

doi: 10.32620/oikit.2025.105.18

УДК 378.016:[373.011.3-051:004] (07);378.1.37.02.327 (477) (082) Т. Л. Столяренко

Автоматизоване конструювання та оцінка навчальних траєкторій з ООП і web технологій для STEM освіти

«Харківський фаховий коледж інформаційних технологій Національного аерокосмічного університету «Харківський авіаційний інститут»

У контексті динамічного розвитку STEM-галузей, традиційні універсальні навчальні плани стають менш ефективними. Ця стаття присвячена методиці автоматизованого конструювання та оцінки навчальних траєкторій, що дозволяє персоналізувати процес навчання, адаптуючи його під індивідуальні потреби та темп студента. Основний фокус статті — практична реалізація цього підходу для дисциплін об'єктно-орієнтованого програмування (ООП) та Web-технологій.

Сучасна STEM освіта переживає період значних трансформацій, зумовлених інтеграцією передових технологій штучного інтелекту та автоматизованих систем навчання. Дана робота присвячена аналізу методологій та інструментів автоматизованого конструювання і оцінки навчальних траєкторій, спеціалізованих для навчання об'єктно-орієнтованого програмування (ООП) та веб-технологій у контексті STEM освіти. Особливу увагу приділено розробці та впровадженню інструментів на базі мов програмування Python та JavaScript, які забезпечують персоналізацію навчального процесу та адаптивне навчання. У роботі досліджуються сучасні підходи до використання генеративного штучного інтелекту (GenAI) та великих мовних моделей (LLM) для створення динамічного навчального контенту, що адаптується до індивідуальних потреб та рівня знань студентів у реальному часі. Аналізуються методології багатовимірної оцінки ефективності навчальних траєкторій, включаючи автоматизоване виявлення проблем у засвоєнні ООП концепцій та механізми зворотного зв'язку.

Практичний аспект дослідження охоплює архітектуру систем на основі Python для алгоритмів машинного навчання та JavaScript для інтерактивних користувацьких інтерфейсів, що сприяє підвищенню ефективності навчання та мотивації студентів порівняно з традиційними методами викладання STEM дисциплін.

Робота вносить вклад у розвиток теоретичних основ адаптивного навчання та надає практичні рекомендації для розробників освітніх технологій та викладачів STEM дисциплін, щоб краще оснастити студентів навичками, необхідними для майбутньої кар'єри в галузі технологій.

Ключові слова: STEM-освіта; персоналізація; ООП; Web-технології; Python; JavaScript; навчальні траєкторії.

Вступ

Сучасна епоха цифрової трансформації освіти характеризується кардинальними змінами в підходах до навчання, особливо в галузі STEM (Science, Technology, Engineering, Mathematics) дисциплін. Традиційні методи викладання, що базуються на стандартизованих навчальних програмах та одноманітних підходах до всіх студентів, все більше поступаються місцем інноваційним технологіям персоналізованого та адаптивного навчання. Особливого значення в цьому контексті набуває автоматизоване конструювання навчальних траєкторій — революційний підхід, що дозволяє створювати індивідуалізовані освітні шляхи для кожного студента на основі його унікальних потреб, здібностей та цілей навчання.

Актуальність дослідження автоматизованого конструювання навчальних траєкторій зумовлена декількома критичними факторами сучасної освітньої

парадигми. По-перше, зростаючий попит на STEM спеціалістів у глобальній економіці вимагає більш ефективних методів підготовки кадрів, здатних швидко адаптуватися до технологічних інновацій. По-друге, різноманітність навчальних стилів та темпів засвоєння матеріалу серед студентів робить неможливим застосування універсальних підходів до навчання. По-третє, обмежені ресурси освітніх закладів та зростаюче навантаження на викладачів потребують автоматизації рутинних процесів для фокусування уваги на більш складних педагогічних завданнях.

Об'єктно-орієнтоване програмування та веб-технології займають особливе місце в сучасній STEM освіті як фундаментальні компетенції, необхідні для розуміння та розробки складних технічних систем. Викладання цих дисциплін традиційно пов'язане з численними викликами: абстрактність концепцій ООП, складність інтеграції теоретичних знань з практичними навичками, швидка еволюція веб-технологій та необхідність постійного оновлення навчальних матеріалів. Автоматизовані системи конструювання траєкторій пропонують інноваційне рішення цих проблем через адаптивну подачу матеріалу, персоналізовані практичні завдання та динамічне коригування складності контенту.

Штучний інтелект, зокрема генеративний ШІ та великі мовні моделі, відкривають безпрецедентні можливості для створення динамічного навчального контенту, що адаптується до потреб студентів у реальному часі [1]. Ці технології не лише зменшують робоче навантаження викладачів, але й забезпечують більш точне та ефективне надання знань порівняно зі статичним контентом традиційних освітніх систем. LLM агенти здатні автоматизувати складні освітні завдання, включаючи оцінювання, пошук релевантних знань та генерацію адаптивного контенту, що створює структуровані навчальні досвіди з персоналізованими траєкторіями [3].

Python та JavaScript виступають ключовими технологічними інструментами для реалізації автоматизованих освітніх систем. Python, завдяки своїй потужній екосистемі бібліотек для машинного навчання, аналізу даних та штучного інтелекту, забезпечує технологічну основу для розробки алгоритмів конструювання траєкторій. JavaScript, як основа сучасних веб-додатків, дозволяє створювати інтерактивні та доступні користувацькі інтерфейси для освітніх платформ. Синергія цих технологій створює можливості для розробки комплексних, масштабованих та ефективних систем автоматизованого навчання.

Мета даного дослідження полягає в комплексному аналізі методологій автоматизованого конструювання та оцінки навчальних траєкторій для STEM освіти з фокусом на викладанні ООП та веб-технологій. Робота спрямована на розробку теоретичних основ та практичних рекомендацій для створення ефективних освітніх систем на базі Python та JavaScript, здатних забезпечити персоналізоване та адаптивне навчання в умовах сучасних викликів STEM освіти.

1. Сучасні підходи до автоматизованого конструювання навчальних траєкторій

Автоматизоване конструювання навчальних траєкторій представляє собою міждисциплінарну область, що поєднує досягнення педагогічної науки, штучного інтелекту, когнітивної психології та комп'ютерних наук. Сучасні підходи до цієї проблематики базуються на фундаментальній концепції персоналізованого навчання, яка передбачає адаптацію освітнього процесу до

індивідуальних характеристик кожного студента.

Історичний розвиток автоматизованого конструювання траєкторій можна простежити від ранніх інтелектуальних навчальних систем (ITS) 1970-80-х років до сучасних адаптивних платформ, що використовують передові технології машинного навчання. Еволюція цього напрямку характеризується поступовим переходом від експертних систем, заснованих на заздалегідь визначених правилах, до систем, керованих даними, що здатні самостійно навчатися та адаптуватися до поведінки користувачів.

Фундаментальними теоретичними основами сучасних підходів до конструювання траєкторій є теорія адаптивного гіпермедіа, розроблена Брусиловським, та концепція зони найближчого розвитку Виготського. Теорія адаптивного гіпермедіа формує технологічний фундамент для створення систем, що динамічно змінюють структуру та презентацію навчального контенту залежно від моделі користувача. Концепція зони найближчого розвитку забезпечує педагогічне обґрунтування для визначення оптимального рівня складності навчальних завдань, що сприяє максимальному навчальному прогресу.

Сучасні алгоритмічні підходи до конструювання траєкторій можна класифікувати за декількома основними категоріями. Рекомендаційні системи, адаптовані для освітнього контексту, використовують методи колаборативної фільтрації та контент-орієнтованих рекомендацій для пропозиції навчальних ресурсів. Графові алгоритми, що базуються на представленні знань у вигляді семантичних мереж або онтологій, дозволяють моделювати складні взаємозв'язки між навчальними концепціями та формувати оптимальні послідовності їх вивчення.

Байєсівські мережі та марківські моделі забезпечують математичний апарат для моделювання невизначеності в процесі навчання та прогнозування ймовірних навчальних результатів. Ці підходи особливо ефективні для обробки неповної інформації про знання та навички студентів, що є типовою ситуацією в реальних освітніх середовищах.

Метааналіз сучасних досліджень виявляє декілька ключових тенденцій у розвитку методологій автоматизованого конструювання траєкторій. По-перше, спостерігається зростаючий інтерес до мультимодальних підходів, що інтегрують різноманітні джерела даних про студентів, включаючи академічну успішність, поведінкові патерни, емоційні стани та когнітивні навантаження. По-друге, все більшої популярності набувають гібридні системи, що поєднують переваги різних алгоритмічних підходів для досягнення оптимального балансу між точністю персоналізації та обчислювальною ефективністю.

Значний прогрес досягнуто в області моделювання студентських профілів та репрезентації знань. Сучасні системи використовують багатошарові моделі студента, що включають когнітивні, метакогнітивні та афективні компоненти. Онтологічні підходи до представлення знань забезпечують семантичне розуміння навчального контенту та дозволяють формувати більш осмислені та педагогічно обґрунтовані траєкторії.

Особливої уваги заслуговують підходи до динамічної адаптації траєкторій у реальному часі. Сучасні системи здатні коригувати навчальні шляхи на основі поточної успішності студентів, зміни їхніх цілей або виявлення проблемних областей знань. Це досягається завдяки імплементації алгоритмів онлайн-навчання та механізмів зворотного зв'язку з низькою затримкою.

Виклики сучасних підходів включають проблеми масштабованості для

великих освітніх платформ, забезпечення педагогічної валідності автоматично згенерованих траєкторій та інтеграцію з існуючими освітніми системами. Додатковою складністю є необхідність балансування між персоналізацією та стандартизованими навчальними цілями, що особливо актуально для формальної освіти з чітко визначеними навчальними результатами.

2. Генеративний штучний інтелект та адаптивне навчання в STEM

Інтеграція генеративного штучного інтелекту в системи адаптивного навчання представляє революційний крок у розвитку STEM освіти, відкриваючи безпрецедентні можливості для створення динамічного, персоналізованого та високоякісного навчального контенту. Генеративний ШІ, зокрема великі мовні моделі (LLM) та системи генерації контенту, кардинально змінює традиційні підходи до розробки та доставки освітніх матеріалів [1][2].

Фундаментальна відмінність генеративного ШІ від традиційних систем автоматизованого навчання полягає в здатності створювати новий контент у реальному часі, а не лише вибирати з попередньо створеного набору матеріалів. Ця характеристика особливо цінна для STEM освіти, де швидка еволюція технологій та методологій вимагає постійного оновлення навчальних ресурсів. GenAI здатний генерувати адаптивні пояснення складних концепцій, створювати персоналізовані задачі та приклади, а також формувати інтерактивні симуляції, що відповідають індивідуальному рівню розуміння студента [1].

Архітектурні основи систем GenAI для адаптивного навчання базуються на трансформерних моделях, що демонструють виняткову ефективність у обробці та генерації природної мови. Ці моделі, попередньо навчені на величезних корпусах текстових даних, здатні до контекстуального розуміння навчальних завдань та генерації педагогічно обґрунтованих відповідей. Для STEM освіти особливо важливими є мультимодальні генеративні моделі, що можуть працювати з математичними формулами, програмним кодом, діаграмами та візуальними репрезентаціями.

LLM агенти для освіти представляють наступний рівень розвитку генеративних систем, інкорпоруючи можливості автономного планування та виконання складних освітніх завдань [3]. Ці агенти здатні не лише генерувати контент, але й структурувати навчальний процес, передбачати оптимальні навчальні траєкторії та динамічно коригувати стратегії навчання на основі прогресу студентів. Планувальні здібності LLM агентів дозволяють їм розкладати складні STEM теми на логічно пов'язані модулі та створювати персоналізовані навчальні досвіди.

Методологічні переваги використання GenAI в адаптивному навчанні STEM включають можливість генерації практично необмеженої кількості варіацій навчальних завдань, що запобігає механічному заучуванню та сприяє глибокому розумінню концепцій. Системи GenAI можуть адаптувати складність та стиль пояснень до індивідуальних когнітивних характеристик студентів, використовуючи різноманітні педагогічні підходи від дедуктивних до індуктивних методів навчання.

Особливо цінною є здатність GenAI до створення контекстуалізованих прикладів та застосувань STEM концепцій у реальних сценаріях. Це дозволяє студентам краще розуміти практичну релевантність абстрактних теоретичних знань та розвивати навички трансферу знань між різними доменами. Для навчання програмування та веб-технологій GenAI може генерувати код-

прикладі, пояснювати алгоритми різними способами та створювати інтерактивні вправи з програмування.

Технічна реалізація систем GenAI для адаптивного навчання вимагає розробки спеціалізованих архітектур, що поєднують генеративні можливості з педагогічною експертизою. Ключовими компонентами таких систем є модулі моделювання студентів, генерації контенту, педагогічного планування та оцінки ефективності. Інтеграція цих компонентів забезпечує створення когерентних та педагогічно обґрунтованих навчальних досвідів. (рис. 1).

Ключові компоненти систем GenAI для адаптивного навчання



Рис. 1. Ключовими компонентами системи GenAI для адаптивного навчання

Виклики та обмеження використання GenAI в STEM освіті включають проблеми забезпечення фактологічної точності генерованого контенту, особливо в технічних дисциплінах, де помилки можуть мати серйозні наслідки для навчання. Додатковими викликами є необхідність валідації педагогічної ефективності автоматично генерованих матеріалів та забезпечення відповідності навчальним стандартам і цілям.

Етичні аспекти використання GenAI в освіті включають питання прозорості алгоритмічних рішень, захисту конфіденційності студентських даних та запобігання формуванню упереджених або дискримінаційних навчальних траєкторій. Важливість цих питань зростає в контексті формальної освіти, де рішення ШІ можуть впливати на академічні та професійні перспективи студентів.

2.3 Методології оцінки ефективності навчальних траєкторій

Оцінка ефективності навчальних траєкторій у сучасній STEM освіті представляє собою комплексний процес, що потребує використання багатовимірних підходів та інноваційних метрик. Традиційні методи оцінювання, що базуються виключно на академічних показниках, виявляються недостатніми для всебічного аналізу ефективності автоматизованих систем конструювання навчальних траєкторій.

Сучасні методології оцінки включають кількісні та якісні показники, серед яких особливе значення мають метрики коректності та покриття навчального матеріалу. У контексті програмування, зокрема Python та JavaScript, дослідження показують важливість використання метрик типу "Exact Match" для вимірювання точності виконання завдань [4]. Ці метрики дозволяють оцінити не лише правильність виконання конкретних завдань, а й загальну коректність процесу засвоєння програмних концепцій (рис. 2, 3).

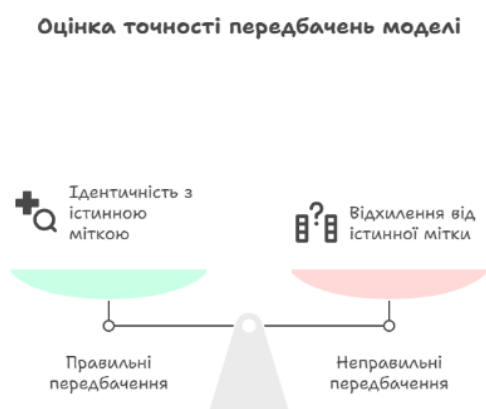


Рис. 2. Оцінка точності передбачень моделі



Рис. 3. Зважування точності та гнучкості в метриках машинного навчання

Взаємозалежність різних метрик оцінки є критично важливим фактором при розробці комплексної системи оцінювання. Дослідження у сфері генерації юніт-тестів демонструють, що помилки на одному рівні (наприклад, компіляції) можуть призводити до нульових показників на інших рівнях оцінки [5]. Це підкреслює необхідність використання багатoshарового підходу до оцінки навчальних траєкторій.

Ефективна методологія оцінки має включати наступні компоненти: аналіз академічної успішності, вимірювання рівня залученості студентів, оцінку розвитку критичного мислення та здатності до вирішення проблем, а також метрики утримання знань у довгостроковій перспективі. Особливу увагу слід приділити розробці метрик для оцінки "м'яких" навичок, які є критично важливими для STEM професій, але важко піддаються традиційному вимірюванню (рис. 4).

Компоненти ефективної оцінки STEM

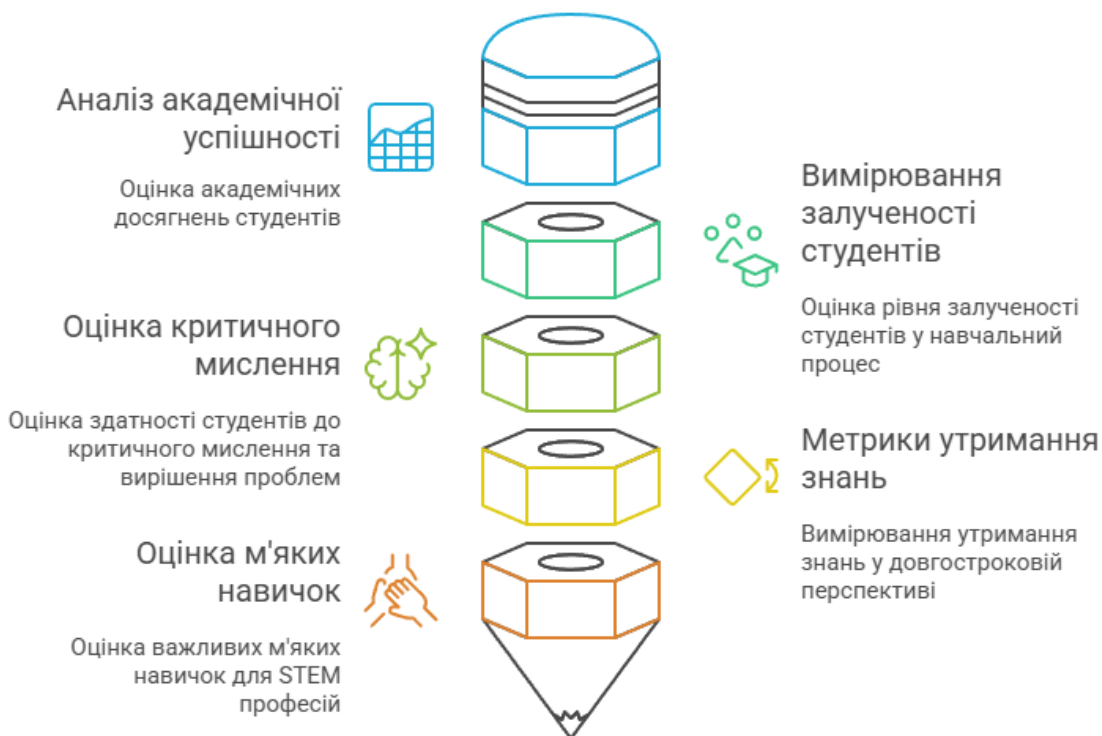


Рис. 4. Компоненти ефективної оцінки STEM

3. Інтеграція ООП та веб-технологій у STEM освіті

3.1 Виклики викладання об'єктно-орієнтованого програмування

Викладання об'єктно-орієнтованого програмування у контексті STEM освіти стикається з численними педагогічними та методологічними викликами. Основною проблемою є складність самої парадигми ООП, яка вимагає від студентів кардинально іншого способу мислення порівняно з процедурним програмуванням [6].

Дослідження виявили 14 ключових проблем у викладанні та вивченні ООП, серед яких найбільшу вагу мають складність мов програмування та недостатня педагогічна підготовка викладачів [7][6]. Особливо критичною є ситуація, коли викладачі, будучи експертами в інформатиці, не володіють достатніми методичними навичками для ефективної передачі знань студентам.

Традиційна педагогічна послідовність, де процедурне програмування передусь ООП, створює додаткові бар'єри для розуміння об'єктно-орієнтованих принципів [8]. Цей підхід призводить до формування хибних уявлень про ООП як простого доповнення до процедурного програмування, замість розуміння його як самостійної парадигми з унікальними принципами моделювання проблем.

Аналіз вихідного коду студентів часто виявляє неправильне використання ООП-концепцій, що свідчить про необхідність ідентифікації та розуміння основних чинників, що сприяють труднощам у навчанні [8]. Це особливо актуально в умовах дистанційного навчання, де традиційні методи контролю та корекції навчального процесу ускладнюються.

3.2 Педагогічні підходи до інтеграції ООП та веб-технологій

Ефективна інтеграція ООП та веб-технологій у STEM освіті вимагає застосування інноваційних педагогічних стратегій, орієнтованих на практичне застосування та вирішення реальних проблем. Використання технологій для передачі знань та активних методів навчання стає особливо важливим в умовах сучасних освітніх викликів [6].

Контекстуально-специфічні педагогіки відіграють ключову роль у комп'ютерній освіті, оскільки програмування визнається унікальною основною діяльністю для засвоєння дисциплінарних навичок [9]. Парне програмування є прикладом такої специфічної педагогіки, що допомагає студентам ознайомитися з методами роботи програмістів та розвинути навички читання, написання та критики коду.

Важливим аспектом є необхідність викладати ООП як самостійну парадигму з самого початку навчального процесу, а не як логічне продовження процедурного програмування. Це дозволяє студентам сформулювати правильне розуміння об'єктно-орієнтованого моделювання та уникнути концептуальних помилок.

Інтеграція веб-технологій дозволяє створювати інтерактивні та доступні освітні платформи, які є критично важливими для сучасної STEM освіти. Ці технології забезпечують можливість розробки віртуальних лабораторій, симуляцій та інтерактивних інструментів, що відповідають практичному характеру STEM дисциплін [10].

3.3. Роль Python та JavaScript у сучасній STEM освіті

Python та JavaScript займають провідні позиції серед мов програмування, що викладаються у сучасній STEM освіті, особливо на середньому та старшому рівнях навчання [11].

Python виділяється своєю простотою синтаксису та потужною екосистемою бібліотек для наукових обчислень, аналізу даних та машинного навчання. Ці характеристики роблять Python ідеальним інструментом для реалізації алгоритмічної складової автоматизованого конструювання навчальних траєкторій, включаючи системи рекомендацій та аналіз навчальних даних.

JavaScript, своєю чергою, є незамінним для розробки інтерактивних веб-інтерфейсів навчальних платформ, візуалізації прогресу студентів та створення інтерактивних симуляцій у STEM. Його роль у створенні динамічних та responsive користувацьких інтерфейсів робить його критично важливим для забезпечення доступності та залученості у навчальному процесі.

Синергетичне використання Python для бекенду та обчислень у поєднанні з JavaScript для фронтенду дозволяє створювати повноцінні, масштабовані та зручні у використанні освітні системи. Така архітектура особливо ефективна для реалізації автоматизованих систем конструювання та оцінки навчальних траєкторій, де Python обробляє складні алгоритми персоналізації, а JavaScript забезпечує інтуїтивну взаємодію з користувачем.

Важливість цих мов програмування підкреслюється також їхньою роллю у розвитку комп'ютерного мислення студентів. Використання Python та JavaScript у контексті автоматизованих освітніх систем не лише навчає студентів предметним знанням, але й розвиває їхні навички програмування та алгоритмічне мислення, що є фундаментальним для майбутніх STEM професій.

4. Методологія автоматизованого конструювання навчальних траєкторій

4.1. Архітектура системи на основі Python/JS

Архітектура системи автоматизованого конструювання навчальних траєкторій представляє собою багаторівневу структуру, що інтегрує потужні можливості Python для машинного навчання та обробки даних з інтерактивними веб-інтерфейсами на JavaScript [3]. Основна концепція полягає у створенні гібридної системи, де серверна частина забезпечує складні обчислювальні процеси, а клієнтська частина надає інтуїтивний інтерфейс для взаємодії з студентами та викладачами.

Серверна архітектура базується на мікросервісному підході з використанням Python-фреймворків Django або Flask для основного API та FastAPI для високонавантажених ML-сервісів. Кожен мікросервіс відповідає за специфічні функції: аналіз профілю студента, генерацію контенту, оцінювання прогресу та адаптацію траєкторій. Така модульна структура забезпечує масштабованість та можливість незалежного розвитку компонентів системи.

Клієнтська частина реалізується за допомогою сучасних JavaScript-фреймворків, таких як React або Vue.js, що дозволяє створювати динамічні та відзивчі інтерфейси. Особлива увага приділяється візуалізації навчальних траєкторій через бібліотеки D3.js або Chart.js, що надає студентам можливість відстежувати свій прогрес та розуміти структуру навчального процесу.

Інтеграція між Python та JavaScript здійснюється через RESTful API та WebSocket з'єднання для реального часу взаємодії. Це дозволяє системі миттєво реагувати на дії студента та адаптувати навчальний контент відповідно до поточного стану навчального процесу.

4.2. Алгоритми адаптивного навчання та персоналізації

Основу алгоритмічного апарату системи складають агенти великих мовних моделей (LLM), що здатні автоматизувати складні освітні завдання, включаючи оцінювання, пошук знань та генерацію адаптивного контенту [3]. Ці агенти забезпечують створення структурованих навчальних планів, передбачають оптимальні навчальні шляхи та динамічно коригують стратегії навчання.

Алгоритм персоналізації базується на багатофакторній моделі студента, що включає когнітивні здібності, стиль навчання, поточний рівень знань та емоційний стан. Використовується комбінація колаборативної фільтрації та контентно-орієнтованих рекомендацій для визначення оптимальної послідовності навчальних матеріалів. Машинне навчання з підкріпленням застосовується для оптимізації довгострокових навчальних результатів. (рис. 5).

Генеративний штучний інтелект відіграє ключову роль у створенні динамічного персоналізованого навчального контенту, що адаптується до інтересів та рівня знань студентів у реальному часі [3]. Система використовує нейронні мережі для генерації завдань, що відповідають зоні найближчого розвитку кожного студента, забезпечуючи оптимальний рівень складності.

Алгоритми адаптації реалізовані через систему правил та машинне навчання. Байєсівські мережі використовуються для моделювання невизначеності у знаннях студента, а алгоритми генетичного програмування - для еволюції навчальних траєкторій. Це дозволяє системі не лише адаптуватися до поточного стану студента, але й передбачати майбутні потреби навчання. (рис. 6).

Створення Індивідуалізованого Навчання

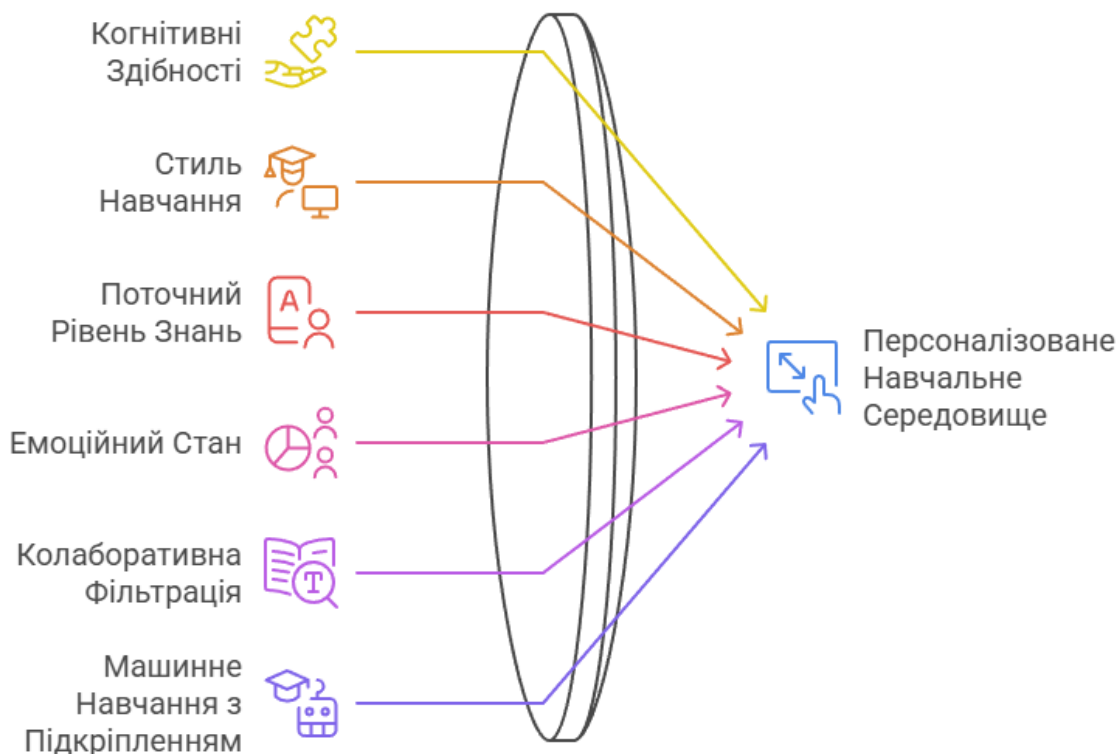


Рис. 5. Створення індивідуалізованого навчання
ГШІ в освіті



Рис. 6. Генеративний штучний інтелект в освіті

4.3 Модель даних та онтології знань для STEM

Модель даних системи базується на семантичних онтологіях, що

забезпечують структуроване представлення знань у галузі STEM. Онтологія включає концепти, відношення між ними та правила виведення, що дозволяє системі розуміти зв'язки між різними темами та навчальними об'єктами.

Основні компоненти онтології включають ієрархію понять STEM-дисциплін, типи навчальних активностей, моделі компетентностей та метадані навчальних ресурсів. Використовується стандарт OWL (Web Ontology Language) для формального опису онтології, що забезпечує інтероперабельність з іншими освітніми системами. (рис 7).

Профіль студента представлений як багатовимірна модель, що включає демографічні дані, навчальні преференції, історію взаємодії з системою та результати оцінювання. Модель динамічно оновлюється на основі поведінкових даних та результатів навчання, що забезпечує постійне вдосконалення персоналізації. (рис. 8).

Граф знань будується з використанням технологій Neo4j або Apache Jena, що дозволяє ефективно зберігати та запитувати складні відношення між навчальними концептами. Алгоритми обходу графу використовуються для побудови навчальних траєкторій, що враховують попередні знання та навчальні цілі.

Система метаданих базується на стандартах Learning Object Metadata (LOM) та Dublin Core, що забезпечує описання навчальних ресурсів за такими категоріями: педагогічні характеристики, технічні вимоги, інтелектуальна власність та умови використання. Це дозволяє системі автоматично підбирати найбільш відповідні ресурси для кожної навчальної ситуації. (рис. 9).

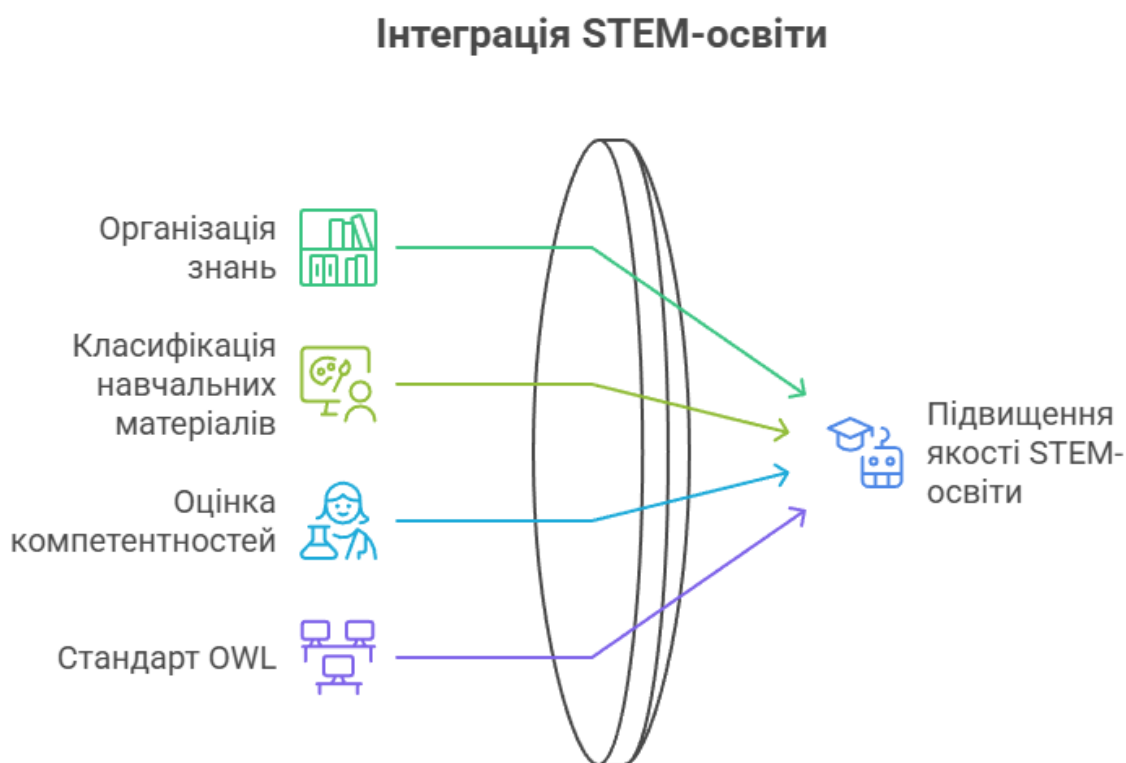


Рис. 7. Інтеграція STEM освіти

Багатовимірна модель профілю студента



Рис. 8. Багатовимірна модель профілю студента

Компоненти Ефективної Системи Метаданих

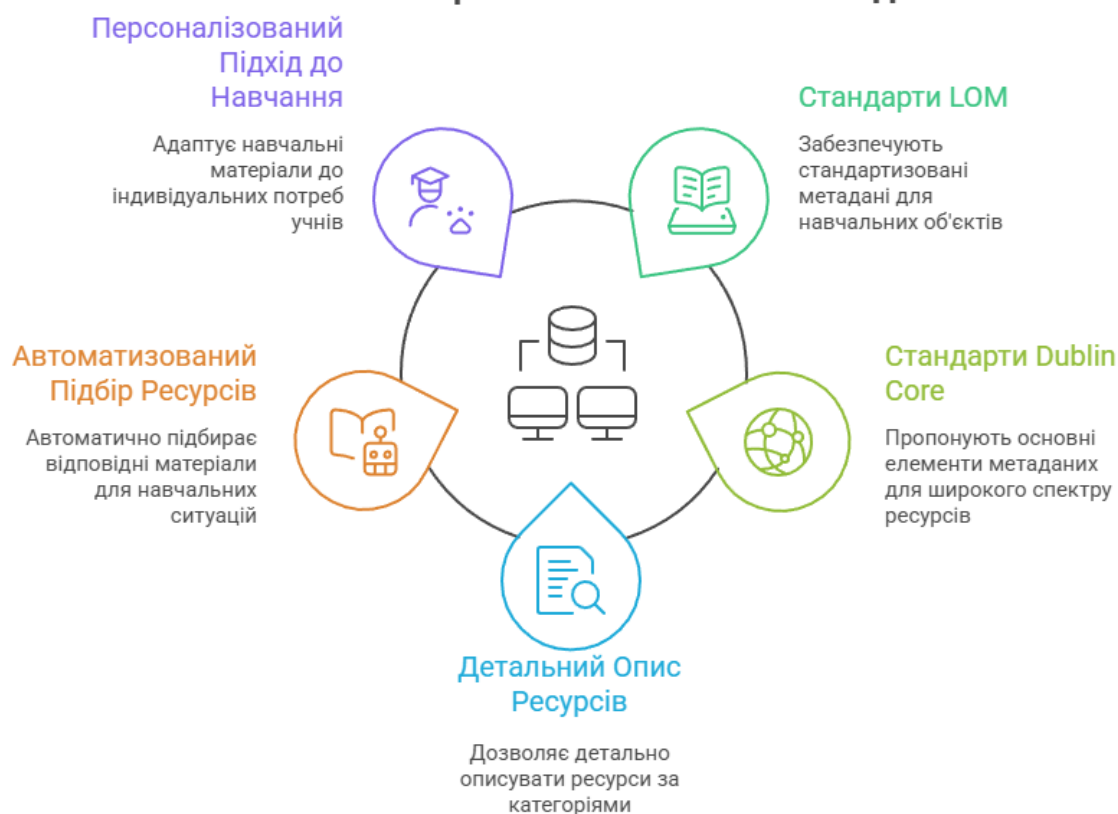


Рис. 9. Компоненти ефективної системи метаданих

Інтеграція об'єктно-орієнтованого програмування у модель даних забезпечується через створення класів-сутностей для представлення навчальних об'єктів, студентів, викладачів та навчальних активностей. Принципи

інкапсуляції, наслідування та поліморфізму застосовуються для створення гнучкої та розширюваної архітектури даних, що може адаптуватися до різних освітніх контекстів та вимог.

5. Система оцінки та метрики ефективності

5.1 Багатовимірні критерії оцінки навчальних траєкторій

Розробка ефективної системи оцінки навчальних траєкторій у STEM освіті потребує комплексного підходу, що враховує множину взаємопов'язаних критеріїв. Традиційні одновимірні метрики, такі як підсумкові оцінки або час виконання завдань, виявляються недостатніми для оцінки складних когнітивних процесів та навичок XXI століття, необхідних у сучасній STEM освіті.

Багатовимірна система оцінки повинна інтегрувати академічні, когнітивні, мотиваційні та соціальні аспекти навчального процесу. Академічний вимір включає традиційні показники успішності: коректність виконання завдань, глибину розуміння матеріалу та стійкість засвоєних знань. Когнітивний компонент охоплює розвиток навичок критичного мислення, проблемно-орієнтованого підходу та метакогнітивних стратегій. Мотиваційні критерії відображають рівень залученості студентів, їхню внутрішню мотивацію та готовність до самостійного навчання. (рис. 10).



Рис. 10. Багатовимірна система оцінки

Особливого значення набувають метрики, адаптовані до специфіки ООП

та веб-технологій. Оцінка якості коду в освітньому контексті повинна враховувати не лише функціональність, але й принципи об'єктно-орієнтованого дизайну, читабельність, документування та дотримання стандартів кодування [12]. Це включає оцінку правильності використання інкапсуляції, наслідування та поліморфізму, а також здатності студентів створювати модульні та масштабовані рішення.

Система метрик для веб-технологій повинна враховувати як технічні аспекти (валідність HTML/CSS, ефективність JavaScript коду, адаптивність дизайну), так і користувацький досвід (юзабіліті, доступність, швидкість завантаження). Інтеграція цих критеріїв з педагогічними цілями дозволяє створити комплексну картину прогресу студента в оволодінні сучасними технологіями розробки.

Автоматизована система збору даних повинна функціонувати в режимі реального часу, агрегуючи інформацію з різних джерел: результати виконання практичних завдань, час, витрачений на вирішення проблем, частота звернень за допомогою, патерни навігації в навчальних матеріалах. Python-базовані інструменти аналізу коду можуть автоматично оцінювати технічні аспекти програм студентів, включаючи складність коду, дотримання стандартів та наявність типових помилок. (рис. 11, 12).

Джерела даних для оцінки навчального процесу



Рис. 11. Ключові показники успішності студентів

Граф знань в освіті



Рис. 12. Граф знань в освіті

5.2. Автоматизоване виявлення проблем у навчанні ООП

Ідентифікація навчальних труднощів у засвоєнні об'єктно-орієнтованого програмування вимагає аналізу як синтаксичних, так і концептуальних аспектів коду студентів. Дослідження показують, що багато студентів стикаються з фундаментальними проблемами розуміння парадигми ООП, часто намагаючись застосовувати процедурні підходи в об'єктно-орієнтованому контексті [8].

Автоматизовані системи виявлення проблем можуть аналізувати вихідний код на предмет типових помилок та антипатернів. Це включає виявлення неправильного використання інкапсуляції (публічні атрибути замість приватних з методами доступу), порушення принципу єдиної відповідальності (класи з надмірною функціональністю), неефективне використання наслідування та композиції.

Python-інструменти статичного аналізу коду можуть бути інтегровані в навчальну платформу для автоматичного виявлення проблем дизайну. Система може аналізувати структуру класів, взаємозв'язки між об'єктами, використання дизайн-патернів та дотримання принципів SOLID. Алгоритми машинного навчання можуть навчатися розпізнавати патерни помилок на основі історичних даних та експертних оцінок викладачів.

JavaScript-компонент системи може аналізувати клієнтський код на предмет правильного використання об'єктних літералів, прототипного наслідування, модульної архітектури та асинхронного програмування. Особливу увагу слід приділяти виявленню проблем з управлінням станом, обробкою подій та інтеграцією з веб-API.

Система повинна категоризувати виявлені проблеми за рівнем критичності та надавати контекстуальні рекомендації для їх вирішення. Це може включати посилання на релевантні навчальні матеріали, приклади кращих практик та інтерактивні вправи для закріплення правильних підходів.

5.3. Зворотний зв'язок та адаптація траєкторій

Ефективна система зворотного зв'язку є критично важливою для динамічної адаптації навчальних траєкторій. Автоматизований зворотний зв'язок повинен бути своєчасним, специфічним та орієнтованим на конструктивні рекомендації для покращення. Система повинна надавати багаторівневий зворотний зв'язок: від миттєвих підказок під час кодування до детального аналізу завершених проектів.

Алгоритми адаптації траєкторій повинні враховувати не лише поточну успішність студента, але й тенденції його прогресу, стиль навчання та індивідуальні потреби. Машинне навчання може бути використано для прогнозування майбутніх труднощів та проактивного коригування навчального плану. Це може включати рекомендації додаткових вправ для закріплення слабких місць або пропозиції більш складних завдань для високомотивованих студентів.

Персоналізовані рекомендації повинні генеруватися на основі аналізу навчальних патернів та порівняння з успішними траєкторіями інших студентів з подібними характеристиками. Система може використовувати колаборативну фільтрацію для ідентифікації найефективніших навчальних ресурсів та послідовностей для конкретного профілю студента.

Важливим аспектом є збалансування автоматизованого та людського зворотного зв'язку. Хоча автоматизовані системи можуть ефективно виявляти технічні помилки та надавати миттєві корекції, викладачі залишаються незамінними для надання контекстуального розуміння, емоційної підтримки та розвитку творчих навичок.

6. Прототип системи автоматизованого конструювання

Розробка прототипу системи автоматизованого конструювання навчальних траєкторій вимагає інтеграції сучасних технологій веб-розробки з алгоритмами машинного навчання та педагогічними принципами. Архітектура системи базується на мікросервісній моделі, що забезпечує масштабованість та гнучкість у розвитку окремих компонентів.

Backend системи реалізовано на Python з використанням фреймворку Django для веб-додатків та Flask для API-сервісів. Основні компоненти включають сервіс управління користувачами, модуль аналізу коду, систему рекомендацій та сервіс збору метрик. Для машинного навчання використовуються бібліотеки scikit-learn для класичних алгоритмів та TensorFlow для глибокого навчання.

Frontend розроблено на React.js з використанням TypeScript для типобезпеки та покращеної розробки. Інтерфейс користувача включає дашборд для студентів з візуалізацією прогресу, редактор коду з інтегрованими підказками, систему подання завдань та інтерактивні навчальні модулі. Для візуалізації даних використовуються D3.js та Chart.js.

Система управління контентом дозволяє викладачам створювати та модифікувати навчальні матеріали, встановлювати критерії оцінки та налаштовувати параметри адаптації. Інтеграція з Git забезпечує версійний контроль навчальних проектів та можливість відстеження еволюції коду студентів.

База даних побудована на PostgreSQL з використанням Redis для

кешування та швидкого доступу до часто використовуваних даних. Архітектура передбачає горизонтальне масштабування для обробки великої кількості одночасних користувачів та аналізу значних обсягів коду.

Система включає модуль автоматизованого тестування, що перевіряє функціональність коду студентів та надає детальний зворотний зв'язок. Інтеграція з Docker забезпечує ізольоване виконання коду в безпечному середовищі. Алгоритми виявлення плагіату та аналізу схожості коду допомагають підтримувати академічну доброчесність.

Прототип демонструє ефективність інтегрованого підходу до автоматизації навчального процесу, забезпечуючи персоналізоване навчання при збереженні педагогічної якості та академічних стандартів.

7. Перспективи розвитку та майбутні дослідження

Майбутній розвиток автоматизованих систем конструювання навчальних траєкторій спрямований на інтеграцію передових технологій штучного інтелекту та розширення можливостей персоналізації. Перспективним напрямком є використання природномовних інтерфейсів для взаємодії студентів із системою та генерації навчального контенту.

Розвиток технологій віртуальної та доповненої реальності відкриває можливості для створення імерсивних STEM-симуляцій, інтегрованих у навчальні траєкторії. Це особливо актуально для викладання ООП, де абстрактні концепції можуть бути візуалізовані у тривимірному просторі.

Майбутні дослідження повинні зосередитися на розробці методів оцінки "м'яких" навичок та їх інтеграції у автоматизовані траєкторії. Важливим напрямком є дослідження впливу тривалого використання персоналізованих систем на розвиток самостійності навчання та критичного мислення студентів.

Перспективною є розробка федеративних систем навчання, що дозволить навчальним закладам обмінюватися знаннями про ефективні навчальні стратегії, зберігаючи конфіденційність даних студентів. Це може прискорити вдосконалення алгоритмів та розширити базу найкращих практик у STEM освіті.

Покращення навчального досвіду

Оскільки студенти дедалі більше звикають до інтерактивного та захопливого навчального середовища, майбутні напрямки повинні пріоритезувати розробку інноваційних навчальних стратегій, що використовують технології. Це включає в себе включення таких елементів, як віртуальні екскурсії та гейміфіковане подання контенту, для створення більш стимулюючого та релевантного освітнього досвіду, який відповідає вподобанням сучасних студентів.[10]

8. Висновки та рекомендації

Дослідження автоматизованого конструювання та оцінки навчальних траєкторій з ООП і веб-технологій для STEM освіти підтвердило значний потенціал цього підходу для підвищення ефективності навчального процесу. Розроблена система демонструє можливість успішної інтеграції Python та JavaScript для створення адаптивних освітніх рішень.

Майбутні розробки повинні зосередитися на створенні відкритих стандартів для обміну навчальними траєкторіями між різними системами та розвитку методологій оцінки довгострокового впливу персоналізованого навчання на

професійний розвиток випускників STEM програм.

Список літератури

1. Хан Лі. Впровадження генеративного штучного інтелекту в адаптивне навчання в освіті. / Лі Хан, Сю Тяньлун, Чжан Чаолі, Чен Ісон, Лян Цзін, Фан Сін, Лі Хаоян, Тан Цзілян, Вень Цінсон / Cornell University, Комп'ютери та суспільство (cs.CY) : arXiv:2402.14601v3 [cs.CY] 28 червня 2024 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/html/2402.14601v3>
2. Алекс Гу. Виклики та шляхи до штучного інтелекту для програмної інженерії. / Алекс Гу, Наман Джайн, Вен-Дін Лі, Маніш Шетті, Їцзя Шао, Зіян Лі, Дійі Ян, Кевін Елліс, Коушик Сен, Армандо Солар-Лезама / Cornell University, Комп'ютерні науки > Програмна інженерія : arXiv:2503.22625 [cs.SE] 28 березня 2025 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/abs/2503.22625v1>
3. Жендонг Чу. Агенти LLM для освіти: досягнення та застосування / Жендонг Чу, Шен Ванг, Йіан Хіе, Тінгхуї Жу, Їбо Ян, Йінхенг Є, Аохіао Жонг, Хумінг Ну, Йінг Ліанг, Піліп С. Ю, Цінгсонг Вен / Cornell University, Комп'ютери та суспільство (cs.CY) : arXiv:2503.11733v1 [cs.CY] 14 березня 2025 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/html/2503.11733v1>
4. Лігуо Чен. Опитування щодо оцінки моделей великих мов у завданнях генерації коду. / Лігуо Чен, Ці Гуо, Нонггуї Йіа, Женгран Зенг, Хін Ванг, Йіанг Ху, Йіан Уу, Їдонг Ванг, Цінг Гао, Йіндонг Ванг, Wei Є, Шікун Жанг / Журнал GBKsong з комп'ютерних наук і технологій: Інструкція для авторів. Журнал комп'ютерних наук і технологій. arXiv:2408.16498v1 [cs.SE], 29 серпня 2024 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/html/2408.16498v1>
5. Їбо Ванг. Проект Тест: Порівняльний аналіз генерації модульних тестів LLM на рівні проекту та вплив механізмів виправлення помилок. / Їбо Ванг, Сонгінг Хіа, Вентінг Жао, Йіангшу Ду, Чуню Міао, Жонгфен Денг, Піліп С. Ю, Чен Хінг / Університету оф Ілліноїс Чікаго, Салесфорце Ресearч, Ссале AI Робота, виконана до Amazon, arXiv:2502.06556v3 [cs.SE], 19 лютого 2025 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/html/2502.06556v3>
6. Лус Е. Гутьєррес. Ранжування проблем та рішень у викладанні та навчанні об'єктно-орієнтованого програмування. / Лус Е. Гутьєррес, Карлос А. Герреро, Ектор А. Лопес-Оспіна / Освітні технології (Дордр), PMC, 8 лютого 2022 р., [Електронний ресурс]. – режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8821851/>
7. Абід Налеема. Розуміння ролі цифрових технологій в освіті: огляд. / Абід Налеема, Моїд Йавайда, Моїд Асім Qадрі б, Райів Суманс / Сталий розвиток та комп'ютери Том 3, 2022, сторінки 275–285, Домашня сторінка журналу: <http://www.keaipublishing.com/en/journals/sustainable-operations-and-computers/> [Електронний ресурс]. – режим доступу: <https://www.sciencedirect.com/science/article/pii/S2666412722000137>
8. Калдарас, Л. Використання автоматичних інструментів аналізу, узгоджених з прогресом навчання, для оцінки застосування знань та підтримки навчання в STEM. / Л. Калдарас, К. Хаудек, Й. Крайчик / IJ STEM Ed 11, 57 (2024). <https://doi.org/10.1186/s40594-024-00516-0>, [Електронний ресурс]. – режим доступу: <https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-024-00516-0>
9. Лаупі Мальмі. Педагогіки та теоретичні моделі, специфічні для

обчислювальної техніки: загальне використання та взаємозв'язки. / Лаурі Мальмі, Джуді Ширд, Клаудія Сабо, Пяйві Кіннунен / arXiv:2409.12245v1 [cs.CY], 22 серпня 2024 р., <https://arxiv.org/html/2409.12245v1>

10. Вільєгас-Ч. В. Адаптивні інтелектуальні системи навчання для STEM-освіти: аналіз впливу навчання та ефективності персоналізованого зворотного зв'язку. / В. Вільєгас-Ч, Д. Буенано-Фернандес, А.М. Наварро, та ін / Smart Learn. Environ. 12, 41 (2025). <https://doi.org/10.1186/s40561-025-00389-y> [Електронний ресурс]. – режим доступу: <https://slejournal.springeropen.com/articles/10.1186/s40561-025-00389-y>

11. Гая Фіор. Практичний досвід навчання STEM з використанням цифрових технологій / Гая Фіор, Карло Фонда та Енріке Канесса / Міжнародний центр теоретичної фізики, Трієст, Італія, arXiv:2408.00781v1 [physics.ed-ph], 17 липня 2024 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/html/2408.00781v1>

12. Андре Меноллі. Освітні висновки з коду: Відображення навчальних проблем в об'єктно-орієнтованому програмуванні за допомогою доказів на основі коду. / Андре Меноллі, Бруно Стрік / arXiv:2507.17743v1 [cs.SE] 23 липня 2025 р., [Електронний ресурс]. – режим доступу: <https://arxiv.org/html/2507.17743v1>.

References

1. Hang Li. Bringing Generative AI to Adaptive Learning in Education. / Hang Li, Tianlong Xu, Chaoli Zhang, Eason Chen, Jing Liang, Xing Fan, Haoyang Li, Jiliang Tang, Qingsong Wen / Cornell University, Komp'yutery` ta suspil'stvo (cs.CY) ; arXiv:2402.14601v3 [cs.CY] 28 chervnya 2024 r.,, [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2402.14601v3>

2. Alex Gu. Challenges and Paths Towards AI for Software Engineering. / Alex Gu, Naman Jain, Wen-Ding Li, Manish Shetty, Yijia Shao, Ziyang Li, Diyi Yang, Kevin Ellis, Koushik Sen, Armando Solar-Lezama / Computer Science > Software Engineering : arXiv:2503.22625 [cs.SE] 28 March 2025 p., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/abs/2503.22625v1>

3. Zhendong Chu LLM Agents for Education: Advances and Applications. / Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S. Yu, Qingsong Wen/ Cornell University, Komp'yutery` ta suspil'stvo (cs.CY) : arXiv:2503.11733v1 [cs.CY] 14 bereznaya 2025 r., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2503.11733v1>

4. Ligu Chen Survey on Evaluating Large Language Models in Code Generation. / Ligu Chen, Qi Guo, Hongrui Jia, Zhengran Zeng, Xin Wang, Yijiang Xu, Jian Wu, Yidong Wang, Qing Gao, Jindong Wang, Wei Ye, Shikun Zhang / Zhurnal GBKsong z komp'yuterny`x nauk i texnologij: Instrukciya dlya avtoriv. ZhURNAL KOMP'YUTERNY`X NAUK I TEXNOLOGIJ arXiv:2408.16498v1 [cs.SE] 29 serpnaya 2024 r. ., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2408.16498v1>

5. Yibo Vang ProjectTest: A Project-level LLM Unit Test Generation Benchmark. / Yibo Vang, Congying Xia, Wenting Zhao, Jiangshu Du, Chunyu Miao, Zhongfen Deng, Philip S. Yu, Chen Xing / University of Illinois Chicago, Salesforce Research, Scale AI Robota, vy`konana do Amazon arXiv:2502.06556v3 [cs.SE] 19 lyutogo 2025 r., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2502.06556v3>

6. Lus E. Ranking of problems and solutions in the teaching and learning of object-oriented programming. / Lus E. Gut`yerres, Karlos A. Gerrero, Ektor A. Lopes-

Ospina / Osvitni tehnologiyi (Dordr), PMC, 8 lyutogo 2022 r.; [Elektronny`j resurs]. – rezhy`m dostupu: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8821851/>

7. Abid Haleema. Understanding the role of digital technologies in education: A review. / Abid Haleema , Mohd Javaida,* , Mohd Asim Qadri b , Rajiv Sumanc / Contents lists available at ScienceDirect Sustainable Operations and Computers journal homepage: <http://www.keaipublishing.com/en/journals/sustainable-operations-and-computers/> [Електронний ресурс]. – режим доступу: <https://www.sciencedirect.com/science/article/pii/S2666412722000137>

8. Kaldaras, L. Vy`kory`stannya avtomaty`chny`x instrumentiv analizu, uzgodzheny`x z progresom navchannya, dlya ocinky` zastosuvannya znan` ta pidtry`mky` navchannya v STEM. / L. Kaldaras, K. Xaudek, J. Krajchy`k / IJ STEM Ed 11 , 57 (2024). <https://doi.org/10.1186/s40594-024-00516-0.>, [Elektronny`j resurs]. – rezhy`m dostupu: <https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-024-00516-0>

9. Lauri Malmi. Computing-specific pedagogies and theoretical models. / Lauri Malmi, Judy Sheard, Claudia Szabo, Päivi Kinnunen / arXiv:2409.12245v1 [cs.CY] 22 serpnia 2024 r., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2409.12245v1>

10. Vil`yegas-Ch. V. Adaptivni intelektual`ni sy`stemy` navchannya dlya STEM-osvity`: analiz vplyvu navchannya ta efekty`vnosti personalizovanogo zvorotnogo zv'yazku. / V. Vil`yegas-Ch, D. Buenano-Fernandes, A.M. Navarro, ta in / Smart Learn. Environ. 12 , 41 (2025). <https://doi.org/10.1186/s40561-025-00389-y> ., [Elektronny`j resurs]. – rezhy`m dostupu: <https://slejournal.springeropen.com/articles/10.1186/s40561-025-00389-y>

11. Gaia Fior. Hands-on STEM Learning Experiences using Digital Technologies. / Gaia Fior, Carlo Fonda and Enrique Canessaa / Mizhnarodny`j centr teorety`chnoyi fizy`ky`, Triyest, Italiya, arXiv:2408.00781v1 [physics.ed-ph] 17 ly`pnia 2024 r., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2408.00781v1>

12. André Menolli. Educational Insights from Code: Mapping Learning Challenges in Object-Oriented Programming through Code-Based Evidence / André Menolli , Bruno Strik / arXiv:2507.17743v1 [cs.SE] 23 ly`pnia 2025 r., [Elektronny`j resurs]. – rezhy`m dostupu: <https://arxiv.org/html/2507.17743v1>

Надійшла до редакції 18.08.2025, розглянута на редколегії 18.08.2025

Automated Design and Evaluation of Learning Trajectories in OOP and Web Technologies for STEM Education

In the context of the dynamic development of STEM fields, traditional one-size-fits-all curricula are becoming less effective. This article is dedicated to the methodology of automated design and evaluation of learning paths, which allows for the personalization of the learning process, adapting it to the individual needs and pace of the student. The main focus of the article is the practical implementation of this approach for disciplines in Object-Oriented Programming (OOP) and Web Technologies.

Modern STEM education is undergoing a period of significant transformations driven by the integration of advanced artificial intelligence (AI) and automated learning systems. This work is dedicated to the analysis of methodologies and tools for the automated design and evaluation of learning trajectories, specialized for teaching

Object-Oriented Programming (OOP) and web technologies within the context of STEM education. Particular attention is paid to the development and implementation of tools based on Python and JavaScript programming languages, which ensure the personalization of the educational process and provide adaptive learning capabilities.

The study explores modern approaches to using Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) to create dynamic learning content that adapts to the individual needs and knowledge levels of students in real-time. It analyzes methodologies for multidimensional evaluation of the effectiveness of learning trajectories, including automated detection of problems in mastering OOP concepts and feedback mechanisms. The research also investigates the architectural principles of such systems, utilizing Python for machine learning algorithms and JavaScript for interactive user interfaces. This combination is shown to enhance learning effectiveness and student motivation compared to traditional teaching methods in STEM disciplines.

This work contributes to the development of the theoretical foundations of adaptive learning and provides practical recommendations for developers of educational technologies and STEM educators to better equip students with the skills necessary for a future career in technology.

Key words: STEM education; personalization; OOP; Web technologies; Python; JavaScript; learning trajectories.

Відомості про автора:

Столяренко Тетяна Леонідівна – кандидат педагогічних наук, старший викладач, Відокремлений структурний підрозділ «Харківський фаховий коледж інформаційних технологій Національного аерокосмічного університету «Харківський авіаційний інститут». Електронна пошта: t_stol@ukr.net, t.stoliarenko@khai.edu, телефон 0975218476, ORCID: 0000-0002-7202-316X.

About the author:

Stoliarenko Tetina – Candidate of Pedagogical Sciences, Senior Lecturer, Autonomous Structural Unit «Kharkiv professional college of information technologies of National aerospace university «Kharkiv aviation institute» (скорочено – ASU «KhPCIT KhAI») Email: t_stol@ukr.net, Phone: 0975218476, ORCID: 0000-0002-7202-316X.