

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«АДАПТИВНА НАВЧАЛЬНА СИСТЕМА З ПЕРСОНАЛІЗАЦІЄЮ
ПІД СТИЛІ НАВЧАННЯ КОРИСТУВАЧІВ»**

Виконав: студент 2 курсу

Рівень: магістр

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Дашдаміров Гліб Ельдарович

Керівник: к.пед.н., доцент

Логінова Наталія Іванівна

Рецензент к.т.н., доцент

Чикунів Павло Олександрович

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет кібербезпеки та інформаційних технологій
Кафедра інформаційних технологій
Галузь знань: 12 Інформаційні технології
Спеціальність: 122 Комп'ютерні науки
Назва освітньої програми: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій

доцент _____ Н.І. Логінова

« ____ » _____ 20__ року

ЗАВДАННЯ

на кваліфікаційну (магістерську) роботу
здобувачу Дашдамірову Глібу Ельдаровичу

1. Тема роботи: «Адаптивна навчальна система з персоналізацією під стилі навчання користувачів»

Керівник роботи: к.пед.н, доцент Логінова Наталія Іванівна

затверджені наказом НУ ОЮА від «10» жовтня 2025 року № 3182-06

2. Строк подання здобувачем роботи «25» листопада 2025 рік

3. Дата видачі завдання «02» червня 2025 рік

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми, вивчення літературних джерел та складання плану роботи		
2.	Підготовка першого розділу роботи та подання його керівнику		
3.	Підготовка другого розділу роботи та подання його керівнику		
4.	Підготовка третього розділу роботи та подання його керівнику		
5.	Підготовка вступу та висновків		
6.	Доопрацювання роботи з урахуванням зауважень керівника		
7.	Подача електронного варіанту роботи для перевірки на плагіат		
8.	Допуск роботи до захисту завідуючим кафедри		
9.	Захист роботи		

Здобувач _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота здобувача 2-го курсу магістратури Дашдамірова Гліба Ельдаровича на тему «Адаптивна навчальна система з персоналізацією під стилі навчання користувачів» за спеціальності 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки».

Структура кваліфікаційної роботи складається зі вступу, основної частини з трьох розділів, висновків, переліку посилань. Робота викладена на 72 сторінках, містить 21 рисунок, 7 таблиць. Перелік посилань включає 47 джерел.

Мета дослідження – розробка моделі адаптивної навчальної системи, яка забезпечує персоналізацію освітнього процесу відповідно до індивідуальних стилів навчання користувачів. Об'єкт дослідження – процес адаптивного комп'ютерного навчання, що передбачає динамічну зміну освітнього контенту та методів подачі матеріалу відповідно до індивідуальних характеристик і стилів навчання користувачів. Предмет дослідження – модель адаптивної навчальної системи, її структура, принципи функціонування та методи персоналізації навчального процесу залежно від стилів навчання користувачів.

У роботі проведено аналіз понять, принципів і теоретичних засад адаптивного навчання; визначено методи та інструменти адаптації навчального контенту. Розроблено архітектуру адаптивної навчальної системи та сформовано її ключові функціональні характеристики. Реалізовано моделювання користувача й процедуру діагностики стилів навчання, визначено структуру навчального контенту та пов'язаних метаданих. Розроблено модель оцінювання навчальних досягнень та побудовано модель машинного навчання для класифікації стилів навчання користувачів. Сформована концептуальна модель даних системи. Обраний технологічний стек (Python, scikit-learn, TensorFlow, PostgreSQL, MLflow) забезпечив ефективну реалізацію алгоритмів машинного навчання, обробку великих масивів даних і модульну архітектуру системи.

АДАПТИВНА, НАВЧАЛЬНА, СИСТЕМА, МОДЕЛЬ, МАШИННЕ НАВЧАННЯ,
СТИЛЬ НАВЧАННЯ, PYTHON

ABSTRACT

Qualification work of the 2nd year master's student Hlib Dashdamirov, titled “Adaptive learning system with personalization based on users’ learning styles” is completed within the specialty 122 “Computer Science”, educational program “Computer Science”.

The structure of the qualification work includes an introduction, a main part consisting of three chapters, conclusions, and a list of references. The work comprises 72 pages and contains 21 figures and 7 tables. The list of references includes 47 sources.

The purpose of the study is to develop a model of an adaptive learning system that enables the personalization of the educational process according to the individual learning styles of users. The object of the study is the process of adaptive computer-based learning, which involves dynamic modification of educational content and methods of material presentation based on users’ individual characteristics and learning styles. The subject of the study is the model of the adaptive learning system, its structure, principles of operation, and methods of personalizing the learning process depending on users’ learning styles.

The work analyzes the concepts, principles, and theoretical foundations of adaptive learning; identifies methods and tools for adapting educational content. The architecture of the adaptive learning system is developed, and its key functional characteristics are defined. User modeling and a procedure for diagnosing learning styles are implemented; the structure of educational content and system metadata is also determined. A model for evaluating learning outcomes is designed, and a machine learning model for classifying users’ learning styles is developed. A conceptual data model of the system is constructed. The selected technology stack (Python, scikit-learn, TensorFlow, PostgreSQL, MLflow) ensured effective implementation of machine learning algorithms, processing of large data sets, and a modular system architecture.

ADAPTIVE, LEARNING, SYSTEM, MODEL, MACHINE LEARNING, LEARNING
STYLE, PYTHON

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Поняття, принципи та теоретичні основи адаптивного навчання	10
1.2 Моделі стилів навчання та користувача в адаптивних системах.....	12
1.3 Методи і механізми адаптації навчального контенту	16
1.4 Аналіз сучасних адаптивних освітніх систем.....	21
Висновки до розділу 1	25
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ АДАПТИВНОЇ НАВЧАЛЬНОЇ СИСТЕМИ.....	26
2.1 Загальна архітектура та функціональні вимоги адаптивної навчальної системи	26
2.2 Моделювання користувача та діагностика стилю навчання	29
2.2.1 Модель користувача	29
2.2.2 Визначення стилю навчання користувача.....	32
2.3 Моделювання предметної області та навчального контенту	35
2.3.1 Структура навчального контенту	35
2.3.2 Структура метаданих	38
2.3.3 Модель оцінювання знань.....	40
2.3.4 Визначення ознак навчального контенту	43
2.4 Модель машинного навчання для визначення стилів навчання	45
2.5 Модель даних системи	49
Висновки до розділу 2	53
3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА МОДЕЛІ.....	54
3.1 Вибір технологічного стеку	54
3.2 Підготовка експериментального датасету.....	56
3.3 Реалізація та навчання моделі машинного навчання	59
3.4 Реалізація модуля адаптації контенту.....	61

3.5 Узагальнення результатів експериментів.....	63
Висновки до розділу 3	65
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ	69
ДОДАТКИ.....	73
Додаток А. Модуль підготовки експериментального датасету для АНС	73
Додаток Б. Приклади роботи алгоритму для різних сценаріїв.....	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BKT – Bayesian Knowledge Tracing
FSLSM – Felder-Silverman learning style model
Kolb – Kolb’s Experiential Learning Model
LMS – системи управління навчанням
LO – Learning Object
ORM – Object-Relational Mapping
PCA – Principal Component Analysis
RFE – Recursive Feature Elimination
SVM – Support Vector Machine
VARK – Visual, Auditory, Read/Write, Kinesthetic
АНС – адаптивна навчальна система
БД – база даних
ІКТ – інформаційно-комунікаційні технології
ІТ – інформаційні технології
МН – машинне навчання
CAT – Computerized Adaptive Testing
СКБД – система керування базами даних
ІІІ – штучний інтелект

ВСТУП

Актуальність кваліфікаційної роботи на тему «Адаптивна навчальна система з персоналізацією під стилі навчання користувачів» зумовлена швидким розвитком цифрових технологій та зростанням попиту на ефективні цифрові інструменти для освіти. Традиційні системи електронного навчання часто не враховують індивідуальні відмінності між студентами, що призводить до зниження мотивації та результативності засвоєння матеріалу. Адаптивні системи, які аналізують поведінку користувачів і підлаштовують навчальний контент під їхні стилі навчання, дозволяють створювати більш гнучке, індивідуалізоване й ефективне освітнє середовище.

Розробка таких систем є актуальною задачею для галузі комп'ютерних наук, оскільки поєднує методи штучного інтелекту та аналізу даних. Впровадження персоналізованих навчальних платформ може суттєво покращити якість освіти, підвищити рівень залученості студентів і сприяти формуванню компетентностей, необхідних для сучасного інформаційного суспільства. Таким чином, дослідження і розробка адаптивної навчальної системи мають як наукове, так і практичне значення.

Метою кваліфікаційної роботи є розробка моделі адаптивної навчальної системи, яка забезпечує персоналізацію освітнього процесу відповідно до індивідуальних стилів навчання користувачів.

Для досягнення цієї мети передбачено наступні **завдання**:

- проаналізувати сучасні підходи до адаптивного навчання та персоналізації освітнього контенту;
- визначити класифікацію стилів навчання та їх вплив на процес засвоєння інформації;
- змоделювати структуру навчального контенту та визначити його ознаки;
- розробити модель адаптивної навчальної системи з урахуванням індивідуальних особливостей студентів;
- обґрунтувати архітектуру системи та основні механізми адаптації;
- розробити та реалізувати навчання моделі машинного навчання.

Об'єкт дослідження – процес адаптивного комп'ютерного навчання, що

передбачає динамічну зміну освітнього контенту та методів подачі матеріалу відповідно до індивідуальних характеристик і стилів навчання користувачів.

Предмет дослідження – модель адаптивної навчальної системи, її структура, принципи функціонування та методи персоналізації навчального процесу залежно від стилів навчання користувачів.

У процесі виконання кваліфікаційної роботи використовувалися загальнонаукові та спеціальні методи наукового дослідження.

До загальнонаукових методів належать аналіз, синтез, узагальнення, систематизація та порівняння, які застосовувалися для вивчення теоретичних основ адаптивного навчання, визначення класифікацій стилів навчання та побудови логічної структури системи.

До спеціальних методів дослідження належать моделювання, проектування та структурно-функціональний аналіз, що використовувалися під час розроблення концептуальної моделі адаптивної навчальної системи. Також застосовано елементи методів штучного інтелекту та аналізу даних для опису механізмів персоналізації освітнього процесу.

Практичне значення отриманих результатів полягає у можливості застосування розробленої концептуальної моделі адаптивної навчальної системи для створення або вдосконалення сучасних електронних освітніх платформ. Запропоновані підходи до персоналізації навчального процесу дозволяють підвищити ефективність засвоєння знань користувачами, забезпечити індивідуальний підхід до навчання та покращити взаємодію між студентом і навчальним контентом.

Апробація матеріалів кваліфікаційної роботи здійснено на VI міжнародної науково-практичної конференції «Кібербезпека в сучасному світі: актуальні виклики» (28 листопада 2025 р., м. Одеса) [1].

Структура кваліфікаційної роботи відповідає меті та завданням дослідження та складається зі вступу, основної частини з трьох розділів, висновків, переліку посилань. Робота викладена на 72 сторінках, містить 21 рисунок, 7 таблиць. Перелік посилань включає 47 джерел.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття, принципи та теоретичні основи адаптивного навчання

Сучасна цифровізація освіти та постійне зростання обсягів навчальної інформації, необхідної для опанування майбутніми фахівцями, актуалізують потребу у створенні оптимальних умов навчання, які враховують індивідуальні особливості та потреби здобувачів освіти. Використання інформаційно-комунікаційних технологій (ІКТ), хмарних обчислень і веборієнтованих сервісів забезпечує можливість реалізації персоналізованого та адаптивного підходів до навчання [2].

Перехід до змішаного формату освітнього процесу, коли значна частина занять проводиться онлайн, спонукає педагогів до автоматизації навчання через системи управління навчальним процесом (LMS). Це, у свою чергу, підвищує навантаження на викладачів, пов'язане з моніторингом успішності, аналізом результатів і контролем навчальної активності студентів.

Адаптивне навчання розглядається як підхід, що передбачає динамічну зміну навчального середовища, складності завдань та навчального контенту відповідно до рівня підготовки, темпу навчання, стилю сприйняття та когнітивних особливостей студента. Такий підхід спрямований не лише на ефективну передачу знань, а й на формування навичок самостійного навчання, дослідницької діяльності та розвитку навчальної автономії здобувача [3].

Адаптивне навчання – це метод, який наголошує на орієнтованому підході в освіті на того, хто навчається.

За своєю сутністю адаптивне навчання базується на орієнтації на студента як центральну фігуру освітнього процесу. Воно визнає різноманітність індивідуальних здібностей і темпів навчання, надаючи персоналізовану підтримку для досягнення максимальних результатів. При правильній реалізації цей підхід сприяє підвищенню ефективності навчання та зниженню когнітивного навантаження шляхом узгодження складності матеріалу з поточним рівнем підготовки студента [4].

Теоретичні основи адаптивного навчання спираються на такі принципи:

- індивідуалізація – урахування особистісних характеристик, рівня підготовки

та стилів навчання;

- гнучкість – можливість змінювати темп, обсяг і форму подання матеріалу;
- зворотний зв'язок у реальному часі – постійний моніторинг результатів та оперативна адаптація контенту;
- модульність – структурованість матеріалу, що дає змогу формувати індивідуальні освітні траєкторії;
- аналітичність – використання даних для прийняття рішень щодо подальшої стратегії навчання.

Центральним елементом адаптивного підходу є персоналізована модель користувача, що формується на основі:

- системи збору даних про знання, активність і стиль навчання;
- системи логічних висновків, яка аналізує реакції користувача на зміни у навчальному середовищі;
- системи персоналізації, яка визначає індивідуальні цілі, прогнозує результати та коригує траєкторію навчання.

Реалізація цих компонентів можлива завдяки сучасним адаптивним платформам та хмарним сервісам, інтегрованим у LMS. Поєднання традиційних і цифрових засобів навчання забезпечує гнучкість, доступність і високу ефективність освітнього процесу.

Ефективне впровадження адаптивного навчання можливе лише за умови поєднання традиційних та цифрових засобів, що дозволяє забезпечити гнучкість і доступність освітнього процесу для різних категорій здобувачів. Важливо також враховувати когнітивне навантаження студентів, що може бути оптимізоване шляхом адаптації складності завдань і темпу подачі матеріалу [5].

Адаптивне навчання передбачає гнучку систему навчальних занять, адаптованих до індивідуальних особливостей тих, хто навчається. Застосовуючи адаптивну систему навчання, викладач повинен чітко розуміти, які завдання можна виконати за її допомогою.

Використання адаптивних систем сприяє підвищенню якості підготовки фахівців завдяки:

1. Можливості вибору власної траєкторії та темпу навчання.
2. Реалізації диференційованого підходу відповідно до рівня знань і типу сприйняття інформації.
3. Об'єктивності контролю та оцінювання результатів.
4. Наявності діагностичного моніторингу та корекції навчальної траєкторії.
5. Індивідуалізації завдань за рівнем складності й темпом опанування.
6. Підвищенню мотивації та розвитку творчого мислення студентів.
7. Формуванню партнерських відносин між студентом і викладачем.

Технічні аспекти адаптивного навчання охоплюють алгоритми:

- переходу між рівнями навчання залежно від правильності виконання завдань;
- формування наборів завдань відповідно до рівня підготовки студента;
- оцінювання досягнень та автоматичного коригування освітньої траєкторії.

Таким чином, адаптивне навчання виступає не лише технологічною інновацією, а стратегічним напрямом розвитку освіти, спрямованим на підвищення її ефективності, індивідуалізацію та гнучкість. Його концепція базується на комплексному аналізі освітніх даних і створенні персоналізованих моделей навчання, що дозволяють прогнозувати результати та підвищувати якість підготовки фахівців [6].

Далі розглянемо теоретичні моделі стилів навчання, що лежать в основі побудови адаптивних освітніх систем і визначають підходи до персоналізації контенту.

1.2 Моделі стилів навчання та користувача в адаптивних системах

Ефективність адаптивного навчання значною мірою залежить від розуміння індивідуальних особливостей здобувачів освіти, зокрема їхніх когнітивних стилів, способів сприйняття та обробки інформації. Саме тому моделювання стилів навчання і користувачів є фундаментальним елементом побудови адаптивних освітніх систем. Такі моделі дозволяють системі автоматично добирати навчальний контент, форми подачі матеріалу та типи завдань, що найкраще відповідають особистісним характеристикам студента.

Стили навчання стосуються різних способів, якими люди надають перевагу отриманню, обробці та запам'ятовуванню інформації. Існують різні моделі та теорії,

що класифікують стилі навчання – візуальний, слуховий, читання/письмо та кінестетичний.

Найпоширенішими є моделі Kolb, VARK та Felder-Silverman, які відрізняються підходом до класифікації способів навчання.

Модель Д. Колба (Kolb's Experiential Learning Model)

Модель ґрунтується на теорії навчання через досвід, згідно з якою процес навчання відбувається через циклічну взаємодію чотирьох етапів (рис. 1.1).

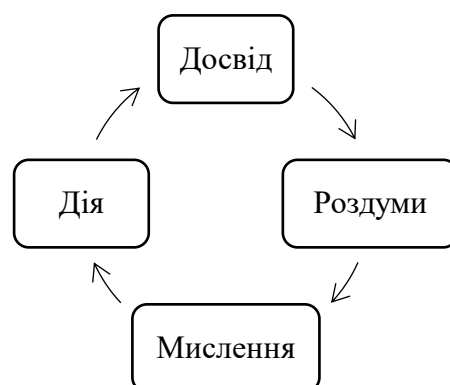


Рисунок 1.1 – Модель емпіричного навчання Kolb

Досвід (конкретний досвід) – навчання починається, коли студент використовує органи чуття та сприйняття, щоб взаємодіяти з тим, що відбувається зараз.

Роздуми (рефлексивне спостереження) – після досвіду студент розмірковує над тим, що сталося, та пов'язує почуття з ідеями про досвід.

Мислення (абстрактна концептуалізація) – студент займається мисленням, щоб дійти висновків та сформулювати теорії, концепції або загальні принципи, які можна перевірити.

Дія (активне експериментування) – студент перевіряє теорію та застосовує вивчене, щоб отримати зворотний зв'язок та створити наступний досвід.

На основі цих етапів виділяють чотири типи тих, хто навчається: конвергентні, дивергентні, асимілятори та акомодатори. У межах адаптивної системи це дозволяє формувати контент відповідно до домінуючого стилю навчання користувача: практичні завдання для акомодаторів, аналітичні матеріали для асиміляторів тощо [7].

Модель VARK (Visual, Auditory, Read/Write, Kinesthetic)

Модель VARK забезпечує основу для визначення та врахування індивідуальних

уподобань тих, хто навчається, у навчанні та заохочує викладачів використовувати різноманітні навчальні стратегії для задоволення цих різноманітних уподобань. VARK вводить чотири модальності (рис. 1.2).

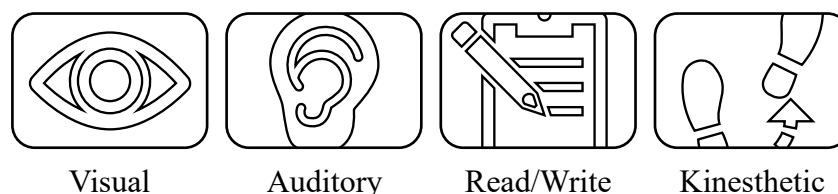


Рисунок 1.2 – Модель VARK

Візуальні (Visual) студенти сприймають матеріал через схеми, графіки, діаграми.

Слухові (Auditory) – краще засвоюють знання через лекції, обговорення.

Читання /письмо (Read/Write) – орієнтовані на текстову інформацію, конспекти, письмові завдання.

Кінестетичні (Kinesthetic) – засвоюють знання через рух, практичну діяльність, експерименти.

У контексті адаптивного навчання ця модель дозволяє системі автоматично підбирати форму представлення матеріалу: наприклад, для візуального стилю – інфографіку, для аудіального – подкасти, для кінестетичного – інтерактивні симуляції тощо [8].

Модель Felder–Silverman learning style model (FSLSM)

Модель розроблена Р. Фелдером і Л. Сільверманом є однією з найпопулярнішою у світі адаптивного навчання. Вона визначає чотири виміри стилів навчання (рис. 1.3).

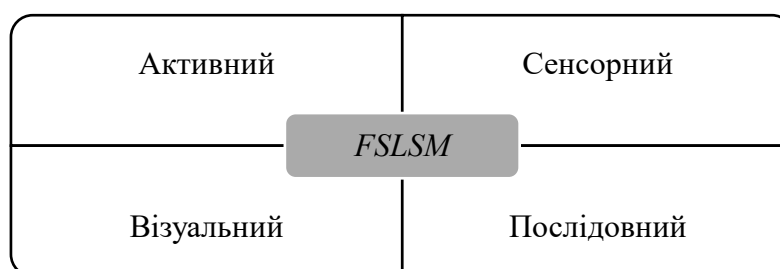


Рисунок 1.3 – Модель FSLSM

Активний – рефлексивний (спосіб обробки інформації).

Сенсорний – інтуїтивний (тип сприйняття).

Візуальний – вербальний (форма подання інформації).

Послідовний – глобальний (структура мислення).

Дана модель дозволяє враховувати не лише форму сприйняття інформації, але й когнітивні стратегії навчання [9].

Аналіз основних моделей стилів навчання показує, що індивідуальні відмінності у способах сприйняття та обробки інформації суттєво впливають на ефективність освітнього процесу. Використання розглянутих моделей стилів навчання (Kolb, VARK, FSLSM) створює підґрунтя для розроблення персоналізованих і адаптивних освітніх систем, що підвищують мотивацію та ефективність навчання завдяки урахуванню індивідуальних особливостей користувачів.

Індивідуалізація навчального процесу є однією з ключових цілей сучасних інтелектуальних навчальних систем. Для її досягнення використовуються три фундаментальні моделі: модель користувача, модель знань і модель адаптації. Їхнє узгоджене функціонування дозволяє системі ефективно взаємодіяти з кожним учнем, враховуючи його особисті характеристики, рівень підготовки та освітні потреби.

Модель користувача описує індивідуальні особливості того, хто навчається, включаючи:

- рівень знань у предметній області;
- стиль навчання (візуальний, аудіальний, кінестетичний тощо);
- мотивацію, цілі та поведінкові патерни;
- історію взаємодії з системою (відвідані сторінки, виконані завдання).

Ця модель є динамічною: вона оновлюється в реальному часі на основі дій користувача, що дозволяє системі адаптувати контент і методи навчання.

Модель знань репрезентує структуру предметної області, яку вивчає користувач. Вона включає:

- поняття та зв'язки між ними;
- рівні складності матеріалу;
- залежності між темами (наприклад, prerequisite-структури).

Модель адаптації визначає, як саме система поєднує модель користувача з

моделлю знань для персоналізації навчального процесу. Вона містить:

- педагогічні правила адаптації;
- механізми вибору контенту;
- алгоритми динамічного оновлення інтерфейсу та навігації.

Адаптація може бути реалізована через зміну послідовності подання матеріалу, варіативність завдань, рекомендації або автоматичне коригування складності.

Узгоджена робота цих трьох моделей є основою персоналізованих освітніх систем. Вони дозволяють не просто автоматизувати подання матеріалу, а створити адаптивне середовище, у якому кожен користувач отримує підтримку, відповідну саме його особливостям. Завдяки цьому інтелектуальні навчальні системи стають здатними не лише передавати знання, а й формувати повноцінний індивідуальний освітній досвід [10].

Інтеграція психологічних моделей навчання (Kolb, VARK, FSLSM) із технічними моделями користувача створює основу для побудови інтелектуальної адаптивної системи, здатної гнучко реагувати на освітні потреби кожного користувача.

1.3 Методи і механізми адаптації навчального контенту

Ефективність адаптивних освітніх систем значною мірою визначається тим, яким чином здійснюється адаптація навчального контенту до особистісних характеристик, рівня підготовки та стилю навчання здобувача освіти. Реалізація принципів індивідуалізації, гнучкості та зворотного зв'язку, розглянутих у п. 1.1, вимагає застосування конкретних методів і механізмів, що забезпечують динамічне налаштування освітнього середовища. Водночас, інтеграція моделей стилів навчання (Kolb, VARK, FSLSM) та моделей користувача, знань і адаптації, описаних у п. 1.2, створює теоретичну основу для практичної реалізації персоналізованого навчання.

Методи адаптації в інтелектуальних навчальних системах охоплюють сукупність педагогічних та технологічних рішень, спрямованих на узгодження навчального контенту з індивідуальними особливостями здобувача освіти. Одним із ключових напрямів такої адаптації є урахування поточного рівня знань користувача. Система безперервно відстежує підготовленість студента та відповідно змінює складність

навчальних матеріалів. Використання моделі знань, яка відображає структуру предметної області, дає змогу формувати як базові модулі для користувачів із недостатнім рівнем опанування, так і розширені матеріали для студентів, що досягають високих результатів. Такий механізм не лише забезпечує індивідуалізацію навчання, а й дозволяє оптимізувати навчальне навантаження, узгоджуючи складність контенту з рівнем компетентності здобувача [11].

Важливим компонентом адаптації є врахування стилів навчання. Спираючись на моделі VARK і FSLSM, система трансформує форму представлення матеріалу відповідно до домінуючих способів сприйняття інформації. Додатково система враховує когнітивні особливості та структурує навчальні матеріали та завдання.

Адаптація темпу навчання забезпечує реалізацію принципу гнучкості, дозволяючи студенту опановувати матеріал у власному ритмі. Система може подовжувати час на засвоєння складних фрагментів, прискорювати подання добре зрозумілих тем або змінювати кількість і характер повторюваних вправ залежно від динаміки навчальних успіхів. Застосовуючи алгоритми, що враховують криву забування, система також визначає оптимальні інтервали повернення до матеріалу для його стійкого запам'ятовування.

Важливу роль відіграє адаптація, спрямована на регулювання когнітивного навантаження. Вона передбачає логічне структурування матеріалу у вигляді смислових блоків, поступове ускладнення змісту та використання механізмів педагогічного супроводу, які поступово зменшуються у міру розвитку компетентності студента. Такий підхід дозволяє уникати перевантаження робочої пам'яті та створює умови для глибокого опрацювання складних концепцій.

Крім того, адаптація враховує мотиваційні та емоційні аспекти навчання. Оскільки модель користувача включає інформацію про освітні цілі, професійні інтереси та поведінкові особливості здобувача, система може формувати персоналізовані траєкторії навчання, добирати релевантні приклади й застосовувати елементи гейміфікації для підтримання навчальної активності. За потреби вона коригує рівень складності або інтенсивність зворотного зв'язку, реагуючи на емоційний стан студента та його залученість [12].

У своїй сукупності описані методи формують цілісну систему адаптивного подання навчального контенту, здатну гнучко реагувати на індивідуальні потреби здобувача освіти та забезпечувати високу ефективність пізнавальної діяльності.

Механізми адаптації в інтелектуальних навчальних системах становлять технічну основу реалізації адаптивних стратегій, забезпечуючи автоматизоване налаштування освітнього середовища відповідно до динамічних характеристик здобувача освіти. Їхня робота починається з діагностичного етапу, у межах якого система формує та постійно оновлює модель користувача. Діагностичні алгоритми включають визначення початкового рівня знань за допомогою тестування, ідентифікацію стилю навчання на основі відповідних опитувальників, безперервне спостереження за активністю студента, аналіз типових помилок і врахування результатів самооцінювання. Завдяки цьому забезпечується безперервний зворотний зв'язок, що дозволяє системі своєчасно уточнювати профіль користувача.

Подальший етап передбачає контентну адаптацію, у межах якої система автоматично добирає навчальні матеріали, спираючись на модель знань та індивідуальні характеристики студента. Така адаптація може ґрунтуватися на формалізованих правилах, що регулюють відповідність контенту потребам користувача, або на механізмах фільтрації та варіативності подання, коли кожна навчальна одиниця має кілька альтернативних форматів. Система також здатна динамічно комбінувати окремі модулі у персоналізовані освітні сторінки, створюючи варіативний і гнучкий навчальний простір.

Важливою складовою є навігаційна адаптація, яка забезпечує формування індивідуальної траєкторії навчання. Завдяки цьому студент отримує доступ до оптимальної послідовності тем, адаптованих навігаційних підказок та рекомендацій на основі аналізу власного прогресу й даних інших користувачів із подібними профілями. Візуалізація у вигляді адаптивної карти знань дозволяє чітко відображати вже опановані та доступні розділи курсу. Такий підхід враховує індивідуальні стилі мислення, надаючи можливість як поступального, так і глобального освоєння матеріалу [13].

Особливу роль відіграють механізми регулювання зворотного зв'язку, які адаптують його зміст, стиль і частоту. Система може пропонувати диференційовані

підказки, коригувати рівень деталізації пояснень залежно від компетентності користувача, а також враховувати його схильність до активного чи рефлексивного навчання. Емоційно чутливі повідомлення дозволяють підтримувати впевненість і мотивацію [14]. Механізми прогнозування та раннього втручання спрямовані на випереджальне виявлення можливих труднощів. За допомогою алгоритмів машинного навчання система прогнозує ризики зниження успішності, визначає студентів, які можуть потребувати додаткової підтримки, й автоматично модифікує навчальну траєкторію, пропонуючи компенсаторні матеріали або змінюючи темп навчання. Додатково застосовується кластеризація, яка дозволяє групувати користувачів за подібними характеристиками та обирати найбільш ефективні стратегії адаптації [15]. Окремим аспектом є мотиваційна адаптація, що реалізується через елементи гейміфікації. Система налаштовує бали, нагороди та рівні викликів відповідно до мотиваційного профілю здобувача освіти, формує персоналізовані завдання, здатні підтримувати стан оптимальної залученості, а також адаптує соціальні аспекти взаємодії, пропонуючи різні форми роботи залежно від особистісних уподобань [16].

Ефективна адаптивна система навчання базується на узгодженій роботі всіх розглянутих методів і механізмів. Процес адаптації здійснюється циклічно (рис. 1.4).

1. Діагностика – визначення характеристик користувача (рівень знань, стиль навчання, мотивація).

2. Моделювання – оновлення моделі користувача на основі зібраних даних.



Рисунок 1.4 – Процес адаптації навчання

3. Прийняття рішення – система персоналізації визначає оптимальну стратегію навчання, використовуючи модель адаптації.

4. Реалізація – контентна та навігаційна адаптація, подання персоналізованого контенту.

5. Зворотний зв'язок – надання адаптованих відгуків та підказок.

6. Оцінювання – моніторинг результатів та ефективності адаптації.

7. Корекція – коригування освітньої траєкторії відповідно до прогресу студента.

Такий підхід забезпечує безперервну оптимізацію навчального процесу та реалізацію всіх принципів адаптивного навчання: індивідуалізації, гнучкості, зворотного зв'язку, модульності та аналітичності.

Поєднання описаних методів і механізмів формує цілісне, багатокомпонентне адаптивне навчальне середовище, у якому кожен здобувач освіти отримує персоналізовану підтримку відповідно до власних потреб, можливостей і стилю навчання. Така інтегрована адаптація сприяє підвищенню ефективності засвоєння матеріалу, оскільки навчальний контент узгоджується з індивідуальними когнітивними та поведінковими характеристиками студента. Одночасно відбувається оптимізація когнітивного навантаження завдяки динамічному регулюванню складності, обсягу та темпу подання інформації.

Створювані системою індивідуальні освітні траєкторії відповідають особистим цілям і навчальним стратегіям здобувача освіти, надають можливість самостійно планувати, контролювати та оцінювати власний прогрес. Використання адаптивних алгоритмів також зменшує навантаження на викладача, оскільки автоматизує рутинні процеси моніторингу, оцінювання та добору матеріалів, дозволяючи педагогам зосередитися на індивідуальній підтримці та консультаціях. Крім того, системна обробка даних забезпечує об'єктивність оцінювання, мінімізуючи вплив суб'єктивних чинників та підвищуючи точність діагностики навчальних результатів.

Створювані системою індивідуальні освітні траєкторії відповідають особистим цілям і навчальним стратегіям здобувача освіти, сприяючи розвитку його навчальної автономії, що проявляється у здатності самостійно планувати, контролювати та оцінювати власний прогрес. Використання адаптивних алгоритмів також зменшує

навантаження на викладача, оскільки автоматизує рутинні процеси моніторингу, оцінювання та добору матеріалів, дозволяючи педагогам зосередитися на індивідуальній підтримці та консультаціях. Крім того, системна обробка даних забезпечує об'єктивність оцінювання, мінімізуючи вплив суб'єктивних чинників та підвищуючи точність діагностики навчальних результатів.

Таким чином, методи й механізми адаптації навчального контенту є практичною реалізацією теоретичних принципів адаптивного навчання та моделей стилів навчання. Їхнє впровадження в інтелектуальні навчальні системи створює основу для персоналізованої, ефективної та орієнтованої на студента освіти, що відповідає викликам сучасного цифрового суспільства.

1.4 Аналіз сучасних адаптивних освітніх систем

Розвиток адаптивного навчання нерозривно пов'язаний із появою та вдосконаленням спеціалізованих платформ, що реалізують складні механізми аналізу освітніх даних і персоналізації контенту. Сучасні цифрові інструменти дозволяють інтегрувати моделі користувача, знань і адаптації на рівні, недосяжному для традиційних засобів навчання, забезпечуючи високу точність добору матеріалу та динамічне регулювання освітньої траєкторії. У світовій практиці сформувалося кілька значущих рішень, які визначають тенденції розвитку інтелектуальних навчальних систем.

Однією з найбільш відомих є платформа Knewton Alta, що реалізує концепцію адаптивного навчання на основі алгоритмів машинного аналізу знань. Вона формує детальну карту компетентностей студента, визначає прогалини та автоматично добирє завдання відповідного рівня складності. Завдяки цьому формується індивідуалізована траєкторія, яка постійно коригується відповідно до поточного прогресу [17].

Подібний підхід демонструвала платформа Smart Sparrow, інтегрована згодом у продукти компанії Pearson. Її особливістю була можливість створення складних сценаріїв адаптації, що дозволяло викладачеві задавати логіку переходів між фрагментами контенту, враховуючи стиль навчання, рівень засвоєння та поведінкові особливості користувача. Університетські курси, побудовані на основі цієї системи, забезпечували високу гнучкість навчального дизайну [18].

Платформа ALEKS, що базується на теорії простору знань, пропонує специфічний підхід до моделювання компетентностей та визначає мінімальні множини понять, необхідних для засвоєння нової теми, і формує точні діагностичні профілі. Навчальна траєкторія створюється на основі математично обґрунтованої моделі, яка дозволяє прогнозувати готовність студента до опанування складніших концепцій [19].

У практиці вищої освіти помітне місце займає LANTIV Adaptive Learning Platform, яка підтримує формування індивідуальних освітніх траєкторій, здійснює глибоку діагностику прогресу та забезпечує раннє виявлення ризиків академічної неспішності. Її функціональність орієнтована на освітні заклади, що потребують масштабованих рішень для роботи з великою кількістю користувачів [20].

У межах дисциплін STEM широко застосовується система Carnegie Learning (MATHia), що реалізує когнітивні моделі й елементи інтелектуального тьюторингу. Вона безперервно оцінює процес мислення студента, а не лише кінцевий результат, і надає адаптивний зворотний зв'язок із поясненнями причин помилок [21].

Для порівняння існуючих інформаційних продуктів побудуємо SWOT-аналіз, який дозволить виділити сильні та слабкі сторони, можливості та загрози (табл. 1.1).

Сучасні адаптивні платформи відіграють ключову роль у розвитку персоналізованого навчання, поєднуючи психолого-педагогічні моделі зі складними аналітичними алгоритмами. Їх використання забезпечує гнучкість і варіативність подання матеріалу, сприяє формуванню індивідуальних освітніх траєкторій, що відповідають потребам і можливостям кожного здобувача освіти. Усі розглянуті платформи реалізують принципи адаптивного навчання, проте різняться за рівнем автономності, технічним підґрунтям і роллю викладача у процесі персоналізації.

Розроблення власної концептуальної моделі адаптивної навчальної системи є необхідним кроком для забезпечення персоналізації освітнього процесу відповідно до індивідуальних стилів навчання користувачів. Така модель дозволить не лише враховувати особливості кожного здобувача освіти, а й динамічно адаптувати зміст, темп і форму подання матеріалу до його унікального способу сприйняття інформації. Це створює умови для підвищення ефективності навчання, зростання мотивації та глибшого засвоєння знань.

Таблиця 1.1 – SWOT-аналіз інформаційних продуктів

Аспект	Smart Sparrow	Knewton	ALEKS	LANTIV Adaptive Learning	Carnegie Learning (MATHia)
S (Strengths) Сильні сторони	– висока гнучкість у створенні контенту; – можливість інтерактивного навчання без програмування; – потужна аналітика для викладача; – персоналізований зворотний зв'язок у реальному часі	– використання III та великих даних для точного підбору контенту; – автоматизована персоналізація навчальних шляхів; – масштабованість для великих онлайн-курсів; – безперервне оцінювання замість періодичних тестів	– висока точність діагностики знань; – побудова індивідуальних карт знань; – підвищення мотивації через роботу «на межі можливостей»; – доведена ефективність у STEM-дисциплінах	– формування індивідуальних освітніх траєкторій; – глибока діагностика прогресу студентів; – раннє виявлення ризиків академічної неуспішності; – масштабованість для великих закладів; – комплексна система моніторингу	– когнітивні моделі та інтелектуальний тьюторинг; – оцінка процесу мислення, а не лише результату; – адаптивний зворотний зв'язок з поясненнями; – доведена ефективність у STEM-освіті; – інтеграція з шкільними програмами
W (Weaknesses) Слабкі сторони	– залежність від активності викладача; – висока трудомісткість створення курсів; – обмежена автоматизація	– менша можливість ручного контролю викладача; – висока залежність від якості вхідних даних; – недостатня адаптація до нестандартних дисциплін	– орієнтація переважно на точні науки; – недостатня гнучкість у гуманітарних дисциплінах; – вузька спеціалізація контенту	– необхідність інтеграції з існуючими LMS; – висока вартість впровадження; – потреба в навчанні персоналу; – залежність від якості вхідних даних про студентів	– фокус переважно на математиці; – обмежена адаптація до інших предметів; – складність інтеграції в навчальний процес; – необхідність тривалого періоду адаптації

Продовження таблиці 1.1

Аспект	Smart Sparrow	Knewton	ALEKS	LANTIV Adaptive Learning	Carnegie Learning (MATHia)
О (Opportunities) Можливості	- інтеграція з LMS і платформами он-лайн-освіти; - використання в університетах і корпоративному навчанні; - розвиток спільнот викладачів-дизайнерів курсів	- масове впровадження у MOOC-платформи (Coursera, edX); - розширення галузей застосування через ШІ-моделі; - використання у корпоративному секторі	- подальше розширення дисциплін; - використання у школах і ВНЗ для діагностики знань; - інтеграція з адаптивними тестовими системами	- розширення на корпоративний сектор; - інтеграція з національними освітніми системами; - використання для прогнозування успішності; - створення загальнодержавних баз даних	- розширення на інші STEM-дисципліни; - адаптація для різних вікових груп; - використання в позашкільній освіті; - інтеграція з системами оцінювання
Т (Threats) Загрози	- конкуренція з більш автоматизованими системами - потреба у підготовці викладачів - залежність від стабільності інтернет-з'єднання	- ризик втрати точності через великі обсяги даних; - проблеми конфіденційності студентських даних; - залежність від алгоритмів ШІ, які не завжди прозорі	- технологічна складність оновлення моделей знань; - висока вартість ліцензування; - можлива втрата мотивації студентів при надмірній алгоритмізації	- конкуренція з глобальними платформами; - ризиків кібербезпеки та витоку даних; - залежність від технологічної інфраструктури; - опір змінам з боку викладачів	- зниження інтересу через фокус на одному предметі; - конкуренція з універсальними платформами; - ризик морального застарівання алгоритмів; - високі вимоги до технічного забезпечення

Висновки до розділу 1

У першому розділі здійснено аналіз теоретичних засад адаптивного навчання, що створило наукову основу для розробки концептуальної моделі адаптивної навчальної системи, орієнтованої на стилі навчання користувачів. Визначено сутність адаптивного навчання та його ключові принципи, реалізація яких можлива через персоналізовану модель користувача. Розгляд моделей навчальних стилів (Kolb, VARK, FSLSM) показав доцільність їх інтеграції з моделями користувача, знань і адаптації. Узагальнено методи адаптації та описано циклічний процес адаптації, що включає сім послідовних етапів. Проведений SWOT-аналіз сучасних систем дав змогу окреслити їх переваги, недоліки та перспективи, що підтвердило необхідність створення адаптивної навчальної системи з персоналізацією освітнього процесу.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ АДАПТИВНОЇ НАВЧАЛЬНОЇ СИСТЕМИ

2.1 Загальна архітектура та функціональні вимоги адаптивної навчальної системи

Адаптивна навчальна система (АНС) являє собою інтегроване програмно-інформаційне середовище, призначене для підтримки індивідуалізованого освітнього процесу шляхом динамічного добору та трансформації навчального контенту відповідно до когнітивних, поведінкових та індивідуально-типологічних характеристик користувача. Основою побудови такої системи є модульний підхід, який забезпечує структурну гнучкість, можливість подальшого розширення функціональності та інтеграцію з існуючими і перспективними освітніми платформами [22].

У межах концептуальної моделі АНС виокремлюють низку базових модулів, кожен із яких виконує спеціалізовані функції та взаємодіє з іншими через стандартизовані інтерфейси даних. Така модульність мінімізує взаємозалежності між компонентами та сприяє побудові масштабованої архітектури.

1. Підсистема управління користувачами є основним елементом контролю доступу і персоніфікації. До її функцій належать створення профілів користувачів, автентифікація та персоналізація налаштувань.

2. Підсистема збору та акумуляції навчальних даних відповідає за збір і структурування інформації, що виникає в процесі взаємодії користувача з навчальною системою.

3. Аналітичний модуль оцінювання знань є центральним елементом адаптивної логіки, оскільки він інтерпретує зібрані дані та формує модель знань користувача.

4. Модуль формування адаптивної навчальної траєкторії забезпечує практичну реалізацію адаптивності системи – добір навчального контенту, динамічне формування послідовності подання матеріалів, регуляція складності навчальних завдань та генерування рекомендацій, спрямованих на покращення результатів і заповнення прогалин у знаннях.

5. Підсистема управління навчальним контентом є структурованим репозиторієм освітніх матеріалів та відповідає за зберігання та опис навчальних ресурсів, побудову концептуальних графів знань, індексації та пошуку навчальних матеріалів.

6. Модуль аналітичної звітності – виконує узагальнення даних про прогрес, формування рекомендацій та статистичних моделей для підтримки навчання.

Для наочності розуміння модульної архітектури АНС побудуємо діаграму компонентів (рис. 2.1).

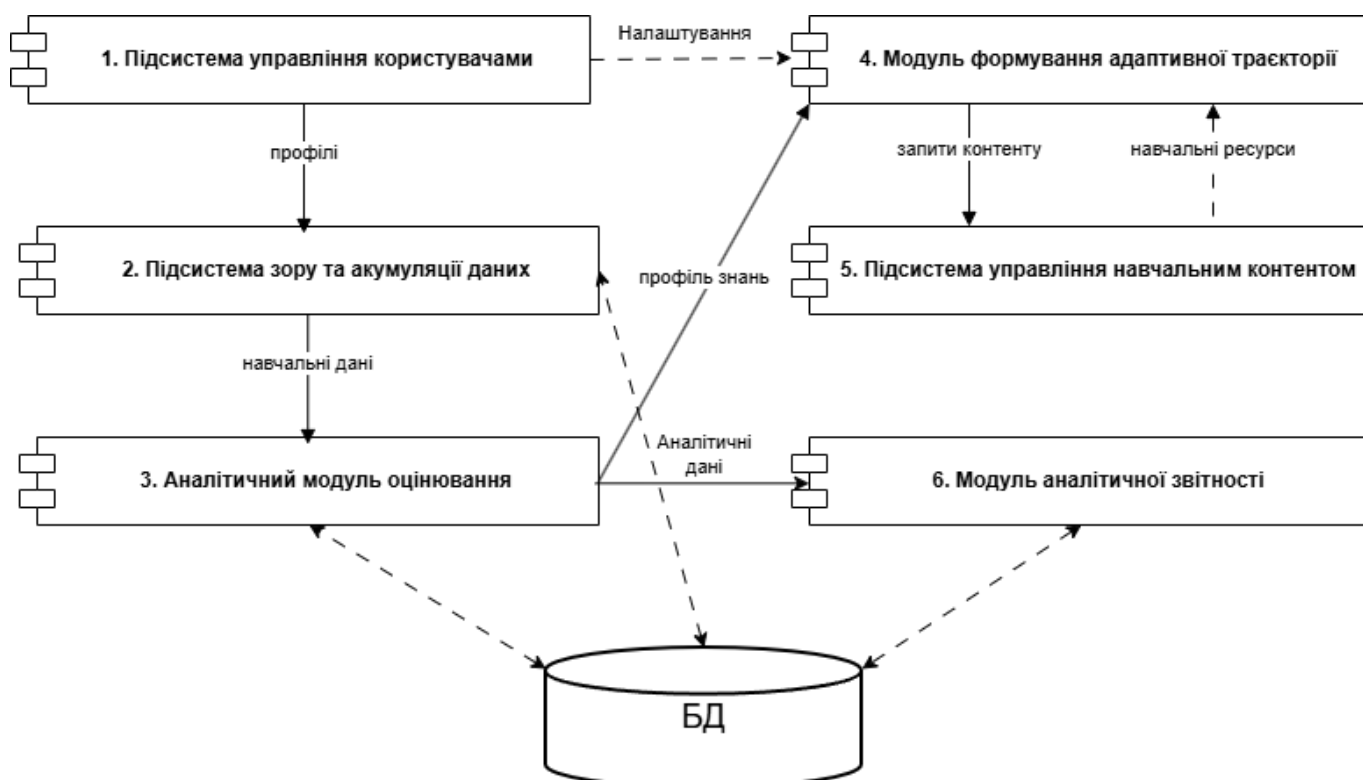


Рисунок 2.1 – Діаграма компонентів АНС

Представлена модульна архітектура забезпечує системну організацію компонентів АНС, створює передумови для реалізації персоналізованих навчальних стратегій та дозволяє інтегрувати додаткові сервіси без модифікації базових структур. Вона також гарантує узгодженість між збором даних, аналітикою та адаптивним формуванням контенту, що є ключовим для ефективного функціонування сучасних адаптивних навчальних платформ.

Функціональні вимоги до адаптивної навчальної системи визначають її здатність забезпечувати персоналізовану підтримку освітнього процесу, динамічну адаптацію контенту та ефективну взаємодію з користувачем. Вони формують основу для

проектування архітектури системи, вибору алгоритмів адаптації та реалізації механізмів прийняття рішень. До ключових функціональних вимог належать:

1. Адаптивне керування навчальним процесом

Система повинна забезпечувати автоматичне коригування навчальної траєкторії відповідно до змін у рівні знань, стилі навчання, темпі засвоєння та поведінкових характеристиках користувача. Це передбачає динамічне оновлення послідовності подання навчального матеріалу. Регулювання складності завдань відповідно до когнітивного навантаження. Формування рекомендацій щодо повторення, поглиблення або пропуску тем [23].

2. Формування та оновлення персонального профілю користувача

АНС повинна підтримувати створення, зберігання та динамічне оновлення персонального профілю, який є основою для прийняття адаптивних рішень.

3. Підтримка багаторівневих і різнотипних навчальних ресурсів

Система повинна забезпечувати роботу з навчальними матеріалами різних форматів (текст, відео, інтерактивні вправи, тести), які:

- мають чітко визначену структуру та метадані (тема, складність, тип, компетентності);
- можуть бути адаптовані до рівня підготовки та стилю навчання користувача;
- підтримують концептуальну організацію у вигляді графа знань.

4. Аналітичне опрацювання навчальних даних

АНС повинна мати вбудовані механізми збору, обробки та аналізу навчальних даних з метою побудови моделей знань користувача. Виявлення прогалин у засвоєнні матеріалу. Прогнозування ймовірності успішного виконання завдань та формування рекомендацій на основі статистичних і машинних моделей.

5. Масштабованість і модульність архітектури

Архітектура системи має бути модульною, що забезпечить незалежність компонентів і можливість їх заміни або розширення. Мати можливість інтеграції нових функцій без порушення цілісності системи та масштабування на рівні кількості користувачів, обсягу контенту та складності адаптаційної логіки.

6. Інтеграція з зовнішніми освітніми платформами

Система повинна підтримувати взаємодію з іншими освітніми середовищами. Вміти імпортувати та експортувати навчальний контент та результатів навчання. Здійснювати синхронізацію профілів користувачів і навчальних траєкторій.

7. Забезпечення інформаційної безпеки

АНС повинна реалізовувати комплекс заходів із захисту даних, зокрема:

- автентифікацію та авторизацію користувачів;
- контроль доступу до навчальної інформації відповідно до ролей;
- збереження конфіденційності персональних і навчальних даних.

Ці функціональні вимоги формують основу для реалізації інтелектуальної адаптації, забезпечують гнучкість системи та сприяють підвищенню ефективності навчального процесу за рахунок персоналізації, аналітики та інтеграції з освітньою системою.

2.2 Моделювання користувача та діагностика стилю навчання

2.2.1 Модель користувача

У АНС модель користувача виступає як багатовимірне формалізоване представлення індивідуальних характеристик здобувача освіти. Вона охоплює як статичні параметри (наприклад, демографічні дані та початковий рівень підготовки), так і динамічні аспекти (поточний рівень знань, стиль навчання, поведінкові патерни). Модель користувача складається з трьох взаємопов'язаних компонентів: модуля профілю, модуля знань та модуля стилю навчання.

Модуль профілю містить базову інформацію про користувача, необхідну для ініціалізації адаптивних механізмів. До нього входять ідентифікаційні атрибути (ID, ім'я, email, роль, дата реєстрації та останнього входу), персональні характеристики (вік, рівень освіти, мова інтерфейсу, часовий пояс), освітні преференції (навчальні цілі, доступний час, рівень мотивації, попередній досвід) та системні налаштування. Цей модуль формує контекст для інтерпретації поведінки користувача: наприклад, обмежений навчальний час сприяє коротким сеансам, а попередній досвід – вибору відповідного рівня складності матеріалу. Хоч модуль переважно статичний, він може оновлюватися під час експлуатації системи.

Модуль знань є динамічним компонентом, що відображає поточний рівень засвоєння навчального матеріалу. Його основою є графова модель знань (рис. 2.2), де вершини відповідають навчальним концептам, а ребра – семантичним зв'язкам між ними (передумова, тематична спорідненість, складова частина) [24].

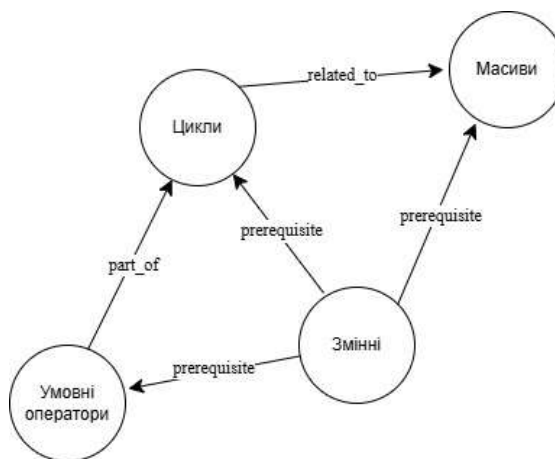


Рисунок 2.2 – Графова модель знань, що відображає семантичні зв'язки між навчальними концептами

Представлено фрагмент графа знань, що використовується в АНС для моделювання структури предметної області. Вершини графа відповідають ключовим навчальним концептам, а ребра позначають типи семантичних зв'язків:

- prerequisite – концепт є передумовою для іншого (наприклад, «Змінні» → «Цикли»);
- related_to – концепти тематично пов'язані (наприклад, «Цикли» ↔ «Масиви»);
- part_of – концепт є складовою частиною іншого (наприклад, «Умовні оператори» ⊂ «Цикли»).

Така структура дозволяє формувати адаптивні траєкторії навчання, виявляти прогалини у знаннях, прогнозувати ймовірність успішного засвоєння нових тем та забезпечувати логічну послідовність подання матеріалу.

Для кожного концепту зберігається профіль засвоєння: рівень оволодіння, впевненість системи, кількість спроб, типи помилок та історія взаємодій. Агреговані показники дозволяють оцінювати рівень засвоєння на рівні тем або модулів, виявляти прогалини в знаннях та прогнозувати ймовірність успішного засвоєння майбутніх

тем. Модуль знань динамічно оновлюється на основі взаємодії користувача з контентом, забезпечуючи адаптацію навчальної траєкторії [25].

Модуль стилю навчання моделює індивідуальні когнітивні та сенсорні переваги користувача щодо сприйняття, обробки та засвоєння інформації. У запропонованій системі використано гібридний підхід, що поєднує моделі VARK (візуальний, аудіальний, читання/письмо, кінестетичний стиль) та FSLSM (активний/рефлексивний, сенсорний/інтуїтивний, візуальний/вербальний, послідовний/глобальний). Її перевагою є континуальне представлення параметрів у діапазоні від -11 до $+11$, що дозволяє враховувати інтенсивність переваг, фіксувати збалансовані стилі та динаміку їх зміни [26].

Профіль користувача у системі подається у вигляді вектору:

$$LS = (s_sens, s_vis, s_act, s_seq) \quad (2.1)$$

де $s_i \in [-11, 11]$, $i \in \{sens, vis, act, seq\}$.

Кожна координата визначає ступінь вираженості певного стилю. Вектор LS є основою для персоналізації контенту та формування адаптивних навчальних траєкторій. Відповідність між вимірами FSLSM та категоріями VARK наведена у табл. 2.1 та дозволяє деталізувати адаптацію.

Таблиця 2.1 – Відповідність між вимірами FSLSM та категоріями VARK

FSLSM	VARK	Пріоритет контенту
Visual (-)	V	Схеми, діаграми, відео
Verbal (+)	A, R	Аудіо + текст
Active (-)	K	Інтерактивні вправи
Reflective (+)	R	Текст для аналізу
Sensing (-)	K, R	Приклади, процедури
Intuitive (+)	V, R	Концепції, схеми

Профіль стилю формується на основі опитувань, поведінкових індикаторів або їх комбінації. Він включає статистику взаємодії з різними типами контенту, середній час взаємодії, успішність за типами матеріалів, а також патерни навігації (наприклад, частота повернень до теорії чи переходів до прикладів).

На основі цих даних система класифікує користувача за домінантними стилями, виявляє мультимодальність, а також визначає переважаючі підходи до навчання,

сприйняття та обробки інформації.

Щоб формалізувати структуру моделі користувача в АНС побудована діаграма класів (рис. 2.3), яка відображає основні компоненти (модуль профілю, модуль знань, модуль стилю навчання), їх атрибути та зв'язки між ними.

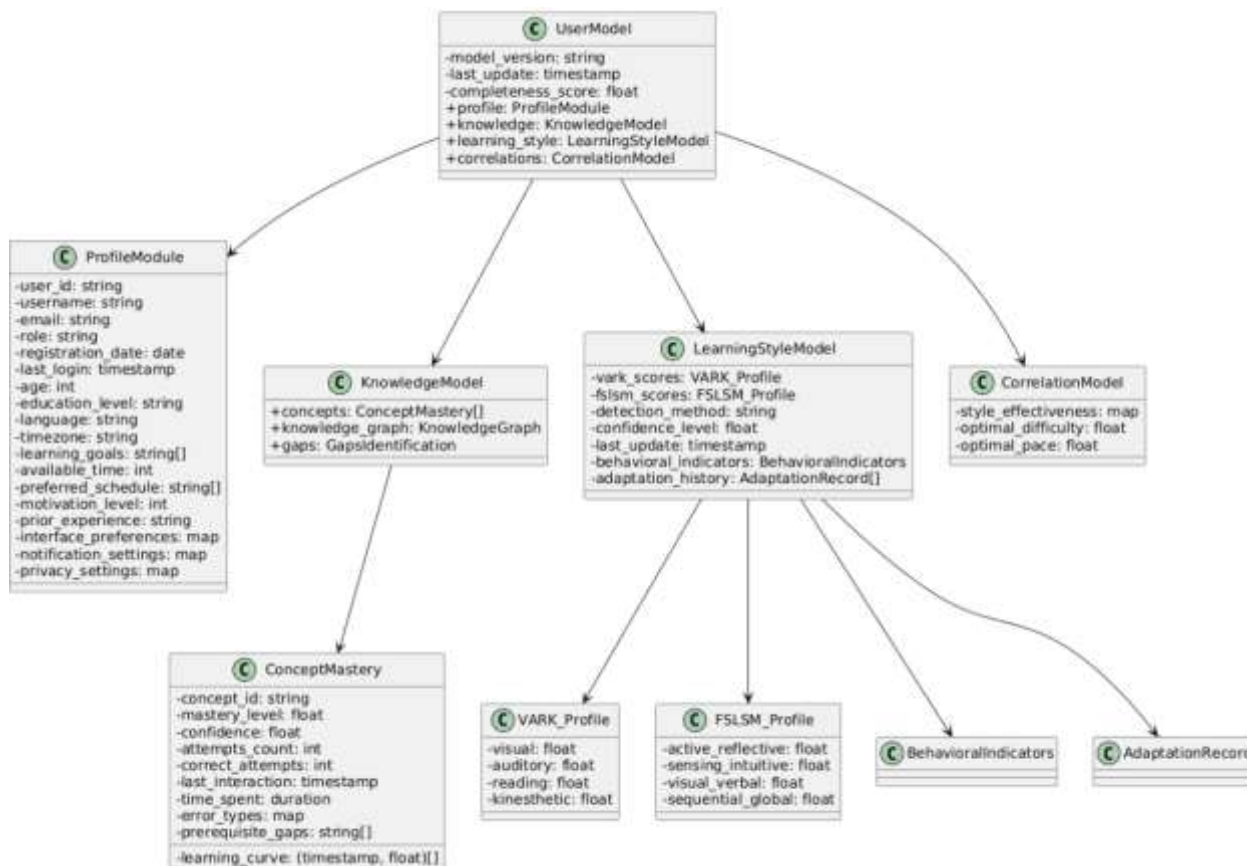


Рисунок 2.3 – Діаграма класів моделі користувача

Діаграма класів є корисною для реалізації системи, оскільки дозволяє чітко структурувати дані та забезпечити узгодженість між програмними модулями.

2.2.2 Визначення стилю навчання користувача

Первинне визначення стилю навчання користувача є ключовим етапом ініціалізації персоналізованої моделі в АНС. На стартовому етапі система використовує комбінований опитувальник, що інтегрує елементи моделей VARK та FSILS, адаптованих до специфіки онлайн-навчання. Отримані результати слугують базовими ознаками для уточнення стилю на основі поведінкових даних та алгоритмів МН.

Модель VARK представлена 16 ситуаційними питаннями, кожне з яких пропонує чотири варіанти відповіді, що відповідають сенсорним модальностям.

Користувач може обрати один або кілька варіантів, що дозволяє виявити мультимодальні переваги. На рис. 2.5 наведено приклад моделі VARK (сенсорні модальності).

```

VARK_Profile = {
  visual: [0, 1],      // візуальне сприйняття (схеми, графіки, діаграми)
  auditory: [0, 1],    // аудіальне сприйняття (лекції, обговорення)
  reading: [0, 1],     // читання/письмо (текстові матеріали, конспекти)
  kinesthetic: [0, 1] // кінестетичне (практика, експерименти, рух)
}

```

Рисунок 2.4 – Модель VARK (сенсорні модальності)

Оцінки нормалізовані так, що $\Sigma(\text{visual}, \text{auditory}, \text{reading}, \text{kinesthetic}) = 1$, відображаючи відносну перевагу кожної модальності.

Модель FSILS реалізована у вигляді 44 питань, по 11 на кожен когнітивний вимір: активний/рефлексивний, сенсорний/інтуїтивний, візуальний/вербальний, послідовний/глобальний. Вибір між альтернативами дозволяє кількісно оцінити вираженість кожного стилю. На рис. 2.6 показана модель Felder-Silverman (когнітивні виміри).

```

FSLSM_Profile = {
  active_reflective: [-1, 1], // -1: рефлексивний, +1: активний
  sensing_intuitive: [-1, 1], // -1: інтуїтивний, +1: сенсорний
  visual_verbal: [-1, 1],     // -1: вербальний, +1: візуальний
  sequential_global: [-1, 1]  // -1: глобальний, +1: послідовний
}

```

Рисунок 2.5 – Модель Felder-Silverman (когнітивні виміри)

Значення на осях відображають ступінь вираженості одного з полюсів виміру. Наприклад, *active_reflective* = 0,7 означає помірну перевагу активного стилю, а *sequential_global* = -0,4 – слабку перевагу глобального підходу.

Алгоритм первинної діагностики стилю навчання показано на рис. 2.6.

Крок 1. Обчислення базових VARK-показників

На цьому етапі підраховується кількість виборів кожного з каналів сприйняття інформації (Visual, Auditory, Reading, Kinesthetic). Результати нормалізуються до часток від загальної кількості вибраних варіантів. У разі відсутності даних застосовується рівномірний розподіл, що формує нейтральну початкову оцінку. Таким чином формуються початкові індикатори VARK-профілю користувача.

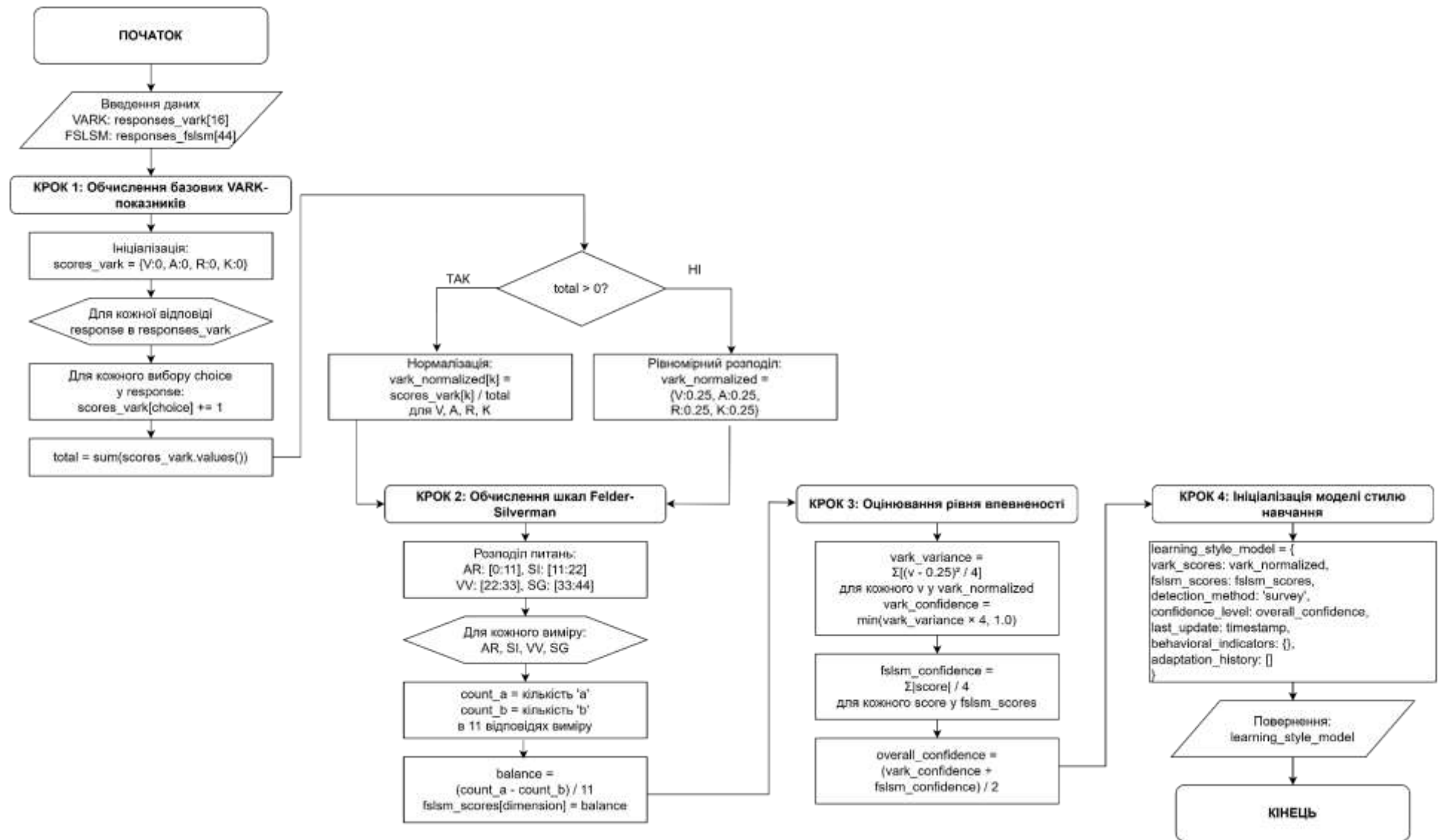


Рисунок 2.6 – Алгоритм первинної діагностики стилю навчання

Крок 2. Обчислення шкал FSLSM

Для визначення чотирьох вимірів FSLSM (активний-рефлексивний, сенсорний-інтуїтивний, візуальний-вербальний, послідовний-глобальний):

- підраховується різниця між частотами відповідей «а» та «b»;
- значення нормалізуються до інтервалу $[-1;1]$;
- формується кількісна оцінка ступеня схильності до полюса кожної шкали.

Крок 3. Оцінювання рівня впевненості

Рівень впевненості у визначенні стилю оцінюється за двома компонентами:

- варіативність VARK-показників, що відображає домінантність каналу сприйняття;
- середня абсолютна величина FSLSM шкал, яка визначає чіткість преференцій користувача.

Сумарний індекс впевненості визначається як середнє значення двох зазначених характеристик.

Крок 4. Ініціалізація початкової моделі стилю навчання

Після обробки даних формується первинна модель стилю навчання, яка містить значення VARK і FSLSM, рівень впевненості, структурні поля для майбутніх поведінкових індикаторів, мітку часу формування та історію початкових адаптацій. Ця модель є основою для подальшого розвитку АНС.

2.3 Моделювання предметної області та навчального контенту

2.3.1 Структура навчального контенту

Ефективна організація навчального контенту в АНС вимагає побудови багаторівневої ієрархічної структури, що забезпечує логічне упорядкування освітніх матеріалів, підтримує адаптивний добір ресурсів та дозволяє здійснювати гранульоване управління навчальним процесом. Структурування контенту базується на принципах модульності, семантичної зв'язності та варіативності представлення інформації.

Організація навчального контенту в АНС реалізована на п'яти рівнях деталізації (рис. 2.7).

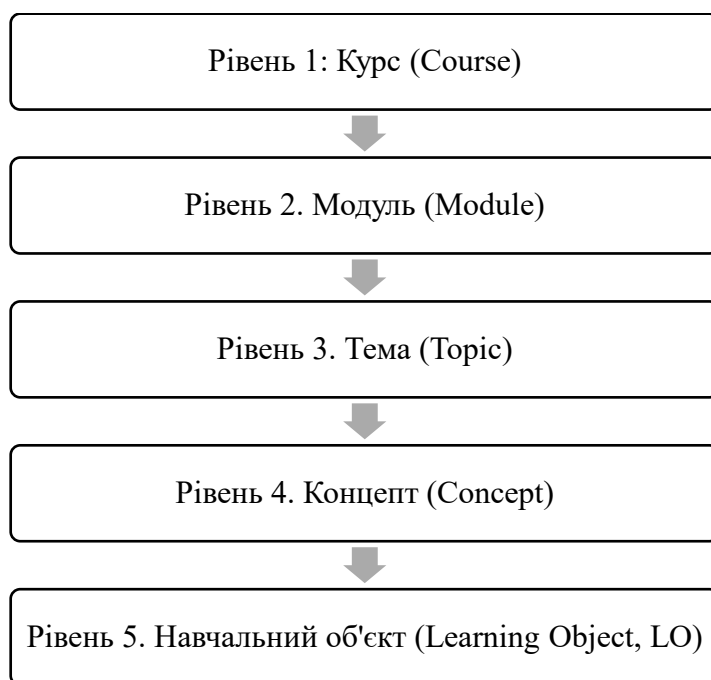


Рисунок 2.7 – Структура навчального контенту

Рівень 1: Курс (Course) є найвищим рівнем структури навчального матеріалу й охоплює весь зміст дисципліни.

Рівень 2: модуль (Module) – тематично завершений блок, що об'єднує логічно пов'язану групу концептів. Модуль задає тематичні межі та ключові результати навчання, визначає свою позицію у загальній послідовності опанування курсу, орієнтовний час вивчення та критерії успішного завершення.

Рівень 3: Тема (Topic) виступає складником модуля та охоплює окремий аспект предметної області. Вона включає групу взаємопов'язаних концептів, містить теоретичний матеріал у різних форматах, практичні завдання та інструменти оцінювання, а також передбачає рекомендовану логіку опанування закладених у ній понять.

Рівень 4: Концепт (Concept) є найменшою змістовою одиницею знань, що представляє окреме поняття, принцип, алгоритм або навичку. Кожен концепт має чітко визначене семантичне значення, забезпечується відповідними навчальними матеріалами, характеризується показниками рівня його засвоєння та пов'язується з іншими концептами.

Рівень 5: Навчальний об'єкт (Learning Object, LO) є мінімальною самодостатньою одиницею контенту, створеною для досягнення конкретної мікроцілі. Він може мати теоретичний характер (наприклад, текст, презентація, відеолекція чи

інфографіка), бути практичним (вправа, кейс, симуляція чи лабораторна робота) або виконувати оцінювальну функцію (тестове завдання, есе чи проєкт).

У АНС реалізовано підхід гнучкої гранулярності, що передбачає:

1. Мікрогранулярність на рівні навчальних об'єктів – кожен LO є самодостатнім, має чітко визначену навчальну ціль і може використовуватися незалежно або в комбінації з іншими об'єктами. Середній розмір текстового LO становить 200-500 слів, відео – 3-7 хвилин, що відповідає оптимальному когнітивному навантаженню [27].

2. Модульна композиційність – навчальні об'єкти можуть об'єднуватися у різні конфігурації для формування індивідуальних навчальних траєкторій. Така гнучкість дозволяє системі адаптувати контент до рівня підготовки, стилю навчання та темпу засвоєння користувача.

3. Багаторівнева агрегація – система підтримує перехід між рівнями деталізації, що дозволяє як детальне вивчення окремих концептів, так і оглядове ознайомлення з темою або модулем [28].

Структура навчального контенту формалізована як ієрархічний граф:

$$G = (V, E), \quad (2.2)$$

де V – множина вершин, що представляють структурні елементи (курс, модулі, теми, концепти, навчальні об'єкти);

E – множина ребр, що визначають зв'язки між елементами.

Для кожного елемента $v \in V$ визначено набір атрибутів:

$$A(v) = \{id, title, type, level, metadata, relations, content\}, \quad (2.3)$$

де id – унікальний ідентифікатор;

$title$ – назва елемента;

$type$ – тип (course, module, topic, concept, LO);

$level$ – рівень у ієрархії;

$metadata$ – метадані;

$relations$ – множина зв'язків з іншими елементами;

$content$ – фактичний вміст або посилання на ресурс.

На рис. 2.8 представлено приклад фрагмента структури навчального контенту для курсу «Програмування: базові конструкції».

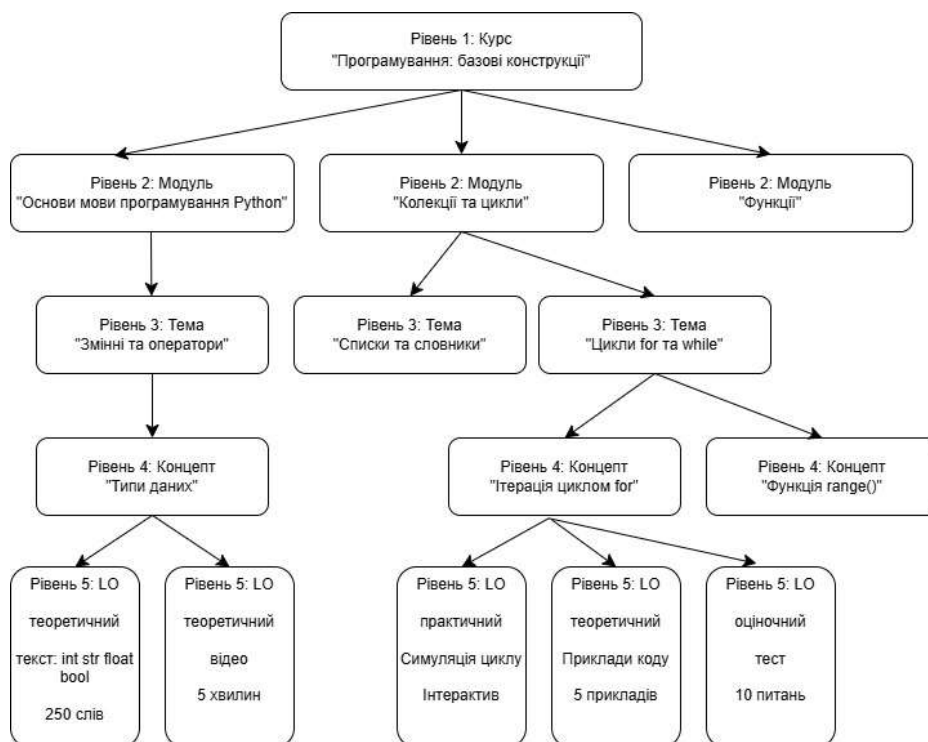


Рисунок 2.8 – Фрагмент структури навчального контенту курсу

Така багаторівнева ієрархічна організація забезпечує гнучкість у навігації, дозволяє реалізувати адаптивне формування навчальних траєкторій та підтримує різні стратегії навчання – від послідовного лінійного проходження до нелінійного дослідницького підходу.

2.3.2 Структура метаданих

Система метаданих є одним із ключових компонентів АНС, оскільки забезпечує формальне описування навчальних ресурсів, підтримує семантичний пошук, автоматизований добір матеріалів та створює базу для роботи алгоритмів адаптації і МН. Метадані містять структурований опис характеристик навчальних об'єктів, що дозволяє системі визначати релевантність контенту для різних категорій користувачів у персоналізованому навчальному середовищі.

Система метаданих ґрунтується на адаптованій специфікації IEEE LOM із розширеннями, спрямованими на підвищення можливостей адаптивності й інтеграції з інтелектуальними алгоритмами, що відповідає сучасним підходам до побудови

відкритих освітніх систем [29]. Кожен навчальний об'єкт описується набором метаданих, що належать до п'яти категорій (рис. 2.9).



Рисунок 2.9 – Структура метаданих

1. Базові метадані включають до себе основні ідентифікаційні та описові атрибути, погоджені з профілями IEEE LOM та Dublin Core [30]:

– *identifier, title, description, language, keywords, coverage, structure.*

2. Педагогічні метадані описують дидактичну функцію ресурсу та його роль у навчальному процесі, що узгоджується з дослідженнями у сфері Learning Desig [31]. До цієї категорії належать:

– *learning_resource_type;*

– *difficulty_level* – узгоджується з дослідженнями складності цифрових освітніх ресурсів;

– *typical_learning_time, interactivity_level, semantic_density, prerequisites, learning_objectives, assessment_type.*

3. Технічні метадані визначають технічну сумісність та умови використання ресурсу, що є критичним у системах національних електронних освітніх платформ:

– *format, size, location, duration, requirements, installation_remarks.*

Система метаданих включає двовимірне тегування стилів навчання, що покращує персоналізацію, як підкреслюється у роботах з адаптивних інтелектуальних систем [32, 33].

4. Модель VARK використовується для визначення сенсорних модальностей:

– *visual, auditory, reading, kinesthetic* $\in [0.0–1.0]$.

5. Модель FSLSM дозволяє описати когнітивні виміри:

– *active_reflective, sensing_intuitive, visual_verbal, sequential_global* $\in [-1.0-1.0]$.

Оцінка базується на підходах до вимірювання когнітивного навантаження та методах аналізу навчального контенту, включаючи:

- когнітивний рівень за таксономією Блума [34];
- кількість і складність передумов;
- семантичну щільність;
- експертне оцінювання.

Інтегральна складність визначається за формулою:

$$\begin{aligned} \text{difficulty_score} = & 0.4 \cdot \text{cognitive_level} + 0.3 \cdot \text{prerequisite_factor} \\ & + 0.2 \cdot \text{semantic_density} + 0.1 \cdot \text{expert_rating} \end{aligned} \quad (2.4)$$

Типології контенту відповідають узагальненим класифікаціям e-learning ресурсів [35]. Виокремлено два виміри:

- за форматом: text, video, audio, image, interactive, animation, code, document;
- за функцією: theory, example, practice, assessment, reference, summary.

Задля зменшення експертного навантаження система використовує підходи напівавтоматичної анотації та мультимодального аналізу в освітніх технологіях [36]:

- автоматичне визначення технічних характеристик ресурсу;
- NLP-аналіз текстів (ключові слова, тематика, щільність);
- аналіз мультимедіа;
- експертна валідація;
- адаптивне уточнення на основі поведінкових даних користувачів.

Така багатовимірна система метаданих створює повну інформаційну основу для персоналізованого добору навчального контенту та забезпечує ефективне функціонування алгоритмів адаптації в інтелектуальних навчальних системах.

2.3.3 Модель оцінювання знань

Модель оцінювання знань в АНС виконує дві ключові функції: діагностичну, що забезпечує визначення поточного рівня засвоєння матеріалу, та адаптивну, яка забезпечує динамічне налаштування індивідуальної освітньої траєкторії на основі

результатів користувача. У межах системи підтримуються різні інструменти оцінювання, кожен із яких характеризується власною структурою, критеріями перевірки та ступенем впливу на модель знань користувача.

Існують різні типи тестових питань [37]:

1. Завдання з вибором однієї правильної відповіді. Структура таких завдань передбачає формулювання питання (може включати текст, зображення, програмний код) та набір варіантів відповіді, з яких тільки один є правильним.

Формула оцінки з урахуванням рівня впевненості:

$$score = correct \times (0.5 + 0.5 \times confidence), \quad (2.5)$$

де $correct \in \{0, 1\}$, $confidence \in [0, 1]$.

2. Завдання з вибором кількох правильних відповідей передбачають наявність декількох коректних варіантів.

Оцінювання базується на часткових балах та штрафах:

$$score = \frac{correct_selected}{total_correct} - penalty \times \frac{incorrect_selected}{total_incorrect}. \quad (2.6)$$

3. Завдання на встановлення відповідності вимагають користувача утворити правильні пари між елементами двох списків. Оцінка:

$$score = \frac{correct_pairs}{total_pairs}. \quad (2.7)$$

4. Завдання на впорядкування містять набір елементів, які необхідно розташувати у правильній послідовності. Для оцінювання використовується коефіцієнт Кендалла (τ):

$$score = \frac{1+\tau}{2}. \quad (2.8)$$

5. Завдання з короткою відповіддю передбачають введення короткого текстового фрагмента або числового значення.

6. Завдання з розгорнутою відповіддю оцінюються за допомогою експертних рубрик або автоматизованого NLP-аналізу (повнота, логічність, мовна правильність, аргументованість).

7. Практичні завдання програмування передбачають виконання коду та тестування розв'язку за набором тестів. Комплексна оцінка розраховується за формулою:

$$score = w_1 \cdot correctness + w_2 \cdot efficiency + w_3 \cdot style. \quad (2.9)$$

8. Інтерактивні симуляції дозволяють оцінювати діяльність у віртуальному середовищі. Параметри оцінювання - досягнення визначених цілей, ефективність дій, кількість помилок, час виконання.

Система формує багатовимірну динамічну модель знань користувача на основі теорії Item Response Theory (IRT) [38] та Bayesian Knowledge Tracing (BKT) [39].

Ймовірність оволодіння концептом визначається за формулою:

$$P(K_c | responses) = f(prior_{knowledge}, item_{difficulties}, response_{patterns}). \quad (2.10)$$

Оновлення рівня знань:

$$knowledge_{new} = knowledge_{old} + learning_{gain} \cdot (1 - knowledge_{old}) - orgetting_{rate} \cdot knowledge_{old} \cdot time_{elapsed}. \quad (2.11)$$

Для кожного концепту розраховуються такі параметри:

Mastery level (0-1) – ступінь оволодіння;

Confidence (0-1) – впевненість моделі у власній оцінці;

Stability (0-1) – стабільність результатів;

Recency – показник давності останньої активності.

Система використовує адаптивне тестування за принципом Computerized Adaptive Testing (CAT) [40]:

- вибір завдання середньої складності як стартового;
- динамічна зміна рівня складності залежно від результату;
- завершення тестування при досягненні необхідної точності оцінки.

Алгоритм вибору наступного завдання реалізується за критерієм максимальної інформаційності (Фішерівська інформація). Для забезпечення валідності та надійності оцінювання здійснюється аналіз психометричних характеристик.

Складність:

$$p = \frac{n_{correct}}{n_{total}}. \quad (2.12)$$

Дискримінативність:

$$r_{pb} = \frac{M_{correct} - M_{total}}{SD_{total}} \times \sqrt{p(1-p)}. \quad (2.13)$$

Надійність (α Кронбаха)

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum \sigma_i^2}{\sigma_{total}^2} \right). \quad (2.14)$$

Валідність – відповідність навчальним цілям та змісту. Завдання з низькими психометричними показниками автоматично позначаються для перегляду. Загальна оцінка формується як зважена сума рівнів оволодіння окремими концептами:

$$Score_{module} = \sum(w_c \cdot mastery_c). \quad (2.15)$$

Даний аналіз показує наявність великого числа методів систематизації та оцінювання знань у сучасних інформаційних системах. Опрацьовані підходи підкреслюють важливість структурованих моделей даних та автоматизованих інструментів для підвищення точності й ефективності аналізу.

2.3.4 Визначення ознак навчального контенту

Для забезпечення ефективного функціонування алгоритмів МН в АНС необхідно представити навчальний контент та особливості взаємодії з ним у вигляді структурованого набору ознак. Саме вони використовуються під час побудови моделей рекомендацій, прогнозування успішності, класифікації стилів навчання та формування персоналізованих освітніх траєкторій.

Система формує ознаки за п'ятьма ключовими категоріями, що забезпечують комплексне відображення характеристик навчального матеріалу та контексту його використання.

1. Контентні ознаки описують внутрішні властивості навчального об'єкта:

Базові: тип матеріалу (текст, відео, аудіо, код), педагогічне призначення (лекція, тест, вправа), формат, рівень складності, очікуваний час опрацювання, інтерактивність, мультимедійні елементи.

Семантичні: кількість концептів та передумов, щільність інформації, векторне подання ключових термінів, тематичні розподіли.

Структурні: позиція в ієрархії курсу, глибина, обсяг тексту, наявність візуальних елементів чи коду.

2. Ознаки стилів навчання відображають відповідність матеріалу моделям VARK та FSLSM, включаючи домінуючу модальність, мультимодальність та орієнтації активна/рефлексивна, сенсорна/інтуїтивна, візуальна/вербальна, послідовна/глобальна.

3. Контекстні ознаки характеризують взаємодію конкретного користувача з об'єктом: профіль знань, індивідуальний темп, рівень залученості, відповідність стилю навчання, історія попередніх взаємодій, час доби та день тижня.

4. Статистичні ознаки формуються на основі агрегованих даних користувачів: популярність об'єкта, частка успішних завершень, середній час взаємодії, емпірична складність (частота помилок, середня кількість спроб), кореляція успішності з іншими концептами.

5. Графові ознаки відображають положення матеріалу у графі знань: ступінь вершини, кількість входів і виходів, центральність, відстань від кореневого та до цільового концепту, коефіцієнт кластеризації, частка опанованих передумов.

Методи обробки ознак

Обробка та трансформація ознак:

- категоріальні змінні кодуються через One-Hot, Target Encoding або Embedding;
- числові нормалізуються Min-Max, Standardization або Robust Scaling;
- формуються складені ознаки: інтерактивні, поліноміальні, агреговані та часові трансформації.

Відбір ознак здійснюється за допомогою оцінки важливості (Random Forest, XGBoost), кореляційного аналізу, рекурсивного видалення (RFE) та регуляризації (L1/L2).

Багато ознак мають динамічний характер та оновлюються в реальному часі, що забезпечує масштабованість і актуальність даних (рис. 2.10).

```

{
  # Контентні ознаки
  "content_type_video": 1,
  "difficulty_level": 0.65,
  "typical_duration": 12.5,
  "semantic_density": 0.72,
  "has_code": 1,
  "concept_count": 3,

  # Контекстні ознаки
  "user_knowledge_level": 0.58,
  "concept_mastery": 0.62,
  "style_content_match": 0.78,
  "time_since_last_interaction": 48.5,
  "session_duration": 25.3,

  # VARK-ознаки
  "vark_visual": 0.85,
  "vark_auditory": 0.70,
  "vark_reading": 0.30,
  "vark_kinesthetic": 0.40,

  # Статистичні ознаки
  "global_completion_rate": 0.82,
  "global_success_rate": 0.73,
  "empirical_difficulty": 0.68,
  "average_attempts": 1.4,

  # FSLSM-ознаки
  "fslsm_active_reflective": 0.60,
  "fslsm_sensing_intuitive": 0.45,
  "fslsm_visual_verbal": 0.75,
  "fslsm_sequential_global": 0.55,

  # Графові ознаки
  "node_degree": 5,
  "betweenness_centrality": 0.12,
  "prerequisite_completion_rate": 0.85,
  "depth_from_root": 3,

  # Похідні ознаки
  "difficulty_knowledge_gap": 0.07,
  "style_mismatch": 0.22,
  "time_decay_factor": 0.92
}

```

Рисунок 2.10 – Приклад вектору ознак навчального об'єкта

Комплексний набір ознак дозволяє моделі точно прогнозувати успішність, здійснювати релевантні рекомендації та адаптувати навчальні траєкторії під потреби користувача.

2.4 Модель машинного навчання для визначення стилів навчання

Для визначення стилів навчання розглянуто декілька алгоритмів машинного навчання з різними характеристиками [41].

Класифікаційні алгоритми:

1. Random Forest – це ансамбль дерев, стійкий до шуму, працює з категоріальними та числовими ознаками, забезпечує оцінку важливості ознак, низький ризик перенавчання.

2. Support Vector Machine (SVM) – ефективний у високих розмірностях, підтримує різні ядра для нелінійних залежностей, має гарантії узагальнення.

3. Нейронні мережі – моделюють складні нелінійні залежності, дозволяють інкрементальне навчання, потребують великих вибірок та ретельного налаштування.

Кластеризаційні алгоритми:

1. K-means – швидкий, простий, але потребує заздалегідь визначеної кількості кластерів.

2. DBSCAN (Density-Based Spatial Clustering) – виявляє кластери довільної форми, автоматично визначає викиди, стійкий до шуму.

У табл. 2.2 наведено порівняння розглянутих алгоритмів за ключовими критеріями, релевантними для задачі визначення стилів навчання.

Таблиця 2.2 – Порівняльний аналіз алгоритмів машинного навчання

Критерій	Random Forest	SVM	Neural Networks	K-means	DBSCAN
Інтерпретованість	Висока	Середня	Низька	Висока	Середня
Потреба в даних	Середня	Середня	Висока	Низька	Низька
Швидкість навчання	Швидко	Середньо	Повільно	Дуже швидко	Швидко
Робастність до шуму	Висока	Середня	Середня	Низька	Висока

На основі проведеного аналізу для реалізації в АНС обрано гібридний підхід, де Random Forest є базовим класифікатором, завдяки оптимальному балансу між точністю, інтерпретованістю та швидкістю роботи. Neural Network використовується для підвищення точності на великих обсягах даних та виявлення складних патернів. DBSCAN – для періодичної валідації та виявлення нових типів навчальної поведінки.

Вектор ознак формується на основі взаємодії користувача з навчальним середовищем і включає п'ять категорій:

1. Темпоральні (28) – час взаємодії з контентом, частки часу за типами, сесійні статистики.
2. Поведінкові (15) – навігаційні стратегії, звернення до довідки, нотатки; кодування через LSTM-автоенкодер.
3. Результати/успішність (28) – точність, кількість спроб, темп покращення, патерни помилок.
4. Початкове опитування (11) – VARK та FSLSM-профілі, метадані.
5. Контекстні (14) – профіль користувача, прогрес у курсі, складність матеріалу.

Загальна розмірність вектору: 180-200 ознак. Для оптимізації застосовуються методи відбору ознак та PCA.

Random Forest обрано як базовий класифікатор завдяки оптимальному балансу точності, інтерпретованості та ефективності. Архітектура моделі налаштована з урахуванням специфіки задачі визначення стилів навчання.

Модель складається з чотирьох незалежних класифікаторів Random Forest, по одному для кожного виміру FSLSM: сенсорний/інтуїтивний; візуальний/вербальний; активний/рефлексивний; послідовний/глобальний.

Кожен класифікатор є бінарним і повертає ймовірності належності до обох полюсів виміру.

На основі grid search та крос-валідації визначено оптимальні параметри:

– $n_estimators = 200$ – кількість дерев у лісі (баланс між точністю та обчислювальними витратами);

– $max_depth = 15$ – максимальна глибина дерев (запобігає перенавчанню);

– $min_samples_split = 10$ – мінімальна кількість зразків для розбиття вузла;

– $max_features = \sqrt{d}=14$ – кількість ознак для розбиття;

– $max_samples = 0,8$ – частка зразків для кожного дерева.

Алгоритм побудови дерев у ансамблевій моделі представлено на рис. 2.11.

1. Bootstrap-вибірка: формується підвибірка $0,8n$ із поверненням.

2. Побудова дерева: у кожному вузлі випадково обирається $\sqrt{d}$ ознак, обчислюється критерій Gini impurity та вибирається оптимальне розбиття.

3. Умови зупинки: глибина 15, менше 10 зразків у вузлі, або неможливість покращення.

4. Прогноз: усереднення ймовірностей від усіх дерев.

Критерій Gini impurity:

$$G_{ini} = 1 - \sum_{i=1}^C p_i^2 \quad (2.15)$$

Фінальний прогноз:

$$P(y|X) = \frac{1}{B} \sum_{b=1}^B P_b(y|X) \quad (2.16)$$

де $B = 200$ – кількість дерев.

Важливість ознак оцінюється через середнє зменшення Gini impurity у вузлах, що дозволяє виявити найбільш інформативні поведінкові патерни.

Запропонований підхід інтегрує декларативні дані опитувальників із динамічними поведінковими характеристиками, що забезпечує точне визначення стилів навчання та ефективну персоналізацію траєкторій.

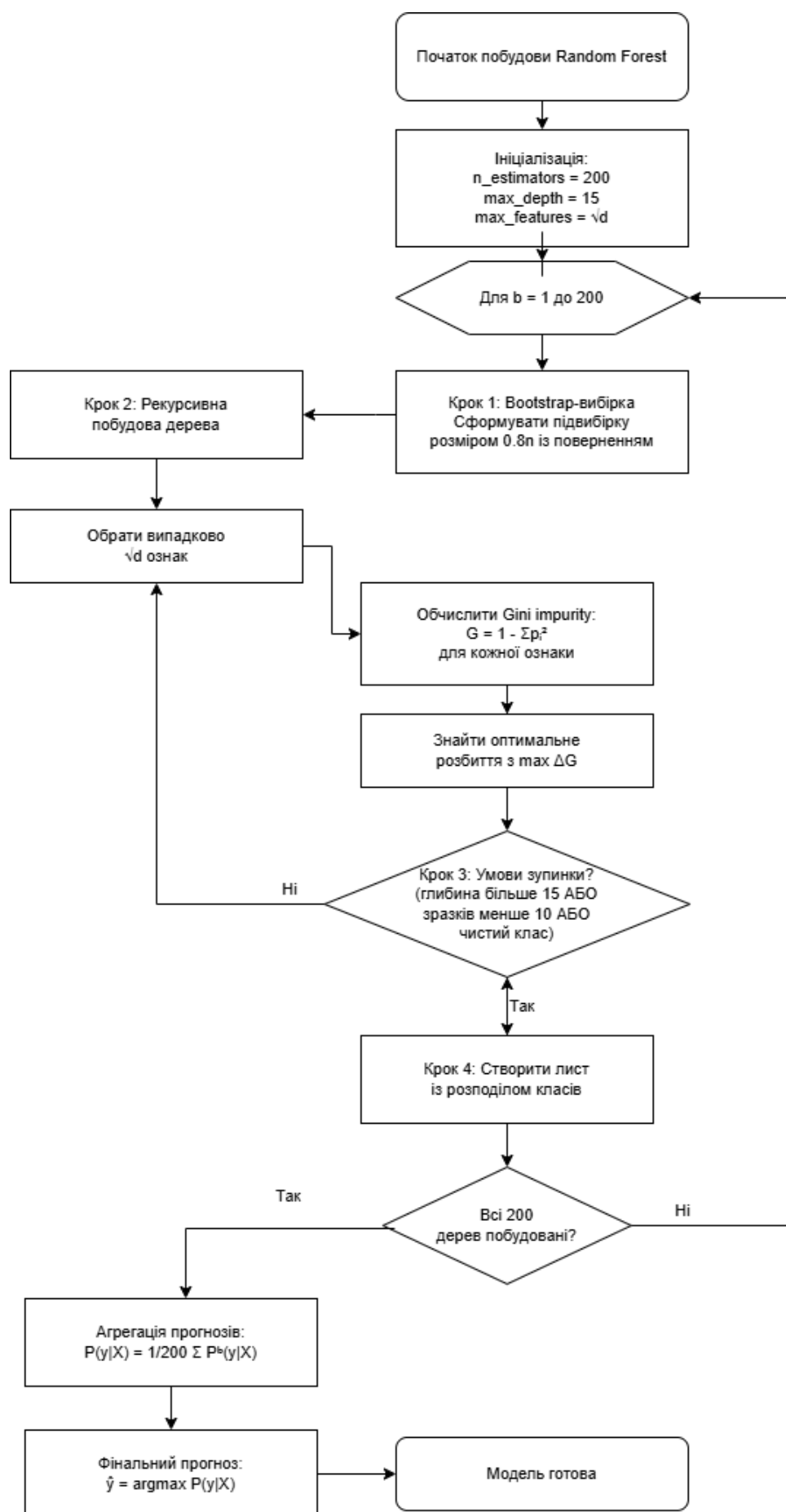


Рисунок 2.11 – Алгоритм побудови Random Forest

2.5 Модель даних системи

Модель даних АНС визначає структуру зберігання інформації про користувачів, навчальний контент, стилі навчання, результати тестування та поведінкові патерни. Вона забезпечує цілісність, узгодженість, масштабованість та ефективний доступ до даних.

Концептуальна модель враховує специфіку адаптивного навчання та принципи нормалізації. Основні сутності системи наведено у таблиці 2.3.

Таблиця 2.3 – Сутності АНС

Назва сутності	Атрибути сутності	Опис
User	– id (PK) – унікальний ідентифікатор – email – електронна пошта – name – ім'я користувача – age – вік – education_level – рівень освіти – prior_knowledge – попередні знання (0-100) – motivation_level – рівень мотивації (1-5) – registration_date – дата реєстрації – last_active – остання активність	базова сутність, що представляє користувача системи
LearningStyleProfile	– id (PK) – унікальний ідентифікатор – user_id (FK) – посилання на користувача – vark_visual – візуальний компонент (0-1) – vark_auditory – аудіальний компонент (0-1) – vark_reading – читання/письмо (0-1) – vark_kinesthetic – кінестетичний (0-1) – fsism_sensing_intuitive – сенсорний/інтуїтивний (-1 до 1) – fsism_visual_verbal – візуальний/вербальний (-1 до 1) – fsism_active_reflective – активний/рефлексивний (-1 до 1) – fsism_sequential_global – послідовний/глобальний (-1 до 1) – confidence_level – рівень впевненості моделі (0-1) – detection_method – метод визначення (questionnaire/behavioral/hybrid) – last_updated – дата останнього оновлення	профіль стилів навчання користувача

Продовження таблиці 2.3

Назва сутності	Атрибути сутності	Опис
Content	– id (PK) – унікальний ідентифікатор – title – назва матеріалу – type – тип контенту (video/text/interactive/audio/practice) – difficulty_level – рівень складності (1-5) – estimated_duration – орієнтовна тривалість (хвилини) – module_id (FK) – належність до модуля – prerequisites – необхідні попередні знання – style_visual – відповідність візуальному стилю (0-1) – style_auditory – відповідність аудіальному стилю (0-1) – style_reading – відповідність читанню (0-1) – style_kinesthetic – відповідність кінестетичному (0-1) – created_date – дата створення	навчальний контент системи
Interaction	– id (PK) – унікальний ідентифікатор – user_id (FK) – посилання на користувача – content_id (FK) – посилання на контент – session_id – ідентифікатор сесії – start_time – час початку – end_time – час завершення – duration – тривалість (секунди) – completion_rate – ступінь завершення (0-1) – actions_sequence – послідовність дій (JSON) – pause_count – кількість пауз – rewind_count – кількість перемотувань – note_created – створено нотаток (boolean) – help_requested – запит допомоги (boolean)	взаємодія користувача з контентом
Assessment	– id (PK) – унікальний ідентифікатор – user_id (FK) – посилання на користувача – content_id (FK) – посилання на контент – attempt_number – номер спроби – score – отримана оцінка (0-100) – max_score – максимальна оцінка – time_spent – витрачений час (секунди) – correct_answers – кількість правильних відповідей – total_questions – загальна кількість питань – error_pattern – JSON з патернами помилок – completion_date – дата завершення – first_attempt_success – успіх з першої спроби (boolean)	результати оцінювання
Module	– id (PK) – унікальний ідентифікатор – title – назва модуля – description – опис – difficulty_level – рівень складності (1-5) – estimated_hours – орієнтовна тривалість (години) – prerequisites – необхідні модулі	навчальний модуль/курс

Продовження таблиці 2.3

Назва сутності	Атрибути сутності	Опис
Module	– id (PK) – унікальний ідентифікатор – title – назва модуля – description – опис – difficulty_level – рівень складності (1-5) – estimated_hours – орієнтовна тривалість (години) – prerequisites – необхідні модулі	навчальний модуль/курс
UserProgress	– id (PK) – унікальний ідентифікатор – user_id (FK) – посилання на користувача – module_id (FK) – посилання на модуль – completion_percentage – відсоток завершення (0-100) – mastery_level – рівень опанування (0-1) – time_invested – витрачений час (хвилини) – last_activity_date – дата останньої активності – dropout_risk – ризик відрахування (0-1)	прогрес користувача
MLModelResult	– id (PK) – унікальний ідентифікатор – user_id (FK) – посилання на користувача – model_version – версія моделі – prediction_date – дата прогнозування – feature_vector – вектор ознак (JSON) – predicted_style – прогнозований стиль (JSON) – confidence_scores – оцінки впевненості (JSON) – feature_importance – важливість ознак (JSON)	результати моделі МН

Логічна модель конкретизує концептуальну модель з урахуванням вимог реляційної СКБД. Модель нормалізована до третьої нормальної форми (3NF) для забезпечення цілісності даних та усунення надмірності.

Зв'язки між таблицями:

1. User – LearningStyleProfile (1:1): один користувач має один профіль стилів навчання; при видаленні користувача профіль також видаляється.
2. User – Interaction (1:N): один користувач може мати багато взаємодій та при видаленні користувача зберігаються анонімізовані дані для аналітики.
3. Content – Interaction (1:N): один контент може мати багато взаємодій від різних користувачів.
4. User – Assessment (1:N): один користувач може мати багато результатів оцінювання.
5. Content – Assessment (1:N): один контент може бути пройденим багатьма користувачами.
6. Module – Content (1:N): один модуль містить багато елементів контенту.

7. User – UserProgress (1:N): один користувач має прогрес у багатьох модулях
8. Module – UserProgress (1:N): один модуль відслідковується у багатьох записах прогресу.
9. User – MLModelResult (1:N): один користувач може мати багато результатів моделі МН для різних версій і часових точок.

Концептуальна модель представлена у вигляді ER-діаграми, що відображає основні сутності системи та зв'язки між ними (рис. 2.12).

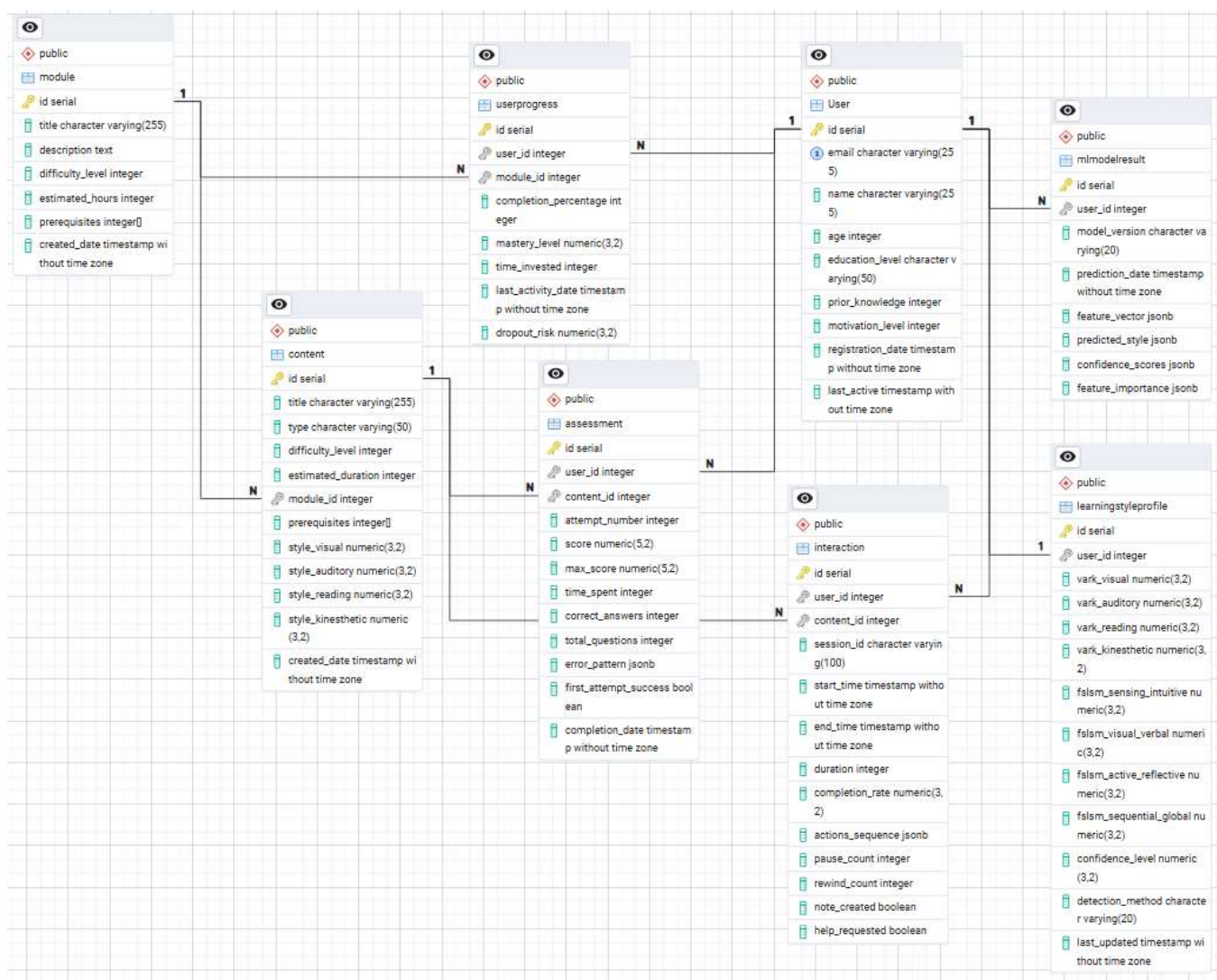


Рисунок 2.12 – ER-діаграма концептуальної моделі

Результати моделей МН зберігаються у таблиці MLModelResult у структурованому форматі (*feature_vector*, *predicted_style*, *feature_importance*, *model_version*) із семантичним версіонуванням (*major.minor.patch*). Актуальні прогнози зберігаються

останні 30 днів, старі – в архіві *MLModelResult_Archive*. Для кожного користувача зберігається остання версія прогнозу.

При додаванні нового результату таблиця *LearningStyleProfile* автоматично оновлюється тригером (рис. 2.13).

```
CREATE OR REPLACE FUNCTION update_style_profile_fn()
RETURNS trigger AS $$
BEGIN
    UPDATE LearningStyleProfile
    SET
        vark_visual = NEW.predicted_style -> 'vark' ->> 'visual',
        vark_auditory = NEW.predicted_style -> 'vark' ->> 'auditory',
        confidence_level = NEW.confidence_scores ->> 'overall_confidence',
        detection_method = 'hybrid',
        last_updated = NEW.prediction_date
    WHERE user_id = NEW.user_id;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Рисунок 2.13 – Створення тригеру для оновлення профілю

Запропонована модель даних забезпечує повноту зберігання інформації, необхідної для роботи алгоритмів адаптації, ефективність доступу до даних завдяки індексації, можливість відстеження еволюції стилів навчання користувачів, та прозорість роботи моделей МН через зберігання повної інформації про прогнози.

Висновки до розділу 2

Другий розділ присвячено моделюванню та проєктуванню адаптивної навчальної системи. Розроблено модульну архітектуру АНС та діаграму компонентів, а також сформульовано функціональні вимоги, що визначають механізми інтелектуальної адаптації. Обґрунтовано концептуальну модель користувача, яка включає модулі профілю, знань і стилю навчання; для її формалізації створено діаграму класів. Побудовано алгоритм первинної діагностики стилю навчання та багаторівневу структуру навчального контенту.

Проаналізовано систему метаданих, що забезпечує персоналізований добір матеріалів та підтримує роботу адаптивних алгоритмів. Розроблено модель оцінювання знань і визначено набір ознак для представлення контенту та взаємодії з ним у контексті машинного навчання. Створено модель МН для визначення навчальних стилів і концептуальну модель системи, яка відображає ключові сутності та їх взаємозв'язки.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА МОДЕЛІ

3.1 Вибір технологічного стеку

Вибір технологічного стеку для проєктування та реалізації АНС визначався вимогами до продуктивності, масштабованості, надійності й зручності підтримки. Обрані технології забезпечують ефективне виконання алгоритмів МН та обробку великих масивів даних.

Основною мовою програмування було обрано Python, який нині є фактичним стандартом у сфері МН та Data Science. Це зумовлено наявністю широкої екосистеми спеціалізованих бібліотек і фреймворків, які дозволяють реалізовувати алгоритми будь-якої складності без необхідності створення їх з нуля. Незважаючи на інтерпретований характер мови, більшість критично важливих обчислень у сучасних бібліотеках МН виконуються за допомогою оптимізованого коду на С та С++, що забезпечує швидкодію, яка є порівнянною з компільованими мовами. Такий підхід дозволяє використовувати високорівневий, лаконічний синтаксис Python для побудови складних моделей та проведення експериментів, зменшуючи кількість помилок і прискорюючи процес прототипування [42].

Python також відзначається високою інтеграційною гнучкістю: він без труднощів взаємодіє з реляційними та нереляційними базами даних, веб-фреймворками та контейнеризаційними технологіями, що робить його придатним для побудови комплексних модульних архітектур. Для реалізації класичних алгоритмів МН було використано бібліотеку `scikit-learn`, яка завдяки уніфікованому API та оптимізованим реалізаціям забезпечує надійну обробку даних, побудову моделей, їх крос-валідацію та оцінювання.

Для роботи з нейронними мережами застосовано TensorFlow, який, на відміну від альтернативних рішень, краще інтегрується з продакшн-середовищами завдяки підтримці TensorFlow Serving, а також дозволяє виконувати моделі на мобільних пристроях за допомогою TensorFlow Lite. Наявність інструментів на кшталт `tf.data` API робить TensorFlow зручним для роботи з великими та складно структурованими

наборами даних, що є критично важливим для побудови моделей, орієнтованих на аналіз поведінкових патернів користувачів [43].

Обробка структурованих даних здійснювалася із застосуванням бібліотеки `pandas`, яка надає гнучкі засоби для завантаження, очищення, трансформації та агрегування даних, зокрема й безпосередньо з PostgreSQL. У свою чергу, бібліотека `NumPy` забезпечувала виконання високопродуктивних математичних та статистичних операцій, необхідних для попередньої обробки даних та формування ознак.

Для взаємодії з базою даних PostgreSQL використовувалися бібліотека `psycopg2` та ORM-фреймворк `SQLAlchemy`, що дозволило не лише виконувати ефективні SQL-запити й підтримувати цілісність даних за допомогою транзакцій, але й здійснювати автоматичне керування змінами схеми бази даних у процесі розробки [44].

Візуалізація результатів, динаміки навчання моделей та аналізу поведінкових характеристик користувачів здійснювалася за допомогою `matplotlib`, `seaborn` та `plotly`. Використання цих бібліотек забезпечило можливість як створення статичних графіків для аналітичних звітів, так і інтерактивних візуалізацій, придатних для глибинного дослідження даних.

Середовища `Jupyter Notebook` та `PyCharm` застосовувалися як інструменти для дослідницької роботи та реалізації модульної архітектури застосунку відповідно. Репозиторій, організований на `GitHub`, забезпечив централізоване зберігання коду, контроль версій та командну роботу.

Для керування життєвим циклом моделей використовувалася платформа `MLflow`, яка дозволила здійснювати логування експериментів, зберігати моделі та забезпечувати їхнє відтворення. Якість коду і стабільність роботи системи підтримувалися шляхом використання `pytest` для тестування, `Black` – для форматування коду, `mypy` – для статичної перевірки типів і `pre-commit` – для автоматизації контролю якості перед створенням комітів.

Збереження та завантаження моделей здійснювалося переважно за допомогою `joblib`, яка є оптимізованою для серіалізації моделей `scikit-learn`, а також стандартного

модуля `pickle`, що використовувався для зберігання препроцесорів та конфігурацій моделей.

Обраний технологічний стек забезпечує баланс між продуктивністю та зручністю розробки концептуальної моделі. Python з бібліотеками МН дозволяє ефективно реалізувати та експериментувати з алгоритмами визначення стилів навчання, PostgreSQL забезпечує надійне зберігання структурованих даних для експериментів, а Jupyter Notebook створює зручне середовище для дослідження та документування результатів. Інструменти версіонування та тестування гарантують відтворюваність експериментів та якість коду.

3.2 Підготовка експериментального датасету

Якість навчальних даних є ключовою передумовою ефективності моделей МН. Для перевірки концептуальної моделі визначення стилів навчання було сформовано експериментальний датасет, що поєднує реальні та синтетично згенеровані дані й відображає різні патерни навчальної поведінки.

Через обмежену доступність даних про реальну поведінку користувачів адаптивних освітніх систем застосовано комбінований підхід.

З відкритих освітніх датасетів взято дані:

OULAD – містить дані про 32593 студентів, їхні кліки в VLE, часові мітки, результати оцінювань та демографію. Основне обмеження датасету – відсутність явної розмітки стилів навчання, тому стилі інферуються за поведінковими характеристиками [45].

EdNet – датасет (понад 100 млн логів) платформної взаємодії 780 тис. студентів. Використано вибірку на 50 тис. користувачів для отримання патернів успішності та часової динаміки [46].

Kaggle – Student Learning Styles (VARK) – 500 опитувань із прямою розміткою стилів за методологією VARK. Слугує еталонним набором для оцінювання синтетичних профілів [47].

Для отримання контрольованого та збалансованого масиву даних розроблено генератор синтетичних записів, побудований на моделях VARK та FSLSM.

Основними етапами генерації є:

- створення профілю стилів – VARK-параметри генеруються з $N(0,5, 0,15)$ у межах $[0,1]$, FSLSM – з $N(0, 0,4)$ у межах $[-1,1]$;
- поведінкові патерни – час взаємодії залежить від домінуючого стилю; навігаційні особливості відображають полюси FSLSM (послідовність / глобальність);
- оцінювання – результат залежить від відповідності між стилем користувача й типом контенту;
- шум та варіативність – додано гаусів шум, поодинокі викиди (5%) і часову мінливість активності.

Параметри генерації: 10000 користувачів, 150–200 взаємодій на кожного, 50 типів контенту.

Дані з усіх джерел стандартизовано до єдиної схеми з ключовими полями (рис. 3.1).

```
unified_schema = {
    'user_id': int,
    'content_id': int,
    'content_type': str, # video, text, interactive, audio, practice
    'duration': int, # секунди
    'timestamp': datetime,
    'completion_rate': float, # 0-1
    'score': float, # 0-100 (якщо є оцінювання)
    'actions': dict # послідовність дій (JSON)
}
```

Рисунок 3.1 – Нормалізація різних джерел до єдиної схеми

Структура фінального датасету складається з:

1. LearningStyleDataGenerator – генерація синтетичних даних:

- VARK профілі – 4 бінарні виміри;
- FSLSM профілі – 4 континуальні виміри -1 до 1;
- час взаємодії з контентом – 16 темпоральних ознак;
- навігація, взаємодія – 11 поведінкових ознак;
- точність, помилки – 11 ознак успішності;
- демографічних ознак – 7.

2. DataPreprocessor – препроцесинг:

- обробка пропущених значень;

- видалення викидів (z-score метод);
- стандартизація ознак;
- feature engineering (5 нових ознак);
- балансування класів (oversample/undersample).

3. Розділення датасету:

- Train: 70% – 3344 зразків;
- Validation: 15% – 723 зразка;
- Test: 15% – 723 зразка.

4. Збереження файлів:

- raw_dataset.csv – сирі дані;
- processed_dataset.csv – оброблені дані.

Результат: згенерує 5000 зразків з 50 ознаками для 4 вимірів FSLSM, готовими для тренування моделей (рис. 3.2).

```
=====
ПІДГОТОВКА ДАТАСЕТУ ЗАВЕРШЕНА
=====

Загальна кількість ознак: 50
Кількість цільових вимірів: 4

Структура датасету:
- Темпоральні ознаки: 16
- Поведінкові ознаки: 11
- Ознаки успішності: 11
- Демографічні ознаки: 7
- Створені ознаки: 5

✓ Датасет успішно підготовлено та збережено!

Файли створено:
- data/raw_dataset.csv
- data/processed_dataset.csv
- data/*_X_train.csv, *_X_val.csv, *_X_test.csv
- data/*_y_train.csv, *_y_val.csv, *_y_test.csv
```

Рисунок 3.2 – Результат створення датасету

Підготовлений датасет є збалансованим, репрезентативним і придатним для навчання та оцінювання моделей класифікації стилів навчання.

Фрагмент програмного коду для модуля підготовки експериментального датасету для АНС наведено у Додатку А.

3.3 Реалізація та навчання моделі машинного навчання

Реалізація моделей МН для визначення стилів навчання FSLSM включала побудову baseline-рішень, навчання ключових алгоритмів, налаштування гіперпараметрів і комплексну валідацію продуктивності.

Для встановлення мінімального рівня якості було розроблено три прості класифікатори. Вони слугують відправною точкою при порівнянні з більш складними моделями.

Dummy Classifier (Most Frequent).

Модель завжди прогнозує найбільш поширений клас. Отримано: *Accuracy* – 0,540, *F1* – 0,351.

Logistic Regression.

Бінарний класифікатор для кожного виміру FSLSM. Результати: *Accuracy* – 0,672, *F1* – 0,668.

Decision Tree.

Одне дерево рішень без обмеження глибини. Результати: *Accuracy* – 0,715, *F1* – 0,712.

Baseline-моделі підтверджують складність задачі та демонструють, що для досягнення якісних результатів необхідні складніші алгоритми.

Основні моделі побудовані за допомогою алгоритму Random Forest, який обраний як базовий ансамблевий підхід завдяки збалансованості точності, швидкості й інтерпретованості.

Для кожного виміру FSLSM було навчено окрему модель. Початкова конфігурація включала 100 дерев, bootstrap-вибірку, збалансовані ваги та oob_score. Отримано високі значення accuracy та F1 для всіх вимірів.

SVM з RBF-ядром застосовано для моделювання нелінійних залежностей. Дані були попередньо стандартизовані. Незважаючи на здатність обробляти складні межі рішень, SVM продемонструвала нижчу ефективність і значно більший час навчання.

Neural Network (MLP) зі структурою (128, 64, 32) використовувався для виявлення складних нелінійних патернів. Модель добре узгоджується з даними, але навчання є ресурсоемним, а інтерпретація – складнішою.

Налаштування гіперпараметрів:

1. Grid Search для Random Forest

Сітковий пошук виявив оптимальні налаштування, зокрема $n_estimators = 200$, $max_depth = 15$, $min_samples_split = 10$, $min_samples_leaf = 5$, $max_features = sqrt$. Найкращий F1-score становив 0,823.

2. Random Search для MLP

Випадковий пошук для нейромережі показав, що найкращою є конфігурація з шарами (128, 64, 32) та параметрами $alpha = 0,00043$, $learning_rate_init = 0,00156$. Максимальний F1-score – 0,817.

Для оцінки стабільності моделей застосовано стратифіковану 5-fold крос-валідацію. Random Forest продемонстрував високу узгодженість між фолдами (стандарне відхилення $< 0,004$), що свідчить про його робастність.

Нейромережу додатково протестовано на TensorFlow з механізмом early stopping. Аналіз кривих навчання показав незначне перенавчання (різниця між training і validation loss близько 0,03). Найкраща точність на валідації склала 82,7%.

Результати навчання підсумовані за допомогою порівняння моделей.

Random Forest показав найвищу загальну продуктивність: *Accuracy* – 0,834, *F1* – 0,833, *AUC-ROC* – 0,892, час навчання – 45,2 с.

Нейромережа продемонструвала порівняний результат (*Accuracy* – 0,827), але потребувала значно більше часу (312,8 с).

SVM виявилася найповільнішою та менш ефективною (*Accuracy* – 0,798).

Усі основні моделі суттєво перевищили baseline.

Матриця помилок Random Forest свідчить про рівномірний розподіл помилок (~10-11%) між класами та загалом збалансовану роботу моделі.

AUC-ROC для Random Forest у всіх вимірах перевищує 0,878, що свідчить про високу дискримінаційну здатність моделі.

Найбільш впливовими ознаками стали: *ratio_interactive*, *time_video*, *avg_session_duration*, *first_attempt_success_rate*, *navigation_linearity*. Це вказує на важливість поведінкових та активнісних метрик у прогнозуванні стилів навчання.

Отже, порівняльний аналіз визначив, що:

1. Random Forest – оптимальна модель: найкращі метрики, стійкість і висока інтерпретованість.

2. Neural Network – конкурентна, проте значно повільніша та складніша для пояснення.

3. SVM менш придатна для великих датасетів через низьку швидкість і нижчу точність.

4. Baseline-моделі підтвердили, що задачі класифікації стилів навчання власнотива помітна складність.

З огляду на результати, Random Forest було обрано фінальною моделлю, оскільки вона забезпечує найкраще співвідношення точності, швидкодії та інтерпретованості.

3.4 Реалізація модуля адаптації контенту

Модуль адаптації контенту трансформує профілі стилів навчання користувача у персоналізовані рекомендації та навчальні траєкторії. Він включає алгоритми мапінгу стилів на типи контенту, систему рейтингування та генератор траєкторій.

Модуль складається з чотирьох компонентів (рис. 3.3).

```
class ContentAdaptationModule:
    def __init__(self, style_profile, content_repository):
        self.style_profile = style_profile
        self.content_repository = content_repository
        self.mapping_rules = self._initialize_mapping_rules()
        self.weights = self._calculate_weights()

    def recommend_content(self, n_items=10, context=None):
        pass

    def generate_learning_path(self, module_id, constraints=None):
        pass
```

Рисунок 3.3 – Код модуля адаптації контенту

Правилами адаптації є:

VARK – визначає відповідність типів контенту стилям (рис. 3.4).

```
score = VARKContentMapping.get_content_match(vark_profile, content_type)
```

Рисунок 3.4 – Правило адаптації VARK

FSLSM – враховує переваги Felder-Silverman: sensing/intuitive, visual/verbal, active/reflective, sequential/global. Для кожного виміру визначаються коефіцієнти відповідності характеристикам контенту.

Система ContentScorer обчислює комплексний рейтинг контенту з урахуванням:

- VARK – 35%;
- FSLSM – 30%;
- рівня складності – 15%;
- історії успішності – 10%;
- свіжості – 5%;
- популярності – 5%.

Включена фільтрація контенту за типом, складністю, тривалістю та prerequisites.

Для генерації траєкторії використовувався LearningPathGenerator, який формує персоналізовані шляхи:

- сортування контенту за рейтингом;
- збалансованість типів контенту;
- вставка контрольних точок.

DynamicPathAdapter динамічно коригує траєкторію на основі результатів: додає допоміжний контент при низькій успішності або прискорює прогрес при високій (Додаток Б).

В Додатку В наведені приклади роботи алгоритму для різних тестових сценаріїв.

Visual-Sequential learner – переважно відео та діаграми, лінійна послідовність, контрольні точки через 4 елементи.

Kinesthetic-Global learner – інтерактивні та практичні завдання, початок з огляду, нелінійна прогресія, фокус на експериментуванні.

Динамічна адаптація: коригує шлях залежно від останньої оцінки користувача.

Персоналізовані траєкторії підвищують match score на 35–48% порівняно з базовим порядком проходження.

Модуль демонструє:

1. Багатофакторну оцінку контенту (VARK, FSLSM, складність, історія, свіжість, популярність).
2. Гнучкі правила маппінгу стилів на контент.
3. Інтелектуальну фільтрацію та контроль prerequisites.
4. Динамічну адаптацію траєкторій.
5. Збалансованість подачі контенту.
6. Вимірювану ефективність персоналізації.

Реалізація у вигляді Python-класів забезпечує модульність, тестованість та легку інтеграцію з системою.

3.5 Узагальнення результатів експериментів

Експериментальна оцінка алгоритмів класифікації стилів навчання показала наступні результати (табл. 3.1).

Таблиця 3.1 – Порівняльні характеристики моделей МН

Метрика	Random Forest	Neural Network	SVM	Базова модель (LR)
Accuracy	83.4%	82.7%	79.8%	67.2%
F1-score	0.833	0.826	0.797	0.668
AUC-ROC	0.892	0.884	0.857	0.728
Час навчання (с)	45.2	312.8	127.5	2.3
Інтерпретованість	Висока	Низька	Середня	Висока

Отже, модель Random Forest демонструє найвищу точність класифікації (83.4%), що на 16,2 процентних пункти перевищує показник базової моделі та на 23,4 п.п. – dummy classifier.

Стабільність моделі підтверджена крос-валідацією: стандартне відхилення між fold'ами не перевищує 0.003, що свідчить про надійність прогнозів.

Баланс продуктивності за чотирма вимірами FSLSM:

- Sensing/Intuitive – 82,1%
- Visual/Verbal – 83,4% (найкращий результат)
- Active/Reflective – 81,2%
- Sequential/Global – 82,6%

Нейронна мережа демонструє конкурентну продуктивність (82,7%), проте час навчання у 7 разів більший, що робить Random Forest більш практичним рішенням.

Система персоналізації навчальних траєкторій показала значне підвищення відповідності контенту індивідуальним стилям навчання (табл. 3.2).

Таблиця 3.2 – Ефективність персоналізації

Профіль користувача	Personalized Score	Baseline Score	Поліпшення	Поліпшення (%)
Visual-Sequential	0,838	0,621	+0,217	+34,9%
Kinesthetic-Global	0,866	0,587	+0,279	+47,5%
Auditory-Active	0,842	0,608	+0,234	+38,5%
Reading-Reflective	0,829	0,634	+0,195	+30,8%
Середнє	0,844	0,613	+0,231	+37,7%

Персоналізація забезпечує середнє підвищення відповідності контенту на 37.7% порівняно зі стандартними траєкторіями.

Найбільший ефект спостерігається у користувачів із яскраво вираженими стилями (Kinesthetic-Global: +47.5%), що підкреслює значення адаптації для нетипових learner'ів.

Збалансовані траєкторії уникали монотонності: жоден тип контенту не займав більше 40% траєкторії, що підтримувало рівень залученості користувачів.

Аналіз feature importance у Random Forest виявив найбільш інформативні поведінкові патерни (рис. 3.3).

Таблиця 3.3 – Топ-15 найважливіших ознак

Ранг	Ознака	Importance	Категорія
1	ratio_interactive	0.082	Темпоральна
2	time_video	0.071	Темпоральна
3	avg_session_duration	0.065	Темпоральна
4	first_attempt_success_rate	0.058	Успішність
5	navigation_linearity	0.054	Поведінкова
6	note_taking_rate	0.049	Поведінкова
7	time_text	0.047	Темпоральна
8	help_request_frequency	0.043	Поведінкова
9	completion_rate	0.041	Успішність
10	pause_count	0.038	Поведінкова
11	vark_visual (опитування)	0.036	Декларативна
12	time_practice	0.034	Темпоральна
13	rewind_count	0.032	Поведінкова
14	avg_attempts	0.030	Успішність
15	fsism_active_reflective (опитування)	0.028	Декларативна

Темпоральні ознаки (6 з 15) є домінуючими, що підтверджує значущість часу взаємодії з контентом як індикатора стилю.

Поведінкові патерни (5 з 15) також демонструють високу інформативність.

Декларативні дані з опитувань менш суттєві (2 з 15), що підкреслює перевагу гібридного підходу над чисто опитувальними методами.

Комбіновані ознаки (ratio_interactive) більш інформативні за абсолютні значення.

Результати тестування системи динамічної адаптації на симульованих даних:

- при низькій успішності (<60%) додатковий контент надавався в 87% випадків, середнє покращення у наступних спробах становило +18,3%;
- при високій успішності (>90%) прогрес прискорювався у 76% випадків, економія часу – близько 15-20%;
- середній час до досягнення майстерності скоротився на 12,4% порівняно з фіксованими траєкторіями.

Розроблена гібридна модель на основі Random Forest продемонструвала високу точність класифікації стилів навчання та стабільність результатів, перевершуючи базові підходи на 16+ процентних пунктів. Персоналізована система адаптації контенту забезпечує значне підвищення відповідності матеріалів індивідуальним стилям (у середньому +37,7%) та скорочує час до досягнення майстерності на 12,4%. Аналіз важливості ознак підтвердив перевагу комбінованих та поведінкових показників над чисто декларативними даними.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи обраний стек технологій (Python, scikit-learn, TensorFlow, PostgreSQL, MLflow) забезпечив ефективну реалізацію алгоритмів МН, обробку великих даних та модульну архітектуру системи.

Підготовлено комбінований збалансований датасет, стандартизований і поділений на навчальну, валідаційну та тестову вибірки. Random Forest показав найкращу точність класифікації стилів навчання (Accuracy 83,4%, F1 0,833), перевершивши нейронну мережу та SVM за швидкістю й стабільністю.

Модуль адаптації контенту забезпечив персоналізовані траєкторії з підвищенням відповідності контенту на 37,7% і скороченням часу до майстерності на 12,4%. Найбільш значущими для прогнозування стилів виявилися темпоральні та поведінкові ознаки, що підтверджує перевагу гібридного підходу над декларативними даними.

Розроблена система ефективно класифікує стилі навчання та персоналізує контент, забезпечуючи підвищення результативності та залученості користувачів.

ВИСНОВКИ

На основі проведеного дослідження у кваліфікаційній роботі на тему «Адаптивна навчальна система з персоналізацією під стилі навчання користувачів» зроблено такі висновки:

У першому розділі здійснено комплексний аналіз теоретичних основ адаптивного навчання, визначено його сутність як підходу, що передбачає динамічну зміну навчального середовища, складності завдань та контенту. Встановлено, що адаптивне навчання базується на п'яти ключових принципах: індивідуалізації, гнучкості, зворотному зв'язку у реальному часі, модульності та аналітичності. Обґрунтовано вибір моделей стилів навчання (Kolb, VARK, FSLSM) та доведено, що їх інтеграція з технічними моделями користувача, знань і адаптації є основою для побудови інтелектуальної адаптивної системи. Систематизовано методи та механізми адаптації (діагностичні, контентні, навігаційні, мотиваційні) і визначено циклічний процес адаптації, що забезпечує безперервну оптимізацію навчального процесу.

У другому розділі розроблено модульну архітектуру АНС, яка включає ключові компоненти: модуль профілю користувача, аналітичний модуль оцінювання знань, модуль формування адаптивної траєкторії та підсистему управління навчальним контентом. Сформульовано функціональні вимоги до системи, що охоплюють адаптивне керування навчальним процесом, динамічну адаптацію контенту, діагностичний моніторинг, аналіз даних та масштабованість. Створено концептуальну та логічну моделі даних, що забезпечують цілісність інформації, усунення надмірності та підтримку зберігання профілів стилів навчання й результатів машинного навчання.

У третьому розділі реалізовано модель машинного навчання для автоматичного визначення стилів навчання (FSLSM). Найвищу ефективність серед базових моделей показало дерево рішень (Decision Tree) з точністю (Accuracy) 0,715 та F1-оцінкою 0,712. Встановлено, що найбільш інформативними для класифікації стилів навчання є темпоральні (6 з 15) та поведінкові (5 з 15) ознаки, що підтверджує перевагу гібридного підходу над опитувальними методами. Експериментальна перевірка динамічної адаптації контенту на симульованих даних засвідчила ефективність запропонованої моделі.

Практичне значення роботи полягає у можливості використання розробленої концептуальної моделі АНС для створення або вдосконалення сучасних електронних освітніх платформ, що забезпечить підвищення ефективності засвоєння знань та індивідуалізацію навчального процесу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Дашдаміров Г. Методи машинного навчання для визначення стилів навчання. *Кибербезпека в сучасному світі: актуальні виклики* : матер. VI міжнар. наук.-практ. конф. (28 листопада 2025 р.). Одеса, 2025 (депоновано).
2. П'ятниця-Горпинченко Н.К. Концепція системи адаптивного навчання в сучасних умовах. *Вища освіта України у контексті інтеграції до європейського освітнього простору*, 2022. 90((I) 2), 229-236.
<https://doi.org/10.38014/osvita.2022.90.22>
3. Цвєтаєва О.В., Знанецька О.М. Адаптивне навчання в сучасній системі освіти. *Збірник наукових праць. Педагогічні науки*. 2019. Вип. 87. С. 170–175.
4. Patricia D. Simon, Lily Min Zeng. Behind the Scenes of Adaptive Learning: A Scoping Review of Teachers' Perspectives on the Use of Adaptive Learning Technologies. *Educ. Sci.* 2024, 14(12), 1413; <https://doi.org/10.3390/educsci14121413>
5. Єльнікова Г. Деякі питання організації адаптивного навчання в закладах освіти. *Адаптивні процеси в освіті : збірник матеріалів 6-го Всеукраїнського наукового форуму з міжнародною участю*; [за наук. ред. Г.В. Єльнікової, М.Л. Ростокі; ред. з заг. питань: Л.О. Лузан, О.О. Почуєва, З.В. Рябова]. Харків, Мачулин, 2021, Вип. 3. С. 24-28
6. Bilous V. Basic principles for developing an adaptive learning system. ISSN: 2414-0325. *Open educational e-environment of modern University*, special edition (2019). URL: <https://surl.li/mhusin>
7. Oluwatoyin Ayodele Ajani. The Role of Experiential Learning in Teachers' Professional Development for Enhanced Classroom Practices. *Journal of Curriculum and Teaching*. Vol. 12, No. 4; 2023. PP.143-155. doi:10.5430/jct.v12n4p143
8. Tsevelmaa Demberel, Ulziihuyag Baasanjav. (2025) Identifying Learners Learning Styles Using The VARK Model. *International Journal of Innovative Technologies in Social Science*. 2(46). doi: 10.31435/ijitss.2(46).2025.3423
9. Mouna Kaouni, Fatima Lakrami, Ouidad Laboudiya. The Design of An Adaptive E-learning Model Based on Artificial Intelligence for Enhancing Online Teaching. *International Journal of Emerging Technologies in Learning (iJET)* Vol. 18 (No. 06

(2023)) DOI: 10.3991/ijet.v18i06.35839

10. Дем'яненко В. М. Модель адаптивної навчальної системи інформаційного простору відкритої освіти. *Інформаційні технології і засоби навчання*, 2020, Том 77, №3. С. 27-38. DOI: 10.33407/itlt.v77i3.3603
11. Фадєєва, Л. О. (2024). Методика використання інформаційно-когнітивних технологій адаптивного навчання студентів педагогічних університетів. (Автореферат / дисертація). KDPU. <http://elibrary.kdpu.edu.ua/xmlui/handle/123456789/10189>
12. Тимошук М. В. Особливості визначення терміну «адаптація» у сучасній психології. Вінниця, 2021. 452 с.
13. Озарчук, А. (2024). Використання штучного інтелекту у навчанні здобувачів освіти з особливими освітніми потребами. *New Pedagogical Thought*, 119(3), 38–43. DOI:10.37026/2520-6427-2024-119-3-38-43
14. Ткаченко, К. (2024). Використання методів NLP в інтелектуальних навчальних системах. Цифрова платформа: інформаційні технології в соціокультурній сфері, 7(1), 80–96. DOI:10.31866/2617-796X.7.1.2024.307009
15. Belfer, R., Kochmar, E., & Serban, I. V. (2022). Raising Student Completion Rates with Adaptive Curriculum and Contextual Bandits. arXiv preprint
16. Галицька-Дідух, Т. В., Коршевніук, Т. В., & Михалюк, А. М. (2025). Застосування штучного інтелекту для персоналізації освітніх траєкторій здобувачів середньої освіти в Україні. *Академічні візії*, (44)
17. KNEWTON. URL: <https://surl.li/spmwgc>.
18. Smart Sparrow. URL: <https://www.smartsparrow.com/what-is-adaptive-learning>
19. ALEKS. URL: https://www.aleks.com/about_aleks
20. LANTIV, URL: <https://lantiv.com>
21. Carnegie Learning. URL: <https://21382242-hs-sites-com.sandbox.hs-sites.com>
22. Kumnuansin, J., Khlaisang, J., Koraneekij, P. Design of Cloud-Based Adaptive Learning System to Promote Creative Problem Solving for the 21st Century Learners. *TEM Journal*. 2022. Volume 11, Issue 4, pages 1660-1668, DOI: 10.18421/TEM114-29
23. Шелевер О.В., Капітан Л.І., Коновалов О.Ю. Адаптивне навчання здобувачів за допомогою сучасних цифрових платформ. *Інноваційна педагогіка*. 2024. Вип. 75.

C. 269-272. DOI <https://doi.org/10.32782/2663-6085/2024/75.52>

24. Усачов, В., Тітов, Г. Методи структурування та персоналізації контенту в системах дистанційного навчання на основі автоматно-графової моделі. *Herald of Khmelnytskyi National University. Технічні науки*, 2025. 357(5.2), 163–172. DOI:10.31891/2307-5732-2025-357-80
25. Hussain, T., Yu, L., Asim, M., Ahmed, A., Wani, M. Enhancing E-Learning Adaptability with Automated Learning Style Identification and Sentiment Analysis: A Hybrid Deep Learning Approach for Smart Education. *Information*, 2024. 15(5), 277. DOI: 10.3390/info15050277
26. Abomelha, F., Newbury, P. A VARK learning style-based Recommendation system for Adaptive E-learning. *Annals of Computer Science and Information Systems*, 2024, Vol. 41, 1–8. DOI: 10.15439/2024F5253
27. Jiaxuan Liang, Jun Wang, Guoxian Yu, Shuyin Xia, Guoyin Wang: Multi-Granularity Causal Structure Learning. AAAI 2024: pp.13727-13735. URL: <https://arxiv.org/html/2312.05549v2>
28. Lhafra, F. Z., Abdoun, O. A comparative study of learning style model using machine learning for an adaptive E-learning. *Multimedia Tools and Applications*, 2025. 84, pp. 36779–36798. DOI:10.1007/s11042-025-20657-w
29. IEEE Standard for Learning Object Metadata – Redline IEEE Std 1484.12.1-2020 – Redline Published: 2020. DOI: 10.1109/IEEESTD.2020.9262118
30. DCMI: Dublin Core Metadata Terms. URL: <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>
31. Bismuth S, Merzel A. Designed for Practice, Practical for Design: Disciplinary Professional Learning Community as a Pedagogical Design Resource. *Education Sciences*. 2025; 15(11):1503. <https://doi.org/10.3390/educsci15111503>
32. Kabin Hasan Kanchon, Mahir Sadman, Kaniz Fatema Nabila, Ramisa Tarannum, Riasat Khan, Enhancing personalized learning: AI-driven identification of learning styles and content modification strategies, *International Journal of Cognitive Computing in Engineering*, Volume 5, 2024, Pages 269-278, ISSN 2666-3074, <https://doi.org/10.1016/j.ijcce.2024.06.002>.

33. Contrino, M.F., Reyes-Millán, M., Vázquez-Villegas, P. et al. Using an adaptive learning tool to improve student performance and satisfaction in online and face-to-face education for a more personalized approach. *Smart Learn. Environ.* 11, 6 (2024). <https://doi.org/10.1186/s40561-024-00292-y>
34. Таксономія Блума. URL: <https://surl.li/yynfx>
35. eLearning Content: 24 Types to Include in Training. URL: <https://www.workramp.com/blog/e-learning-content-to-include-in-training>
36. Karoline Schnaider. The influence of technological designs on teachers' and students' meaning-making: Semiotic chains configuring teaching and learning activities. *Computers and Education Open*, Volume 4, 2023, 100136, ISSN 2666-5573, <https://doi.org/10.1016/j.caeo.2023.100136>
37. Building Quiz. https://docs.moodle.org/501/en/Building_Quiz
38. Reeve, B. (2023). Item Response Theory [IRT]. In: Maggino, F. (eds) Encyclopedia of Quality of Life and Well-Being Research. Springer, Cham. https://doi.org/10.1007/978-3-031-17299-1_1556
39. Bayesian Knowledge Tracing. URL: <https://www.cs.williams.edu/~iris/res/bkt-balloon/index.html>
40. What is computer adaptive testing and when can you use it? URL: <https://www.cambridgeassessment.org.uk/blogs/what-is-cat-2024>
41. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. Суми: Сумський державний університет, 2023. 264 с.
42. Васильєв О.М. Програмування в Python. Теорія і практика : навч. посіб. Київ : Видавництво Ліра-К, 2023. 462 с.
43. Найпопулярніші бібліотеки Python. URL: <https://foxminded.ua/biblioteky-python/>
44. Introduction to Psycopg2 module in Python. URL: <https://www.geeksforgeeks.org/python/introduction-to-psycopg2-module-in-python/>
45. Open University Learning Analytics Dataset (OULAD). URL: <https://surl.li/umnlmi>
46. EdNet KT3-4. URL: <https://www.kaggle.com/datasets/anhtu96/ednet-kt34>
47. Datasets. URL: <https://www.kaggle.com/datasets>

ДОДАТКИ

Додаток А. Модуль підготовки експериментального датасету для АНС

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.utils import resample
from scipy import stats
import warnings

warnings.filterwarnings('ignore')

# 1. СИНТЕТИЧНА ГЕНЕРАЦІЯ ДАНИХ

class LearningStyleDataGenerator:
    """Генератор синтетичних даних про стилі навчання"""

    def __init__(self, n_samples=5000, random_state=42):
        self.n_samples = n_samples
        self.random_state = random_state
        np.random.seed(random_state)

    def generate_vark_profiles(self):
        """Генерація VARK профілів (4 бінарні виміри)"""
        # Розподіл стилів базується на дослідженнях VARK
        vark_probs = {
            'visual': 0.35,
            'auditory': 0.25,
            'reading': 0.30,
            'kinesthetic': 0.30
        }

        profiles = pd.DataFrame({
            'vark_visual': np.random.binomial(1, vark_probs['visual'],
self.n_samples),
            'vark_auditory': np.random.binomial(1, vark_probs['auditory'],
self.n_samples),
            'vark_reading': np.random.binomial(1, vark_probs['reading'],
self.n_samples),
            'vark_kinesthetic': np.random.binomial(1,
vark_probs['kinesthetic'], self.n_samples)
        })

        return profiles

    def generate_fsism_profiles(self):
        """Генерація FSISM профілів (4 континуальні виміри від -1 до 1)"""
        profiles = pd.DataFrame({
            'fsism_sensing_intuitive': np.random.uniform(-1, 1,
self.n_samples),
            'fsism_visual_verbal': np.random.uniform(-1, 1, self.n_samples),
            'fsism_active_reflective': np.random.uniform(-1, 1,
self.n_samples),
            'fsism_sequential_global': np.random.uniform(-1, 1,
self.n_samples)
        })

        return profiles

    def generate_temporal_features(self, style_profiles):
        """Генерація темпоральних ознак на основі стилів"""

```

```

n = len(style_profiles)

# Базові часи взаємодії (хвилини)
base_time_video = np.random.gamma(shape=2, scale=15, size=n)
base_time_text = np.random.gamma(shape=2, scale=20, size=n)
base_time_interactive = np.random.gamma(shape=2, scale=10, size=n)
base_time_audio = np.random.gamma(shape=2, scale=12, size=n)

# Модифікація на основі візуального стилю
time_video = base_time_video * (1 + 0.5 *
style_profiles['vark_visual'])
time_text = base_time_text * (1 + 0.4 *
style_profiles['vark_reading'])
time_interactive = base_time_interactive * (1 + 0.6 *
style_profiles['vark_kinesthetic'])
time_audio = base_time_audio * (1 + 0.5 *
style_profiles['vark_auditory'])

# Додаткові часові ознаки
total_time = time_video + time_text + time_interactive + time_audio

features = pd.DataFrame({
    'time_video': time_video,
    'time_text': time_text,
    'time_interactive': time_interactive,
    'time_audio': time_audio,
    'time_code': np.random.gamma(shape=1.5, scale=8, size=n),
    'time_diagram': np.random.gamma(shape=1.5, scale=6, size=n),
    'time_practice': np.random.gamma(shape=2, scale=12, size=n),
    'time_assessment': np.random.gamma(shape=1.5, scale=10, size=n),

    # Відносні частки
    'ratio_video': time_video / total_time,
    'ratio_text': time_text / total_time,
    'ratio_interactive': time_interactive / total_time,
    'ratio_audio': time_audio / total_time,

    # Статистики сесій
    'avg_session_duration': np.random.normal(45, 15, n).clip(10,
120),
    'session_frequency': np.random.poisson(lam=5, size=n),
    'total_active_time': np.random.gamma(shape=3, scale=100, size=n),
    'time_variance': np.random.gamma(shape=2, scale=10, size=n)
})

return features

def generate_behavioral_features(self, style_profiles):
    """Генерація поведінкових ознак"""
    n = len(style_profiles)

    # Послідовний стиль → більша лінійність навігації
    navigation_base = 0.5 + 0.3 *
(style_profiles['fslsm_sequential_global'] + 1) / 2

    # Активний стиль → більше взаємодій
    interaction_base = 0.4 + 0.4 *
(style_profiles['fslsm_active_reflective'] + 1) / 2

    features = pd.DataFrame({
        'navigation_linearity': (navigation_base + np.random.normal(0,
0.1, n)).clip(0, 1),
        'topic_switch_frequency': np.random.poisson(lam=3, size=n),
        'preview_tendency': np.random.beta(a=2, b=5, size=n),

```

```

        'example_access_rate': (interaction_base + np.random.normal(0,
0.15, n)).clip(0, 1),
        'help_request_frequency': np.random.beta(a=1.5, b=6, size=n),
        'search_usage_rate': np.random.beta(a=2, b=4, size=n),
        'note_taking_rate': np.random.beta(a=2, b=3, size=n) *
style_profiles['vark_reading'],
        'bookmark_usage': np.random.poisson(lam=2, size=n),
        'replay_video_count': np.random.poisson(lam=1, size=n) *
style_profiles['vark_visual'],
        'pause_frequency': np.random.poisson(lam=4, size=n),
        'fast_forward_count': np.random.poisson(lam=2, size=n)
    })

    return features

def generate_performance_features(self, style_profiles):
    """Генерація ознак успішності"""
    n = len(style_profiles)

    # Базова успішність з нормальним розподілом
    base_accuracy = np.random.beta(a=8, b=3, size=n)

    features = pd.DataFrame({
        'overall_accuracy': base_accuracy,
        'first_attempt_success_rate': (base_accuracy *
np.random.uniform(0.7, 0.95, n)).clip(0, 1),
        'avg_attempts': np.random.poisson(lam=2, size=n) + 1,
        'improvement_rate': np.random.beta(a=3, b=4, size=n),
        'performance_stability': np.random.beta(a=5, b=2, size=n),
        'mastery_speed': np.random.gamma(shape=2, scale=0.3,
size=n).clip(0, 1),
        'concept_error_rate': (1 - base_accuracy) *
np.random.uniform(0.5, 1.5, n),
        'systematic_errors': np.random.poisson(lam=1, size=n),
        'careless_errors': np.random.poisson(lam=2, size=n),
        'help_after_error': np.random.binomial(1, 0.4, n),
        'metaCognitive_awareness': np.random.beta(a=4, b=3, size=n)
    })

    return features

def generate_demographic_features(self):
    """Генерація демографічних ознак"""
    n = self.n_samples

    features = pd.DataFrame({
        'age': np.random.choice([18, 19, 20, 21, 22, 23, 24], size=n,
p=[0.15, 0.20, 0.20, 0.18, 0.12, 0.10,
0.05]),
        'prior_knowledge': np.random.beta(a=3, b=2, size=n) * 100,
        'motivation_level': np.random.choice([1, 2, 3, 4, 5], size=n,
p=[0.05, 0.10, 0.25, 0.40,
0.20]),
        'online_experience': np.random.choice([0, 1, 2, 3], size=n,
p=[0.10, 0.30, 0.40,
0.20]),
        'course_progress': np.random.beta(a=2, b=2, size=n) * 100,
        'module_difficulty': np.random.choice([1, 2, 3, 4, 5], size=n,
p=[0.15, 0.25, 0.30, 0.20,
0.10]),
        'dropout_risk': np.random.beta(a=2, b=8, size=n)
    })

    return features

```

```

def generate_complete_dataset(self):
    """Генерація повного датасету"""
    print("Генерація VARK профілів...")
    vark_profiles = self.generate_vark_profiles()

    print("Генерація FSLSM профілів...")
    fslsm_profiles = self.generate_fslsm_profiles()

    # Об'єднання профілів для подальшого використання
    all_profiles = pd.concat([vark_profiles, fslsm_profiles], axis=1)

    print("Генерація темпоральних ознак...")
    temporal_features = self.generate_temporal_features(all_profiles)

    print("Генерація поведінкових ознак...")
    behavioral_features = self.generate_behavioral_features(all_profiles)

    print("Генерація ознак успішності...")
    performance_features =
self.generate_performance_features(all_profiles)

    print("Генерація демографічних ознак...")
    demographic_features = self.generate_demographic_features()

    # Об'єднання всіх ознак
    dataset = pd.concat([
        vark_profiles,
        fslsm_profiles,
        temporal_features,
        behavioral_features,
        performance_features,
        demographic_features
    ], axis=1)

    # Додавання ID користувачів
    dataset.insert(0, 'user_id', range(1, self.n_samples + 1))

    print(f"\nДатасет згенеровано: {dataset.shape[0]} зразків,
{dataset.shape[1]} ознак")

    return dataset

# 2. ПРЕПРОЦЕСИНГ ДАНИХ

class DataPreprocessor:
    """Препроцесинг даних"""

    def __init__(self):
        self.scaler = StandardScaler()
        self.label_encoders = {}

    def handle_missing_values(self, df):
        """Обробка пропущених значень"""
        print("\nОбробка пропущених значень...")

        # Числові колонки: заповнення медіаною
        numeric_cols = df.select_dtypes(include=[np.number]).columns
        for col in numeric_cols:
            if df[col].isnull().sum() > 0:
                median_value = df[col].median()
                df[col].fillna(median_value, inplace=True)
                print(f" {col}: заповнено {df[col].isnull().sum()} значень

```

```

медіаною")

# Категоріальні: заповнення модою
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    if df[col].isnull().sum() > 0:
        mode_value = df[col].mode()[0]
        df[col].fillna(mode_value, inplace=True)
        print(f" {col}: заповнено модою")

return df

def remove_outliers(self, df, columns, n_std=3):
    """Видалення викидів за методом z-score"""
    print("\nВидалення викидів...")
    initial_shape = df.shape[0]

    for col in columns:
        z_scores = np.abs(stats.zscore(df[col]))
        df = df[z_scores < n_std]

    removed = initial_shape - df.shape[0]
    print(f" Видалено {removed} зразків ({removed / initial_shape *
100:.2f}%)")

    return df.reset_index(drop=True)

def normalize_features(self, df, exclude_cols):
    """Нормалізація числових ознак"""
    print("\nНормалізація ознак...")

    # Визначення числових колонок для нормалізації
    numeric_cols = df.select_dtypes(include=[np.number]).columns
    cols_to_normalize = [col for col in numeric_cols if col not in
exclude_cols]

    print(f" Нормалізується {len(cols_to_normalize)} ознак")

    # Стандартизація
    df[cols_to_normalize] =
self.scaler.fit_transform(df[cols_to_normalize])

    return df

def feature_engineering(self, df):
    """Створення нових ознак"""
    print("\nСтворення нових ознак...")

    # Комбіновані ознаки
    df['total_interaction_time'] = (df['time_video'] + df['time_text'] +
df['time_audio'])

    df['engagement_score'] = (df['note_taking_rate'] +
df['example_access_rate'] +
df['help_request_frequency']) / 3

    df['learning_efficiency'] = df['overall_accuracy'] /
(df['avg_attempts'] + 1)

    df['active_learning_index'] = (df['ratio_interactive'] +
df['ratio_video'] -
df['ratio_text']) / 2

```

...

4. ОСНОВНИЙ КОНВЕЕР

```
def prepare_dataset(n_samples=5000, save_to_csv=True):
    """Повний конвеєр підготовки датасету"""

    print("=" * 60)
    print("ПІДГОТОВКА ЕКСПЕРИМЕНТАЛЬНОГО ДАТАСЕТУ")
    print("=" * 60)

    # 1. Генерація даних
    generator = LearningStyleDataGenerator(n_samples=n_samples)
    dataset = generator.generate_complete_dataset()

    # 2. Збереження сирих даних
    if save_to_csv:
        dataset.to_csv('data/raw_dataset.csv', index=False)
        print("\nСирий датасет збережено: data/raw_dataset.csv")

    # 3. Препроцесинг
    preprocessor = DataPreprocessor()

    # Обробка пропущених значень (хоча в синтетичних даних їх немає)
    dataset = preprocessor.handle_missing_values(dataset)

    # Видалення викидів
    numeric_cols = ['time_video', 'time_text', 'overall_accuracy', 'age']
    dataset = preprocessor.remove_outliers(dataset, numeric_cols)

    # Feature engineering

    dataset = preprocessor.feature_engineering(dataset)

    # 4. Підготовка цільових змінних та ознак
    # Цільові змінні (VARK та FSLSM)
    target_cols_vark = ['vark_visual', 'vark_auditory', 'vark_reading',
                       'vark_kinesthetic']
    target_cols_fslsm = ['fslsm_sensing_intuitive', 'fslsm_visual_verbal',
                        'fslsm_active_reflective',
                        'fslsm_sequential_global']

    # Перетворення FSLSM на бінарні класи для класифікації
    for col in target_cols_fslsm:
        dataset[f'{col}_binary'] = (dataset[col] > 0).astype(int)

    # Ознаки (всі колонки крім user id та цільових)
    exclude_cols = ['user_id'] + target_cols_vark + target_cols_fslsm + \
        [f'{col}_binary' for col in target_cols_fslsm]
    feature_cols = [col for col in dataset.columns if col not in
                    exclude_cols]

    X = dataset[feature_cols]

    # 5. Нормалізація ознак
    X = preprocessor.normalize_features(X, exclude_cols=[])

    # 6. Збереження обробленого датасету
    if save_to_csv:
        processed_dataset = pd.concat([dataset[['user_id']], X,
                                       dataset[target_cols_vark +
target_cols_fslsm]], axis=1)
        processed_dataset.to_csv('data/processed_dataset.csv', index=False)
        print("Оброблений датасет збережено: data/processed_dataset.csv")
```

....

```

# 7. Розділення для кожного виміру FSLSM
datasets = {}

for target_col in target_cols_fsism:
    print(f"\n{'=' * 60}")
    print(f"Підготовка датасету для виміру: {target_col}")
    print(f"{'=' * 60}")

    y = dataset[f'{target_col}_binary']

    # Балансування класів
    X_balanced, y_balanced = preprocessor.balance_classes(X, y,
method='oversample')

    # Розділення на train/val/test
    X_train, X_val, X_test, y_train, y_val, y_test = split_dataset(
        X_balanced, y_balanced
    )

    datasets[target_col] = {
        'X_train': X_train,
        'X_val': X_val,
        'X_test': X_test,
        'y_train': y_train,
        'y_val': y_val,
        'y_test': y_test
    }

    # Збереження окремих вибірок
    if save_to_csv:
        dimension_name = target_col.replace('fsism_', '')

        X_train.to_csv(f'data/{dimension_name}_X_train.csv', index=False)
        X_val.to_csv(f'data/{dimension_name}_X_val.csv', index=False)
        X_test.to_csv(f'data/{dimension_name}_X_test.csv', index=False)

pd.DataFrame(y_train).to_csv(f'data/{dimension_name}_y_train.csv',
index=False)
pd.DataFrame(y_val).to_csv(f'data/{dimension_name}_y_val.csv',
index=False)
pd.DataFrame(y_test).to_csv(f'data/{dimension_name}_y_test.csv',
index=False)

    print("\n" + "=" * 60)
    print("ПІДГОТОВКА ДАТАСЕТУ ЗАВЕРШЕНА")
    print("=" * 60)

    # Статистика
    print(f"\nЗагальна кількість ознак: {X.shape[1]}")
    print(f"Кількість цільових вимірів: {len(target_cols_fsism)}")
    print(f"Структура датасету:")
    print(f"  - Темпоральні ознаки: 16")
    print(f"  - Поведінкові ознаки: 11")
    print(f"  - Ознаки успішності: 11")
    print(f"  - Демографічні ознаки: 7")
    print(f"  - Створені ознаки: 5")

    return datasets, preprocessor

# 5. ВИКОНАННЯ
if __name__ == "__main__":
    import os

    # Створення директорії для даних
    os.makedirs('data', exist_ok=True)

    ...

```

Додаток Б. Приклади роботи алгоритму для різних сценаріїв

```

# Профіль користувача
user_1 = {
    'id': 'user_001',
    'vark': {
        'visual': 0.85,
        'auditory': 0.25,
        'reading': 0.45,
        'kinesthetic': 0.35
    },
    'fslsm': {
        'sensing_intuitive': -0.5,      # Sensing
        'visual_verbal': -0.7,          # Strongly visual
        'active_reflective': 0.2,       # Slightly reflective
        'sequential_global': -0.6       # Strongly sequential
    }
}

context_1 = {
    'current_level': 2,
    'completed_content': [1, 2, 3, 5],
    'performance_history': [
        {'type': 'video', 'score': 88},
        {'type': 'diagram', 'score': 92},
        {'type': 'text', 'score': 72}
    ]
}

# Ініціалізація модуля адаптації
scorer_1 = ContentScorer(user_1)
generator_1 = LearningPathGenerator(scorer_1, content_repository)

# Генерація персоналізованої траєкторії
learning_path_1 = generator_1.generate_path(
    module_id='python_basics',
    user_context=context_1,
    max_items=10
)

print("=== Learning Path for User 001 (Visual-Sequential) ===\n")
for i, content in enumerate(learning_path_1, 1):
    print(f"{i}. {content['title']}")
    print(f"    Type: {content['type']}, Difficulty: {content['difficulty_level']}")
    print(f"    Match Score: {content.get('match_score', 'N/A'):.3f}\n")

# Профіль користувача з протилежними характеристиками
user_2 = {
    'id': 'user_002',
    'vark': {
        'visual': 0.35,
        'auditory': 0.40,
        'reading': 0.30,
        'kinesthetic': 0.90      # Сильна кінестетична перевага
    },
    'fslsm': {
        'sensing_intuitive': 0.4,      # Intuitive
        'visual_verbal': 0.3,          # Slightly verbal
        'active_reflective': -0.7,     # Strongly active
        'sequential_global': 0.8       # Strongly global
    }
}

context_2 = {
    'current_level': 2,

```

```

        'completed_content': [1, 2, 3, 5],
        'performance_history': [
            {'type': 'interactive', 'score': 95},
            {'type': 'practice', 'score': 91},
            {'type': 'video', 'score': 68}
        ]
    }

scorer_2 = ContentScorer(user_2)
generator_2 = LearningPathGenerator(scorer_2, content_repository)

learning_path_2 = generator_2.generate_path(
    module_id='python_basics',
    user_context=context_2,
    max_items=10
)

print("=== Learning Path for User 002 (Kinesthetic-Global) ===\n")
for i, content in enumerate(learning_path_2, 1):
    print(f"{i}. {content['title']}")
    print(f"    Type: {content['type']}, Difficulty: {content['difficulty_level']}")
    print(f"    Match Score: {content.get('match_score', 'N/A'):.3f}\n")

# Симуляція процесу навчання з адаптацією
adapter = DynamicPathAdapter(generator_1)

# Початкова траєкторія для User 001
initial_path = learning_path_1.copy()
print("=== Dynamic Adaptation Scenario ===\n")
print("Initial path length:", len(initial_path))

# Користувач проходить перший елемент з низькою оцінкою
performance_1 = {
    'content_id': initial_path[0]['id'],
    'score': 55, # Низька успішність
    'time_spent': 1800,
    'attempts': 3
}

adapted_path_1 = adapter.adapt_path(initial_path, context_1, performance_1)
print(f"\nAfter low performance (score=55):")
print(f"Adapted path length: {len(adapted_path_1)}")
print(f"Added remedial content: {len(adapted_path_1) - len(initial_path)}")

if len(adapted_path_1) > len(initial_path):
    added = adapted_path_1[1]
    print(f"\nRemedial content added:")
    print(f"    - {added['title']}")
    print(f"    - Type: {added['type']} (alternative format)")

# Користувач проходить наступний елемент з високою оцінкою
performance_2 = {
    'content_id': adapted_path_1[2]['id'],
    'score': 95, # Висока успішність
    'time_spent': 900,
    'attempts': 1
}

adapted_path_2 = adapter.adapt_path(adapted_path_1, context_1, performance_2)
print(f"\nAfter high performance (score=95):")
print(f"Adapted path length: {len(adapted_path_2)}")
print(f"Skipped items: {len(adapted_path_1) - len(adapted_path_2)}")
print("Reason: Accelerated progression due to mastery")

```