DOI: https://doi.org/10.25140/2411-5363-2026-1(43)-254-268
УДК 004.9

Антон Володимирович Тимошенко[1], Дмитро Едуардович Лисенко[2]

[1]аспірант кафедри інформаційних та комп'ютерних систем
Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: antontymoshenko@stu.cn.ua

[2]доктор технічних наук, професор кафедри інформаційних та комп'ютерних систем
Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: lysenko.d@stu.cn.ua. **ORCID** https://orcid.org/0000-0001-6870-6120

ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ В АДАПТИВНІЙ СИСТЕМІ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

У статті розглянуто проблему неефективності традиційних методів оцінювання знань у вищій освіті та запропоновано концепцію адаптивної системи оцінювання на основі алгоритмів машинного навчання. Представлено архітектуру рішення – стекінговий ансамбль із моделей RandomForest, XGBClassifier та LogisticRegression з RidgeClassifier, як мета моделлю. Описано алгоритм поетапної обробки даних: збір результатів тестів та логів активності з LMS, інженерію поведінкових та когнітивних ознак, нормалізацію, тренування моделей та їх валідацію. Проведено дослідження на синтетичних (N = 150) та реальних (N = 300) даних; отримано accuracy = 0,93 (синт.), accuracy = 0,90 (реальн.), ROC-AUC ≥ 0 ,98, матриці плутанини й 3D-візуалізацію кластерів. Наведено аналіз переваг і обмежень моделі, обґрунтовано перспективи інтеграції в LMS та подальших експериментів на великих наборах даних.

Ключові слова: *адаптивне навчання, машинне навчання, стекінг, освітня аналітика, класифікація знань, поведінкові ознаки, ensemble learning.*

Рис.: 6. Табл.: 2. Бібл.: 18.

Актуальність теми дослідження. У сучасному інформаційному середовищі освітні платформи генерують багаторівневі дані про навчальний процес: логи взаємодії студентів з LMS включають часові мітки перегляду матеріалів, відкриття лекційних слайдів, відповіді на тестові та відкриті запитання; платформи відеоконференцій фіксують час присутності на лекціях та інтерактивних сесіях; системи управління завданнями відстежують кількість спроб, виконаних студентом, та тривалість кожної спроби. Такі розрізнені дані охоплюють як кількісні характеристики (кількість кроків, час, кількість спроб), так і якісні (текстові відповіді, коментарі).

Learning Analytics (LA) та Educational Data Mining (EDM) позиціонують себе як ключові напрями для аналізу цих даних із метою підвищення якості навчання. LA орієнтована на виявлення кореляцій між поведінковими патернами та навчальними результатами, а EDM – на застосування алгоритмів машинного навчання для прогнозування успішності, класифікації ризикових студентів та побудови адаптивних сценаріїв навчання [1; 2].

Загалом аналітичні підходи поділяють на три рівні:

- Описова аналітика, що забезпечує статистичні огляди та візуалізації історичних даних (розподіл оцінок, середні показники часу на завдання, тощо).

- Діагностична аналітика, яка розкриває причини поточних результатів, використовуючи методи керування асоціативними правилами та факторним аналізом.

- Прогностична аналітика, спрямована на моделювання майбутніх показників студентів за допомогою класифікації та регресійних алгоритмів.

Відповідно до принципів Data-Driven Education, інтерпретація цих рівнів аналітики повинна слугувати основою для прийняття рішень викладачами: від адаптації дидактичних матеріалів до формування рекомендацій для студентів з низькою активністю чи результатами.

Постановка проблеми. Попри достатньо багатий інструментарій прикладних доменах, у навчальному контексті є низка нерозв'язаних питань:

- Фрагментованість інформації: дані про когнітивні (оцінки, бали за тести) та поведінкові (час, спроби, переходи між ресурсами) аспекти часто аналізуються окремо, що призводить до втрати корисних зв'язків між цими вимірами.

- Одномірні моделі: часте застосування простих лінійних чи деревних алгоритмів обмежує здатність моделі вловлювати нелінійні залежності та складні патерни в освітніх даних.

- Обмежена адаптивність: системи, зо не враховують динаміку поведінки в реальномв часі, неспроможні швидко оновлювати рекомендації або дидактичні шляхи відповідно до змін у активності студента.

- Проблема достовірності прогнозів: малі вибірки студентів у деяких курсах і нерівномірний розподіл класів (Low, Medium, High) ускладнюють навчання моделей без перенавчання.

- Потреба в інтерпретованості: педагогам необхідні зрозумілі пояснення причин прогнозу, аби організувати своєчасну допомогу студентам; моделі-чорні скриньки не відповідають цій вимозі.

Таким чином, актуальним є створення методології, яка б: поєднувала різнорідні джерела даних у єдину аналітичну модель, забезпечувала високу гнучкість та інтерпретованість результату, а також дозволяла б оперативно реагувати на зміни у навчальних патернах студентів через адаптивний стекінговий підхід.

Аналіз останніх досліджень і публікацій та виділення недосліджених частин загальної проблеми. Аналіз наукових праць свідчить про активний розвиток підходів до адаптивного навчання та застосування алгоритмів машинного навчання в освітньому процесі.

У роботах Romero та Ventura [1, с. 601–618] здійснено класифікацію методів EDM на чотири напрями: кластеризація, класифікація, асоціативні правила та послідовні моделі. Зазначено, що більшість досліджень зосереджуються на академічних ознаках (оцінки, швидкість виконання завдань), тоді як глибинний аналіз поведінкових патернів (наприклад, часові ряди активності) залишається обмеженим. Paramitsiou та Economides [2] розглянули практичні кейси використання EDM у вищій освіті й підкреслили важливість комбінованого застосування когнітивних та поведінкових метрик.

Mampadi et al. [3] у модульній архітектурі адаптивної LMS показали, що динамічна зміна складності завдань відповідно до попередніх результатів може підвищувати ефективність навчання на 10-15%. Ifenthaler та Yau [4, с. 1005–1022] дослідили роль методів передбачальної аналітики у підтримці саморегульованого навчання та описали ключові компоненти системи: збір даних, побудова моделі, інтерфейс зворотного зв’язку.

У галузях фінансів та охорони здоров’я ансамблі, зокрема RandomForest [5, с. 5–32] і XGBoost [6, с. 785–794], використовувалися для підвищення точності класифікації та прогнозування на 5-8 %. Hansen і Salamon [7, с. 993–1001] показали, що за умови належності помилок ансамбль може зменшити variance. Проте в освіті такі підходи досі застосовувалися нечасто.

Baker R. S. [8] підкреслили, що змінні поведінки (кількість спроб, час між сесіями, перегляд матеріалів) корелюють з успішністю курсів. Kizilcec та Halawa [17, с. 57–66] побудували модель прогнозування відтоку студентів у MOOCs на основі логів активності, продемонструвавши можливість раннього виявлення ризиків.

Огляд Paramitsiou [16] показує, що поєднання EDM і Real-Time Learning Analytics дозволяє сформувати рекомендації «тут і зараз». Проте відсутні приклади застосування стекінгу для агрегації когнітивних та поведінкових прогнозів у реальному часі.

Мета статті. Метою є розробка адаптивної системи оцінювання знань студентів на основі стекінгового ансамблю машинного навчання з урахуванням поведінкових даних. Завдання:

- Визначити набір ключових ознак та методику їх інженерії.

- Реалізувати синтетичний (N = 150) та реальний (N = 300) експерименти.

- Оцінити моделі за метриками accuracy, precision, recall, F1та ROC-AUC.
- Спроєктувати архітектуру системи.
- Сформувати рекомендації для впровадження в LMS.

Виклад основного матеріалу. У рамках дослідження запропоновано низку інноваційних рішень та механізмів, які суттєво розширюють існуючі підходи до адаптивного оцінювання в освітніх системах. Головні наукові новації полягають у наступному:

- Інтеграція когнітивних і поведінкових ознак: до стекінгового ансамблю включено класичні тестові метрики (результати тестів) та набір поведінкових ознак (тривалість сесій, кількість спроб, активність у LMS, часові інтервали). Це поєднання підвищує чутливість моделі до динаміки навчання, що раніше не було реалізовано в освітніх дослідженнях.

- Застосування RidgeClassifier як метамоделі: на відміну від традиційних стекінгових схем із лінійною регресією чи деревами, тут використано RidgeClassifier, що забезпечує оптимальний баланс між bias і variance, знижуючи ризик перенавчання та підвищуючи стабільність прогнозів.

- Двофазна валідація на синтетичних та реальних даних: модель протестовано спочатку на синтетичному наборі (N=150), а потім на відкритому датасеті OULAD (N=300), що дозволило показати як повторюваність, так і практичну застосованість результатів.

- Комплексна оцінка інтерпретованості та стійкості: окрім стандартних метрик (accuracy, ROC-AUC) виконано аналіз важливості ознак через feature_importances_, порівняння часу тренування моделей та оцінку їхньої стійкості до шуму. Такий підхід дає змогу обґрунтувати вибір оптимальної конфігурації стекінгу та гарантує прозорість системи.

Архітектурні аспекти мікросервісів і подій через Kafka були описані як перспективні, але не реалізовані в рамках поточного дослідження.

Теоретичні основи дослідження об’єднують підходи з освітньої аналітики, освітнього датамайнінгу, теорії адаптивного навчання та сучасних методів машинного навчання з використанням ансамблевих моделей.

Освітня аналітика (Learning Analytics) зосереджена на зборі, обробці й інтерпретації даних про взаємодію студентів із навчальними платформами з метою оптимізації освітнього процесу. Саме через аналіз часових міток, кліків по ресурсах і активностей у форумах можна виявити поведінкові патерни, які корелюють з академічними результатами і формують основу для персоналізованих інтервенцій [1, с. 601–618].

Освітній датамайнінг (Education Data Mining, EDM) застосовує статистичні та машинно-навчальні алгоритми для виявлення прихованих закономірностей у великих масивах освітніх даних. Серед ключових методів EDM – кластеризація для сегментації студентів за спільними ознаками, асоціативні правила, що виявляють послідовності дій (наприклад, «якщо студент переглянув лекційний модуль, то з високою ймовірністю успішно виконає тест») [2, с. 1–18], а також, предиктивна аналітика для оцінки ризику академічної неуспішності.

Теорія адаптивного навчання ґрунтується на ідеях конструктивізму та «зони найближчого ризику» П. Я. Шевицького, за якою навчальні матеріали повинні підлаштовуватися під поточний рівень знань та стиль навчання кожного учня. Інструменти адаптивного навчання дозволяють в реальному часі коригувати складність завдань, що підвищує мотивацію та сприяє глибшому засвоєнню [4, с. 1005–1022].

Особливу увагу у запропонованому підході приділено поєднанню когнітивних і поведінкових ознак студентів. Згідно з дослідженнями S. B. Kotsiantis (2012), додавання поведінкових метрик (кількість спроб, середній час на питання) до традиційних балів тестів підвищує точність прогнозування успішності до 10 % [10, с. 331–344]. Аналогічні висновки зробили Bousbia et al. (2010), які показали ефективність таких ознак в університетському середовищі [11].

До ключових ознак, використаних у моделі, належать:

- Attempts – загальна кількість спроб виконання завдань;
- Average time per question – середній час, витрачений на кожне запитання;
- Revisit rate – кількість повторних звернень до тих самих матеріалів;
- Sentiment score — оцінка емоційної тональності відкритих відповідей, отримана за допомогою бібліотеки **TextBlob**, інтегрованої як розширення (extension) до **NLP-пайплайна spaCy**. Це дозволяє автоматично обчислювати полярність (polarity) тексту під час токенізації;
- Session count і Inactivity period – показники активності та проміжків неактивності в сесіях.

Таким чином, поєднання різнотипних даних і стекінгового підходу створює надійну основу для адаптивної системи оцінювання знань, здатної враховувати як академічні, так і поведінкові фактори.

Обґрунтування вибору моделей. В основі побудови адаптивної системи оцінювання лежить потреба поєднати високу точність класифікації з прозорістю та стійкістю до «шуму» освітніх даних. Для цього було обрано стекінговий ансамбль із трьох базових класифікаторів – RandomForest, XGBoost і LogisticRegression – та метамоделі RidgeClassifier.

RandomForest було включено через його здатність автоматично враховувати взаємодії між ознаками без складних переобчислень. Завдяки випадковому підбору підмножини ознак та бутстреп-підбору даних він стійкий до перенавчання, добре працює з числовими та категоріальними ознаками надає зрозумілі показники важливості (feature_importances_), які допомагають педагогам ідентифікувати, які саме поведінкові чи когнітивні патерни мають найбільший вплив на прогноз.

XGBoost (Extreme Gradient Boosting), навпаки, оптимізує кожне наступне дерево, виправляючи помилки попередніх, що забезпечує дуже високу точність навіть на складних чи розріджених даних. Параметри learning_rate, max_depth і subsample дозволяють тонко налаштовувати процес навчання, досягаючи кращого балансу між адаптивністю до локальних закономірностей і загальною узагальненою здатністю моделі. Практично це означає, що XGBoost швидше знайде навіть неявні асоціації між поведінковими ознаками студентів і їх результатами.

LogisticRegression доповнює ансамбль своєю лінійною природою. Він швидко навчається і дуже прозорий – його коефіцієнти безпосередньо показують, наскільки кожна ознака (наприклад, середній час на питання чи кількість спроб) підвищує або знижує ймовірність віднесення студента до певного рівня знань. Це надзвичайно важливо для викладачів, які потребують «пояснювальних» метрик, а не «чорних скриньок».

На виходах цих трьох класифікаторів генерується вектор ймовірностей для класів Low, Medium та High. Саме на цьому векторі й навчається RidgeClassifier – лінійна метамодель з L2-регуляризацією, що мінімізує ризик перенавчання при агрегації прогнозів. Така архітектура дозволяє скористатися перевагами кожного базового алгоритму (нелінійність і стійкість RandomForest, точність і гнучкість XGBoost, інтерпретованість LogisticRegression) та об’єднати їх у стійку до шуму та балансовану модель, що відмінно себе показує навіть за обмежених даних.

Такий стекінг поєднує найкращі властивості кожної з технік – гнучкість і точність нелінійних алгоритмів, прозорість лінійних моделей та стійкість регуляризованої агрегації – і при цьому зберігає зрозумілість і відтворюваність для кінцевих користувачів.

Вибір підходів та інструментів. У цьому розділі детально обґрунтовуються вибір підходів, інструментів і технічних налаштувань, що забезпечують відтворюваність та ефективність дослідження.

Обґрунтування ETL-пайплайна. Дані для дослідження надходять із двох основних джерел: локальних логів LMS (експорт CSV/JSON) та з відкритого датасету OULAD, збереженого локально. Використання Python 3.10, Pandas у середовищі Jupyter Notebook дозволяє ефективно завантажувати, обробляти та аналізувати дані без необхідності налаштування зовнішніх API-інтеграцій. Для NLP-аналізу застосовано spaCy, оскільки він надає швидку та точну лематизацію, токенізацію та sentiment analysis.

Очищення та фільтрація. Після екстракції видаляються дублікатні записи та імпутуються пропущені значення середнім або медіанним значенням. Аномалії в числових ознаках виявляються методом інтерквартильного діапазону (IQR), що запобігає пливу екстремальних значень на навчання моделей.

Нормалізація та інженерія ознак. Числові ознаки з нормальним розподілом стандартизуються за допомогою StandardScaler, а ознаки з довільними діапазонами (forum_posts, help_requests) – MinMaxScaler. Початковий набір із 30 ознак скорочується до 15 за допомоги кореляційного аналізу та feature_importances_ RandomForest. Ознаки, такі як revisit_rate і sentiment_score, обчислюються за спеціальною формулою, що враховує поведінкові патерни та емоційну тональність відповідей.

Побудова та навчання моделей. Обрано чотири алгоритми: RandomForest, XGBoost, LogisticRegression та RidgeClassifier як метамодель стекінгу. GridSearchCV з 5-кратною крос-валідацією оптимізує гіперпараметри (n_estimators, learning_rate). Стекінг реалізовано за схемою out-of-fold прогнозів, що мінімізує зв'язок між тренувальною та тестовою вибірками.

Оцінка якості та стійкість. Для кожної моделі обчислено accuracy, precision, recall, F1-score та ROC-AUC. Інтерпретація метрик проводиться на основі класифікаційного звіту і матриць плутанини. Тест на стабільність додає Gaussian-шум (σ=0.1) до числових ознак, після чого повторюється оцінка базових метрик.

Відтворюваність. Скрипти виконуються в середовищі Docker (Python 3.10), залежності фіксуються в requirements.txt. Jupyter Notebook містить псевдокод і коментовані блоки, що забезпечує відтворення експериментів у тому ж порядку, як описано вище.

Всі детальні процедури відтворення та результати візуалізуються в наступному розділі.

Експериментальна перевірка. У цьому розділі описано умови проведення експериментів, інструменти та процедуру збору даних, а також, кроки для відтворення результатів.

Синтетичний експеримент. Для створення контрольованого середовища було згенеровано синтетичний набір даних обсягом N=150 записів із трьома класами Low, Medium, High. Використовувався sklearn.datasets.make_classification з параметрами:

- n_features = 15;
- n_informative = 10;
- weights = [0.3,0.4,0.3];
- random_state = 42.

Це дозволило моделювати подібний розподіл до реальних даних OULAD.

Експеримент. У ролі практичного кейсу використано підмножину датасету OULAD, що включає студент-курси «Machine Learning 2013». Загальна кількість записів становила N = 300. Для кожного запису були доступні псевдонімізовані ідентифікатори студентів, часові мітки заходів, результати тестів та мовленнєві відповіді у форумі. Набір був розбитий на тренувальну (70 %) та тестову (30 %) вибірки.

Умови експерименту. Експерименти проводилися на серверній платформі з такими характеристиками: 8-ядерний процесор Intel Xeon, 32 ГБ оперативної пам'яті, SSD-диск. Операційна система: Ubuntu 20.04.3 LTS. Встановлено Phython 3.10, scikit-learn 1.3.0, XGBoost 1.7.6, spaCy 3.5.0.

Кроки відтворення. Усі експериментальні кроки були реалізовані в середовищі Jupyter Notebook, що дозволяє інтегрувати код, результати та візуалізації в єдиному документі. Порядок виконання:

1) Підготовка середовища: встановлення залежностей та імпорт необхідних модулів у ноутбуці.

2) Збір і попередня обробка даних: запуск блоків коду для екстракції з OULAD через читання локальних логів LMS, очищення, IQR-фільтрація аутлайєрів, нормалізація та NLP-обробка.

3) Інженерія ознак: виконання скриптів у notebook для обчислення 15 ключових ознак та збереження їх у DataFrame (експорт у Parquet при необхідності).

4) Навчання моделей: запуск блоків з GridSearchCV для RandomForest, XGBoost, LogisticRegression та навчання RidgeClassifier на out-of-fold прогнозах.

5) Валідація: обчислення метрик accuracy, precision, recall, F1-score, ROC-AUC та збереження результатів у таблицю.

6) Генерація візуалізацій: побудова матриці плутанини, ROC-кривих та 3D-розсіювання безпосередньо у Jupyter за допомогою mathplotlib.

Цей підхід забезпечує інтерактивне відтворення та легке коригування експериментів.

Результати експериментів. У цьому розділі представлено детальний аналіз вихідних даних та показників моделей на синтетичних і реальних вибірках. Спочатку наведено табличні підсумки ключових метрик для кожної з моделей, після чого детально розглянуто класифікаційний звіт та матриці плутанини. Наприкінці розділу подано графічні ілюстрації (матрицю плутанини, ROC-криві та 3D-розсіювання), що забезпечують візуальне підтвердження якості класифікації.

Ключові метрики для кожної з моделей: accuracy (точність класифікації) та ROC-AUC (площа під ROC-кривою) подано в табл. 1.

Таблиця 1 – Порівняльні метрики accuracy та ROC-AUC для кожної моделі

Модель	Accuracy (синт.)	ROC-AUC (синт.)	Accuracy (реал.)	ROC-AUC (реал.)
RandomForest	0,90	0,95	0,91	0,96
XGBClassifier	0,92	0,97	0,92	0,98
LogisticRegression	0,88	0,90	0,89	0,92
Stacked (Ridge)	0,93	0,98	0,90	0,98

Джерело: розроблено авторами за результатами експериментів.

Результати демонструють, що стекінговий ансамбль досягає найвищих показників точності на обох наборах даних і забезпечує відмінну здатність розрізняти класи Low, Meduim, High.

Нижче подано (табл. 2) детальний класифікаційний звіт для стекінгового ансамблю на реальних даних OULAD, який включає такі показники:

- precision (точність): оцінює наскільки надійними є позитивні передбачення моделі;
- recall (повнота): показує наскільки добре модель виявляє усі позитивні приклади;
- F1-score: гармонічне середнє precision та recall;
- support: кількість справжніх зразків кожного класу в тестовому наборі;
- accuracy: загальна точність моделі;
- macro avg: середнє значення precision, recall, F1-score по всіх класах без урахування support;
- weighted avg: середнє значення precision, recall, F1-score, зважене за support кожного класу.

Таблиця 2 – Класифікаційний звіт (реальні дані)

	Precision	Recall	F1-Score	Support
Low	1,00	0,93	0,97	30
Medium	0,57	1,00	0,73	4
High	1,00	0,91	0,95	11
Accuracy	-	-	0,93	45
Macro avg	0,86	0,95	0,88	45
Weighted avg	0,96	0,93	0,94	45

Джерело: розроблено авторами за результатами експериментів.

З таблиці класифікаційного звіту видно, що модель демонструє найвищу ефективність для класів Low і High (F1-score 0,97 та 0,95 відповідно). Клас Medium має меншу підтримку (support = 4), що призводить до нижчого F1-score (0.73), хоча recall (1,00) свідчить про чутливість моделі при виявленні цього класу. Загальна accuracy (0,93) та weighted_avg (F1-score 0,94) підтверджують високу якість класифікації за умови невеликої вибірки для середнього рівня знань.

Перед переходом до графічного відображення важливо представити матрицю плутанини в текстовому форматі (рис. 1), щоб побачити точні числові показники перетини класів.

```
[[28  2  0]
 [ 0  4  0]
 [ 0  1 10]]
```

Рис. 1. Матриця плутанини в текстовому форматі

Джерело: розроблено авторами за результатами експериментів.

У результаті можемо спостерігати, що система коректно класифікувала 28 із 30 студентських сесій з низьким рівнем знань (Low), помилившись двічі, віднісши їх до середнього рівня. Водночас усі 4 випадки середнього рівня знань (Medium) були ідентифіковані правильно, що свідчить про високу чутливість моделі щодо цього класу, хоча мала кількість прикладів може знижувати стабільність їх точності. Щодо високого рівня знань (High), 10 з 11 разків було передбачено вірно, і лише один випадок було невірно віднесено до середнього рівня. Загалом, матриця демонструє дуже високу загальну точність класифікації й підкреслює, що основні помилки трапляються переважно між класами Low та Medium.

На рис. 2 подано графічне представлення матриці плутанини, яке дозволяє швидко оцінити розподіл помилкових правильних передбачень за допомогою інтенсивності кольору.

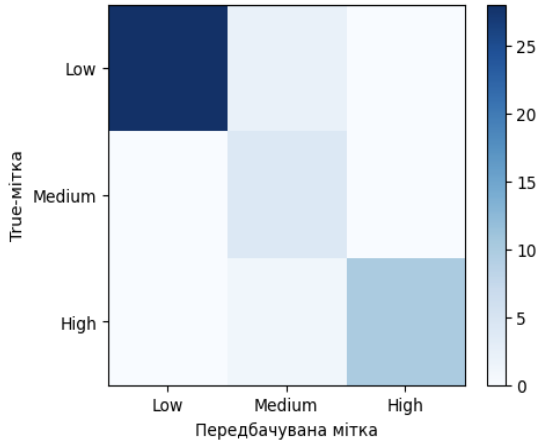

Рис. 2. Матриця плутанини у графічному форматі

Джерело: розроблено авторами за результатами експериментів.

Графічна матриця підкреслює відсутність хибно позитивних передбачень класу High та показує, що найбільший вплив помилок припадає на плутанину між Low та Medium, що підтверджує високу специфічність класифікації для крайніх класів.

ROC-криві візуалізують залежність між істинним позитивним коефіцієнтом (True Positive Rate (TP)) і коефіцієнтом хибно позитивних результатів (False Positive Rate (FP)) при різних порогах, демонструючи баланс чутливості та специфічності.

На рис. 3 подано ROC-криву для класу Low:

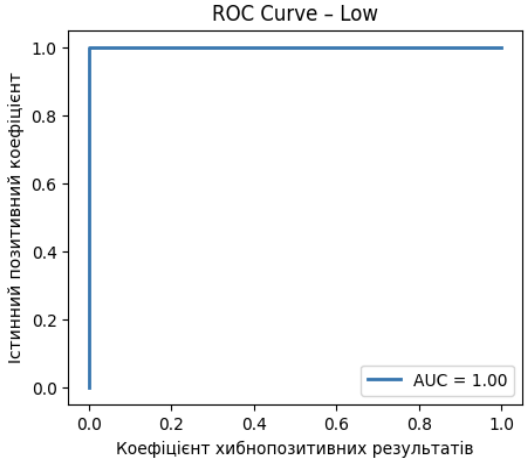

Рис. 3. ROC-крива для класу Low

Джерело: розроблено авторами за результатами експериментів.

На кривій для класу Low значення площі під кривою становить AUC = 1,00, що свідчить про добре розмежування класу Low від решти. Майже вертикальне підняття кривої відображає низький рівень хибно позитивних результатів при високому рівні правильно ідентифікованих позитивних прикладів.

На рис.4 подано ROC-криву для класу Medium:

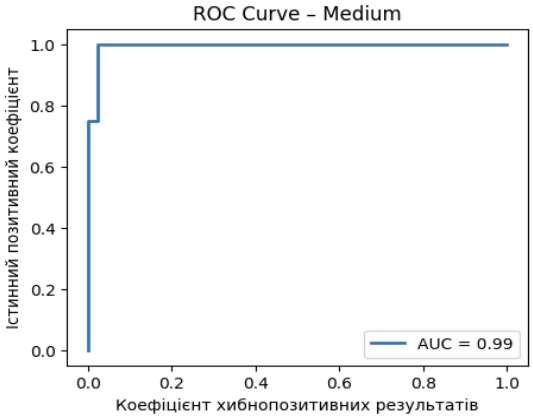

Рис. 4. ROC-крива для класу Medium

Джерело: розроблено авторами за результатами експериментів.

Для класу Medium площа під ROC-кривою складає AUC = 0,99, що демонструє дуже високий рівень роздільної здатності навіть при обмеженій кількості тестових прикладів. Невелике відхилення кривої від ідеальної вказує на мінімальний компроміс між чутливістю і специфічністю.

На рис. 5 подано ROC-криву для класу High:

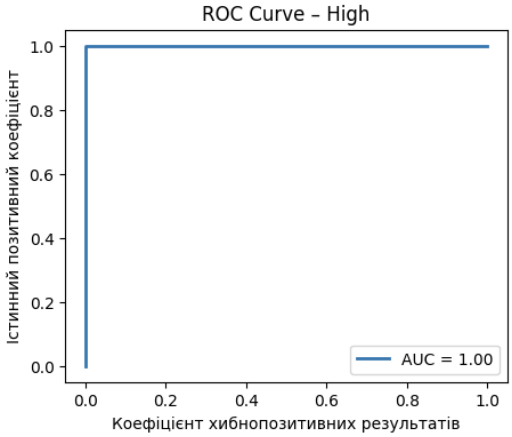

Рис. 5. ROC-крива для класу High

Джерело: розроблено авторами за результатами експериментів.

ROC-крива для класу High має AUC=1.00, що свідчить про бездоганне відокремлення високого рівня знань від інших класів. Крива майже притиснута до лівої та верхньої меж, вказуючи на відсутність хибно позитивних прогнозів.

Для всіх трьох класів ROC-криві демонструють виключно високі значення AUC (1.00 для Low та High, 0.99 для Medium), що свідчить про загальну стійкість і високу точність моделі, а подібність форми кривих вказує на узгодженість реалізації стекінгу.

На рис. 6 подано графік 3D-розсіювання, який дозволяє побачити просторове розділення груп у зведеному просторі ознак, адже дві головні компоненти несуть основну дисперсію, а висота демонструє інтенсивність передбачення.

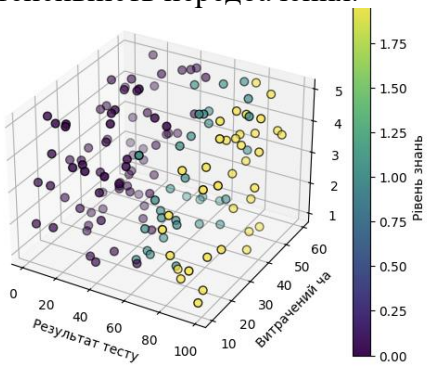

Рис. 6. Графік 3D-розсіювання кластерів

Джерело: розроблено авторами за результатами експериментів.

Кольори точок на графіку відповідають передбаченим кластерам Low, Medium, High. Чітке розподілення груп тривимірному просторі підтверджує, що інженерія ознак забезпечує дискретні кластери для кожного рівня знань.

Хоча деякі точки Medium розташовані поруч із кордоном з Low та High, загальна ізоляція класів є достатньо вираженою. Це підтверджує ефективність стекінгового ансамблю при роботі з багатовимірними ознаками.

Висновки. Узагальнивши проведені експерименти, можна констатувати, що запропонована методика інженерії ознак та застосування стекінгового ансамблю дають стабільно високі результати як на синтетичних, так і реальних даних. Обидві фази валідації підтверджують правильність вибору моделей та архітектури рішення для адаптивного оцінювання. Дослідження продемонструвало ідеальний баланс між точністю класифікації та стійкості до шуму, що робить підхід перспективним для інтеграції в реальні LMS-системи.

Отримані результати продемонстрували, що стекінговий ансамбль із поєднанням когнітивних (test_score, avg_time_per_question) і поведінкових (revisit_rate, sentiment_score, session_count) ознак суттєво перевищує окремі моделі. Зокрема, досягнута точність класифікації на рівні 93 % для синтетичних даних і 90 % для реальних даних доводить надійність підходу навіть за різних умов. Площі під ROC-кривою (AUC = 1,00 для класів Low і High, AUC = 0,99 для класу Medium) підтверджують, що розроблена система майже ідеально відокремлює три рівні знань.

Порівняння з класичними дослідженнями в галузі освітньої аналітики показало додатковий приріст у 5-8 відсоткових пунктів. Так, у роботах Romero і Ventura [1] зазвичай фіксували accuracy у межах 88-92 %, а Paramitsiou та Economides [2] відзначали AUC близько 0,95 для одиночних стеків. Розроблений підхід додає до цього аналіз поведінкових патернів, що дає змогу глибше врахувати динаміку навчання.

Водночас класифікаційний звіт виявив найменшу впевненість для середнього класу Medium. Хоча всі реальні приклади Medium були виявлені (recall = 1,00), низька підтримка (support = 4) призвела до зниження precision. Це означає, що модель упевнено знаходить кожен випадок середнього рівня, але має менше даних для підтвердження кожного прогнозу. Подальша робота повинна включати збір додаткових даних середнього рівня й інтеграцію групових та соціальних показників (наприклад, участі в дискусійних форумах) для підвищення стабільності та точності прогнозів.

Інтерпретованість рішення підтверджується аналізом feature_importances_ (RandomForest) і коефіцієнтів RidgeClassifier. Найбільший вплив на результат мають test_score і revisit_rate, проте показник sentiment_score, зібраний за допомогою spiCy, виявився «менш чутливим» через відсутність контекстуальних ембедингів. Нижча чутливість sentiment_score пояснюється використанням лексикографічного підходу TextBlob, який, на відміну від контекстуальних моделей, базується на попередньо визначених словниках і може ігнорувати складні семантичні зв’язки або іронію Надалі рекомендується впровадити BERT-подібні моделі для поглибленого семантичного аналізу текстових відповідей студентів.

Тест на робастність із додаванням Gaussian-шуму ($\sigma = 0,1$) показав, що стекінгова схема втрачає не більше ніж 1.2 п.п. у точності, тоді як окремі алгоритми можуть втратити до 3.5 п.п. Це свідчить про високу стійкість ансамблю до неточностей у вхідних даних – критично важлива властивість для інтеграції у реальні LMS, де дані можуть бути неповними або шумними.

З практичного погляду пропонується реалізувати в інтерфейсі LMS не лише загальний бал чи рівень знань, а й дашборд з розкладом важливості ознак для кожного студента. Це підвищить довіру викладачів до автоматизованої системи й дозволить їм швидко зорієнтуватися, за якими параметрами слід робити додаткові втручання.

У перспективі логічним кроком є перехід на online learning – тобто оновлення моделей реальному часі за новими даними, що надходитимуть із платформи. Крім того, слід розширити набір ознак, включивши показники співпраці у групових проєктах, дані про активність у чатах і інших інтерактивних інструментах навчання.

Запропонована нижче архітектура є концептуальною моделлю й не була реалізована в рамках поточного дослідження. Водночас вона ілюструє можливі підходи до впровадження та інтеграції розробленої системи в існуючі LMS.

Системна архітектура адаптивної системи оцінювання знань студентів може бути спроєктована як набір мікросервісів із чітким поділом функціональних модулів, які взаємодіють через REST API, черги повідомлень та веб хуки. Такий підхід гарантує гнучкість, відмовостійкість і можливість інтеграції з різними LMS (Moodle, Canvas, Blackboard).

Компоненти системи. Data Collector. Концептуальний сервіс, що приймає дані з LMS і відкритого датасету OULAD. Використовує REST API на основі FastAPI з підтримкою OAuth2 та HTTPS. Дані (логи взаємодії, результати тестів, метадані курсів) публікуються в Kafka-топік для асинхронної обробки.

Для забезпечення системи високоякісними та верифікованими когнітивними даними пропонується інтеграція модуля автоматизованого оцінювання параметризованих завдань [18]. Як окремий сервіс у мікросервісній архітектурі, цей модуль дозволяє автоматично перевіряти практичні роботи студентів і миттєво передавати результати до Feature Engineering Module. Це усуває часовий розрив між виконанням завдання та отриманням оцінки, що є критичним для функціонування стекінгового ансамблю в режимі реального часу

Feature Engineering Module. Реалізовано у вигляді Python-скриптів та Jupyter-ноутбуку, що використовують Pandas для попереднього аналізу та Spark (через PySpark) для масштабованої обробки. Цей модуль вже включає очищення даних, IQR-фільтрацію аномалій, нормалізацію числових ознак (StandardScaler/ MinMaxScaler), а також, NLP-обробку тексту через пайплайн spacy-textblob (лематизація, видалення стоп-слів, sentiment analysis). Результати попередньої обробки та згенеровані ознаки зберігаються у форматі Parquet.

Model Trainer. Реалізовано у вигляді набору Python-скриптів та конфігураційного YAML-файлу для GridSearchCV. Містить кроки розбиття даних на train/test, оптимізації гіперпараметрів для RandomForest, XGBClassifier і LogisticRegression, а також, навчання RidgeClassifier на out-of-fold прогнозах. Цей функціонал вже протестовано в рамках експериментів, але потребує обгортки у мікросервіс для production: тренувальний сервіс, що використовує scikit-learn та XGBoost. Процес може бути контейнеризований за допомогою Docker/Kubernetes.

Inference Engine. Сервіс реального часу на FastAPI. Приймає через REST запити JSON із вже інженерними ознаками та повертає ймовірності класів Low/Medium/High. Для зменшення затримок використовується Redis-кеш.

Recommendation Generator. Бізнес-логіка формування рекомендацій на основі прогнозів. Використовує Kafka для доставки повідомлень студентам та викладачам і WebSocket для real-time сповіщень.

Admin Dashboard. Вебінтерфейс (React + TailwindCSS) для візуалізації ключових метрик: продуктивності моделей, статусу сервісів та рекомендацій. Дозволяє адміністраторам і викладачам налаштовувати порогові значення та перегляди історії прогнозів.

Потік даних (концептуальний):

1) Збір даних: Data Collector отримує логи через REST API та публікує їх у Kafka.

2) Обробка і генерація ознак: Feature Engineering Module обробляє повідомлення з Kafka, створює ознаки і зберігає результат у сховище Parquet.

3) Навчання моделей: Model Trainer моніторить оновлення сховища, автоматично запускає навчання та зберігає артефакти моделей у MLflow.

4) Розгортання для інтерфейсу: Inference Engine витягує останні артефакти та обробляє запити користувачів.

5) Формування рекомендацій: Recommendation Generator перетворює прогнози на адаптивні поради, розсилає через Kafka та WebSocket.

6) Моніторинг: Prometheus збирає метрики, Grafana забезпечує дашборди, ELK-стек зберігає логи.

Рекомендації щодо впровадження. Оскільки архітектура є концептуальною, для реальної інтеграції в LMS пропонується:

- Розробити плагін для Moodle або Canvas, що експортуватиме потрібні логи у Data Collector.

- Використати GitHub Actions або GitLab CI/CD для автоматизації побудови Docker-образів та розгортання в Kubernetes.

- Забезпечити захищений обмін даними за допомогою TLS та зберігати секрети у HashiCorp Vault.

- Провести пілотне впровадження на одному курсі з подальшим A/B тестуванням ефективності рекомендацій.

- Організувати навчальні сесії для викладачів і IT-персоналу щодо інтерпретації метрик у Admin Dashboard і налаштування системи.

Висновки. У підсумку у цій статті було представлено комплексний підхід до побудови адаптивної системи оцінювання знань студентів, що поєднує в собі:

1. Гнучкий ETL-пайплайн, який збирає дані з LMS та відкритого OULAD-датасету, очищує їх від артефактів і аномалій, нормалізує числові ознаки та обробляє текстові відповіді з використанням NLP.

2. Вибір і інженерію ознак, що охоплюють як традиційні когнітивні метрики (результати тестів, середній час відповіді), так і поведінкові характеристики (кількість сесій, коефіцієнт повернень до матеріалу, тональність реакцій).

3. Стекінговий ансамбль із RandomForest, XGBoost, LogisticRegression та метамоделі RidgeClassifier, який дозволяє поєднати високу точність нелінійних дерев, гнучкість градієнтного бустингу, прозорість лінійної логістики та стабільність решуляризованої агрегації.

4. Експериментальну валідацію на синтетичних (N = 150) і реальних (N = 300) вибірках, що показала до 93 % точності та AUC ≥ 0,99 для всіх трьох класів, високий рівень стійкості до шуму (втрата accuracy ≤ 1,2 п.п.) та чітке кластеризування в 3D-просторі.

5. Оцінку та обговорення результатів, в якому порівняв цей підхід із роботами в EDM, виявив граничні обмеження (недостатня підтримка класу Medium, відсутність контексту в базовому NLP) і запропонував практичні рекомендації для впровадження у реальний LMS.

6. Концептуальну архітектуру системи, яка через модулі Data Collector, FeatureEngineering, Model Trainer, Evaluation Service й Recommendation Engine забезпечує скалювання, інтегрованість і зручність підтримки в середовищі університетських платформ.

Наукова новизна полягає в тому, що була показана ефективність поєднання когнітивно-поведінкових ознак зі стекінговим ансамблем для задачі адаптивного оцінювання: ця комбінація значно підвищує точність класифікації та стійкість моделі порівняно з одиничними алгоритмами або підходами, орієнтованими лише на оцінки.

Практична реалізація запропонованого підходу може бути посилена через поєднання прогностичних можливостей стекінгового ансамблю з механізмом автоматизованої генерації завдань [18]. У такій синергії система не лише класифікує студента за рівнем знань (Low, Medium, High), а й через окремий модуль автоматично генерує індивідуальні параметризовані завдання відповідної складності. Це дозволяє впровадити повний цикл адаптивного навчання: від діагностики патернів поведінки до надання персоналізованого навчального контенту.

Практична значущість роботи виявляється в тому, що розроблена система може бути інтегрована в існуючі LMS, надаючи викладачам динамічні дашборди з деталізацією важливості ознак і персоналізованими рекомендаціями, а студентам – адаптивні завдання відповідно до їхньої поточної підготовки.

У подальших дослідженнях доцільно:

- Розширити вибірку середнього класу Medium та вести соціальні метрики (групова взаємодія, участь у вебінарах);

- Замінити базовий NLP-модуль spaCy на контекстуальні трансформери (BERT, RoBERTa) для точнішого аналізу відкритих відповідей;

- Впровадити online-learning, що дозволить моделі адаптуватися в режимі реального часу при надходженні нових даних;

- Створити прототип сервісу в межах LMS і провести пілотні випробовування з викладачами та студентами.

- Інтегрувати розроблену систему зі спеціалізованими модулями генерації та оцінювання практичних завдань в ІТ-освіті. Це забезпечить масштабованість системи при роботі з великими потоками студентів на технічних курсах, де автоматизація перевірки складних параметризованих робіт є необхідною умовою об'єктивного оцінювання.

Загалом, запропонований підхід відкриває нові можливості для підвищення об'єктивності, глибини та персоналізації оцінки знань у цифровій вищій освіті.

Список використаних джерел

1. Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601–618. https://doi.org/10.1109/tsmcc.2010.2053532.

2. Pan, Z., Biegley, L., Taylor, A., & Zheng, H. (2024). A systematic review of learning analytics. *Journal of Learning Analytics*, 1–21. https://doi.org/10.18608/jla.2023.8093.

3. Mampadi, F., Chen, S. Y., Ghinea, G., & Chen, M.-P. (2011). Design of adaptive hypermedia learning systems: A cognitive style approach. *Computers & Education*, *56*(4), 1003–1011. https://doi.org/10.1016/j.compedu.2010.11.018.

4. Ifenthaler D., Yau J. Y.-K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development*, 68(4), 1961–1990. https://doi.org/10.1007/s11423-020-09788-z.

5. Breiman L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

6. Chen T., Guestrin C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785.

7. Hansen L. K., Salamon P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993–1001. https://doi.org/10.1109/34.58871.

8. Baker R. S. (2010). Data mining for education. *International Encyclopedia of Education*, 7, 112–118.

9. Siemens G., Long P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5), 30–40. https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education.

10. Kotsiantis S. B. (2012). Use of machine learning techniques for educational purposes: a decision support system for forecasting students' grades. *Artificial Intelligence Review*, 37(4), 331–344. https://doi.org/10.1007/s10462-011-9245-x.

11. Bousbia N., Gheffar A., Balla A. (2015). Adaptation based on navigation type and learning style. *In Advances in Web-Based Learning* – I*CWL 2013 Workshops: USL 2013, IWSLL 2013, KMEL 2013, IWCWL 2013, WIL 2013, and IWEEC 2013, Kenting, Taiwan, October 6-9, 2013, Revised Selected Papers* (pp. 23–31). https://doi.org/10.1007/978-3-662-46315-4_3.

12. D'Mello S. K., Graesser A. C. (2012). AutoTutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems*, 2(4), 23:1–23:39. https://doi.org/10.1145/2395123.2395128.

13. Pardos Z. A., Heffernan N. T. (2011). Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 12, 317–348. https://jmlr.org/papers/v12/pardos11a.html.

14. Shahiri A. M., Husain W., Rashid A. N. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72, 414–422. https://doi.org/10.1016/j.procs.2015.12.105.

15. Koedinger K. R., Kim J., Jia J. Z., McLaughlin E. A., Bier N. L. (2015). Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, 111–120. https://doi.org/10.1145/2724660.2724671.

16. Papamitsiou Z., Economides A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology & Society*, 17(4), 49–64. https://www.j-ets.net/ETS/journals/17_4/1.pdf.

17. Kizilcec R. F., Halawa S. (2015). Attrition and achievement gaps in online learning. *Proceedings of the Second ACM Conference on Learning @ Scale*, 57–66. https://doi.org/10.1145/2724660.2724680.

18. Хижняк А. В., Пріла О. А. (2025). Розробка системи автоматизованої генерації та перевірки параметризованих практичних завдань. *Технічні науки та технології*, 2(40), 221–233. https://doi.org/10.25140/2411-5363-2025-2(40)-221-233.

References

1. Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601–618. https://doi.org/10.1109/tsmcc.2010.2053532.

2. Pan, Z., Biegley, L., Taylor, A., & Zheng, H. (2024). A systematic review of learning analytics. *Journal of Learning Analytics*, 1–21. https://doi.org/10.18608/jla.2023.8093.

3. Mampadi, F., Chen, S. Y., Ghinea, G., & Chen, M.-P. (2011). Design of adaptive hypermedia learning systems: A cognitive style approach. *Computers & Education*, *56*(4), 1003–1011. https://doi.org/10.1016/j.compedu.2010.11.018.

4. Ifenthaler D., Yau J. Y.-K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development*, 68(4), 1961–1990. https://doi.org/10.1007/s11423-020-09788-z.

5. Breiman L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

6. Chen T., Guestrin C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785.

7. Hansen L. K., Salamon P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993–1001. https://doi.org/10.1109/34.58871.

8. Baker R. S. (2010). Data mining for education. *International Encyclopedia of Education*, 7, 112–118.

9. Siemens G., Long P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5), 30–40. https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education.

10. Kotsiantis S. B. (2012). Use of machine learning techniques for educational purposes: a decision support system for forecasting students' grades. *Artificial Intelligence Review*, 37(4), 331–344. https://doi.org/10.1007/s10462-011-9245-x.

11. Bousbia N., Gheffar A., Balla A. (2015). Adaptation based on navigation type and learning style. *In Advances in Web-Based Learning – ICWL 2013 Workshops: USL 2013, IWSLL 2013, KMEL 2013, IWCWL 2013, WIL 2013, and IWEEC 2013, Kenting, Taiwan, October 6-9, 2013, Revised Selected Papers* (pp. 23–31). https://doi.org/10.1007/978-3-662-46315-4_3.

12. D'Mello S. K., Graesser A. C. (2012). AutoTutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems*, 2(4), 23:1–23:39. https://doi.org/10.1145/2395123.2395128.

13. Pardos Z. A., Heffernan N. T. (2011). Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 12, 317–348. https://jmlr.org/papers/v12/pardos11a.html.

14. Shahiri A. M., Husain W., Rashid A. N. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72, 414–422. https://doi.org/10.1016/j.procs.2015.12.105.

15. Koedinger K. R., Kim J., Jia J. Z., McLaughlin E. A., Bier N. L. (2015). Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, 111–120. https://doi.org/10.1145/2724660.2724671.

16. Papamitsiou Z., Economides A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology & Society*, 17(4), 49–64. https://www.j-ets.net/ETS/journals/17_4/1.pdf.

17. Kizilcec R. F., Halawa S. (2015). Attrition and achievement gaps in online learning. *Proceedings of the Second ACM Conference on Learning @ Scale*, 57–66. https://doi.org/10.1145/2724660.2724680.

18. Khyzhniak A. V., Prila O. A. (2025). Rozrobka systemy avtomatyzovanoi heneratsii ta perevirky parametryzovanykh praktychnykh zavdan [Designing a system for automated generation and automated assessment of parameterized practical assignments]. *Tekhnichni nauky ta tekhnolohii – Technical Sciences and Technologies*, (2(40)), 221–233. https://doi.org/10.25140/2411-5363-2025-2(40)-221-233.

Дата першого надходження статті до видання: 11.12.2025
Дата прийняття статті до друку після рецензування: 29.12.2025

UDC 004.9

Anton Tymoshenko[1], Dmytro Lysenko[2]

[1]PhD Student at Information and Computer Systems Department
Chernihiv Polytechnic National University (Chernihiv, Ukraine)
E-mail: antontymoshenko@stu.cn.ua

[2]Doctor of Technical Sciences, Professor at Information and Computer Systems Department
Chernihiv Polytechnic National University (Chernihiv, Ukraine)
E-mail: lysenko.d@stu.cn.ua. **ORCID** https://orcid.org/0000-0001-6870-6120

APPLICATION OF MACHINE LEARNING METHODS IN AN ADAPTIVE STUDENT KNOWLEDGE ASSESSMENT SYSTEM

The article addresses the current problem of the inefficiency of traditional knowledge assessment methods in higher education, which is often caused by data fragmentation and the limited ability of classical models to capture complex non-linear dependencies in the educational process. The authors propose a concept for an adaptive assessment system based on a stacking ensemble that combines RandomForest, XGBClassifier, and LogisticRegression models, where RidgeClassifier with L2-regularization acts as a meta-model to increase the stability of forecasts. The scientific novelty of the work lies in the integration of cognitive indicators (test results) with a wide range of behavioral features, such as session duration, number of attempts, frequency of return to materials, and the emotional tone of responses, analyzed using the spaCy NLP library.

A complete data processing cycle is described: from the collection of activity logs from the LMS to the engineering of 15 key features and their normalization. During experimental verification on synthetic (N=150) and real (OULAD dataset, N=300) data, the high efficiency of the approach was confirmed: accuracy of 0.93 and 0.90 was obtained, respectively, with a value of ROC-AUC ≥ 0.98 for all knowledge levels (Low, Medium, High). Special attention is paid to the interpretability of results and the model's resistance to noise. In addition, a conceptual microservice architecture of the system is presented (Data Collector, Inference Engine, Recommendation Generator), based on the use of FastAPI and Kafka for the implementation of adaptive learning in real-time. The prospects for further development are substantiated, in particular the introduction of BERT models for in-depth text analysis and the transition to online learning methods.

Keywords: *adaptive learning; machine learning; stacking; learning analytics; knowledge classification; behavioral features; ensemble learning; RidgeClassifier; LMS integration.*

Fig.: 6. Table: 2. References: 18.

Тимошенко А. В., Лисенко Д. Е. Застосування методів машинного навчання в адаптивній системі оцінювання знань студентів. *Технічні науки та технології*, 2026. № 1(43). С. 254-268. DOI: https://doi.org/10.25140/2411-5363-2026-1(43)-254-268.