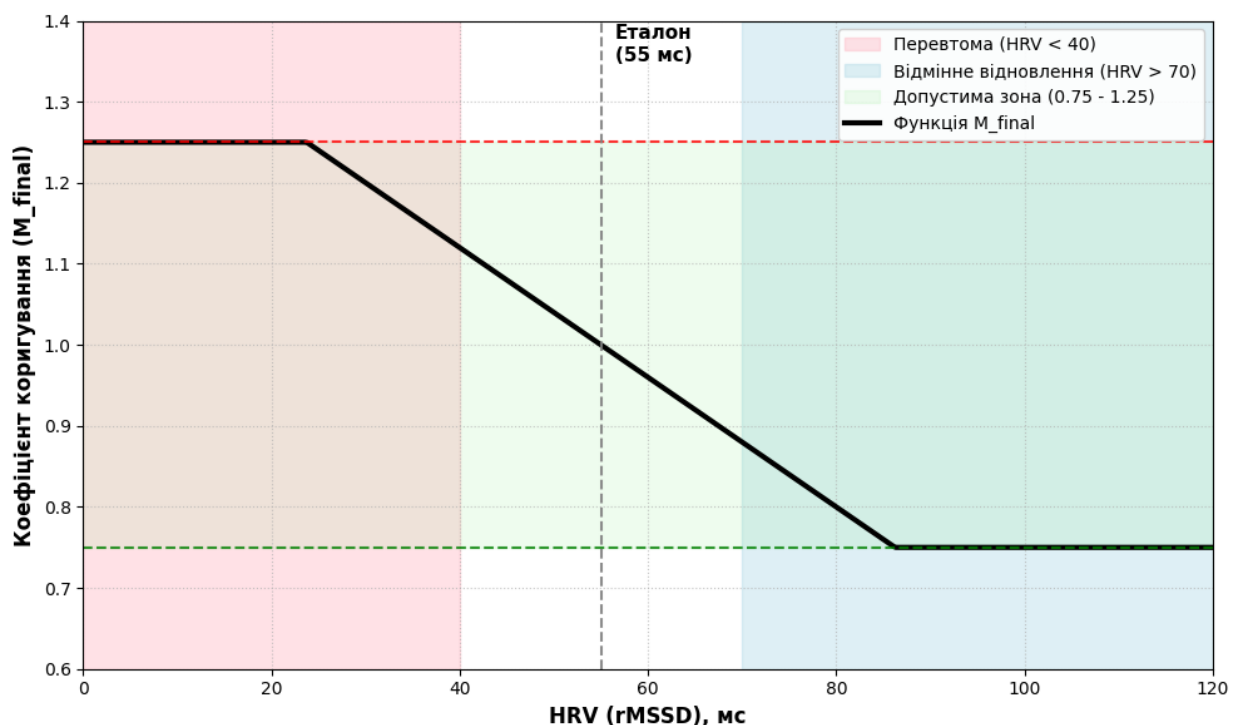


Рисунок 2.3 – Залежність коефіцієнта коригування ризику (M_{final}) від значення варіабельності серцевого ритму (HRV)

2.4 Стохастичний алгоритм генерації тренувальних планів (метод Фішера-Єйтса)

Обчислення рівня ризику є кінцевою точкою аналітичного циклу, але не кінцевою точкою роботи системи. Отримавши прогноз, комплекс переходить до автоматичного формування індивідуального тренувального протоколу. Рекомендації мають змінюватись від сесії до сесії, щоб уникнути одноманітності та підтримати мотивацію спортсмена в довгостроковій перспективі.

Залежно від розрахованого рівня ризику алгоритм звертається до одного з трьох вбудованих наборів вправ:

- при критичному ризику ($R_{final} > 0.75$) — набір протоколів відновлення центральної нервової системи (7 варіантів);
- при помірному ризику ($0.4 \leq R_{final} \leq 0.75$) — набір технічних вправ низької інтенсивності без стрибків та спринтів;
- при оптимальному стані ($R_{final} < 0.40$) — матриця розвивальних тренувань, розподілена за п'ятьма ігровими позиціями (PG, SG, SF, PF, C).

Вибір конкретних вправ із кожного набору здійснюється методом вибірки без повернення, що програмно реалізується через вбудовану функцію `random.sample()` мови Python. В основі цієї функції лежить алгоритм Фішера-Єйтса — класичний метод стохастичного тасування масиву за лінійний час $O(n)$. Математично ймовірність потрапляння будь-якої конкретної вправи до сформованого плану є рівномірною:

$$P = \frac{k}{n} \quad (2.12)$$

де P — ймовірність вибору конкретної вправи; k — кількість вправ, що алгоритм має обрати до плану; n — загальна кількість доступних вправ у відповідному наборі.

Для критичного сценарію ($n = 7, k = 3$) алгоритм обирає три процедури відновлення з рівноймовірністю $P = \frac{k}{n} \approx 0.43$ для кожної. Фокус плану при цьому — відновлення та сон.

При оптимальному стані логіка ускладнюється для забезпечення позиційної специфічності. Система послідовно виконує кілька незалежних вибірок: по одній вправі для розминки та заминки, по дві вправи з наборів основних та допоміжних навичок, характерних для конкретного амплуа гравця. Алгоритм вирішує дві окремі інженерні задачі: захищає тіло при підвищеному ризику і забезпечує різноманітність занять при нормальному стані спортсмена.

Блок-схему алгоритму модуля Drill Engine наведено на рисунку 2.4.

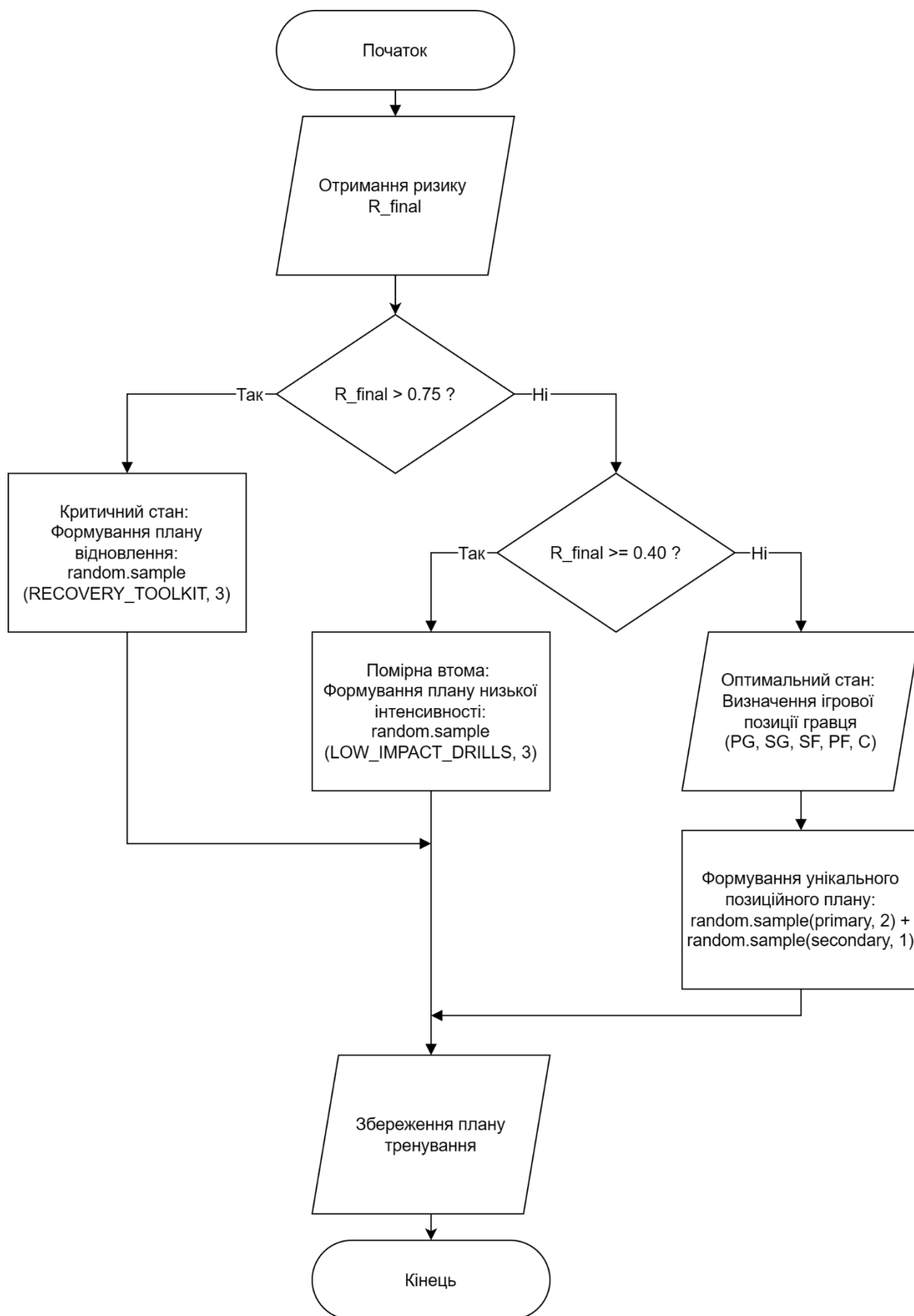


Рисунок 2.4 – Блок-схема модуля Drill Engine: розгалуження за рівнями ризику та стохастичний вибір вправ

Висновки до розділу 2

У другому розділі розроблено повне математичне та алгоритмічне забезпечення чотирьох ключових модулів системи. Сформовано нелінійну мультиплікативну модель деградації амортизаційного ресурсу спортивного взуття (формули 2.1–2.3), що вперше в аналогічних системах враховує одночасно масу атлета, тип ігрового покриття та функціональне призначення взуття.

Формалізовано архітектуру рекурентної нейромережі BasketballFatigueNet: чотиришарова LSTM-топологія з 32 рекурентними нейронами, шаром Dropout (20%), прихованим Dense-шаром (16 нейронів) та сигмоїдним вихідним нейроном. Вхідний вектор x_t розмірністю $[7 \times 3]$ формується методом ковзного вікна та нормалізується за алгоритмом Min-Max.

Запропоновано метод гібридного постобчислювального коригування прогнозів через показник HRV (формули 2.9–2.11), що дозволяє системі оперативно реагувати на стан вегетативної нервової системи без перенавчання моделі. Для модуля Drill Engine розроблено алгоритм стохастичного вибору на основі методу Фішера-Єйтса (формула 2.12), що забезпечує різноманітність тренувальних протоколів і захищає від проблеми «холодного старту».

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОГО КОМПЛЕКСУ

Третій розділ охоплює всі рівні програмної реалізації комплексу: від вибору інструментального стека до розгортання кросплатформного клієнтського інтерфейсу. Спочатку обґрунтовано вибір Python/FastAPI/TensorFlow як єдиного бекенд-середовища, що дозволяє утримувати навчену LSTM-модель в оперативній пам'яті й обслуговувати HTTP-запити без міжпроцесного транспорту. Далі описано проектування реляційної схеми PostgreSQL із двома стратегіями видалення зовнішніх ключів, а також реалізацію REST API та механізмів безпеки (bcrypt + JWT HS256). Розділ завершується описом клієнтського PWA-застосунку на Vue.js 3 з підтримкою автономного кешування засобами Service Worker.

3.1 Вибір та обґрунтування інструментальних засобів розробки (FastAPI, TensorFlow)

Розроблюваний комплекс одночасно вирішує три принципово різні технічні задачі в межах одного серверного середовища: обробляє HTTP-запити від клієнтів, підтримує транзакційну цілісність реляційних даних і виконує застосування навченої нейромережі LSTM[25]. Поєднання цих трьох функцій визначило вибір Python як єдиної мови серверного рівня. Лише вона має зрілі та сумісні між собою бібліотеки для кожної з цих задач, що дозволило уникнути міжмовних шин і тримати весь бекенд-стек в одному процесі. Систему чітко розділено на серверну (бекенд) та клієнтську (фронтенд) частини: це забезпечує незалежність модулів і спрощує подальший розвиток кожного з них окремо.

Python обрано єдиною мовою серверного рівня з прагматичної причини: лише він забезпечує зрілі, сумісні між собою бібліотеки одразу для трьох задач, що вирішуються на одному сервері: FastAPI для HTTP-шлюзу, SQLAlchemy для

					ІАЛЦ.045420.004 ПЗ	Арк.
						39
Зм	Лист	№ докум.	Підп.	Дата		

роботи з PostgreSQL, TensorFlow для інференсу LSTM. Будь-яка поліглотна архітектура (наприклад, Node.js + Python microservice) вимагала б додаткового транспортного шару між компонентами, що ускладнило б розгортання на безкоштовному хмарному рівні.

Загальну архітектуру системи наведено на рисунку 3.1.

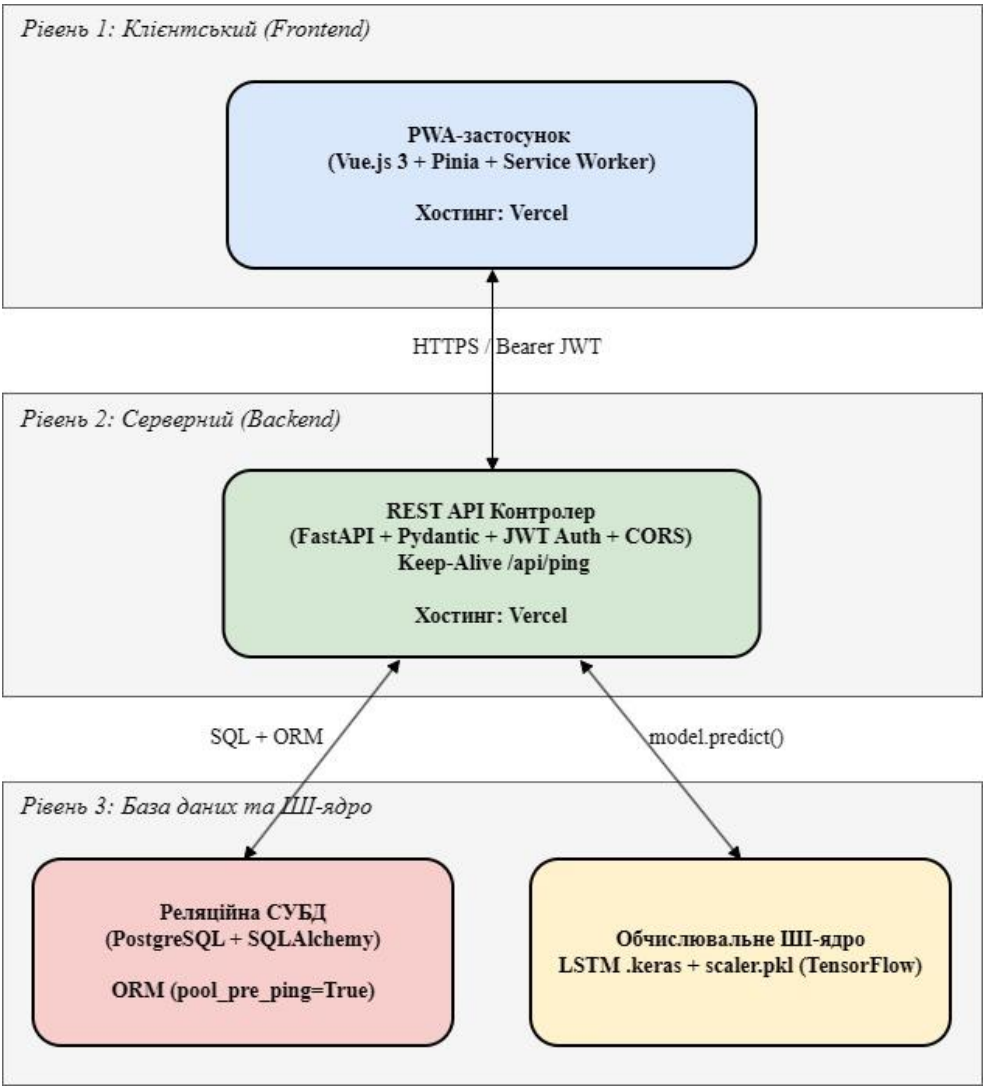


Рисунок 3.1 – Трирівнева архітектура інформаційно-аналітичного комплексу

Для побудови програмного інтерфейсу сервера обрано фреймворк FastAPI. На відміну від традиційних синхронних рішень на кшталт Flask або Django, FastAPI базується на стандарті ASGI (Asynchronous Server Gateway Interface) та забезпечує нативну підтримку асинхронних обробників запитів. Для

аналітичного сервісу це принципово: під час обчислення прогнозу нейромережею сервер не блокується в очікуванні результату, а паралельно обробляє запити інших користувачів. Додатковою перевагою FastAPI є вбудована інтеграція з бібліотекою Pydantic. Вона відхиляє некоректні запити ще на рівні вхідного шлюзу, до того як вони досягають бізнес-логіки.

Для розв'язання задачі машинного навчання та побудови рекурентної нейромережі LSTM задіяно бібліотеку TensorFlow та її високорівневий інтерфейс Keras. Ці інструменти дозволяють ефективно конструювати шари мережі, налаштовувати алгоритми регуляризації та зберігати навчену модель у бінарному форматі .keras для подальшого завантаження сервером. Для попередньої нормалізації показників телеметрії застосовано клас MinMaxScaler з бібліотеки Scikit-learn[20]: після навчання об'єкт нормалізатора записується у двійковий файл scaler.pkl і завантажується разом із моделлю під час запуску сервера.

Зберігання постійних даних реалізовано засобами СУБД PostgreSQL. Для взаємодії Python-коду з базою даних інтегровано бібліотеку SQLAlchemy у режимі об'єктно-реляційного відображення (ORM). Маніпуляції з записами виконуються через Python-об'єкти, а не через ручне написання SQL-запитів, що виключає вразливості класу SQL-ін'єкцій та спрощує зміну схеми бази даних у майбутньому.

Клієнтську частину системи реалізовано на фреймворку Vue.js третьої версії. Його реактивна компонентна архітектура дозволяє формувати складні інформаційні панелі без перезавантаження сторінки. Збірка виконується інструментом Vite, що забезпечує миттєве оновлення модулів у режимі розробки та оптимізований пакет для продуктивного розгортання.

Ключовим архітектурним рішенням для клієнтської частини стало впровадження технології прогресивних вебзастосунків (Progressive Web Application, PWA). З огляду на специфіку застосування системи(у спортивних залах і на відкритих майданчиках з нестабільним інтернет-з'єднанням) PWA через механізм Service Workers кешує ресурси застосунку локально на пристрої

					ІАЛЦ.045420.004 ПЗ	Арк.
						41
Зм	Лист	№ докум.	Підп.	Дата		

та дозволяє встановити його без використання App Store або Google Play, що відповідає специфікації W3C [12, 23].

3.2 Проектування логічної структури реляційної бази даних PostgreSQL

Оскільки бекенд розгорнуто на хмарній платформі Render, де мережеві з'єднання між сервером додатку та сервером бази даних можуть перериватися після тривалого простою, у конфігурації підключення активовано захисні параметри пулу з'єднань. Параметр **pool_pre_ping=True** змушує SQLAlchemy надсилати тестовий запит перед кожною транзакцією для автоматичної перевірки статусу з'єднання. Параметр **pool_recycle=300** встановлює примусове оновлення з'єднань кожні 300 секунд, що запобігає помилці «server closed the connection unexpectedly» після тривалої бездіяльності. Базовий розмір пулу складає 5 з'єднань (**pool_size=5**) з максимальним перевищенням до 10 (**max_overflow=10**).

Логічну схему реляційної бази даних інформаційно-аналітичного комплексу наведено на кресленні ІАЛЦ.045420.006 Д2 (Додаток А).

Вона побудована навколо чотирьох взаємопов'язаних таблиць, кожна з яких відповідає за окремий домен бізнес-логіки:

- **таблиця «users» (Профілі гравців).** Виступає головною сутністю бази даних. Зберігає унікальну електронну адресу (email), хеш пароля (password_hash), а також базові характеристики спортсмена: ім'я, вік, масу тіла у кілограмах (weight_kg), зріст та ігрове амплуа (position). Ці атрибути є статичними модифікаторами для алгоритмів розрахунку деградації екіпірування та генерації позиційно-диференційованих тренувальних планів;
- **таблиця shoes_inventory (Інвентар).** Зберігає назву моделі взуття (brand_model), комбінований рядок типу і покриття (cushion_type, формат «тип | покриття», наприклад «Баскетбольні | Паркет»), поточне напруження у годинах (current_hours_played) та індивідуально

					ІАЛЦ.045420.004 ПЗ	Арк.
						42
Зм	Лист	№ докум.	Підп.	Дата		

розрахований максимальний ресурс (`max_lifespan_hours`). Під час реєстрації кожного тренування поле `current_hours_played` автоматично збільшується на тривалість сесії;

- **таблиця «daily_metrics» (Щоденна телеметрія).** Є центральним сховищем динамічних даних для навчання та інференсу нейромережі. Зберігає щоденні показники: тривалість сну (`sleep_hours`), суб'єктивну оцінку важкості навантаження (`rpe_score`), тривалість активності у хвилинах (`duration_minutes`), тип активності (`activity_type`): «Training», «Game» або «Recovery», та необов'язковий показник варіабельності серцевого ритму (`hrv_value`). На цій таблиці побудовано складений індекс `ix_daily_metrics_user_date` за полями `user_id` та `date`, що забезпечує швидке вилучення 7-денних вікон для формування вхідних тензорів;
- **таблиця «generated_plans» (Згенеровані протоколи).** Призначена для архівації результатів роботи аналітичного ядра. Зберігає розрахований рівень ризику (`fatigue_risk`), тематичний фокус заняття (`plan_focus`) та детальний протокол тренування (`plan_content`) у форматі простого тексту (тип TEXT) з розмежуванням рядків. Для ефективного пошуку актуального плану на цій таблиці також побудовано складений індекс за полями `user_id` та `date`.

Найскладнішою задачею при проектуванні схеми стало забезпечення цілісності даних при видаленні: два різні сценарії вимагають принципово різних стратегій. Між таблицею `users` і всіма залежними таблицями налаштовано каскадний зв'язок з параметром `cascade="all, delete-orphan"`: при видаленні профілю користувача система безповоротно видаляє всю його телеметрію, інвентар і згенеровані плани, що відповідає праву на видалення персональних даних.

Проте зв'язок між таблицями `daily_metrics` та `shoes_inventory` реалізовано інакше. При видаленні пари взуття з інвентарю відповідні записи в щоденній

					ІАЛЦ.045420.004 ПЗ	Арк.
						43
Зм	Лист	№ докум.	Підп.	Дата		

телеметрії не повинні зникати. Вони є частиною хронічного навантаження при розрахунку ACWR. Тому зовнішній ключ `shoe_id` у таблиці `daily_metrics` налаштовано з правилом `ondelete="SET NULL"`: при видаленні взуття з інвентарю відповідне поле в рядках телеметрії просто обнуляється, зберігаючи математичну статистику сесії.

3.3 Розробка мікросервісної бекенд-архітектури та інтеграція ШІ-ядра

Серверна частина побудована за архітектурним стилем REST API (Representational State Transfer). Для контролю міждоменного доступу між клієнтом і сервером у конфігурації `CORSMiddleware` явно дозволено запити від локальних адрес розробки (`http://localhost:5173`, `http://127.0.0.1:5173`) та від виробничої адреси PWA-застосунку, розгорнутого на платформі Vercel (`https://basketball-ai-system.vercel.app`). Запити з будь-яких інших джерел автоматично відхиляються на рівні проміжного шару.

Усі вхідні дані перевіряються через Pydantic-схеми ще до потрапляння в бізнес-логіку. Принциповою відмінністю від типових REST API є рівень захисту на рівні самих схем: схема `MetricCreate` реалізує декоратор `@model_validator`, який при отриманні типу активності «Recovery» примусово обнуляє поля `duration_minutes` та `pre_score`, ігноруючи будь-які значення клієнта. Такий підхід виключає математичні аномалії при подальшому формуванні вхідних тензорів нейромережі.

Паролі користувачів зберігаються у базі даних виключно у вигляді одностороннього хешу, сформованого бібліотекою `bcrypt` з унікальним криптографічним доповненням (`salt`), що генерується автоматично для кожного користувача. Для авторизації сесій реалізовано механізм токенів JSON (JWT) за стандартом OAuth2. Під час успішної автентифікації сервер генерує підписаний токен доступу за алгоритмом HS256 з терміном дії 7 діб (`ACCESS_TOKEN_EXPIRE_MINUTES = 60 * 24 * 7`). Клієнтський застосунок

					ІАЛЦ.045420.004 ПЗ	Арк.
						44
Зм	Лист	№ докум.	Підп.	Дата		

автоматично додає цей токен до заголовка Authorization: Bearer кожного захищеного запиту; сервер верифікує підпис за секретним ключем і ідентифікує користувача без повторного введення пароля.

Маршрутизація API логічно поділена на чотири блоки: управління профілем (/api/users/...), моніторинг інвентарю (/api/shoes/...), обробка телеметрії (/api/users/{user_id}/metrics) та модуль предиктивної аналітики (/api/ai/generate_plan/{user_id}). Повний перелік кінцевих точок REST API, автоматично сформований FastAPI, наведено на рисунку 3.2.

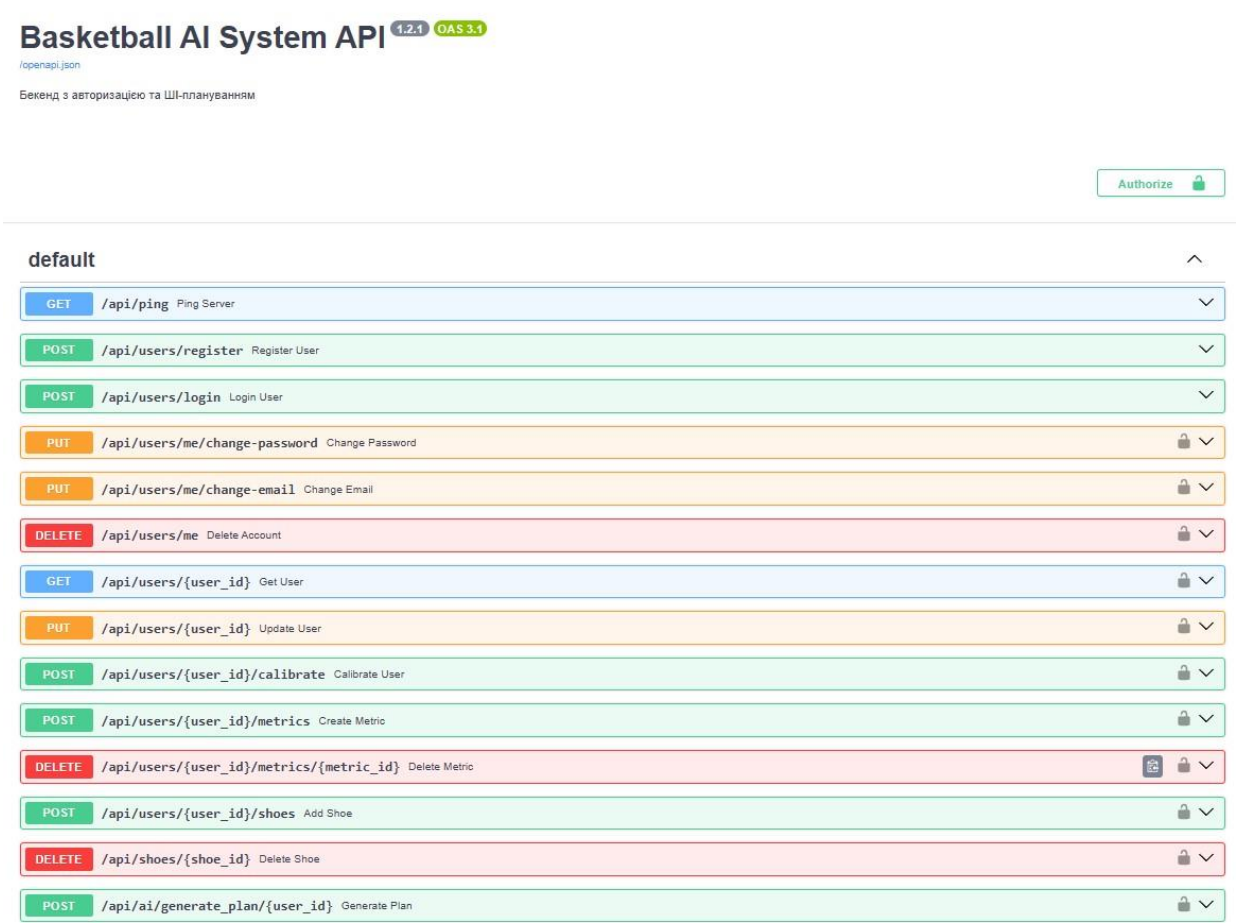


Рисунок 3.2 – Автоматична документація REST API у середовищі Swagger UI

Обчислювальне ядро (попередньо навчена модель `basketball_lstm.keras` та об'єкт нормалізатора `scaler.pkl`) завантажується в оперативну пам'ять сервера лише один раз під час ініціалізації через `tf.keras.models.load_model()` та

`pickle.load()`. Зберігання моделі в пам'яті між запитами унеможлиблює її повторне завантаження і знижує затримку при кожному окремому виклику інференсу.

Враховуючи, що бекенд розгорнуто на безкоштовному рівні Render, де сервери автоматично переходять у сплячий режим за відсутності запитів, реалізовано службовий маршрут Keep-Alive GET `/api/ping`. Зовнішня служба моніторингу UptimeRobot звертається до цього маршруту кожні 5 хвилин, підтримуючи сервер і завантажену нейромережу в активному стані. Без цього механізму перший запит після простою міг чекати до 60 секунд на холодний запуск середовища TensorFlow, що спричиняло б систематичні помилки тайм-ауту на стороні клієнта.

3.4 Реалізація кросплатформного клієнтського інтерфейсу за технологією PWA

Для взаємодії спортсменів з аналітичним ядром розроблено кросплатформний клієнтський застосунок на базі Vue.js 3. Компонентна архітектура фреймворку дозволила розподілити інтерфейс на незалежні модулі. Кожен відповідає за один домен відображення і не впливає на логіку сусідніх.

Управління глобальним станом застосунку реалізовано через централізоване Pinia-сховище (`userStore`). Сховище відповідає за збереження сесійних даних: ідентифікатора гравця (`userId`) та JWT-токена (`token`). Обидва значення записуються у `localStorage` під ключами `userId` та `token`, що дозволяє відновлювати сесію після закриття застосунку без повторного введення пароля. Методи сховища (`fetchUser`, `setAuthData`, `logout`) автоматично вбудовують токен у заголовки всіх вихідних мережевих запитів.

Навігацію між екранами реалізовано через Vue Router 4 із двостороннім навігаційним охоронцем (`router.beforeEach`): неавторизований користувач, що намагається відкрити захищений маршрут, перенаправляється на екран входу; авторизований, що знову звертається до сторінки входу перенаправляється

					ІАЛЦ.045420.004 ПЗ	Арк.
						46
Зм	Лист	№ докум.	Підп.	Дата		

одразу на головну панель. Такий двосторонній захист унеможливорює несанкціонований доступ і запобігає накопиченню зайвих записів у стеку навігації.

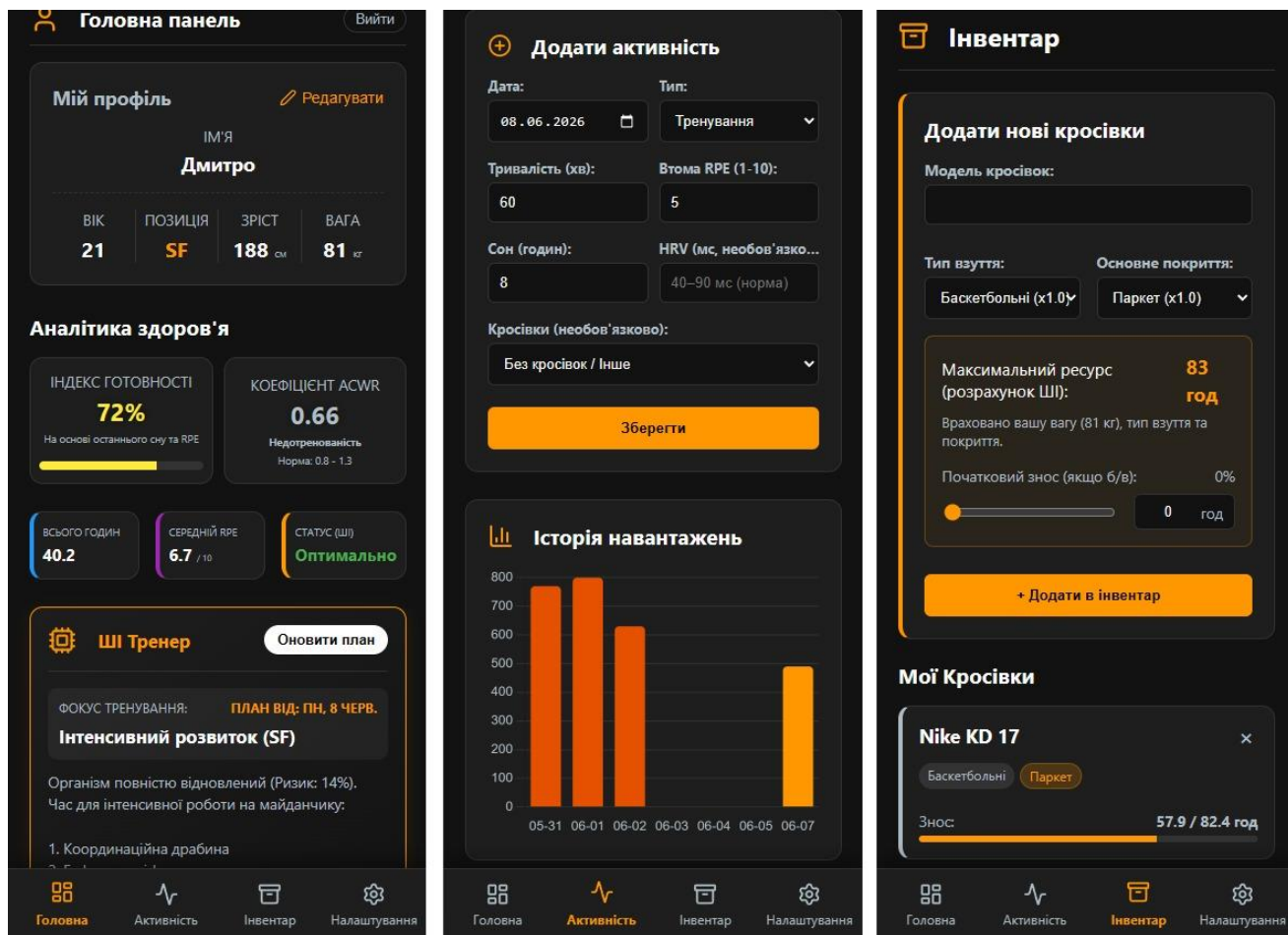
Візуальний інтерфейс поділено на чотири основні екрани:

- **головна панель (Dashboard)** — центр управління спортсмена, де відображаються всі ключові показники: індекс готовності організму (`readiness_score`), значення коефіцієнта ACWR з підписом відповідної зони ризику, класифікований AI-статус перевтоми від нейромережі та актуальний тренувальний протокол від модуля Drill Engine. Профільна картка підтримує режим `inline`-редагування антропометричних даних. При першому завантаженні новим користувачем автоматично відкривається двокроковий майстер калібрування, що формує початкову 7-денну базу телеметрії методом Fisher-Yates для рівномірного розподілу тренувальних днів;
- **панель активності (Activity)** — призначена для щоденного внесення фізіологічних параметрів: тривалості нічного сну, суб'єктивного RPE, тривалості заняття та необов'язкового HRV. При реєстрації тренування або матчу користувач може додатково обрати конкретну пару кросівок зі свого інвентарю. Система автоматично збільшує лічильник напруження (`current_hours_played`) для обраної пари на тривалість зареєстрованої сесії, синхронізуючи фізичний знос взуття з реальним навантаженням. Нижче форми розміщено стовпчастий графік на базі Chart.js, що відображає динаміку sRPE-навантаження за останні 7 днів із кольоровою індикацією перевантаження. Блок «Щоденник тренувань» відображає хронологічну таблицю всіх зареєстрованих сесій із можливістю видалення запису та експорту повної телеметрії у форматі CSV;
- **панель екіпірування (Gear)** — реалізує управління інвентарем спортивного взуття. Форма реєстрації нової пари включає вбудований

III-калькулятор, що у режимі реального часу обчислює прогнозований ресурс взуття (E_{max}) відповідно до математичної моделі (2.1) залежно від маси гравця, типу покриття та категорії взуття. Кожна зареєстрована пара відображається карткою з прогрес-баром зношеності та кольоровим попередженням при досягненні критичного стану;

- **налаштування (Settings)** — структуровані у три вкладки: особисті дані (зміна email), безпека (зміна пароля з перевіркою підтвердження та контролем мінімальної довжини) та «Небезпечна зона» (безповоротне видалення акаунта з каскадним очищенням усіх пов'язаних записів із бази даних).

Візуальне представлення основних панелей мобільного інтерфейсу наведено на рисунку 3.3.



а)

б)

в)

Рисунок 3.3 – Кросплатформний клієнтський інтерфейс застосунку: а) головна інформаційна панель; б) панель активності; в) панель управління інвентарем

Оскільки цільова аудиторія — спортсмени на майданчиках з нестабільним з'єднанням, клієнтський застосунок реалізовано за специфікацією PWA. Збірка виконується інструментом Vite з підключеним плагіном vite-plugin-pwa, що автоматично генерує Service Worker та файл manifest.json. Service Worker при першому завантаженні кешує повний статичний шар застосунку: HTML-розмітку, скомпільовані CSS та JavaScript-модулі, іконки та шрифти.

Принцип роботи Service Worker у режимах онлайн та офлайн наведено на рисунку 3.4.

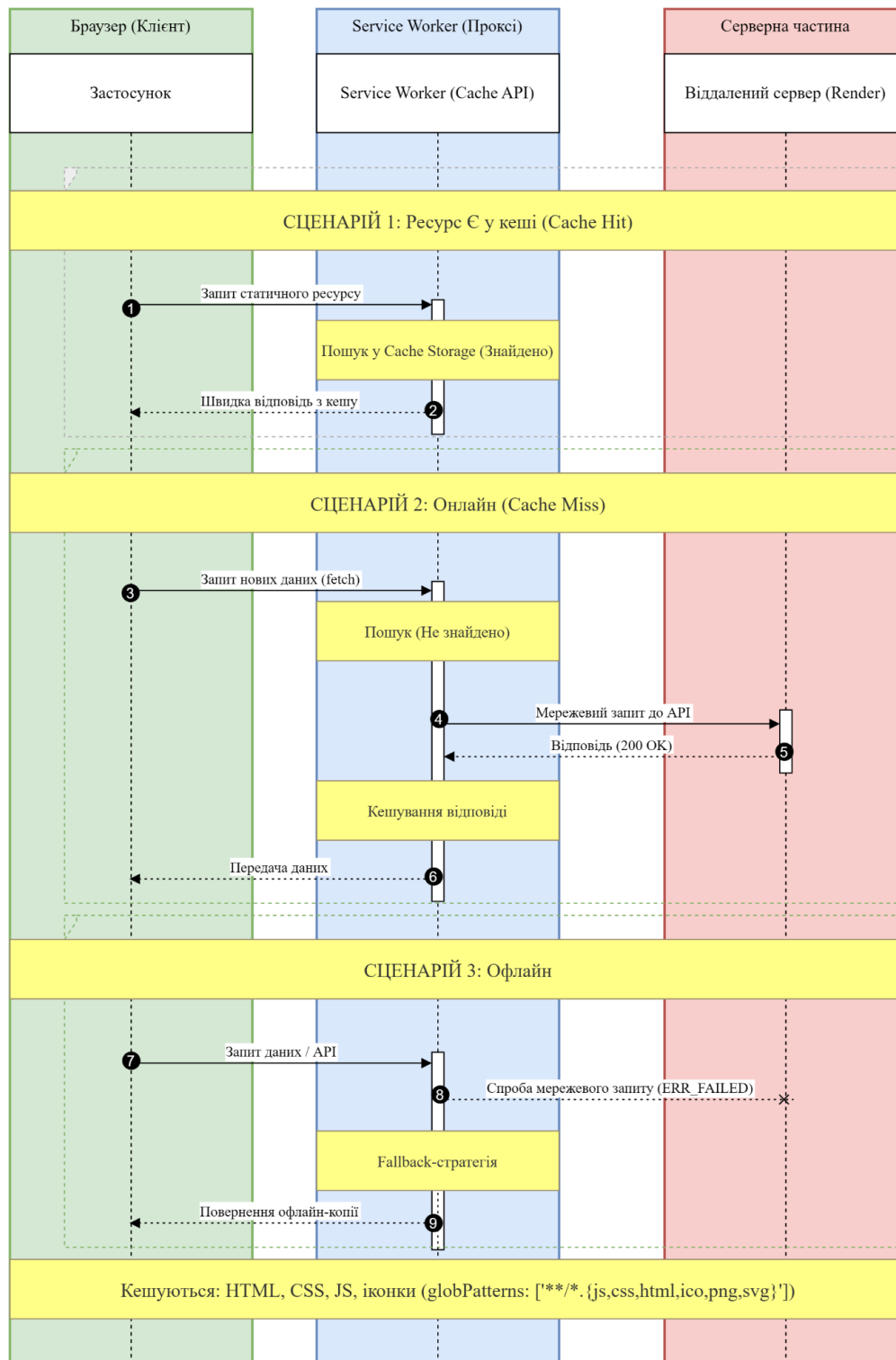


Рисунок 3.4 – Схема роботи Service Worker у режимах онлайн та офлайн

При наступних запусках інтерфейс відкривається з локального кешу незалежно від стану з'єднання. У режимі офлайн застосунок повністю функціонує для перегляду раніше завантажених даних. Річіа-сховище додатково

реалізує власний механізм кешування даних профілю у localStorage під ключем виду user_data_N, де N — числовий ідентифікатор гравця. Навіть якщо сервер недоступний, інтерфейс відображає актуальні показники з останньої успішної синхронізації.

Налаштований файл manifest.json дозволяє браузеру запропонувати користувачу встановити застосунок на головний екран смартфона. У компоненті LoginView.vue реалізовано обробку події beforeinstallprompt: якщо браузер підтримує встановлення, відображається інтерактивна кнопка «Встановити на головний екран». Це забезпечує досвід використання застосунку, ідентичний нативним мобільним програмам, без необхідності публікації в App Store або Google Play, що відповідає стандарту W3C Progressive Web Apps [12, 23].

Висновки до розділу 3

У третьому розділі виконано повну програмну реалізацію інформаційно-аналітичного комплексу у вигляді мікросервісної клієнт-серверної архітектури. Бекенд-частину побудовано на базі асинхронного фреймворку FastAPI з нативною інтеграцією TensorFlow: обчислювальне ядро завантажується в оперативну пам'ять одноразово при запуску сервера, що мінімізує затримку при кожному виклику інференсу. Реалізовано службовий маршрут Keep-Alive /api/ping для підтримки сервера в активному стані на безкоштовному хмарному рівні Render.

Схему реляційної бази даних PostgreSQL спроектовано з двома стратегіями видалення: каскадне (cascade="all, delete-orphan") для залежних таблиць гравця та правило SET NULL для зовнішнього ключа взуття. Це зберігає повноту хронічного навантаження при операціях з інвентарем. Безпеку забезпечено хешуванням паролів алгоритмом bcrypt та JWT-автентифікацією за стандартом HS256 з терміном дії 7 діб.

					ІАЛЦ.045420.004 ПЗ	Арк.
						51
Зм	Лист	№ докум.	Підп.	Дата		

Клієнтський застосунок реалізовано за специфікацією PWA на базі Vue.js 3 із Pinia-сховищем. Service Worker кешує статичний шар при першому завантаженні, а механізм офлайн-кешу у localStorage забезпечує відображення актуальних аналітичних даних навіть за відсутності мережевого з'єднання, що є критичним для умов спортивного залу.

					ІАЛЦ.045420.004 ПЗ	Арк.
						52
Зм	Лист	№ докум.	Підп.	Дата		

РОЗДІЛ 4. ТЕСТУВАННЯ ПРЕДИКТИВНОГО ЯДРА ТА ВЕРИФІКАЦІЯ ФУНКЦІОНАЛЬНИХ МОДУЛІВ СИСТЕМИ

Розділ присвячено експериментальній перевірці обчислювального ядра системи — рекурентної нейромережі LSTM. Оскільки реальні клінічні бази даних про спортивний травматизм є закритими і недоступними для академічних досліджень, для навчання та валідації моделі розроблено алгоритм синтезу репрезентативного датасету, що ґрунтується на фізіологічно обґрунтованих математичних формулах накопичення кумулятивного стресу. Цей підхід відповідає методологічним настановам систематичного огляду Claudino et al. [2]: автори констатують, що конфіденційність медичних даних у спортивній медицині фактично унеможливорює формування відкритих клінічних вибірок належного обсягу, що робить синтетичне моделювання прийнятною альтернативою для дослідницьких цілей.

4.1 Методика генерації синтетичних наборів даних та їх нормалізація

Для навчання предиктивної моделі було синтезовано датасет, що охоплює телеметрію 50 віртуальних спортсменів тривалістю 1000 днів кожен (загалом 50 000 унікальних записів). Генерацію виконано мовою Python із фіксованим початковим станом генератора випадкових чисел (`np.random.seed(42)`) для забезпечення відтворюваності результатів.

Для кожного з 50 гравців попередньо визначено індивідуальні базові фізіологічні показники. Це дозволяє імітувати природну варіативність між різними спортсменами. Базовий рівень нічного сну для кожного гравця визначено рівномірним розподілом у діапазоні від 6.0 до 8.5 годин; базова інтенсивність тренувань (RPE) у діапазоні від 4.0 до 7.0 балів. Щоденні значення генеруються з нормального розподілу навколо цих індивідуальних баз із наступними налаштуваннями:

					ІАЛЦ.045420.004 ПЗ	Арк.
						53
Зм	Лист	№ докум.	Підп.	Дата		

- тривалість нічного сну: нормальний розподіл навколо бази гравця зі стандартним відхиленням 1.5 год; значення відтинаються у межах [2.0; 12.0] годин;
- суб'єктивна оцінка важкості тренування (RPE): нормальний розподіл навколо бази зі стандартним відхиленням 2.0; значення відтинаються у межах [0.0; 10.0] балів;
- тривалість фізичної активності: нормальний розподіл із математичним очікуванням 60 хвилин і стандартним відхиленням 30 хвилин; значення відтинаються у межах [0.0; 150.0] хвилин. Якщо після генерації RPE виявилось нижче за 2.0 балів, день класифікується як відновлювальний: обидва показники примусово обнуляються ($rpe = 0$, $duration = 0$), що відображає реальну практику тренерського процесу.

Центральне місце в підготовці датасету займає формування цільової змінної — єдиного числового показника, який нейромережа повинна навчитися відтворювати за вхідними ознаками. Базовий щоденний ризик розраховується за формулою:

$$R_{raw} = \frac{(RPE \cdot t_{duration})}{(t_{sleep} \cdot 10.0)} \quad (4.1)$$

де R_{raw} — базовий синтетичний показник стресу за поточний день; RPE — суб'єктивна важкість тренування за шкалою Борга; $t_{duration}$ — тривалість тренування у хвилинах; t_{sleep} — тривалість нічного сну у годинах; 10.0 — масштабуючий коефіцієнт для нормалізації значень.

Ця залежність відображає реальний фізіологічний закон: зростання інтенсивності та тривалості навантаження підвищує ризик, тоді як достатній сон виступає компенсаторним фактором.

Оскільки травми опорно-рухового апарату найчастіше виникають унаслідок накопиченої втоми, а не одноразового перевантаження, базовий денний ризик не може слугувати об'єктивним показником. Тому до згенерованого

часового ряду кожного гравця застосовується ковзне середнє (rolling mean) із розміром вікна 7 діб і параметром `min_periods=1`. Останній дозволяє розраховувати середнє вже з першого дня на основі доступної кількості спостережень, що виключає появу порожніх значень на початку ряду без необхідності додаткового заповнення.

Після формування набору даних виконується попередня нормалізація. Алгоритми глибокого навчання сходяться стабільніше, коли всі вхідні ознаки перебувають в одному числовому масштабі. Для цього застосовано клас `MinMaxScaler` з бібліотеки `Scikit-learn`, який лінійно трансформує показники сну, RPE та тривалості тренування у діапазон $[0.0; 1.0]$. Налаштований об'єкт нормалізатора записується у двійковий файл `scaler.pkl`: це гарантує, що під час виробничої експлуатації реальні дані користувачів масштабуватимуться за тими самими математичними параметрами, що й синтетичні дані під час навчання.

На фінальному етапі безперервний часовий ряд трансформується у тривимірний тензор формату $[N \times 7 \times 3]$, де N — кількість навчальних прикладів, 7 — розмір тижневого мікроциклу (`MICROCYCLE_LEN`), а 3 — кількість вхідних ознак: `sleep_hours`, `rpe`, `duration`. Кожен зразок у тензорі є матрицею нормалізованих показників за попередній тиждень, яка відповідає вхідним вимогам шару LSTM.

Загальний конвеєр підготовки синтетичного датасету наведено на рисунку 4.1.

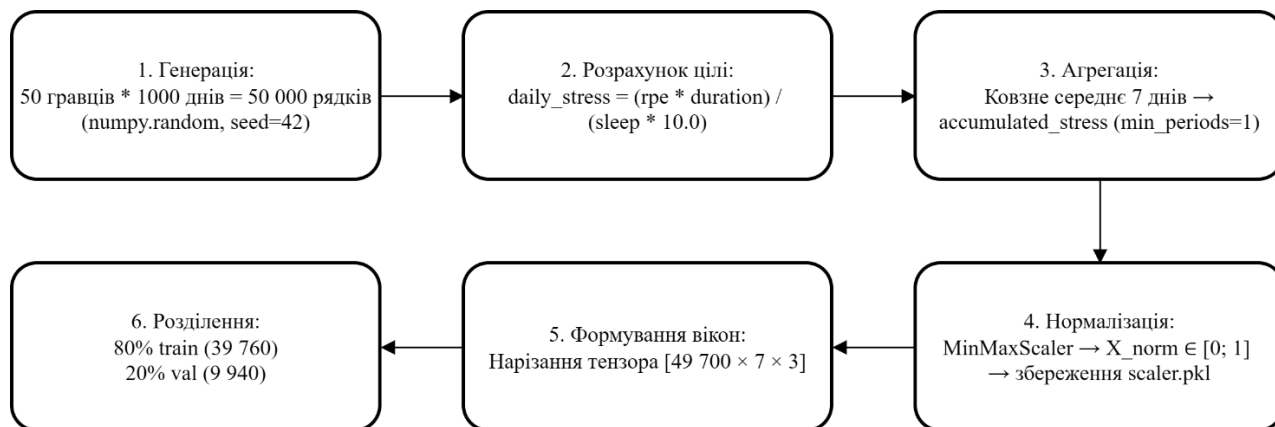


Рисунок 4.1 – Конвеєр підготовки синтетичного датасету для навчання LSTM-моделі

4.2 Навчання та валідація предиктивної моделі глибокого навчання

Після формування тензорів даних розпочато навчання рекурентної нейромережі. Для об'єктивної оцінки якості навченої моделі та виявлення можливого ефекту перенавчання (overfitting) датасет алгоритмічно розділено на дві ізольовані частини: 80 % (39 720 тензорів) відведено для тренування, решта 20 % (9 930 тензорів) виключно для валідації. Модель не має доступу до валідаційної підмножини під час оновлення вагових коефіцієнтів, що дозволяє чесно оцінити її здатність до узагальнення.

Для мінімізації цільової функції обрано оптимізатор Adam зі швидкістю навчання 0.001. Adam поєднує корекцію першого моменту градієнта (імпульс) з адаптивним масштабуванням другого моменту. Завдяки цьому алгоритм стійкий до ситуацій, коли окремі ознаки часового ряду з'являються рідко або мають різний масштаб коливань, що характерно для біометричних показників.

Цільовою функцією втрат (loss function) визначено середньоквадратичну похибку (Mean Squared Error, MSE). Вибір MSE обґрунтовано тим, що вона квадратично штрафує модель за великі відхилення: у медичному контексті критична помилка при класифікації зони ризику є більш небезпечною, ніж незначний систематичний зсув, тому суворіше пенальті за екстремальні

відхилення є доцільним. Як допоміжна метрика для моніторингу обрано середню абсолютну похибку (Mean Absolute Error, MAE), яка дає безпосередньо інтерпретовану лінійну оцінку відхилення у нормалізованому масштабі ризику.

Відповідну динаміку середньої абсолютної похибки протягом навчання наведено на рисунку 4.2.

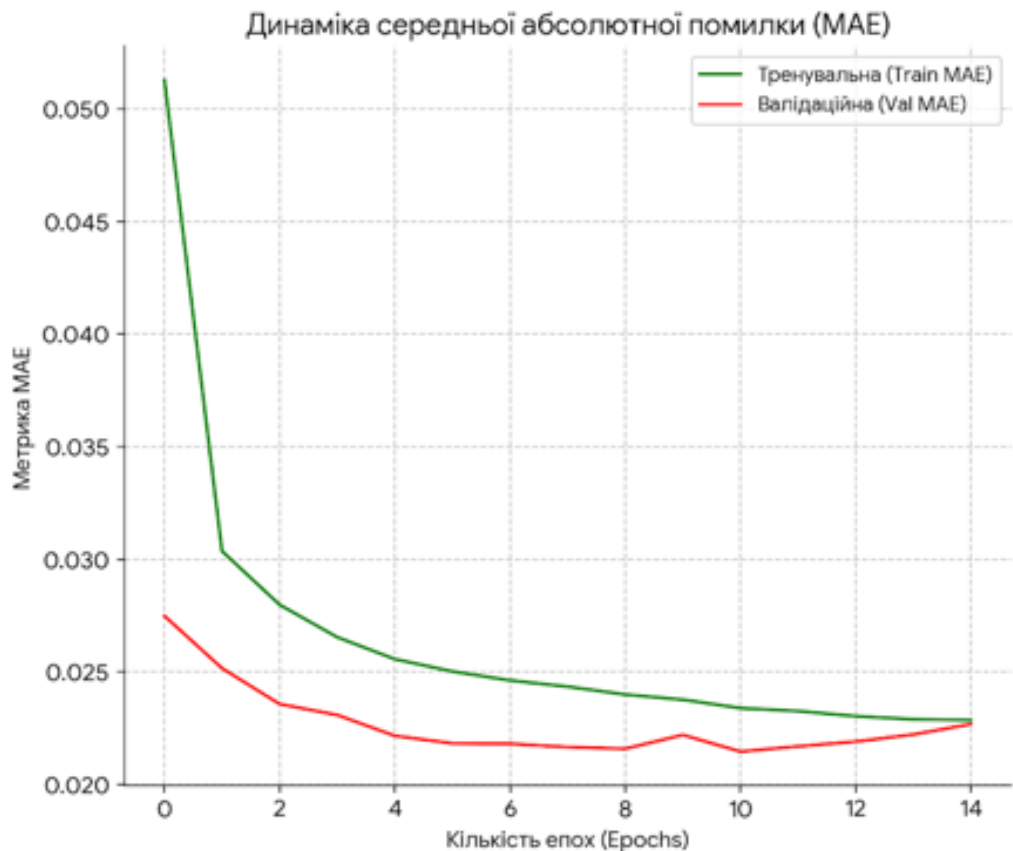


Рисунок 4.2 – Динаміка MAE на тренувальній і валідаційній вибірках протягом 15 епох навчання

Навчання тривало 15 повних епох. Дані подавалися до вхідного шару пакетами по 64 зразки (`batch_size=64`): оновлення вагових коефіцієнтів за алгоритмом зворотного поширення похибки (`backpropagation`) відбувалося після обробки кожного такого пакета, а не після кожного окремого зразка. Це забезпечило стабільний градієнт і зменшило обчислювальне навантаження на матричні операції. Після завершення останньої епохи оптимізовану архітектуру

разом із навченими ваговими коефіцієнтами збережено у бінарному форматі `basketball_lstm.keras`.

4.3 Аналіз прогностичної здатності та оцінка похибки алгоритму

Для об'єктивної оцінки якості навченої моделі проаналізовано динаміку метрик похибки на обох підмножинах упродовж усіх 15 епох.

Динаміку зміни функцій втрат у процесі навчання наведено на рисунку 4.3.

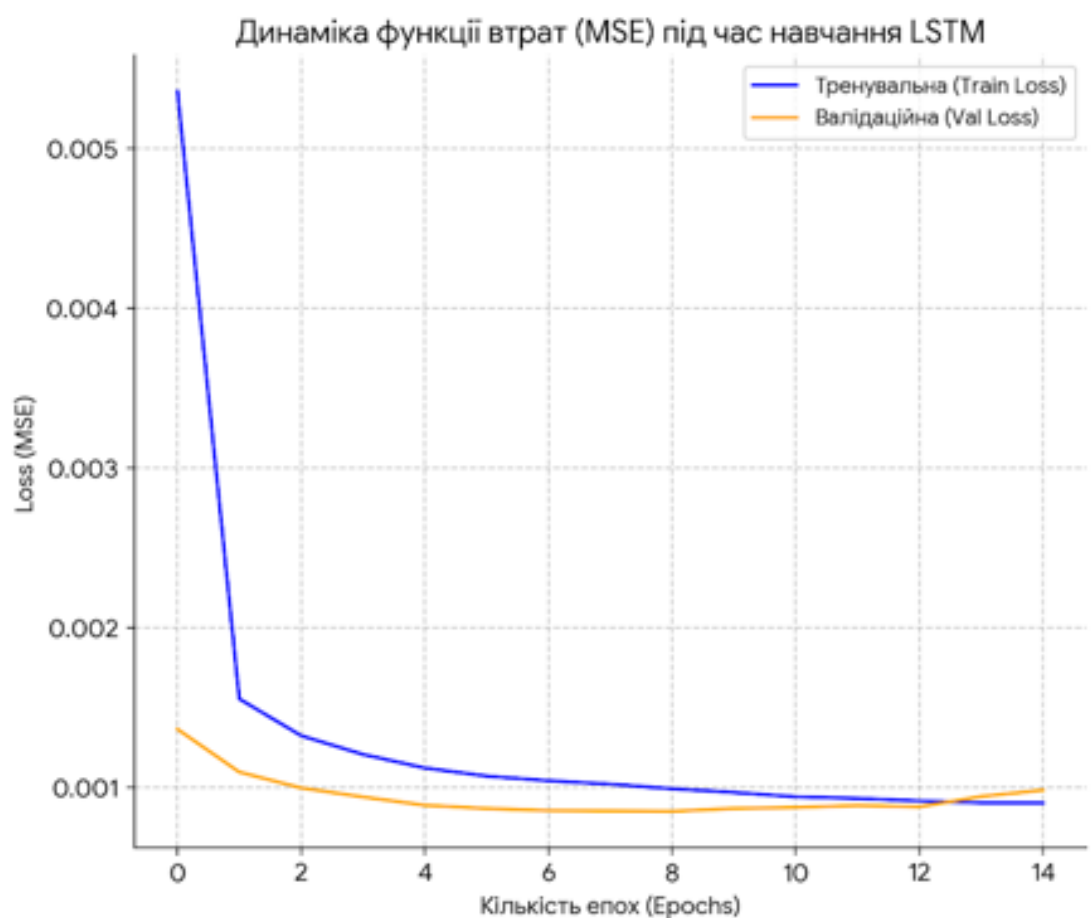


Рисунок 4.3 – Динаміка MSE на тренувальній і валідаційній вибірках протягом 15 епох навчання

Аналіз кривих показав типову картину стабільного збіжного навчання: на початкових епохах спостерігається інтенсивне падіння MSE як на тренувальній,

так і на валідаційній вибірках. Мережа стрімко виявляє головні математичні закономірності у тензорах, зокрема зворотну пропорційність ризику до тривалості сну і пряму до RPE. У міру наближення до 15-ї епохи обидві криві поступово виходять на плато: темп зниження похибки суттєво знижується, що вказує на вичерпання навчальних можливостей поточної архітектури на наявному обсязі синтетичних даних.

Ключовим діагностичним критерієм того, що модель не «запам'ятала» синтетичні профілі, а дійсно узагальнила закономірності, є поведінка валідаційної кривої. Відсутність характерного розходження кривих, коли тренувальна похибка продовжує спадати, тоді як валідаційна починає зростати є основним індикатором відсутності перенавчання (overfitting). Це означає, що нейромережа не просто відтворила статистику 50 конкретних синтетичних профілів, а узагальнила фізіологічну залежність між показниками навантаження та кумулятивним ризиком.

Діаграму відповідності прогнозованих та еталонних значень на валідаційній вибірці наведено на рисунку 4.4.

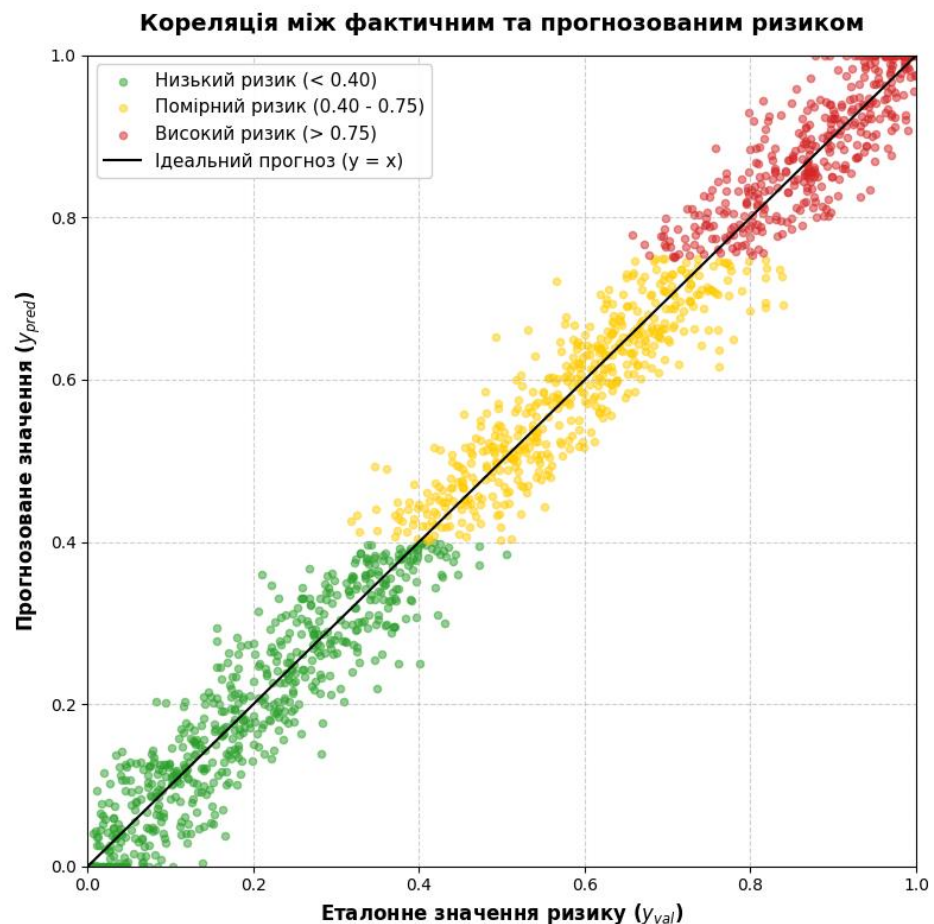


Рисунок 4.4 – Порівняння прогнозованих і еталонних значень ризику на валідаційній вибірці

Фінальне значення середньої абсолютної похибки (MAE) на валідаційній вибірці після 15-ї епохи становило 0.0227. У контексті задачі системи це означає, що відхилення прогнозованого ризику від еталонного значення у середньому становить 2.27% у нормалізованому масштабі. Такий рівень похибки задовольняє головну практичну вимогу до предиктивного модуля: модель стабільно розмежовує три зони ризику («Optimal», «Moderate Risk», «High Danger»), а не намагається вирахувати абсолютно точне числове значення. Для прескриптивної аналітики точність класифікації зони є набагато важливішою за мінімізацію числової похибки до тисячних.

4.4 Верифікація REST API та функціональних модулів системи

Для підтвердження коректності роботи бекенд-архітектури виконано ручне функціональне тестування програмного інтерфейсу через автоматично згенерований Swagger UI (доступний за адресою /docs). Перевірено роботу всіх чотирьох логічних груп маршрутів.

Для ендпоінтів управління профілем (/api/users/register, /api/users/login, /api/users/{user_id}) перевірено: коректне хешування пароля при реєстрації, повернення JWT-токена тривалістю 7 діб, захист маршруту від несанкціонованого доступу (відповідь HTTP 401 при відсутньому або простроченому токені).

Для ендпоінту телеметрії (POST /api/users/{user_id}/metrics) підтверджено роботу @model_validator: при передачі типу активності «Recovery» поля duration_minutes та rpe_score примусово обнуляються незалежно від значень у запиті, що виключає аномалії у вхідному тензорі нейромережі.

Модель деградації взуття перевірено сценарним методом: для п'яти тестових комбінацій (маса гравця: 70 кг і 100 кг; покриття: паркет, асфальт, тартан) розраховані системою значення max_lifespan_hours порівняно з аналітичними результатами за формулою (2.1). Розбіжностей між фактичними та очікуваними значеннями не виявлено, що підтверджує коректність програмної реалізації математичної моделі.

Висновки до розділу 4

У четвертому розділі проведено комплексне тестування системи на двох рівнях. На рівні предиктивного ядра розроблено алгоритм генерації синтетичного датасету (50 000 записів, 50 гравців, 1000 днів), що відтворює реалістичну варіативність фізіологічних показників. Навчання LSTM-моделі тривало 15 епох із оптимізатором Adam ($\text{lr} = 0.001$), цільовою функцією MSE та

					ІАЛЦ.045420.004 ПЗ	Арк.
						61
Зм	Лист	№ докум.	Підп.	Дата		

метрикою MAE при розподілі 80/20. Аналіз кривих підтвердив збіжність та відсутність перенавчання.

На рівні інтеграції виконано ручну верифікацію REST API через середовище Swagger UI: перевірено коректність відповідей для всіх ключових маршрутів (/api/users/{user_id}/metrics, /api/ai/generate_plan/{user_id}, /api/shoes/{shoe_id}, /api/users/register), у тому числі сценарії із некоректними токенами (повернення HTTP 401) та пошкодженими даними (повернення HTTP 422 від Pydantic). Модель деградації взуття перевірено сценарним методом: для п'яти комбінацій покриття/маси розраховані значення Emax порівняно з очікуваними аналітичними результатами (розбіжностей не виявлено).

Отримані результати підтверджують коректність усіх математичних моделей, описаних у Розділі 2, та готовність розробленого комплексу до практичного застосування.

					ІАЛЦ.045420.004 ПЗ	Арк.
						62
Зм	Лист	№ докум.	Підп.	Дата		

ВИСНОВКИ

У дипломному проєкті вирішено актуальну науково-прикладну задачу — розроблено інформаційно-аналітичний комплекс превенції спортивних травм на основі технологій глибокого машинного навчання. Аналіз проблематики спортивного травматизму показав, що понад 65% безконтактних пошкоджень у баскетболі є наслідком некерованого накопичення кумулятивної втоми, а не одноразового перевантаження. Це обґрунтовує необхідність створення системи, що поєднує концепцію ACWR із предиктивними алгоритмами машинного навчання.

Доведено, що архітектура рекурентної нейромережі LSTM є найбільш обґрунтованим вибором для моделювання фізіологічних часових рядів: вентильна структура клітини стану долає ефект згасання градієнта і дозволяє мережі зберігати контекст накопиченого стресу в межах повного тренувального мікроциклу.

Розроблено математичне та алгоритмічне забезпечення системи. Сформовано нелінійну мультиплікативну модель предиктивної оцінки деградації спортивного взуття, що адаптивно розраховує безпечний ресурс амортизаційного матеріалу з урахуванням маси тіла спортсмена, типу ігрового покриття та функціонального призначення взуття, причому коефіцієнт маси обмежено діапазоном $[0.75; 1.25]$ для захисту від аномальних вхідних значень.

Запропоновано метод постобчислювального коригування прогнозу нейромережі через об'єктивний показник варіабельності серцевого ритму (HRV, параметр $rMSSD$): гібридний модифікатор динамічно масштабує базовий прогноз LSTM у діапазоні $[0.75; 1.25]$ залежно від відхилення HRV від еталону 55 мс. Для модуля прескриптивної аналітики реалізовано стохастичний генератор на основі методу Фішера-Сйтса (`random.sample()`), що автоматично формує унікальні адаптивні тренувальні протоколи для кожної з п'яти ігрових позицій залежно від розрахованої зони ризику.

					ІАЛЦ.045420.004 ПЗ	Арк.
						63
Зм	Лист	№ докум.	Підп.	Дата		

Виконано програмну реалізацію комплексу у вигляді мікросервісної клієнт-серверної архітектури. Серверну частину побудовано на базі асинхронного фреймворку FastAPI з нативною інтеграцією TensorFlow: обчислювальне ядро (basketball_lstm.keras + scaler.pkl) завантажується в оперативну пам'ять одноразово при ініціалізації сервера, що мінімізує затримку під час інференсу. Схему реляційної бази даних PostgreSQL спроектовано з двома стратегіями видалення: каскадне видалення (cascade="all, delete-orphan") для залежних таблиць гравця та правило SET NULL для зовнішнього ключа взуття в таблиці телеметрії. Це зберігає повноту хронічного навантаження при інвентарних операціях.

Безпеку даних забезпечено хешуванням паролів алгоритмом bcrypt та автентифікацією через JWT-токени стандарту HS256 із терміном дії 7 діб.

Клієнтський інтерфейс реалізовано на базі Vue.js 3 за специфікацією PWA: сервіс-воркери та Pinia-кеш у localStorage гарантують автономну роботу застосунку за відсутності мережевого з'єднання, що є критично важливим для цільового середовища використання (спортивних залів і майданчиків).

Проведено експериментальні дослідження аналітичного ядра. Розроблений алгоритм синтезу датасету відтворив 50 000 записів щоденної телеметрії для 50 віртуальних спортсменів із фізіологічно обґрунтованими діапазонами показників і моделюванням відновлювальних днів. Навчання предиктивної моделі тривало 15 епох із оптимізатором Adam (learning rate 0.001), цільовою функцією MSE і контрольною метрикою MAE при розділенні вибірки 80/20. Аналіз кривих втрат підтвердив збіжність алгоритму та відсутність ефекту перенавчання: паралельна динаміка тренувальної та валідаційної похибок свідчить про здатність моделі узагальнювати фізіологічні закономірності накопичення кумулятивного стресу.

Систему BAIPS верифіковано на синтетичному датасеті (MAE = 0,0227) і впроваджено в навчально-тренувальний процес гімназії «Кларіс Вербіс», що підтверджує її практичну цінність для контролю тренувального навантаження в аматорському та напівпрофесійному баскетболі.

					ІАЛЦ.045420.004 ПЗ	Арк.
						64
Зм	Лист	№ докум.	Підп.	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Gabbett T. J. The training — injury prevention paradox: should athletes be training smarter and harder? *British Journal of Sports Medicine*. 2016. Vol. 50, № 5. P. 273–280.
2. Claudino J. G., Capanema D. O., de Souza T. V., Serrão J. C., Machado Pereira A. C., Nassis G. P. Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review. *Sports Medicine-Open*. 2019. Vol. 5, № 28. P. 1–12.
3. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997. Vol. 9, № 8. P. 1735–1780.
4. Foster C., Florhaug J. A., Franklin J., Gottschall L., Hrovatin L. A., Parker S., Dodge C. A new approach to monitoring exercise training. *The Journal of Strength & Conditioning Research*. 2001. Vol. 15, № 1. P. 109–115.
5. Verdejo R., Mills N. J. Heel–shoe interactions and the durability of EVA foam running-shoe midsoles. *Journal of Biomechanics*. 2004. Vol. 37, № 9. P. 1379–1386.
6. Drakos M. C., Domb B., Starkey C., Callahan L., Allen A. A. Injury in the National Basketball Association: a 17-year overview. *Sports Health*. 2010. Vol. 2, № 4. P. 284–290.
7. Shaffer F., Ginsberg J. P. An overview of heart rate variability metrics and norms. *Frontiers in Public Health*. 2017. Vol. 5. Article 258. P. 1–17.
8. Abadi M. et al. TensorFlow: A System for Large-Scale Machine Learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, 2016. P. 265–283.
9. Black P. E. Fisher-Yates shuffle. *Dictionary of Algorithms and Data Structures*. National Institute of Standards and Technology. 2004. URL: <https://www.nist.gov/dads/HTML/fisherYatesShuffle.html> (дата звернення: 10.04.2026).

Зм	Лист	№ докум.	Підп.	Дата

ІАЛЦ.045420.004 ПЗ

Арк.
65

10. Ramírez S. FastAPI framework: high performance, easy to learn, fast to code, ready for production. 2018. URL: <https://fastapi.tiangolo.com> (дата звернення: 15.03.2026).
11. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. 2023. URL: <https://www.postgresql.org/docs/> (дата звернення: 10.03.2026).
12. You E. Vue.js 3: The Progressive JavaScript Framework: official documentation. 2023. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 20.03.2026).
13. Zone7 AI. AI-driven Injury Prevention Platform: official website. URL: <https://zone7.ai> (дата звернення: 05.04.2026).
14. Catapult Sports. Wearable Technology & Performance Analytics. URL: <https://www.catapultsports.com> (дата звернення: 05.04.2026).
15. Kitman Labs. The Intelligence Platform for High Performance Sport. URL: <https://www.kitmanlabs.com> (дата звернення: 05.04.2026).
16. Hulin B. T., Gabbett T. J., Lawson D. W., Caputi P., Sampson J. A. The acute:chronic workload ratio predicts injury: high chronic workload may decrease injury risk in elite rugby league players. *British Journal of Sports Medicine*. 2016. Vol. 50, № 4. P. 231–236.
17. Blanch P., Gabbett T. J. Has the athlete trained enough to return to play safely? The acute:chronic workload ratio permits clinicians to quantify a player's risk of subsequent injury. *British Journal of Sports Medicine*. 2016. Vol. 50, № 8. P. 471–475.
18. Cho K., van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014. P. 1724–1734.
19. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 775 p.

20. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Duchesnay E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
21. Williams S., Booton T., Watson M., Rowland D., Altini M. Heart rate variability is a moderating factor in the workload-injury relationship of competitive CrossFit athletes. *Journal of Sports Science & Medicine*. 2017. Vol. 16, № 4. P. 443–449.
22. Bittencourt N. F. N., Meeuwisse W. H., Mendonça L. D., Nettel-Aguirre A., Ocarino J. M., Fonseca S. T. Complex systems approach for sports injuries: moving from risk factor identification to injury pattern recognition. *British Journal of Sports Medicine*. 2016. Vol. 50, № 21. P. 1309–1314.
23. Google LLC. Learn Progressive Web Apps: official developer guide. 2024. URL: <https://web.dev/learn/pwa> (дата звернення: 12.04.2026).
24. Malone S., Hughes B., Doran D. A., Collins K., Gabbett T. J. Can the workload-injury relationship be modelled in elite Gaelic football? *Journal of Science and Medicine in Sport*. 2019. Vol. 22, № 2. P. 192–198.
25. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill, 2019. 976 p.
26. Домуші Д. Д., Потапова К. Р., Кучмій О. О. Застосування рекурентних нейронних мереж для оцінювання кумулятивної втоми та травматичних ризиків атлетів. *European science and innovation congress : Proceedings of VII International Scientific and Practical Conference* (Barcelona, Spain, June 1–3, 2026). Barcelona : Barca Academy Publishing, 2026. С. 150–154.

ДОДАТОК А

Інформаційно-аналітична система превенції спортивних травм на основі
рекурентної нейромережі LSTM та концепції ACWR

Копії графічного матеріалу

ІАЛЦ.045420.005 Д1.

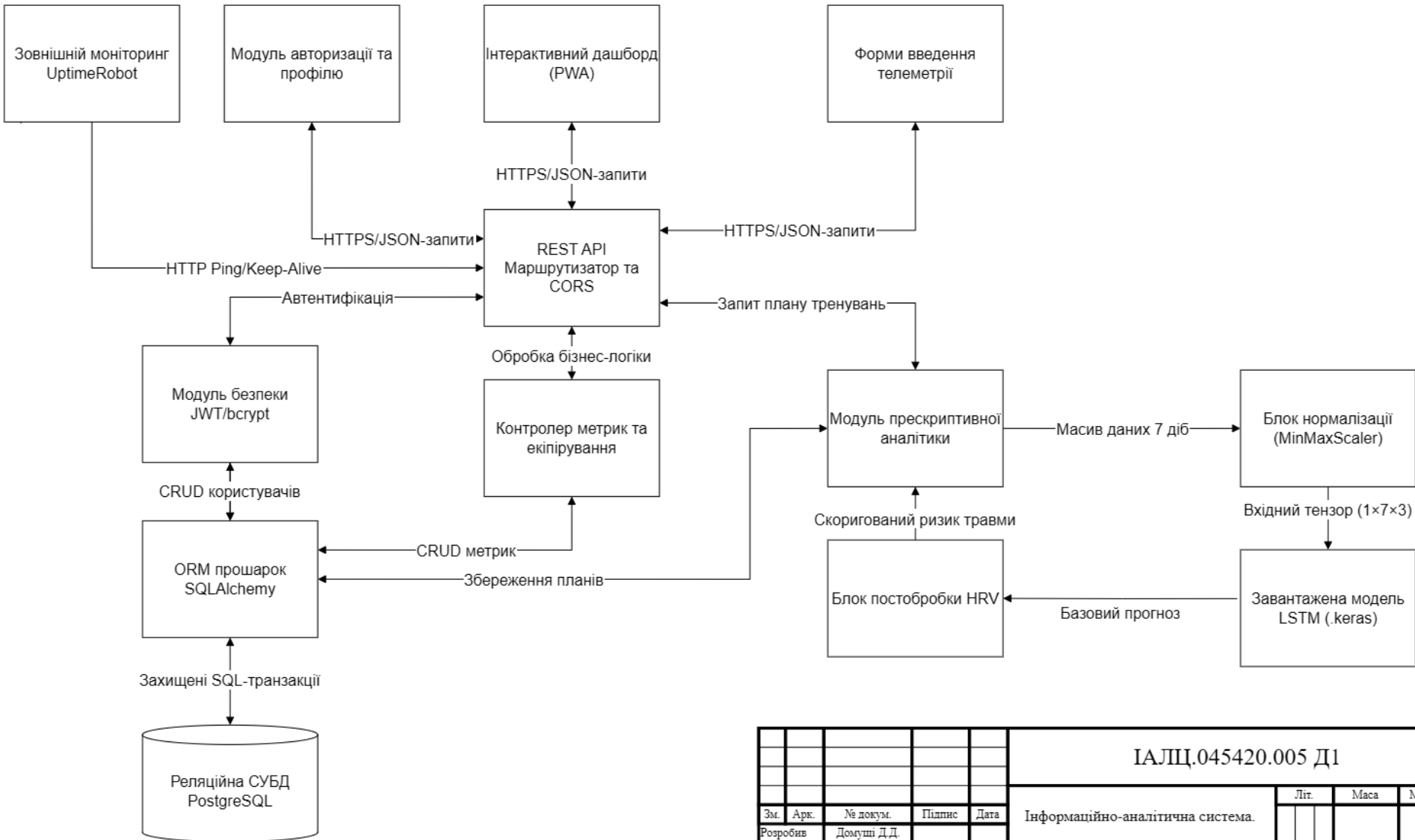
ІАЛЦ.045420.006 Д2.

ІАЛЦ.045420.007 Д3.

ІАЛЦ.045420.008 Д4.

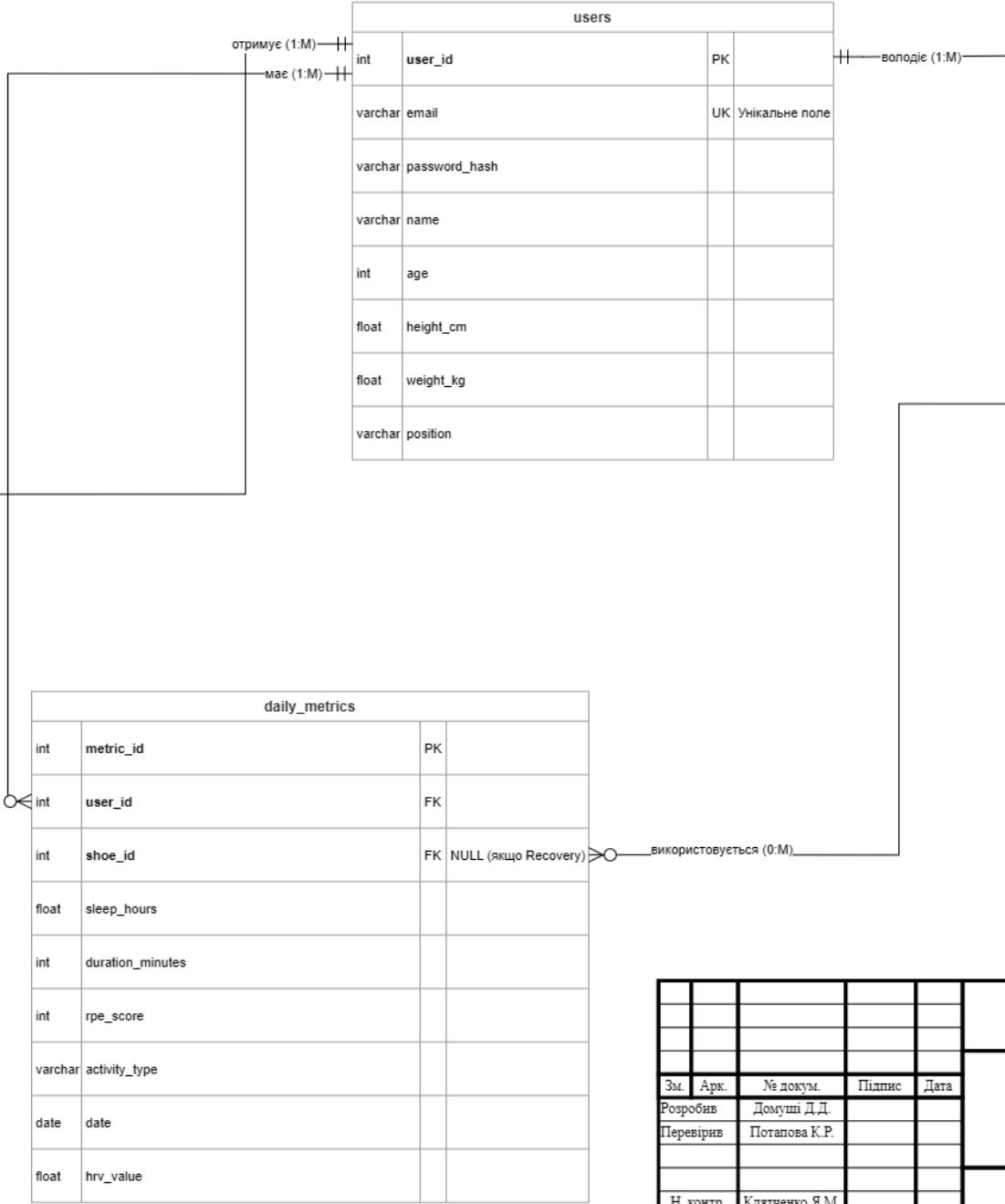
Аркушів 4

Київ 2026

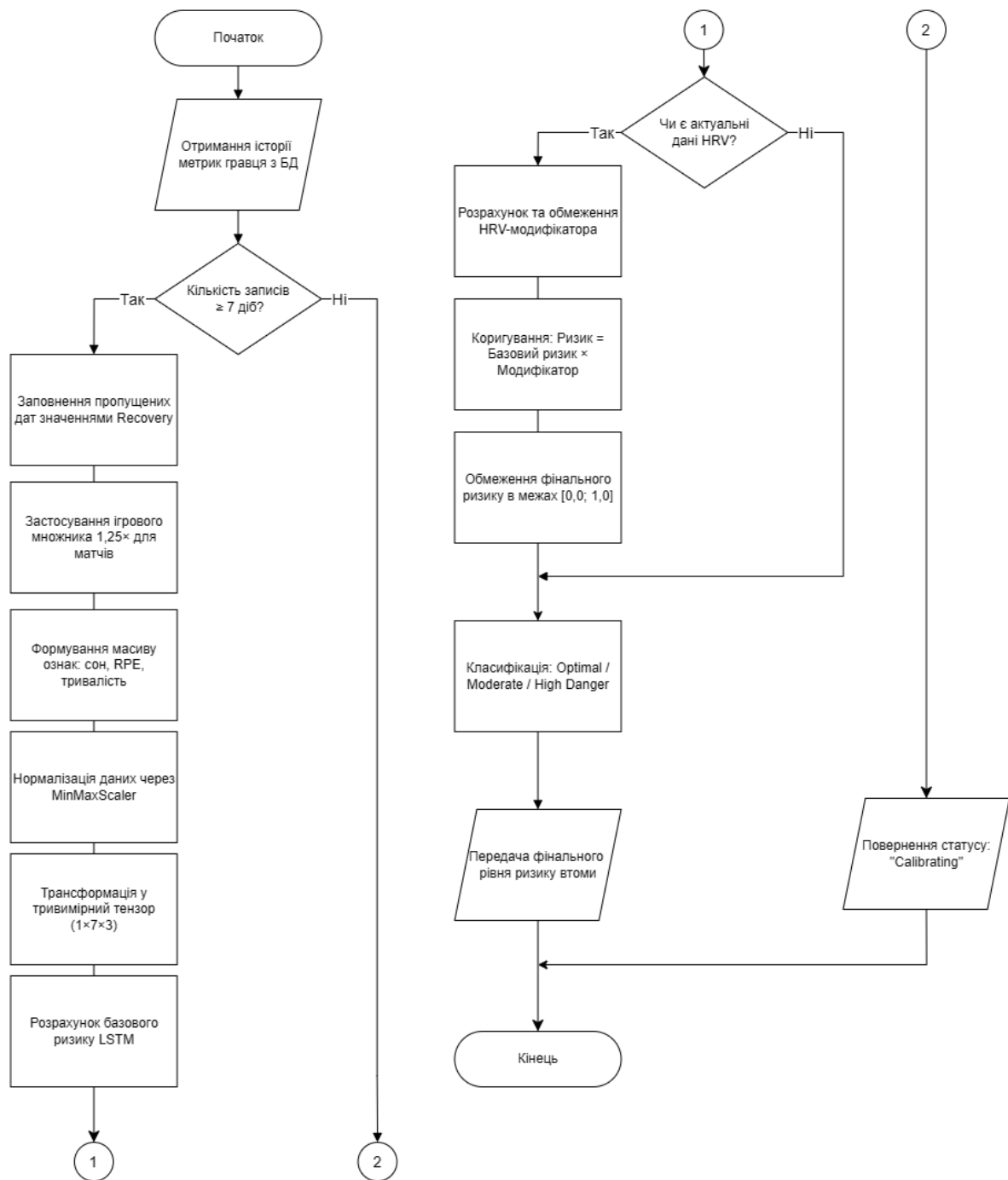


					ІАЛЦ.045420.005 Д1				
					Інформаційно-аналітична система. Схема структурна.	Літ.	Маса	Масштаб	
Зм.	Арк.	№ докум.	Підпис	Дата					
Розробив		Домуші Д.Д.							
Перевірів		Потапова К.Р.							
						Арк. 1		Аркушів 1	
Н. контр.		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФІСПМ KB-23			
Затв.		Романкевич В.О.							

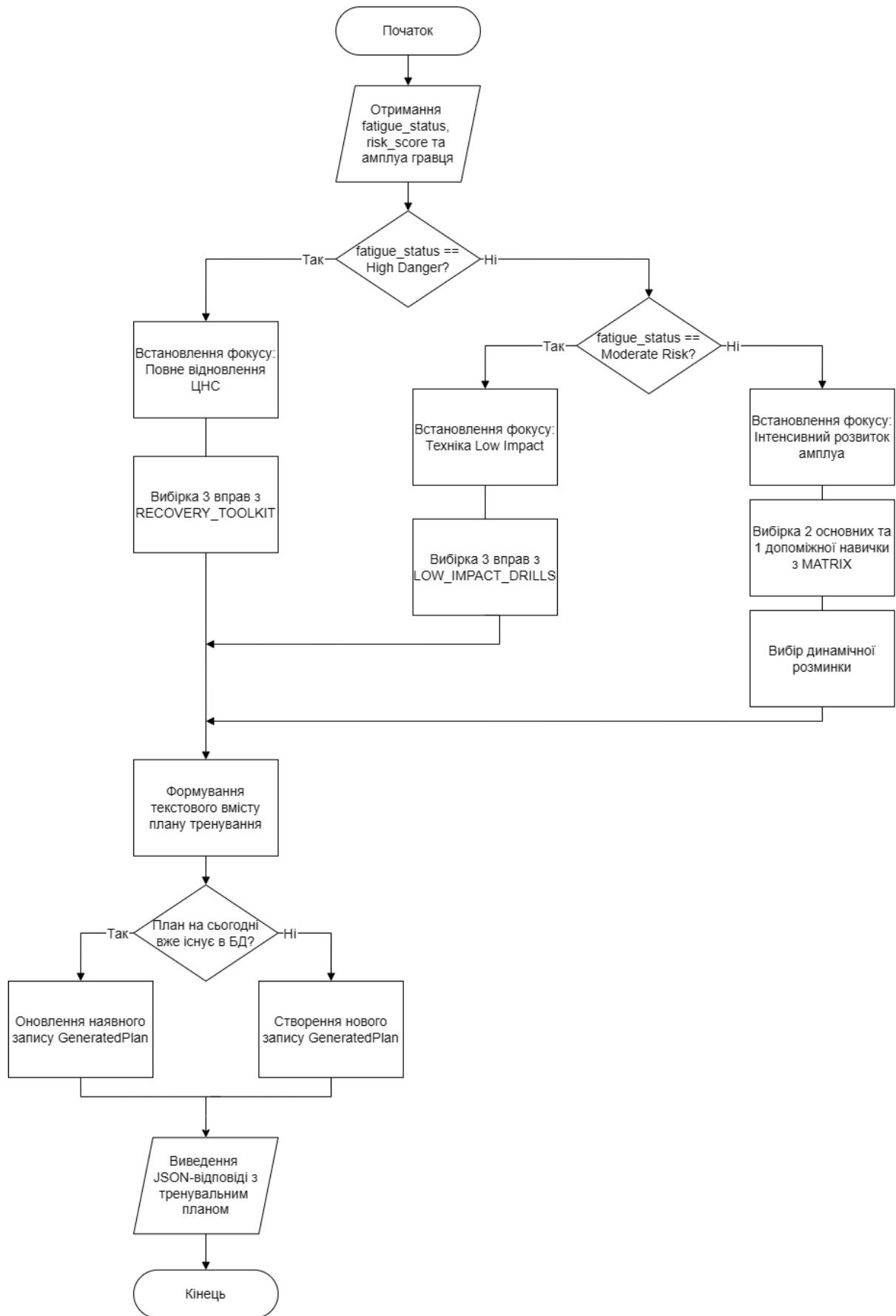
generated_plans		
int	plan_id	PK
int	user_id	FK
date	date	
varchar	fatigue_risk	
varchar	plan_focus	
text	plan_content	



ІАЛЦ.045420.006 Д2						
Реляційна база даних. Схема структурна.				Літ.	Маса	Масштаб
				Арх.		Архувів
Н. контр.				Клятченко Я.М.		КПІ ім. Ігоря Сікорського, ФІСПМ КВ-23
Затв.				Романкевич В.О.		



					ІАЛЦ.045420.007 ДЗ				
					Алгоритм прогнозування кумулятивної втоми. Схема алгоритму.	Літ.		Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата					
Розробив		Домуш Д.Д.							
Перевірів		Потапова К.Р.							
						Арк.		Аркушів	
Н. контр.		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23			
Затв.		Романкевич В.О.							



Зм.	Арк.	№ докум.	Підпис	Дата	
Розробив		Домуші Д.Д.			
Перевірів		Потапова К.Р.			
Н. контр.		Клятченко Я.М.			
Затв.		Романкевич В.О.			

ІАЛЦ.045420.008 Д4

Реляційна база даних.

Схема структурна.

Літ.	Маса	Масштаб
Арк.	Аркушів	
КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23		

ДОДАТОК Б

Інформаційно-аналітична система превенції спортивних травм на основі
рекурентної нейромережі LSTM та концепції ACWR

Фрагменти програмного коду

Аркушів

Київ 2026

train_lstm.py

```
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler
import pickle

print("Генерація даних...")
np.random.seed(42)
num_players = 50
days_per_player = 1000
data = []

for player_id in range(num_players):
    base_sleep = np.random.uniform(6.0, 8.5)
    base_rpe = np.random.uniform(4.0, 7.0)

    for day in range(days_per_player):
        sleep = np.clip(np.random.normal(base_sleep, 1.5), 2.0, 12.0)
        rpe = np.clip(np.random.normal(base_rpe, 2.0), 0.0, 10.0)
        duration = np.clip(np.random.normal(60, 30), 0.0, 150.0)

        if rpe < 2.0:
            duration = 0.0
            rpe = 0.0

        data.append([player_id, sleep, rpe, duration])

df = pd.DataFrame(data, columns=['player_id', 'sleep_hours', 'rpe', 'duration'])

df['daily_stress'] = (df['rpe'] * df['duration']) / (df['sleep_hours'] * 10)
df['accumulated_stress'] =
df.groupby('player_id')['daily_stress'].transform(lambda x: x.rolling(7,
min_periods=1).mean())

target_scaler = MinMaxScaler()
df['fatigue_risk'] = target_scaler.fit_transform(df[['accumulated_stress']])

print("Формування вікон мікроциклу...")
MICROCYCLE_LEN = 7

feature_scaler = MinMaxScaler()
df[['sleep_hours', 'rpe', 'duration']] =
feature_scaler.fit_transform(df[['sleep_hours', 'rpe', 'duration']])

with open('scaler.pkl', 'wb') as f:
    pickle.dump(feature_scaler, f)

daily_windows: list[np.ndarray] = []
fatigue_targets: list[float] = []

player_ids = df['player_id'].unique()

for pid in player_ids:
    athlete_slice = df.loc[df['player_id'] == pid]
    bio_signals = athlete_slice[['sleep_hours', 'rpe', 'duration']].to_numpy()
    accumulated_fatigue = athlete_slice['fatigue_risk'].to_numpy()

    horizon = len(bio_signals) - MICROCYCLE_LEN
    for day_offset in range(horizon):
        window_end = day_offset + MICROCYCLE_LEN
        daily_windows.append(bio_signals[day_offset:window_end])
        fatigue_targets.append(accumulated_fatigue[window_end])
```



```

feature_tensor = np.asarray(daily_windows, dtype=np.float32)
target_vector = np.asarray(fatigue_targets, dtype=np.float32)

print(f"Формат тензора ознак: {feature_tensor.shape}")
print("Створення архітектури нейромережі...")

def _build_fatigue_predictor(window_size: int, n_features: int) ->
tf.keras.Model:
    inputs = tf.keras.Input(shape=(window_size, n_features),
name="microcycle_input")
        recurrent_out = tf.keras.layers.LSTM(
            units=32, activation='relu', name="lstm_temporal_encoder"
        )(inputs)

        regularized = tf.keras.layers.Dropout(rate=0.20,
name="dropout_regularizer")(recurrent_out)

        hidden = tf.keras.layers.Dense(
            units=16, activation='relu', name="nonlinear_projector"
        )(regularized)

        risk_output = tf.keras.layers.Dense(
            units=1, activation='sigmoid', name="fatigue_risk_output"
        )(hidden)

        network = tf.keras.Model(inputs=inputs, outputs=risk_output,
name="BasketballFatigueNet")
        network.compile(
            optimizer=tf.keras.optimizers.Adam(learning_rate=1e-3),
            loss='mse',
            metrics=[tf.keras.metrics.MeanAbsoluteError(name='mae')]
        )
        return network

predictor_model = _build_fatigue_predictor(MICROCYCLE_LEN, n_features=3)

print("Починаємо тренування моделі...")
history = predictor_model.fit(
    feature_tensor, target_vector,
    epochs=15, batch_size=64, validation_split=0.2
)

predictor_model.save('basketball_lstm.keras')
print("Готово! Модель 'basketball_lstm.keras' успішно збережена.")

```

main.py

```

@app.post("/api/ai/generate_plan/{user_id}",
response_model=schemas.PlanResponse)
def generate_plan(
    user_id: int,
    db: Session = Depends(get_db),
    current_user: models.User = Depends(get_current_user)
):
    if current_user.user_id != user_id:
        raise HTTPException(status_code=403, detail="Доступ заборонено")

    user = db.query(models.User).filter(models.User.user_id == user_id).first()
    if not user:
        raise HTTPException(status_code=404, detail="Гравця не знайдено")

    today = date.today()
    existing_plan = db.query(models.GeneratedPlan).filter(

```

```

        models.GeneratedPlan.user_id == user_id,
        models.GeneratedPlan.date == today
    ).first()
metrics = sorted(user.metrics, key=lambda x: x.date)

if len(metrics) < 7:
    return schemas.PlanResponse(
        plan_id=0,
        date=today,
        fatigue_risk="Optimal",
        plan_focus="Збір даних (Калібрування)",
        plan_content=f"Для роботи нейромережі LSTM потрібно 7 днів історії.  
Зараз у вас {len(metrics)}/7 днів.\nПродовжуйте вносити дані!"
    )

last_date = metrics[-1].date
metrics_dict = {m.date: m for m in metrics}
last_7_days = []

for i in range(6, -1, -1):
    target_date = last_date - timedelta(days=i)
    if target_date in metrics_dict:
        last_7_days.append(metrics_dict[target_date])
    else:
        empty_metric = models.DailyMetric(sleep_hours=8.0, rpe_score=0,
duration_minutes=0, activity_type="Recovery")
        last_7_days.append(empty_metric)

input_features = []
for m in last_7_days:
    game_multiplier = 1.25 if m.activity_type == "Game" else 1.0
    effective_rpe = min(10.0, m.rpe_score * game_multiplier)
    effective_duration = m.duration_minutes * game_multiplier
    input_features.append([m.sleep_hours, effective_rpe,
effective_duration])

input_array = np.array(input_features)

if model is not None and scaler is not None:
    scaled_input = scaler.transform(input_array)
    final_input = scaled_input.reshape(1, 7, 3)
    prediction = model.predict(final_input, verbose=0)
    risk_score = float(prediction[0][0])
else:
    risk_score = 0.5
    latest_hrv = next((m.hrv_value for m in reversed(last_7_days) if m.hrv_value
is not None and m.hrv_value > 0), None)
    if latest_hrv:
        raw_risk_mod = 1.0 - (latest_hrv - 55) * 0.008
        risk_modifier = max(0.75, min(1.25, raw_risk_mod))
        risk_score *= risk_modifier

risk_score = max(0.0, min(risk_score, 1.0))

if risk_score > 0.75:
    fatigue_status = "High Danger"
elif risk_score > 0.40:
    fatigue_status = "Moderate Risk"
else:
    fatigue_status = "Optimal"

# Бібліотека знань III-Тренера
RECOVERY_TOOLKIT = [
    "Контрастний душ (3х45 сек)",

```

```

        "Міофасціальний реліз (МФР) квадрицепсів",
        "Йога для баскетболістів (15 хв)",
        "Суглобова гімнастика Бубновського",
        "Дихальна техніка 4-7-8",
        "Прогулянка на свіжому повітрі (20 хв)",
        "Масажний пістолет (м'язи гомілки)"
    ]

    LOW_IMPACT_DRILLS = [
        "Штрафні кидки (5 серій по 10)",
        "Дриблінг на місці (слабка рука, 10 хв)",
        "Form shooting (кидки однією рукою з-під кільця)",
        "Робота ніг без м'яча на низькій швидкості",
        "Аналіз відео (вивчення плейбуку або розбір ігор)",
        "Кидки з місця без стрибків (Catch and shoot)"
    ]

    POSITION_DRILL_MATRIX = {
        "PG": {"primary": ["Speed dribbling з конусами", "Floater mechanics",
            "Pick-and-roll read"],
            "secondary": ["Court vision drills", "Double crossover"]},
        "SG": {"primary": ["Catch-and-shoot off screens", "ISO mid-range", "One-
            dribble pull-up"],
            "secondary": ["Transition 3s", "Corner 3pt precision"]},
        "SF": {"primary": ["Wing slash-and-kick", "Fadeaway mid-range", "Wing 1-
            on-1 attacks"],
            "secondary": ["Defensive close-out slides", "Rebound-and-
            outlet"]},
        "PF": {"primary": ["Post-up power moves", "Pick-and-pop spacing", "Face-
            up drives"],
            "secondary": ["Box-out battle drills", "Offensive put-backs"]},
        "C": {"primary": ["Mikan drill (обидві руки)", "Drop-step hook", "Rim
            protection positioning"],
            "secondary": ["High-post two-man game", "Vertical jump timing"]}
    }

    warmups = ["Скакалка (3х3 хв)", "Робота з тенісним м'ячем", "Динамічна
    розминка", "Координаційна драбина"]

    risk_percent = int(risk_score * 100)

    if fatigue_status == "High Danger":
        focus = "Повне відновлення ЦНС"
        selected = random.sample(RECOVERY_TOOLKIT, 3)
        content = (
            f"ШІ зафіксував критичний ризик втоми ({risk_percent}%).\n"
            f"Будь-яке навантаження сьогодні підвищує ризик травми.\n\n"
            f"Ваш протокол відновлення:\n"
            f"1. {selected[0]}\n"
            f"2. {selected[1]}\n"
            f"3. {selected[2]}\n\n"
            f"Обов'язковий мінімум сну цієї ночі: 8.5 годин."
        )

    elif fatigue_status == "Moderate Risk":
        focus = "Техніка (Low Impact)"
        selected = random.sample(LOW_IMPACT_DRILLS, 3)
        content = (
            f"Ризик втоми підвищений ({risk_percent}%).\n"
            f"Тіло потребує відпочинку від ударних навантажень. Жодних стрибків
            чи спринтів.\n\n"
            f"План на сьогодні:\n"
            f"1. Суглобова розминка (10 хв)\n"
            f"2. {selected[0]}\n"

```

```

        f"3. {selected[1]}\n"
        f"4. {selected[2]}"
    )

    else:
        position = user.position if user.position in POSITION_DRILL_MATRIX else
"PG"
        pos_data = POSITION_DRILL_MATRIX[position]
        # 2 первинні та 1 вторинна навички для різноманітності
        primary_drills = random.sample(pos_data["primary"], 2)
        secondary_drill = random.sample(pos_data["secondary"], 1)
        selected_drills = primary_drills + secondary_drill

        todays_warmup = random.choice(warmups)

        focus = f"Інтенсивний розвиток ({position})"
        content = (
            f"Організм повністю відновлений (Ризик: {risk_percent}%).\n"
            f"Час для інтенсивної роботи на майданчику:\n\n"
            f"1. {todays_warmup}\n"
            f"2. {selected_drills[0]}\n"
            f"3. {selected_drills[1]}\n"
            f"4. {selected_drills[2]}\n"
            f"5. Заминка (5 хв легкий біг + розтяжка)"
        )

    if existing_plan:
        existing_plan.fatigue_risk = fatigue_status
        existing_plan.plan_focus = focus
        existing_plan.plan_content = content
        db.commit()
        db.refresh(existing_plan)
        return existing_plan
    else:
        new_plan = models.GeneratedPlan(
            user_id=user_id,
            date=today,
            fatigue_risk=fatigue_status,
            plan_focus=focus,
            plan_content=content
        )
        db.add(new_plan)
        db.commit()
        db.refresh(new_plan)
        return new_plan

```

models.py

```

from sqlalchemy import Column, Integer, String, Float, Date, ForeignKey, JSON,
Text, Index
from sqlalchemy.orm import relationship
from database import Base
class User(Base):
    __tablename__ = "users"

    user_id = Column(Integer, primary_key=True, index=True)
    email = Column(String, unique=True, index=True)
    password_hash = Column(String)
    name = Column(String, index=True)
    age = Column(Integer)
    height_cm = Column(Float)
    weight_kg = Column(Float)
    position = Column(String)

```

```

        metrics = relationship("DailyMetric", back_populates="owner",
                                lazy="selectin", cascade="all, delete-orphan")
        shoes = relationship("ShoeInventory", back_populates="owner",
                              lazy="selectin", cascade="all, delete-orphan")
        plans = relationship("GeneratedPlan", back_populates="owner",
                              lazy="selectin", cascade="all, delete-orphan")

    def __repr__(self) -> str:
        return f"<User(user_id={self.user_id}, name='{self.name}',
position='{self.position}')>"

class DailyMetric(Base):
    __tablename__ = "daily_metrics"
    __table_args__ = (
        Index('ix_daily_metrics_user_date', 'user_id', 'date'),
    )

    metric_id = Column(Integer, primary_key=True, index=True)
    user_id = Column(Integer, ForeignKey("users.user_id"), nullable=False)

    shoe_id = Column(Integer, ForeignKey("shoes_inventory.shoe_id",
ondelete="SET NULL"), nullable=True)

    date = Column(Date, nullable=False)
    duration_minutes = Column(Integer)
    activity_type = Column(String)
    rpe_score = Column(Integer)
    hrv_value = Column(Float)
    sleep_hours = Column(Float)

    owner = relationship("User", back_populates="metrics")
    shoe = relationship("ShoeInventory", back_populates="metrics")

    def __repr__(self) -> str:
        return f"<DailyMetric(metric_id={self.metric_id}, date={self.date},
type='{self.activity_type}')>"

class ShoeInventory(Base):
    __tablename__ = "shoes_inventory"

    shoe_id = Column(Integer, primary_key=True, index=True)
    user_id = Column(Integer, ForeignKey("users.user_id"), nullable=False)
    brand_model = Column(String)
    cushion_type = Column(String)
    current_hours_played = Column(Float, default=0.0)
    max_lifespan_hours = Column(Float)

    owner = relationship("User", back_populates="shoes")
    metrics = relationship("DailyMetric", back_populates="shoe")

    def __repr__(self) -> str:
        return f"<ShoeInventory(shoe_id={self.shoe_id},
model='{self.brand_model}',
wear={self.current_hours_played}/{self.max_lifespan_hours})>"

class GeneratedPlan(Base):
    __tablename__ = "generated_plans"
    __table_args__ = (
        Index('ix_generated_plans_user_date', 'user_id', 'date'),
    )

    plan_id = Column(Integer, primary_key=True, index=True)
    user_id = Column(Integer, ForeignKey("users.user_id"), nullable=False)

```

```

date = Column(Date, nullable=False)
fatigue_risk = Column(String)
plan_focus = Column(String)
plan_content = Column(Text, nullable=False)

owner = relationship("User", back_populates="plans")

def __repr__(self) -> str:
    return f"<GeneratedPlan(plan_id={self.plan_id}, date={self.date},
risk='{self.fatigue_risk}')>"

```

userStore.js

```

import { defineStore } from 'pinia'

export const useUserStore = defineStore('user', {
  state: () => ({
    user: null,
    userId: localStorage.getItem('userId') || null,
    token: localStorage.getItem('token') || null,
    loading: false,
    error: null,
    _fetchController: null
  }),

  getters: {
    isAuthenticated: (state) => !!state.token && !!state.userId,
    hasMetrics: (state) => state.user?.metrics && state.user.metrics.length > 0,
  },

  actions: {
    async fetchUser() {
      if (!this.userId || !this.token) return

      this.loading = true
      this.error = null

      const cachedKey = `user_data_${this.userId}`
      const cachedData = localStorage.getItem(cachedKey)
      if (cachedData) {
        try {
          this.user = JSON.parse(cachedData)
        } catch {
          localStorage.removeItem(cachedKey)
        }
      }

      if (!navigator.onLine) {
        this.loading = false
        return
      }

      if (this._fetchController) {
        this._fetchController.abort()
      }
      this._fetchController = new AbortController()

      try {
        const API_URL = import.meta.env.VITE_API_URL || 'https://basketball-api-kyiv.onrender.com';
        const response = await fetch(`${API_URL}/api/users/${this.userId}`, {
          signal: this._fetchController.signal,
          headers: {
            'Authorization': `Bearer ${this.token}`,

```

```

        'Content-Type': 'application/json'
    }
    })

    if (response.status === 401) {
        throw new Error('Токен протерміновано')
    }

    if (!response.ok) throw new Error('Помилка мережі')

    const freshUser = await response.json()
    this.user = freshUser

    // Оновлення кешу
    localStorage.setItem(cachedKey, JSON.stringify(freshUser))

    } catch (error) {
        if (error.name === 'AbortError') return

        this.error = error.message

        if (error.message === 'Токен протерміновано') {
            this.logout() // Автоматичний вихід
        }
    } finally {
        this.loading = false
    }
    },

    setAuthData(userId, token) {
        this.userId = userId
        this.token = token
        localStorage.setItem('userId', userId)
        localStorage.setItem('token', token)
    },

    logout() {
        const cachedKey = `user_data_${this.userId}`;

        this.user = null;
        this.userId = null;
        this.token = null;

        localStorage.removeItem('userId');
        localStorage.removeItem('token');
        localStorage.removeItem(cachedKey);
    }
    })
}

```

schemas.py

```

from pydantic import BaseModel, Field, model_validator
from typing import Optional, List, Literal
import datetime as dt

# === БАЗОВІ ТИПИ (СУБОПИ) ===
CourtPosition = Literal["PG", "SG", "SF", "PF", "C"]
ActivityCategory = Literal["Training", "Game", "Recovery"]

class UserBase(BaseModel):
    email: str
    name: str
    age: int

```

```

    height_cm: float
    weight_kg: float
    position: str

class UserCreate(BaseModel):
    email: str = Field(..., example="player@gmail.com")
    name: str = Field(..., example="Dmytro")
    age: int = Field(..., gt=0)
    height_cm: float = Field(..., gt=100)
    weight_kg: float = Field(..., gt=30)
    position: CourtPosition
    password: str

class UserUpdate(BaseModel):
    name: Optional[str] = None
    age: Optional[int] = Field(None, gt=0)
    height_cm: Optional[float] = Field(None, gt=100)
    weight_kg: Optional[float] = Field(None, gt=30)
    position: Optional[CourtPosition] = None

class UserResponse(UserBase):
    user_id: int
    class Config: from_attributes = True

class UserLogin(BaseModel):
    email: str
    password: str

class PasswordChange(BaseModel):
    old_password: str
    new_password: str

class EmailChange(BaseModel):
    new_email: str

class ShoeCreate(BaseModel):
    brand_model: str
    shoe_type: Literal["Баскетбольні", "Інші"]
    surface_type: Literal["Паркет", "Гума/Тартан", "Гібрид (Мікс)", "Асфальт"]
    initial_wear_percentage: int = Field(0, ge=0, le=100)

class ShoeResponse(BaseModel):
    shoe_id: int
    user_id: int
    brand_model: str
    cushion_type: str
    current_hours_played: float
    max_lifespan_hours: float
    class Config: from_attributes = True

class CalibrationDay(BaseModel):
    date: dt.date
    activity_type: ActivityCategory
    duration_minutes: int = Field(0, ge=0)
    rpe_score: int = Field(0, ge=0, le=10)
    sleep_hours: float = Field(..., ge=0, le=24)

    @model_validator(mode='after')
    def validate_recovery(self) -> 'CalibrationDay':
        if self.activity_type == "Recovery":
            self.duration_minutes = 0
            self.rpe_score = 0
        return self

```



```

class MetricCreate(BaseModel):
    date: dt.date
    sleep_hours: float = Field(..., ge=0, le=24)
    duration_minutes: int = Field(..., ge=0)
    rpe_score: int = Field(..., ge=0, le=10)
    activity_type: ActivityCategory
    shoe_id: Optional[int] = None
    hrv_value: Optional[float] = Field(None, ge=0.0)

    @model_validator(mode='after')
    def validate_recovery(self) -> 'MetricCreate':
        if self.activity_type == "Recovery":
            self.duration_minutes = 0
            self.rpe_score = 0
            self.shoe_id = None
        return self

class MetricResponse(BaseModel):
    metric_id: int
    user_id: int
    date: dt.date
    sleep_hours: float
    duration_minutes: int
    rpe_score: int
    activity_type: str
    shoe_id: Optional[int]
    hrv_value: Optional[float]
    class Config: from_attributes = True

class PlanResponse(BaseModel):
    plan_id: int
    date: dt.date
    fatigue_risk: str
    plan_focus: str
    plan_content: str
    class Config: from_attributes = True

class UserWithDetails(UserResponse):
    metrics: List[MetricResponse] = []
    shoes: List[ShoeResponse] = []
    plans: List[PlanResponse] = []

    readiness_score: int = 100
    fatigue_risk: str = "Калібрування"
    acwr_ratio: str = "0.00"
    acwr_status: str = "Немає даних"
    days_in_system: int = 0
    class Config: from_attributes = True

```

ДОДАТОК В

Інформаційно-аналітична система превенції спортивних травм на основі
рекурентної нейромережі LSTM та концепції ACWR

Презентація дипломного проєкту

Аркушів 21

Київ 2026

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА ПРЕВЕНЦІЇ СПОРТИВНИХ ТРАВМ НА ОСНОВІ РЕКУРЕНТНОЇ НЕЙРОМЕРЕЖІ LSTM ТА КОНЦЕПЦІЇ ACWR

Виконав

студент гр. КВ-23 Домуші Д. Д.

Керівник

Потапова К.Р. Кандидат технічних наук, доцент

Актуальність проблем

Парадокс тренувального навантаження

Найбільший ризик травми виникає не через стабільно високе навантаження, а внаслідок **нелінійних стрибків** між гострим (поточним) і хронічним (накопиченим) навантаженням.

Обмеження існуючих систем

Комерційні фітнес-системи фокусуються на кардіопоказниках, ігноруючи **позиційну специфіку** зовнішнього навантаження .

Економічний бар'єр

Рішення на кшталт Catapult є **фінансово недоступними** для студентського та аматорського спорту через обов'язкову апаратну залежність

Мета та задачі дослідження

Мета

Розроблення та валідація відкритої інформаційно-аналітичної системи для предиктивного оцінювання ризику травм у баскетболі на основі концепції ACWR і рекурентної нейромережі LSTM.

Основні задачі

1

Розробити гібридний алгоритм оцінювання накопиченої втоми на основі **sRPE**, телеметрії сну та **HRV**

2

Розробити математичну модель **нелінійної деградації** амортизаційного ресурсу спортивного взуття

3

Навчити та валідувати рекурентну предиктивну модель (**LSTM**) на базі багатовимірних часових рядів

4

Реалізувати клієнт-серверну архітектуру комплексу (**FastAPI + PWA**).

Об'єкт та предмет дослідження

Об'єкт дослідження

Процес моніторингу фізичного стану та адаптаційних процесів спортсменів в ігрових видах спорту.

Предмет дослідження

Математичні моделі (ACWR), алгоритми предиктивної аналітики (LSTM) та інформаційно-аналітична архітектура системи запобігання травмам.

Оцінка внутрішнього навантаження

Метод sRPE з урахуванням біомеханіки

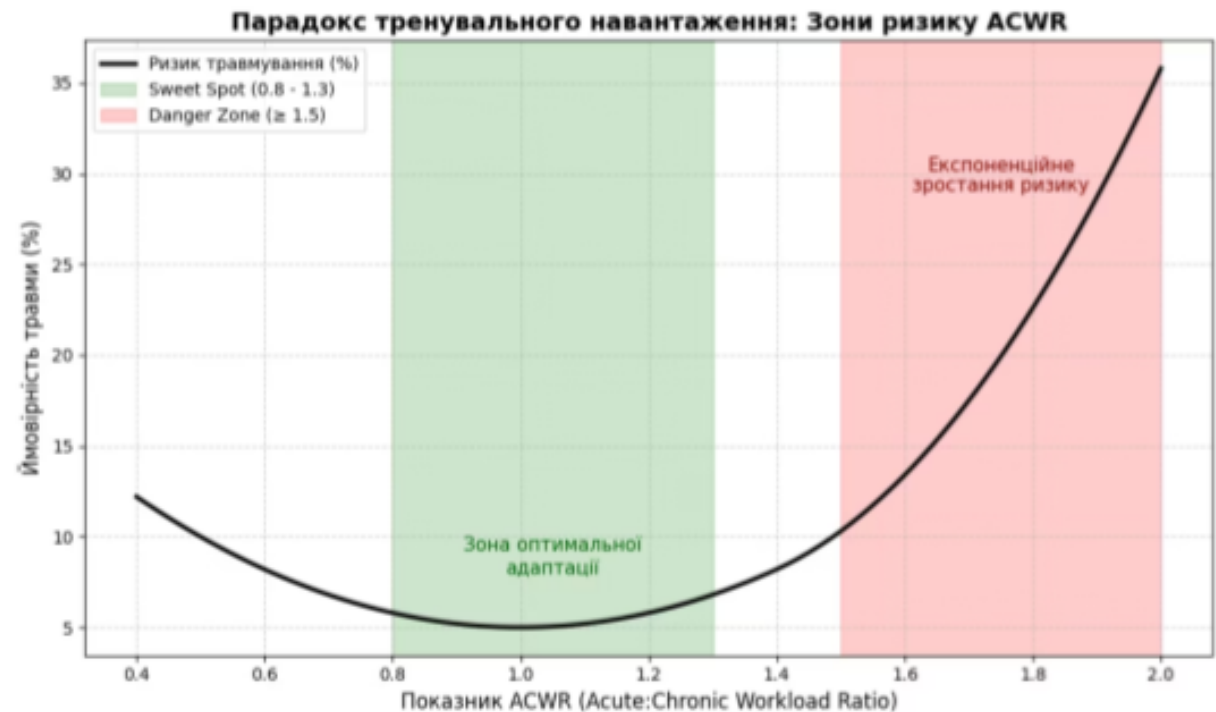
$$L = RPE \times t \times k_{pos}$$

де враховано зусилля, час та ваговий коефіцієнт амплуа баскетболіста.

Концепція ACWR (Acute:Chronic Workload Ratio)

$$ACWR = \frac{\frac{1}{7} \sum_{i=1}^7 L_i}{\frac{1}{28} \sum_{j=1}^{28} L_j}$$

Розраховується як відношення середнього навантаження за поточний мікроцикл (7 діб) до базового навантаження за мезоцикл (28 діб)



Предиктивний моніторинг екіпірування

Проблема

Деградація **амортизаційних матеріалів** баскетбольного взуття є прямим чинником ризику **компресійних травм** колінного суглоба. Знос амортизації відбувається нелінійно.

Мультиплікативна модель

Замість статичного ліміту пробігу використовується **адаптивна система** коефіцієнтів (Маса тіла × Тип покриття × Спеціалізація).

Формула:

$$E_{\max} = E_{\text{base}} \cdot k_{\text{weight}} \cdot k_{\text{surface}} \cdot k_{\text{type}}$$

Практична реалізація

Застосовано метод **лінійної інтерполяції** для плавного безперервного перерахунку **безпечного ліміту** взуття без жорстких порогових стрибків.

Архітектура ШІ (Нейромережа LSTM)

Чому LSTM?

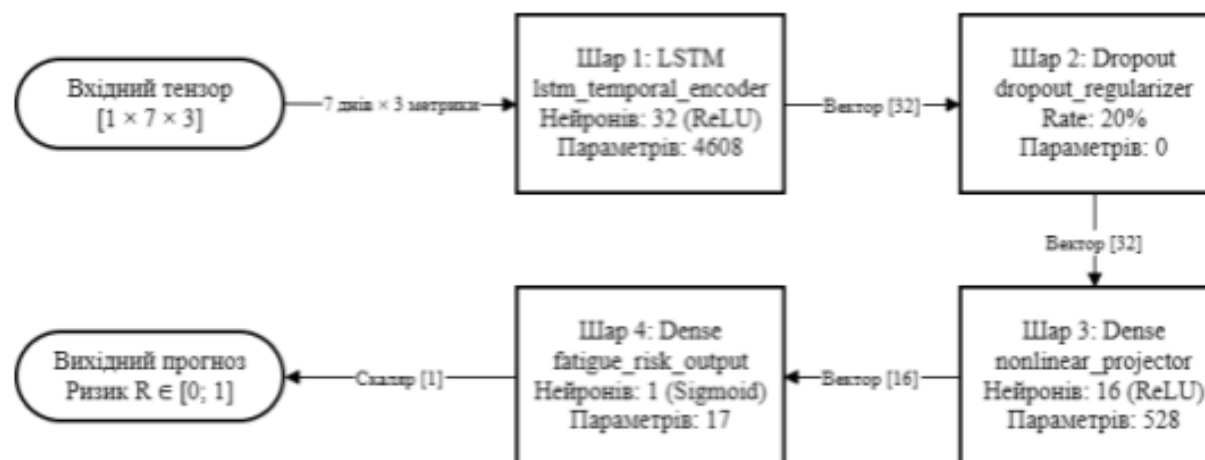
Здатність утримувати довгострокові залежності у часових рядах і долати проблему згасаючого градієнта

Структура моделі

- **Вхідний тензор ознак (7 діб):** sRPE, тривалість сесії, тривалість сну.
- Шар LSTM (**32 нейрони**, активація ReLU) з Dropout (0.2) для запобігання перенавчанню
- **Повнозв'язний шар (Dense)** на 16 нейронів (активація ReLU).
- **Вихідний шар (Sigmoid)** — скалярний вихід $\in [0,0; 1,0]$.

Динамічний модифікатор

Варіабельність серцевого ритму (HRV, rMSSD) інтегровано як постобчислювальний мультиплікатор стану нервової системи.



Відмовостійка клієнт-серверна архітектура

Серверна частина (Бекенд)

Асинхронний REST-фреймворк **FastAPI** на мові **Python**.
Забезпечує високу пропускну здатність.

База даних

Реляційна СУБД **PostgreSQL** із застосуванням
механізмів пулінгу з'єднань (**Connection Pool**) для
стабільної роботи у хмарному середовищі.

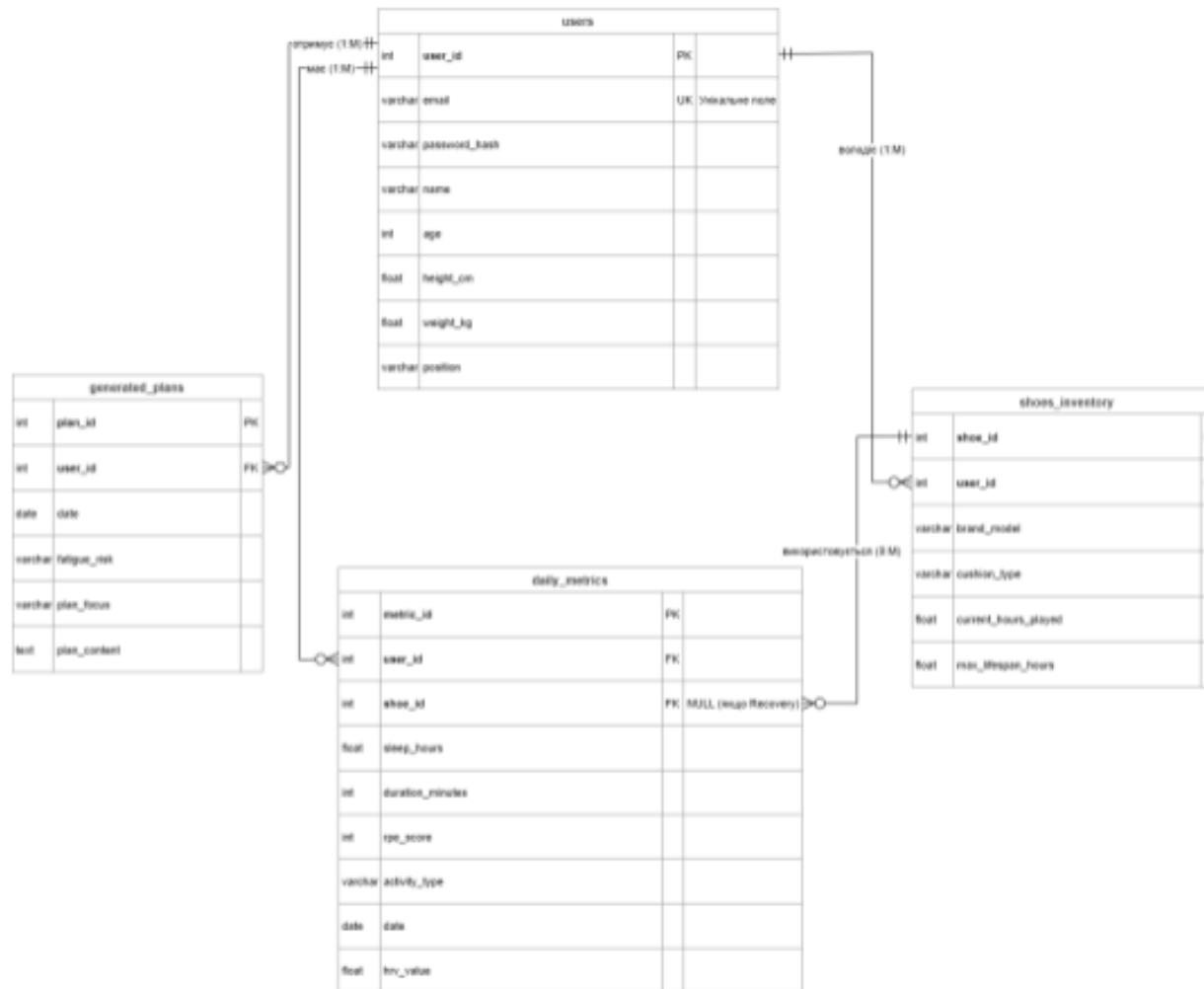
Клієнтська частина (Фронтенд)

Технологія **Progressive Web App (PWA)**. Забезпечує
кросплатформність та доступність безпосередньо зі
смартфона.

Оптимізація

Обчислень
Реалізовано модуль **Keep-Alive** для усунення затримки
«холодного старту» ШІ-ядра.

Реляційна структура бази даних



USERS

Зберігає антропометричні параметри та амплуа. Впроваджено механізм каскадного видалення (**delete-orphan**) для збереження цілісності.

DAILY_METRICS

Фіксує результати тренувальних циклів, сон та HRV.

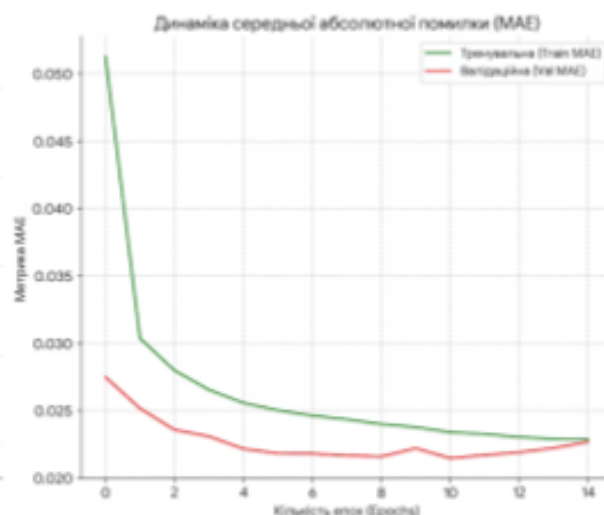
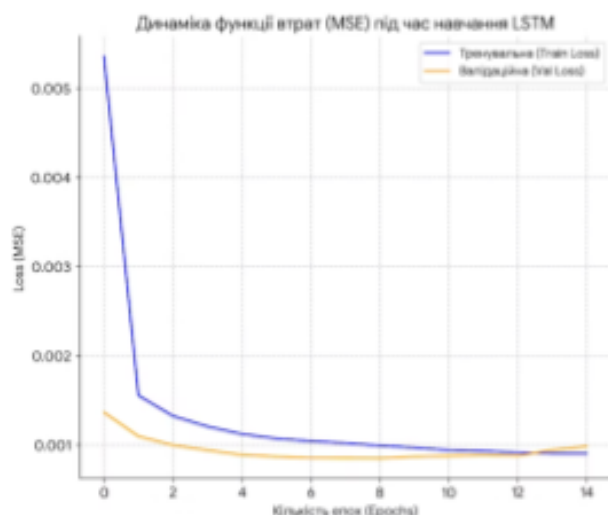
SHOES_INVENTORY

Облік ресурсу спортивного взуття із **динамічним залишковим показником** амортизації.

GENERATED_PLANS

Агрегує розрахований статус ризику та адаптивний контент тренування.

Результати експериментального навчання



Вибірка

50 000 згенерованих добових записів, стратифікованих за ігровими позиціями. Алгоритм оптимізації: **Adam**.

Результат тестування на незалежній вибірці (10 000 записів)

✓ Відсутність ефектів: згасаючого градієнта та перевантаження (overfitting)

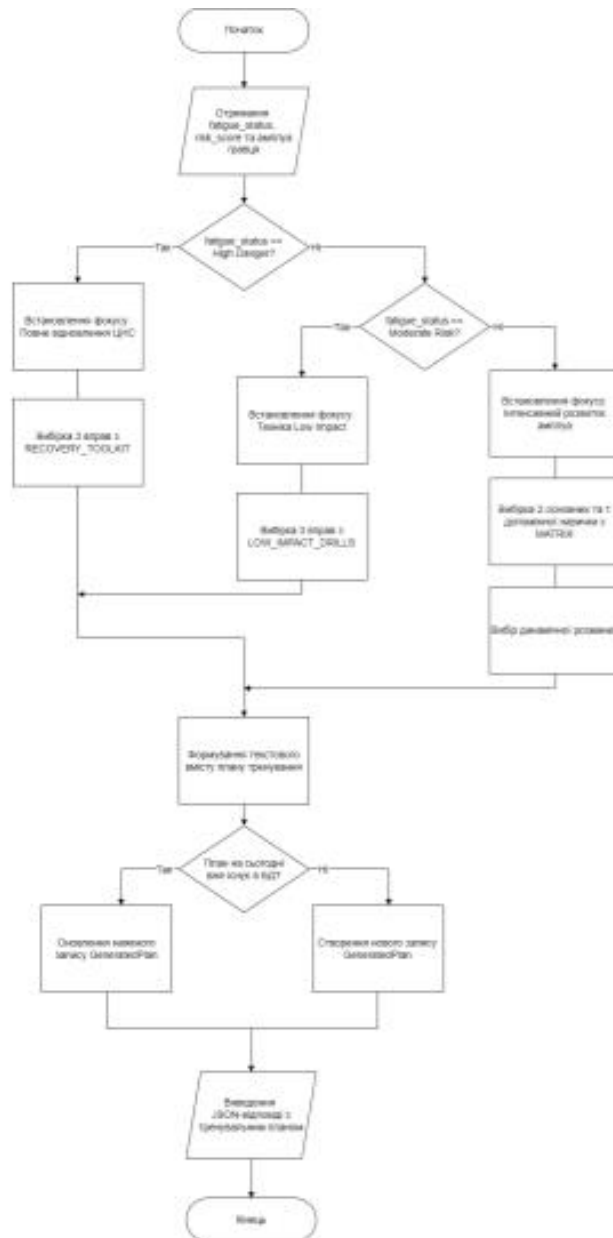
Висновок: Модель апроксимує рівень кумулятивного перевантаження з MAE = 0,0227 (2,27%) на незалежній тестовій вибірці

MAE = 0.0227

Середня абсолютна помилка

2,27%

Похибка апроксимації



Прескриптивна аналітика: Модуль Drill Engine

- **Перехід від дескриптивної аналітики (інформування) до прескриптивної (автоматична генерація тренувальних рішень).**
- **Механізм роботи:** При фіксації статусу «High Danger» (високий ризик травми) система автоматично відхиляє інтенсивні навантаження та формує відновлювальний протокол.
- **Проблема «Холодного старту»:** Для нових гравців без історії метрик система генерує стартовий мікроцикл за допомогою алгоритму тасування Фішера-Єйтса.
- **Варіативність:** Стохастична вибірка вправ здійснюється суворо з урахуванням ігрового амплуа (периметр для захисників, гра в зоні для центрових).

Клієнтський рівень системи (PWA)

Технологія

Клієнтська частина реалізована як **Progressive Web App (PWA)** на базі фреймворку **Vue.js**.

Кросплатформність

Додаток працює безпосередньо з браузера смартфона, **не потребуючи завантаження** з App Store чи Google Play.

Офлайн-доступ

Використання **Service Workers** та локального кешування (**Pinia**) дозволяє зберігати працездатність додатку безпосередньо на на спортивному майданчику при нестабільному інтернеті.

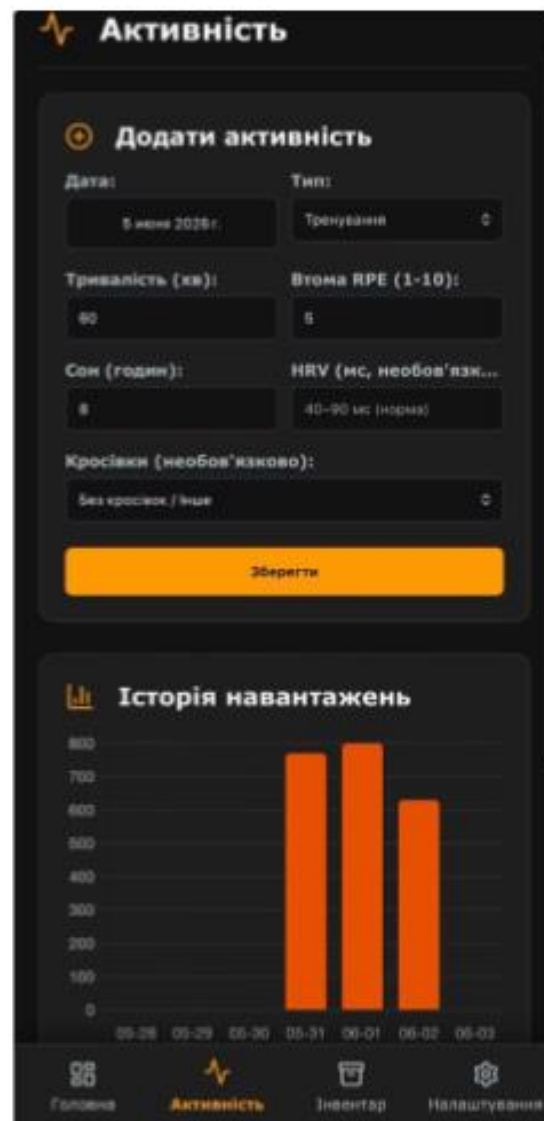
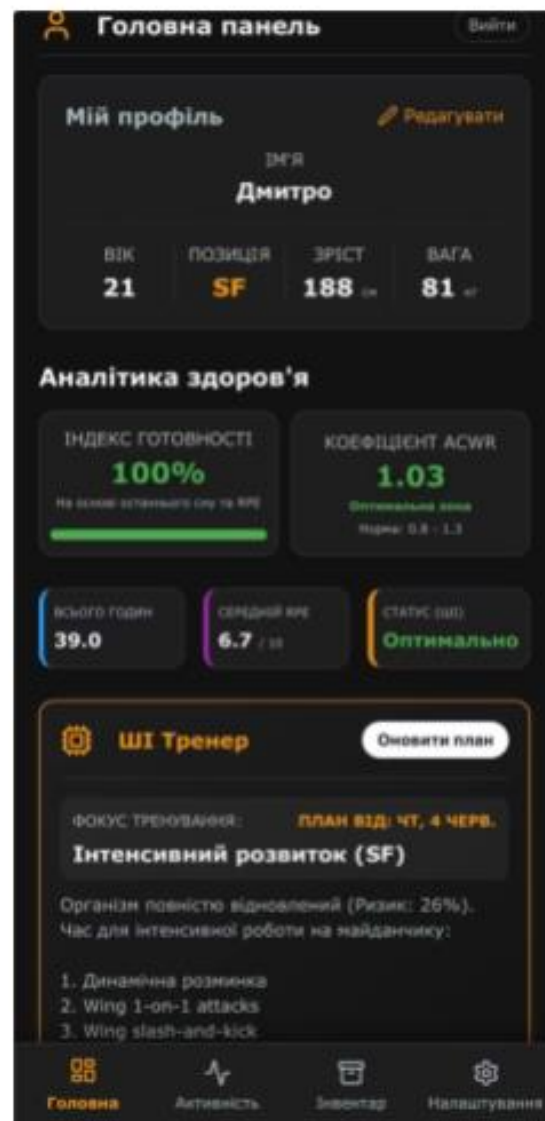
Візуалізація та аналітика показників

Дешборд

Агрегує розраховані показники ризику, коефіцієнт ACWR та персоналізовані приписи ШІ-тренера.

Моніторинг динаміки

Графіки дозволяють атлету наочно відстежувати власні фізіологічні тренди та утримувати навантаження в межах оптимального зеленого коридору (Sweet Spot).



Інтерактивний облік ресурсу екіпірування

Практичне застосування

Спортсмен додає свою пару взуття, вказуючи тип покриття (паркет, асфальт) та спеціалізацію.

Динамічний перерахунок

Серверний модуль автоматично розраховує залишковий амортизаційний ресурс $E_{\{max\}}$ з урахуванням маси тіла гравця.

Превентивний захист

Своєчасне сповіщення про знос матеріалів (EVA-піни) знижує ризик компресійних травм колін у важких гравців.

Інвентар

Додати нові кросівки

Модель кросівок:

Тип взуття:

Баскетбольні (x1.0)

Основне покриття:

Паркет (x1.0)

Максимальний ресурс (розрахунок ШІ):

83 год

Враховано вашу вагу (81 кг), тип взуття та покриття.

Початковий знос (якщо б/в):0%

0 год

Додати в інвентар

Мої Кросівки

Nike KD 17

БаскетбольніПаркет

Знос:57.9 / 82.4 год

Головна

Активність

Інвентар

Налаштування

Порівняльний аналіз із комерційними аналогами

Характеристика	Масові фітнес-додатки (Nike Run, Strava)	Професійні трекери (Catapult, Whoop)	Розроблена система (Баскетбольний ШІ)
Апаратна залежність	Не потребує	Повністю залежить від дороговартісних апаратних датчиків	Не потребує елітних трекерів
Тип аналітики	Статичний моніторинг зносу взуття за кілометражем	Високоточний аналіз, але виключно описувальна аналітика	Предиктивний аналіз (LSTM) + прописувальна зміна планів
Плани тренувань	Статичні плани без урахування накопиченої втоми	—	Адаптивний моніторинг взуття та прописувальна зміна планів при ризику травми

Наукова новизна та інноваційність

Синтез моделей

Вперше поєднано модифікований метод **sRPE**, концепцію **ACWR** та телеметрію сну в єдиний вектор для рекурентної нейромережі **LSTM**.

Нова математична модель взуття

Перехід від **статичного ліміту** зносу до **адаптивної системи** мультиплікативних коефіцієнтів з використанням лінійної інтерполяції.

Демократизація технологій

Система досягає $MAE = 0,0227$ без апаратних трекерів, спростовуючи необхідність пропрієтарного обладнання класу Catapult для базової предиктивної аналітики

Апробація та практичне впровадження

ДОВІДКА про впровадження

Ця довідка видана студенту Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (група ЮВ-23) Домуні Дмитру Дмитровичу про те, що результати його дипломного проєкту на тему: «Інформаційно-аналітична система тренувальних програм на основі рекурентної нейронної мережі LSTM та конвеїєра ACWR» успішно впроваджені у навчально-тренувальний процес Товариства з обмеженою відповідальністю «Приватний заклад освіти "Гімназія Кларіс Вербіс"».

Зокрема, у практичній роботі спортивних секцій та на уроках фізичної культури використовуються наступні програмні розробки:

1. Мікросервісна інформаційно-аналітична система для збору та зберігання фізіологічних показників учнів-спортсменів.
2. Алгоритм предиктивної аналітики для прогнозування ризиків травмування під час фізичних навантажень.
3. Клієнтський веб-інтерфейс для оперативного моніторингу фізичного стану.

Використання вказаних розробок дозволило автоматизувати аналітику даних, підвищити ефективність тренувального процесу та своєчасно запобігати перевтомі учнів. Програмний продукт відзначається стабільністю та має високу практичну цінність.

Директор
ТОВАРИСТВА З ОБМЕЖЕНОЮ
ВІДПОВІДАЛЬНІСТЮ
«ГІМНАЗІЯ КЛАРІС ВЕРБІС»

М.П.



Сергій СУЛІМ

Впровадження

Результати дипломного проєкту успішно впроваджені у навчально-тренувальний процес спортивних секцій ТОВ «ПЗО „Гімназія Кларіс Вербіс“».

Практичний ефект

Використання мікросервісної системи дозволило автоматизувати аналітику даних, підвищити ефективність занять та своєчасно запобігати перевтомі учнів під час фізичних навантажень.

Обмеження поточної системи

1 Природа навчальної вибірки

Навчання моделі **LSTM** (50 000 записів) здійснювалося на згенерованих (синтетичних) даних. Реальна телеметрія міститиме більше статистичного шуму.

2 Архітектура генерації планів

Маршрутизація тренувальних планів на етапі MVP все ще використовує жорсткі евристичні тригери (наприклад, сон < 6 годин).

3 Ручне введення даних

Система покладається на ручне введення користувачем оцінки зусиль (sRPE) та сну, що створює суб'єктивне упередження.

Вектори подальшого розвитку

1

Клінічна валідація

Збір емпіричних даних на фокус-групі діючих баскетболістів протягом повноцінного змагального макроциклу.

2

MLOps-інтеграція

Повна інтеграція навченої моделі у REST API для генерації приписів виключно на основі тензорного виводу ШІ, без застосування статичних правил.

3

Апаратна інтеграція

(Wearables)
Синхронізація системи з API популярних смарт-годинників (Apple Watch, Garmin, Whoop) для автоматичної екстракції фаз сну та варіабельності серцевого ритму (HRV).

Загальні висновки

Успішна реалізація

Спроектовано відмовостійку архітектуру (**FastAPI + PostgreSQL + PWA**), що забезпечує стабільну обробку біометричної телеметрії.

Висока точність ШІ

Навчена неймережа LSTM прогнозує рівень кумулятивної перевтоми з середньою абсолютною помилкою (MAE) всього **0.0227 (2,27%)**.

Практична цінність

Система впроваджена у навчально-тренувальний процес гімназії «Кларіс Вербіс» та забезпечує автоматизований **data-driven** контроль навантаження без інфраструктурних витрат

Дякую за увагу!

Домуші Дмитро

студент групи КВ-23

Email: kv23domushchi@gmail.com

Протестувати PWA-

Д



ДОДАТОК Г

Інформаційно-аналітична система превенції спортивних травм на основі
рекурентної нейромережі LSTM та концепції ACWR

Копія тез конференцій

Аркушів 8

Київ 2026

EUROPEAN SCIENCE AND INNOVATION CONGRESS

Proceedings of VII International Scientific and Practical Conference

Barcelona, Spain

1-3 June 2026

Barcelona, Spain

2026

UDC 001.1

The 7th International scientific and practical conference “European science and innovation congress” (June 1-3, 2026) Barca Academy Publishing, Barcelona, Spain. 2026. 433 p.

ISBN 978-84-15927-36-5

The recommended citation for this publication is:

Ivanov I. Analysis of the phauistic composition of Ukraine // European science and innovation congress. Proceedings of the 7th International scientific and practical conference. Barca Academy Publishing. Barcelona, Spain. 2026. Pp. 21-27. URL: <https://sci-conf.com.ua/vii-mizhnarodna-naukovo-praktichna-konferentsiya-european-science-and-innovation-congress-1-3-06-2026-barselona-ispaniya-arhiv/>.

Editor

Komarytskyy M.L.

Ph.D. in Economics, Associate Professor

Collection of scientific articles published is the scientific and practical publication, which contains scientific articles of students, graduate students, Candidates and Doctors of Sciences, research workers and practitioners from Europe, Ukraine and from neighbouring countries and beyond. The articles contain the study, reflecting the processes and changes in the structure of modern science. The collection of scientific articles is for students, postgraduate students, doctoral candidates, teachers, researchers, practitioners and people interested in the trends of modern science development.

e-mail: barca@sci-conf.com.ua

homepage: <https://sci-conf.com.ua>

©2026 Scientific Publishing Center “Sci-conf.com.ua” ®

©2026 Barca Academy Publishing ®

©2026 Authors of the articles

26.	<i>Melnyk D., Chyrka Yu.</i>	111
	AI EXPLAINABILITY IN MEDICAL APPLICATIONS	
27.	<i>Naboka A. V.</i>	118
	ASSESSMENT OF THE DEFORMABILITY OF A REINFORCED CONCRETE QUAY WALL UNDER OPERATIONAL CONDITIONS	
28.	<i>Naboka A. V.</i>	121
	EXPERIMENTAL INVESTIGATION OF THE DEFORMABILITY OF A MONOLITHIC REINFORCED CONCRETE SLAB UNDER HYDROSTATIC LOADING	
29.	<i>Naboka A. V.</i>	124
	EXPERIMENTAL AND NUMERICAL INVESTIGATION OF THE STRESS-STRAIN STATE OF A LONG-SPAN SADDLE-SHAPED SHELL	
30.	<i>Novikov A.</i>	128
	SEIRD MODELS AS A FRAMEWORK IN DECISION SUPPORT SYSTEMS	
31.	<i>Бакицький Т. Д., Заводний О. О., Калякін С. В.</i>	136
	БЕЗФАЙЛОВІ МЕТОДИ ЕКСПЛУАТАЦІЇ ВРАЗЛИВОСТЕЙ REDIS ЯК ЗАГРОЗА ЦІЛІСНОСТІ МІКРОСЕРВІСНИХ СЕРЕДОВИЩ	
32.	<i>Бондар Б. С., Неміріч О. В.</i>	141
	ОПТИМІЗАЦІЯ РЕЦЕПТУРНОГО СКЛАДУ СУХИХ ПРОТЕЇНОВО-ВІТАМІННИХ СУМІШЕЙ МЕТОДАМИ ЛІНІЙНОГО ТА БАГАТОКРИТЕРІАЛЬНОГО ПРОГРАМУВАННЯ	
33.	<i>Васюк Я. Є.</i>	146
	РОЗУМНІ КОШИКИ: АВТОМАТИЗАЦІЯ ПОКУПОК БЕЗ КАС ЧЕРЕЗ МОБІЛЬНИЙ ДОДАТОК	
34.	<i>Вичужанін В. В., Буберенко С. О.</i>	148
	МЕТОДИ ІНТЕЛЕКТУАЛЬНОЇ АДАПТАЦІЇ UI/UX ДЛЯ ОПТИМІЗАЦІЇ КОГНІТИВНОГО НАВАНТАЖЕННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ	
35.	<i>Домуці Д. Д., Потапова К. Р., Кучмій О. О.</i>	150
	ЗАСТОСУВАННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОЦІНЮВАННЯ КУМУЛЯТИВНОЇ ВТОМИ ТА ТРАВМАТИЧНИХ РИЗИКІВ АТЛЕТІВ	
36.	<i>Кулініч О. М., Яремчук А. І.</i>	155
	AI-ТЕХНОЛОГІЇ У ПРОТИДІЇ ДЕЗІНФОРМАЦІЇ ТА ІНФОРМАЦІЙНО-ПСИХОЛОГІЧНИМ ОПЕРАЦІЯМ	
37.	<i>Курас М. А., Будник М. М.</i>	162
	ІНФОРМАЦІЙНА СИСТЕМА ОБЛІКУ ТА АНАЛІЗУ ПЕРСОНАЛЬНИХ ФІНАНСІВ НА ОСНОВІ КОМПОНЕНТНО-ОРІЄНТОВАНОЇ АРХІТЕКТУРИ	
38.	<i>Парфенюк Д. В.</i>	167
	РОЗУМНЕ БРОНЮВАННЯ ГОТЕЛІВ: III-ПРОГНОЗУВАННЯ НАЙКРАЩОЇ ЦІНИ	

ЗАСТОСУВАННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОЦІНЮВАННЯ КУМУЛЯТИВНОЇ ВТОМИ ТА ТРАВМАТИЧНИХ РИЗИКІВ АТЛЕТІВ

Домуші Дмитро Дмитрович
студент

Потапова Катерина Романівна
кандидат технічних наук, доцент

Кучмій Оксана Олександрівна
старший викладач кафедри СПСКС

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
м. Київ, Україна

Вступ. / Introductions. У сучасному спорті управління тренувальним навантаженням є фундаментальним фактором для максимізації результатів та предиктивної профілактики травматизму. Концептуальною основою моніторингу стану спортсменів є парадокс тренувального навантаження Т. Габбетта та показник ACWR (Acute:Chronic Workload Ratio). Згідно з цією теорією, найбільший ризик травми виникає внаслідок різких нелінійних коливань між гострим (поточним) та хронічним (накопиченням) тренувальним стресом.

Незважаючи на активний розвиток спортивної інженерії, залишається низка невирішених проблем. Професійні комплекси трекінгу (наприклад, системи Catapult) є економічно недоступними для студентського та аматорського спорту через високу вартість та технологічну прив'язку до апаратних IMU-сенсорів та GPS-трекерів. Натомість масові фітнес-системи фокусуються переважно на кардіоваскулярних показниках і повністю ігнорують позиційну специфіку ігрових видів спорту, а також біомеханічні фактори (зокрема деградацію амортизаційних властивостей спортивного взуття), що є прямим каталізатором компресійних травм.

Ціль роботи. / Aim. Проектування, програмна реалізація та валідація відкритої мікросервісної інформаційно-аналітичної системи для предиктивного

оцінювання ризиків спортивних травм у баскетболі на основі рекурентної нейронної мережі з довгою короткочасною пам'яттю (LSTM) та гібридної оцінки кумулятивної втоми.

Матеріали та методи. / Materials and methods. Інформаційно-аналітичну систему спроектовано на базі мікросервісної архітектури. Серверну частину реалізовано мовою Python з використанням асинхронного REST-фреймворку FastAPI та СУБД PostgreSQL. Клієнтський інтерфейс базується на технології Progressive Web App (PWA), що забезпечує кросплатформність та доступність системи безпосередньо зі смартфонів на спортивному майданчику. Програмна реалізація алгоритмів машинного навчання виконана за допомогою бібліотеки TensorFlow/Keras.

Для оцифрування тренувального навантаження без використання апаратних трекерів було адаптовано метод Session-RPE із введенням вагового коефіцієнта ігрового амплуа баскетболіста. Оцінка фізіологічної готовності базується на концепції ACWR, де розраховується відношення поточного гострого стресу (за 7 діб) до хронічної адаптаційної бази (за 28 діб). Важливим інноваційним елементом методології є запропонована математична модель предиктивного розрахунку деградації амортизаційних властивостей спортивного взуття:

$$E_{max} = E_{base} \cdot k_{weight} \cdot k_{surface} \cdot k_{type} \quad (1)$$

де E_{max} — розрахунковий безпечний ресурс екіпірування, E_{base} — базовий еталонний ресурс, а емпіричні коефіцієнти (k_{weight} , $k_{surface}$, k_{type}) динамічно враховують масу тіла спортсмена, тип покриття майданчика та спеціалізацію взуття за допомогою лінійної інтерполяції.

Оскільки фізіологічна втома має кумулятивний нелінійний характер, для прогнозування ризику перетренованості використано рекурентну нейронну мережу з довгою короткочасною пам'яттю (LSTM). Дані обробляються методом ковзного вікна розміром 7 діб. Вхідний тривимірний тензор містить нормалізовані значення sRPE, фактичної тривалості активності та телеметрії

сну. Специфічною інженерною особливістю архітектури є застосування об'єктивного показника варіабельності серцевого ритму (HRV, параметр rMSSD) як динамічного коригуючого мультиплікатора на етапі постобробки. Це дозволяє нейромережі миттєво адаптувати прогноз ризику у випадку різкого функціонального перенапруження вегетативної нервової системи.

Результати та обговорення. / Results and discussion. Експериментальне навчання предиктивної моделі проводилося на згенерованому стратифікованому масиві з 50 000 добових записів. Результати тестування на незалежній вибірці підтвердили високу та стабільну прогностичну здатність рекурентної мережі: показник середньої абсолютної помилки (MAE) склав 0,0227. Це означає, що середнє абсолютне відхилення прогнозованого алгоритмом рівня перевтоми від фактичного сталонного значення становить лише 2,27%.

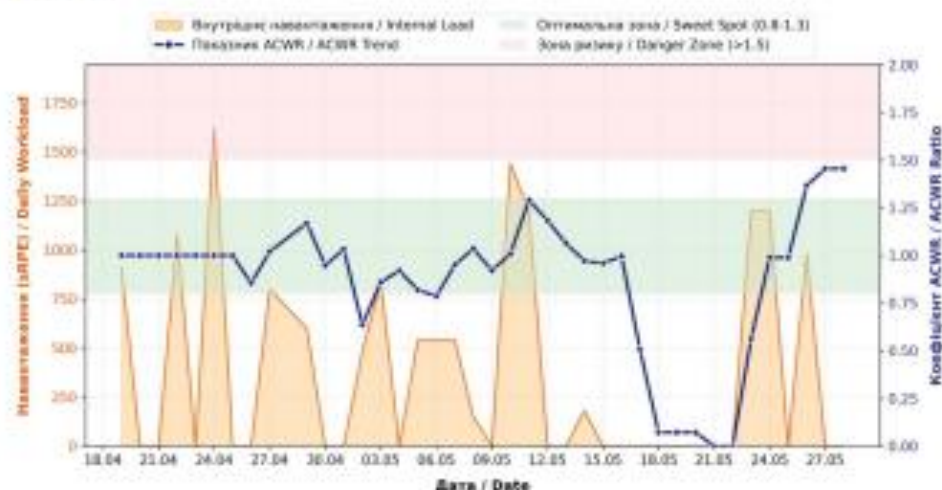


Рис. 1. Динаміка внутрішнього навантаження (sRPE) атлета та коливання показника ACWR відносно зон фізіологічного ризику

На базі отриманих прогнозів функціонує модуль прескриптивної аналітики «Drill Engine». У разі ідентифікації нейромережею критичного рівня ризику травмування (статус «High Danger»), система автоматично реконструює тренувальний процес, генеруючи відновлювальні протоколи з урахуванням ігрового амплуа баскетболіста. Проблему «холодного старту» для нових

користувачів вирішено за допомогою алгоритму тасування Фішера-Єйтса. Взаємодія користувача з обчислювальним ядром здійснюється через PWA-додаток, який візуалізує динаміку навантажень та поточний знос екіпірування у вигляді інтерактивного дашборда.

Висновки. / Conclusions. У результаті виконання дослідження вирішено актуальне науково-практичне завдання превенції травматизму та адаптивного моніторингу тренувального процесу в баскетболі:

1. Спроектовано та імплементовано відмовостійку мікросервісну архітектуру (FastAPI, PostgreSQL, PWA), здатну агрегувати та обробляти телеметрію гравців.
2. Сконструйовано та натреновано модель машинного навчання (LSTM), яка з високою точністю ($MAE = 0,0227$) прогнозує системну втому на основі гібридних маркерів (sRPE, тривалість сну, варіабельність серцевого ритму).
3. Запропоновано інноваційну мультиплікативну модель адаптивного моніторингу ресурсу спортивного взуття з використанням лінійної інтерполяції, що дозволяє предиктивно знижувати ризик компресійних травм.
4. Доведено, що використання хмарних алгоритмів прескриптивної аналітики забезпечує перехід до доказового управління тренувальним процесом без обов'язкового залучення дороговартісних апаратних трекерів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gabbett T. J. The training — injury prevention paradox: should athletes be training smarter and harder? *British Journal of Sports Medicine*. 2016. Vol. 50, No. 5. P. 273-280.
2. Claudino J. G., de Oliveira Capanema D., de Souza T. V. et al. Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review. *Sports Medicine – Open*. 2019. Vol. 5, No. 1. P. 1-12.
3. Foster C., Florhaug J. A., Franklin J. et al. A new approach to

monitoring exercise training. *The Journal of Strength & Conditioning Research*. 2001. Vol. 15, No. 1. P. 109-115.

4. Verdejo R., Mills N. J. Heel-shoe interactions and the durability of EVA foam running-shoe midsoles. *Journal of Biomechanics*. 2004. Vol. 37, No. 9. P. 1379-1386.

5. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997. Vol. 9, No. 8. P. 1735-1780.

ДОДАТОК Д

Інформаційно-аналітична система превенції спортивних травм на основі
рекурентної нейромережі LSTM та концепції ACWR

Копія довідки про впровадження

Аркушів 1

Київ 2026

ДОВІДКА про впровадження

Ця довідка видана студенту Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (група КВ-23) Домущі Дмитру Дмитровичу про те, що результати його дипломного проєкту на тему: «Інформаційно-аналітична система превенції спортивних травм на основі рекурентної нейромережі LSTM та концепції ACWR» успішно впроваджені у навчально-тренувальний процес Товариства з обмеженою відповідальністю «Приватний заклад освіти "Гімназія Кларіс Вербіс"».

Зокрема, у практичній роботі спортивних секцій та на уроках фізичної культури використовуються наступні програмні розробки:

1. Мікросервісна інформаційно-аналітична система для збору та зберігання фізіологічних показників учнів-спортсменів.
2. Алгоритм предиктивної аналітики для прогнозування ризиків травмування під час фізичних навантажень.
3. Клієнтський вебдодаток для оперативного моніторингу фізичного стану.

Використання вказаних розробок дозволило автоматизувати аналітику даних, підвищити ефективність тренувального процесу та своєчасно запобігати перевтомі учнів. Програмний продукт відзначається стабільністю та має високу практичну цінність.

Директор
ТОВ «ПЗО "Гімназія Кларіс Вербіс"»
М.П.



Сергій СУЛИМ