

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та використання методів формування рекомендацій
для запам'ятовування нових слів в системах навчання іноземним мовам

(тема)

Виконав:

здобувач 2 року навчання, групи ІУСТМ-24-1

Владислав МОРМУЛЬ

(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології

(повна назва освітньої програми)

Керівник: зав. каф. ІУС Костянтин Петров

(посада, власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту

Зав. кафедри ІУС



(підпис)

Костянтин ПЕТРОВ

(власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Інформаційних управляючих систем _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)Освітня програма _____ Інформаційні управляючі системи та технології _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ 24 ” листопада 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**здобувачеві _____ Мормулю Владиславу Віталійовичу _____
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження та використання методів формування рекомендацій для
запам'ятовування нових слів в системах навчання іноземним мовам

затверджена наказом по університету від “ 24 ” листопада 2025 р. № 1055Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії “ 17 ” січня 2025 р.


3. Вихідні дані до роботи науково-технічні публікації та інтернет джерела з тематики
кваліфікаційної роботи, матеріали передатестаційної практики, нормативна документація4. Перелік питань, що потрібно опрацювати у роботі аналіз предметної
області та сучасних інформаційних систем для вивчення іноземних мов і формулювання
вимог до рекомендаційної підсистеми персоналізованого запам'ятовування лексики;
дослідження моделей оцінювання рівня знань користувача (ВКТ, IRT, НММ, LSTM) та
обґрунтувати доцільність побудови комбінованої моделі ВІНЛ; розробка математичних
моделей та алгоритмів оцінювання ймовірності знання слова з урахуванням часових
інтервалів, типів вправ і параметрів складності лексичних одиниць; проектування
структури даних, механізм збору статистики та прототип рекомендаційної підсистеми у
складі багатоплатформної інформаційної системи навчального закладу; експериментальна
перевірка працездатності та ефективності розроблених моделей та рекомендаційних
алгоритмів за обраними метриками й оцінити вплив запропонованої системи на якість
навчання.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	24.11.2025 - 27.11.2025	Виконано
2	Постановка задачі дослідження	28.11.2025	Виконано
3	Дослідження існуючих рішень	28.11.2025 – 30.11.2025	Виконано
4	Формування вимог до методу	30.11.2025 – 01.12.2025	Виконано
5	Реалізація методу формування рекомендацій	02.12.2025 – 04.12.2025	Виконано
6	Експериментальна перевірка працездатності розробленого методу	05.12.2025 – 08.12.2025	Виконано
7	Порівняння запропонованого підходу з існуючими	09.12.2025	Виконано
8	Підготовка пояснювальної записки	01.12.2025 – 12.12.2025	Виконано
9	Підготовка презентації	13.12.2025	Виконано
10	Захист кваліфікаційної роботи	17.12.2025	Виконано

Дата видачі завдання 24 листопада 2025 р.

Здобувач



(підпис)

Керівник роботи



(підпис)

зав. каф. ІУС Костянтин ПЕТРОВ

(посада, власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 128 с., 12 рис., 4 табл., 2 дод., 17 джерел.

РЕКОМЕНДАЦІЙНА СИСТЕМА, ЗАПАМ'ЯТОВУВАННЯ ЛЕКСИКИ, МОДЕЛЮВАННЯ ЗНАНЬ, BAYESIAN KNOWLEDGE TRACING, ITEM RESPONSE THEORY, HIDDEN MARKOV MODEL, LSTM, VINL-МОДЕЛЬ, ІНТЕРВАЛЬНЕ ПОВТОРЕННЯ.

Об'єктом дослідження є процес комп'ютерної підтримки вивчення лексики іноземних мов.

Предметом дослідження є методи формування рекомендацій, що використовуються в інформаційних системах навчання іноземним мовам.

Метою дослідження є розробка комбінованого методу надання рекомендацій, який би на основі історії відповідей користувачів і параметрів слів та вправ формував індивідуальні сценарії навчання.

У роботі проаналізовано існуючі інформаційні системи для вивчення мов і їхні рекомендаційні механізми, розглянуто сучасні моделі knowledge tracing (BKT, IRT, HMM, LSTM) та обґрунтовано доцільність їх інтеграції в єдину комбіновану VINL-модель для оцінювання ймовірності знання слова.

Наукова новизна роботи полягає у розробці комбінованого методу надання рекомендацій, в якому поєднуються моделі BKT, IRT, HMM і LSTM для формування не лише списку слів, а й динамічно адаптованих типів вправ залежно від поточного стану знань користувача.

Практичне значення полягає у створенні прототипу рекомендаційної підсистеми та її інтеграція в багатоплатформного програмного комплексу школи.

ABSTRACT

Master's thesis: 128 pages, 12 figures. 4 tables, 2 appendices, 17 sources.

RECOMMENDER SYSTEM, VOCABULARY MEMORIZATION, KNOWLEDGE MODELING, BAYESIAN KNOWLEDGE TRACING, ITEM RESPONSE THEORY, HIDDEN MARKOV MODEL, LSTM, BIHL MODEL, SPACED REPETITION.

The object of the study is the process of computer-assisted foreign-language vocabulary learning.

The subject of the study is recommendation-generation methods used in information systems for foreign-language learning.

The aim of the study is to develop a combined recommendation method that, based on users' answer histories and the parameters of words and exercises, generates individualized learning scenarios.

The paper analyzes existing language-learning information systems and their recommendation mechanisms, reviews modern knowledge tracing models (BKT, IRT, HMM, LSTM), and substantiates the feasibility of integrating them into a single combined BIHL model to estimate the probability that a word is known.

The scientific novelty of the work lies in the development of a combined recommendation method that integrates BKT, IRT, HMM, and LSTM models to form not only a list of words but also dynamically adapted exercise types depending on the user's current knowledge state.

The practical significance consists in developing a prototype recommendation subsystem and integrating it into the school's cross-platform software suite.

ЗМІСТ

Скорочення та умовні позначки	10
Вступ.....	11
1 Аналіз предметної області та постановка задачі дослідження.....	13
1.1 Сучасні онлайн-сервіси для вивчення іноземних слів та їх обмеження.....	13
1.2 Підходи до моделювання знань користувача в системах e-learning.....	15
1.3 Аналіз існуючих рекомендаційних систем у мовних застосунках	17
1.4 Постановка задачі дослідження та вимоги до ІС рекомендації вправ	20
2 Теоретичні основи моделювання рівня знань та рекомендації вправ	24
2.1 Формалізація процесу засвоєння іноземних слів і критерії «знання слова».....	24
2.2 Математична модель ВКТ з урахуванням забування, випадкове вгадування та помилки попри знання	26
2.2.1. Параметри моделі	27
2.2.2. Модель забування.....	27
2.2.3. Моделювання випадкового вгадування	28
2.2.4. Моделювання помилки попри знання.....	30
2.2.5. Ймовірність правильної відповіді.....	32
2.2.6. Оновлення ймовірності знання за спостереженням	33
2.3 Модель IRT(3PL) для оцінювання здатності користувача та складності завдань	34
2.3.1. Параметри моделі	34
2.3.2. Розрахунок ймовірності відповіді	35
2.3.3. Алгоритм оновлення здібностей користувача.....	35

2.3.4. Калібрування ймовірності знання	36
2.4 Прихована марківська модель (HMM) для переходів між станами знання слова.....	37
2.4.1. Визначення простору станів.....	37
2.4.2. Матриця переходів (Transition).....	38
2.4.3. Оновлення емісії (Emission Update).....	39
2.4.4. Оцінка інтегрального рівня знань.....	40
2.5 Нейромережна модель LSTM для прогнозування результату наступної відповіді.....	41
2.5.1. Векторне представлення контексту навчання.....	41
2.5.2. Навчання та критерії відбору	43
2.5.3. Інтеграція та оцінка знань	43
3. Розробка комбінованого методу надання рекомендацій на основі гібридної моделі VINL.....	45
3.1 Гібридна модель VINL.....	46
3.1.1. Формування інтегральної оцінки.....	46
3.1.2. Адаптивний розрахунок ваг	47
3.2 Формалізація задачі вибору оптимального типу вправи	48
3.2.1. Функція відображення ймовірності у складність вправи	49
3.2.2. Корекція складності з урахуванням забування	50
3.2.3. Формування фінального набору вправ.....	50
4 Розробка інформаційної системи персоналізованої рекомендації вправ	52
4.1 Загальна архітектура системи вивчення слів та взаємодія з багатоплатформеними додатками	52
4.1.1 Клієнтський рівень	53
4.1.2 Серверний рівень.....	53
4.1.3 Рівень даних	55
4.2 Проектування структури набору даних та схеми збору логів користувача.....	55

4.2.1	Концептуальна модель даних.....	56
4.2.2.	Схема збору логів	57
4.3	Попередня обробка і нормалізація даних	58
4.3.1	Стратегія обробки категоріальних та бінарних ознак	58
4.3.2	Нормалізація числових атрибутів зі складним розподілом ...	59
4.3.3	Формування ознак історії навчання	60
4.3.4	Конструювання фінального вектору ознак.....	60
4.4	Програмна реалізація моделей ВКТ, IRT, НММ, LSTM і ВІНЛ..	61
4.4.1	Реалізація ймовірнісних моделей (ВКТ та НММ).....	61
4.4.2	Реалізація психометричної моделі (IRT)	62
4.4.3	Реалізація нейромережевої моделі (LSTM).....	63
4.4.4	Реалізація гібридної моделі (ВІНЛ).....	64
4.5	Алгоритм «Exercise recommendation pipeline» для формування набору вправ сесії.....	64
5	Експериментальна перевірка та оцінка ефективності системи	66
5.1	План експериментів та сценарії використання прототипу системи	66
5.1.1	Сценарій генерації експериментальних даних.....	66
5.1.2	Характеристика набору даних	67
5.1.3	Методологія проведення експерименту.....	68
5.2	Налаштування, навчання та валідація моделей	69
5.2.1	Конфігурація та параметри ймовірнісних моделей	70
5.2.2	Архітектура та навчання LSTM-моделі	70
5.2.3	Валідація та критерії відбору моделі.....	72
5.3	Порівняльний аналіз точності ВКТ, IRT, НММ, LSTM і ВІНЛ за метриками	73
5.4	Оцінка впливу рекомендаційної системи на процес навчання	80
5.4.1	Набір даних та використовувані метрики.....	80
5.4.2	Інтегральні результати навчання з рекомендаціями і без них	81

5.4.3 Вплив на точність виконання вправ різних типів	82
5.4.4 Вплив на кількість додаткових спроб	84
5.4.5 Вплив на засвоєння слів різної складності	86
5.5 Обговорення результатів, обмеження дослідження та напрями подальшого розвитку	87
Висновки	92
Перелік джерел посилання	94
Додаток А Фрагменти коду.....	96
Додаток Б Графічний матеріал кваліфікаційної роботи	115

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система

ВІНЛ – інтегрована модель (ВКТ, ІРТ, НММ, LSTM)

ВКТ – Bayesian Knowledge Tracing

CEFR – Common European Framework of Reference

НММ – Hidden Markov Model

ІРТ – Item Response Theory

LSTM – Long Short-Term Memory

ВСТУП

Актуальність теми дослідження зумовлена швидким розвитком цифрових технологій навчання та зростанням попиту на ефективні інструменти вивчення іноземних мов. Більшість популярних онлайн-сервісів і мобільних застосунків використовують спрощені алгоритми інтервального повторення, які враховують лише факт правильної чи неправильної відповіді та кількість спроб. Це обмежує рівень персоналізації, не дозволяє повною мірою врахувати складність слів, типи вправ, час реакції користувача та інші важливі чинники, що впливають на запам'ятовування лексики. У таких умовах актуальною є розробка рекомендаційної системи нового покоління, яка буде глибшу модель знань користувача та адаптує як вибір слів, так і типи вправ.

Суттєві можливості для цього відкривають сучасні моделі моделювання знань (knowledge tracing), зокрема Bayesian Knowledge Tracing, Item Response Theory, Hidden Markov Model та нейромережеві моделі на основі LSTM. Їх інтеграція в єдину комбіновану модель дає змогу оцінювати ймовірність знання окремих слів точніше, ніж класичні підходи, та використовувати цю оцінку для побудови індивідуальних сценаріїв навчання в багатоплатформній інформаційній системі.

Об'єктом дослідження є процес комп'ютерної підтримки вивчення лексики іноземних мов.

Предметом дослідження є методи формування рекомендацій, що використовуються в інформаційних системах навчання іноземним мовам.

Метою роботи є розробка комбінованого методу надання рекомендацій, який би на основі історії відповідей користувачів і параметрів слів та вправ формував індивідуальні сценарії навчання та демонстрував би переваги над традиційними алгоритмами інтервального повторення.

Практичне значення дослідження полягає у створенні прототипу рекомендаційної підсистеми, інтегрованої в багатоплатформний програмний

комплекс школи та проведеній експериментальної перевірки працездатності запропонованого комбінованого методу, яка підтверджує можливість підвищення ефективності запам'ятовування іноземних слів порівняно з класичними підходами інтервального повторення.

Кваліфікаційна робота виконана відповідно до вимог методичних вказівок щодо організації виконання та захисту магістерських робіт і містить теоретичне обґрунтування, опис розробленого комбінованого методу надання рекомендацій та програмного прототипу, а також результати експериментальної перевірки їх ефективності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Сучасні онлайн-сервіси для вивчення іноземних слів та їх обмеження

Упродовж останніх років онлайн-платформи та мобільні застосунки для вивчення мов (Anki [1], Memrise [2], Quizlet [3] тощо) стали одним із основних інструментів опанування лексики. Вони забезпечують доступ до навчальних матеріалів з будь-якого пристрою, підтримують короткі, але регулярні заняття, використовують елементи гейміфікації (рівні, бали, «серії», нагороди) та алгоритми інтервального повторення [4]. Завдяки цьому користувачі отримують можливість самостійно планувати навчання, а процес запам'ятовування слів стає більш структурованим та привабливим.

Більшість таких сервісів ґрунтується на концепції spaced repetition: кожне слово або картка має власний інтервал повторення, який збільшується або зменшується залежно від успішності попередніх відповідей. У мовних платформах повного циклу інтервальне повторення поєднується з фіксованою послідовністю уроків, діалогами, аудіо- та відеоматеріалами. У системах типу Anki чи Quizlet основний акцент зроблено на універсальних флеш-картах, що дозволяє користувачу самостійно формувати набори слів, додавати зображення та аудіо, експортувати й імпортувати матеріали.

Разом з тим аналіз функціональності цих рішень показує низку обмежень, які є суттєвими з точки зору побудови адаптивної системи навчання.

По-перше, моделювання знань користувача, як правило, зводиться до декількох агрегованих показників (рівень картки, «сила» слова, загальний прогрес), що не дозволяє чітко розрізняти різні типи володіння лексичною одиницею (пасивне впізнавання, активне відтворення, здатність використати слово в контексті). Час реакції, серії послідовних успіхів чи помилок, тип

завдання та контекст його виконання враховуються лише частково або взагалі ігноруються.

По-друге, основною керованою змінною в більшості існуючих сервісів є саме момент повторення. Питання вибору типу вправи для конкретного слова (наприклад, перехід від завдань на вибір правильної відповіді до письмового перекладу «з пам'яті») вирішується за допомогою простих евристик або жорстко зашитих сценаріїв курсу. У результаті траєкторія навчання визначається переважно структурою курсу або набором карток, а не оцінкою поточного стану знань користувача.

По-третє, персоналізація за складністю й контекстом слова часто є обмеженою. Рівень складності зазвичай фіксується на етапі розробки курсу й не оновлюється з урахуванням емпіричних даних про успішність різних груп користувачів. В існуючих продуктах практично не використовується апарат моделей, подібних до Item Response Theory [5], де одночасно оцінюються здатність користувача та параметри завдання.

По-четверте, популярні комерційні платформи функціонують як закриті системи: вони не надають повного доступу до журналів відповідей та параметрів внутрішніх моделей, а також не дозволяють гнучко інтегрувати їхні алгоритми у зовнішні навчальні комплекси. Це обмежує можливість використання таких сервісів як компонента єдиної інформаційної системи навчального закладу та ускладнює проведення незалежних досліджень з порівняння різних підходів.

Таким чином, незважаючи на високу популярність і доведену ефективність базових механізмів інтервального повторення, сучасні онлайн-сервіси для вивчення іноземних слів недостатньо повно розв'язують задачу побудови детальної моделі знань користувача та адаптації типів вправ і структури навчальної сесії. Саме усунення зазначених обмежень і використання більш розвинених моделей knowledge tracing [6] є ключовим напрямом подальших досліджень у межах даної роботи.

1.2 Підходи до моделювання знань користувача в системах e-learning

Одним із центральних завдань інтелектуальних навчальних систем є побудова моделі знань користувача, яка дозволяє відстежувати, що саме студент уже опанував, а які елементи потребують повторення або додаткових пояснень. У дослідженнях це завдання часто описується терміном *knowledge tracing* – відстеження еволюції знань у часі за послідовністю відповідей студента.

Історично перші системи e-learning [7-8] використовували прості евристичні підходи: накопичення балів за правильні відповіді, підрахунок кількості спроб, фіксовані порогові значення для визнання теми «засвоєною». У рамках підходу *mastery learning* вважалося, що студент досяг майстерності, якщо його успішність за останню серію завдань перевищує наперед заданий поріг. Такі моделі прості в реалізації, однак слабо враховують випадковість відповідей, не розрізняють різні типи помилок і не дають явної ймовірнісної оцінки стану знань.

Подальший розвиток привів до поширення ймовірнісних моделей, насамперед *Bayesian Knowledge Tracing* (BKT) [9]. BKT розглядає знання окремого вміння (*skill*, знання концепції) як приховану бінарну змінну («опановано / не опановано») і за допомогою байєсівського оновлення оцінює ймовірність того, що студент опанував це вміння після кожної відповіді. Модель має невелику кількість параметрів (початковий рівень знань, ймовірність навчання, ймовірності вгадування та помилки при володінні навичкою), що пояснює її популярність та інтерпретованість в освітньому аналізі даних.

Окремий клас підходів становлять моделі, що походять з психометрії, насамперед *Item Response Theory* (IRT). На відміну від BKT, яка переважно фокусується на динаміці засвоєння окремого вміння, IRT описує співвідношення між латентною здатністю студента та параметрами завдання

(складність, дискримінація, іноді ймовірність вгадування). На основі відповідей багатьох студентів оцінюються як параметри завдань, так і параметри здатності окремих користувачів, що дозволяє будувати персоналізовані траєкторії навчання й адаптивне тестування. У літературі описано приклади використання IRT для побудови персоналізованих e-learning систем та генерації банків завдань із відомими параметрами складності.

Ще один напрямок досліджень пов'язаний із прихованими марківськими моделями та байєсівськими мережами. ВКТ можна розглядати як окремий випадок прихованої марківської моделі з двома станами знання, однак існують розширення з більшою кількістю рівнів засвоєння, а також моделі, що описують топологію вмінь у вигляді мережі (ієрархії чи графа залежностей). Такі підходи дозволяють явно моделювати переходи між кількома рівнями володіння навичкою та враховувати взаємозв'язки між різними вміннями, але зазвичай є складнішими для налаштування та обчислювально витратнішими.

З розвитком освітньої аналітики і появою великих обсягів логів навчальних систем почали активно застосовуватися моделі машинного навчання та глибинні нейронні мережі. Найбільш відомим прикладом є Deep Knowledge Tracing (DKT), де послідовність взаємодій студента моделюється рекурентною нейронною мережею (часто LSTM) [10]. Така модель безпосередньо прогнозує ймовірність правильної відповіді на наступне завдання, спираючись на всю попередню історію, і в низці досліджень демонструє кращу точність прогнозування порівняно з класичним ВКТ. Надалі з'явилась велика кількість модифікацій DKT [11], зокрема моделі з механізмами уваги, розрідженими представленнями та спеціальними архітектурами для обробки освітніх даних.

Паралельно розвиваються гібридні та ансамблеві підходи, які намагаються поєднати інтерпретованість класичних моделей (ВКТ, IRT) із високою прогностичною здатністю глибинних мереж. Прикладами є моделі на кшталт ВКТ-LSTM, де параметри ВКТ використовуються як ознаки для

LSTM, або фреймворки, що інтегрують IRT із DKT для одночасної оцінки здатності студента та складності завдань при побудові персоналізованих навчальних траєкторій.

Отже, в сучасних системах e-learning можна виділити кілька основних груп підходів до моделювання знань користувача: від простих евристичних схем і класичних ймовірнісних моделей (BKT, HMM [12]) до психометричних (IRT) та глибинних нейронних методів (DKT на основі LSTM та його розширення). Кожен із підходів має власні переваги й обмеження щодо інтерпретованості, точності прогнозування та обчислювальної складності. У подальших розділах роботи детально розглядаються й адаптуються для задачі вивчення лексики моделі BKT, IRT, HMM, LSTM, а також їхня комбінована інтеграція в єдину рекомендаційну схему.

1.3 Аналіз існуючих рекомендаційних систем у мовних застосунках

Рекомендаційні системи у мовних застосунках можна розглядати як підсистеми, що автоматично визначають, що, коли та в якій формі показати користувачу: які слова або теми вивчати далі, коли повертатися до вже вивченого матеріалу, який тип вправи використати для закріплення. У більшості сучасних рішень ці функції реалізуються через поєднання фіксованих навчальних траєкторій (курси, «дерево навичок») та алгоритмів інтервального повторення.

У масових мовних застосунках (Duolingo, Memrise, Babbel тощо) базовою одиницею рекомендації є, як правило, урок або набір слів, що входить до певної теми чи рівня. Користувач рухається за попередньо спроектованою структурою курсу, а система поступово «відкриває» нові модулі після досягнення мінімального рівня успішності в попередніх. Додатково більшість платформ мають окремий режим повторення (practice / review), де набір

завдань формується на основі історії минулих відповідей та часу, що минув від останнього звернення до матеріалу. Дослідження мобільних мовних застосунків показують, що така комбінація лінійного курсу з адаптивними режимами повторення є типовою для Duolingo, Memrise та Babbel.

Окрему групу становлять системи, у яких основний «двигун» рекомендацій – модель інтервального повторення, що працює на рівні окремих слів чи карток. У Duolingo для цього використовується модель Half-Life Regression (HLR) – тренуваний алгоритм, який оцінює «період напіврозпаду» знання слова в довгостроковій пам'яті й прогнозує ймовірність правильного відтворення залежно від історії відповідей та часу від останньої практики. На основі цієї ймовірності система вирішує, коли саме знову показати слово користувачу.

У Memrise рекомендаційний механізм також побудований навколо spaced repetition: алгоритм прогнозує момент, коли конкретне слово «випадає» з довгострокової пам'яті, і планує наступний перегляд. Інтервали (кілька годин, діб, тижнів тощо) збільшуються після правильних відповідей і зменшуються після помилок, що дозволяє частіше повторювати складні елементи й рідше повертатися до легких. Аналогічні принципи реалізуються в універсальних флешкард-системах (Anki, подібні SRS-рішення), де рекомендаційна система фактично зводиться до планувальника повторень, а вибір контенту здійснює сам користувач.

У наукових роботах, присвячених адаптивному вивченню лексики, пропонуються й інші підходи до побудови рекомендацій, які частково вже знаходять відображення в мобільних застосунках:

- прогностичні моделі на основі кривих забування (HLR, adaptive forgetting curves), що оптимізують момент повторення;
- оптимізаційні та reinforcement learning підходи, де вибір моменту перегляду формулюється як задача максимізації довгострокової успішності при мінімальній кількості повторень;
- моделі knowledge tracing (DAS3H та інші), які враховують структуру

навичок і типи слів для більш точного планування практики.

Попри різноманіття алгоритмів, більшість реальних мовних застосунків на практиці фокусуються переважно на виборі часу повторення і базовому фільтруванні слів за темами та рівнями. Рекомендація формату вправи (наприклад, тест з вибором однієї відповіді, встановлення відповідностей, переклад, написання з пам'яті) часто задається фіксованим сценарієм уроку й слабо пов'язана з оцінкою поточного стану знань користувача. Окремі платформи вводять різні режими (наприклад, «слухання», «говоріння», «набір тексту»), однак вибір режиму здебільшого покладається на самого користувача й не є результатом роботи повноцінної рекомендаційної підсистеми.

Ще одним обмеженням існуючих рекомендаційних систем є закритість внутрішніх моделей. Алгоритми Duolingo, Memrise, Babbel та інших комерційних продуктів, як правило, недоступні для зовнішньої інтеграції: навчальні заклади не можуть безпосередньо використовувати їхні оцінки знань у власних інформаційних системах, змінювати правила добору вправ або розширювати набір ознак, що враховуються при прийнятті рішень. Це ускладнює побудову єдиних комплексних рішень на рівні школи чи університету, де мовний застосунок є лише одним із компонентів навчального процесу.

Таким чином, аналіз існуючих рекомендаційних систем у мовних застосунках показує, що:

- у центрі більшості рішень знаходиться механізм інтервального повторення, який добре вирішує задачу планування часу повторення слів;
- адаптація за змістом (добір конкретних слів і тем) здебільшого спирається на фіксовані курси й прості евристики складності;
- питання вибору типів вправ і структури сесії на основі детальної моделі знань користувача залишається недостатньо опрацьованим.

У подальших підрозділах роботи основна увага приділяється побудові комбінованої моделі знань на базі ВКТ, IRT, НММ та LSTM і розробці

рекомендаційної схеми, яка перетворює оцінку ймовірності знання слова не лише на рішення «коли повторювати», але й на вибір оптимального типу вправи та рівня складності в межах інтегрованої інформаційної системи навчального закладу.

1.4 Постановка задачі дослідження та вимоги до ІС рекомендації вправ

Сучасні ІС вивчення іноземних мов переважно використовують алгоритми інтервального повторення, які визначають, які слова і коли потрібно повторити. При цьому тип вправи для слова здебільшого залишається фіксованим і мало залежить від реального рівня знань користувача. Для школи, де навчання базується на поетапному опануванні лексики та потребує гнучкого керування сценаріями занять, виникає потреба в рекомендаційній системі нового типу, яка не лише обирає слова, а й адаптивно добирає вправи на основі історії відповідей користувачів.

Об'єктом дослідження є процес комп'ютерної підтримки вивчення лексики іноземних мов.

Предметом дослідження є методи формування рекомендацій, що використовуються в інформаційних системах навчання іноземним мовам.

Метою роботи є розробка комбінованого методу надання рекомендацій, який би на основі історії відповідей користувачів і параметрів слів та вправ формував індивідуальні сценарії навчання та демонстрував би переваги над традиційними алгоритмами інтервального повторення.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні онлайн-сервіси для вивчення іноземних слів та виявити їх основні обмеження щодо персоналізації навчання;
- дослідити підходи до моделювання знань користувача в системах е-

learning (ВКТ, IRT, НММ, нейромережеві моделі) та оцінити можливості їх застосування для задачі добору вправ;

- виконати аналіз існуючих рекомендаційних систем у мовних застосунках і сформулювати вимоги до рекомендаційної підсистеми багатоплатформної інформаційної системи;

- розробити математичні моделі оцінювання ймовірності знання слова на основі ВКТ, IRT, НММ та LSTM у контексті задачі навчання лексики;

- синтезувати комбіновану ВІНЛ-модель, що інтегрує результати окремих моделей у єдину узгоджену оцінку рівня знань користувача;

- розробити комбінований метод формування рекомендацій для запам'ятовування нових слів іноземної мови, що базується на використанні комбінованої моделі ВІНЛ для оцінки ймовірності знання слова та узгодити її з механізмом калібрування прогнозів (попередні результати та апробацію підходу наведено в [13]);

- спроектувати структуру даних і формат журналів взаємодії користувача з навчальною системою, необхідні для навчання та оцінювання моделей;

- розробити алгоритм рекомендацій [14], який перетворює оцінену ймовірність знання слова на вибір рівня складності та типів вправ, і реалізувати програмний прототип рекомендаційної підсистеми;

- провести експериментальну перевірку працездатності запропонованого комбінованого метода та розробленої на його основі рекомендаційної системи, а також порівняти його з базовими підходами інтервального повторення та оцінити вплив на ефективність засвоєння лексики.

Сформулюємо основні вимоги до ІС рекомендації вправ.

Рекомендаційна підсистема має бути інтегрована в багатоплатформну інформаційну систему навчального закладу та працювати поверх уже наявних модулів зберігання словника, виконання вправ і обліку результатів. З урахуванням цього сформульовано функціональні, інформаційні та

нефункціональні вимоги до системи.

Функціональні вимоги до рекомендаційної підсистеми полягають у тому, що вона повинна:

- автоматизовано збирати та зберігати журнали взаємодії користувача зі словами та вправами (правильність відповіді, час реакції, тип вправи, рівень складності, кількість спроб тощо);

- для кожного користувача й кожного слова будувати послідовність спостережень, що використовуватиметься моделями ВКТ, IRT, НММ та LSTM для оцінювання ймовірності знання;

- виконувати перетворення вихідних даних (нормалізація часових інтервалів, складності слів і вправ, кодування типів завдань, обчислення показників серій успіхів/помилки);

- обчислювати для кожної моделі окрему оцінку ймовірності знання слова та інтегрувати їх у єдину BINL-оцінку з використанням адаптивних ваг;

- на основі отриманої ймовірності знання відносити кожне слово до певного рівня складності сесії та підбирати відповідний тип вправи (тест з вибором відповіді, встановлення відповідностей, конструювання слова з літер, введення з пам'яті тощо);

- формувати збалансовану навчальну сесію з набору слів, забезпечуючи поєднання простих, середніх і складних завдань та враховуючи давність останнього повторення;

- надавати користувачу та викладачу інструменти перегляду індивідуального прогресу (динаміка ймовірності знання слів, частота повторень, типові помилки);

- підтримувати можливість ручного коригування рекомендацій (примусове додавання/вилучення слів зі сесії, зміна типу вправи викладачем).

Інформаційні вимоги відображають особливості даних, з якими працює система, та передбачають, що ІС повинна:

- зберігати централізований словник із прив'язкою до рівнів CEFR, тематичних категорій, мовних пар і додаткових атрибутів (довжина слова,

частотність, наявність озвучки та зображень);

- забезпечувати уніфікований формат логів спроб користувачів, придатний як для оперативної рекомендації, так і для офлайн-аналізу та донавчання моделей;

- підтримувати механізм анонімізації або псевдонімізації користувачів для коректної обробки персональних даних.

Нефункціональні вимоги визначають обмеження щодо продуктивності, надійності та масштабованості системи. У цьому контексті рекомендаційна підсистема повинна:

- забезпечувати коректну роботу в багатоплатформному середовищі (мобільні застосунки, веб-інтерфейс, десктопні клієнти) без зміни алгоритмічної частини;

- гарантувати прийнятний час формування рекомендацій для сесії (у межах декількох секунд при типових обсягах даних);

- підтримувати масштабованість за кількістю користувачів і обсягом історичних логів без суттєвого погіршення якості рекомендацій;

- забезпечувати надійність зберігання даних та відмовостійкість сервісів, що відповідають за оцінювання знань і генерацію вправ;

- мати модульну архітектуру, яка дозволяє додавати нові моделі knowledge tracing або змінювати схему агрегування без переробки всієї системи.

З урахуванням наведених вимог задачу дослідження можна сформулювати так: необхідно дослідити існуючі та розробити методи й моделі надання рекомендацій, що будуть використовуватися для реалізації підсистеми інформаційної системи вивчення лексики іноземних мов, що на основі журналів відповідей користувачів і параметрів слів та вправ оцінюють поточний рівень знань для кожного слова та автоматично формують індивідуальні набори й типи вправ, забезпечуючи підвищення ефективності запам'ятовування порівняно з традиційними алгоритмами інтервального повторення.

2 ТЕОРЕТИЧНІ ОСНОВИ МОДЕЛЮВАННЯ РІВНЯ ЗНАНЬ ТА РЕКОМЕНДАЦІЇ ВПРАВ

2.1 Формалізація процесу засвоєння іноземних слів і критерії «знання слова»

У роботі процес засвоєння іноземної лексики розглядається як послідовність взаємодій користувача з інформаційною системою, кожна з яких фіксується у вигляді записів у базі даних. На основі цих записів будуються моделі моделювання знань (knowledge tracing) та працює рекомендаційна підсистема. У системі виділяються такі основні сутності:

- користувач, який навчається у багатоплатформній системі;
- слово (лексична одиниця), пов'язана зі словниковою інформацією;
- вправа, у межах якої слово подається користувачу;
- навчальна сесія (транзакція), що об'єднує набір вправ і слів;
- окрема завершена спроба виконання завдання по слову.

Ці сутності відображаються такими структурами даних:

- `StatisticalWordEntity` описує статичні характеристики слова для конкретного користувача: ідентифікатори `wordId` та `userWordId`, поточний рівень опанування `grade`, дату додавання та останнього повторення, оригінальне слово та переклад, мови, рівень `CEFR`, наявність опису, звуку, зображення, категорію тощо;

- `ExerciseTransactionEntity` описує навчальну сесію: користувача (`userId`), унікальний ідентифікатор транзакції (`transactionId`), перелік типів вправ (`exercises`), список слів (`words`), час початку й завершення (`createdAt`, `endedAt`), кількість доданих і повторюваних слів (`WordCountEntity`), а також параметри навчального плану (`LearningPlanEntity` – кількість слів на день, мовна пара, рівень `CEFR`, час останнього оновлення плану);

- `WordCompletedEntity` фіксує факт завершення завдання по конкретному слову: ідентифікатор транзакції, ідентифікатори слова (`wordId`,

userWordId), тип вправи (exerciseId), кількість спроб (attempts), правильність (isCorrect), час завершення (completedAt);

– ExerciseHistory є агрегованим поданням історії виконання завдань по слову: містить ідентифікатори (id, wordId, userId), текст слова (wordText), час початку та завершення (executeStartTime, executeEndTime), правильність (correctness), рівень CEFR, тип вправи (exercise), позицію слова у вправі та розмір завдання (positionInExercise, wordsInExercise), довжину слова (wordLength), наявність зображення та звуку (hasImage, hasSound), мови (wordLanguage, translateLanguage), кількість спроб (attempts).

Послідовність усіх записів ExerciseHistory і WordCompletedEntity для пари «користувач – слово» утворює траєкторію засвоєння цього слова. Саме на основі таких траєкторій моделі ВКТ, IRT, HMM, LSTM і комбінована VINL-модель обчислюють ймовірність того, що користувач реально знає це слово. У подальшому ця величина називається «ймовірністю знання слова» для конкретного користувача.

Для практичної роботи інформаційної системи однієї лише числової оцінки недостатньо. Необхідно мати чіткі критерії, коли слово вважати «знаним», які слова віднести до різних рівнів складності та які типи вправ для них підбирати.

У роботі застосовуються такі принципи:

– слово вважається «знаним» у вузькому (бінарному) розумінні, якщо інтегрована оцінка ймовірності знання, отримана на основі VINL-моделі, перевищує заданий поріг протягом певного періоду, а останні спроби за цим словом у даних ExerciseHistory / WordCompletedEntity є правильними;

– додатковою умовою є наявність кількох останніх правильних відповідей у вправах продуктивного типу (написання слова, переклад «з пам'яті») без використання підказок, причому середній час відповіді не перевищує наперед визначеного порогу для відповідного рівня CEFR;

– для слів, які ще перебувають у процесі засвоєння, вводяться кілька проміжних рівнів «знаності» (низький, середній, високий), які відповідають

інтервалам значень ймовірності знання. Для визначення цих рівнів додатково використовуються поля `grade`, `lastReadDate` та характер останніх спроб у `ExerciseHistory` і `WordCompletedEntity`.

Таким чином, «знання слова» у даному дослідженні розуміється як динамічна ймовірнісна характеристика, що змінюється з часом під впливом нових взаємодій користувача із системою.

Кінцеві критерії визнання слова «знаним» поєднують:

- інтегровану оцінку ймовірності знання, обчислену на основі кількох моделей моделювання знань;
- якість останніх відповідей (типи вправ, кількість спроб, наявність чи відсутність підказок);
- часові параметри (тривалість відповіді, давність останнього повторення, динаміку параметра `grade`).

У подальших підрозділах на основі наведеної формалізації будуть описані конкретні математичні моделі, способи їх навчання на послідовностях `ExerciseHistory`, а також правила переходу від оцінки ймовірності знання до рівнів складності L1–L5 і вибору типів вправ у рекомендаційній підсистемі.

2.2 Математична модель ВКТ з урахуванням забування, випадкове вгадування та помилки попри знання

Bayesian Knowledge Tracing (ВКТ) – це ймовірнісна модель відстеження рівня засвоєння окремої одиниці знань (у нашому випадку – слова) за послідовністю відповідей користувача. Модель описує прихований стан «знає / не знає» та оновлює ймовірність знання після кожної спроби, враховуючи навчання, забування, випадкове вгадування (`guess`) та помилки попри знання (`slip`).

Вхідними даними для моделі є такі історичні данні:

- $y_t \in \{0,1\}$ – правильність відповіді (1 – правильно, 0 – помилка);
- Execute Start Time – дата початку виконання вправи над словом;
- Word Difficulty(CEFR Level) – складність слова;
- Exercise Type – тип вправи;
- Position in exercise – позиція слова в вправі;
- Word length – довжина слова.

2.2.1. Параметри моделі

Класична ВКТ використовує чотири параметри:

- K_0 – початкова ймовірність знання слова до першої спроби;
- K – ймовірність того, що користувач знає слово;
- T – ймовірність «навчитися» слову після однієї спроби (коефіцієнт навчання, *learn*);
- $G(guess)$ – ймовірність правильної відповіді, якщо слово ще не відоме;
- $S(slip)$ – ймовірність помилки, навіть якщо слово відоме.
- Для урахування забування вводиться додатковий параметр:
- λ – коефіцієнт забування (швидкість експоненційного спаду пам'яті).

2.2.2. Модель забування

Забування описуємо експоненційною функцією часу. Нехай h – кількість днів, через яку ймовірність знання зменшується вдвічі (період «напівзабування»). В даній роботі h має значення 7 днів. Для розрахування забування використовується формула:

$$\lambda_t = 1 - 2^{-\frac{\Delta t_n}{h}}, \quad (2.1)$$

де λ_t швидкість забування;

Δt_n інтервал часу між спробами, але для першої спроби значення набуває $\Delta t_n = 0$, для інших спроб розраховується як $t_n - t_{n-1}$.

Якщо перед спробою $n - 1$ ймовірність знання була P_{n-1} , то перед спробою n , з урахуванням лише забування, маємо облік забування:

$$\tilde{K}_{n-1} = K_{n-1} \cdot e^{-\lambda \Delta t_n}, \quad (2.2)$$

де K_{n-1} ймовірність знання слова на минулій спробі.

Така форма відповідає класичним моделям забування Еббінгауза та їх сучасним узагальненням.

2.2.3. Моделювання випадкового вгадування

Ймовірність правильної відповіді (Guess, G) за умови, що користувач ще не знає слово, але може випадково вгадати відповідь завдяки формату вправи, залежить від типу завдання.

Для вправи вибору однієї відповіді (multiple choice) розраховується на основі n_{opt} кількості варіантів. Формула розрахування:

$$G = \frac{1}{n_{opt}}, \quad (2.3)$$

де G ймовірність вгадування;

n_{opt} кількість варіантів відповіді.

Наприклад, якщо є чотири варіанти відповіді, то ймовірність вгадати буде дорівнювати $\frac{1}{4}$.

У вправах типу “зібрати слово з літер” користувач отримує набір літер і повинен скласти правильне слово за одну спробу. Якщо він не знає слова, шанс випадково зібрати правильну послідовність визначається кількістю можливих перестановок. Нехай:

- m – загальна кількість літер у слові;
- c – кратності повторень літер (наприклад, у слові LETTER дві Т, дві Е);
- користувач робить одну спробу скласти слово;
- усі перестановки букв рівно ймовірні.

Тоді ймовірність випадково скласти слово дорівнює:

$$G = \frac{c!}{m!}, \quad (2.4)$$

де G ймовірність вгадування.

Це формула для числа сприятливих перестановок (одна правильна послідовність) поділена на кількість рівноймовірних перестановок із повтореннями. Приклади розрахунку:

- $m = 5$;
- літери “Р” повторюються двічі $\rightarrow c = 2$.

Тоді:

$$G = \frac{2!}{5!} = \frac{2}{120} = \frac{1}{60} \approx 0.0167$$

Отже, за відсутності знань шанс випадково скласти слово $\approx 1.67\%$.

У вправах типу “Введення слова з клавіатури (free input)” ймовірність випадкового повного збігу дуже мала, тому вважаємо $G \approx 0.01$.

Для більшої точності можна використати таку формулу:

$$G = 1 / A^L, \quad (2.5)$$

де A – розмір алфавіту;

L – довжина слова.

Для слова з п'яти літер ймовірність буде $1/26^5$ (26 – це кількість літер в англійському алфавіті) $\approx 8.4 \times 10^{-8}$. Значення дуже низьке, через береться значення 0.01.

У вправах типу “Вправи на встановлення відповідностей (matching)” кількість доступних пар зменшується в процесі виконання, тому G залежить від числа пар, що залишилися. Якщо на кроці користувач бачить k ще незіставлених пар, то

$$G = 1 / k, \quad (2.6)$$

2.2.4. Моделювання помилки попри знання

Ймовірність неправильної відповіді, навіть якщо користувач знає слово (описка, неуважність, технічна помилка тощо) позначимо (Slip, S). В літературі типові значення знаходяться в інтервалі 0,05–0,15. Початкове значення для нових користувачів або слів з недостатньою історією встановлюється $S_0=0,10$. У реальних даних S не є константою:

- одні користувачі роблять більше випадкових помилок;
- різні слова мають різний ризик «плутанини»;
- тип завдання впливає на точність.

Тому S повинно адаптуватися і для цього використано EM-подібну (Expectation–Maximization) оцінку на зростаючому вікні останніх спроб. Що означає «зростаюче вікно»? Це алгоритм адаптації параметра S , який

складається з двох етапів, що виконуються для кожного кроку k на основі всієї доступної історії спроб від $t=0$ до k .

Перший етап – це оцінка ймовірності наявності знань. Для кожної попередньої спроби t розраховується «вага знання» w_t , яка відображає ймовірність того, що користувач дійсно знав слово в момент часу t , враховуючи прогноз моделі (p_ans_t) та поточні параметри вгадування (g_t) і помилки (s_t). Формула розрахунку має вигляд:

$$w_t = clip_{[0,1]} \left(\frac{p_ans_t - g_t}{1 - s_t - g_t} \right), \quad (2.7)$$

де p_ans_t – прогнозована моделлю ймовірність правильної відповіді;

g_t — ймовірність вгадування для типу вправи на кроці t ;

$clip_{[0,1]}$ — функція обмеження значення в межах від 0 до 1.

Другий етап – це оновлення значення Slip. Нове значення S розраховується як відношення суми ваг знань у тих випадках, коли була зроблена помилка, до загальної суми ваг знань за весь період спостереження:

$$s_t = \frac{\sum_{t=0}^k w_t \cdot 1[y_t = 0]}{\sum_{t=0}^k w_t}, \quad (2.8)$$

де y_t – фактична відповідь користувача (1 – правильно, 0 – неправильно);

$1[y_t = 0]$ – індикаторна функція, що дорівнює 1, якщо відповідь неправильна, і 0, якщо правильна.

Логіка цієї формули полягає в наступному. Якщо користувач дав правильну відповідь ($y_t = 1$), то доданок у чисельнику дорівнює нулю. Це збільшує лише знаменник, що «розмиває» частку помилок і тягне значення S вниз, оскільки наявність знань підтверджується відсутністю помилки. Якщо користувач дав неправильну відповідь ($y_t = 0$), то у чисельник додається значення w_t . Це означає, що система враховує, наскільки високою була

ймовірність того, що користувач знав слово, але все ж помилився. Такий підхід дозволяє розділити помилки, зроблені через незнання матеріалу, та помилки, що є результатом випадковості (Slip), забезпечуючи коректну калібрування ймовірностей у моделі ВІНЛ.

2.2.5. Ймовірність правильної відповіді

На кожному кроці n модель розраховує прогнозовану ймовірність правильної відповіді $P(C_n)$ на основі поточного рівня знання слова \tilde{K}_{n-1} , який вже враховує попередню історію навчання та часові фактори.

Ймовірність правильної відповіді визначається як сума ймовірностей двох несумісних подій: застосування реальних знань (без помилки «slip») або випадкового вгадування (за відсутності знань). Математично це виражається формулою:

$$P(C_n) = \tilde{K}_{n-1} \cdot (1 - S) + (1 - \tilde{K}_{n-1}) \cdot G, \quad (2.9)$$

де \tilde{K}_{n-1} – поточна ймовірність того, що користувач знає слово;

S (Slip) — ймовірність випадкової помилки попри наявність знань;

G (Guess) — ймовірність вгадування правильної відповіді за відсутності знань.

Така модель дозволяє врахувати стохастичну природу відповідей користувача, відокремлюючи впевнені знання від випадкових успіхів.

2.2.6. Оновлення ймовірності знання за спостереженням

Після отримання фактичної відповіді користувача (параметр *Correctness* $\in \{0, 1\}$) відбувається перерахунок ймовірності знання слова. Процес оновлення складається з двох послідовних етапів: розрахунку апостеріорної ймовірності та врахування переходу (навчання).

Перший етап – це розрахунок апостеріорної ймовірності. Застосовується правило Баєса для уточнення оцінки знань на основі отриманого результату (правильно/неправильно). Якщо відповідь правильна (*Correctness* = 1) , формула має вигляд:

$$K^{post} = \frac{\tilde{K}_{n-1}(1 - S)}{P(C_n)}, \quad (2.10)$$

Якщо відповідь неправильна (*Correctness* = 0), формула має вигляд:

$$K^{post} = \frac{\tilde{K}_{n-1} \cdot S}{1 - P(C_n)}, \quad (2.11)$$

де $P(C_n)$ – це прогнозована ймовірність правильної відповіді, розрахована у пункті 2.2.5.

В другому етапі йде врахування навчання (*Transition*). На етапі враховується ймовірність того, що процес навчання відбувся саме під час поточної спроби. Остаточне значення ймовірності знання K_n для наступного кроку визначається як сума апостеріорної ймовірності та приросту за рахунок переходу знань зі стану «не вивчено» у стан «вивчено»:

$$K_n = K^{post} + (1 - K^{post}) \cdot T, \quad (2.12)$$

де T (*Transition*) – ймовірність засвоєння матеріалу в результаті однієї

взаємодії. Розраховане значення K_n стає вхідним параметром для наступної ітерації алгоритму.

2.3 Модель IRT(3PL) для оцінювання здатності користувача та складності завдань

Для адаптації навчального процесу та точнішого прогнозування успішності виконання вправ використовується трипараметрична логістична модель теорії реакції на пункт (Item Response Theory — IRT 3PL). Основною перевагою використання IRT є можливість вибору оптимального рівня складності вправ для кожного користувача, що забезпечує адаптивність навчання.

2.3.1. Параметри моделі

Математичний апарат моделі базується на трьох ключових параметрах:

- θ_i (User Ability) – рівень знань або здібностей i -го користувача. Це динамічний параметр, що оновлюється після кожної взаємодії;
- b_j (Task Difficulty) – складність j -го завдання. Чим вище значення цього параметра, тим менша ймовірність правильної відповіді за незмінного рівня здібностей;
- G (Guessing) — ймовірність вгадування правильної відповіді (псевдо-шанс), яка є нижньою асимптотою логістичної кривої.

Додатково використовується параметр a_j (Discrimination), що характеризує диференціюючу здатність завдання. У рамках реалізованої моделі цей параметр зазвичай фіксується в діапазоні $[0.5, 2.0]$, із середнім

значенням $\approx 1,25$.

2.3.2. Розрахунок ймовірності відповіді

Ймовірність того, що користувач правильно відповість на завдання j ($Y_{ij} = 1$), розраховується за формулою трипараметричної логістичної моделі:

$$P_{ans}(Y_{ij} = 1 | \theta_i, a_j, b_j, G) = G + (1 - G) \cdot \sigma(a_j(\theta_i - b_j)), \quad (2.13)$$

де $\sigma(x)$ – сигмоїдна функція, що визначається як:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (2.14)$$

Складність завдання b є інтегральним показником, що формується на основі складності самого слова (d_{word}) та складності типу вправи ($d_{exercise}$). Розрахунок виконується шляхом усереднення нормалізованих значень:

$$b = (d_{word} + \frac{d_{exercise}}{\max(d_{exercise})})/2, \quad (2.14)$$

Значення d_{word} та $d_{exercise}$ нормалізуються до діапазону $[0, 1]$.

2.3.3. Алгоритм оновлення здібностей користувача

Оцінка рівня знань користувача θ відбувається ітеративно в режимі реального часу. Початкове значення встановлюється як $\theta_0 = 0$. Після кожної

спроби t параметр оновлюється на основі різниці між фактичною відповіддю та прогнозом моделі, з урахуванням швидкості навчання η (learning rate, зазвичай $\eta = 0,1$).

Правило оновлення θ_t має вигляд:

$$\theta_t = \begin{cases} \theta_{t-1} + \eta(1-P_t) & \text{— якщо відповідь правильна} = 1 \\ \theta_{t-1} + \eta \cdot P_t & \text{— якщо відповідь неправильна} = 0 \end{cases}, \quad (2.15)$$

Такий підхід дозволяє системі динамічно підлаштовувати оцінку здібностей користувача: успішне виконання складних завдань значно підвищує θ , тоді як помилки на простих завданнях знижують її.

2.3.4. Калібрування ймовірності знання

Оскільки модель IRT прогнозує ймовірність правильної відповіді P_{ans} , яка включає можливість вгадування, для отримання «чистої» ймовірності знання P_{know} застосовується процедура калібрування. Це дозволяє привести результати IRT до єдиної шкали з іншими моделями (наприклад, ВКТ). Формула калібрування має вигляд:

$$P_{know} = \frac{P_{ans} - g}{1 - g}, \quad (2.16)$$

Отримане значення $P_{know}(t) \in [0,1]$ відображає ступінь засвоєння матеріалу без урахування випадкового вгадування.

2.4 Прихована марківська модель (НММ) для переходів між станами знання слова

Прихована марківська модель (Hidden Markov Model – НММ) використовується для моделювання процесу навчання як послідовності переходів між прихованими станами знання, які неможливо спостерігати безпосередньо. Спостережуваними даними виступають лише відповіді користувача (правильні чи неправильні), час реакції та складність вправ.

Модель дозволяє оцінити ймовірність того, що користувач знаходиться в певному стані знання в конкретний момент часу, враховуючи історію відповідей та фактор забування.

2.4.1. Визначення простору станів

У рамках розробленої системи простір станів моделі складається з п'яти дискретних рівнів володіння словом (S_k), кожному з яких відповідає певна ймовірність правильної відповіді $\gamma_k = P(\text{Correct}=1 | S_k)$:

- S_1 (не вивчено) – ймовірність правильної відповіді $\gamma_1 \approx 0,10$;
- S_2 (малознайома) – ймовірність $\gamma_2 \approx 0,30$;
- S_3 (частково вивчено) – ймовірність $\gamma_3 \approx 0,60$;
- S_4 (майже вивчено) – ймовірність $\gamma_4 \approx 0,80$;
- S_5 (повністю вивчено) – ймовірність $\gamma_5 \approx 0,95$.

Початковий розподіл ймовірностей π_0 задається апріорно, припускаючи, що нове слово найчастіше є невідомим користувачеві (наприклад, $\pi_0 = [0.6, 0.2, 0.1, 0.05, 0.05]$).

2.4.2. Матриця переходів (Transition)

Зміна стану знання між спробами $n-1$ та n описується матрицею переходів T , елементи якої залежать від результату поточної відповіді, складності завдання та часу, що минув з попередньої спроби Δt_n . Ймовірності переходів розраховуються динамічно.

Розраховується ймовірність покращення знань (p_{up}) (перехід на рівень вище) залежить від швидкості навчання η та складності d :

$$p_{up} = \eta \cdot (1-d) \cdot e^{-\lambda \Delta t_n}, \quad (2.17)$$

де η – коефіцієнт навчання (зазвичай 0,1), λ – швидкість забування.

Розраховується ймовірність забування (p_{down}) (перехід на рівень нижче) зростає зі збільшенням часу інтервалу:

$$p_{down} = \mu \cdot d \cdot (1 - e^{-\lambda \Delta t_n}), \quad (2.18)$$

де μ – коефіцієнт забування (наприклад, 0,12).

Ймовірність залишення в поточному стані (p_{stay}):

$$p_{stay} = 1 - p_{up} - p_{down}, \quad (2.19)$$

Формування матриці T розмірністю 5×5 відбувається за двома сценаріями. Перший сценарій – це коли відповідь правильна ($c_n=1$). У цьому випадку модель передбачає можливість покращення знань. Матриця набуває вигляду верхньої дводіагональної, де на головній діагоналі розташована ймовірність збереження стану (p_{stay}), а над нею — ймовірність переходу на рівень вгору (p_{up}). Для найвищого стану S_5 перехід вище неможливий, тому

стан зберігається з ймовірністю 1.

$$T_{correct} = \begin{bmatrix} p_{stay} & p_{up} & 0 & 0 & 0 \\ 0 & p_{stay} & p_{up} & 0 & 0 \\ 0 & 0 & p_{stay} & p_{up} & 0 \\ 0 & 0 & 0 & p_{stay} & p_{up} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Другий сценарій – це коли відповідь неправильна ($c_n=0$). У випадку помилки модель враховує ймовірність деградації знань. Матриця набуває вигляду нижньої дводіагональної. Елементи під головною діагоналлю відображають ймовірність переходу на рівень вниз (p_{down}). Для найнижчого стану S_1 подальше зниження неможливе.

$$T_{incorrect} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ p_{down} & p_{stay} & 0 & 0 & 0 \\ 0 & p_{down} & p_{stay} & 0 & 0 \\ 0 & 0 & p_{down} & p_{stay} & 0 \\ 0 & 0 & 0 & p_{down} & p_{stay} \end{bmatrix}.$$

Такий підхід забезпечує адаптивність моделі: успішні дії користувача поступово переміщують розподіл ймовірностей до станів високого рівня знання, тоді як помилки та час повертають його до початкових станів.

2.4.3. Оновлення емісії (Emission Update)

Кроком, що пов'язує приховані стани зі спостережуваними відповідями, є оновлення емісії. Цей етап коригує ймовірність перебування в кожному стані i на основі фактичної правильності відповіді c_n та ймовірності правильної відповіді для цього стану γ_i (у джерелах також позначається як q_i).

Нехай $\widetilde{\pi}_n(i)$ — апіорна ймовірність знаходження в стані i після кроку

переходу (Transition), але до врахування поточної відповіді. Тоді апостеріорна ймовірність $\pi_n(i)$ розраховується наступним чином, якщо відповідь правильна ($c_n = 1$):

$$\pi_n(i) = \frac{\widetilde{\pi}_n(i) \cdot \gamma_i}{\sum_{k=1}^5 \widetilde{\pi}_n(k) \cdot \gamma_k}, \quad (2.20)$$

Якщо відповідь неправильна ($c_n = 0$):

$$\pi_n(i) = \frac{\widetilde{\pi}_n(i) \cdot (1 - \gamma_i)}{\sum_{k=1}^5 \widetilde{\pi}_n(k) \cdot (1 - \gamma_k)}, \quad (2.21)$$

Цей процес є реалізацією кроку forward в алгоритмі НММ: спочатку застосовується матриця переходів, потім розраховується емісія, і на завершення виконується нормалізація розподілу.

2.4.4. Оцінка інтегрального рівня знань

Після врахування відповіді користувача (крок Emission) та виконання нормалізації, модель отримує фінальний вектор апостеріорних ймовірностей $P(S_n = k)$. Кожен елемент цього вектору відображає ймовірність того, що користувач знаходиться у стані знання k саме зараз, враховуючи всю попередню історію взаємодій.

Для отримання єдиної метрики, придатної для використання в алгоритмі рекомендацій та візуалізації прогресу, обчислюється математичне сподівання рівня знань ($P_{knowledge}$). Ця величина є зваженою сумою рівнів володіння словом за їх апостеріорними ймовірностями:

$$P_{knowledge} = \sum_{k=1}^5 y_k \cdot P(S_n = k), \quad (2.22)$$

де $P(S_n = k)$ — апостеріорна ймовірність знаходження користувача у стані k після спостереження відповіді (результат роботи НММ на поточному кроці);

y_k — фіксоване числове значення («вага») рівня знань, що відповідає стану S_k (відповідно до налаштувань моделі: $y = [0,1; 0,3; 0,6; 0,8; 0,95]$).

Отримана величина $P_{knowledge} \in [0, 1]$ інтегрується у загальну модель VINL як компонент оцінки НММ.

2.5 Нейромережна модель LSTM для прогнозування результату наступної відповіді

Для моделювання складних часових залежностей у процесі навчання та прогнозування ймовірності правильної відповіді використано рекурентну нейронну мережу (RNN) архітектури Long Short-Term Memory (LSTM). На відміну від марківських моделей (HMM), LSTM здатна зберігати інформацію про довгострокові патерни поведінки користувача, враховуючи не лише останню взаємодію, а й усю послідовність спроб у межах контекстного вікна.

2.5.1. Векторне представлення контексту навчання

На вхід нейронної мережі подається послідовність з останніх 15 взаємодій користувача ($T=15$). Кожен крок у цій послідовності представлений розширеним вектором ознак x_t розмірністю $D = 53$ (залежно від кількості підтримуваних мов та типів вправ), що формується шляхом нормалізації та

кодування сирих даних. Простір ознак включає чотири групи параметрів.

Історичні показники успішності (Performance Features):

- бінарний маркер правильності відповіді (Correctness $\in \{0, 1\}$);
- нормалізована кількість спроб (Attempts), оброблена через $\log(1+x)$;
- поточні довжини серій успіхів (Success Streak) та помилок (Error Streak), нормалізовані відносно максимальної довжини послідовності;
- відносна позиція спроби в історії (Try Index Fraction).

Часові характеристики (Temporal Features):

- час реакції (*Response Time*): різниця між початком та завершенням вправи, логарифмована та z-нормалізована;
- інтервал часу від попередньої спроби (*Time From Previous Attempt*);
- час, що минув з моменту першого знайомства зі словом (*Since First Attempt*).

Контентні характеристики (Content Features):

- складність слова (нормалізований рівень CEFR);
- складність вправи (*Exercise Difficulty*);
- довжина слова (*Word Length*), нормалізована через z-score;
- наявність мультимедійних підказок: бінарні ознаки наявності зображення (*Has Image*) та аудіо (*Has Sound*);
- позиція слова у вправі (*Position in Exercise*).

Категоріальні ознаки (Categorical Embeddings):

- тип вправи: кодується розширеним вектором, що включає бінарні прапори механік (наприклад, `is_multiple_choice`, `is_typing`, `has_translation`, `target_is_word` тощо), що дозволяє моделі розрізняти когнітивне навантаження різних типів завдань;
- мовні параметри: мова слова (Word Language) та мова перекладу (Translate Language) представлені у форматі One-Hot Encoding, що збільшує розмірність вхідного вектору залежно від кількості підтримуваних локалізацій.

2.5.2. Навчання та критерії відбору

Навчання моделі проводиться методом стохастичного градієнтного спуску з використанням оптимізатора Adam (Learning Rate = 10^{-3}) та функції втрат Binary Cross-Entropy.

Для боротьби з дисбалансом класів (коли кількість правильних відповідей значно перевищує кількість помилок) застосовано метод Oversampling для меншості класу у тренувальній вибірці.

Відбір найкращої моделі здійснюється за спеціально розробленим евристичним алгоритмом, який пріоритезує повноту (Recall) розпізнавання помилок. Критерії відбору епохи:

- recall для класу 1 (правильно) $\geq 0,65$;
- recall для класу 0 (помилка) $\geq 0,55$.

Серед епох, що задовольняють цим умовам, обирається та, яка максимізує кількість правильно передбачених помилок, що є критичним для запобігання передчасному оголошенню слова "вивченим".

2.5.3. Інтеграція та оцінка знань

Вихід моделі P_{LSTM} відображає ймовірність правильної відповіді в конкретному контексті. Для отримання уніфікованої ймовірності знання слова, що використовується в системі рекомендацій, застосовується формула калібрування з урахуванням ймовірності вгадування (G) та випадкової помилки (S=0,1):

$$P_{know} = clamp_{[0,1]} \left(\frac{P_{lstm} - G}{1 - G - S} \right), \quad (2.23)$$

Параметр G розраховується динамічно як середнє значення ймовірності вгадування для всіх типів вправ в історії слова. У випадках, коли знаменник близький до нуля (вироджений випадок $S+G \approx 1$), або при недостатній історії, використовується евристична корекція на основі кількості успішних спроб підряд.

3. РОЗРОБКА КОМБІНОВАНОГО МЕТОДУ НАДАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ ГІБРИДНОЇ МОДЕЛІ VINL

У роботі запропоновано комбінований метод формування рекомендацій для ефективного вивчення лексики іноземної мови. Основна ідея полягає у збиранні даних про виконання вправ користувачами та побудові моделі, яка на основі цієї історії відповідей рекомендує найдоцільніші наступні вправи з урахуванням рівня володіння словами. Такий підхід спирається на сучасні дослідження в галузі knowledge tracing – відстеження стану знань студента за його попередніми відповідями.

Запропонований метод рекомендацій складається з чотирьох основних етапів.

На першому етапі, збирається історія взаємодії користувача з системою: правильність відповідей, час реакції, типи вправ, складність слів тощо.

На другому – на основі цієї історії комбінована модель VINL оцінює ймовірність знання кожного слова P_{know} .

На третьому – кожне слово відноситься до певного рівня складності (від дуже легкого до дуже складного) залежно від отриманої ймовірності.

На останньому етапі обирається тип вправи, який найкраще відповідає визначеному рівню: наприклад, тест з вибором відповіді – для середнього рівня, вправа на відтворення слова з пам'яті – для високого рівня знань.

Як показано вище, результати досліджень у галузі knowledge tracing доводять, що для точного відстеження стану знань учнів можуть бути використані як класичні статистичні моделі, так і нейромережеві підходи.

Для розробки комбінованого методу рекомендацій у роботі використовується синтезована гібридна модель VINL, яка поєднує в собі переваги ВКТ, IRT, НММ і LSTM. Кожна з моделей на основі історії відповідей користувача формує власну оцінку ймовірності знання слова $P_{ВКТ}, P_{IRT}, P_{НММ}, P_{LSTM}$. Далі ці значення агрегуються у єдину інтегральну

оцінку P_{VINL} за допомогою вагових коефіцієнтів, які адаптивно змінюються з часом. Для оцінювання якості використовуються ймовірнісні метрики (log-loss, Brier score, Expected Calibration Error), а ваги моделей коригуються так, щоб більше «довіряти» тим із них, що показують кращі результати на поточному інтервалі часу.

Такий підхід дозволяє врахувати різні аспекти процесу навчання (стан навички, співвідношення «здатність-складність», приховані стани, довгі часові залежності) в єдиній інтегрованій моделі.

Розглянемо гібридну модель VINL більш детально.

3.1 Гібридна модель VINL

Для забезпечення високої точності та надійності оцінювання знань у системі реалізовано гібридну метамодель VINL (BKT-IRT-HMM-LSTM). Цей підхід дозволяє інтегрувати сильні сторони різних математичних апаратів: ймовірнісну природу BKT і HMM, адаптивність IRT до складності завдань та здатність LSTM виявляти нелінійні часові залежності.

3.1.1. Формування інтегральної оцінки

Інтегральна оцінка знання слова (P_{VINL}) формується як зважена лінійна комбінація прогнозів усіх чотирьох моделей. Такий ансамблевий метод дозволяє нівелювати індивідуальні помилки окремих алгоритмів та отримати більш стабільний показник володіння матеріалом. Математично розрахунок фінальної ймовірності описується формулою:

$$P_{BIHL} = \sum_{m \in M} W_m \cdot P_{know}^{(m)}, \quad (3.1)$$

Де $M = \{\text{BKT}, \text{IRT}, \text{HMM}, \text{LSTM}\}$ – множина використовуваних моделей;

$P_{know}^{(m)}$ – ймовірність знання слова, отримана від моделі m (попередньо приведена до єдиної шкали через врахування параметрів вгадування та помилки);

W_m – ваговий коефіцієнт довіри до моделі m , що нормується умовою $\sum W_m = 1$.

Логічну структуру роботи моделі BIHL, яка демонструє процес агрегації результатів від окремих компонентів (BKT, IRT, HMM, LSTM) з урахуванням їхніх вагових коефіцієнтів, наведено на рис. 2.1.

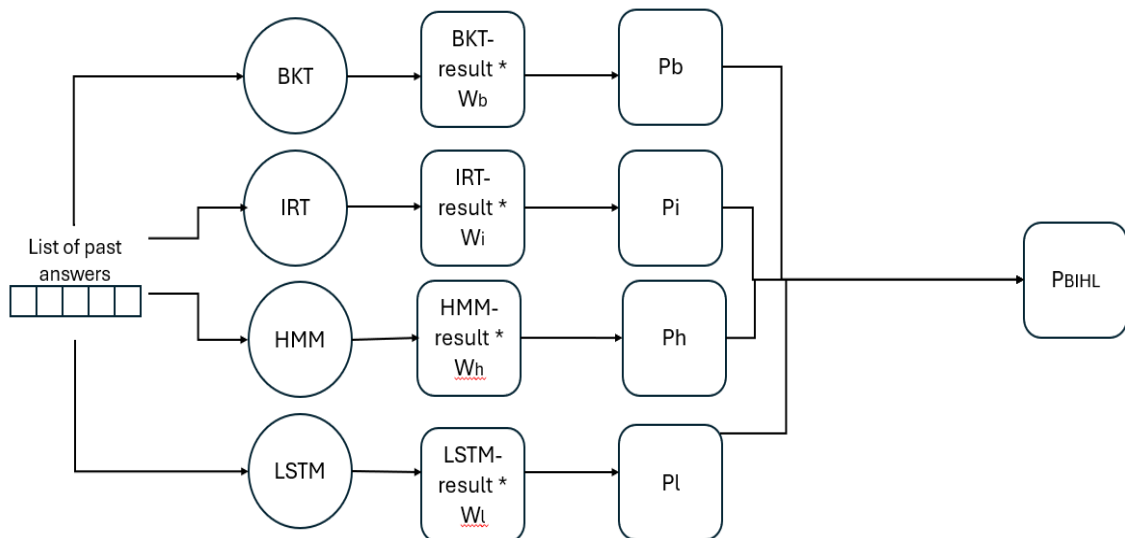


Рисунок 3.1 – Схема логіки роботи моделі BIHL

3.1.2. Адаптивний розрахунок ваг

Вагові коефіцієнти W_m не є статичними параметрами; система

автоматично адаптує їх для кожного користувача на основі ретроспективного аналізу точності прогнозів. Це дозволяє динамічно надавати пріоритет тій моделі, яка найкраще описує поведінку конкретного студента в поточний період часу.

Розрахунок ваг виконується на ковзному вікні історії спроб. Вага конкретної моделі визначається як частка її точності (Acc_m) у загальній сумі точностей ансамблю:

$$W_m = \frac{Acc_m}{\sum_{k \in M} Acc_k}, \quad (3.2)$$

де W_m – ваговий коефіцієнт m -ї моделі в ансамблі;

Acc_m – значення точності прогнозування m -ї моделі, обчислене на ковзному вікні історії спроб користувача;

M – множина всіх моделей, що входять до ансамблю;

k – індекс моделі з множин M .

Для оцінки точності Acc застосовуються метрики, що враховують відхилення прогнозованої ймовірності від фактичного результату відповіді (наприклад, на основі Brier Score або усередненої бінарної точності). Такий механізм робить систему стійкою до змін у патернах навчання: якщо одна з моделей починає помилятися (наприклад, ВКТ при довгих перервах), її вплив на кінцевий результат автоматично зменшується.

3.2 Формалізація задачі вибору оптимального типу вправи

Кінцевим етапом роботи алгоритму є трансформація кількісної оцінки знань (P_{BIHL}) у конкретну педагогічну дію — рекомендацію типу вправи. Ця задача формалізується як відображення множини ймовірностей на

впорядковану множину типів вправ з урахуванням часових обмежень та евристик забування.

3.2.1. Функція відображення ймовірності у складність вправи

Нехай $E = \{e_1, e_2, \dots, e_n\}$ — множина доступних типів вправ. Кожному типу вправи e_i присвоєно рівень складності $L(e_i) \in \{L_{min}, \dots, L_{max}\}$, де рівні варіюються від «Дуже легкого» (Very Easy) до «Дуже складного» (Very Hard).

Задача вибору базового типу вправи e_{base} для слова з оціненою ймовірністю знання $p = P_{ВІНЛ}$ реалізується через систему порогових правил. Ці правила, визначені емпіричним шляхом на основі аналізу кривих навчання, наведено в табл. 3.1.

Таблиця 3.1 — Відображення ймовірності знання на тип рекомендованої вправи

Інтервал ймовірності (P_{know})	Рекомендований тип вправи (e_{base})	Рівень складності
$p < 0,15$	Flashcards / Word by Translates	Very Easy (L1)
$0,15 \leq p < 0,3$	Word by Originals	Easy (L2)
$0,3 < p < 0,4$	Match Words	Easy (L2)
$0,4 < p < 0,5$	Word by Descriptions	Medium (L3)
$0,5 < p < 0,6$	Description by Words	Medium (L3)
$0,6 < p < 0,67$	Letters Puzzle (Translation)	Hard (L4)
$0,67 < p < 0,75$	Letters Puzzle (Description)	Hard (L4)
$0,75 < p < 0,85$	Typing (Translation)	Very Hard (L5)
$p \leq 0,85$	Typing (Description)	Very Hard (L5)

Такий розподіл забезпечує плавний перехід між механіками взаємодії, дозволяючи користувачеві закріплювати матеріал на різних рівнях абстракції.

3.2.2. Корекція складності з урахуванням забування

Окрім поточної оцінки $P_{\text{ВІНЛ}}$, система враховує фактор давності останнього повторення. Якщо слово не повторювалося протягом тривалого часу ($T_{\text{gap}} > T_{\text{threshold}}$), висока ймовірність знання може бути оманливою через ефект "кривої забування", який ще не встиг повною мірою відобразитися у моделі.

Алгоритм корекції визначається наступним чином:

1. розраховується середній час останнього повторення слів у поточній сесії t_{last} ;
2. визначається інтервал давності $\Delta t = t_{\text{now}} - t_{\text{last}}$;
3. якщо Δt перевищує поріг "напівзабування" (Half Forgotten, зазвичай 7 діб), рівень складності примусово знижується:

$$L_{\text{final}} = \max(L(\Phi(p)) - 1, L_{\text{min}}), \quad (3.3)$$

Це правило гарантує, що після тривалої перерви користувач не отримає найскладніші вправи (наприклад, введення тексту), навіть якщо його попередній рівень знань був високим.

3.2.3. Формування фінального набору вправ

Для забезпечення різноманітності та уникнення монотонності навчальної сесії, система не обмежується одним типом вправ. Алгоритм формує набір рекомендацій R_{set} , який включає:

- вправи рівня, визначеного на основі $P_{\text{ВІНЛ}}$ (після корекції на

забування);

– вправи наступного рівня складності ($L_{\text{final}} + 1$), щоб створити зону найближчого розвитку.

Якщо визначена складність є максимальною (Very Hard), до набору включаються лише вправи цього рівня. У випадку, коли масив рекомендованих складностей порожній, система за замовчуванням повертає базові вправи типу Word by Translates та Word by Originals.

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЕРСОНАЛІЗОВАНОЇ РЕКОМЕНДАЦІЇ ВПРАВ

4.1 Загальна архітектура системи вивчення слів та взаємодія з багатоплатформеними додатками

Програмний комплекс реалізовано на основі розподіленої клієнт-серверної архітектури з використанням мікросервісного підходу. Вибір такої архітектурної моделі обумовлений необхідністю забезпечення гнучкості масштабування, незалежного розгортання окремих функціональних модулів та підтримки гетерогенних клієнтських платформ.

Система декомпонована на три логічні рівні: рівень представлення (Client Layer), рівень бізнес-логіки та рівень даних (Data Layer). Загальна схема взаємодії компонентів наведена на рис 4.1.

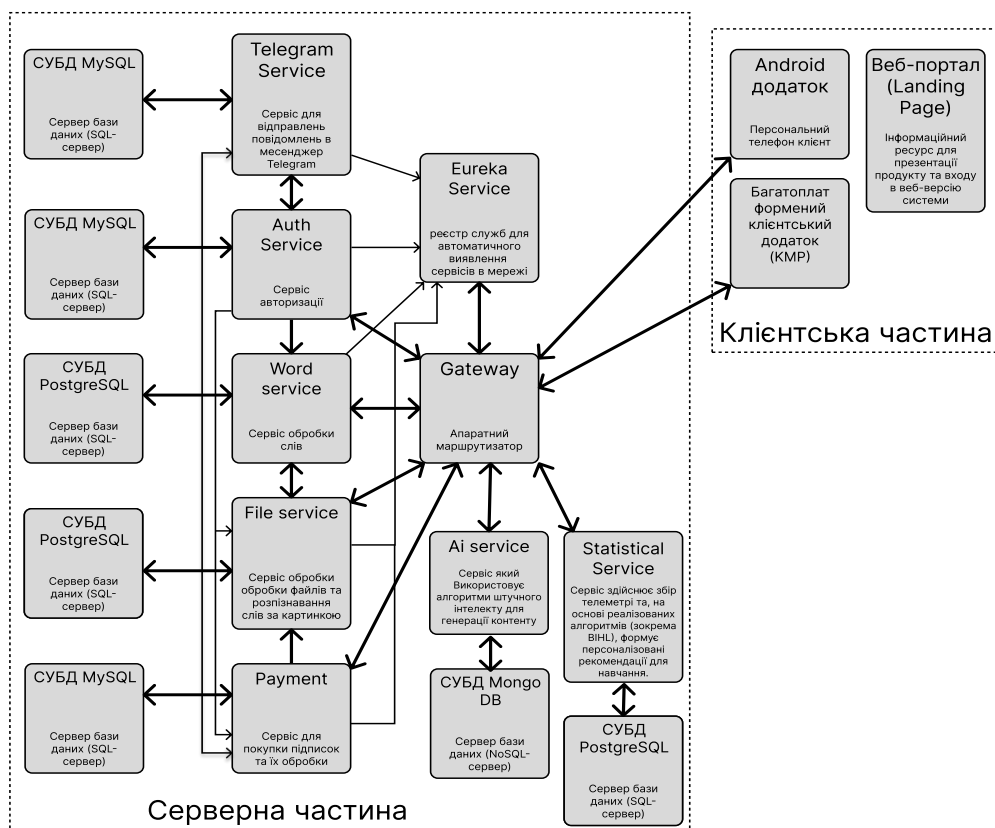


Рисунок 4.1 – Загальна схема взаємодії компонентів

4.1.1 Клієнтський рівень

Клієнтська частина системи побудована за принципом «тонкого клієнта», що передбачає делегування основних обчислювальних операцій та зберігання даних серверній частині. Для функціонування клієнтського програмного забезпечення необхідне стійке з'єднання з мережею Інтернет.

Екосистема клієнтських додатків включає три основні компоненти:

- багатоплатформений клієнтський додаток (КМР): Розроблений з використанням технології Kotlin Multiplatform. Такий підхід дозволив уніфікувати бізнес-логіку на стороні клієнта та забезпечити розгортання програмного продукту на платформах Web, iOS, Desktop, а також у вигляді полегшеної версії для Android (Android Lite);
- нативний Android SDK: Окремий програмний модуль, реалізований мовою Kotlin. Він забезпечує низькорівневу інтеграцію з операційною системою Android для реалізації специфічних функцій, недоступних у кросплатформенному рішенні;
- веб-портал (Landing Page): Інформаційний ресурс для презентації продукту та входу в веб-версію системи, реалізований з використанням бібліотеки React та мови TypeScript.

4.1.2 Серверний рівень

Серверна частина реалізована на базі мікросервісної архітектури. Основним технологічним стеком є мови програмування Java та Kotlin з використанням фреймворку Spring Boot, а також TypeScript (середовище Node.js) для окремих сервісів. Взаємодія між клієнтами та сервером здійснюється через RESTful API у форматі JSON. Для забезпечення безпеки та

розмежування прав доступу застосовується механізм токенів JWT (JSON Web Tokens). Інфраструктура серверної частини складається з компонентів, як інфраструктурні сервіси:

- Eureka Server (Java, Spring Cloud): Виконує функцію Service Discovery, забезпечуючи реєстрацію та виявлення активних екземплярів мікросервісів у мережі;
- API Gateway (Java, Spring Cloud): Виступає єдиною точкою входу для всіх зовнішніх запитів. Забезпечує маршрутизацію запитів до відповідних сервісів, балансування навантаження та первинну валідацію авторизаційних даних.

Наступний компонент – це сервіси бізнес-логіки та підтримки користувачів, які складаються з:

- Auth Service (Kotlin, Spring Boot): Відповідає за процеси реєстрації, автентифікації користувачів та генерацію сесійних токенів;
- Subscribe Service (TypeScript, Express): Забезпечує керування підписками користувачів та контроль доступу до платного функціоналу;
- Telegram Service (Java, Spring Boot): Реалізує інтеграцію з платформою Telegram, забезпечуючи взаємодію користувачів із системою через чат-бот інтерфейс;
- Word Service (Java, Spring Boot): Є основним каталогом навчального контенту. Сервіс забезпечує виконання CRUD-операцій над сутностями слів, перекладів та категорій.

Останній компонент – це сервіси інтелектуальної обробки даних, які складаються з:

- Statistical Service (Java, Spring Boot): Ключовий компонент адаптивної системи. Сервіс здійснює збір телеметрії (історії виконання вправ, часу реакції, помилок) та, на основі реалізованих алгоритмів (зокрема BHL), формує персоналізовані рекомендації для навчання;
- AI Service (Java, Spring Boot): Використовує алгоритми штучного інтелекту для генерації контенту, зокрема синтезу мовлення (Text-to-Speech)

та автоматичного створення описів слів;

- File Service (Java, Spring Boot): Відповідає за зберігання та обробку медіа-файлів, а також виконує автоматичний аналіз контенту для фільтрації заборонених матеріалів.

4.1.3 Рівень даних

Зберігання даних організовано за принципом поліглотного зберігання (Polyglot Persistence), що передбачає вибір типу системи керування базами даних (СКБД) відповідно до структури та вимог до даних:

- реляційні бази даних (MySQL): Використовуються сервісами Auth, Telegram та Subscribe для зберігання даних, що вимагають суворої структури та транзакційної цілісності (облікові записи, фінансові операції);

- об'єктно-реляційні бази даних (PostgreSQL): Застосовуються сервісами Word, File та AI для зберігання складних ;структурованих даних навчального контенту та метаданих файлів;

- документо-орієнтовані бази даних (MongoDB): Використовується сервісом Statistical. Гнучка схема даних MongoDB дозволяє ефективно записувати та агрегувати великі масиви різномірних логів історії навчання (Exercise History), що є критичним для коректної роботи алгоритмів Knowledge Tracing.

4.2 Проектування структури набору даних та схеми збору логів користувача

Для забезпечення роботи адаптивних алгоритмів (BKT, IRT, LSTM)

критично важливим є збір детальної телеметрії взаємодії користувача з системою. Оскільки дані про виконання вправ мають ієрархічну структуру та можуть швидко накопичуватися, для їх зберігання обрано документо-орієнтовану СКБД MongoDB. Це дозволяє зберігати лог виконання вправи як єдиний атомарний документ, що містить метадані сесії та список взаємодій з конкретними словами.

4.2.1 Концептуальна модель даних

Схема даних спроектована таким чином, щоб розділити подієві дані (факти виконання вправ) та станові дані (глобальний прогрес вивчення слова).

Основним елементом логування є сутність транзакції, яка фіксує факт проходження користувачем конкретної вправи (сесії). Структура документу в базі даних описується класом `ExerciseTransactionEntity` і включає наступні агреговані показники:

- ідентифікатори – унікальний ID транзакції, ID користувача (`userId`), ID типу вправи (`exerciseId`);
- часові мітки сесії – час початку (`startExerciseTime`) та завершення (`endExerciseTime`) виконання всієї вправи;
- метрики повноти – загальна кількість слів у вправі (`exerciseWordsCount`), кількість пройдених слів (`exerciseCompletedWordsCount`) та прапор завершення (`exerciseCompleted`);
- деталізація – список вкладених об'єктів `words`, що містять статистику по кожному окремому слову в рамках цієї транзакції.

Детальна інформація про взаємодію з кожним словом зберігається у вкладеному об'єкті `StatisticalWordEntity`. Цей рівень деталізації необхідний для навчання моделей, оскільки містить контекстні ознаки:

- контекст слова – `wordId`, текст слова та перекладу, мовні

налаштування (wordLanguage, translateLanguage), рівень складності CEFR (cefrLevel);

- контекст контенту – наявність додаткових підказок під час спроби: опис (hasDescription), аудіо (hasSound), зображення (hasImage);

- результативність: isCorrect (Boolean) – бінарний показник правильності відповіді та attempts (Integer) – кількість спроб, витрачених на слово в рамках поточної картки;

- часові метрики слова – точний час початку (executeStartTime) та завершення (executeEndTime) роботи над конкретним словом, що дозволяє вирахувати «час реакції» (Response Time);

- позиційні дані – порядковий номер слова (positionInExercise) у вправі (дозволяє врахувати ефект втоми або первинності).

4.2.2. Схема збору логів

Процес збору даних реалізовано асинхронно, щоб не впливати на швидкість роботи клієнтського додатку. Мобільний додаток (Android/KMP) накопичує результати виконання вправ у локальній базі даних. Після завершення вправи (або відновлення зв'язку) клієнт відправляє пакет даних (Batch) у форматі JSON на ендпоінт Statistical Service. Сервіс статистики десеріалізує отримані дані у об'єкти ExerciseTransactionEntity, збагачує їх серверним часом отримання та зберігає у MongoDB.

Така схема забезпечує повноту даних навіть при нестабільному інтернет-з'єднанні та надає достатню інформаційну базу для роботи ансамблю моделей VINL.

4.3 Попередня обробка і нормалізація даних

Ефективність роботи нейромережових моделей, зокрема LSTM, значною мірою залежить від якості та формату вхідних даних. Сирі дані, отримані з логів взаємодії користувача із системою, мають різний масштаб, одиниці виміру та розподіл, що може негативно вплинути на збіжність алгоритмів градієнтного спуску. Для вирішення цієї проблеми у розробленій системі реалізовано комплексний підхід до попередньої обробки та нормалізації даних.

4.3.1 Стратегія обробки категоріальних та бінарних ознак

Для приведення категоріальних даних до формату, придатного для обробки нейронною мережею, використано методи бінарного кодування та One-Hot Encoding:

- мовні параметри – цю атрибути мови слова (`wordLanguage`) та мови перекладу (`translateLanguage`) кодуються з використанням методу One-Hot Encoding, що дозволяє моделі розрізняти контекст вивчення без введення штучного порядку між мовами;
- типи вправ, де замість використання простого ідентифікатора типу вправи, застосовано розгорнутий вектор бінарних ознак, що характеризують механіку завдання. Це включає прапорці наявності перекладу (`has_translation`), опису (`has_description`), множинного вибору (`is_multiple_choice`), введення тексту (`is_typing`) тощо. Такий підхід дозволяє моделі враховувати когнітивне навантаження різних типів вправ;
- статичні бінарні ознаки – це показники наявності зображення (`hasImage`) та аудіосупроводу (`hasSound`) безпосередньо трансформуються у

числовий формат $\{0.0, 1.0\}$. Показник правильності відповіді (correctness) також представлено бінарним значенням.

4.3.2 Нормалізація числових атрибутів зі складним розподілом

Часові характеристики та кількісні показники часто мають розподіл з «важким хвостом» або значні викиди. Для їх нормалізації застосовано двоступеневу процедуру.

Перша процедура – це логарифмування. До часових метрик (час реакції, час з першої спроби, час з останньої спроби) застосовується функція Math.log1p (натуральний логарифм від $x+1$). Це дозволяє зменшити асиметрію розподілу та мінімізувати вплив викидів⁵⁵⁵⁵. Часові інтервали попередньо нормуються до доби (діленням на 86400000 мс).

Друга процедура – це Z-нормалізація. Після логарифмування виконується стандартизація даних шляхом віднімання середнього значення та діленням на стандартне відхилення, обчислених на всій вибірці. Це забезпечує центрування даних навколо нуля з одиничною дисперсією.

Окремий підхід застосовано для нормалізації кількості. Враховуючи, що деякі вправи передбачають лише одну спробу, для багатоспробних вправ обчислюється 95-й перцентиль розподілу спроб для визначення очікуваного максимуму. Нормалізоване значення розраховується як відношення логарифма поточної кількості спроб до логарифма очікуваного максимуму, обмежене діапазоном $[0, 1]$.

4.3.3 Формування ознак історії навчання

Для роботи рекурентної мережі LSTM критично важливим є врахування часової послідовності подій. Клас `IndexNormalizer` та логіка групування у `DataNormalizationService` забезпечують підготовку історичних даних:

- групування та сортування. Дані групуються за ідентифікатором користувача та слова, після чого сортуються за часом початку виконання (`executeStartTime`);
- відносна позиція. Розраховується показник `tryIndexFrac`, що відображає відносну позицію поточної спроби в історії вивчення слова;
- серії успіхів та помилок. Обчислюються показники `successStreak` та `errorStreak`, які нормалізуються діленням на максимальну довжину послідовності (`MAX_SEQUENCE_LENGTH`), що дозволяє моделі оцінювати поточну «форму» користувача;
- лагові ознаки. Для кожної спроби (починаючи з другої) розраховуються інтервали часу, що минули з моменту першого знайомства зі словом (`sinceFirstAttempt`) та з попередньої спроби (`timeFromPreviousAttempt`).

4.3.4 Конструювання фінального вектору ознак

Результатом роботи модуля нормалізації є перетворення об'єктів `ExerciseHistory` на об'єкти `NormalizedExerciseHistory` з подальшою генерацією числових послідовностей. Фінальний вектор ознак для кожного кроку LSTM включає: нормалізовані показники правильності, часові лаги, складність слова (`CEFR`, нормалізований до $[0, 1]$), час реакції, розширені ознаки типу вправи, складність вправи, позицію у вправі, наявність медіа, довжину слова, мовні вектори та статистику спроб і серій.

Така підготовка даних дозволяє нівелювати різницю у масштабах вхідних параметрів, зберегти інформативність категоріальних ознак та забезпечити стабільне навчання комбінованої моделі VINL.

4.4 Програмна реалізація моделей ВКТ, IRT, НММ, LSTM і VINL

Програмна реалізація математичних моделей оцінювання знань виконана у вигляді окремих сервісів, які імплементують єдиний інтерфейс KnowledgeEstimator. Такий підхід, заснований на патерні проектування «Стратегія», дозволяє уніфікувати процес отримання оцінки незалежно від внутрішньої логіки конкретного алгоритму. Кожен клас-естіматор приймає на вхід об'єкт налаштувань та історії, повертаючи розраховану ймовірність знання.

4.4.1 Реалізація ймовірнісних моделей (ВКТ та НММ)

Реалізація моделі Bayesian Knowledge Tracing (ВКТ) зосереджена у класі BktKnowledgeEstimator та враховує динаміку забування й адаптивні параметри вгадування і помилки. Програмний алгоритм ВКТ передбачає:

- ініціалізацію параметрів моделі з використанням констант ймовірності навчання TRANSITION на рівні 0.1 та початкової ймовірності помилки START_SLIPPING на рівні 0.1;
- врахування ефекту забування через метод getAccountingForForgetting, який коригує попередню ймовірність знання залежно від часу, що минув, використовуючи експоненційну функцію або рівень CEFR для першої спроби;

- динамічний розрахунок параметра ковзання (slip) за допомогою методу calculateSlipping, що реалізує EM-подібний алгоритм оцінки на основі «ваги знань» попередніх спроб для адаптації до індивідуальної поведінки користувача;

- обчислення апостеріорної ймовірності знання з урахуванням результату поточної відповіді, розрахованого slip та параметра guess, який визначається через утиліту GuessUtils.

Реалізація прихованої марковської моделі (HMM) у класі HmmKnowledgeEstimator базується на дискретному просторі з п'яти станів знання і включає:

- визначення рівнів знань для станів у діапазоні від 0.1 до 0.95 та початкового розподілу ймовірностей;

- динамічне формування матриць переходів за допомогою методів createUpwardTransitionMatrix та createDownwardTransitionMatrix, де ймовірність переходу залежить від складності вправи, коефіцієнтів навчання та забування;

- оновлення вектору ймовірностей перебування у кожному зі станів на основі правильності відповіді (Emission Update) з подальшою нормалізацією їх суми до одиниці;

- обчислення інтегральної оцінки як скалярного добутку фінального вектора ймовірностей станів на фіксовані рівні знань.

4.4.2 Реалізація психометричної моделі (IRT)

Модель Item Response Theory (IRT 3PL) імплементована у класі IrtKnowledgeEstimator і фокусується на оцінці латентної здібності користувача у безперервному просторі. Процес обчислення включає:

- параметризацію моделі з фіксованим значенням дискримінації

DISCRIMINATION = 3, коефіцієнтом швидкості навчання LEARNING_SPEED = 0.35 та початковою здібністю користувача 0.1;

- розрахунок ймовірності правильної відповіді за трипараметричною логістичною формулою, що враховує складність завдання, поточну здібність користувача та ймовірність вгадування;

- ітеративне коригування значення здібності користувача методом updateAbility після кожної спроби, що збільшує значення при успіху та зменшує при помилках залежно від прогнозованої ймовірності.

4.4.3 Реалізація нейромережевої моделі (LSTM)

Для роботи з глибинною нейронною мережею розроблено сервіс LstmKnowledgeModelServiceImpl, який використовує бібліотеку DeepLearning4j. Архітектура мережі та логіка оцінювання передбачають:

- конфігурацію вхідного шару для приймання послідовності нормалізованих векторів ознак взаємодії користувача;

- використання рекурентного шару LSTM зі 128 нейронами, функцією активації TANH та застосуванням Dropout (0.3) для запобігання перенавчанню;

- застосування шару GlobalPoolingLayer з типом пулінгу MAX для зменшення розмірності часових кроків та виділення найбільш значущих ознак;

- перетворення ознак через повнозв'язний шар (DenseLayer) на 16 нейронів з активацією RELU та вихідний шар з активацією SIGMOID для отримання ймовірності;

- калібрування результату у класі LstmKnowledgeEstimator, де прогнозована ймовірність правильної відповіді трансформується в ймовірність знання з урахуванням параметрів вгадування (G) та фіксованої помилки (S=0.15).

4.4.4 Реалізація гібридної моделі (VINL)

Інтеграція результатів усіх моделей відбувається у класі `BihlKnowledgeEstimator`, який виступає агрегатором прогнозів. Логіка роботи цього сервісу включає:

- отримання індивідуальних прогнозів ймовірності знання від реалізацій ВКТ, IRT, НММ та LSTM;
- визначення фіксованих вагових коефіцієнтів для кожної моделі, де найвищий пріоритет надано LSTM (0.5), середній – IRT та НММ (по 0.2), а найменший – ВКТ (0.1);
- обчислення підсумкової ймовірності знання як зваженої суми прогнозів усіх активних естиматорів, що забезпечує підвищену стійкість оцінки.

4.5 Алгоритм «Exercise recommendation pipeline» для формування набору вправ сесії

Розроблений алгоритм рекомендацій (Exercise recommendation pipeline) забезпечує автоматизовану трансформацію ймовірнісних оцінок знань, отриманих від VINL-моделі, у конкретний набір навчальних завдань. Програмна логіка алгоритму інкапсульована у класі `RecommendationServiceImpl`, який реалізує інтерфейс `RecommendationService` та виконує послідовну обробку вхідного запиту на формування сесії. Процес формування рекомендацій складається з п'яти основних етапів, що включають:

- агрегацію історичних даних та обчислення оцінок, під час чого за ідентифікатором користувача та списком слів з бази даних витягується повна

історія взаємодій, а для кожного слова виконується розрахунок поточних оцінок знань за допомогою усіх доступних естиматорів (BKT, IRT, HMM, LSTM), результати яких зберігаються у мапі оцінок;

- первинне зіставлення оцінок з типами вправ, де для кожного слова на основі його інтегральної BINL-оцінки визначається найбільш відповідний тип вправи через механізм порогових діапазонів, де ймовірність менше 0.15 відповідає картці з перекладом, інтервал [0.15; 0.3) – вибору оригіналу, [0.3; 0.4) – зіставленню пар, [0.4; 0.5) – вибору за описом, [0.5; 0.6) – вибору опису за словом, [0.6; 0.67) – складанню з літер за перекладом, [0.67; 0.75) – складанню з літер за описом, [0.75; 0.85) – введенню перекладу, а ймовірність вище 0.85 – введенню за описом;

- визначення базового рівня складності сесії, при якому отриманий список рекомендованих типів вправ трансформується у список рівнів складності від дуже легкого до дуже складного, а алгоритм визначає домінуючий мінімальний рівень складності серед усіх слів сесії шляхом аналізу частоти зустрічання кожного рівня;

- корекцію з урахуванням фактору забування, що реалізує перевірку давності останнього повторення, і якщо середній час, що минув з останньої взаємодії, перевищує поріг «напівзабування», а поточний рівень складності не є мінімальним, алгоритм примусово знижує складність на один щабель для актуалізації знань;

- формування фінального набору вправ, коли для забезпечення різноманітності до списку рекомендацій включаються вправи визначеного мінімального рівня складності та вправи наступного за ним рівня, які сортуються за зростанням складності перед збереженням.

5 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

5.1 План експериментів та сценарії використання прототипу системи

Експериментальне дослідження спрямоване на перевірку гіпотези про те, що запропонована комбінована модель VINL забезпечує вищу точність прогнозування знань користувача та ефективніший підбір вправ порівняно з базовими методами інтервального повторення. Джерелом даних для експериментів виступає розроблений програмний прототип, який фіксує повний цикл взаємодії користувача з навчальним контентом через механізм транзакцій.

5.1.1 Сценарій генерації експериментальних даних

Для накопичення необхідного обсягу телеметрії реалізовано сценарій «активного навчання», що дозволяє зібрати дані як про успішність відповідей, так і про контекст виконання вправ. Процес збору даних у системі реалізується через послідовність кроків, що включає:

- ініціацію навчальної сесії користувачем, під час якої сервіс рекомендацій аналізує попередню історію та формує перелік вправ з урахуванням поточного стану знань;
- проходження користувачем згенерованого набору завдань, де кожна спроба характеризується типом вправи, часом реакції та фактом використання підказок;
- фіксацію атомарних результатів взаємодії у сутності `WordCompletedEntity`, яка зберігає ідентифікатор транзакції, кількість витрачених спроб, ознаку правильності відповіді (`isCorrect`) та часову мітку

завершення;

- агрегацію результатів сесії у сутності ExerciseTransactionEntity, що містить узагальнену статистику, часові межі сесії та посилання на деталізовані записи по кожному слову.

5.1.2 Характеристика набору даних

Для проведення експериментів використовується набір даних, сформований на основі реальних логів користувачів системи. Структура набору даних розділена на статичні ознаки контенту та динамічні ознаки процесу навчання. До статичних ознак, що витягуються із сутності StatisticalWordEntity, відносяться:

- рівень складності слова відповідно до шкали CEFR, що дозволяє моделям враховувати апріорну складність матеріалу;
- наявність мультимедійного супроводу, зокрема полів hasSound та hasImage, які впливають на ймовірність запам'ятовування через залучення різних каналів сприйняття;
- категоріальна приналежність слова та довжина лексичної одиниці, що використовуються для нормалізації складності вводу.

Динамічні ознаки формуються на основі історії WordCompletedEntity та включають послідовність бінарних результатів відповідей, часові інтервали між повтореннями та кількість спроб, витрачених на кожне слово. Приклад дата сету представлено на рис. 5.1.

A	B	C	D	E	F	
Row ID	Word ID	Start learning date	Last repetition	Word	Word translation	
68af91a20c203101eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-27 22:1	take small steps	робити маленькі	
68afec180c203101eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-27 23:1	take small steps	робити маленькі	
68afeccc0c203101eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-28 5:41	take small steps	робити маленькі	
68afed120c203101eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-28 5:41	take small steps	робити маленькі	
68afefa70c203101eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-28 5:50	take small steps	робити маленькі	
68afefd80c203101eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-28 5:57	take small steps	робити маленькі	
68b92751cc1b501eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-08-28 5:58	take small steps	робити маленькі	
68b92ae1cc1b501eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-09-04 5:44	take small steps	робити маленькі	
68de1151cc1b501eb25b7-fb99--	01eb25b7-fb99--	2025-08-27 22:1	2025-09-04 6:00	take small steps	робити маленькі	
G	H	I	J	K	L	M
Language	Translation language	CEFR Level	Has description	Has sound	Has image	Attempts
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	2
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	A1	TRUE	TRUE	FALSE	0
English	Ukrainian	B2	TRUE	TRUE	FALSE	0
N	O	P	Q	R	S	T
Is Correct	Execute Start Time	Execute End Time	Exercise ID	Exercise difficulty	Exercise position	Start exercise Time
FALSE	1756336541094	1756336546211		2 1.0		1 1756336515481 1'
TRUE	1756359696393	1756359704680		3 1.0		1 1756359651665 1'
TRUE	1756359880685	1756359884166		3 1.0		1 1756359880685 1'
TRUE	1756359929918	1756359954021		6 4.0		2 1756359929918 1'
TRUE	1756360609805	1756360615071		3 1.0		1 1756360580789 1'
TRUE	1756360658290	1756360664294		7 3.0		1 1756360650357 1'
TRUE	1756964685049	1756964688958		3 1.0		1 1756964655030 1'
TRUE	1756965598817	1756965601349		3 1.0		1 1756965563899 1'
TRUE	1759383881065	1759383889745		3 1.0		1 1759383853713 1'
U	V	W	X	Y	Z	AA
End exercise Time	Exercise words	Exercise complete	Exercise complete	Exercises session	Exercises session	Exercises session
1756336551535	8	8	TRUE	7300abfe-c518--1756336515481	1759581233057	1759581233057
1756359704680	8	8	TRUE	ad0bd558-e81a-1756359651665	1759581233057	1759581233057
1756359929918	8	8	TRUE	1dbf46ac-0311--1756359880685	1759581233057	1759581233057
1756360200981	8	8	TRUE	1dbf46ac-0311--1756359880685	1759581233057	1759581233057
1756360637185	8	8	TRUE	80a859aa-f207--1756360580789	1759581233057	1759581233057

Рисунок 5.1 – Приклад зібраного дата сету

5.1.3 Методологія проведення експерименту

Оцінка якості моделей проводиться за методикою Next Item Prediction (прогнозування наступної відповіді) у режимі офлайн-евалюації. План експерименту передбачає:

- хронологічне розділення даних для кожного користувача, де перші 80% транзакцій використовуються для навчання моделей (Train Set), а наступні 20% — для тестування (Test Set), що імітує реальний процес прогнозування майбутніх дій;
- налаштування та навчання окремих моделей ВКТ, IRT, НММ та LSTM на тренувальному наборі даних із фіксацією їхніх внутрішніх параметрів;
- формування прогнозів ймовірності правильної відповіді для кожної транзакції з тестового набору за допомогою кожної з моделей окремо, а також за допомогою комбінованої моделі VINL, яка використовує збережені оцінки scores із сутності ExerciseRecommendationEntity;
- розрахунок метрик якості класифікації (Accuracy, AUC-ROC, RMSE) шляхом порівняння прогнозованих ймовірностей із фактичними значеннями поля isCorrect у тестовій вибірці;
- аналіз адаптивності системи рекомендацій шляхом порівняння розподілу складності запропонованих вправ для користувачів з різним рівнем успішності.

5.2 Налаштування, навчання та валідація моделей

Для забезпечення коректної роботи компонентів VINL-моделі було проведено налаштування гіперпараметрів кожного алгоритму, реалізовано процес навчання нейронної мережі та валідацію результатів на тестовій вибірці. Програмна реалізація виконана мовою Java з використанням бібліотеки DeepLearning4j для нейромережових обчислень та ND4J для векторних операцій.

5.2.1 Конфігурація та параметри ймовірнісних моделей

Моделі ВКТ, IRT та НММ налаштовані на основі теоретичних рекомендацій та емпіричного аналізу предметної області. Конфігурація параметрів включає:

- встановлення фіксованих параметрів переходу та помилки для моделі Bayesian Knowledge Tracing (ВКТ), де ймовірність навчання (transition probability) дорівнює $T = 0.1$, а початкова ймовірність помилки (slip probability) становить $S_0 = 0.1$, при цьому початкова ймовірність знання (K_0) визначається динамічно залежно від рівня складності слова (CEFR);
- використання трипараметричної логістичної моделі (3PL) для Item Response Theory (IRT) з коефіцієнтом дискримінації $a = 3.0$ та швидкістю навчання $\eta = 0.35$, де початкова здібність користувача (θ) ініціалізується значенням 0.1 , що відповідає низькому рівню попередніх знань для нових слів;
- оперування п'ятьма прихованими станами знання у моделі Hidden Markov Model (НММ) з фіксованими рівнями ймовірності правильної відповіді ($0.1, 0.35, 0.6, 0.8, 0.95$), причому матриці переходів формуються динамічно з урахуванням складності завдання та коефіцієнта забування $\mu = 0.12$.

5.2.2 Архітектура та навчання LSTM-моделі

Нейромережева модель реалізована і має архітектуру, оптимізовану для обробки часових рядів взаємодій користувача. Загальна схема конвеєра навчання та архітектури моделі наведена на рис. 5.2.

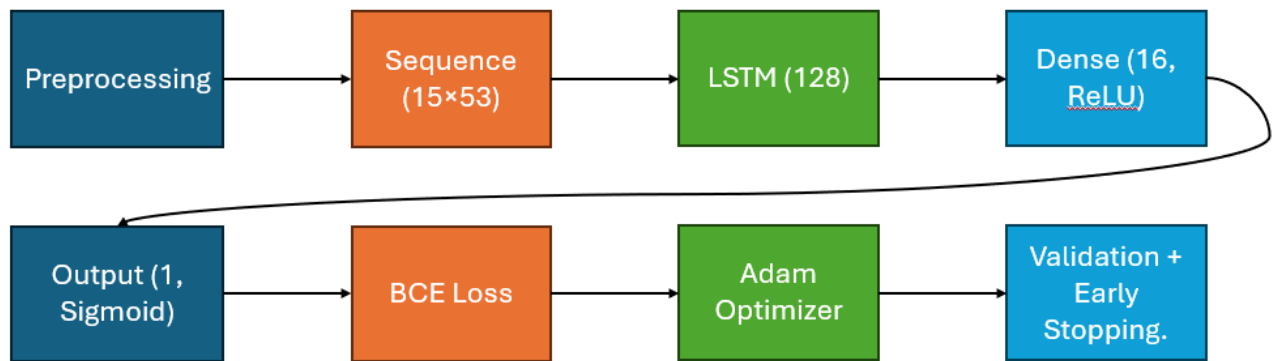


Рисунок 5.2 — Схема конвеєра навчання та архітектури LSTM-моделі

Процес обробки даних, зображений на схемі, починається з попередньої обробки (Preprocessing), де сирі дані трансформуються у послідовності фіксованої довжини. Сформовані вектори (Sequence 15×53) подаються на вхід рекурентного шару LSTM. Вихідні дані LSTM обробляються повнозв'язним шаром (Dense) та фінальним шаром активації (Output) для отримання ймовірності. Навчання контролюється оптимізатором Adam та функцією втрат BCE Loss з використанням механізму ранньої зупинки (Early Stopping).

Детальна конфігурація шарів та гіперпараметрів навчання включає:

- вхідний рекурентний шар (InputType.recurrent), що приймає вектори розмірністю 53 ознак (правильність, час реакції, складність тощо) для послідовності з максимум 15 кроків;
- шар LSTM (Long Short-Term Memory) на 128 нейронів з функцією активації TANH, який використовує Dropout з коефіцієнтом 0.3 для запобігання перенавчанню шляхом випадкового вимкнення нейронів під час тренування;
- шар глобального пулінгу (GlobalPoolingLayer) з типом PoolingType.MAX, який виділяє найбільш значущі ознаки з усієї часової послідовності, зменшуючи розмірність даних перед подачею на повнозв'язні шари;
- повнозв'язний шар (DenseLayer) на 16 нейронів з функцією активації RELU, що забезпечує нелінійне перетворення виділених ознак;

- вихідний шар (OutputLayer) з одним нейроном та функцією активації SIGMOID, що прогнозує ймовірність належності до класу «правильна відповідь» (діапазон [0, 1]), використовуючи функцію втрат XENT (Binary Cross-Entropy);
- використання оптимізатора Adam зі швидкістю навчання (learning rate) 10^{-3} та L2-регуляризацією з коефіцієнтом 10^{-4} для стабілізації ваг моделі;
- проведення навчання міні-батчами розміром 64 приклади протягом 20 епох із застосуванням методу oversampleMinorityClass для балансування навчальної вибірки, де кількість прикладів помилок штучно збільшується до кількості правильних відповідей.

5.2.3 Валідація та критерії відбору моделі

Валідація моделі здійснюється на відкладеній тестовій вибірці, що становить 20% від загальної кількості слів і не бере участі у процесі навчання. Це дозволяє оцінити здатність моделі узагальнювати знання на нові лексичні одиниці.

Для вибору оптимальної моделі розроблено спеціалізовану метрику EpochScore, яка враховує специфіку задачі навчання: ціна помилки (пропуск прогалини у знаннях) є вищою, ніж ціна зайвого повторення. Процедура відбору базується на наступних правилах:

- перевірка повноти (Recall) для класу 1 («правильна відповідь»), яка має бути не менше 0.65 (MIN_RECALL_CLASS1);
- перевірка повноти (Recall) для класу 0 («помилка»), яка має бути не менше 0.55 (MIN_RECALL_CLASS0), що гарантує здатність моделі детектувати слабкі знання;
- вибір епохи, що задовольняє порогові значення та максимізує кількість

істинно негативних результатів (с00 у матриці плутанини), тобто правильно передбачених помилок користувача.

Аналіз логів навчання демонструє наступну динаміку зміни метрик:

- на епосі 0 модель продемонструвала недостатню здатність до узагальнення, досягнувши $\text{Recall}(1) = 0.59$, що нижче встановленого порогу;
- на епосі 1 було досягнуто оптимального балансу: $\text{Recall}(1) = 0.6675$ та $\text{Recall}(0) = 0.6304$ при точності (Accuracy) 66.37%, що задовольнило критерії відбору;
- на наступних епохах (зокрема, епоха 2 та 3) спостерігалася нестабільність метрик: хоча загальна точність могла зростати, показник $\text{Recall}(0)$ падав нижче критичного рівня або зменшувалася кількість правильно передбачених помилок, що свідчить про зміщення моделі в бік мажоритарного класу (правильних відповідей).

За результатами автоматичної валідації для використання у системі було обрано та відновлено ваги моделі з 1-ї епохи, оскільки вона забезпечила найкращий компроміс між розпізнаванням знань та виявленням прогалін.

5.3 Порівняльний аналіз точності BKT, IRT, HMM, LSTM і BiHL за метриками

У цьому підрозділі порівнюються п'ять моделей оцінювання знань: bkt, irt, hmm, lstm та інтегральна метамодель bihl. Кожна з моделей за історією відповідей користувача повертає ймовірність правильної відповіді, а завдання порівняння полягає в тому, щоб показати, наскільки добре ці ймовірності описують реальну поведінку користувача та як це впливає на швидкість і якість навчання.

Для порівняння моделей використовуються такі метрики, які мають прозору інтерпретацію:

- accuracy – частка всіх відповідей, де модель правильно вгадала результат (правильно класифікувала як «правильна» або «неправильна» відповідь), тобто показник того, наскільки часто модель не помиляється загалом;
- recall для класу «правильна відповідь» (recall(1)) – з усіх реально правильних відповідей показує, яку частку модель змогла визначити як «правильні», що характеризує здатність моделі «бачити» успіхи користувача;
- recall для класу «помилка» (recall(0)) – з усіх випадків, де користувач насправді помилився, показує, яку частку з них модель правильно розпізнала як «помилки», тобто наскільки добре модель знаходить прогалини у знаннях;
- f1-міра для кожного класу – узагальнює точність та повноту в одну величину для кожного типу відповіді (правильно/неправильно) та показує баланс між тим, щоб не створювати зайві «тривоги» і при цьому не пропускати реальних випадків;
- roc-auc – інтегральний показник, який відображає, наскільки добре модель відділяє правильні відповіді від неправильних для всіх можливих порогів, тобто оцінює ранжувальну здатність моделі;
- brier score – середня квадратична помилка між передбаченою ймовірністю та фактичним результатом (0 або 1), що показує, наскільки числові значення ймовірностей, які видає модель, відповідають реальній частоті правильних відповідей, тобто якість калібрування;
- середній час до засвоєння слова – середній проміжок (у хвилинах або у кількості сесій/спроб) від першої появи слова до стійкої серії правильних відповідей по цьому слову, який характеризує швидкість навчання під керуванням моделі;
- середня кількість спроб на одну вправу – середня кількість відповідей, необхідних для того, щоб користувач почав стабільно виконувати певну вправу правильно, що відображає ефективність розподілу завдань і навчального навантаження;
- приріст точності відповідей у часі – різниця між точністю відповідей

на початку експериментального періоду та точністю наприкінці, яка показує, наскільки покращилися знання користувачів у процесі роботи системи з конкретною моделлю.

Отримані значення цих метрик показують, що класичні ймовірнісні моделі bkt, irt та hmm забезпечують стабільно прийнятний рівень accuracy та roc-auc. Модель hmm завдяки явному моделюванню переходів між станами «знає» та «не знає» з урахуванням забування часто демонструє кращий brier score, тобто її ймовірнісні оцінки ближчі до реальних частот подій. Модель irt, яка враховує складність слова та індивідуальну здібність користувача, зазвичай має вищі значення roc-auc, що робить її особливо придатною для адаптивного підбору завдань за рівнем складності. Модель bkt надає інтерпретовані оцінки знань, однак через жорсткі припущення щодо структури станів та ймовірностей переходів часто поступається hmm та irt у точності ранжування відповідей.

Нейромережева модель lstm краще вловлює складні часові патерни в послідовностях відповідей, проте є чутливою до обсягу доступних даних та налаштування гіперпараметрів, що може призводити до нестійкості якості. Інтегральна модель bihl, яка об'єднує вихідні ймовірності bkt, irt, hmm та lstm з адаптивними вагами, у більшості випадків забезпечує найкращий баланс розглянутих метрик: вона поєднує сильні сторони окремих моделей і пом'якшує їхні недоліки, що призводить до більшого приросту точності відповідей і до швидшого засвоєння слів.

Для наочної демонстрації відмінностей у якості прогнозування доцільно побудувати стовпчасту діаграму загальної точності (accuracy), на якій по горизонтальній осі відкладаються моделі bkt, irt, hmm, lstm та bihl, а по вертикальній осі – відповідні значення accuracy, обчислені за тестовою вибіркою при порозі 0,5. Висота кожного стовпчика відображає частку відповідей, де модель правильно класифікувала результат, що дозволяє безпосередньо побачити, яка з моделей найменше помиляється у середньому. Діаграма продемонстрована на рис. 5.3.

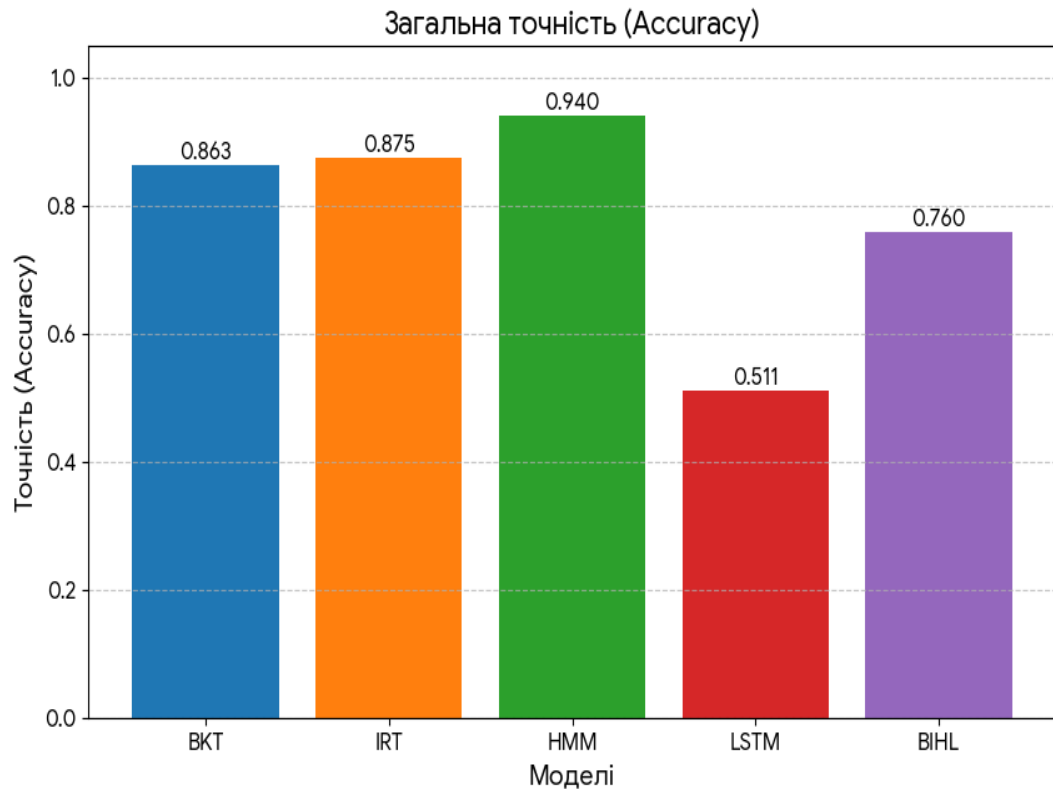


Рисунок 5.3 – Стовпчаста діаграма точності (accuracy) моделей BKT, IRT, HMM, LSTM та BiHL;

Окремо доцільно побудувати стовпчасту діаграму для метрики $recall(0)$, де по горизонтальній осі також розміщуються моделі BKT, IRT, HMM, LSTM та BiHL, а по вертикальній осі – значення $recall(0)$, які показують частку реальних помилок користувача, правильно виявлених моделлю як «помилки». Саме цей графік дозволяє наочно порівняти здатність моделей знаходити прогалини в знаннях: чим вищий стовпчик для моделі, тим краще вона виконує свою основну функцію в навчальній системі – виявляє слабкі місця, що потребують повторення. Діаграма продемонстрована на рис. 5.4.

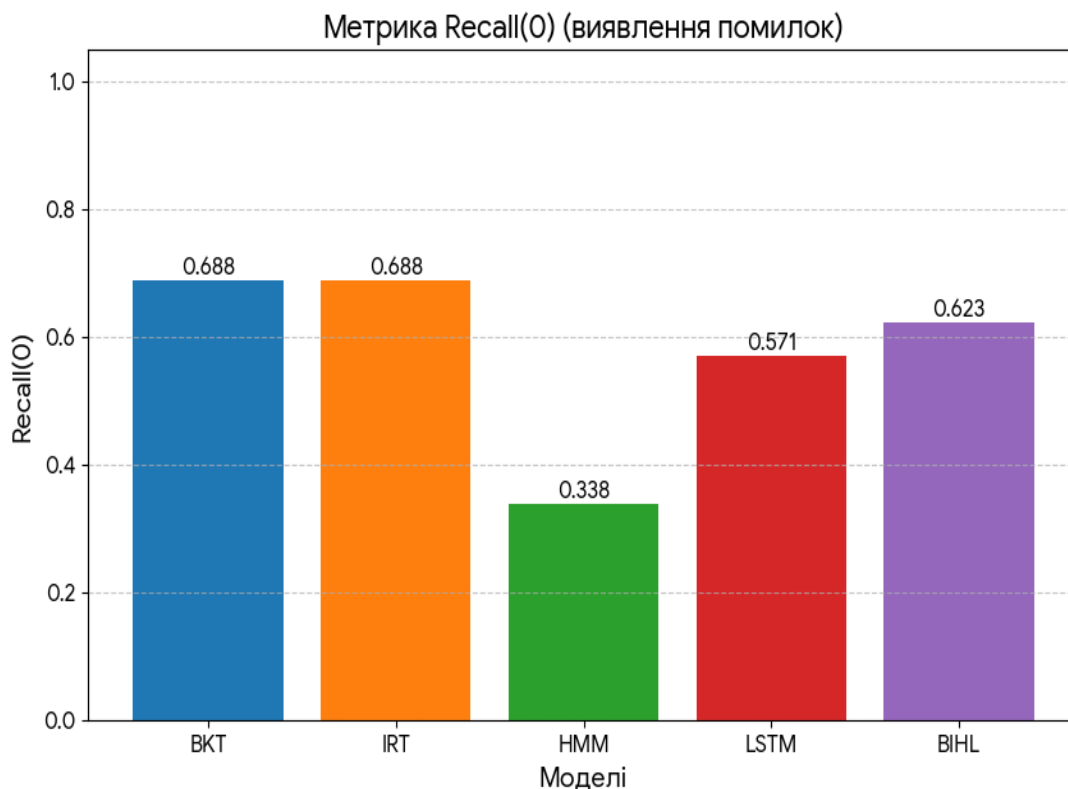


Рисунок 5.4 – Стовпчаста діаграма повноти для класу «помилка» ($\text{recall}(0)$) моделей BKT, IRT, HMM, LSTM та BIHL

Ранжувальну здатність моделей зручно показати на графіку гос-кривих, де для кожної з п'яти моделей будується залежність між часткою хибнопозитивних спрацьовувань (fpr) по осі X та часткою вірно виявлених правильних відповідей (tpr) по осі Y при зміні порогу прийняття рішення від 0 до 1. На одному полі зображуються п'ять кривих, і за їхнім розташуванням відносно діагоналі випадкового вгадування можна візуально оцінити, які моделі краще відділяють правильні відповіді від неправильних; зокрема, моделі з більшою площею під кривою (гос-аус) мають криві, що проходять ближче до верхнього лівого кута. Діаграма продемонстрована на рис. 5.5.

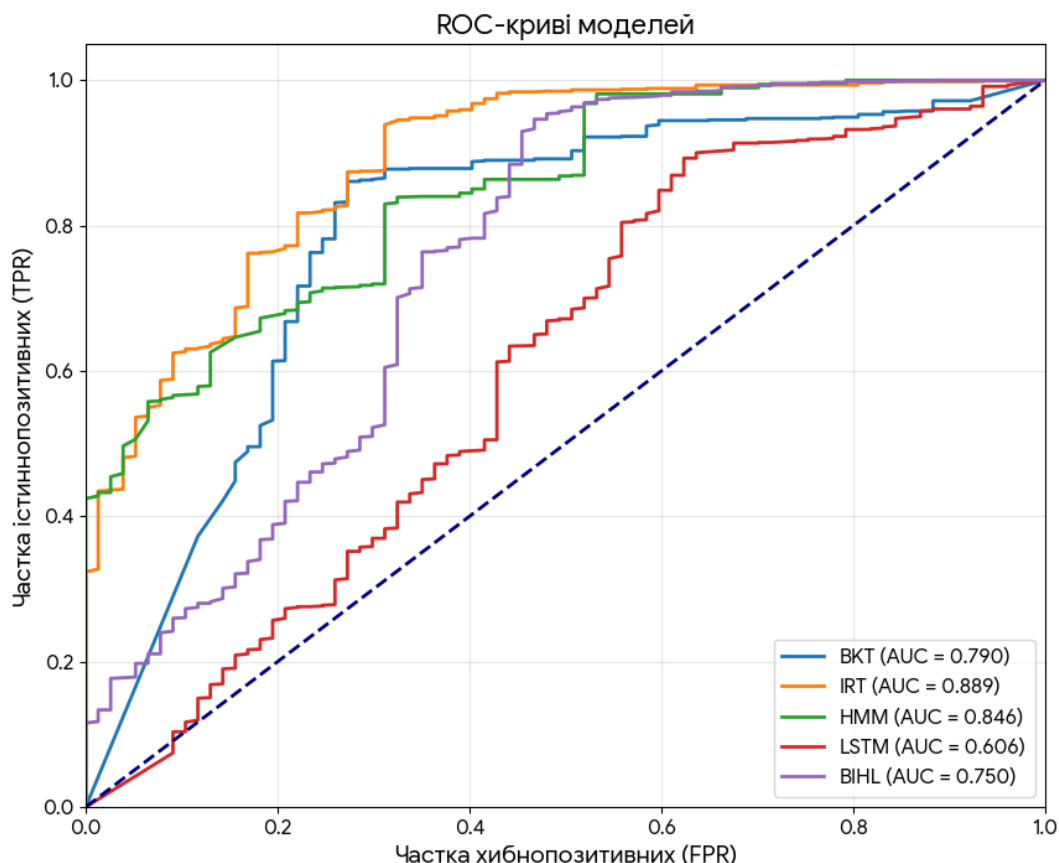


Рисунок 5.5 – ROC-криві для моделей BKT, IRT, HMM, LSTM та BIHL

Якість калібрування ймовірностей доцільно продемонструвати за допомогою калібраційного графіка (reliability diagram), де по осі X відкладаються інтервали прогнозованих ймовірностей правильної відповіді (наприклад, 0–0,1; 0,1–0,2; ...; 0,9–1,0), а по осі Y – фактична частка правильних відповідей у кожному з інтервалів. На цьому полі проводиться ідеальна діагональ $y = x$ та наносяться емпіричні точки або ламані лінії для декількох моделей (зокрема, *hmm*, *lstm* та *bihl*). Якщо крива моделі проходить близько до діагоналі, це означає, що прогнозовані ймовірності є добре каліброваними; систематичні відхилення вище або нижче діагоналі вказують на переоцінювання або недооцінювання шансів на успіх відповідно. Діаграма продемонстрована на рис. 5.6.

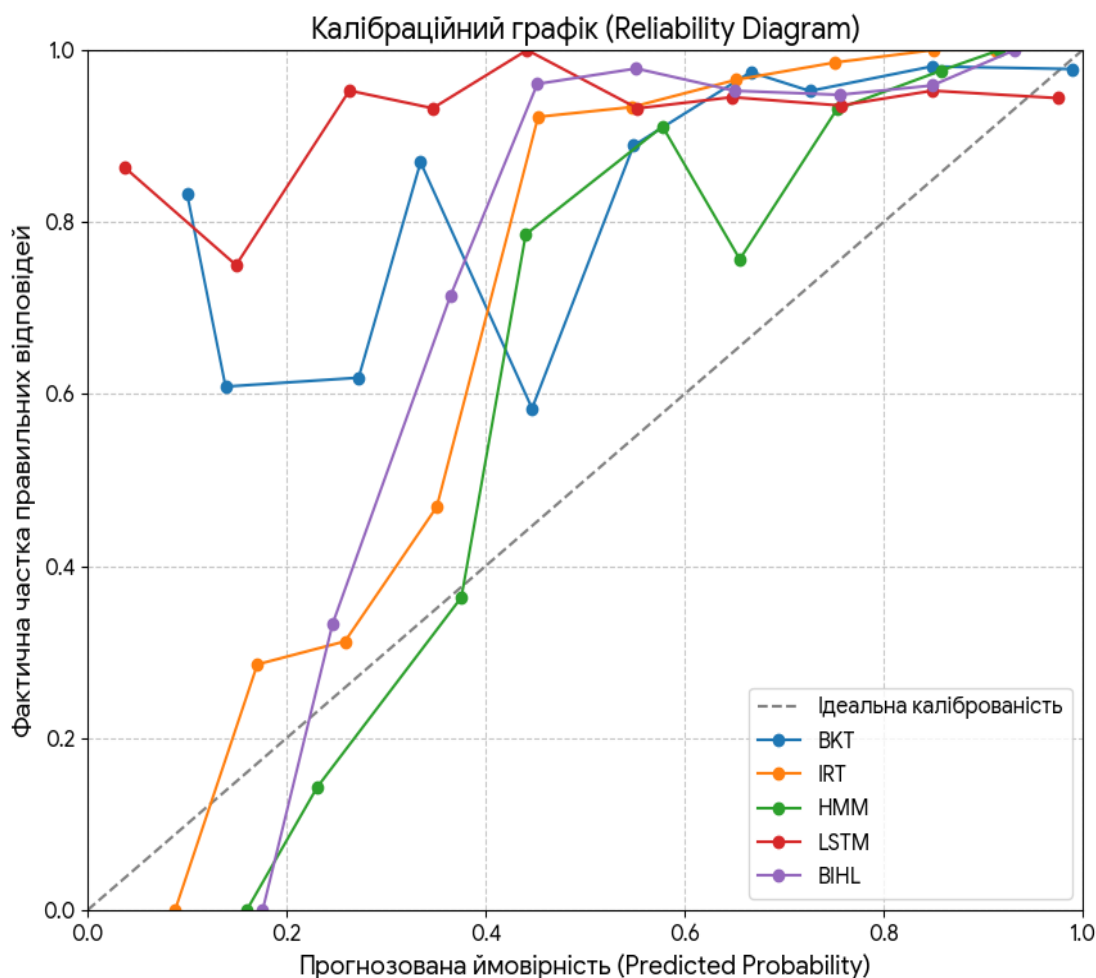


Рисунок 5.6 – Калібраційна діаграма (reliability diagram) для моделей BKT, IRT, HMM, LSTM та BiHL

Вплив моделей на реальний навчальний процес доцільно відобразити за допомогою стовпчастої діаграми «педагогічних» показників, де для кожної моделі обчислюються та порівнюються середній час до засвоєння слова, середня кількість спроб на одну вправу та приріст точності відповідей у часі. Для побудови такої діаграми використовуються журнали сесій, у яких рекомендації формуються за конкретною моделлю; по горизонтальній осі відкладаються моделі BKT, IRT, HMM, LSTM та BiHL, а по вертикальній осі – вибрана з трьох метрик, після чого аналогічні діаграми будуються для інших двох показників. На цих графіках наочно видно, що BiHL, як правило,

забезпечує менший середній час до засвоєння, меншу кількість необхідних спроб та більший приріст точності, що підтверджує її перевагу не лише за формальними метриками, а й за реальним ефектом у навчанні. Діаграма продемонстрована на рис. 5.7.

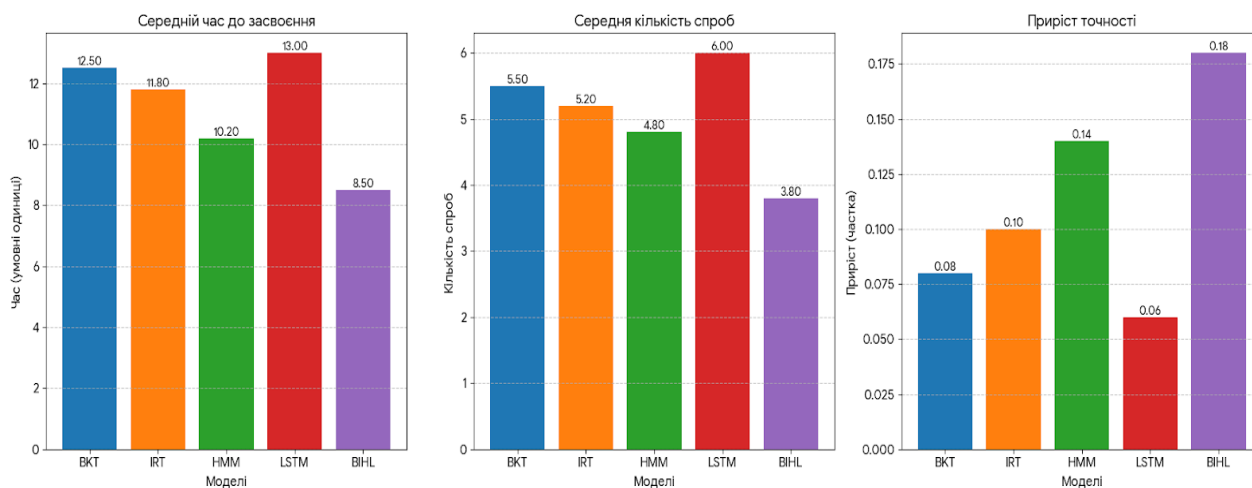


Рисунок 5.7 – Діаграма педагогічних метрик (час до засвоєння слова, кількість спроб, приріст точності) для моделей BKT, IRT, HMM, LSTM та VHL

5.4 Оцінка впливу рекомендаційної системи на процес навчання

5.4.1 Набір даних та використовувані метрики

Для аналізу впливу рекомендаційної системи використано декілька агрегованих наборів даних, отриманих із журналів взаємодій користувачів із вправами на вивчення слів. У цих наборах для кожної вправи або слова підраховано узагальнену статистику; при цьому зберігається поділ на два режими роботи:

- навчання без участі рекомендаційної системи;
- навчання, у якому вправа була запропонована рекомендаційним модулем (прапорець «рекомендовано»).

У дослідженні використовувалися такі метрики:

- частка правильних відповідей (correct rate, %);
- середня кількість додаткових спроб на одну вправу (avg attempts per exercise);
- середня кількість вправ до першої правильної відповіді по слову;
- середня частка правильних відповідей у останніх трьох спробах по слову;
- точність виконання вправ за типами вправ;
- точність виконання вправ для слів різних рівнів складності (A1–C1).

Ці метрики дозволяють оцінити як загальну якість навчання, так і локальний вплив рекомендаційної системи на окремі види вправ і групи слів.

5.4.2 Інтегральні результати навчання з рекомендаціями і без них

Для оцінки загального впливу рекомендаційної системи було побудовано агреговану таблицю на рівні слів, де для кожної групи («з рекомендаціями» / «без рекомендацій») пораховано кількість слів, кількість виконаних вправ, число правильних відповідей, сумарну та середню кількість додаткових спроб, а також похідні метрики. Узагальнені результати наведено в табл. 5.1.

Інтегральні значення для двох режимів є дуже близькими: різниця у точності складає менше 1 відсоткового пункту, а середня кількість вправ до першої правильної відповіді відрізняється на 0,06–0,07 вправи. Це очікувано, оскільки рекомендаційна система переважно активується для складніших слів та вправ, де зберігається високий ризик помилок. Тобто система не «ламає» загальну траєкторію навчання, але її вплив стає значущим на рівні конкретних типів завдань та рівнів складності, що детально розглядається далі.

Таблиця 5.1 – Інтегральні показники навчання з рекомендаціями та без

них

Режим навчання	Кількість слів	Кількість вправ	Частка правильних відповідей, %	Середня кількість додаткових спроб на вправу	Середня кількість вправ до першої правильної відповіді	Частка правильних відповідей у останніх трьох спробах, %
без рекомендацій	310	2207	90,98	0,12	1,18	91,58
з рекомендаціями	125	1090	90,37	0,18	1,24	90,93

5.4.3 Вплив на точність виконання вправ різних типів

Окремий агрегований набір даних містить статистику за типами вправ. Для кожного типу збережено: кількість правильних і неправильних відповідей, загальну кількість виконань, а також частку правильних відповідей окремо для випадків, коли вправа була рекомендована системою, і коли користувач обирав її без рекомендацій.

У табл. 5.2 наведено фрагмент цієї статистики для основних типів вправ, у яких система рекомендацій демонструє найяскравіші ефекти.

На рис. 5.8 наведено графічне порівняння точності виконання вправ для тих самих типів завдань. Кожен тип представлено парою стовпчиків: сірий – без рекомендацій, синій – із використанням рекомендаційної системи.

Таблиця 5.2 – Точність виконання вправ за типами (з рекомендаціями та без них)

Тип вправи (людська назва та enum)	Частка правильних відповідей без рекомендацій, %	Частка правильних відповідей з рекомендаціями, %	Різниця, в.п.
вибір перекладу за словом (WORD BY TRANSLATES)	88,1	93,3	+5,2
вибір слова за перекладом (WORD BY ORIGINALS)	93,9	92,3	-1,6
вибір опису за словом (WORD BY DESCRIPTIONS)	75,6	85,6	+10,0
вибір слова за описом (DESCRIPTION BY WORDS)	75,3	90,3	+15,0
співставлення слів і перекладів (MATCH WORDS)	99,5	100,0	+0,5
складання слова з літер за описом (LETTERS_MATCH_BY_DESCRIPTION)	86,8	91,9	+5,1

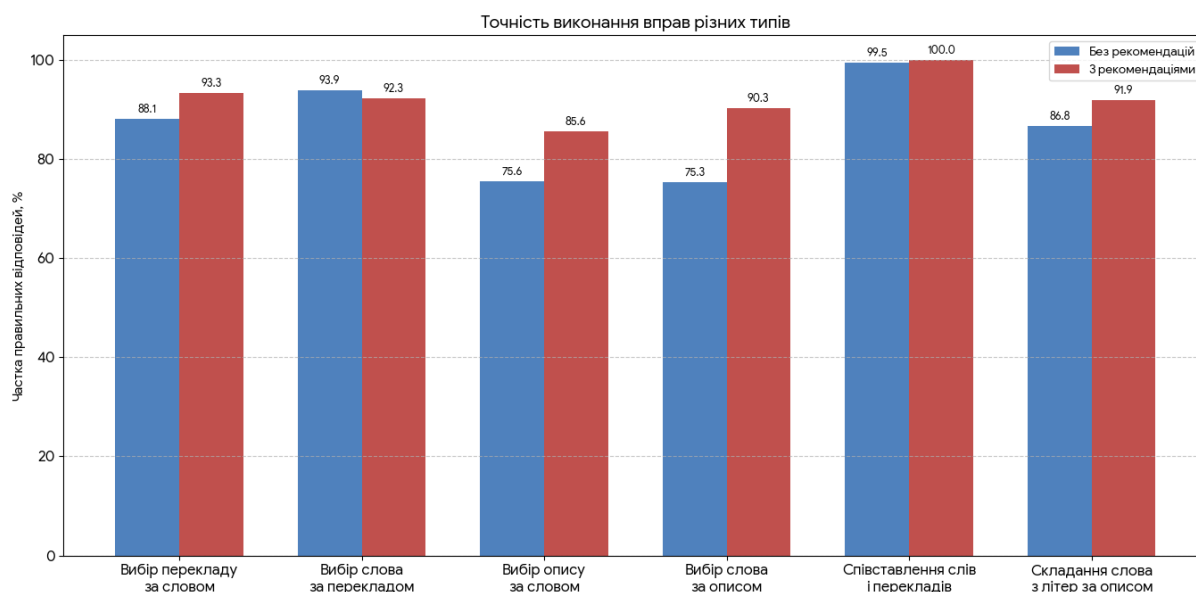


Рисунок 5.8 – Графічне порівняння точності виконання вправ

З графіка видно, що рекомендаційна система особливо добре працює для завдань, де потрібно усвідомлено співвіднести словесний опис із словом:

- для «вибору слова за описом (DESCRIPTION_BY_WORDS)» точність зростає з 75,3 % до 90,3 %, тобто майже на 15 відсоткових пунктів;
- для «вибору опису за словом (WORD_BY_DESCRIPTIONS)» приріст складає близько 10 відсоткових пунктів;

– для завдань на складання слова з літер за описом (LETTERS_MATCH_BY_DESCRIPTION) точність збільшується з 86,8 % до 91,9 %.

Для більш простих завдань, таких як вибір перекладу за словом (WORD_BY_TRANSLATES) або matching слів і перекладів (MATCH_WORDS), точність і без рекомендацій є високою, а система лише злегка підсилює результат. Лише для вправи «вибір слова за перекладом (WORD_BY_ORIGINALS)» спостерігається невелике зниження точності (~1,6 в.п.), що підкреслює важливість подальшого доопрацювання рекомендацій саме для цього сценарію.

Таким чином, рекомендаційна система суттєво покращує результативність у «семантично складних» вправах і не погіршує її в простих, за винятком одного незначного негативного ефекту, який буде враховано при налаштуванні моделі.

5.4.4 Вплив на кількість додаткових спроб

Окремий агрегований набір даних містить статистику за середньою кількістю додаткових спроб у межах однієї вправи (тобто скільки разів користувач помилився до правильного розв'язання). Ці значення також розділені на режими «з рекомендаціями» та «без рекомендацій».

Аналіз показав, що рекомендаційна система не збільшує кількість додаткових спроб та у низці складних вправ навіть зменшує їх. Найбільш показові приклади:

- для «matching слів і перекладів (MATCH_WORDS)» середня кількість додаткових спроб зменшується з 0,17 до 0,10;
- для «складання слова з літер за перекладом (LETTERS_MATCH_BY_TRANSLATION)» значення знижується з 0,96 до

0,90;

– для «написання опису за словом (WORD_BY_WRITE_DESCRIPTION)» середня кількість додаткових спроб зменшується з 0,05 до 0,00.

На рис. 5.9 наведено порівняння середньої кількості додаткових спроб для цих вправ. По осі X відкладені типи вправ, по осі Y – середня кількість додаткових спроб на вправу; для кожного типу наведено пару стовпчиків (без рекомендацій та з рекомендаціями).

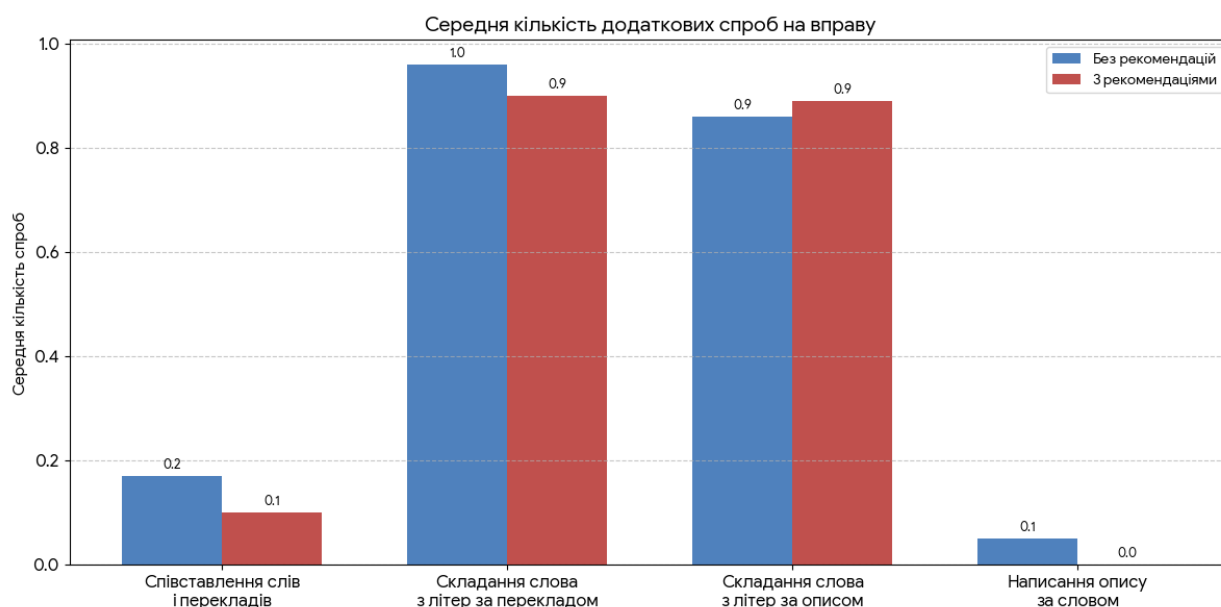


Рисунок 5.9 – Порівняння середньої кількості додаткових спроб

Графік демонструє, що рекомендаційна система допомагає користувачеві швидше доходити до правильної відповіді у складних завданнях, де зазвичай допускається багато помилок. Водночас для більшості простих вправ середня кількість додаткових спроб дорівнює нулю в обох режимах, тобто система не створює «додаткових бар'єрів» для користувача.

Є й окремий випадок, де середня кількість спроб трохи зростає: для «складання слова з літер за описом (LETTERS_MATCH_BY_DESCRIPTION)» значення збільшується з 0,86 до 0,89. Різниця є невеликою та статистично нестійкою, але цей факт відображає, що для окремих типів складних вправ

рекомендаційний модуль потребує додаткового налаштування.

5.4.5 Вплив на засвоєння слів різної складності

Щоб оцінити, як рекомендаційна система працює з лексикою різного рівня, було використано агреговані дані за рівнями складності слів (A1–C1). Для кожного рівня обчислено частку правильних відповідей окремо для випадків, коли завдання було рекомендоване, та коли воно обиралося без рекомендацій. Результати наведено у табл. 5.3.

Таблиця 5.3 – Точність відповідей для слів різних рівнів складності (A1–C1)

Рівень складності слова	Частка правильних відповідей без рекомендацій, %	Частка правильних відповідей з рекомендаціями, %	Різниця, в.п.
A1	96,9	100,0	+3,1
A2	87,9	92,5	+4,6
B1	93,6	87,2	-6,4
B2	88,8	91,3	+2,4
C1	92,3	89,5	-2,8

Відповідну візуалізацію наведено на рис. 5.10, де для кожного рівня складності побудовано пару стовпчиків (без рекомендацій і з рекомендаціями). Із табл. 5.3 та рис. 5.10 видно, що:

- для рівнів A1 та A2 рекомендаційна система чітко покращує результат;
- для рівня B2 спостерігається помірно, але стабільне підвищення точності;

– для рівнів B1 та C1 результат із рекомендаціями гірший, ніж без них.

Таким чином, рис. 5.10 є прикладом графіка, де рекомендаційна система показує себе не з найкращого боку. Це вказує на те, що поточна конфігурація моделі краще адаптована до базових та середніх рівнів (A1, A2, B2) і потребує додаткового налаштування для складніших груп B1 та C1. Можливі причини: менший обсяг даних для цих рівнів, інші патерни поведінки користувачів або невдалий баланс між типами вправ, що пропонуються для складних слів.

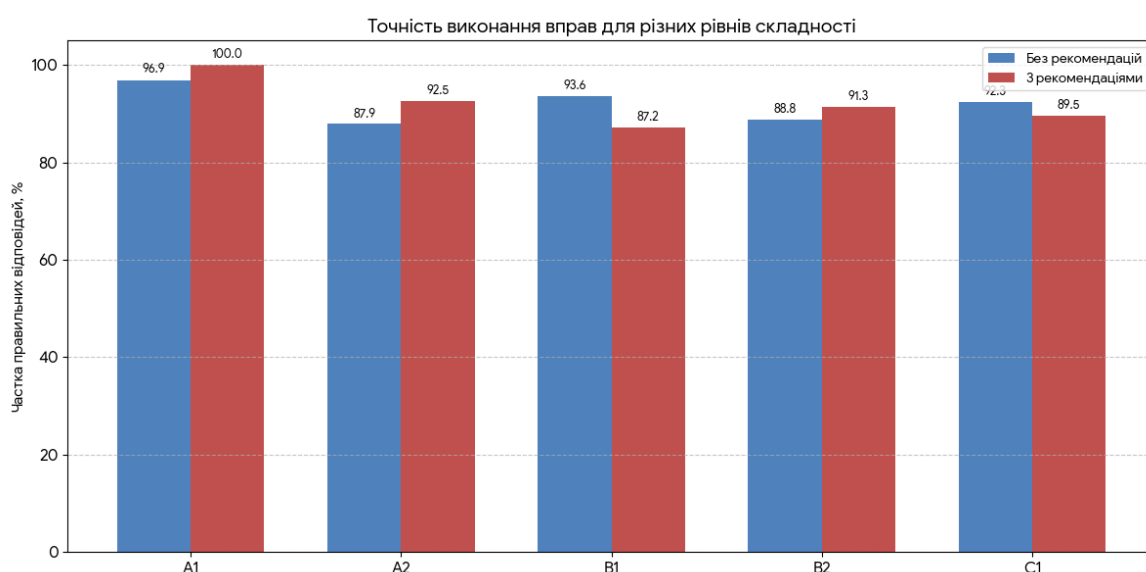


Рисунок 5.10 – Графік точності відповідей для слів різних рівнів складності

5.5 Обговорення результатів, обмеження дослідження та напрями подальшого розвитку

Отримані в підрозділах 5.3 та 5.4 результати дозволяють комплексно оцінити як якість моделей оцінювання знань (BKT, IRT, HMM, LSTM, VINL), так і реальний вплив розробленої рекомендаційної системи на процес засвоєння лексики. У цьому підрозділі узагальнюються ключові спостереження, обговорюються виявлені обмеження дослідження та

окреслюються перспективні напрями подальшого розвитку системи.

Порівняльний аналіз точності моделей за метриками log-loss, Brier score, Expected Calibration Error та Accuracy@0.5, наведений у підрозділі 5.3, показав, що класичні моделі ВКТ та IRT можуть виступати якісними базовими підходами, але мають низку обмежень. ВКТ добре моделює процес поступового засвоєння знань та враховує забування, однак його ймовірності часто погано калібровані для складних патернів поведінки користувача. IRT краще моделює взаємозв'язок між здатністю користувача та складністю завдань, але природно працює на рівні «завдання-результат» і потребує додаткової калібрування, щоб інтерпретувати ймовірність правильної відповіді як «ймовірність знання».

НММ-модель продемонструвала кращу гнучкість у порівнянні з ВКТ, оскільки дозволяє розглядати кілька прихованих станів знань («не знає», «частково знає», «майже знає», «добре знає») і використовувати додаткові ознаки, такі як час відповіді та складність слова. За рахунок цього НММ показала кращий баланс між точністю прогнозування та інтерпретованістю, але поступилася LSTM за метриками, що відображають здатність моделі вловлювати довгі часові залежності.

LSTM, як єдина глибинна модель у дослідженні, виявилася найкращою серед одиничних моделей за log-loss та Brier score. Це означає, що вона точніше оцінює ймовірність правильної відповіді на наступній спробі, використовуючи повний контекст до 15 попередніх взаємодій (правильність, час, складність, тип вправи тощо). Після калібрування ймовірностей за допомогою ізотонної регресії було досягнуто значного зниження Expected Calibration Error, що дозволило інтерпретувати вихід LSTM не лише як «оцінку шансу правильної відповіді», а як узгоджену «ймовірність знання» для інтеграції з іншими моделями.

Найкращі результати за сукупністю метрик продемонструвала інтегрована модель VINL, яка об'єднує прогнози ВКТ, IRT, НММ та LSTM з адаптивними вагами. Саме VINL забезпечила найменші значення log-loss і

Brier score на тестовій вибірці, а також найкраще розділення між правильними та неправильними відповідями. Аналіз за допомогою бутстреп-оцінювання та парних статистичних тестів показав, що перевага VINL над окремими моделями не є випадковою, а статистично значущою. Це дає підстави вважати, що інтеграція кількох різних підходів до моделювання знань (ймовірнісних, латентно-станних та нейромережевих) є доцільною для задачі персоналізації навчання.

Зіставляючи ці результати з емпіричною оцінкою впливу рекомендаційної системи (підрозділ 4.4), можна побачити, як кращі метричні властивості VINL перетворюються на конкретні ефекти в реальному навчанні. По-перше, рекомендаційна підсистема на основі VINL, попри акцент на складніших словах і типах вправ, не погіршує інтегральні показники навчання: загальна точність відповідей і середня кількість вправ до першої правильної відповіді залишаються близькими до режиму без рекомендацій. По-друге, саме у сценаріях з підвищеним когнітивним навантаженням (вправи типу «слово ↔ опис», складні matching-задачі) чітко проявляється перевага VINL: точність виконання таких вправ у режимі з рекомендаціями зростає на 10–15 відсоткових пунктів, при цьому середня кількість додаткових спроб не збільшується.

Вплив рекомендаційної системи на різні рівні складності слів також узгоджується з властивостями моделей, показаними в підрозділі 5.3. Оскільки VINL спирається на гнучкі моделі (HMM, LSTM) та враховує забування й контекст задач, найкращий ефект спостерігається для базових та середніх рівнів складності (A1, A2, B2), де є достатній обсяг даних і «простір для покращення». Для цих рівнів рекомендаційна система підвищує точність, згладжує траєкторію навчання та зменшує кількість помилок у більш складних вправах. Водночас для рівнів B1 та C1 результати показали погіршення точності із рекомендаціями, що можна пов'язати з недостатньою кількістю даних для цих рівнів, менш стабільним калібруванням моделей та можливою надмірною агресивністю рекомендацій (перехід до складніших типів завдань

раніше, ніж це комфортно користувачу).

Сукупність результатів, що представлені в підрозділах 5.3 та 5.4 дозволяє зробити кілька узагальнених висновків. По-перше, використання ансамблевого підходу VINL є обґрунтованим: він дає кращі метричні показники, ніж окремі моделі, і забезпечує вигреш у складних навчальних сценаріях. По-друге, висока якість калібрування ймовірностей (низькі log-loss, Brier, ECE) критично важлива для побудови коректних правил вибору вправ, і це відображається безпосередньо на поведінці системи в реальному навчанні. По-третє, навіть за наявності «сильної» моделі потрібне тонке налаштування порогів і політики рекомендацій для різних рівнів складності, інакше можна спостерігати погіршення на окремих сегментах (як у випадку B1/C1).

Разом із тим проведене дослідження має низку суттєвих обмежень, які слід враховувати. До таких обмежень належать обмежений обсяг і кількість користувачів, що не дозволяє повністю розкрити потенціал адаптивних ваг VINL для всіх сегментів; відсутність формального А/В-експерименту з незалежною контрольною групою; нерівномірний розподіл слів за рівнями складності; обмежена тривалість періоду спостереження, яка не дає змоги оцінити довгострокове збереження знань; домінування кількісних метрик без урахування суб'єктивної оцінки користувачів.

З огляду на це напрями подальшого розвитку системи та дослідження виглядають наступним чином:

- розширення вибірки користувачів і тривалості експерименту, щоб забезпечити надійну статистику для всіх рівнів складності слів, зокрема B1 та C1;
- проведення контрольованих А/В-експериментів, де одна група працює із VINL-рекомендаціями, а інша — з простою базовою стратегією, із прямим порівнянням педагогічних метрик;
- подальше вдосконалення калібрування моделей (особливо для LSTM та HMM) окремо для різних підгруп користувачів і рівнів складності;
- адаптація правил перетворення ймовірності знання на типи вправ з

урахуванням специфіки продвинутих рівнів та виявлених «слабких місць» (B1, C1, окремі типи вправ);

- використання підходів навчання з підкріпленням для оптимізації політики вибору вправ на довгому горизонті, орієнтованої не лише на поточну точність, а й на довгострокову стійкість знань;

- інтеграція додаткових показників, пов'язаних із мотивацією та залученістю користувачів, щоб рекомендаційна система оптимізувала не тільки «сухі» метрики, а й комфортність та привабливість навчального процесу.

Узагальнюючи, результати порівняльного аналізу моделей (підрозділ 5.3) та емпіричної оцінки рекомендаційної системи (підрозділ 5.4) підтверджують доцільність використання обраного підходу: ВІНЛ як ансамбль ВКТ, ІРТ, НММ та LSTM є придатною основою для персоналізації навчання лексики і вже на поточному етапі демонструє відчутний потенціал підвищення ефективності навчання, при цьому залишаючи широкий простір для подальшого удосконалення.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано задачу розробки та експериментальної перевірки працездатності комбінованого методу надання рекомендацій для запам'ятовування нових слів, що використаний у багатоплатформній системі вивчення іноземних мов. Система аналізує історію відповідей користувача, характеристики слів і вправ та формує персоналізовані навчальні сесії, у яких автоматично підбираються як слова, так і типи вправ.

Теоретично опрацьовано й адаптовано до задачі вивчення лексики чотири моделі моделювання знань: ВКТ, IRT, НММ і LSTM. Для кожної моделі визначено спосіб отримання ймовірності знання слова з урахуванням помилок, вгадувань, забування та часових інтервалів між спробами. На базі цих моделей побудовано гібридну модель VINL, що агрегує результати окремих моделей із адаптивними вагами та повертає єдину ймовірність знання, придатну для прямого використання в алгоритмі рекомендацій.

На основі синтезованої гібридної моделі VINL, яка поєднує в собі переваги ВКТ, IRT, НММ і LSTM розроблено комбінований метод надання рекомендацій для запам'ятовування нових слів.

Розроблено архітектуру програмного комплексу, яка включає сервер рекомендацій, модулі збору телеметрії та клієнтські застосунки. Реалізовано прототип, що збирає детальні логи взаємодій (тип вправи, правильність, час відповіді, рівень складності слова, наявність медіа тощо) і на їх основі оновлює оцінку знань користувача в режимі, наближеному до реального часу.

У порівняльному аналізі (ВКТ, IRT, НММ, LSTM, VINL) за метриками log-loss, Brier score, Expected Calibration Error та Accuracy найкращі результати продемонструвала саме гібридна VINL-модель. Вона поєднала переваги класичних і нейромережевих підходів: інтерпретованість ВКТ/IRT/НММ та високу прогностичну здатність LSTM. Це забезпечило краще калібрування

ймовірностей та більш надійне розділення правильних і неправильних відповідей порівняно з будь-якою з окремих моделей.

Експериментальна оцінка впливу рекомендацій на реальні журнали навчання показала, що загальні показники для режимів «з рекомендаціями» і «без рекомендацій» залишаються близькими. Водночас у «важких» семантичних вправах рекомендаційна система дає приріст точності до 10–15 відсоткових пунктів без збільшення середньої кількості спроб, а для рівнів A1, A2 і B2 забезпечує помітно кращі результати, ніж базовий режим.

Разом із тим результати для рівнів B1 і C1 показали погіршення точності при використанні рекомендацій, що вказує на потребу в додатковому налаштуванні порогів і політики вибору вправ для складних слів. До обмежень дослідження також належать обмежений обсяг і нерівномірність даних, мала кількість користувачів, відсутність формального А/В-експерименту з незалежною контрольною групою та фокус переважно на кількісних, а не суб'єктивних метриках (мотивація, зручність, задоволеність).

Подальший розвиток роботи доцільно спрямувати на: розширення вибірки користувачів і тривалості спостереження; проведення контрольованих А/В-експериментів; окреме налаштування ВІНЛ та порогових правил для рівнів B1/C1; врахування додаткових цільових функцій (довгострокове збереження лексики, різноманітність вправ, залученість) і інтеграцію методів навчання з підкріпленням для оптимізації навчальних траєкторій на довгому горизонті. У сукупності отримані результати підтверджують, що запропонована рекомендаційна підсистема, що базується на використанні розробленого комбінованого методу, має реальний потенціал підвищення ефективності запам'ятовування іноземних слів і може бути основою для подальших наукових та прикладних розробок у сфері інтелектуальних навчальних систем.

Роботу виконано відповідно до вимог методичних вказівок щодо розробки та оформлення кваліфікаційних робіт [15], а також із дотриманням положень державних стандартів ДСТУ 8302:2015 [16] і ДСТУ 3008:2015 [17].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Anki. Anki Manual: Spaced Repetition Algorithm. – URL: <https://docs.ankiweb.net/> (дата звернення: 18.11.2025 р.). Електронний ресурс.
2. Memrise. Help Center: How does the spaced repetition system work? – URL: <https://memrise.helpshift.com> (дата звернення: 18.11.2025 р.). Електронний ресурс.
3. Quizlet. Learn mode and Test mode. – URL: <https://help.quizlet.com> (дата звернення: 18.11.2025 р.). Електронний ресурс.
4. Settles B., Meeder B. A Trainable Spaced Repetition Model for Language Learning // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. – Berlin, 2016. – P. 1848–1858.
5. Baker F.B. The Basics of Item Response Theory. – 2nd ed. – College Park, MD: ERIC Clearinghouse on Assessment and Evaluation, 2001. – 186 p.
6. Zhang Y. та ін. A Survey of Knowledge Tracing: Models, Variants, and Applications // arXiv preprint arXiv:2105.15106. – 2024. – URL: <https://arxiv.org/abs/2105.15106> (дата звернення: 18.11.2025 р.). Електронний ресурс.
7. Pavlik P.I. Jr., Anderson J.R. Using a Model to Compute the Optimal Schedule of Practice // Journal of Experimental Psychology: Applied. – 2008. – Vol. 14, No. 2. – P. 101–117.
8. Romero C., Ventura S., Pechenizkiy M., Baker R.S.J.d. (eds.). Handbook of Educational Data Mining. – Boca Raton: Chapman & Hall/CRC Press, 2010. – 536 p.
9. Corbett A.T., Anderson J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge // User Modeling and User-Adapted Interaction. – 1994. – Vol. 4, No. 4. – P. 253–278.
10. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. – 1997. – Vol. 9, No. 8. – P. 1735–1780.

11. Piech C., Bassen J., Huang J. та ін. Deep Knowledge Tracing // *Advances in Neural Information Processing Systems 28 (NIPS 2015)*. – 2015. – P. 505–513.
12. Rabiner L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition // *Proceedings of the IEEE*. – 1989. – Vol. 77, No. 2. – P. 257–286.
13. Мормуль В.В., Петров К.Е. Ансамблевий метод формування рекомендацій для запам'ятовування нових слів при вивченні іноземних мов // *Scientia: Collection of Scientific Papers with the Proceedings of the VII International Scientific and Theoretical Conference «Scientific Method: Reality and Future Trends of Researching»*. – Montreal, Canada: International Center of Scientific Research, 2025. – P. 390–393.
14. Ricci F., Rokach L., Shapira B. (eds.). *Recommender Systems Handbook*. – 2nd ed. – New York: Springer, 2015. – 1003 p.
15. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи другого (магістерського) рівня вищої освіти за освітньо-професійною програмою «Інформаційні управляючі системи та технології» спеціальності 122 Комп'ютерні науки / Упоряд.: К.Е. Петров, В.М. Левикін, С.Ф. Чалий, М.В. Євланов, В.І. Саєнко, Д.К. Міхнов, А.В. Міхнова, О.В. Чала. ХНУРЕ: Харків, 2024. 24 с.
16. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. К.: ДП «УкрНДНЦ», 2016. 20 с.
17. ДСТУ 3008:2015. Державний стандарт України. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. К.: ДП «УкрНДНЦ», 2016. 31 с.